UNIVERSITE D'ALGER

ECOLE NATIONALE POLYTECHNIQUE

département ECONOMIE

6/75
2 ex

المدرسة الوطنية المتعددة التقنيات
المكتبة
ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHÈQUE

THESE DE FIN D'ETUDES

A.R.O.M.E

Algorithme de Recherche d'un Optimum par le

Multiplicateur d'Everett

المدرسة الوطنية للعلوم الهندسية
المكتبة
ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHÈQUE

sujet proposé par:

Mr F.D.ARMINGAUD

étudié par:

.D. FERROUKHI

.M.GUEMIRI

PROMOTION .75.

المدرسة الوطنية للعلوم الهندسية
— المكتبة —
ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHÈQUE

UNIVERSITE D'ALGER

ECOLE NATIONALE POLYTECHNIQUE

département ECONOMIE

THESE DE FIN D'ETUDES

A.R.O.M.E

Algorithme de Recherche d'un Optimum par le

Multiplicateur d'Everett

sujet proposé par:

M. F_D_ARMINGAUD

étudié par:

.D. FERROUKHI

.M_GUEMIRI

PROMOTION _75_

*****REMERCIEMENTS*****
****

NOUS TENONS A REMERCIER TOUS CEUX QUI, D'UNE MANIERE OU D'UNE AUTRE, ONT CONTRIBUE A NOTRE FORMATION, AINSI QU'A CEUX QUI NOUS ONT AIDE DANS L'ELABORATIONDE NOTRE TRAVAIL : EN PARTICULIER

M^r F.D. ARMINGAUD(PROMOTEUR), M^r BOUMAHRAT (CHEF DE DEPARTEMENT ECONOMIE A L'E.N.P.A.)

AINSI QUE MM^r-BEKOUCHA M, EDISSIA, MALOUM D.

******DEDICACES******

****

-JE DEDIE CE PETIT TRAVAIL A MES PARENTS EN PARTICULIER A MON FRERE ALI ET MA MERE KHADIDJA.

-FERROUKHI DJAMAL-

-JE DEDIE CE PETIT TRAVAIL A MES PARENTS , TOUS MES AMIS AINSI QU'A MES CAMARADES.

-GUEMIRI MOUSSA-

-LA SCIENCE N'ATTEINT LA PERFECTION QU'AVEC LES MATHEMATIQUES.

-KARL MARX-

======SOMMAIRE=====

==========

=====§§§§§§§§§=====

-INTODUCTION.

On peut définir l'optimisation comme étant, la meilleure façon de faire ce que l'on doit faire.

-L'optimisation, revêt un intêret certain dans le domaine de la production, de la gestion, du commerce, et bien entendu celui de la politique, pour lesquels un petit écart d'éfficacité peut sighifier la différehce entre le succès et la faillite. L'optimisation a de nos jours acquis une place importante, avec le développement technologique et la complexité des moyens de production.

-Le problème revêt un intêret particulier, dans un pays socialiste, où toutes les activités doivent être planifiées d'une façon scientifique et rationnelle ;pour parer au gaspillage, économique, rentabiliser les forces productives, et utiliser d'une manière optimale les ressources existentes.

-Comme nous pouvons le constater, le champs d'application de l'optimisation est très vaste; dans la totalité des actes de la vie courante, l'homme a l'habitude d'optimiser; le plus souvent cette démarche est faite instinctivement; mais les impératifs de rentabilité du monde moderne ont rendu nécéssaire l'élaboration des différentesthéories de l'optimisation qui permettent d'accomplir rationnallement cette démarche.

-Dans la pratique le problème se pose de la façon suivante:

-Optimiser (maximiser ou minimiser) une fonction F(X) à plusieures variables, cette fonction est appelée objectif ou fonction économique.

-Sous des contraintes $G(X) \leq B$

Les variables peuvent être des facteurs économiques.

Par exemple dans l'industrie, on sera amené à Minimiser les coûts de production, sachant que l'on est affronté à des exigences particulières Minimiser des pertes d'énergies dans un réseau electrifié, Maximiser un revenu .....etc.

..../....

-Pour cela ,il existe des méthodes dites spécialisées telles que:

-Programmation linéaire

- " dynamique

-Méthode du gradiant

- " statistique ...

Notre méthode qui s'appuie et utilise le Multiplicateur d'Everett,est générale du fait qu'elle ne fait aucune hypothèse sur la fonction à optimiser(dérivabilité,continuité,linéarité,....).

-Il est certain que notre méthode est moins performante que les méthodes spécialisées,mais on en fera appel lorsque ces dernières ne seront pas disponibles ou non applicables .

Notre thèse comporte deux parties séparées:

-un support théorique(une synthèse des méthodes de recherche avec des tests pratiques vérifiés sur ordinateur IBM 1130

-une partie destinée à l'utlisateur, qui décrit la marche à suivre pour arriver à l'optimisation et une idée assez approchée du temps de calcul par conséquent du coût d'exécution.

Ces deux parties sont éxécutéss sur deux polycopiés distincts;pour faciliter d'une part la compréhension pour celui qui veut approfondir la recherche mathématique , et d'autre part pour aider dans l'utilisation celui qui sera intéressé par une quelconque application de notre méthode .

BIBLIOGRAPHIE.

- D-J -WILDE (recherche d'un optimum)

- THESE de M^r F.D. AR [illegible] (mutiplicateur d'Everett)

- CONFERENCES DONNEES A L'ENPA (département de génie chimique)

-KOWALIK J (methodes for uncontrainned optimisation).

.../...

CHAPITRE -I-

MECANISME D'EVERETT

§§§§§§§§§§§§§§§§§§§§§§§§
§§§
§§§§§§§§§§§§§§§§§§§§§§§§

I-Problème de base

On à maximiser la fonction $F(x1,x2,x3,.....xn)$,

sous les containtes : $G(x1,x2,x3,.....xn) \leq b1$

$Gm(x1,x2,x3,.....xm) \leq bm$

On ecira le problème sous la forme:

$$\begin{matrix} \text{Max } F(\vec{X}) \\ G(\vec{X}) \leq b \end{matrix} \qquad \begin{matrix} \vec{X} \in En \\ \vec{G}, b \in Em \end{matrix}$$

II-Fonction H d'Everett.

Everett introduit la fonction $H(x,y)$ telle que :

$H(x,y)=F(x)-\vec{y}.\vec{G}(x)$ avec $y \in Em$, est appelé le multiplicateur d'Everett .les composantes de ysont positives ou nulles.

Le mécanisme d'Everett set un procédé itératif.il consiste à se donner une valeur quelconque en général à y° et chercher le maximum de $H(x,y)$sur En:

$$\text{Max } H(x)=H(x,y^\circ) \qquad (II)$$

Si F et G sont dérivables,il existe de nombreuses méthodes permétant de résoudre (II).Ce sont les techniques ppe appelées de "descentes".on supposera dans notre étude ,que l'on a aucune information sur les dérivées de F et G. le problème ne sera que plus gé néral.

-une fois ,ayant trouver un x optimal pour un y donné, deux cas peuvent se présenter:

--x vérifie (I), alors x est l'optimum et on a fini la recherche.

--x ne vérifie pas les containtes de (I).

Par conséquent le mécanisme d'Everett consiste à adapter les suites y(k) en fonction de y(k+1) et de x(k) de façon à aboutir àx un x optimal.

-Da ns ce qui suivra ,exposé brièvement les différentes propriétés des fonctions Gi(x) et des Yi . Cet exposé a eté tiré du rapport interne de MONSIEUR F.D. ARMINGAUD,pour le service du développement scientifique I.B.M.,PARIS 1971.

III-Le multiplicateur d'Everett.

IIIa)-Propriétés.

Théorème I. Pour toute valeur de Y dans R+(m) le multiplicateur H, maximise F(x) sous les containtes $G(x) \leq G(x^\circ)$.

Démonstation:

$H(x,y)=F(x)-y.G(x)$

Soit $x^\circ=\max H(x,\hat{y})$ et $x=g(x)=g(x^\circ)-e$

supposons:

$f(x)=f(x^\circ)+d$ alors: $H(x,\hat{y})=f(x^\circ)+d-y.g(x^\circ)+y.e$

$=H(x^\circ,\hat{y})+y.e+ d \quad H(x^\circ,\hat{y})$.

.../...

Ce qui est contraire à l'hypotèse:

IL n'existe pas de x / $f(x) > f(x^\circ)$ et $g(x) \leq g(x^\circ)$.

Par conséquent le problème posé : Max f(x) sous g(x) b serait résolu si nous pouvons résoudre le problème posé par Everett à savoir(Max H(x), trouver y°

$y^\circ \geq 0$/ $g(x^\circ) \leq b$ ---> x°= MaxH(x,y°)).

III-a1/Influence du multiplicateur sur domaine d'optimalité

Théorème 2 :

-les gi(x°) sont des fonctions non croissantes de yi.

Soitx x1= Max H(x,y1) et x2 = Max H(x,y2)

Par définition:

$$H(x1,y1) \geq H(x2,y1) \qquad (1)$$

$$H(x2,y2) \geq H(x1,y2) \qquad (2)$$

(1) +(2) nous donnent :

$$(y2-y1)(g(x2)-g(x1)) \leq 0$$

En particulier si y2= y1+dki avec d> 0

Alrs :

$$gi(x2) \leq gi(x1) .$$

III-a2/Cas des containtes inégalités

Les containtes non saturées n'interviennent pas dans la solution optimale si l'on arrive à déterminer leur rang. Il devient possible de les éliminer et se ramener du problème des containtes saturées.

Théorème 3/

S'il existe un y° tel que :

$g(x^\circ) \leq b$ et $y^\circ.g(x^\circ) = y^\circ.b$ Alors:

-a)les contraintes non saturées à l'optimum correspondent à une composante nulle du multiplicateur.

-b)x° est le Max cherché

Soit x1 / $f(x1) = f(x^\circ) + d$

$$H(x1,y^\circ) = f(x^\circ) + d - y^\circ.g(x1) > f(x^\circ) - y^\circ.g(x^\circ)$$

$$y^\circ(g(x1)-b) \geq d$$

Posons:

B= ($y^\circ = 0$)

Alors :

$\beta.y^\circ.(g(x1)-b > 0$ Avec B = Matrice booléénne telle que :

$$B + \beta \pm I \quad \text{et} \quad B.\beta = 0 .$$

Toutes les composantes de y étant positives,on en déduit :

..../.....

$\forall$/ $g(xi) > b$,Donc $\nexists$ x1 / $f(x1) > f(x^\circ)$. x° =optimum cherché.

III-a3)Correspondance avec le Lagrangien classique

Théorème 4:

Si x° maximise $H(x,y^\circ)$avec $y^\circ.g(x^\circ) = y^\circ.b$ Alors :

$$L(x,y^\circ) \leqslant L(x^\circ,y^\circ) \leqslant L(x^\circ,y) .$$

Conséquences:

— Le théorème 3 ,nous permet de conclure: tout procédé de recherche du mutiplicateur optimal dans le cas des contraintes saturées,pourra être étendu au cas des contraintes non saturées par remplacement du test d'arrêt:

$g(x^\circ) = b$ Par : $y.(g(x) = y.b$

—les théorème 1;2,3,4 ne font appel àaucune hypothèse sur les caractéristiques de f et g(continuité,dérivabilité,convéxité,) et du domaine d'optimalité à la différence du Lagrangien.

III-b/Domaine d'appartenance du multiplicateur optimal.

Soient y1,y2,g(x1),g(x2).

des deux relations:

$$\begin{cases} (y3-y1).(g(x3)-g(x1)) \leqslant 0 \\ (y3-y2).(g(x3)-g(x2)) \leqslant 0 \end{cases}$$

On tire en posant : $y3 = ty1 + (1-t).y2$

$$(1-t).(y2-y1).(g(x3)-g(x1)) \leqslant 0$$
$$-t.(y2-y1).(g(x3)-g(x2)) \leqslant 0$$

En désignant par $\hat{g}$la projection de g(x)sur y2-y1, on aura :

$t < 0$	$\hat{g}3 \leqslant \hat{g}2$
$0 < t < 1$	$\hat{g}2 \leqslant \hat{g}3 \leqslant g1$
$t > 1$	$\hat{g}1 \geqslant \hat{g}3$.

C'est à dire la fonction$(y2-y1).g(x) = (y2-y1).L(x)$ est monotone non croissante sur $\overline{y1.y2}$

III-b1/ Restriction du domaine de recherche I.

-Soit $y' \longrightarrow l(y')$ et $B = (l(y') < b)$ On a donc :

$L(y') - b = (\not\beta - B).W$ Avec $W > 0$

Supposons que $y^\circ = y' + (B - \not\beta).W$

Alors : $(y^\circ - y').(L(y^\circ) - L(y')) \leqslant 0$

Et $(B - \not\beta).y.(L(y^\circ) - L(y')) \leqslant 0$

La relation $\not\beta.yL(y^\circ) \geqslant \not\beta.y.L(y')$est impossible par définition. .../...

et $B.y.(L(y^\circ) - L(y') \leqslant 0$ est impossible lorsque les containtes sont saturées à l'optimum .

Dans ce dernier cas :

$$\forall y \neq 0 \ , \ y^\circ \notin (Bi - Bi).v + yi \text{ avec } v > 0$$

Donc, chaque essai élimine de la recherche un orthant centré sur Y.

III-b2/ Restriction du domaine de recherche II.

Si toutes les contraintes sont saturées en X° alors :

$$(y^k - y^\circ)(G(x^k) - G(x^\circ)) \leq 0 \Rightarrow y^\circ(G(x^k) - b) \geq y^k(G(x^k) - b)$$

La projection de y° sur $G(x^k) - b$ est supérieure à celle de y^k, chaque essai de y^k élimine un 1/2 espace délimité par l'hyperplan d'équation

$$(y^k - y) \cdot (G(x^k) - b) = 0.$$

Remarque :

- Pour les contraintes non saturées, on essayera des algorithmes faisant tendre vers o les composantes de y associés à ces contraintes.

IV. RECHERCHE DU MULTIPLICATEUR OPTIMAL :

4-A)-Algorithmes Utilisant (III.b.2)

$$y^\circ(G(x^k) - b) \geq y^k(G(x^k) - b) \Rightarrow \forall y^k, \quad \exists r \geq 0 \ / :$$

$$\| y^k + r\,(G^k - b) - y^\circ \| \leq \| y^k - y^\circ \|$$

et l'algorithme peut être défini par $y^{k+1} = y^k + r^k \cdot v^k$ où v^k est un vecteur unitaire dans les composantes sont les cosinus directeurs pondérés ou non.

4-B)- Algorithme 2 :

$$v^k = \frac{G^k - b}{\| G^k - b \|} \qquad r^k = r^{k-1} \cdot C$$

$C = 0.5$ si $G^k \leq b$ et $G^{k-1} \cdot w \leq b$

$C = 1$ sinon

La loi d'évolution de 2 a pour but d'assurer un point de fonctionnement au voisinage des contraintes . La distance de y°° a un point saturant les contraintes est au plus = a r

4-C)- Point Stationnaire :

Le seul point stationnaire de l'algorithme est en toute rigueur est le point y° / $l(y^\circ) = b$

Soit : $y^+ / u^+ r.v = y^+$ avec $y^+ \neq y^\circ$

.../...

$\forall i \Rightarrow \left| \frac{r.v^i}{y^i} \right| < p$ = précision des calculs .

$y, r > 0$

$r < \min \frac{y^i}{|v^i|} . p \Rightarrow r < \frac{y^i}{|v^i|} . p \Rightarrow |v^i| < \frac{y^i}{r} . p$ (a)

le vecteur v est normé $\Rightarrow 1 = \sum (v^i)^2 \leq n.\max(v^i)^2 ; \max|v^i| \geq \frac{1}{n^{1/2}}$ (b)

une condition nécessaire pour eviter l'arret de la progresser des y^k sur un point stationnaire non optimal est conséquent d'imposer d'aprés (a) et (b)on a $r > p \sqrt{n} . \max v^i$ ($p \simeq 2^{-23} = 4.10^{-7}$ en 1130, r $r > 10^{-6}$)

* Programmation discréte .

Les contraintes sont pseudo-saturées

W tel que : $G(x^\circ) = b-W$

W vecteur défini positif

Domaine de Recherche :

$(y-y^\circ)\ (G(x)-b) \leq (y^\circ - y)\ W.$

le 2° algorithme s'applique içi si on a :

-1) y doit majorer y°

-2) si $v^k . v \leq 0 \Rightarrow$ revenir à y^{k-1} en diminuant r.

y° est alors approché ; et v doit vérifier (1)

V-) Algorithme d'Everett :

y est modifié de la façon suivante :

$$y^{k+1} = y^k + s^k . v^k$$

$s_i^k = \pm 1$ détermine le sens de variation de y_i

v_i^k = amplitude de y_i

$s_i^k = 1$ si $G_i^k > b_i$

$s_i^k = -1$ sinon

v_i^k sera adapté selon le régime de recherche .

- a .Convergence

si G_i^k meilleur que G_i^{k-1} alors $v_i^{k+1} = v_i^k * C$

pour accélérer la convergence du multiplicateur.

- b . Encadrement

si $b_i \in (G^k, G^{k-1})$

alors $v_i^{k+1} = v_i^k * e$ avec $e < 1$ et $v_i^{k+1} \in$ intervalle d'encadrement.

c- Divergence :

si G_i^{k-1} est meilleur que G_i^k la variation de y_j ($j \neq i$) a contrarié l'effet de y_j (Théorème 2) alors :

$$v_i^{k+1} = v_i^k * d \quad (d = \text{coefficient})$$

V-a) Choix des paramètres :

Everett propose empiriquement C =1.3 , e =0.25 , d=2. (Voir suggestions sur les paramètres dans l'exemple traité) . On optimise la recherche une fois un encadrement obtenu.

V-b) Oscillations :

si y_i^k tend vers zéro les oscillations v_i^k seront amortées .

y_i ne peut revenir en y_i^k si elle ne rencontre pas un encadrement dans le cas le plus défavorable la condition $v_i^{k+j} < v_i^{k-1}$ (y_i a subi j convergence, $e < 1$), nous donne $v_i^{k+j} < v_i^{k-1}$ $e.c^j < 1$

En effet : $\Rightarrow \begin{cases} v_i^{k+j} = v_i^k \cdot e\, c^{j-1} \\ v_i^{k-1} = v_i^k \end{cases} \Rightarrow v_i^k \cdot e\, c^{j-2} < \frac{v_i^k}{c} \Rightarrow e\, c^j < 1$

- Début de recherche . on n'a pas d'idée sur l'encadrement au début de la recherche.Mais on connait une valeur majorante de y_i (Prix duaux,Khen-tucker)

a1= valeur estimée du multiplicateur.

α =Risque de ne pas trouver d'encadrement en K itérations:

$a_i < y_i^k = V^\circ(1+c+.....+c^{k-1}) + y^\circ$ (valeur de départ y°).

C'est à dire:

$$V_i > (c-1)(a_i - y_{i^\circ})/(c^k - 1)$$

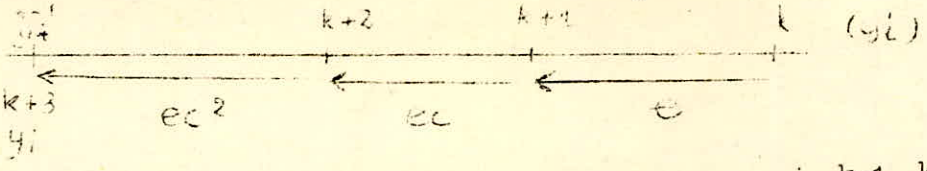
-Fin de recherche. en fin de recherche les y_i sont voisins de leur valeur optimale ,on se limitera àdes convergences et encadrement,en posant;

(I) $e.(c^j - 1)/(c - 1) = 1$ on pert un minimum d'information pour la valeur de j choisie . En effet d'apres A ,on est proche de la va leur optimale , donc très proche de 1.

En particulier :pour j=3 (I) $e.(c - 1).(c^2 + c + 1)/(c - 1) = 1$

Ce qui nous donne: $c^2 + c = \frac{1 - e}{e} = 0$

Les paramètres d'Everett satisfont à cette équation.



- Perte minimale d'information quand on suppose que : $y_i^{k-1} = y_i^{k+3}$.

.../...

VI-Taux de contraction par itération;

VI-a/Taux

Si l'on pose $t.\beta = c^j * e$

$$t = \sqrt[j]{\beta} = c.\sqrt[j]{e}$$

t représente le taux de contractionpar itération .

VII-b/Risque

Si y_i^o se trouve dans l'intervalle d'encadrement;le risque de ne pas l'avoir à nouveau encadré après y-1 itérations est:

$$e.c^{j-1} = \beta / c$$

Si ε est la probabilité estimée d'avoir y_i^j hors de l'intervalle avec c optimal

alors: $$\varepsilon = (1-\varepsilon) \cdot \beta / c \cdot$$

Alors deux remarques peuvent être faites:

a-) j peut être choisi indépendamment de la valeur présumée de ε.

b-)choisir β et c revient à donner à la valeur implicite:

$$\varepsilon = \beta / c + \beta$$

Le taux de contraction sera: $$t = \sqrt[j]{\frac{\varepsilon - c}{1 - \varepsilon}}$$

c-)Si comme on l'a présumé ε diminue en fin de recherche,on pourra augmenter la vitesse de convergence en estimantau fur et à mesure sa valeur à l'aide de la formule précédente, et ce d'autant plus que j est grand.

d-)c et ε fixés , la convergence est d'autant meilleure que j est faible.

Application.

Pour j = 2

$$e.(1 + c)= 1 \text{ ou :}$$
$$e.(c^2 - 1) = c - 1$$
$$e.c^2 - c + (1 - e) = 0$$
$$\text{Delta} = + 4.e^2 - 4.e + 1$$

Racine unique:

$$e = 0.5$$
$$c = 1.$$

Le risque aassocié est de 0.33 et $\beta = 0.5$ avec $t = 0.702$

Voire exemple traité.

...../....

VII-Exemple d'application du multiplicateur d'Everett.

Soit à maximiser, la fonction $F(x1,x2) = x1 + x2$ sous les containtes suivantes:

$$\begin{cases} \frac{x1^2}{9} + \frac{x2^2}{64} \leq 1 \\ \frac{x1^2}{49} + \frac{x2^2}{25} \leq 1 \end{cases}$$

Au départ Y est :

$$\begin{cases} y1 = 0,7 = y1^\circ \\ y2 = 0,4 = y2^\circ \end{cases}$$

Le cacul donne:

$$NV1 = \frac{G1(X) - 1}{\|G\|} = 0,6551$$

$\|G\|$: norme euclidienne

$$NV2 = \frac{G2(X) - 1}{\|G\|} = 0,75551$$

On en conclut que X ne vérifie pas les contraintes

$$g_1(X) - 1 > 0$$
$$g_2(X) - 1 > 0$$

Si $S_i = +1$ Alors:

$$y1 = \begin{cases} y_1^1 = y_{10} + NV1 * 1 \\ y1 = y20 + NV2 * 1 \end{cases} \qquad R^1 = R^* * 1$$

Et : $X2 = \begin{cases} 3,2 \\ 6,9 \end{cases} \Rightarrow NV1 > 0 , \quad NV2 > 0$

On refait les mêmes calculs que précédemment. On trouve:

$$X3 = \begin{cases} 2,0875 \\ 4,41118 \end{cases} \quad NV1 < 0 \quad , NV2 < 0$$

Par conséquent X3 vérifie les contraintes avec $R^3 = R^2 * 0.5$

Remarque:

G^4 est moins bon que G^3; donc on revient vers y^3 , l'intervalle d'encadrement est à ce niveau:

$$1.8780 \leq Y_1 \leq 2.1396$$
$$1.3495 \leq Y_2 \leq 1.7755$$

On continue le processus j'usqu'à ce que l'on est:

$$\|\vec{g}(x) - \vec{b}\| \leq 0,001$$

Ce qui donne:

...../.....

$$\overrightarrow{Y^\circ} = \begin{cases} y_1 = 2.1093 \\ y^\circ_2 = 1.4876 \end{cases}$$

$$\overrightarrow{X^\circ} = \begin{cases} x^\circ_1 = 2.4367 \\ x^\circ_2 = 4.6960 \end{cases}$$

Conclusions:

D'aprées les résultats obtenus à l'aide de l'exemple traité, deux remarques peuvent être faites :

a-) X^2 ne safisfait pas les contraintes , donc R est multiplié par le coéfficiant c = 1.

NV1 passe par la valeur 0,7845 à la valeur -0,5233

NV2 passe de la valeur 0,6200 à la valeur -0851

Everett propose alors de multiplier R par 0.5, coéfficient qui est trops grand dans notre cas car on obtient:

NV1 = 0,7371

NV2 = 0,6757

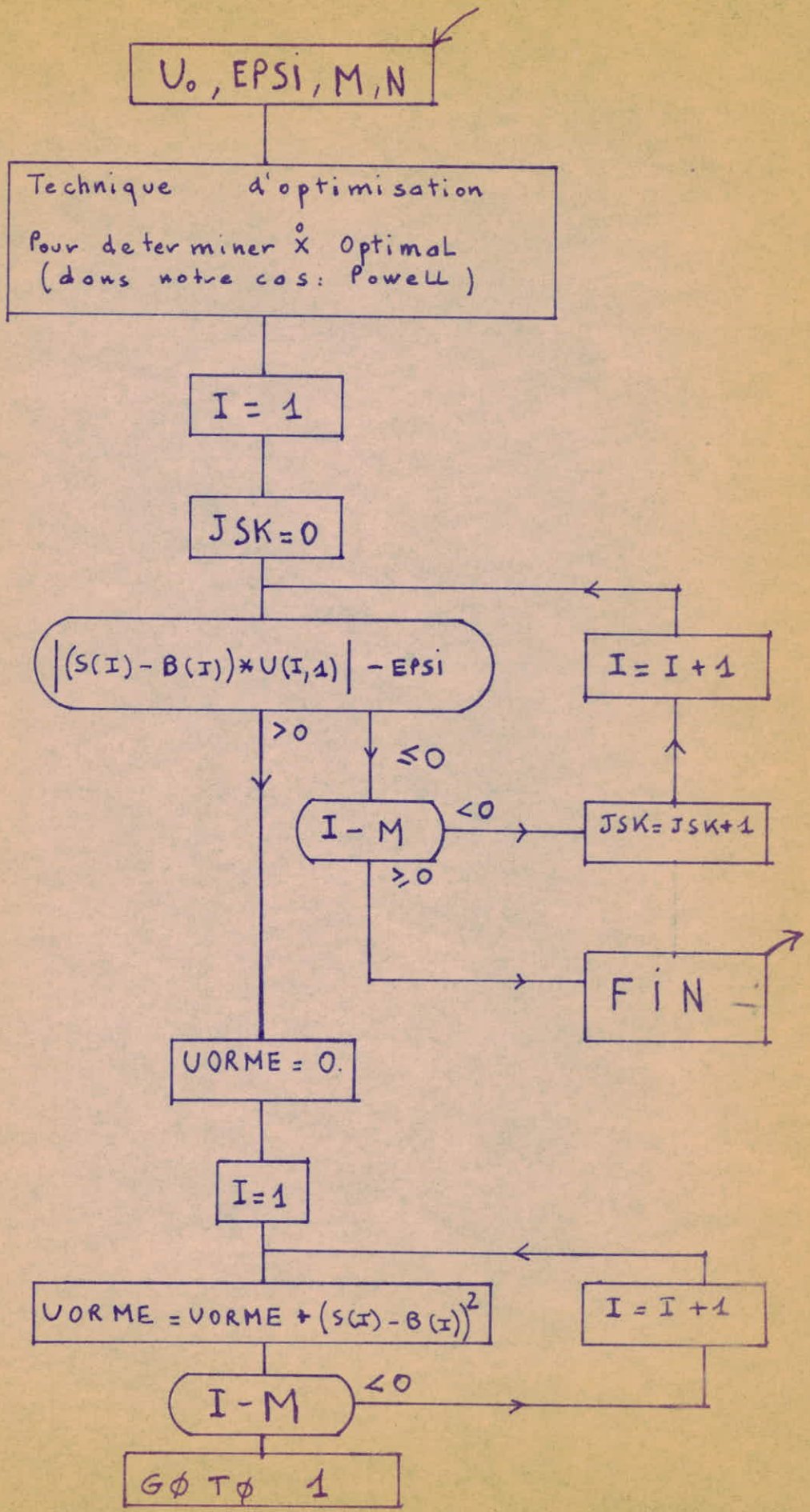
C'est à dire qu'on sort du domaine des contraintes.

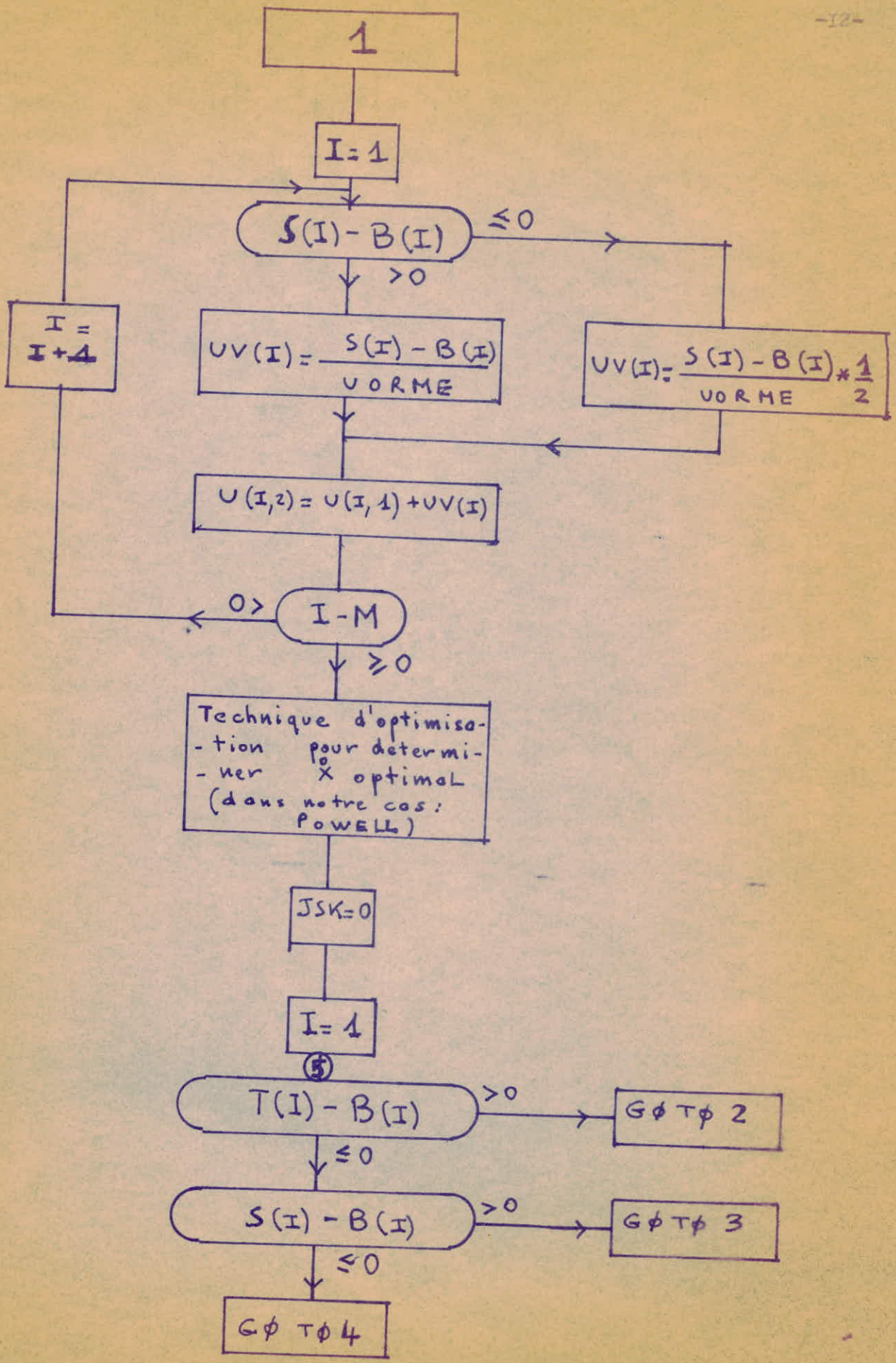
Donc il aurait été préférable de donner à c une valeur plus petite pour avoir plus de chance de rester dans le domaine.

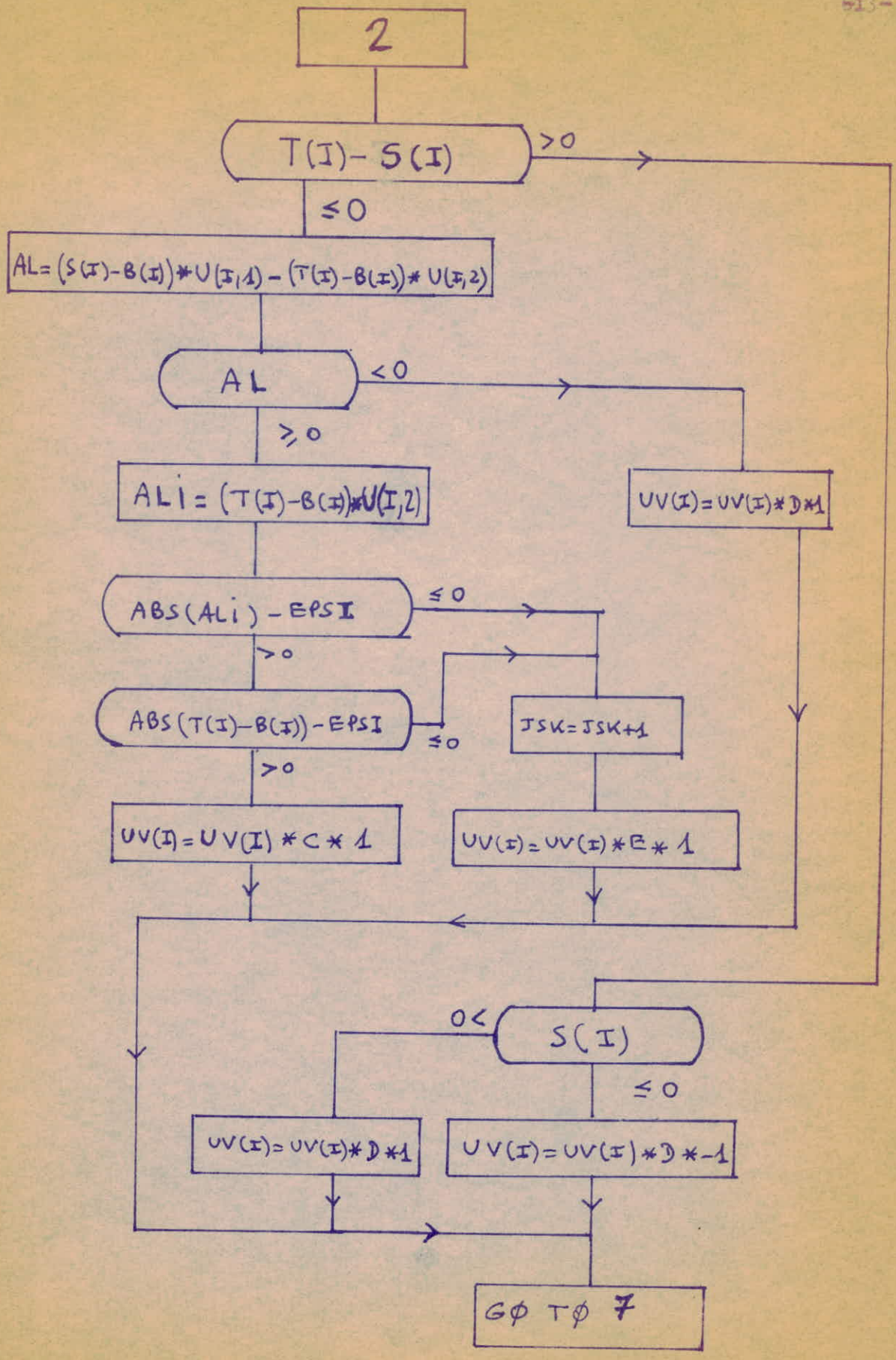
D'une manière générale , il faudrait adapter les paramètres pour chaque cas que l'on traite.

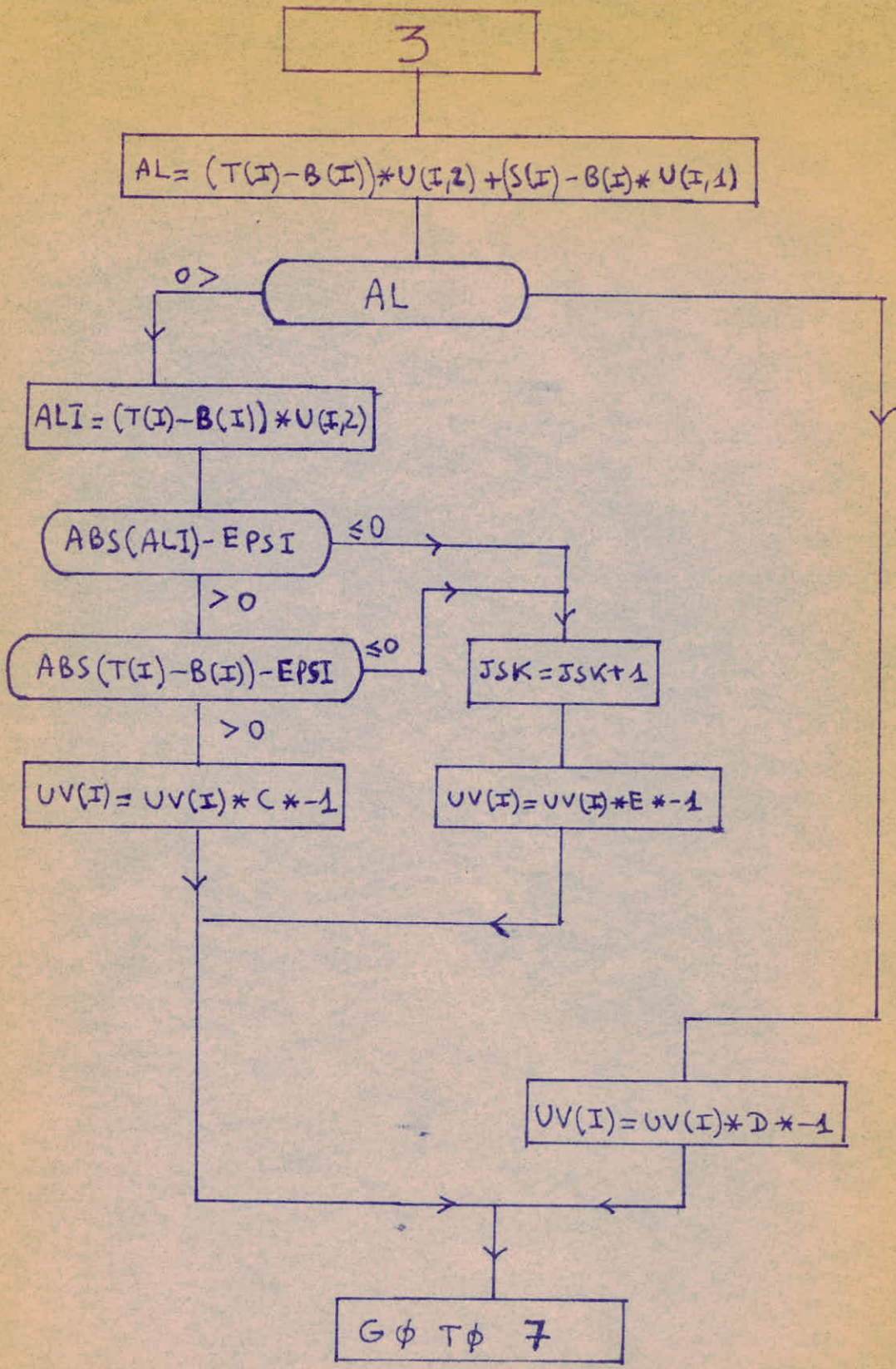
b-) Il n'est pas nécéssaire de faire intervenir $S_i = \pm 1$, car la valeur de G(X) - b inclut justement ce signe .

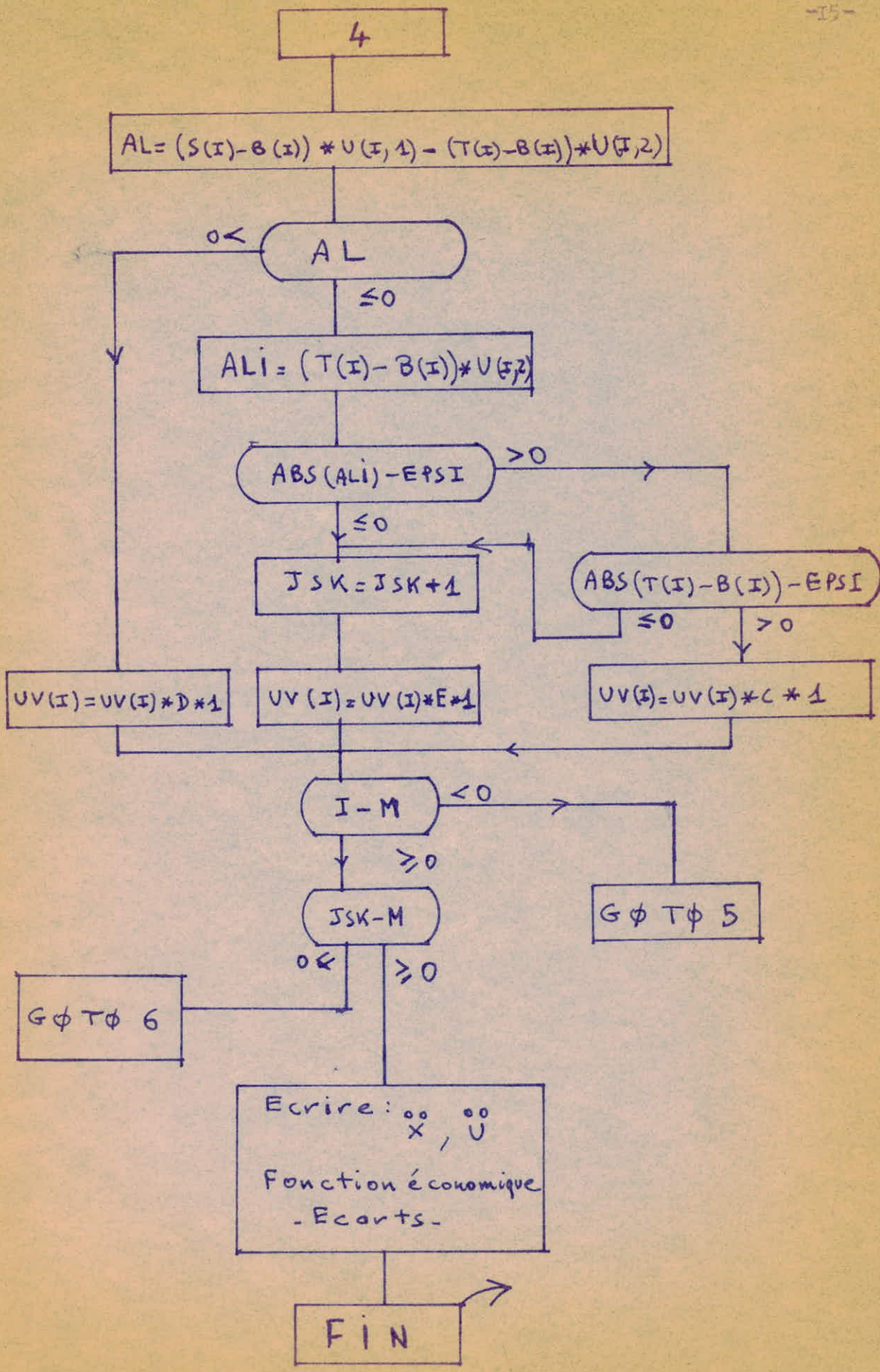
(Voir dans ce suit ,courbes et programme .)

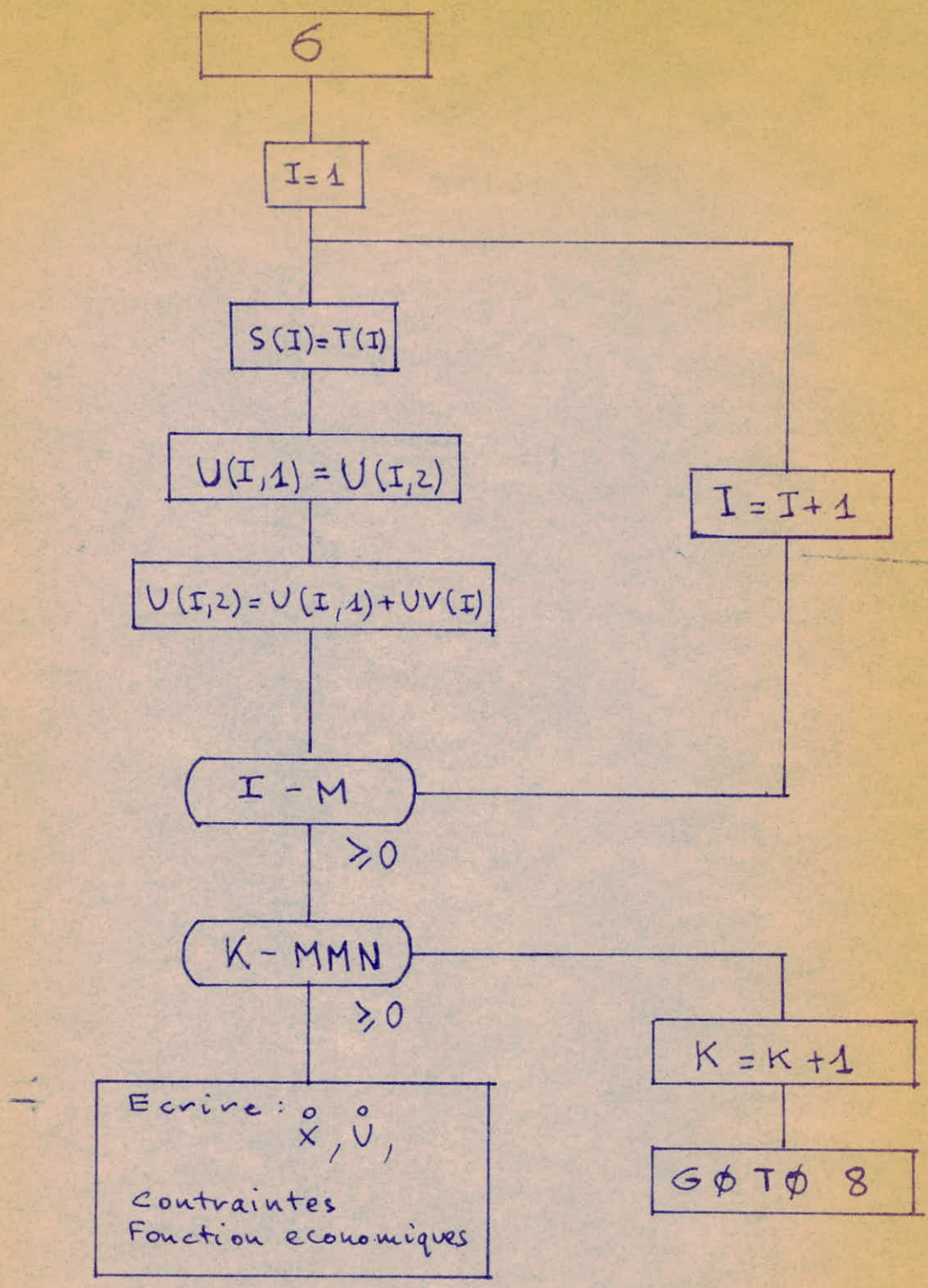

U_o, EPSI, M, N
Technique d'optimisation
Pour determiner $\overset{\circ}{x}$ Optimal
(dans notre cas: Powell)
I = 1
JSK = 0
$|(S(I) - B(I)) * U(I,1)| - EPSI$
I = I + 1
>0
≤0
I - M
<0
JSK = JSK + 1
≥0
FIN
UORME = 0.
I = 1
$UORME = UORME + (S(I) - B(I))^2$
I = I + 1
I - M
<0
GØ TØ 1


1
I=1
S(I)-B(I)
≤0
>0
I=I+1
UV(I)=S(I)-B(I)/UORME
UV(I)=S(I)-B(I)*1/2/UORME
U(I,2)=U(I,1)+UV(I)
0>
I-M
≥0
Technique d'optimisation pour determiner X optimal (dans notre cas: POWELL)
JSK=0
I=1
5
T(I)-B(I)
>0
GØ TØ 2
≤0
S(I)-B(I)
>0
GØ TØ 3
≤0
GØ TØ 4


2
T(I)-S(I)
>0
≤0
AL=(S(I)-B(I))*U(I,1)-(T(I)-B(I))*U(I,2)
AL
<0
≥0
ALi=(T(I)-B(I))*U(I,2)
UV(I)=UV(I)*D*1
ABS(ALi)-EPSI
≤0
>0
ABS(T(I)-B(I))-EPSI
≤0
JSK=JSK+1
>0
UV(I)=UV(I)*C*1
UV(I)=UV(I)*E*1
0<
S(I)
≤0
UV(I)=UV(I)*D*1
UV(I)=UV(I)*D*-1
GØ TØ 7


3
AL=(T(I)-B(I))*U(I,2)+(S(I)-B(I)*U(I,1)
AL
0>
ALI=(T(I)-B(I))*U(I,2)
ABS(ALI)-EPSI
≤0
>0
ABS(T(I)-B(I))-EPSI
≤0
JSK=JSK+1
>0
UV(I)=UV(I)*C*-1
UV(I)=UV(I)*E*-1
UV(I)=UV(I)*D*-1
GϕTϕ 7


4
AL= (S(I)-B(I)) * U(I,1) - (T(I)-B(I))*U(I,2)
0<
AL
≤0
ALI = (T(I)-B(I))*U(I,2)
ABS(ALI)-EPSI
>0
≤0
JSK=JSK+1
ABS(T(I)-B(I))-EPSI
≤0
>0
UV(I)=UV(I)*D*1
UV(I)=UV(I)*E*1
UV(I)=UV(I)*C*1
I-M
<0
≥0
JSK-M
GϕTϕ 5
0<
≥0
GϕTϕ 6
Ecrire : °° °°
X , U
Fonction économique
-Ecarts-
FIN

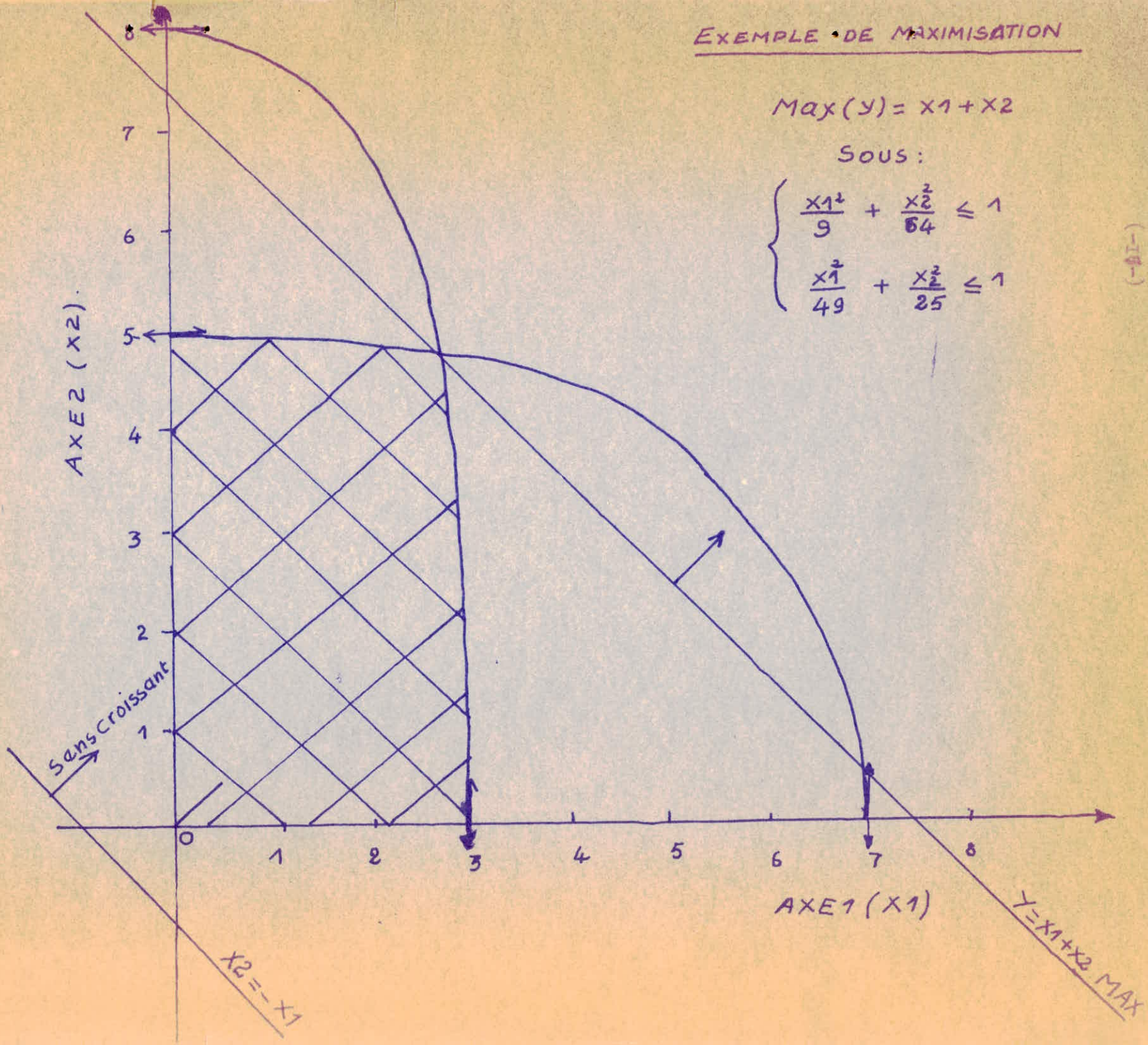

6
I=1
S(I)=T(I)
U(I,1)=U(I,2)
U(I,2)=U(I,1)+UV(I)
I=I+1
I-M
≥0
K-MMN
≥0
K=K+1
GØTØ 8
Ecrire : X°, U°,
Contraintes
Fonction economiques

EXEMPLE DE MAXIMISATION

$$Max(y) = x1 + x2$$

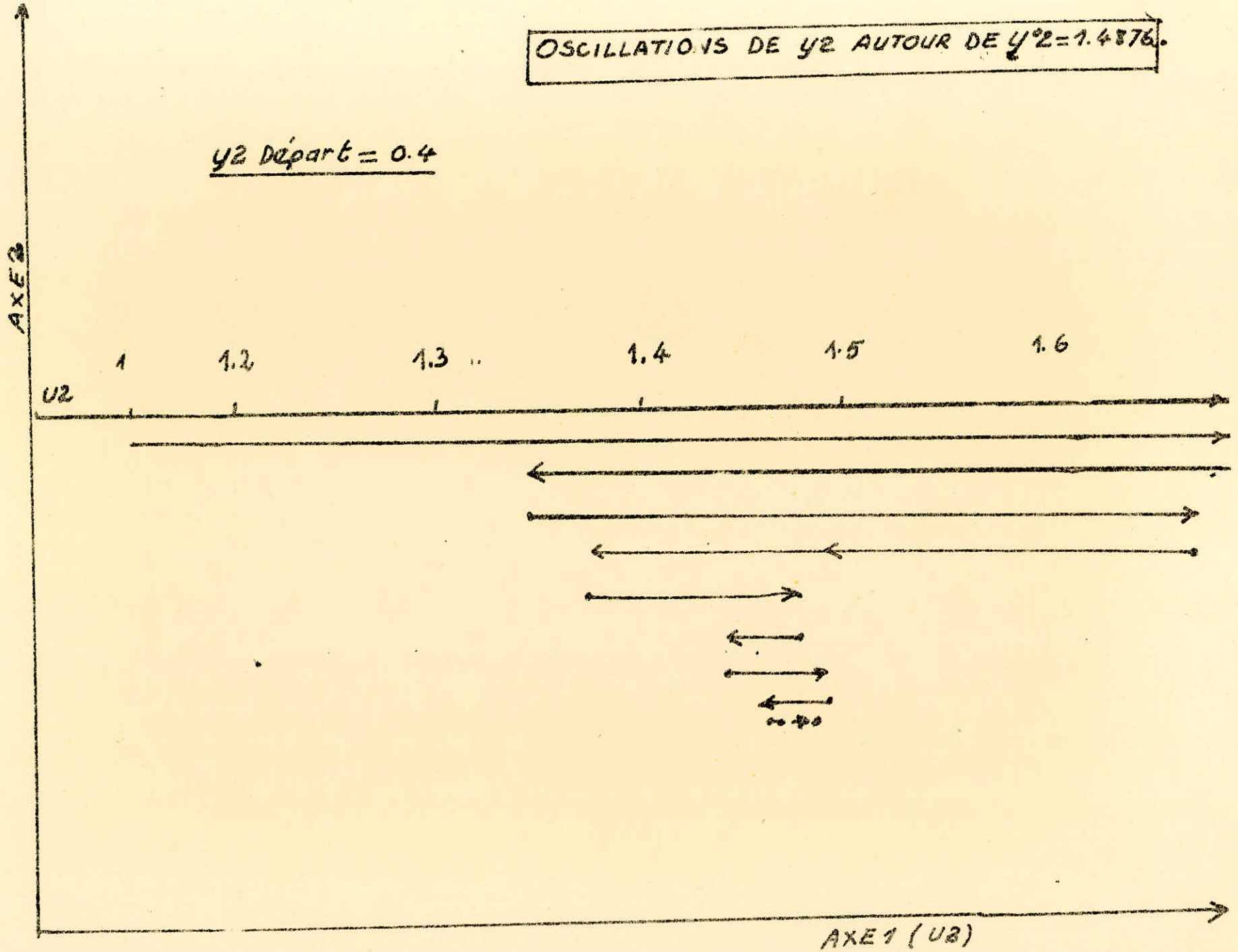
SOUS :

$$\begin{cases} \frac{x1^2}{9} + \frac{x2^2}{64} \leq 1 \\ \frac{x1^2}{49} + \frac{x2^2}{25} \leq 1 \end{cases}$$



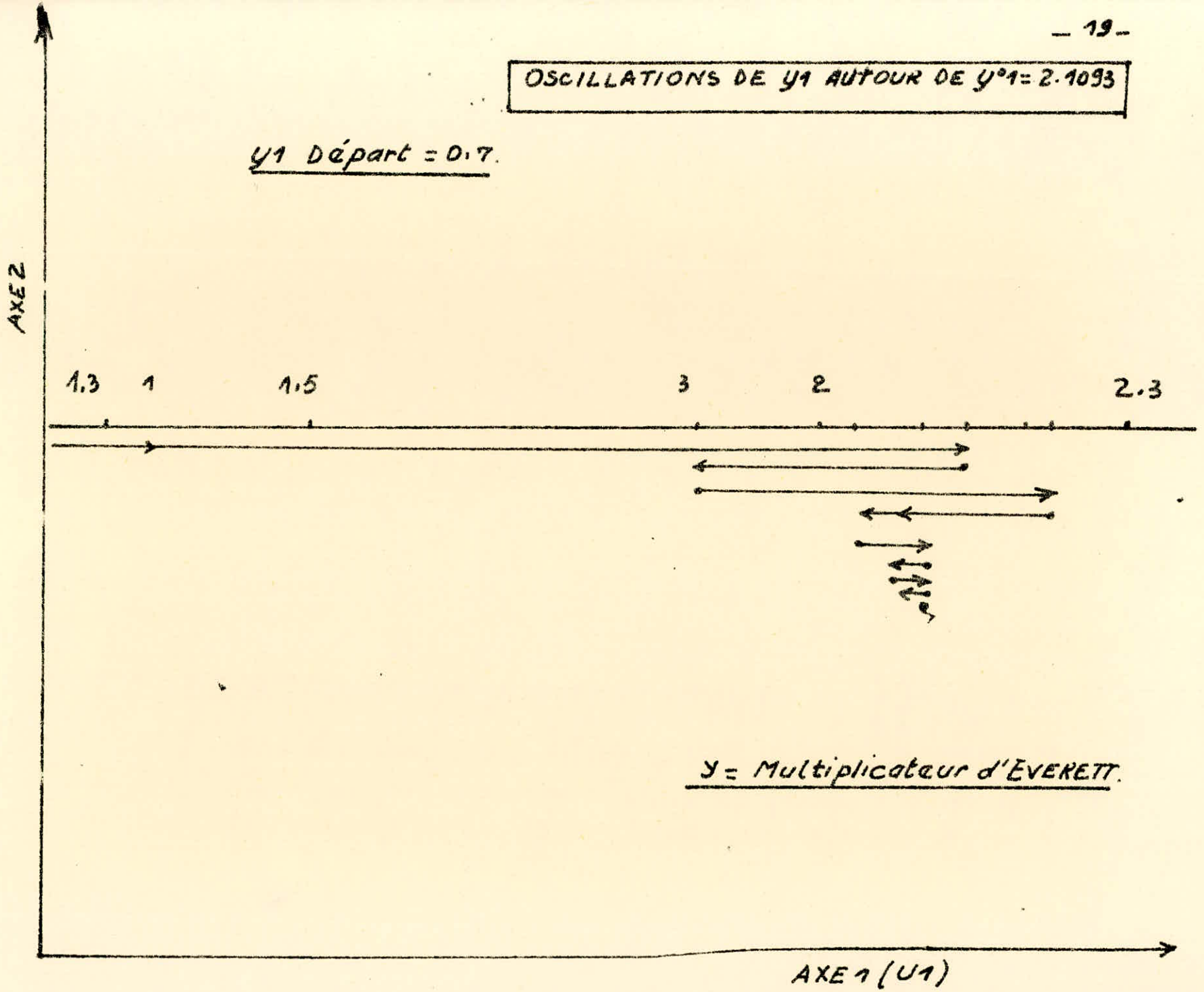
OSCILLATIONS DE y2 AUTOUR DE $y^{*}2 = 1.4876$.

y2 Départ = 0.4



OSCILLATIONS DE y_1 AUTOUR DE $y^*_1 = 2.1093$

y_1 Départ = 0.7



y = Multiplicateur d'EVERETT.

- TEST DU PROGRAMME D'EVERETT.

Exemple : Maximiser. $F(X_1, X_2) = X1 + X2$.

Sous :

$$\begin{cases} \frac{X_1^2}{9} + \frac{X_2^2}{64} \leq 1 \\ \frac{X_1^2}{49} + \frac{X_2^2}{25} \leq 1 \end{cases}$$

X1	X2	Y1	Y2	R	NV1	NV2	NORME.
8.6135	14.5986	1.3591	1.1555	1.0000	0.6554	0.7555	13.7434
3.2041	5.9194	2.1196	1.7755	1.0000	0.7845	0.6200	1.4336
2.0750	4.4118	1.8780	1.3495	0.5000	-0.5233	-0.8521	0.2553
2.6557	5.1971	2.2465	1.6874	0.5000	0.7371	0.6757	0.3043
2.1428	4.3018	2.0871	1.4948	0.2500	-0.6375	-0.7704	0.2604
2.3959	4.6797	2.0466	1.3765	0.1250	-0.3241	-0.9459	0.0211
2.5678	4.8967	2.1194	1.4781	0.1250	0.5824	0.8128	0.1207
2.4097	4.6850	2.0721	1.4373	0.0625	-0.7569	-0.6535	0.0292
2.4753	4.7463	2.1111	1.4861	0.0625	0.6233	0.7819	0.0419
2.4013	4.6439	2.0904	1.4627	0.0312	-0.6611	-0.7502	0.0297
2.4367	4.6960	2.1093	1.4876	0.0312	0.6037	0.7971	0.0054

Valeurs Initiales :

$$\begin{cases} y_{10} = 0.7 \\ y_{20} = 0.4 \\ R = 1. \end{cases}$$

L'optimum (Max) est obtenu pour :

$$\begin{cases} X1 = 2.4367 \\ X2 = 4.6960 \\ Y1 = 2.1093 \\ Y2 = 1.4876 \end{cases}$$

CHAPITRE-I-

METHODE DE FLETCHER -POWELL

I-ORIGINE DE LA METHODE FLETCHER -POWELL

II-METHODE DE FLE-POWELL

- II-1-Hypothese sur la fonction à optimiser
- II-2-Directions Conjuguèes
  - II-2-1-Exemple
  - II-2-2-Theoreme
- II-3- Description de la methode
- II-4- Algorithme

III- AMELIORATION DE L'ALGORITHME DE FLETCHER-POWELL

- III-1- Demonstration
- III-2- Lemme -1
- III-3- Lemme- 2
- III-4- Corollaire
- III-5- Remarque
- III-6- Conclusion
- III-7- Algorithme de Fle-Powell Modifiè

IV- EXEMPLE NUMERIQUE DE PROCEDURE DE FLE-POWELL

V- EXEMPLE GRAPHIQUE DE LA PROCEDURE

VI- QUELQES REMARQUES SUR LA METHODE DE FLE-POWELL

- VI-1-Avantages
- VI-2-inconvenients

VII- ORGANIGRAMME ET TESTS

METHODE DE FLETCHER POWELL

I- ORIGINE DE LA METHODE

On rappelle que le probleme de maximisation de $f(\vec{x})$ sous les contraintes :

$$G_i(x) \leq B_i \qquad i=1,2,.....,m$$

a été transformé en probleme de:

$$\text{Maximisation : } H(\vec{x},\vec{y})=f(\vec{x})-\vec{y}.\vec{G}(x) \qquad y \quad E_M \qquad (I)$$

Pour une valeur donnée $\vec{y}_o$ de $\vec{y}$, il existe de nombreuses permettant de resoudre (I)

ON citera en particulier :

- -La methode de plus grande pente
- -La methode de MONTE-CARLO
- -Etc .....

La premiere methode est basée sur le Gradient de H(x,y), vecteur noté: $\vec{\nabla H}$

et dont les composantes les derivées partielles de H par rapport à chacune des variables x_i i=1 ,2,.....,N

$$\vec{\nabla}_H \begin{cases} \partial H/\partial x_1 \\ \vdots \\ \partial H/\partial x_i \\ \vdots \\ \partial H/\partial x_n \end{cases}$$

Lors de l'explication du mecanisme d'Everett , il n'a été , à aucun moment , fait appel à la notion de derivée.

On essaydra de resoudre (I) sans faire appel à la notion de derivée. En effet ,meme dans le cas ou la fonction H est connue, le calcul des dérivées partielles de H est assez delicat;d'autre part , une fonction Revenue ou Cout ,est connue en des points de fonctionnement, mais on ignore sa forme Analytique.PAR conséqu--ent cette premiere methode ne se prete guere a la resolution de (I).

Quant a la seconde méthode , elle est difficilement maitrisable q le nombre de variables est assez grand.

Donc ,arrivés à l'etape de maximisation de H , on pourra employer la methode de HOOKE-JEEVES ou celle de ROSENBROCK.ON emploiera la " FLETCHER-POWELL , qui est une methode recente et plus generale que les deux methodes citées ci-dessus .

II- METHODE DE FLE-POWELL

II-1- Hypothèse Importante:

On suppose que la fonction H peut etre etroitement approximée au voisinage du maximum par une forme quadratique definie negative. (ou positive s'il s'agit d'un minimum)

Cette hypothèse est aussi valable pour les methodes de plus grande pente. En fait, il existe presque toujours une quadrique osculatrice au voisinage de l'optimum.

II-2-Directions conjuguées: (D-C)

Deux directions sont dites conjuguées, U et V ,en respectant la matrice A definie positive si:

$$U^T.A.V = 0 \qquad (U^T = \text{transposée de } U)$$

Il est possible de generer une suite de directions conjuguées par un processus analogue à celui de GRAM-SCHMIDT de procédure d'orthogonalisation . (voir D-C)

II-2-1-Exemple: supposons que l'on démarre avec un vecteur d1 ; $d2=A.d1-b11.d1$ si :

$$b11=\frac{d1^T.A.d1}{d1^T.A.d1} \qquad \text{et}$$

$d3=A.d2 -b22.d2- b21.d1$ si :

$$b22=\frac{d2^T.A^2.d2}{d2^T.A.d2} \qquad \text{et} \qquad b21=\frac{d1^T.A^2.d2}{d1^T.A.d1}$$

.../...

D'une manière générale :

$$di + 1 = A.di - \frac{di^T . A2\ di}{di^T . Adi}\ di - \frac{di^T . Adi}{di^T_{-1} . Adi_{-1}} - di_{-1}$$

et pour j = 1,2 - - - - -, i on a.

$$di + 1 = A\ di - \sum_{s=1}^{i} bis . d_s$$

avec bis = 0 si S (i - 1.

Si les vecteurs D_1 , d_2, - - -, dn, sont mutuellement conjugués, ils sont linéairement indépendant.

La signification des directions conjuguées est la suivante :

2.1. THEOREME.

Si D_1 , d_2 , ... dn est une suite de vecteurs mutuellement conjugués en respectant la matrice positive définie A , alors le minimum de la quadrique :

$$F = a + b^T x + \frac{1}{2}\ x^T . AX$$

peut être atteint à partir d'un point x^o de départ arbitraire en un nombre fini de decentes dans lesquelles chaque vecteur di est utilisé comme direction de descente à la fois - l'ordre dans lequel les di sont utilisés est indiférent.

Démonstration Voire (2-6) - page 40.

.../...

.../...

3°. DESCRIPTION DE LA METHODE DE POWELL.

Le principe de la méthode est basé sur le fait que si 2 minimums sont trouvés à partir de panits différents suivant une direction donnée $\overrightarrow{V}$, alors le vecteur joignant ces 2 minimums est conjugué à $\overrightarrow{V}$ - (voire figure 3 - a).

Soit 2 minimums x1 et x2 obtenus en partant de 2 points différents et dans la même direction de $\overrightarrow{V}$.

Si x1 et x2 sont les minima, alors :

Le vecteur joignant 2 minima trouvés dans la direction de V est conjugué à $\overrightarrow{V}$.

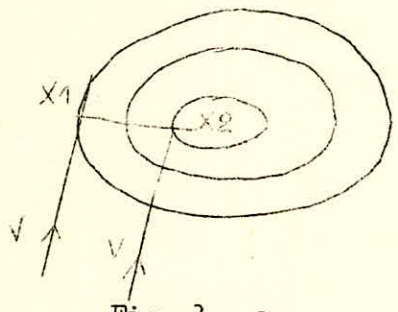


Fig. 3 - a.

Conditions nécessaires d'optimalité de X1 et X2.

$$\left.\begin{array}{l} V^T \cdot (AX1 + b) = 0 \\ V^T \cdot (AX2 + b) = 0 \end{array}\right\} \Longrightarrow$$

$$\boxed{V^T A (X1 - X2) = 0}$$

est donc le vecteur figurant les minimas suivant un vecteur $\overrightarrow{V}$ est conjugué à $\overrightarrow{V}$

.../...

4°. ALGORITHME DE FLETCHER-POWELL.

L'Algorithme de Powell utilise le résultat suivant : Si la recherche pour chaque minimum est faite le long de plusieurs directions conjuguées, alors si l'on joint ces deux minimas, le vecteur faisant le point est conjugué à toutes ces p directions.

Ceci étant un résultat du théorème 21 = En effet, le résultat de chaque recherche dépend seulement des points de départ et pas de l'ordre dans lequel chaque de " descente " est réalisé. Par conséquent, ce résultat peut être exploité et donnera l'algorithme d'optimisation suivant :

Voir exemple dans le cas de 2 dimensions.

4 - A.

Algorithme : Supposons qu'il existe . vecteurs indépendants
D1, D2, ... d initiaux

A) Choisir $\lambda 0$ qui minimise $F (X^\circ + \lambda dn)$ et poser

$$X^1 = X^\circ + \lambda 0 \; dn$$

B) pour i = 1, 2, ..., calculer λI qui minimise $F (X^i + \lambda_i di)$ et poser

$$X^{i+1} = X^i + \lambda i \; di$$

C) Prendre di = di + 1 pour i = 1, ..., n - 1.

D) Prendre $dn = X^{n+1} - X^i$ comme nouvelle direction à la place de dn et comme nouveau point de départ $X = X^{n+1}$

$$X^\circ = X^{n+1}$$

E) Revenir en A.

La procédure s'arrête lorsqu'il n'y a plus moyen de minimiser davantage la fonction.
En effet, si l'algorithme a été exécuté K fois, et que la fonction objectif a diminué constamment au cours de ces K itération et si la (K + 1,) une exécution de l'algorithme, on ne peut plus améliorer la fonction, alors l'optimum atteint est celui de la $K^{ième}$ itération.

A la fin du 1er cycle d'itération $-X^{n+1}-X1$ est conjugué à
d_n - fin du 2ème cycle d'itération $X^{n+1}-X1$ " " à
dn -1 etc.....
Après (-1) cycles d'itérations, n directions conjuguées sont obtenues.

A la fin d'un cycle d'itérations , on utilise un critère permettant d'estimer si la nouvelle direction générée est efficace ou non.- On rejette cette direction générée si elle ne satisfait pas le critère et on répète un nouveau cycle en utilisant les anciens vecteurs conjugués. Dans le cas contraire, on accepte la nouvelle direction qu'on pose à la place d'un des vecteurs utilisés à cette étape. Lequel des vecteurs sera remplacé ?

LEMME 1. X1, X2 , ...Xn sont des vecteurs tels que :
$Xi^T . A . Xi-1, Vj (X) = Xj$. Alors le déterminant de X est maximum quand les vecteurs Xj sont mutuellement conjugués en respectant A.

(Démonstration - Voir - D.C -).

Ce lemme procure le critère de Powell suivant :
On teste la nouvelle direction pour voir si'il est possible de l'ajouter à la place de l'une des directions conjuguées utilisées à cette étape. Pour cela il faut que la matrice obtenue après telle substitution ait un déterminant non décroissant. Evidemment, ceci étant fait en ne faisant appel qu'aux valeurs de la fonction. Pour cela, on aura besoin de :

LEMME 2.

Si Xi est un vecteur scolaire tel que $Xi^T . A . Xi = 1$,
Alors , la composante du vecteur $(X^{n+1} - X^1)$ suivant la direction de Xi est :

$$\lambda i^2 = 2 (F (xi) - F (x^{i+1}))$$

COROLLAIRE.

Si $n^T . A . n = 1$, alors le minimum de la fonction F suivant la direction n à partir d'un point y est à :

$$x = y + \sqrt{2 (F (y) - F (x))} . n \qquad (A)$$

(Démonstration - voir - (D C))

Donc on doit remplacer le dp par d (nouvelle direction conjuguée obtenue à cette étape) pour lequel la quantité.

$$\sqrt{2 (F (x^p) - F (x^{p+1}))} \qquad p = 1, ..., n$$

est maximale.

Ceci nous assurera que le déterminant de la matrice des directions nouvelles est le plus grand d'après la relation (A).

En effet : $x^{n+1} - x^1 = \lambda d$ où $d^T A d = 1$

(B) deter ($\xi_1, \xi_2, \ldots, \xi_{p-1}, d, \xi_{p+1}, \ldots \xi_n$) = $\frac{\alpha}{\lambda}$ det ($\xi_1 \ldots \xi_n$)

.../...

Le r sultat de B permétra de nous indiquer si le déterminant de la matrice des coefficients des directions a augmenté ou non de l'incorporation de la nouvelle direction .

-Démonstration .

Posons : $X^{n+I} - X^{I} = \mu . d$

où $d^T . A . d = I$

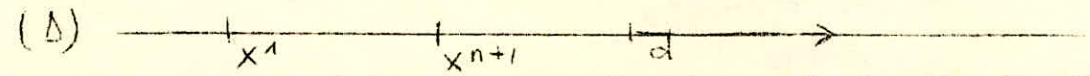
$$\text{Deter}(\varepsilon_1, \varepsilon_2, \ldots \varepsilon_{p-1}, d, \varepsilon_p, \ldots \varepsilon_n) = \frac{\alpha_p}{\mu} \text{Deter}(\varepsilon_1, \ldots \varepsilon_p, \ldots \varepsilon_n)$$

On remplacera ε_p par d pour lequel la quantité /

$\sqrt{2.(F(X^p) - F(X^{p+I}))}$ est maximale .

Pour calculer désignons par $X^{n+I} + Y¥.d$, le minimum suivant la direction d à partir du point X^{n+I} . A partir du corollaire précédent :

$$X^{n+I} + Y.d = X^{n+I} + \sqrt{2.(F(X^{n+I}) - F_s)}.d \qquad (A)$$



Sur la droite (Δ) le minimum atteint par la fonction F suivant la direction d est le même que si l'on partait de X^I ou de X^{n+I} :

$$X^{n+1} + Y.d = X^1 + \sqrt{2(F(X^1) - F_s)}.d \qquad (B)$$

$$\text{avec} \quad F_s = (F(X^{n+1} + Y d)).$$

Si $X^{n+I} + Y.d$ sont des minimums suivant d , il est nécessaire que l'on ait:

$$d^T . A \left(X^{n+1} + Y.d + (-X^{n+1} - Yd) \right) = 0$$

$$d^T . A \left(X^{n+1} - X^1 + \sqrt{2(F(X^{n+1}) - F_s)}.d - \sqrt{2(F(X^1) - F_s)}.d \right) = 0$$

Donc on aura :

$$d^T . A \left(\mu . d + \left(\sqrt{2(F(X^{n+1}) - F_s)} - \sqrt{2(F(X^1) - F_s)} \right) . d \right) = 0$$

$$\Rightarrow \mu + \sqrt{2(F(X^{n+1}) - F_s)} - \sqrt{2(F(X^1) - F_s)} = 0$$

$$\Rightarrow \boxed{\mu = \sqrt{2(F(X^1) - F_s)} - \sqrt{2(F(X^{n+1}) - F_s)}} \qquad (D)$$

.../...

-Remarque

Remarque

Si le point $X^{n+1} + Y.d$ se trouve entre les points X^{n+1} et X^1, Alors :

$$\alpha_d^2 = 2.(F(X^{n+1}) - F(X^{n+1} + Y.d)) \text{ avec } Y.d < 0$$

Et par conséquent(II)devient:

$$\mu = \sqrt{2.(F(X^1) - F_s)} + \sqrt{2(F(X^{n+1}) - F_s)} \qquad (III)$$

De (III) pntire :

$$\mu = \sqrt{2.(F(X^1) - F(X^{n+1}) + F(X^{n+1}) - F_s)} + \sqrt{2.(F(X^{n+1}) - F_s)}$$

D'où:

$$|\Delta| < \sqrt{2.(F(X^1) - F(X^{n+1}))}$$

On sait que :

$$|\Delta| < \sqrt{2.(F(X^1) - F(X^{n+1}))}$$

Donc:

$$|\mu| > |\Delta|$$

Finalement, le déterminant de la matrice des nouvelles directions est croissant si/

$$|\Delta| / |\mu| \geq 1$$

Alors on est sûr que $DET(\varepsilon_1, \varepsilon_{n+1}, \ldots, \varepsilon_n) \quad DET(\varepsilon_1, \ldots, \varepsilon_n)$

D'après (3) on peut conclure;

a-)Si $|\Delta| / |\mu| \geq 1$ on remplace l'ancienne direction par $X^{n+1} - X^1$ et on continue le processus.

b-)Sinon on continue un nouveau cycle d'itérations en utilisant les anciennes directions.

Si le minimum de la fonction F se trouve entre X^1 et $Xn+1$, alors le critère est automatiquement non vérifié et on se branchera en B.

-III-6)CONCLUSION

Elle repose sur un changement de direction après chaque cycle d'itérations (N itérations par cycle) .

Chaque cycle correspond à N recherches (si H est une fonction à N variables) à une direction le long de chacune des directions précédentes .

La nouvelle direction de recherche est celle qui joint les deux points correspondant à la plus grande et la plus petite valeur de la fonction H .

Les directions prises correspondent au voisinage de l'optimum, aux diamètres conjugués des courbes de niveau qui deviennent des ellipses .

.../...

III-7 - Algorithme de FLE-POWELL modifié

Est-ce que les directions générées par l'algorithme initial sont toujours performantes? Non, car ces directions peuvent devenir de plus en plus mauvaises donc moins efficaces quand on augmente le nombre d'itérations.

POWELL emploie un critère qui permet d'affirmer si la nouvelle direction générée est bonne ou non.

I) Choisir qui minimise $F(X^{\circ} + \lambda d_n)$ et poser :

$$X^I = X^{\circ} + \lambda d_n$$

II) pour i =I, 2,.....,n calculer λ_i qui minimise :

$F(X^i + \lambda_i d_i)$ et poser:

$X^{i+I} = X^i + \lambda_i d_i$

III)Choisir m tel que $I \leq m \leq n$

$\Delta = (2(F(X^p) - F(X^{p+I}))^{I/2}$ soit minimum .

IV) Calculer :

$$\mu = (2.(F(X^I) - F_s))^{I/2} - (2.(F(X^{n+I}) - F_s))^{I/2}$$

F_s est telle que : y minimise :

$F_y = F(X^{n+I} + y.d)$

avec $d = X^{n+I} - X^I$

alors $F_s = F_y$

v)Si $\mu/\Delta \leq I$ alors on remplace la direcrion d_m par la direction $X^{n+I} - X^I$ et on continue la procédure avec le nouveau point de départ:

$$X^{\circ} = X^{n+I} + y.d$$

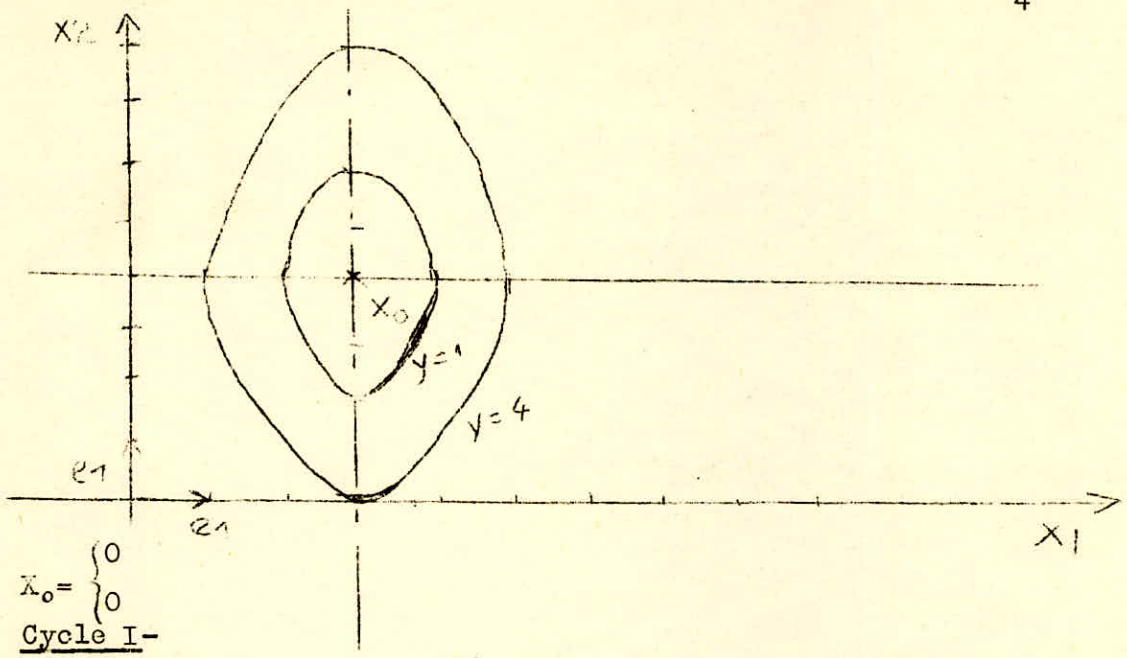
VI-Sinon on fait un autre cycle d'itérations : $X^{\circ} = X^{n+I}$

VII-La procédure s'arrete lorsqu'on ne peut plus minimiser F suivant aucune direction .l'optimum est obtenu pour $X = X^{n+I}$ .../...

V-EXEMPLE NUMERIQUE DE LA PROCEDURE DE POWELL.

Soit à minimiser la fonction $f(x_i) = Y + (X_I - 3)^2/I + (X_2 + 4)^2/4$

Les surfaces de niveau sont des ellipses concentriques en $\begin{matrix}3\\4\end{matrix}$



$X_0 = \begin{cases}0\\0\end{cases}$

Cycle I-

$*1)$ - $X^1 = X_0 + \lambda e_2 = \begin{vmatrix}0\\0\end{vmatrix} + \begin{vmatrix}0\\\lambda\end{vmatrix} = \begin{vmatrix}0\\\lambda\end{vmatrix} \Rightarrow Y(\lambda) = 9 + 1/4(\lambda^2 - 8\lambda + 16)$

$\partial Y/\partial\lambda = 0 \Rightarrow \lambda = 4$ et $Y(4) = 9$.

$*2)$ - $X^2 = X^1 + \lambda e_1 = \begin{vmatrix}\lambda\\4\end{vmatrix} \Rightarrow Y(\lambda) = \lambda - 1 \Rightarrow \lambda = 1$ et $Y(1) = 4$

$*3)$ - $X^3 = X^2 + \lambda e_2 = \begin{vmatrix}1\\4+\lambda\end{vmatrix} \Rightarrow Y = 4 + \lambda^2/4 \Rightarrow \lambda = 0$

et $Y(0) = 4$.

Direction Générée : $X^3 - X^1 = \begin{vmatrix}1\\0\end{vmatrix} \equiv e_1$.

Cycle II

$X_0 = \begin{vmatrix}1\\4\end{vmatrix}$ $*1)$ - $'X^1 = X_0 + \lambda e_2 = \begin{vmatrix}1\\4+\lambda\end{vmatrix} \Rightarrow \lambda = 0$ et $Y(0) = 4$

$*2)$ - $'X^2 = X^1 + \lambda e_1 = \begin{vmatrix}1+\lambda\\4\end{vmatrix} \Rightarrow \lambda = 2$ et $Y(2) = 0$

$*3)$ - $'X^3 = X^2 + \lambda e_2 = \begin{vmatrix}3\\4+\lambda\end{vmatrix} \Rightarrow \lambda = 0$ et $Y(0) = 0$

.../...

Direction générée $'X^3 - 'X^1 = \begin{vmatrix} 2 \\ 0 \end{vmatrix} = d.$

On voit que si l'on remplace e_1 ou e_2 par d, le déterminant

$\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ ou $\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ est supérieur à celui de $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

On remplacera par exemple EI par d.

CYCLE III.

$"XO = \begin{vmatrix} 3 \\ \\ 4 \end{vmatrix}$ Car $"XO = {}'X + Y\,d$ et on trouve que $y = 0$ rend minimum Y

$"X^1 = X"^o + \lambda e_2 = \begin{vmatrix} 3 \\ 4 + \lambda \end{vmatrix} \Rightarrow \lambda = 0 \Rightarrow Y = 0$

$"X^2 = X^{"1} + \lambda d = \begin{vmatrix} 3 + 2\lambda \\ 4 \end{vmatrix} \Rightarrow \lambda = 0 \Rightarrow Y = 0$

$"X^3 = "X^2 + \lambda e_2 = \begin{vmatrix} 3 \\ 4 + \lambda \end{vmatrix} \Rightarrow \lambda = 0 \Longrightarrow Y = 0$

On remarque qu'au cours de 3ème cycle, Y n'a pu être diminué . donc l'optimum est le point de départ du IIIème cycle, c'est à dire :

$"XO \quad \begin{vmatrix} 3 \\ 4 \end{vmatrix}$ et $Y = 0$

Ce qu'on vérifie facilement pas la méthode classique.

EXEMPLE GRAPHIQUE DE LA METHODE DE FLETCHER-POWELL.

0 : point de départ.

0, 1, 2 : points optimaux obtenus successivement lors de la première étape.

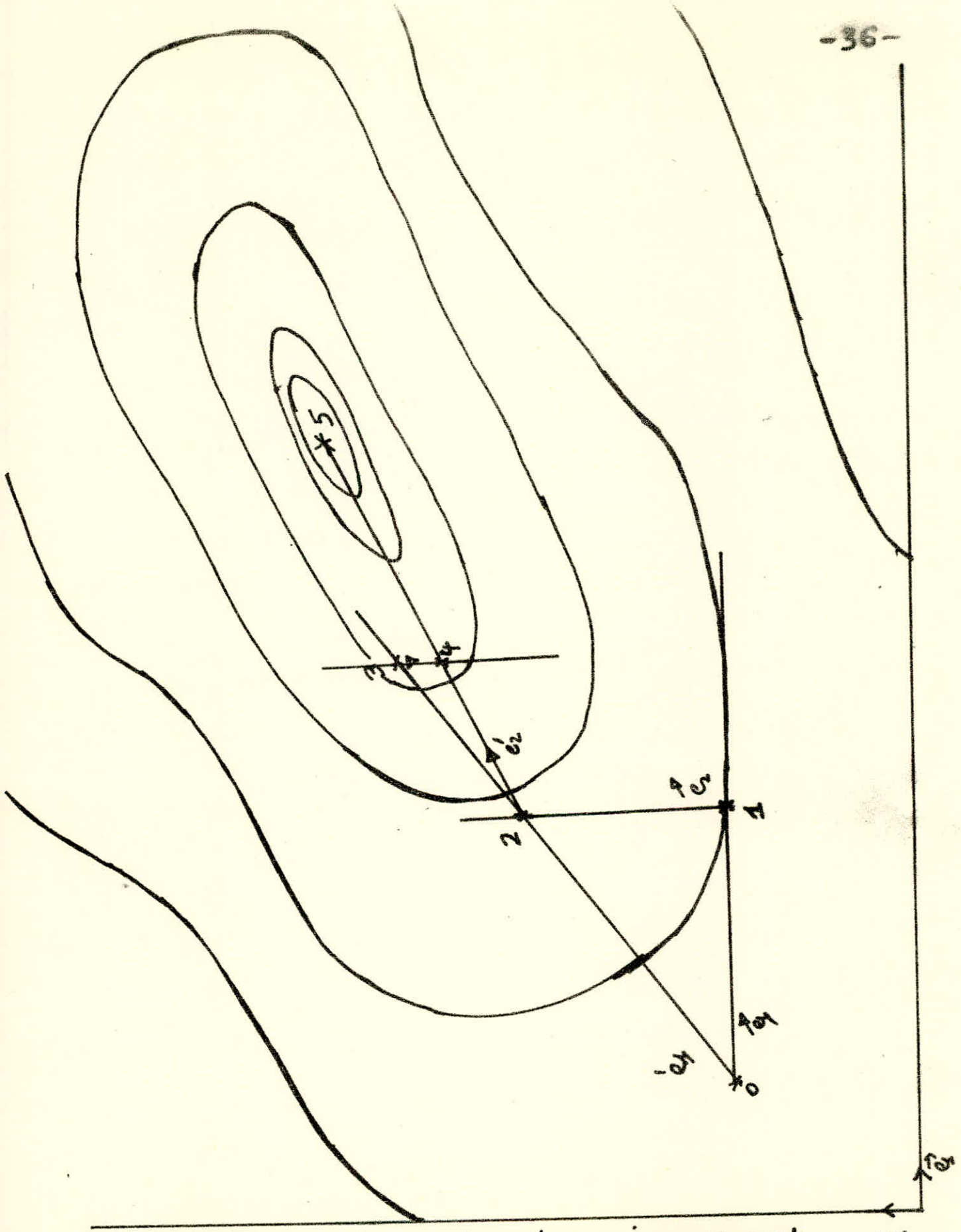
Direction générée lors de cette itération : $\overrightarrow{02}$.

2, 3, 4 : points optimaux obtenus successivement lors de la deuxième étape.

Nouvelle direction générée lors de cette itération : $\overrightarrow{24}$

Cette direction ($\overrightarrow{2.4.}$), dans ce cas particulier, nous mène directement à l'optimum (5).

On remarque que les directions de recherche correspondent, pour cette dernière itération, aux diamètres conjugués des courbes de niveaux au voisinage de l'optimum.



exemple de cheminement

VII - QUELQUES REMARQUES SUR LA METHODE DE POWELL.

La méthode de POWELL, comme toutes les autres techniques d'optimisation possède des avantages et des inconvénients.

AVANTAGES /- en ce sens qu'elle ne nécessite pas d'emploi de dérivés.

INCONVENIENTS /- A. pour optimiser une fonction H, on a besoin de 2 techniques d'optimisation : dans notre cas, c'est le nombre d'or, et la recherche aléatoire.

- Par la recherche aléatoire, on commet une certaine erreur car toute méthode compte une erreur inhérente.

  On détermine l'intervalle qui contient l'optimum avec une certaine erreur (intervalle où H (λ) sera supposé unimodale).

- Une fois ayant déterminé cette intervalle, on cherche la valeur de λ qui optimise H (λ) par la méthode du nombre d'or. Cette valeur de λ étant toujours déterminée avec une certaine erreur (qu'on peut rendre aussi petite que l'on voudra : il suffit en effet d'augmenter le nombre d'expériences).

CONCLUSIONS /-

Il est donc indispensable d'ammener nos efforts sur la méthode aléatoire car c'est à partir d'elle que le processus d'optimisation de la fonction d'everett sera engendré. Et toute erreur commise à ce niveau là risque de nous donner des résultats très érronés.

-

.../...

B). Dans le procédé de l'algorithme de POWELL, il faut se donner un point de départ X 0 - Le comportement de la méthode dépend en effet du point de départ de recherche.

EXEMPLE /-

Considérons la fonction H (X1, X2) dont les courbes de niveaux sont représentées sur la figure (II)

L'Optimum réel (Maximum) est F (X1, X2) = 90

a) prenons comme point de départ Xo et rechercons l'optimum suivant $\overrightarrow{e_1}$ ===> 70 = F (X1*).

A partir de X*, recherchons l'optimum suivant $\overrightarrow{e_2}$ ==> 80 = F (X*2).

La recherche suivant direction (X*2 - X*1) donnera toujours = F = 80.

donc F = 80 Si X'0 est pris comme point de départ (Maximum relatif)

b) Prenons comme point de départ X"0. En appliquant l'algorithme de POWELL et en prenant comme première direction de recherche $\overrightarrow{e_1}$, on aboutit au vrai maximum (Absolu).

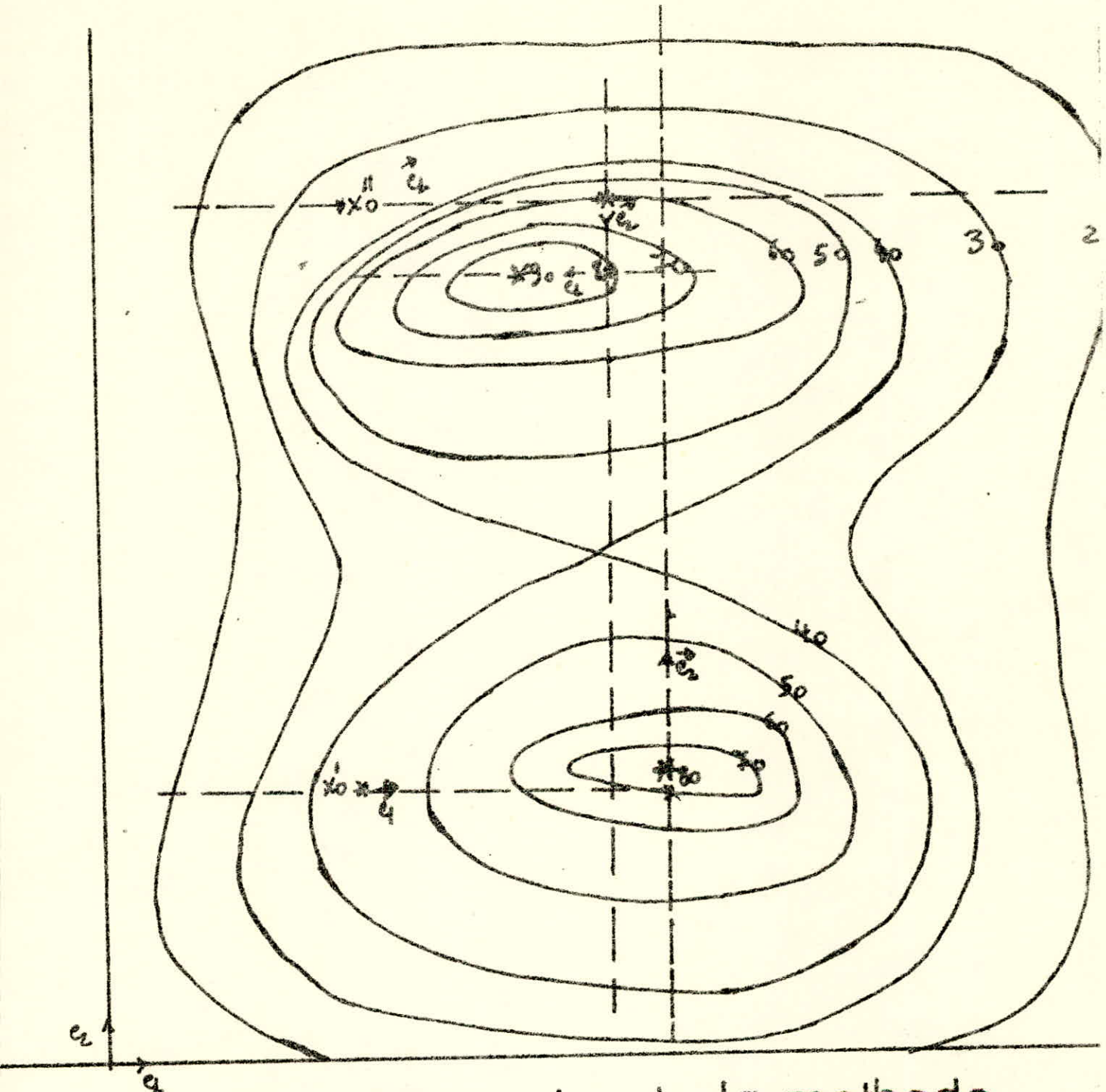
F = 90 Si X"0 est le point de départ.

* Remarque que l'hypothèse de POWELL est vérifié dans ce cas de figure :

En effet, au voisinage de l'optimum X* , F (X*) = 90, la fonction (F- est F *) est approximée par une quadrique définie négative.

* La remarque (B) n'est pas spécifique au procédé de POWELL, mais s'applique aussi aux méthodes de descentes.

L'inconvénient (A) provient surtout du fait que l'on utilise pas de dérivées.



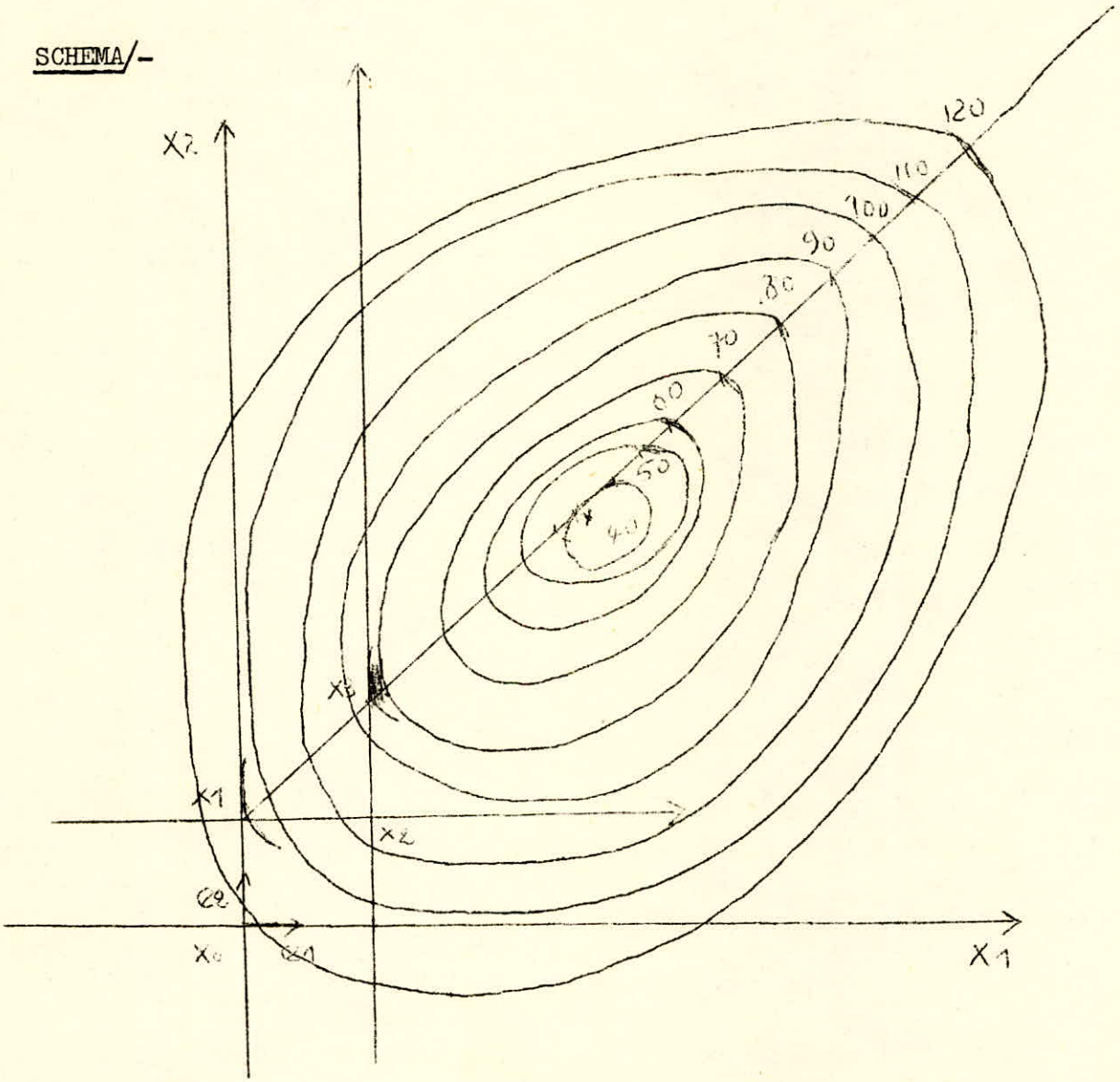
comportement de la methode de Powell en fonction du point de depart

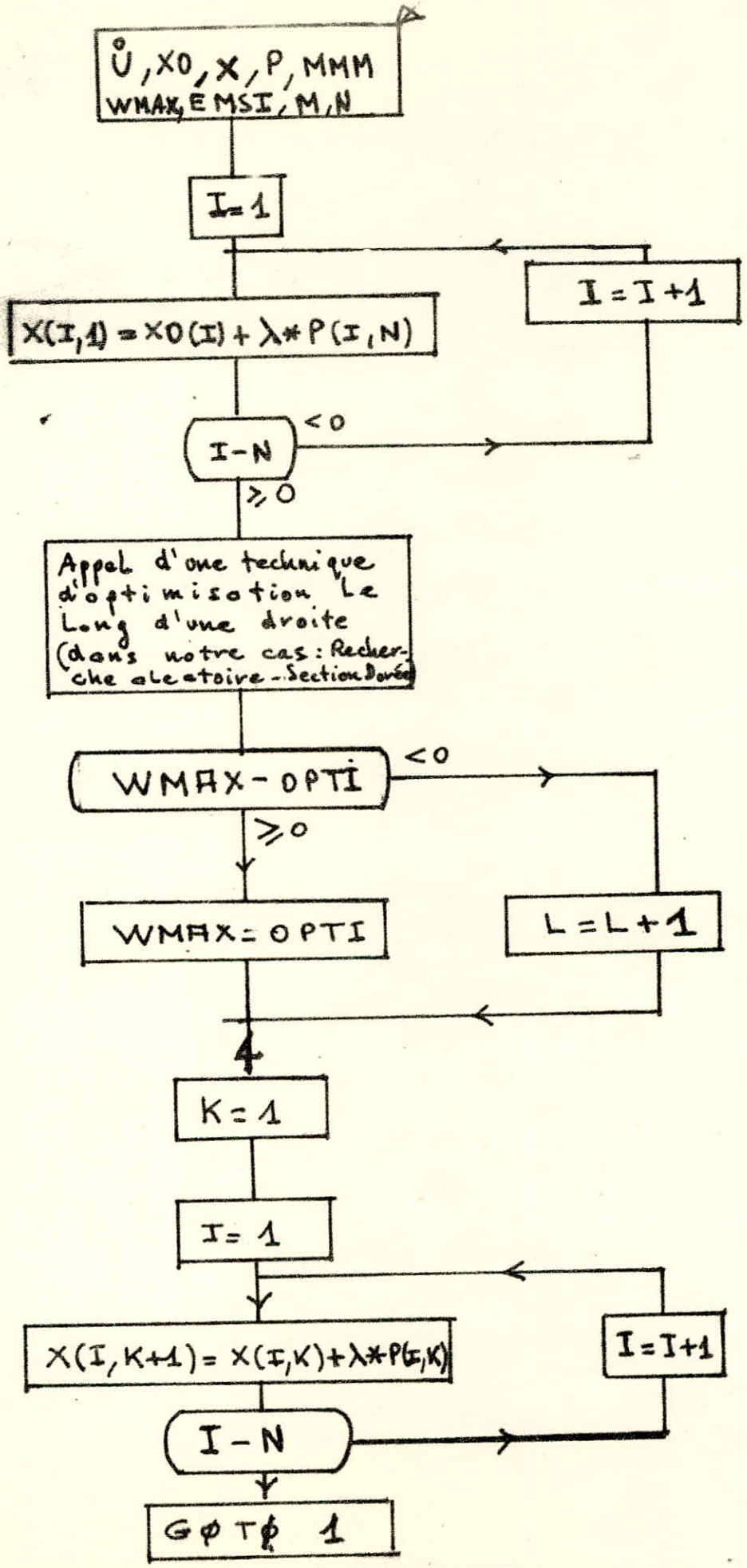
R.1.

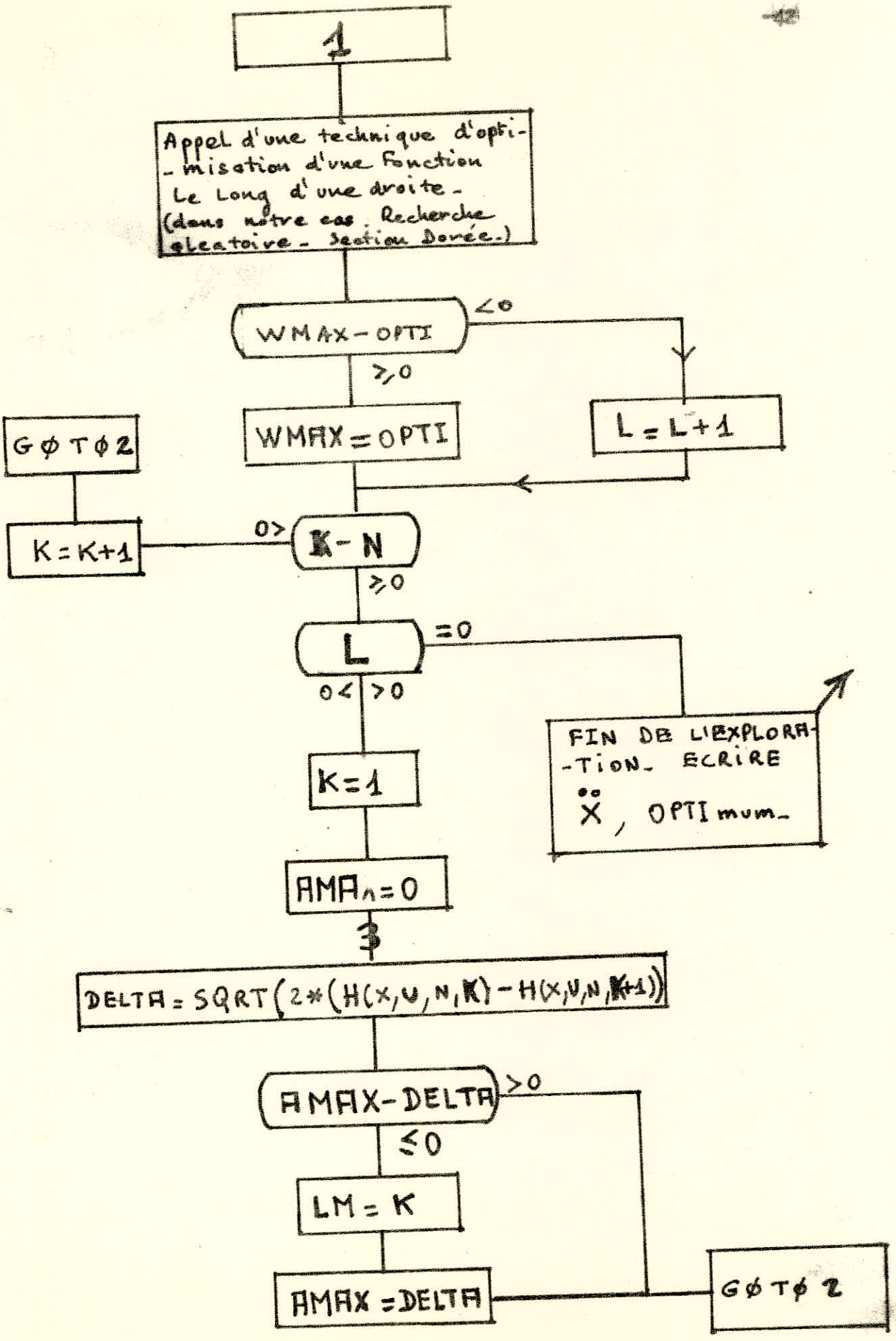
Si l'expression de la fonction H est une forme quadrique (positive) si l'on cherche un minimum, ou négative si l'on cherche un maximum, alors il est inutile de faire appel à la méthode aléatoire, car surtoute direction, F (λ) est unimodale.

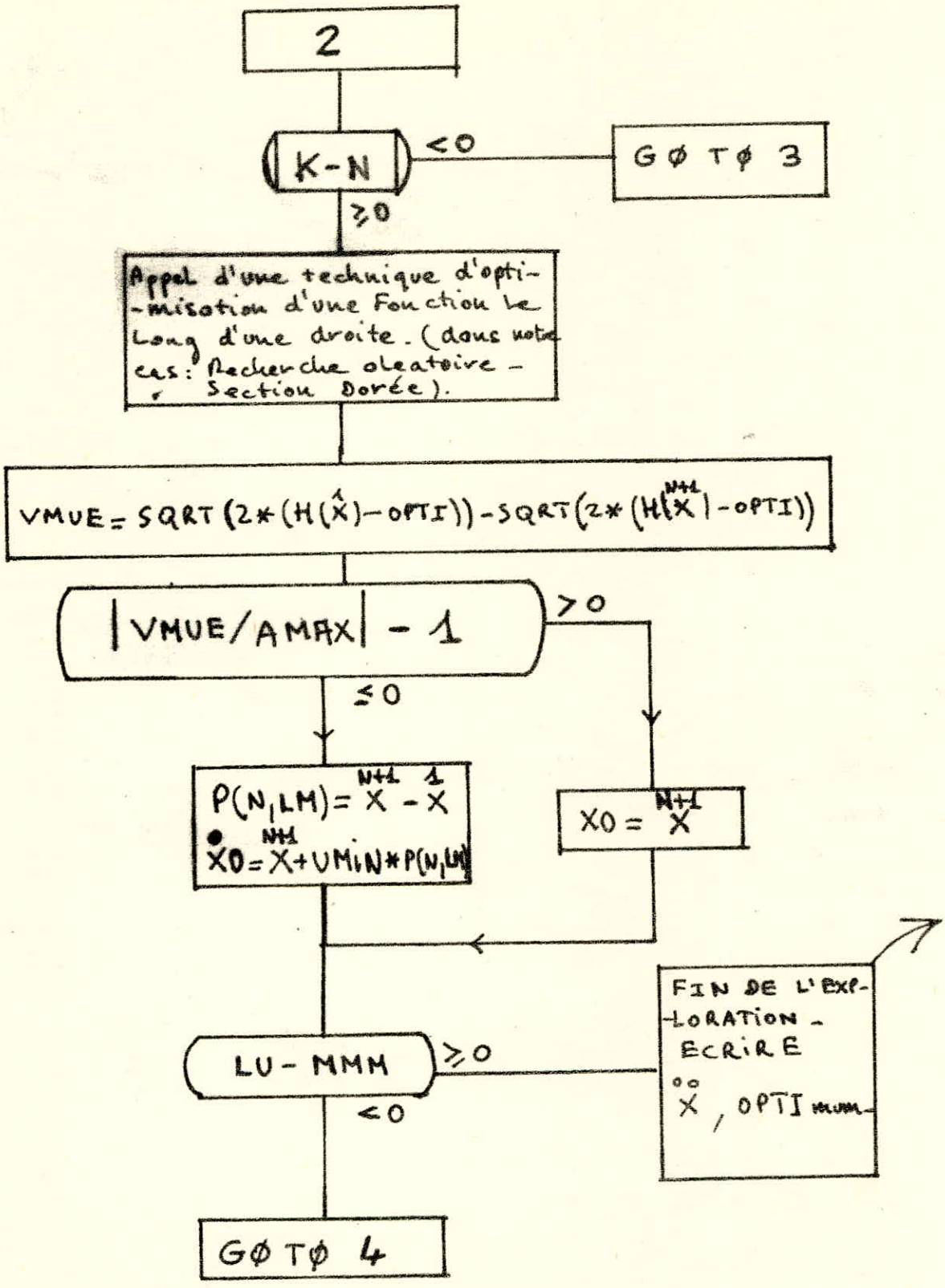
En effet, suivant toute direction , on atteint un seul minimum ou un seul maximum.

SCHEMA/-




Ů, X0, X, P, MMM
WMAX, EMSI, M, N
I=1
I = I+1
X(I,1) = X0(I) + λ*P(I,N)
I-N
<0
≥0
Appel d'une technique d'optimisation Le Long d'une droite (dans notre cas: Recherche aleatoire - Section Dorée)
WMAX-OPTI
<0
≥0
WMAX=OPTI
L=L+1
K=1
I=1
X(I,K+1) = X(I,K)+λ*P(I,K)
I=I+1
I-N
GOTO 1


1
Appel d'une technique d'optimisation d'une fonction Le Long d'une droite. (dans notre cas, Recherche aleatoire - Section Dorée.)
WMAX-OPTI
<0
≥0
WMAX=OPTI
L=L+1
GOTO2
K=K+1
0>
K-N
≥0
L
=0
0<
>0
FIN DE L'EXPLORATION. ECRIRE X, OPTImum.
K=1
AMAX=0
3
DELTA=SQRT(2*(H(X,U,N,K)-H(X,U,N,K+1)))
AMAX-DELTA
>0
≤0
LM=K
AMAX=DELTA
GOTO2


2
K-N
<0
GØTØ 3
≥0
Appel d'une technique d'optimisation d'une Fonction Le Long d'une droite. (dans notre cas: Recherche aleatoire - Section Dorée).
VMUE = SQRT(2*(H(X̂)-OPTI)) - SQRT(2*(H(X^{N+1})-OPTI))
|VMUE/AMAX| - 1
>0
≤0
P(N,LM) = X^{N+1} - X
XO = X+UMIN*P(N,LM)
XO = X^{N+1}
FIN DE L'EXPLORATION - ECRIRE XOPTIMUM
LU - MMM
≥0
<0
GØTØ 4

- TEST DU PROGRAMME. POWELL -

1. Exemple :

Minimiser : $F(x) = -x^2 e^{-x}$

le programme nous donne :

$$\text{optimum} = \begin{cases} X = 1.98946 \\ F(x) = -0.541 \end{cases}.$$

2. Exemple :

Minimiser : $F(x) = -x e^{-x}$

$$\text{optimum} : \quad \begin{cases} x = 1.00465 \\ F(x) = -0.367. \end{cases}$$

* optimisation sur la droite $X1 = 0 \Rightarrow$ ce qui donne $X2 = 0$ car F est une fonction d'une seule variable.

* 2e itération : Réduction de l'intervalle initial :

$$[-5, 4] \longrightarrow [1.0013, 1.0079] \text{ . (section dorée).}$$
$$X1 = 0, \; X2 = 1.0046 \; ; \; F(x) = -0.367.$$

* 3e itération :
optimisation suivant $\begin{cases} X1 = 1.00465. \\ F(x) = -0.367. \end{cases}$

3. Exemple.

Minimiser : $F(x) = e^{\left((x_1-1)^2 + (x_2-2)^2\right)}$

optimimum :
$$x1 = 1.00465$$
$$x2 = 1.99204$$

$$F(x1, x2) = 1.00.$$

* Point de départ $x \begin{cases} 0 \\ 0 \end{cases}$.

- Réduction de l'intervalle $(-5, 4) \longrightarrow (2.0044, 2.0110)$
valeur de $F = 2.718$ au point $x \begin{cases} 0 \\ 2.0776 \end{cases}$
- Reduction de l'intervalle $(-5, 4) \longrightarrow (1.0079, 1.0013)$.
- On arrive à l'optimum à la 3ème itération.

- /- ORMES QUADRATIQUES -

RAPPELS.

L'hypothèse fondamentale lors de l'élaboration de la méthode POUVELL, on a supposé que la fonction à optimiser était approximée par une forme quadratique au voisinage de l'optimum. On va montrer que cela n'est pas suffisant en général et que l'optimum peut être atteint dans certains cas.

Rappelons brièvement la définition mathématique d'une forme quadratique.

DEFINITION.

L'expression mathématique d'une forme quadratique homogène de degré K est :

(I) $\sum_{i=1}^{k} \sum_{j=1}^{k} \Delta x_i \cdot \Delta x_j \cdot m_{ij}$ m_{ij} = est .

Au voisinage de l'optimum, et quand les coefficient de degré égal en supérieur à 3 du développement de TAYLOR sont négligeables, la fonction peut petre approximée par une forme quadratique et l'on peut écrire :

$$F = F - F^* = \frac{1}{2}\left[\frac{\partial^2 f}{\partial x_1^2}(\Delta x_1)^2 + \frac{\partial^2 f}{\partial x_1 \partial x_2}(\Delta x_1 \cdot \Delta x_2) + \frac{\partial^2 f}{\partial x_2^2}(\Delta x_2)^2\right] + 0 * (\Delta x)^3$$

Dans le cas d'une première fonction de deux variables.
Si toute combinaison de Dx_1 et Dx_2 ne fait que diminuer DF (c'est à dire dans le cas d'un maximum), ou l'augmenter (si c'est un minimum), alors lafonction F est dite définie négative (Maxi) ou définie positive (Minimum) et elle possède des contours elliptiques.

.../....

.../...

REMARQUE : (I)

Comme on s'intéresse aux fonctions dont on ignore les dérivées, les coefficients " ", c'est à dire les dérivées secondes peuvent être approximées.

Si une fonction de plusieurs variables , désignons par H la matrice des dérivées partielles du deuxième ordre.

$$H = \begin{pmatrix} \partial^2 f/\partial x_1^2 & \cdots & \partial^2 f/\partial x_1 \partial x_2 \\ \vdots & & \\ \partial^2 f/\partial x_1 \partial x_2 & \cdots & \partial^2 f/\partial x_n^2 \end{pmatrix}$$

Le théorème de Sylvester précise que F.F* sera définie positive (donc le définit stationnaire est un minimum) si les mineurs principaux de H sont tous positifs :

$$H_1 = h\,11 \qquad H_2 = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \cdots Hn = \begin{bmatrix} h_{11} & \cdots & hn \\ h_{11} & \cdots & hnn \end{bmatrix}$$

* Si (F-F*) est définit positive le point stationnaire est un maximum.

* Si ni (F-F*) ni - (F-F*=) ne sont définies positives, le point stationnaire est un col.

CONTOURS HYPERBOLIQUES.

Il est bien connu qu'une forme quadratique peut avoir, à part les contours elliptiques, des contours hyperboliques. Le point x est alors appelé col et n'est pas l'optimum.

CONCLUSION.

On constate que même si la fonction peut être approximée à un quadrique au voisinage de l'optimum, cela ne nous garantie pas de trouver l'optimum à coup sûr.

Même si (F-F*) sera définie positive (ou négative), le prinit stationnaire risque d'être un optimum relatif.

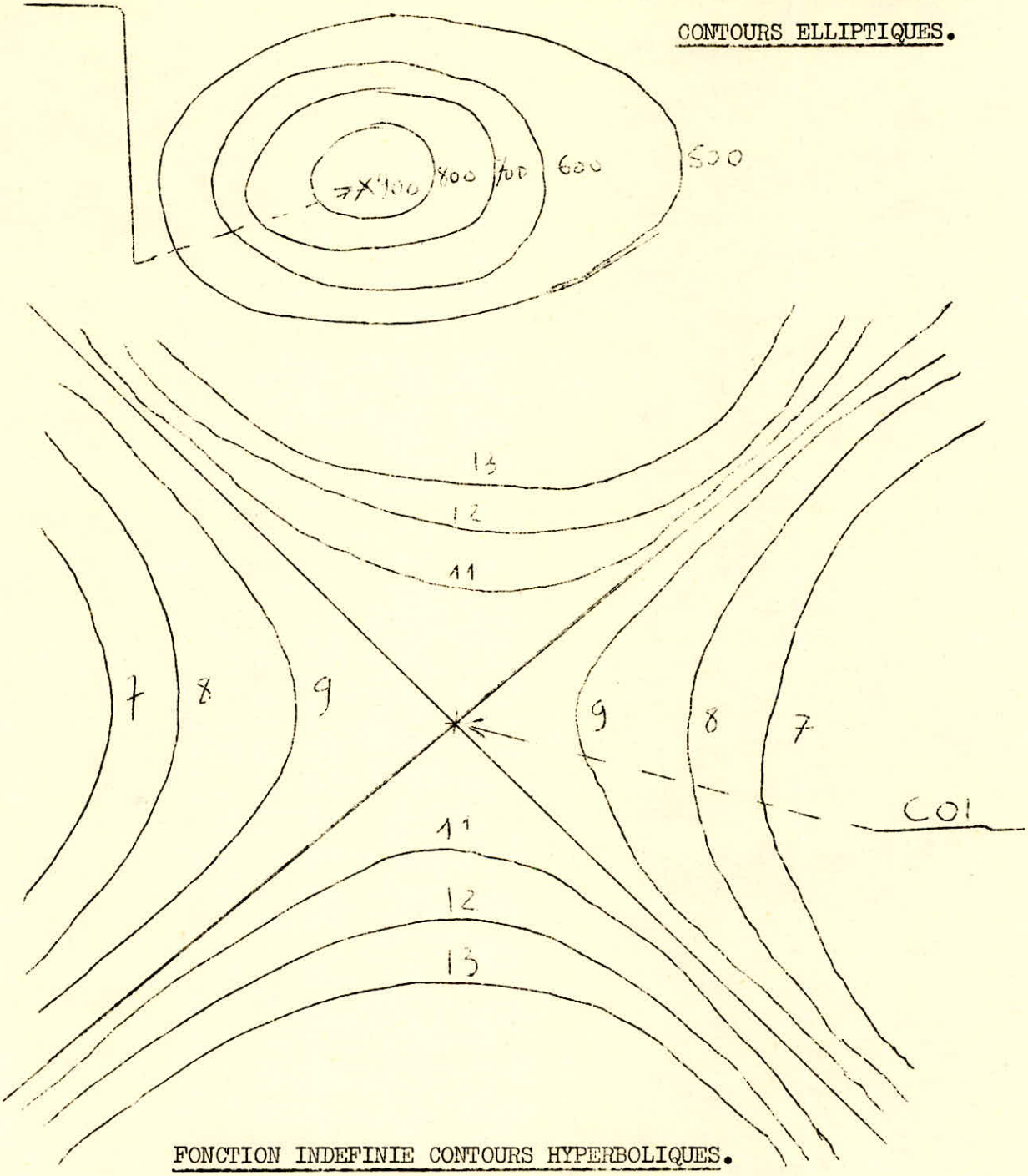
(voire discussion de la méthode de Powell).

SCHEMA.

Maximum.

FONCTION DEFINIE NEGATIVE

CONTOURS ELLIPTIQUES.



FONCTION INDEFINIE CONTOURS HYPERBOLIQUES.

-CHAPITRE -III-

C/ -REDUCTION DE L'INTERVALLE DE RECHERCHE DE $\hat{\lambda}$.

CI-CAS OU ON NE DISPOSE AUCUNE INFORMATION SUR L'INTERVALLE

CII-CAS OU CONNAIT LES LIMITES DE L'INTERVALLE

CIII-ORGANIGRAMME ET PROGRAMME

CIV-TEST.

-CI-Cas où l'on nepossède aucune information

Si on ne possède aucune information sur l'intervalle de regherche, on prendra tout simplement tout l'intervalle que peut nous fournir la machine.

-CII-Cas où on possède l'intervalle.

Généralement l'utilisateur a toujours une "idée"sur le domaine de variation des variables X_i,et porra de ce fait nous fournir une borne supérieure et une borne inférieure de l'intervalle de recherche.

Pour chaque variable X_i,les limites seront donc:

$$A(I,1) \leq X_i \leq A(I,2)$$

A chaque étape, de l'algorithme de POWELL on est amené à chercher:

$$X^{i+1} = X^i + \lambda_i d_i \qquad d_i = \text{direction donnée.}$$

Ce qui équivalent à:

$$A(i,1) \leq X(I,LJ) + \lambda P(I,JK) \leq A(I,2)$$

A-)Si $P(I,JK) > 0$

La borne inférieure de l'intervalle est donnée par:

$$DEB = MAX((A(I,1)-X(I,LJ))/P(I,JK))$$

La borne supérieure dans ce cas est:

$$FIN = MIN((A(I,2) - X(I,LJ))/P(I,JK)) .$$

B-)Si $P(I,JK) < 0$

alors:

$$DEB = MAX((X(I,LJ) - A(I,2))/P(I,JK))$$

$$FIN = MIN((X(I,LJ) - A(I,1))/P(I,JK)) .$$

Avec $I_i \cap I_{i+1} = \emptyset$

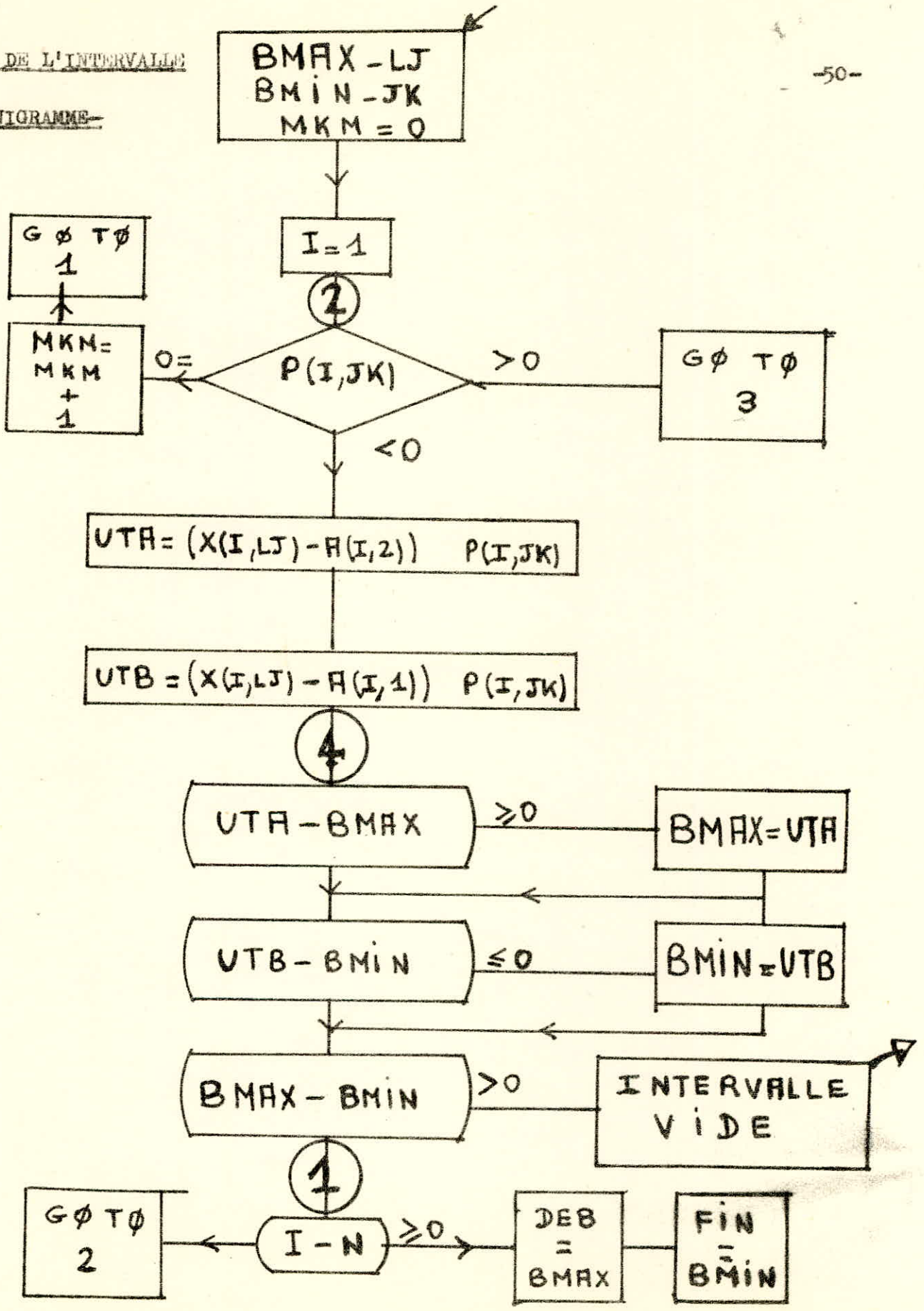
$I_1 = \{ I \in N / P(I,JK) > 0 \}$

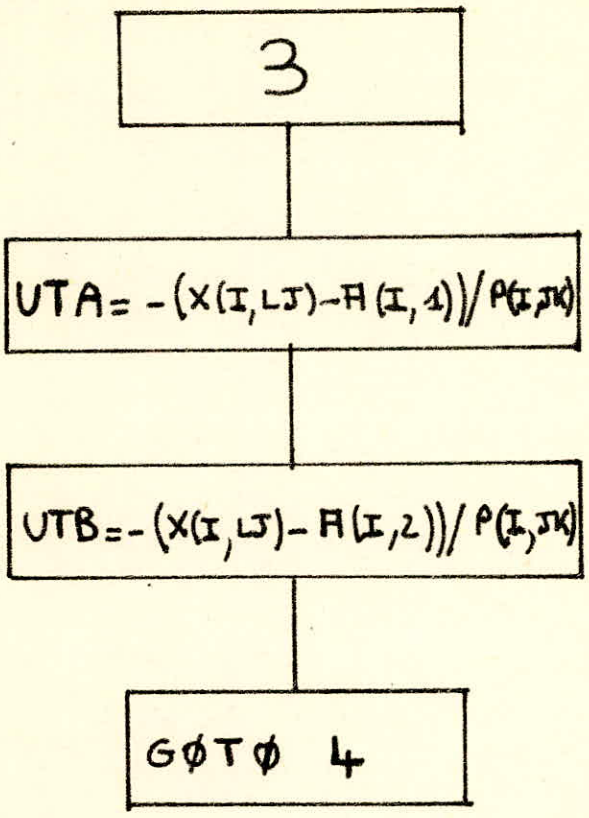
$I_2 = \{ I \in N / P(I,JK) < 0 \}$

.../...

REDUCTION DE L'INTERVALLE

-ORGANIGRAMME-





reduction de l intervalle

de

recherche

- Programme Test.

```
      DIMENSION X(2,2),A(2,2),P(2,2).
      N=2
      JK=2
      LJ=1
      BMIN=32100.
      BMAX=-32100.
      READ(2,1)  ((X(I,J),P(I,J),A(I,J),J=1,N),I=1,N)
    1 FORMAT (6(F4.1))
      DO 101 LM=1,N
      IF(P(LM,JK))102,103,104
  104 UTA=-(X(LM,LJ)-A(LM,1))/P(LM,JK)
      UTB=-(X(LM,LJ)-A(LM,2))/P(LM,JK)
      GO TO 105
  102 UTA=(X(LM,LJ)-A(LM,2))/P(LM,JK)
      UTB=(X(LM,LJ)-A(LM,1))/P(LM,JK)
  105 IF(UTA-BMAX)106,107,107
  107 BMAX=UTA
  106 IF(UTB-BMIN)108,108,109
  108 BMIN=UTB
  109 IF(BMAX-BMIN)101,191,191
  101 CONTINUE
  103 DEB=BMAX
      FIN=BMIN
      WRITE(3,2) FIN,DEB
    2 FORMAT (5X,2(F5.2,2X))
      GO TO 111
  191 WRITE(3,3)
    3 FORMAT (6X,'INTERVALLE  VIDE')
  111 CALL EXIT
      END
```

Ce Programme a été Testé Pour:

$$\begin{cases} 5 \leq X_1 \leq -6 \\ 8 \leq X_2 \leq 4 \end{cases}$$

Ce qui nous a donné un domaine de Variation Vide :

$$\begin{cases} 4 \leq 3\lambda \leq -7 \\ -3 \leq 2\cdot\lambda \leq -7 \end{cases}$$

-REDUCTION DE L'INTERVALLE DE RECHERCHE-

CHAPITRE -IV-

-RECHERCHE ALEATOIRE

A-RECHERCHE ALEATOIRE -I-

A-1 /origine

A-2 /problèmes

A-3 /exemples I

A-4 /exemples II

A-5/remarques

A-6 /conclusions

B-GENERATION DES NOMBRES AU HASARD

B-1 /nombres pseudo au hasard

B-2 /utilisation

B -3 /conclusion

C-RECHERCHE ALEATOIRE II-

C-1 /exposé de l'algorithme

C-2 /détermination R_oI et de X_oI

C-3 /organigramme et test

§§§

A-ORIGINE

Au dernier stade de résolution du problème de maximisation de H , on a abouti à optimiser H par la methode du nombre d'or .Celle-ci n'etant efficace que $H(\lambda)$ est unimodale sur l'intervalle de variation de .

A-II- Deux problème surgissent alors :

1) En travaillant sur un ordinateur, l'intervalle de variation de est limité du fait que la capacité du mot est limitée.

Ex = en simple précision sur un ordinateur IBM 1130.

$\lambda \in [-32767, +32768]$ pour un Entier

L'optimum peut tés bien se trouver à l'éxtérieur de cet intervalle.

2) H n'est pas fixement unimodale, mais généralement de forme quelconque Pour résoudre alors H, tout en utilisant la méthode du nombre d'or, on a va diviser l'intervalle H = en intervalles éga ux , asse z tra petits, de façon à supposer que sur chacuh d'eux, la fonction $H(\lambda)$ est unimodale.

Par la suite, essayer de tirer 1 de ces intervalles ayant la plus grande probabilité de contenir la valeur de λ optimale.

Comment déterminer cet intervalle ?

A-III- Exemple I : Supposons que notre intervalle soit divisé en 1000 intervalles égaux et supposons qu'on cherche les 100 meilleurs intervalle classés par ordre décroissant de H sur chacun d'eux pour tirer un intervalle au hasard et qui l'appartienne aux 100 meilleurs, est la probabilité est :

$$\frac{100}{1000} = 0,1$$

par conséquent, la probabilité de ne pas tirer 1 des 100 meilleurs en un seul éssai est : $1-0,1=0,9$

La probabilité d'échouer 2 fois est: $0,9 \times 0,9 = 0,81$ en tirant 2 éléments au hasard.

D'une façon générale, prés tirage au hasard de n intervalles, la probabilité p(0,1) d'en tirer au moins un qui appartienne au 100 meilleurs est:

$$p(0,1) = 1 - (0,9)^n$$

Si f est la fonction la plus intéressante de la région expérimentale, ~~alor~~ alors le nombre de tirages ou d'essais est donné par:

$$n = \log(1 - p(f)) / \log(1 - f)$$

ou p(f) est la probabilité de trouver au moins un élément appartenant à la fonction f . Ce procédé de recherche aléatoire présente deux caractéristiques intéressantes:

- a)il ne nécéssite aucune hypothèse sur la forme de H(λ)
- b) la fonction f est indépendante du nombre de dimension.

(voir , le WLDE ,pour plus dep'précisions.)

Tableau du nombre d'essais pour la recherche d'un maximum à l'aide de la méthode aléatoire :

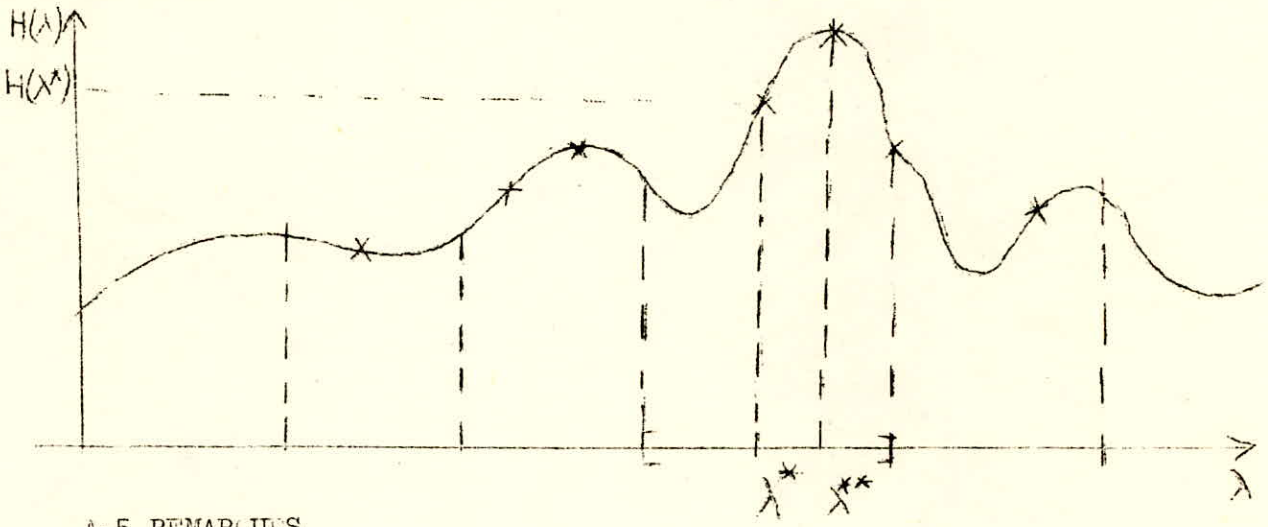
f	P(f)			
	0.80	0.90	0.95	0.99
0.1	16	22	29	44
0.05	32	45	59	90
0.025	64	91	119	182
0.01	161	230	299	459
0.05	322	460	598	919

A-4/ EXEMPLE II

Supposons que $H(\lambda)$ ait la forme de figure (I), on à calculer la valeur de $H(\lambda)$ en cinq POINTS (n =5).

Le point λ^* s'avère d'après ces calculs ,le meilleur point.Et comme l'intervalle auquel appartient λ^* est assez réduit ,H est unimodale.

On pourra alors appliquer la méthode du nombre d'or pour déterminer λ^{**}.



A-5 REMARQUES

La méthode aléatoire n'est pas sans inconvénéents .En effet c'est une méthode essentiellement simultanée et séquentielle. La méthode séquentielle est de placer la $n^{ième}$ expérience ou essai en fonction des (n-1) essais précédents , ce qui accélère la recherche de l'optimum

— Pour terminer l'étude complète du problème ,, il ne reste qu'à déterminer la longueur de l'intervalle et comment placer les n essais sur l'intervalle [illegible] , pour un [illegible]

Si m est le nombre d'intervalle ,la probabilité de tirer le meilleur en un seule essai est : I/m

La probabilité d'echouer est (I-I/m)

" " " " pour deux essais est : $(I-I/m)^2$

etc .....

La probabilité de tirer le meilleur intervalle au bout de n essais:

$$p(I/m) = I - (I-I/m)^n$$

Ce qui donne :

$$N = \log(I-p(I/m)) \;/\; \log(I-I/m) \;/$$

Si l'on fixe le nombre d'intervalles m et la probabilité avec laquelle on veut obtenir le meilleur intervalle p(I/m), il est facile de déterminer le nombre d'essais à réaliser.

Pour trouver l'emplacement de n essais il suffit de générer des nombres au hasard appartenant à: [ ] .

Si m =IO I/m =0,I et p(0,I) =0,95

le nombre d'essais est de : N =29

A-6/CONCLUSION

ON est confronté au problème suivant;

-Si ON choisit un nombre élevé d'intervalles, on aura un nombre d'essais pour la localisation de l'optimum assez élevé mais par contre une meilleur précision de l'optimum .(il ya une grande chance pour que H soit unimodale sur cet intervalle)

-Si l'on choisit un nombred'intervalles faible, le nombre d'essais pour la localisation de l'optimum sera réduit mais une précision moins bonne sur l'optimum.

B-GENERATION DES NOMBRES AU HASARD

B-I /NOMBRES PSEUDO AU HASARD

La meilleure méthode pour obtenir des nombres au hasard(pseudo) (car la plupart des problèmes pratiquess s'accomodent fort bien à ce que nous appelons des series pseudo -nombres au hasard); sur une machine arithmétique binaire, consiste à considérer l'équation suivante:

$$R_{n+I} \equiv R_n \cdot k \pmod{2^N}$$

avec :

(I)	R_n	Nombre au hasard numéro un
	R_{n+I}	" " " n+I
	k	Constante multiplicative
	N	Nombre de chiffres binaires d'un

- mot du calculateur .

Dans la plupart des machines , il est facile de réaliser l'opération MOD 2^n en effectuant d'abord exatement la multiplication K.Rn, puis en prenant pour R_{n+1} la moitié du produit .En partant de R_o impair on tirera 2^{n-2} nombres avant de retouver le même . Ces nombres peuvent être considérés comme indépendants et uniformément distribués entre 0 et I.0

-Remarque:

la relation établie plus haut n'est pas la seule pour l'obtention des nombres"pseudo au hasard" Il existe bien d'autres :méthode du milieu du carré de VON NEUMAN, etc..

On ne peut utiliser les tables de nombres au hasard (fisher and yates, MG KENDALL and Babington rand corporation ...) ,pour la bonne raison quelles nécéssitent d'être enregistrées sur calculateur pour pouvoir être consultées. Ce qui prendrait beaucoup de place sur ordinateur .

Pour toute recherche aléatoire, on emploira le S/P RANDU qui existe sur I.B.M II30 ,dans la bibliothèque des S/P;

Ce S/P calcule des variables aléatoires uniformément distribuées réelles entre 0 et I.0 et des entiers compris entre: 0 et $2^{**}15$

B- UTILISATION

CALL RANDU (IX,IY,YFL)

IX:premier nombre entier qui doit être compris entre (I et 32767)

IY: résultat entier ou nombre aléatoire entier $\in [I, 32767[$

YFL : nombre aléatoire réel $\in]0,I[$

IY : sera la prochaine variable d'entrée de RANDU.

.../...

-B-3 CONCLUSION

Ce chapitre a été introduit pour mèttre en évidence la difficulté de : recherche d'un optimum d'une fonction à une seule variable lorsque cette fonction n'est pas unimodale sur l'intervalle de recherche .
Plus le nombre de calcul est grand (c'est à dire le nombre d'expériences àréaliser),plus les chances d'approcher l'optimum augmentent.

Comme notre méthode est destinée à être appliquée par des utilisateurs il faudrait que la recherche de l'optimum soit fonction du coût que l'utilisateur est prêt à sacrifier.D'autre part diviser le segment en segments égaux demanderait plus de temps Machine, de ce fait un coût élevé .Dans qui suivra nous verrons comment eviter cela tout en tenant compte du temps de calcul.

.../...

C-RECHERCHE ALEATOIRE -II- méthode heuristique

-EXPOSE DE L'ALGORITHME

La recherche au hasard ,méthode due à LUIS et JAKOLA (1974) ,permet une réduction systématique de l'intervalle de recherche.

Soit une fonction $H(\vec{X})$; $\vec{X} \in R^n$

a-) Choisir un point de départ $\vec{X}_o$, et une amplitude de variation pour chaque variable $\vec{X}_o$, R_{oi}

$$X_{oi} - 0.5 * R_{oi} \leq X_i \leq X_{oi} + 0.5 R_{oi}$$

$$i = 1, \ldots, n$$

b-)Choisir p nombres au hasard entre -0.5 et +0.5 et les noter

p_k ; $k = 1, \ldots, p$ mettre le compteur d'itérations j à 1

c-) Définir pour chaque variable X_i p valeurs: chaque valeur est calculée par la formule suivante :

$$X_i^j = X_i^{j-1} + p_k . R_i^{j-1} \qquad k = 1, \ldots p$$

d-)Trouver les X_i^j ,c'est à dire $\vec{X}$,qui optimisent H(X) et incrémenter j de plus un (1)

e-)Arrêter le calcul si le nombre d'itérations fixé est atteint.

f-)Réduire l'amplitude de variation R_i (i=1,...n) d'une quantité

$$R^j = (1 - \varepsilon) R^{j-1} \qquad 0$$

g-)Retourner en b-

Cette méthode a l'avantage de converger vers l'optimum global de la fonction $H(\vec{X})$ si celle-ci présente des optimums locaux.

C -2)DETERMINATION DE R_{oi} ET DE X_{oi}

Lors de l'étude de réduction de l'intervalle de recherche du optimal , on est arrivé à déterminer les bornes de cet intervalle DEB et FIN. Dans notre cas , ce n'est pas une fonction à plusieures variables,que l'on a à étudier mais une fonction à une seule variable :$H(\lambda)$.

..../...

$$DEB \leq \lambda \leq FIN$$

D'où :

$$\lambda_o - 0.5.R_o = DEB$$

$$\lambda_o + 0.5.R_o = FIN$$

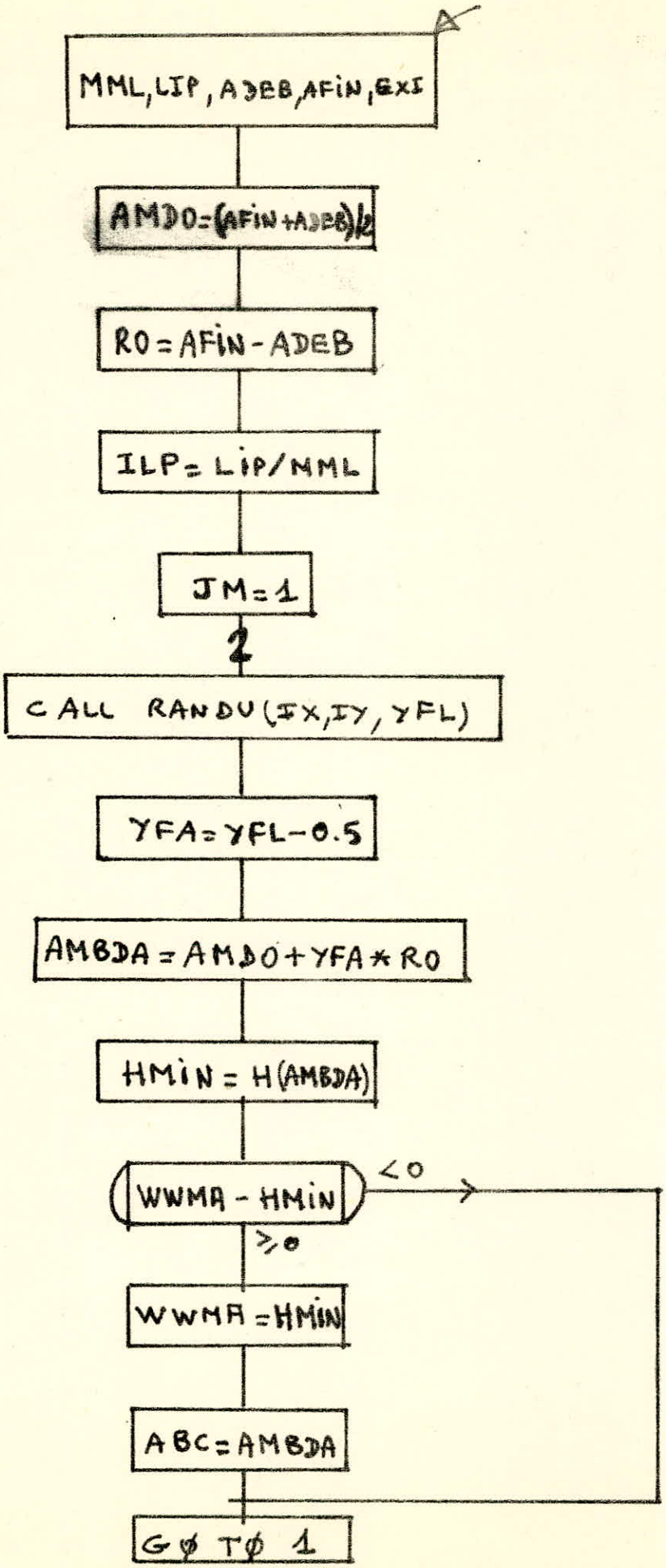
Ce qui donne :

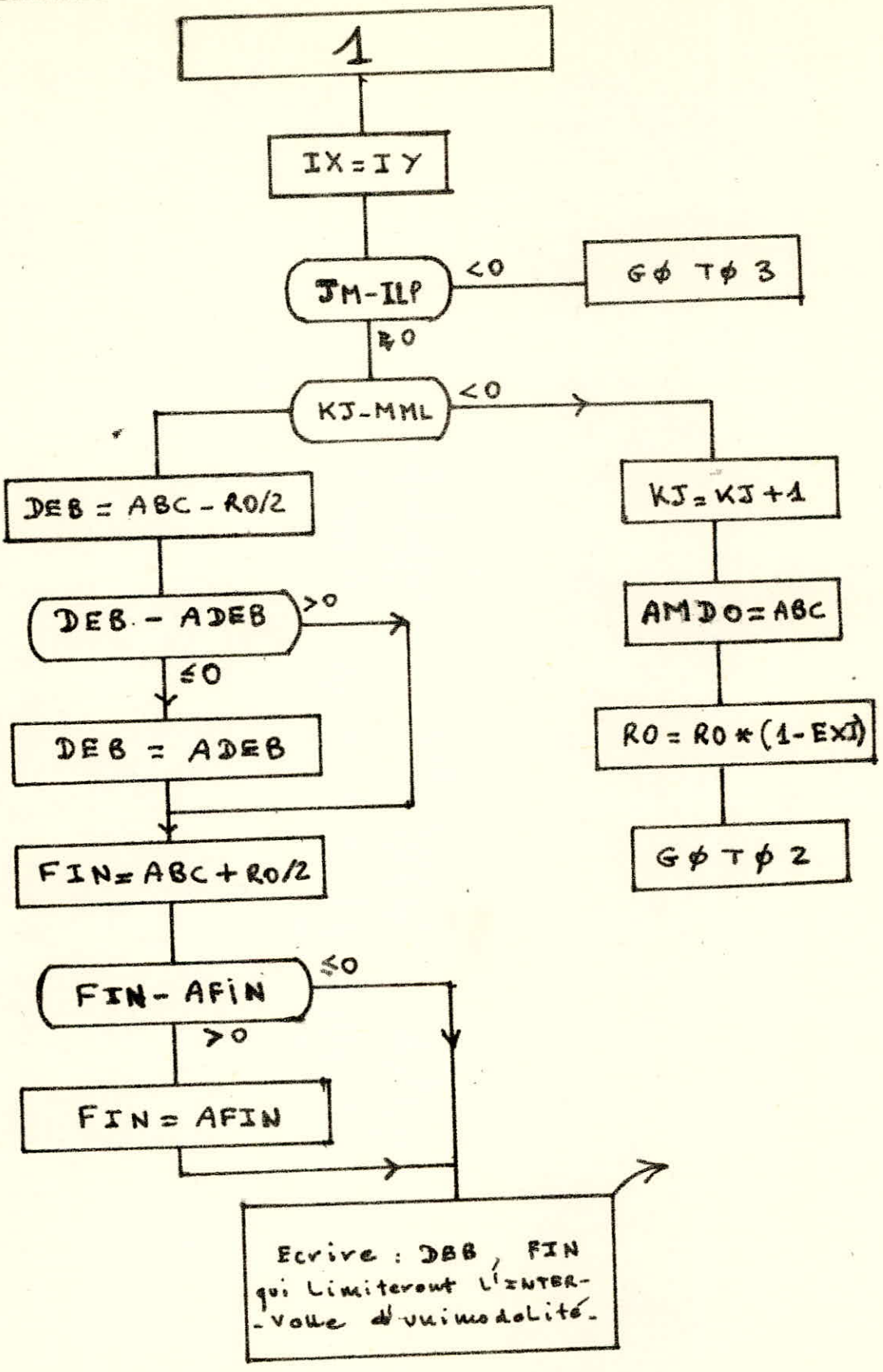
$$\lambda_o = (DEB + FIN) / 2$$

$$R_o = FIN - DEB$$

C-3-ORGANIGRAMME ET TEST -

(voir pages suivantes)


MML, LIP, ADEB, AFIN, EXI
AMDO=(AFIN+ADEB)/2
RO=AFIN-ADEB
ILP=LIP/MML
JM=1
2
CALL RANDU(IX,IY,YFL)
YFA=YFL-0.5
AMBDA=AMDO+YFA*RO
HMIN=H(AMBDA)
WWMA-HMIN
<0
≥0
WWMA=HMIN
ABC=AMBDA
GØTØ 1



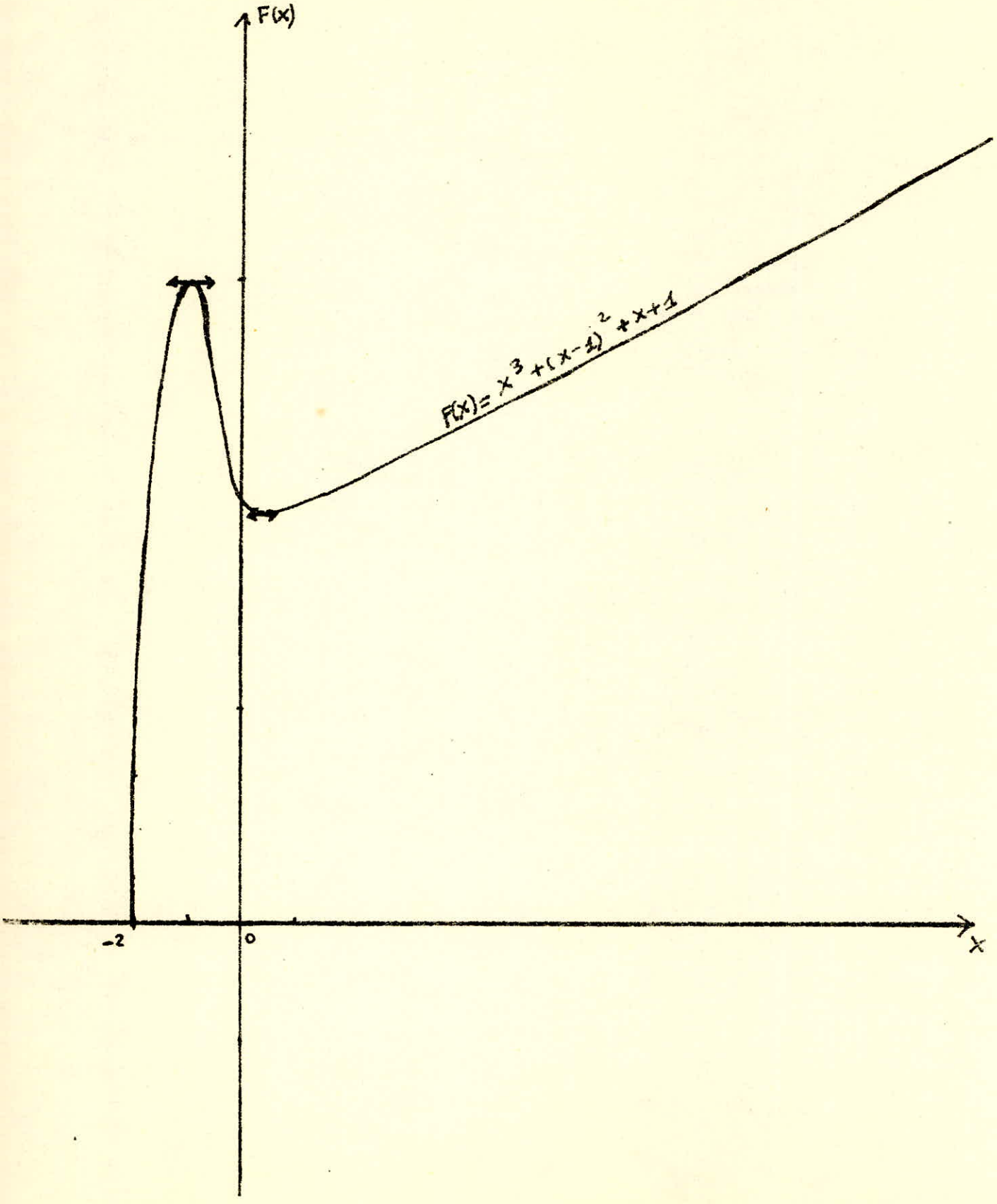
Recherche - Aléatoire -

Avec 160 points, et sur un intervalle de $[-2, 90]$, la recherche aléatoire donne :

$$-0{,}35937 \leq \text{Min}(F(x)) \leq 0{,}35937$$

Le vrai optimum ne se situe pas dans cet intervalle, comme le montre la figure.

($\text{Min } F(x) = -2 \notin [-0{,}35937 ; 0{,}35537]$)


F(x)
F(x)= x^3 +(x-1)^2 +x+1
-2
0
x

-----CHAPITRE IV ---

-METHODE DU NOMBRE DAOR-

I -ORIGINE

II-NOTION D'UNIMODALITE

-II-1 DEFINITION

-II-2 EXEMPLE

III-SUITE DE FIBONACCI

IV- METHODE DU NOMBRE D'OR

V-ALGORITHME

VI-REMA RQUE

VII-ORGANIGRAMME, PROGRAMME

-VII-1TEST 1

-VII-2 TEST 2

METHODE DU NOMBRE D' OR

I-ORIGINE

Dans le chapitre précédent ,lors de l' élaboration de l ' algorithme de FLE-POWELL ,on était à chaque fois amené à trouver un ' λ ' qui maxi--mise $H(\lambda)$:

$$\lambda \ / \ \text{Max} \ (H(\overset{i}{x} + \lambda . \overset{i}{d}))$$

Il s 'agit de trouver le λ optimal sans toujours faire appel à la notion de derivée. Comme H une fonction d 'une seule variable (λ) , on recherchera le ' λ ' par une des methodes ' de recherche directe '. SI l ' on suppose que la fonction H est 'Unimodale' sur l 'intervalle de variation de λ (on verra que meme si H ne possède pas cette propriété , comment reduire l ' intervalle de variation de λ au voisinage de l ' optimum pour que l 'Unimodalité soit verifiée.)

On a choisi la ' methode du nombre d' or ' car:

- On ne sait pas ,au depart , combien il faudra realiser d ' experiences pour determiner le λ optimal.

On aurait pu ,pour une erreur admise sur le λ optimal , determiner le nombre d ' experiences à réaliser et employer la ' methode de FIBO--NACCI ; mais cette methode est plus complexe à programmer , ne diffère de celle du ' nombre d ' or ' que par une experiences (une experiences revient à donner une valeur à λ et calculer H (H) , ce qui ne coute pratiquement rien en temps de cal cul pour un ordinateur.

- D ' autre part la methode de FIBONACCI prend beaucoup plus de place sur un ordinateur ,car il faudra stocker la suite de FIBONACCI dont on aura besoin pour l ' emplacement des experiences.

II - UNIMODALITE

-DEFINITION-

On dit qu'une fonction est unimodale dans un espace donné lorsqu'elle n'a qu'un maximum(minimum) dans cet espace. Elle peut ne pas être continue. Elle peut aussi présenter des discontinuités et même ne pas être définie dans une partie de l'espace.

(Voir exemple à la page suivante)

-METHODE DU NOMBRE D'OR.

-SUITES DE FIBONACCI

On appelle suite de FIBONACCI la suite suivante F_n telles que:

$$F_n = F_{n-1} + F_{n-2} \qquad \text{pour } n \geq 2$$

avec : $F_0 = F_1 = 1$

Si n = 2,3,4,5,.....,8 par récurence on trouve:

F_n = 2,3,5,8,.....34 etc...

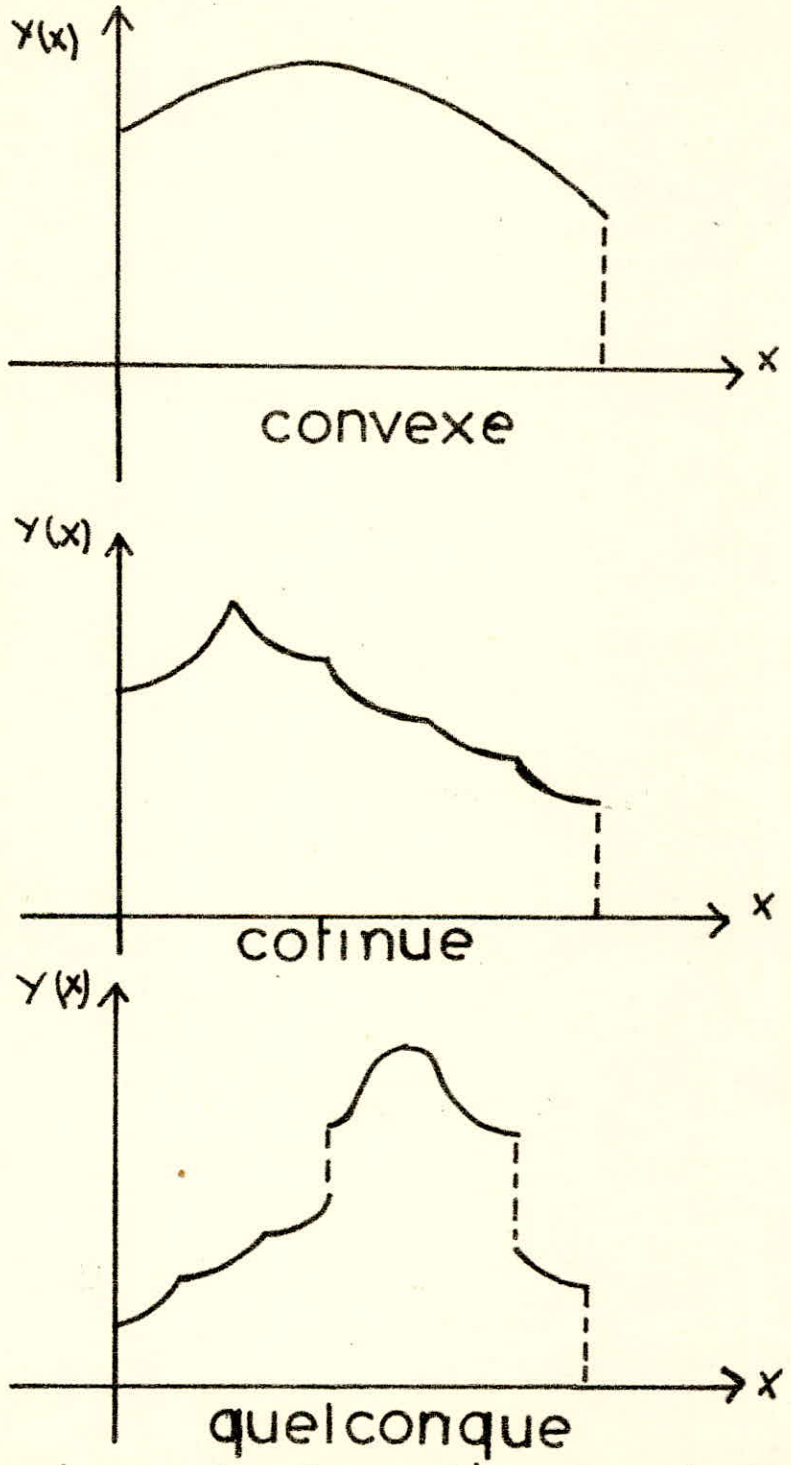
La méthode du nombre d'or consiste à déterminer le minimumou le maximum d'une fonction unimodale à une seule variable quand on ne connaît pas le nombre d'expériences à réaliser pour cette détermination.

-NOMBRE D'OR.

Le nombre d'or est défini par : $t = \frac{1+\sqrt{5}}{2} = 1{,}618033$

On démontre la formule de LUCAS.

$$\lim (F_n - 1)/F_n = 1/t .$$



exemples de fonctions Unimodales

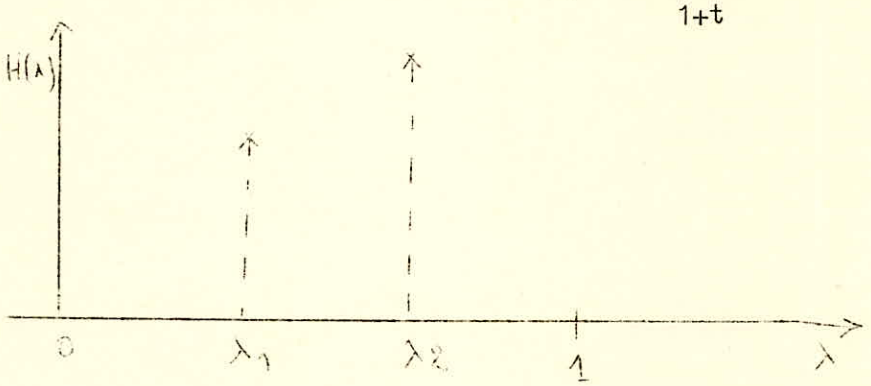
ALGORITHME

Cet algorithme résout le problème suivant: Etant donnée $H(\lambda)$ pour $0 \leq \lambda \leq 1$ trouver avec le minimum d'essais, le maximum de H à ε prés, le nombre d'essais n'est pas limité .

Si n etait connu (nombre d'essais) ,on utiliserait théoriquement la méthode de fibonacci.

-On placera la première expérience à : $\lambda 1 = \frac{1}{1+t}$

la deuxième " " $\lambda 2 = \frac{t}{1+t}$



Ce qui permet de réduire l'intervalle après deux expériences au pire à $1/t$ et l'expérience qui est dans cet intervalle joue le rôle soit de 1 , soit de 2 pour les deux expériences suivantes puisque on a: $t^2 = t + 1$

Exemple:

Dans notre cas $H(\lambda_2) > H(\lambda_1)$

l'expérience 3 sera placée à:

$$\lambda 3 = 1/(1+t) + t / (1+t) * 1/(1+t) = 1/t$$

réduction de l'intervalle où se trouve optimal à :

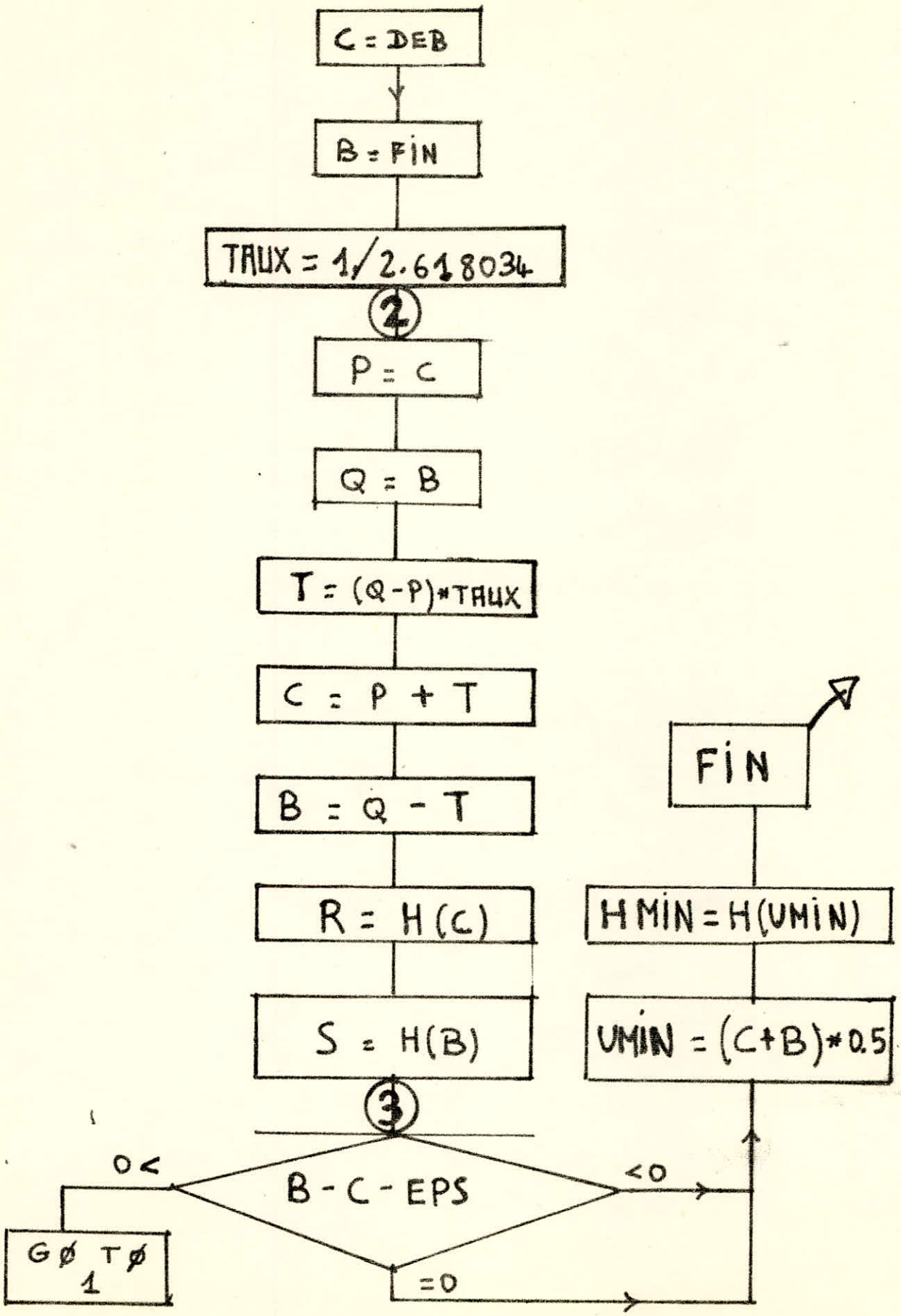
$1/t^2$,etc ....., .

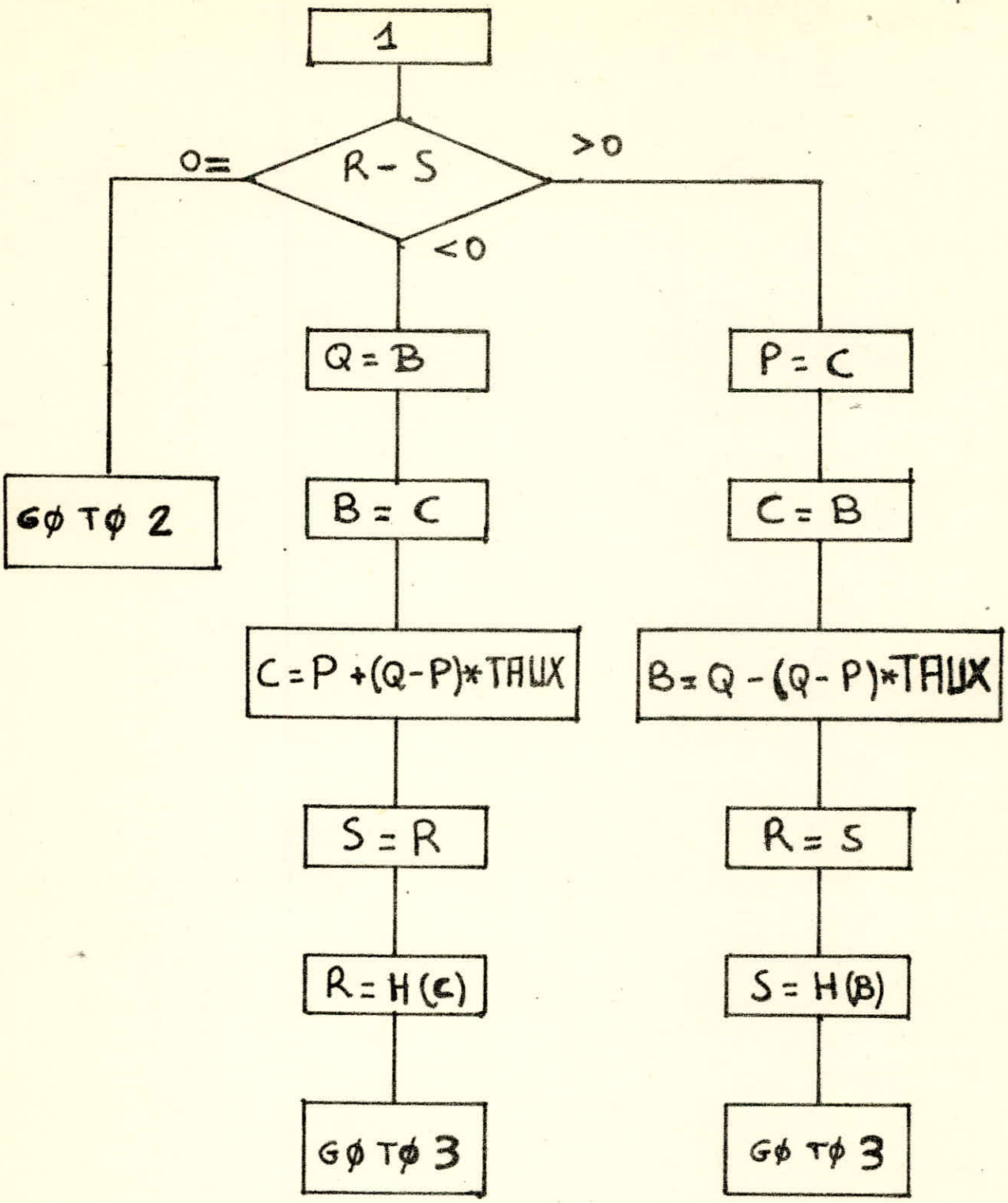
Apres n expériences , l'intervalle est réduit à ; $1/(t^{n-1}) = L_n$

-REMARQUE /

On peut cependant utiliser quand même ,dans ce cas la méthode de fibonacci.

En effet , posé est de chercher la meilleur procédure pour localiser l'optimum d'un intervalle connu à l'avance , on peut alors savoir le nombre d'essais que demande la méthode de fibonacci par la relati on : $L_n = 1/F_n \leq \varepsilon$ -/- - -


C = DEB
B = FIN
TAUX = 1/2.618034
2
P = C
Q = B
T = (Q-P)*TAUX
C = P + T
B = Q - T
R = H(C)
S = H(B)
3
B - C - EPS
0<
<0
=0
GØ TØ 1
FIN
HMIN = H(UMIN)
UMIN = (C+B)*0.5



organigramme de la methode
du
nombre d'or

ALGORITHME DE RECHERCHE D'OPTIMISATION PAR LE

-PROGRAMME

```
      SUBROUTINE NBOR (DEB,FIN,EPS,F,  MIN ,HMIN )
      C = DEB
      B = FIN
      TAUX =1./2.618034
100   P =C
      Q =B
      T =(Q - P)*TAUX
      C =P+T
      B =Q - T
      R =H(C)
      S = H(B)
200   IF (B-C-EPS)60,60,70
70    IF (R-S)10,20,30
20    GO TO 100
30    P =C
      C =B
      B =Q -(Q - P) *TAUX
      R =S
      S =H(B)
      GO TO 200
10    Q =B
      B =c
      C =P + (Q - P)*TAUX
      S =R
      R =H(C)
      GO TO 200
60      MIN =( C + B )*0.5
      HMIN =H(  MIN)
      RETURN
      END.
```

.../....

- TEST 1 PROGRAMME. NOMBRE D'OR.

* Fonction à une seule variable.

Minimiser : $F(X) = Y = -\text{Exp}(-X^2/2)$

$-1 \leq X \leq 1$ Précision de 0.01.

optimum :

$$\begin{cases} F(X) = 1.0 \\ X = 0.00 \end{cases}$$

- TEST 2.

* Fonction Unimodale Le Long d'une droite.

Minimiser : $H(X_1, X_2) = (X_1 - 1)^2 + (X^2 - 3)^2/4$

Le Long de La droite $X_1 = 1$.

$-4 \leq X_2 \leq 4$

optimum :

$$\begin{cases} H = 0.00 \\ X_2 = X_2^\circ + L\,\text{Min} \times 1 = 2.979 \\ X_1 = 1.00 \\ L\,\text{Min} = 3.979. \end{cases}$$

Avec Comme point de départ :

$$X \begin{cases} 1 \\ -1. \end{cases}$$

C O N C L U S I O N G E N E R A L E

--------- 0 ---------

Nous nous sommes efforcés au cours de ce travail, d'élaborer une méthode d'optimisation permettant de résoudre les problèmes quotidiens d'optimisation auquels sont confrontés les étudiants, enseignants et chefs d'entreprises.

Pour les deux premiers, le PACKAGE AROME sera stocké sur disque de l'ordinateur IBM 1130 et il suffira de se documenter sur la partie correspondante AROME pour les modalités d'utilisation.

En ce qui conserne son utilisation à l'extérieur de l'ENPA le problème qui se posera sera seulement une adaptation des cartes de controle au calculateur. Aussi il est certain que beaucoup de modifications peuvent être apportées au programme AROME en ce qui conserne :

* l'étude des scalaires C, E et D pour une éventuelle amélioration de la convergence,
* l'étude du test d'arrêt en ce qui conserne le produit scalaire $\vec{y} * (g(\vec{x})-\vec{b})$. En effet, celui-ci n'est jamais nul au voisinage de l'optimum. Si l'optimum sature une contrainte à 10^{-3} et si le coût de cette contrainte est de 10^{6}, le produit scalaire est trés supérieur à zéro et donc la convergence ne se réalisera jamais si l'on prend pour le test d'arrêt une erreur de 10^{-1} (ce qui est assez grand),
* de remplacer le sous-programme POWELL par une méthode plus approchée si la fonction possède des propriétés remarquable (dérivabilité, dérivées partielles connues etc ...).

UNIVERSITE D'ALGER

ECOLE NATIONALE POLYTECHNIQUE

département _ ECONOMIE

6/75

2 ex

المدرسة الوطنية للعلوم الهندسية
— المكتبة —
ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

THESE DE FIN D'ETUDES

A_R_O_M_E

Algorithme de Recherche d'un Optimum par le Multiplicateur d'Everett

PACKAGE .

المدرسة الوطنية للعلوم الهندسية
المكتبة
ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

sujet proposé par:

Mr F_D_ARMINGAUD

étudié par :

_ D_ FERROUKHI

_ M_ GUEMIRI

PROMOTION _75_

المدرسة الوطنية للعلوم الهندسية
— المكتبة —
ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHÈQUE

UNIVERSITE D'ALGER

ECOLE NATIONALE POLYTECHNIQUE

département _ ECONOMIE

EXCLU DU PRÊT

THESE DE FIN D'ETUDES

A_R_O_M_E

Algorithme de Recherche d'un Optimum par le Multiplicateur d'Everett

PACKAGE .

sujet proposé par :

Mr F_D_ARMINGAUD

étudié par :

_ D_ FERROUKHI

_ M_ GUEMIRI

PROMOTION _75_

Digitized by Google

ALGORITHME DE RECHERCHE D'OPTIMUM PAR LE MECANISME D'EVERETT.

-PACKAGE-

A-PREMIERE PARTIE -DESCRIPTION-

I-INTRODUCTION

-LES PROBLEMES D'OPTIMISATION

-INCONVENIENTS DES METHODES SPECIALISEES

-A.R.O.M.E.

-AVANTAGES -INCONVENIENTS

II-LES MODULES DE A.R.O.M.E.

-A/ EVERETT

A1-PRINCIPE

A2-DESCRIPTION DU MODULE -ALGORITHME-

-B/METHODE DE FLETCHER-POWELL.

B1-PRINCIPE

B2-DESCRIPTION DU MODULE-ALGORITHME-

-C/REDUCTION DE L'INTERVALLE DE RECHERCHE

C1-PRINCIPE

C2-DESCRIPTIONDDU MODULE -ALGORITHME-

-D/METHODE ALEATOIRE DE RECHERCHE DE L'INTERVALLE D'UNIMODALITE

D1-PRINCIPE

D2-DESCRIPTION DU MODULE -ALGORITHME-

-E/METHODE DU NOMBRE D'OR.

E1-PRINCIPE

E2-DESCRIPTION DU MODULE -ALGORITHME-

B-DEUXIEME PARTIE-MANUEL D'UTILISATION.

-I-INTRODUCTION

-II-PRESENTATION DES PARAMETRES

-III-CARTES DE CONTROLE

-IV-TEMPS DE CALCUL ET PRECISION DES RESULTATS

-V-PROGRAMME -TYPE-

-VI-EXEMPLE.

.../...

ALGORITHME DE RECHERCHE D'OPTIMUM PAR LE MECANISME D'EVERETT.

--PACKAGE--

-A- PREMIERE PARTIE--DESCRIPTION--

.../...

I-INTRODUCTION

Le package AROME permet de déterminer l'optimum d'une fonction quelconque sous des contraintes quelconques. Exemple:

-Max F(X) = Fonction économique

sous G(X) = Contraintes.

Ces problèmes d'optimisation sont souvent rencontrés dans l'industrie sous la forme :

-minimiser un coût à objectifs donnés

-maximiser une fiabilité à coût donné

-minimiser des pertes énergétiques, etc....

Les solutions habituelles sont:

-Utiliser un package spécialisé : programmation linéaire, programmation dynamique, programmation quadratique, en nombre entiers, statistique, gadient.

-Ecrire un programme général. (ad hoc)

La premiere solution concerne les problèmes à stucture simple et un grand nombre de paramètres, le second concerne des structures plus complexes avec moins dee paramètres.

Toutes deux demandent un grand effort soit de programmation ,soit de documentation.

Dans un grans nombre de cas ,l'utilisateur désire avant tout "se faire une idée" de la valeur optimale :il prêt pour cela àfaire appel à un programme général, moin s performa nt que les premiers ,mais plus simple à l'utilisateur que les premiers.

La va leur ainsi déterminée peut éventuellement êtrerapportée dans les programmes classiques.

C'est ce créneau qui vient combler le PACKAGE A.R.O.ME.

.../....

II-LES MODULES DE A.R.O.M.E.

A/EVERETT.

A1/PRINCIPE

Le problème de base est le suivant :

-maximiser $F(X)$

sous les contraintes $G_i(X) \leq B_i$

On transforme ce problème en posant: $H(x,y) = f(x) - y.g(x)$ qui la fonction d'Everett,avec y = Multiplicateur d'Everett,si on arrive à maximiser $H(x,y)$ on aura maximiser $f(x)$; (se référer à la brochure I.B.M.de F.D.ARMINGAUD.)

Pour cela aya nt trouver un X° deux cas peuvent surgir :

-a)X°vérifie les contraintes ($g_i(x) \leq b_i$),le travail estfini.

-b)X° ne vérifie pas les contraintes .

Ce qui dire ,que aboutir à un X° optimal ,on est ramené à adapter les paramètres $Y_i(k)$ en fonction de $Y_i(k-1)$et $X(k)$.

A2/ALGORITHME D'EVERETT

EVERETT propose l'algorithme suivant: en modifiant le multiplicateur de la façon qui suit:

$$y_i(k+1) = y_i(k) + S_i(k).V_i(k)$$

$S_i(k) = \pm 1$ Détermine le sens de variation de yi

$V_i(k) =$ Amplitude de y_i (adaptée selon le régime de recherche.)

On sera ramené à observer la variation de y_i (si la suite y_i converge ou diverge après un certain nombre d'itérations.) Pour cela on ajustera l'amplitude en la multipliant par des paramètres c,d,ou e selon le cas.

- si $g_i(k)$ est meilleur que $g_i(k-1)$, alors on multiplie l'amplitude par c un coefficient établi empiriquement par Everett.

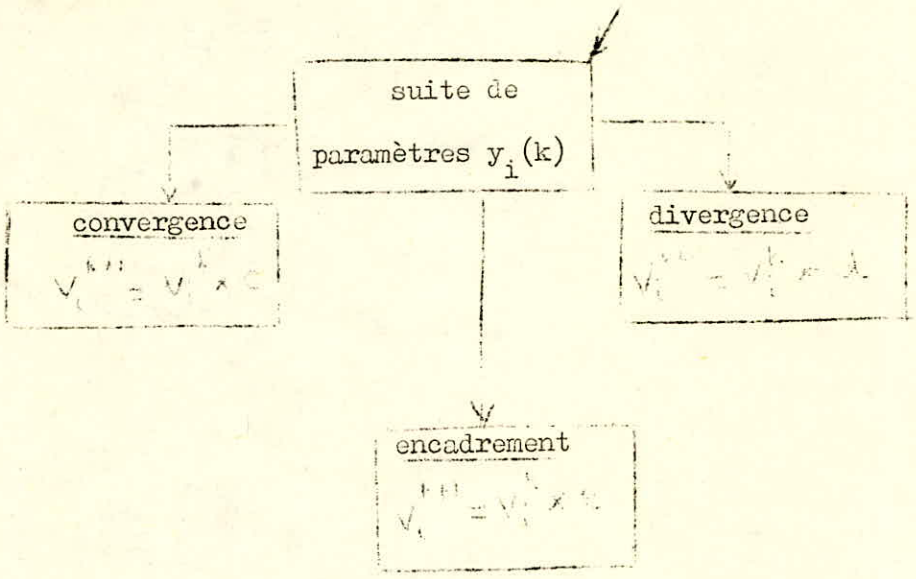
- si on a un encadrementde y ,ou $b_i \in (g_i(k);g_i(k-1))$,alors: on multiplie l'amplitude par e =coéfficient < 1

-s'il y a divergence ou si $g_i(k-1)$ est meilleur que $g_i(k)$;

.../...

C'est à dire au lieu que l'amplitude diminue à chaque itération, elle augmente.La variation de y_j $(j \neq i)$ a contrarié l'effet de y_i .

Récapitulation:



Everett propose pour les paramètres:

c =1.3

e =0.25

d =2.

Ce module est réalisé par le programme EVERT.

B/METHODE DE FLETCHER & POWELL.

-PRINCIPE

Pour maximiser la fonction H(x,y, on emploie la méthode de méthode de Flether-Powell, vue qu'elle ne fait pass appel à la notion de dérivée.(par conséquent, plus générale.)

-la méthode consiste à touver le Maximum (Minnimum) d'une fonction à l'aide des directions conjuguées.

-a)directions conjuguées

Souent deux directions U et V,et A =une matrice positive

U et V sont conjuguées si l' on a :

$$U^t.A.V = 0$$

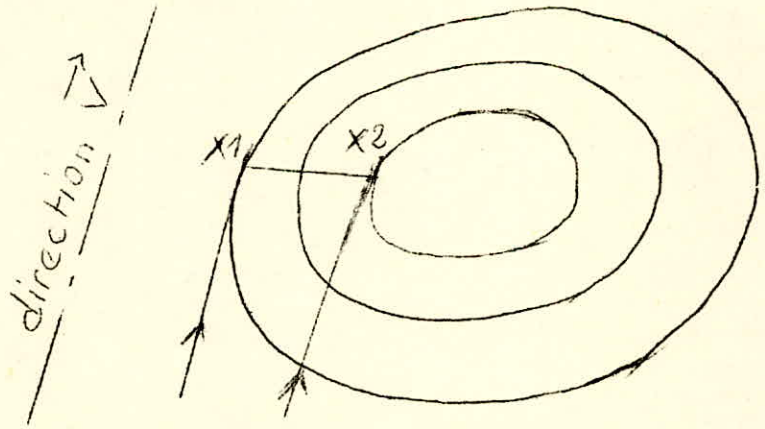
U^t =transposée de U

-b)description brêve de la méthode .

Le principe consiste à touver deux minimums à partir de points différents suivant une direction don née V, alors le vecteur joignant ces minimums(max) est conjugué à V.

-Exemple:(preuve)

X1 et X2 sont deux minimums obtenus en partant de deux points diffé rents d ans la même direction V.



Si X1 et X2 sont deux minimums on a:

(I) $V^t.(AX1 + b) = 0$

(II) $V^t.(A.X2 + b) = 0$

De (I) et (II) on tire :

$$V.A(X1 - X2) = 0$$

Donc U=X1 -X2 est conjugué à V.

.../...

-B2- ALGORITHME

L'algorithme de powell utilise le des directions conjuguées .En effet si la recherche pour chaque minimum est faite le long de p directions , alors si on joint cse minimums ,le vecteur ainsi obtenu est conjugué à css p directions.

-1- Soit F(° + dn) = fonction ,avec dn =direction donnée. et qui minimise l'expréssion :

$$x^1 = x^\circ dn$$

-2-Pour(i =1,2,....n) ,on cacule qui minimise :

$F(x^i +$ idi) et on pose :

$x^{i+1} = x^i +$ di .

-3-choisir m tel que : 1 m n et que :

$$G = (2.(F(x^p)) - F(x^{p+1}))^{1/2}$$ soit minimum.

-4- oncalcule :

$$K = (2.(F(x^1) - F_s))^{1/2} - 2.(F(x^{n+1}) - F_s))^{1/2}$$

où $F_s = F_y$ avec y minimisant $F_y = F(x^{n+1} + y.d)$

et $d = x^{n+1} - x^1$

-5-on teste si:

K/G 1, alrs on remplace la direction dm par la direction $x^{n+1} - x^1$,et on continue la procédure avec comme nouveau point

$$X^\circ = X^{n+1} + y.d \qquad (d = x^{n+1} - x^1)$$

Cet algotrithme est réalisé par le module (ou le programme) POW.

Pour minimiser la fonction H( ) par la méthode de POWELL est amené à chaque fois à trouver ( ) qui minimise H( ) .

On recherche le optimal par la méthode de recherche directe : nous emploirons ici la méthode du nombre d'or ou dection dorée.

..../....

C-REDUCTION DE L'INTERVALLE DE RECHERCHE DE

-C-1-PRINCIPE

Il serait intéressant de pouvoir réduire l'intervalle de recherche, ou le domaine de variation des variables X_i (i = 1, n).

Nous pourons de ce fait réduire le temps de recherche et par conséquent minimiser les coûtsd'emploi de la machine.

-a) si l'utilisateur ne nous fournit aucune indication sur l'intervalle, on prendra X 10^{-23}, 10^{+23} pour X ,reel et positif ou négatif .

-b) Mais généralement l'utilisateur a une idée sur le domaine de variation des X_i , et il nous donnera les deux bornes ,supérieures et inférieures telles que:

$$A(I,1) \leq X_i \leq A(I,2)$$

C-2-ALGORITHME

Achaque étape de l'algorithme de POWELL, onnest amené à chercher :

$$X^{i+1} = X^i + \lambda d_i \qquad \text{avec } d_i = \text{direction donnée.}$$

ce qui donne :

$$A(I,1) \leq X(1,LJ) + P(I,JK) \leq A(I,2).$$

Si P(I,JK) $\neq$ 0 Alors :

la borne inférieure est donnée par :

$$DEB = MAX((A(I,1) - X(I,LJ))/P(I,JK)$$

la borne supérieure est donnée par :

$$FIN = MIN(A(I,2) - X(I,LJ))/P(I,JK).$$

Si P(I,JK) $<$ 0 Alors :

$$DEB = MAX(X(I,LJ) - A(I,2))/P(I,JK)$$

et $$FIN = MIN(X(I,LJ) - A(I,1))/P(I,JK) .$$

..../....

E/METHODE DU NOMBRE D'OR .

E1-PRINCIPE

Elle consiste à déterminer le maximum d'une fonction unimodale H( ) avec 0 1 ;avec le minimum d'essais à k% près.Le nombre d'essais n'est pas connu.Si le nombre d'essais etait connu à l'avance on utiliserait la méthode de Fibonacci.n (nombre d'essais),est déterminé par :

$$L_n^* = 1/F_n$$

E-2-ALGORITHME

Le nombre d'OR est déterminé par :

$$t = (1 + \sqrt{5}) /2 = 1,618033$$

on démontre que $\lim F_{n-1} / F_n = 1/t$ (LUCAS)

où F_n est la suite de Fibonacci avec :

$$F_n = F_{n-1} + F_{n-2} \quad , F_0 = F_1 = 1$$

On placera la première expérience à :

$$\lambda 1 = 1/(1 + t)$$

la deuxième expérience à :

$$\lambda 2 = t /(1 + t)$$ 0 1

Pour la toisième expérience on aura :

$$\lambda 3 = 1/(1 + t) + t/(1 + t)*1/(1 + t) = 1/t$$

Au bout de n expériences l'intervalles se réduit à:

$$L_n = \frac{1}{t^{n-1}}$$

(POUR PLUS DE PRECISION SE REFERER AU :WILDE D.J CHAP:II)

.../...

D-1/PRINCIPE

Si la fonction à optimiser n'est pas unimodale sur l'intervalle de recherche ,c'est à dire quelle ne possède pas de minimum (respectivement de maximum)sur cet intervalle, on essayera de la rendre unimodalepar la recherche aleatoire.La méthode consiste à diviser l'intervalle H en petits intervalles égaux de façon à touver au moins un intervalle où la fonction est unimodale ,ensuite prendre parmis ces intervalles le meilleur ,c'est à dire celui qui nous procure la plus grande probabilité de cerner la valeur de optimale.

D-2/ALGORITHME

Cette méthode est due à LUIS et JAKOLA (1974),elle est aussi appelée méthode heuristique . Soit une fonction H(X) $X \in R^n$

a-)choisir un point de dépard X_o et une amplitude de variation pour chaque variable $x^*_{o_i}$, R_{o_i}

$$x_{o_i} = 0.5.R_{o_i}$$

$$i = 1,\ldots,n$$

b-) choisir p nombres au hasard entre -0.5 et +0.5 et les noter P_k $k = 1,\ldots,p$,mettre le compteur d'itération j à +1

c-) Définir , pour chaque variable X_i p valeurs :chaque valeur étant calculée par la formule suivante :

$$X_i^j = X_i^{j-1} + P_k.R_i^{j-1}$$

$$k = 1,\ldots.p$$

d-)Trouver les X_i^j , c'est à dire X^j ,qui optimisent H(X) incrémenter j de plus un , écrire H(X) et X^j .

e-)Arreter le calcul si le nombre d'itérationsfixé est atteint.

f-)Réduire l'amplitude R_i (i= 1,.....,n)d'une quantité:

$$R^j = (1 - \ )-R^{j-1}$$

.../...

g-) Retourner en b.

Cette méthode a l'avantage de converger vers l'optimum global de la fonction H(X) si celle-ci présente des optimums locaux.

-vDétermination de R_{o_I} etde X_{o_i}

Lors de l'étude de réduction de l'intervalle de recherche du optimal .. on est arrivéà déterminer les bornes supérieures et inférieures DEB etFIN Dans notre cas ,ce n'est pas une fonction à plusieursvvariables mais à une seule variable ,d'où on aura :

$$DEB \leq \quad \leq FIN$$

Donc ;

$$-0.5.R_o = DEB$$

$$+0.5.R_o = FIN$$

ce quidonne :

$$= (DEB + FIN)/2$$

$$= FIN - DEB$$

—CONCLUSION.

Vues le différentes formes de problèmes d'optimisation que rencontre quotidiènement le chef d'entreprise , il lui est quelquefois impossible de préférer uneméthode de résolution à une autre. Bien que dans la pratique il est difficile de formuler le problème sous forme mathématique et de lui affecter telle ou telle procédure pour arriver au résultat désiré.

L'avantage de notre méthode est qu'elle soit générale , donc beaucoup proche de la réalité que vit l'entreprise et la complexité de l'appareillage technologique.

Il est bien entendu que la méthode exposée ici, n'est pas des plus performantes telles que les méthodes spécialisées(programmation linéaire, dynamique ,statistique ,gradient ,.....etc ;)

mais elle procure à l'utilisateur un camps plus vaste de recherche,une utilisation facile,sans que le coût de sonapplication soit élevé.

--ALGORITHME DE RECHERCHE DE L'OPTIMUM PAR LE MECANISME D'EVERETT--

--PACKAGE--

B--DEUXIEME PARTIE--MANUEL D'UTILISATION--

-I-INTRODUCTION

Rappelons notre problème fondamentale :ils'agit d'optimiser

$H(X,Y) = F(\vec{X}) - Y.\vec{G}(\vec{X})$,transformé du problème général qui est:

optimiser: $F(\vec{X})$ suivant les containtes : $\vec{G}(\vec{X}) \leq \vec{B}$

Pour cela on aura besoin des modules suivants déjà exposés dans la partie "description" , à savoir ;

-mécanisme d'EVERETT

- méthode de POWELL(FLETCHER)

-Réduction de l'intervalle de recherche

-méthode du nombre d'or

-Eventuellement ,méthode aléa toire pour l'unimodalité de la fonction.

Les programmes à utiliser seront stokes sur disque,mais il serait préférable de les stoker séparémment pour faciliter le taitement de chaque type de problème qui pourrait se poser à l'utilisateur.Les différents modules nous les appelerons respec tivement:

1-)EVERT

2-) POW

3-)RIN

4-)NDØR

5-) ALEA

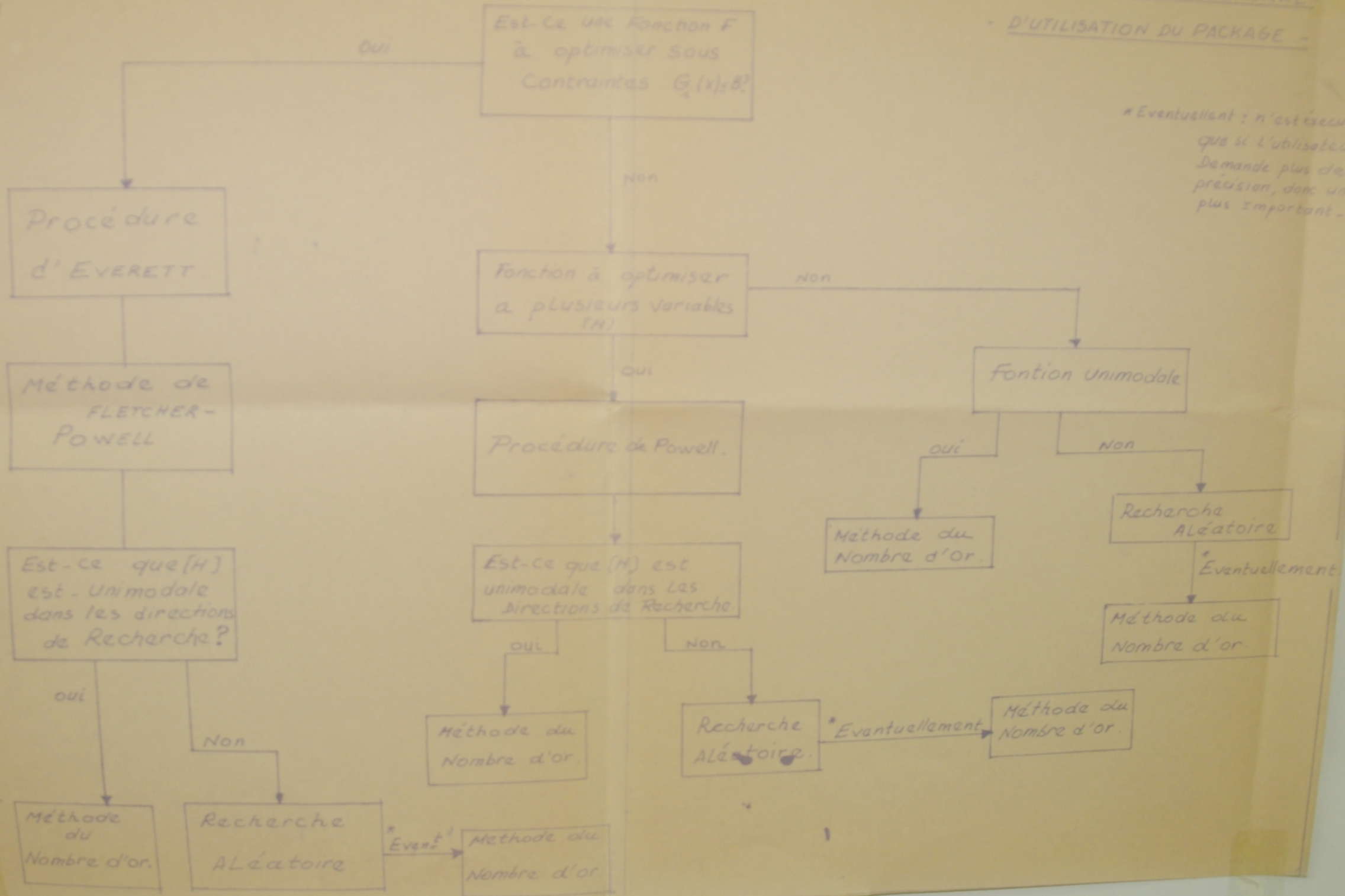
D'u ne façon générale ,si l'utilisateur a une fonction à optimiser avec des containtes il fera appel à EVERT qui enchaînera a vec POW,RIN,ALEA, et eventuellement NDØR;mais souvent il sera affronté à des problèmes plus simples et il prendra un autre cheminement .(voir organigramme d'utilisation du package)

.../...

P2006 95
Recueil p. 14
. PACKAGE.

- ORGANIGRAMME GENERAL -

- D'UTILISATION DU PACKAGE -



* Eventuellement : n'est executée que si l'utilisateur demande plus de précision, donc un cas plus important.

- II-PRESENTATION DES PARAMETRES

Ce que nous appelerons paramètres ,ce sont les données principales que l'utilisateur devrait avoir en possession.C'est à dire en fait le problème pratique qui se pose àl'entreprise,formulé d'une manière mathématique;àsavoir:

-la fonction économique à optimiser ou objectif.

-les contraintes auquellesil est affronté.

-le nombre des contraintes

-le nombre de variables

-l'intervalle de recherche de l'optimum dans la mesure du possible.

-l'erreur sur l'optimum

-les coûts d'utilisation (prévision)

-III-CARTES DE CONTROLE

Tous nos programmes ont été étudiés sur ordinateur IBM 1130,toutefois nous rappelons les cartes de commande utilisées:

```
-//JØB
-//FØR
-*IØCS(CARD,1132PRINTER)
-ØNE WRD INTEGERS
-LIST SOURCE PROGRAM
-C          commentaires,noms,;..ETC
              PROGRAMME
-// XEQ
```

Données eventuelles

.../...

-cartes de contrôles

Nous allons considérer le programme le général ,à sàvoir celui qui utilise les cinq modules déjà cités: nous aurons besoin de:

-F	Fonction économique
- W	contraintes
-A	fonction d'EVERETT
-X	Matrice des points optimaux au court d'un cycle(POW)
-U	Multiplicateurd'EVERETT
-UX	Variables
-G	Bornes de variation (RIN)
-P	Matrices de direction (POW)
-LIP	Nombre d'expériences pour déterminer l'intervalle d'unimodalité par la recherche aléatoire.
-MMM	Nombre d'itérations (POW)
-EXI	Paramètres de contraction de l'intervalle.
-EPS	Erreur sur l'optimum (NDOR)
-EMSI	Erreur sur l'optimum (POW)
-EPSI	Erreur sur l'optimum (EVERT)
-MMN	Nombre d'itérations "
-B	Second membre des contraintes (EVERT)
-UV	Amplitude des multiplicateurs (U) -(EVERT)
-S	Valeur initiale des contraintes "
-T	Valeur finale des contraintes d'une itération "
-M	Nombre de contraintes (EVERT)
-N	Nombre de variables "

DIMENSION: A(N,N+2)
P(N,N+I)
U(M,2)
UX(N)
G(N,2)

.../...

-B-4/TEMPS DE CALCUL

Il est difficile de parler de temps de calcul pour un programme donné car plus le problème à traiter est compliqué, plus le travail pour le calcul est complexe et fastidieux. Dans notre cas on tiendra compte uniquement des opérations usuelles ,àsavoir:

-l'additon (soustraction)

-la multiplicatioh(division)

-appel de fonction et de S/P

-rangement

On pourrait faire d'une autre façon ,c'est à dire effectuer plusieurs tests sur des exemples de difficultéés croissantes et essayer d'établir une loi de probabilité W(t),à partir de là on pourrait estimer le temps de calcul ,mais ceci nous prendrait un temps très long et une étu étude plus complexe.

On examinera dans ce qui suit les modules successivement:

-EVERT

nous considérons la branche la plus longue ,ensuite comme à un certain niveau il ya 3branches on divisera par trois après avoir fait la somme pondérée.

-grande branche: $C = M(16add + 3mult) + 2app\ def + 2appdde\ POW)$

-petite " $C' = M/3\ (23add + 15mult)$

nombre total d'opérations pour une itération:

$$(C + C').$$

-Comme MMN =Nombre d'itérations on aura donc:

$$(C + C').MMN$$

Si :

t_1=temps pour effectuer une addition

t_2= " " " "multiplication

.../...

t_3 = temps pour un appel de fonction

t_4 = " " " "POW

On aura pour les différentes opérations:

$$T_I = ((4I/3.t_I;M).MMN)$$

$$T_2 = ((8.t_2).M).MMN)$$

$$T_3 = MMN(2.t_3) .M$$

$$T_4 = MMN(2.t_4) .M$$

Le temps total des opérations sera la somme de ces 4 temps .

Si on tiend compte du temps de rangement ,on multipliera ce temps par deux (2) et on aura en fin de compte /

$$T = 2.(T_I + T_2 + T_3 + T_4)$$

Il est évident que ceci n'est qu'une approximation .

Pour le temps d'appel de POW on l'explicitera dans ce suit .

-POW

t'_4 =temps d'appel pour le S/P NDOR

POUR UNE ITERATION on aura :

$$-T_I = N(3 + N).t_I$$

$$-T_2 = (N^2 + N + 2).t_2$$

$$-T3 = (4 + 3N).t_3$$

$$-T'_4 = (N + N^2 + 2).t'_4$$

Si on fait comme pour EVERT,on obtient:

$$T = 2.(T_I + T_2 + T_3 + T'_4).MMM$$

Avec MMM =Nombre d'itérations.

N =itérations au court d'un cycle .

.../...

-REIN-

Dans ce programme il n'y a pas d'appel de S/P

Soit N le nombre de variables

le temps total est de :

$$T = N(5t_I + 4t_2)$$

Remarquons que les opérations utilisées sont des additions et des multiplications.

-ALEA-

Soient LIP =nombres d'expériences

MML = " D'itérations

$T_I = (2 + 6.LIP/MML).MML.t_I$

$T_2 = (LIM/MML + I).MML.t_2$

$T_3 = 3.LIP/MML.MML.t_3 = 3.LIP.t_3$

D'où le temps total est:

$$T=2.(T_I + T_2 + T_3)$$

On a procéder de la même façon que pour EVERT et les autres programmes pour le temps de rangement .

Pour être plus précis il aurait fallut tenir compte du temps de calcul du lecteur de cartes,de la latence du disque ,du temps de compilation; MAIS ces temps dépendent surtout du type de machine utilisée.

-MDOR -

D'après l'organigramme de la procédure, on est amené à prendre l'une des 3 branches ,suivant le resultat du test (R-S),on fera la moyenne des 3branches .

$T_I = N.6.t_I$

$T_2 = N.t_2$

$T_3 = N.4.t_3$

Avec N =nombre d'itérations, et t_3 =temps d'appel de la fonction H

.../...

Temps de calcul moyen :

$$T = 2(T_I + T_2 + T_3).N/3$$

Remarquons que pour le temps de calcul de H,on fera appel aussi à deux autres fonctions ,Fet le produit Y.G,dont ilfaut en tenir compte ,dans notre cas on l'a appelé t_3,On a aussi multiplié par 2 pour le Rangement

-MOTS MEMOIRES POUR LA SECTION DOREE :

-La place prise par le programme de la méthode du nombre d'or est fonction du nombre de variables de la fonction à optimiser .

Si l'utilisateur a besoin de cette méthode par la procédure d'EVERETT la place prise est alors moindre que si l'on utilisait la méthode du nombre d'or (c'està dire fonction à une seule variable) .

En effet le tableau X,P,UX seront communs aux programmes EVERT,et POW et donc seront utilisés en commun.

-a) l'utilisateur a besoin de la procédure d'EVERETT, on on aura :

2.I8 = 36 MOTS MEMOIRES

les variables sont réelles

-b)L'utilisateur une fonction à plusieures variables mais sans contraintes alors,il fera appel à POW, X,P;UX sont utilisés en commun.

La place sera:

2.I8 =36 MOTS MEMOIRES

-c)si l'utilisateur a besoin d'optimiser une fonction à une seule variable ,il fera directement appel à au nombre d'or cell-ci a été prevue pour une seulevariable on utilisera le tableau suivant pour l'adapter au cas de deux variables

UX(2),P(2,2),X(2,2)

avec $P = \begin{matrix} I & 0 \\ 0 & 0 \end{matrix}$ et $X = \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$

La place sera de :

2(2+2+4) + 2.I8 =56 MOTS MEMOIRES

.../...

-CONCLUSION.

Le travail fait ici n'est évidemment qu'approximatif (pour la détermination du temps de calcul), mais néanmoins nous avons montré une des manieres de procedurer.

Dans un programme il faut tenir compte du temps que prendrait la compilation, le stockage sur disque, de la mémoire, du lecteur de cartes, du temps d'écriture et aussi des opérations usuelles ainsi que du temps d'appel de S/P, ou de fonctions. mais ces derniers sont surtout fonction du type de machine utilisée. Toutefois nous verrons dans les exemples traités ci-après.

-EXEMPLE -I-

-Chercher la valeur de c qui minimise la fonction suivante

$$= IO.N(\ f_i.\log_{IO}(f_i/p_i))$$

$$c = (0\ ,\ IO) \quad \text{et} \quad f_i = y_i/$$

$$p_i = a.i^{-c}$$

$$a = \ i^{-c}$$

Par résolution avec la méthode de POWELL qui a fait appel autres programmes, à savoir RIN, RALEA, NDØR, nous avons trouvé les réultats suivants

Intervalle : (0 ; IO)

6.I803	3.8I96
7.6393	6.I803
7.0820	6.7376
6.9504	6.869I
6 6.9I94	6.9002
6.9I20	6.9075
6.90982	0.00000

Donc l'optimum (Minimum) est : y =6.909 et la fonction est: 0.000

.../...

- Package - 2ème Exemple -

- Minimiser $\Psi = 10 * N \left(\sum_{i=1}^{36} f_i \, Log_{10} (f_i / p_i) \right)$

avec : $f_i = y_i / \sum_{1}^{36} y_i$

$p_i = a (b+i)^{-c}$

$a = \sum_{1}^{36} (b+i)^{-c}$

$b \in (0, 100)$

$c \in (0, 10)$

* On Emploira pour cela la Méthode de POWELL qui enchainera avec les autres (NOQA, REIN, ~~NOQA~~ ALEA).

* 1° on traitera L'exemple sans ALEA.

optimum :

$$\begin{cases} a = 6.909 \;, \; b = 1.909. \\ \Psi = 0.000 \end{cases}$$

*2° En utilisant la Recherche ALéatoire.

optimum :

$$\begin{cases} a = 6.909 \\ b = 1.525 \\ \Psi = 0.000. \end{cases}$$

On a utilisé directement la Méthode de POWELL, car on se trouve devant un problème sans Contrainte

- PROGRAMME - TYPE -

```
// JOB
// FOR
* ONE WORD INTEGERS
* LIST SOURCE PROGRAM
      SUBROUTINE REIN(X, P, G, LJ, JK, DEB, FIN, MKM)
C     REDUCTION DE L'INTERVALLE DE RECHERCHE
      DIMENSION X(2,4), P(2,2), G(2,2)
      N=2
      MKM = 1
      MLM = 0
      BMIN = 32100.
      BMAX = -32100.
      DO 101 LM = 1, N
      IF( P(LM, JK) 102, 101, 104
  104 UTA = -(X(LM, LJ) - G(LM,1)) / P(LM, JK)
      UTB = -(X(LM, LJ) - G(LM,2)) / P(LM, JK)
      MLM = MLM + 1
      GOTO 105
  102 UTA = (X(LM, LJ) - G(LM, 2)) / P(LM, JK)
      UTB = (X(LM, LJ) - G(LM,1)) / P(LM, JK)
      MLM = MLM + 1
  105 IF(UTA - BMAX) 106, 107, 107
  107 BMAX = UTA
  106 IF(UTB - BMIN) 108, 108, 109
  108 BMIN = UTB
  109 IF(BMAX - BMIN) 101, 191, 191
  101 CONTINUE
      IF(MLM) 103, 191, 108
C     DEB = DEBUT DE L'INTERVALLE
  103 DEB = BMAX
      FIN = BMIN
      WRITE (3,2) FIN, DEB
    2 FORMAT( 5X, 2(F7.2, 2X))
      GO TO 111
  191 WRITE (3,3)
    3 FORMAT( 6X, 'INTERVALLE VIDE')
      MKM = 0
  111 RETURN
      END
// DUP
* STORE WS UA REIN
* DELETE REA
```

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE REA(X,P,LJ,JK,LIP,MML,EXI,ADEB,AFIN,DEB,FIN,A,N,M,U,KK)
C       DETERMINATION DE L INTERVALLE DE UNIMODACITE PAR RE-ALEA
      DIMENSION X(2,4), P(2,3), VX(2)
      DIMENSION U(2,2)
      WWMA = 32767.
      ILP = LIP/MML
      N = 2
      IX = 99
      KJ = 1
      AMDO = (AFIN + ADEB) / 2.
      RO = AFIN - ADEB
 2004 DO 2000 JM = 1, ILP
      CALL RANDU (IX, IY, YFL)
      YFA = YFL - 0,5
      AMBDA = AMDO + YFA * RO
      DO 2013 NY = 1, N
 2013 VX(NY) = X(NY, LJ) + AMBDA * P(NY, JK)
      HMIN = A(VX, U, M, KK)
      IF(WWMA - HMIN) 2005, 2006, 2006
 2006 WWMA = HMIN
      ABC = AMBDA
 2005 IX = IY
 2000 CONTINUE
      IF(KJ - MML) 2002, 2007, 2007
 2002 KJ = KJ + 1
      AMBDO = ABC
      RO = RO * (1.0 - EXI)
      GO TO 2004
 2007 DEB = ABC - RO/2.
      IF(DEB - ADEB) 2008, 2008, 2009
C       DEB =   DEBUT DE L INTERVALLE POUR SECTION DOREE
 2008 DEB = ADEB
 2009 FIN = ABC + RO/2.
      IF(FIN - AFIN) 2011, 2011, 2010
C     FIN = FIN DE L'INTERVALLE POUR SECTION DOREE
 2010 FIN = AFIN
 2011 WRITE (3, 2001) DEB, FIN
 2001 FORMAT (15X, 'DEB = ', F10.5, 5X, 'FIN = ', F10.5)
      RETURN
      END
```

```
// DUP
* STORE    WS  UA  REA
* DELETE            NBOR
// FOR   NBOR
* ONE WORD INTEGERS
* LIST SOURCE PROGRAM
* *   RECHERCHE D'OPTIMUM SUR UN SEGMENT VIA SECTION DOREE
      SUBROUTINE NBOR(X,P,A,LJ,JK,HMIN,UMIN,DEB,FIN,UX,EPS,U,KK,N,M)
      DIMENSION UX(2), X(2,4), P(2,3)
      DIMENSION U(2,2)
      N=2
      LEC = 3
      C = DEB
      B = FIN
      TAUX = 1./2.618034
  100 Z=C
      Q=B
      E=(Q-Z) * TAUX
      C = Z +E
      B = Q-E
      DO 110 KW = 1, N
  110 UX(KW) = X(KW, LJ) + C*P(KW, JK)
      R = A(UX, U, M, KK)
      DO 111 KW= 1, N
  111 UX(KW) = X(KW, LJ) + B* P(KW, JK)
      WRITE (3, 1000) B, C
      S = A(UX, U, M, KK)
C     TEST DE PRECISION
  200 IF (B-C-EPS) 60, 60, 70
C     EMPLACEMENT PROCHAINE EXPERIENCE
   70 IF (R-S) 10, 20, 30
   20 GO TO 100
   30 Z = C
      C = B
      B = Q-(Q-Z)* TAUX
      WRITE (3, 1000) B, C
      R = S
      DO 120 KW = 1, N
  120 UX(KW) = X(KW, LJ) + B * P(KW, JK)
      S = A(UX, U, M, KK)
      GO TO 200
   10 Q = B
      B = C
      C = Z + (Q - Z) * TAUX
      WRITE (3, 1000) B, C
      S = R
```

```
      DO 121 KW = 1, N
121   UX(KW) = X(KW, LJ) + C * P(KW, JK)
      R = A(UX, U, M, KK)
      GO TO 200
60    UMIN = (B+C) * 0.5
      DO 131 KW = 1, N
      UX(KW) = X(KW, LJ) + UMIN * P(KW, JK)
131   X(KW, LJ+1) = UX(KW)
C     POIN OPTIMAL
      WRITE(3, 41) (UX(KW), KW = 1, 2)
41    FORMAT(5X, 2(F9.5, 2X))
C     OPTIMUM
      HMIN = A(UX, U, M, KK)
      WRITE(LEC, 40) UMIN, HMIN
40    FORMAT(5X, F6.3, 4X, F7.3 //).
1000  FORMAT(2(F8.4, 4X)).
      RETURN
      END.
// DUP
* STORE        WS   UA   NBOR
* DELETE                  A
// FOR        A (= EXEMPLE DE FONCTION)
* ONE WORD INTEGERS
* LIST SOURCE PROGRAM
* * EXEMPLE DE FONCTION
      FUNCTION A(X, U, M, KK)
      DIMENSION X(2), U(2,2)
      A = F(X)
      DO 2222 I = 1, M
2222  A = A + U(I, KK) * W(I, X)
      RETURN
      END
// DUP
* STORE        WS   UA   A
* DELETE       POW
// FOR
* ONE WORD INTEGERS
* LIST SOURCE PROGRAM
      SUBROUTINE POW(X, P, G, A, U, N, M, KK, MML, LIP, EXI, JREA, EPS, EMSI,
     1UX, REIN, REA, NBOR)
      EXTERNAL A
      DIMENSION X(2, 4), P(2, 3), G(2, 2)
      DIMENSION U(2,2)
      LM = 0
      L = 0
      LJ = 1
      JK = 2
```

```
      WMAX=32767.
      LU = N+1
      LW= N+2
      MN = 0
      CALL REIN(X, P, G, LJ, JK, DEB, FIN, MKM).
      IF (MKM) 24, 33, 26
C     INITIALISATION
26    IF(JREA) 47, 46, 47
47    ADEB = DEB
      AFIN = FIN
      CALL REA(X, P, LJ, JK, LIP, MML, EXIT, ADEB, AFIN, DEB, FIN,
     1 A, N, M, U, KK)
46    CALL NBOR (X, P, A, LJ, JK, HMIN, UMIN, DEB, FIN, UX, EPS, U, KK, N, M)
12    IF(ABS(WMAX-HMIN) - EMSI) 13, 13, 33
13    WMAX = HMIN
      L = L+1
      LOP = 2
33    DO 14 K=2, LU
C     PROCEDURE D'OPTIMISATION.
      CALL REIN(X, P, G, K, K-1, DEB, FIN, MKM)
      IF (MKM) 34, 14, 34
34    IF( JREA) 48, 49, 48
48    ADEB = DEB
      AFIN = FIN
      CALL REA (X, P, K, K-1, LIP, MML, EXI, ADEB, AFIN, DEB, FIN, A,
     1 N, M, U, KK)
49    CALL NBOR (X, P, A, K, K-1, HMIN, UMIN, DEB, FIN, UX, EPS, U, KK, N, M)
      IF(WMAX-HMIN) 14, 14, 16
16    IF (ABS(WMAX-HMIN) - EMSI) 11, 11, 14
11    WMAX = HMIN
      LOP = K+1
      L = L+1
14    CONTINUE
      IF(L) 17, 91, 17
C     RECHERCHE D'UNE NOUVELLE DIRECTION CONJUGUEE
17    AMAX = 0.
      DO 18 K = 2, N
      BTA = 0.
      DO 27 MU = 1, N
27    UX(MU) = X(MU, K)
      BTA = - BTA - A(UK, U, M, KK)
      DO 28 MU = 1, N
28    UX(MU) = X(MU, K+1)
      BTA = -BTA - A(UK, U, M, KK)
      DELTA = - SQRT (BTA)
      IF (AMAX - DELTA) 19, 19, 18
19    LM = K
      AMAX = DELTA
18    CONTINUE
```

```
      DO 21  MU = 1,N
   21 P(MU,N+1) = X(MU,N+1) - X(MU,1)
      CALL REIN(X,P,G,N+1,N+1,DEB,FIN,MKM)
      IF (MKM) 35 ,36 ,35
   35 IF (JREA) 51, 52, 51
   51 ADEB = DEB
      AFIN = FIN
      CALL REA(X,P,N+1,N+1,LIP,MML,EXI,ADEB,AFIN,DEB,FIN,A,N,MU,MKM)
   52 CALL NBOR(X,P,A,N+1,N+1,HMIN,UMIN,DEB,FIN,UX,EPS,U,KK,M,MK)
      GOTO 37
   36 DO 38  MU = 1,N
   38 UX(MU) = X(MU,N+1)
      HMIN = A(UX,U,M,KK)
   37 DO 29  MU = 1,N
   29 UX(MU) = X(MU,1)
      VMA = SQRT(A(UX,U,M,KK))
      DO 31  MU = 1,N
   31 UX(MU) = X(MU,N+1)
      VMB = SQRT(A(UX,U,M,KK))
      VMUE = VMA - VMB
      IF (ABS(VMUE/AMAX) - 1.) 22, 23, 23
   23 DO 24  MU = 1,N
      P(MU,LM-1) = X(MU,N+1) - X(MU,1)
   24 X(MU,1) = X(MU,N+1) + UMIN * P(MU,LM-1)
      L = 0
      MN = MN+1
      WRITE (3,43)  MN, LOP
      WRITE (3,44) ((P(I,0), J = 1,LU), I = 1,N)
      GOTO 26
   22 DO 25  MU = 1,N
   25 X(MU,1) = X(MU,N+1)
      L = 0
      MN = MN+1
      WRITE (3,43)  MN, LOP
      WRITE (3,44) ((P(I,J), J = 1,LU), I = 1,N)
      GOTO 26
C     ECRITURE DE L'OPTIMUM
   98 DO 32  MU = 1,N
   32 UX(MU) = X(MU,1)
C     OPTIMUM
      OPTI = A(UX,U,M,KK)
      WRITE (3,42)  OPTI
   42 FORMAT (15X, F9.3 //)
C     POINT OPTIMAL
      WRITE (3,41) UX(MU), MU = 1,N)
   41 FORMAT (5X, 2 (F6.3, 4X))
   43 FORMAT (15X, 2 (I9, 5X) //)
   44 FORMAT (15X, 3 (F6.2) //)
      RETURN
      END
```

```
// DUP
* STORE          WS    UA    POW
* DELETE                     F
// FOR
* ONE WORD INTEGERS
* LIST SOURCE PROGRAM
      FUNCTION F(X)
      DIMENSION X(2)
      F = ____________
      RETURN
      END

// DUP
* STORE          WS    UA    F
* DELETE
// FOR
* ONE WORD INTEGERS
* LIST SOURCE PROGRAM
      FUNCTION W(K,X)
      DIMENSION X(2)
      GO TO (81, 82) K
 81   UL = ____________
 83   W = UL
      RETURN
      END
// DUP
* STORE          WS    UA    W
// FOR
* IOCS (CARD, 1132 PRINTER)
* ONE WORD INTEGERS
* LIST SOURCE PROGRAM
      EXTERNAL A
      EXTERNAL REA
      EXTERNAL NBOR
      EXTERNAL REIN
      DIMENSION X(2), T(2), U(2,2), B(2), UY(2)
      DIMENSION X(2,4), P(2,3), G(2,2), UX(2)
      UREA = 0
      MML = 2
      LIP = 6.60
      EXI = 0.5
      EMSI = 0,001
      EPS = 0.01
      N = 2
      M = 2
      LU = N+1
      LW = N+2
      READ (2,64) (( LU(I,J), J= 1, M), I = 1, M)
 64   FORMAT(2(F 5.2))
```

```
      READ(2,65)( B(I), I = 1, M)
   65 FORMAT (2( F5.2))
      READ(2, 1003)(( X(I,J), J = 1, LW), I = 1, N)
 1001 FORMAT( 4 F4.1)
      READ (2, 1002) (( P( I,J) , J= 1, LU) , I = 1, N)
 1002 FORMAT( 3 F4.1)
      READ (2, 1) (( G(I,J) , J = 1, N) , I = 1, N)
    1 FORMAT( 4 F7.3)
      EPSI = 0.5
      D = 2.
      C = 1.3
      E = 0.25
      J = 1
      KK = 1
      CALL POW (X, P, G, A, U, N, M, KK, MML, LIP, EXI, JREA,
     1 EPS, EMSI, UX, REIN, REA, NBOR)
      JSK = 0
      DO 35 I = 1, M
      S(I) = W(I, UX)
      ATEST = (S(I) - B(I)) * U(I,J)
      ABL = ABS (ATEST) - EPSI
      IF( ABL) 30, 30, 35
   30 JSK = JSK + 1
   35 CONTINUE
      IF( JSK - M) 37, 61, 37
   37 UORME = 0.0
      DO 51 I = 1, M
   51 UORME = UORME + (S(I) - B(I))**2
      DO 31 I = 1, M
      TEST = (S(I) - B(I)) / UORME
      IF(TEST) 32, 32, 33
   32 UV(I) = TEST/2.
      GO TO 34
   33 UV(I) = TEST * 1.0
   34 U(I, J+1) = U(I,J) + UV(I)
   31 CONTINUE
      DO 47 MMM = 1, 10
      KK = 2
      CALL POW (X, P, G, A, U, N, M, MML, LIP, EXI, JREA, E
     1 EMSI, UX, REIN, REA, NBOR)
      JSK = 0
      DO 38 I = 1, M
      T(I) = W(I, UX)
      IF (T(I)) - B(I)) 39, 39, 41
   39 IF( S(I) - B(I)) 42, 42, 47
   42 AL = (S (I) - B(I)) * U(I,J) - ( T(I) - B(I)) * U(I, JM)
      IF (AL) 53, 53, 44
   53 ALI = ( T(I) - B(I)) * U(I, J+1)
```

```
      IF(ABS(ALI)-EPSI)54,54,66
   54 JSK = JSK+1
      AU = 1.0
      UV(I) = UV(I)*E*AU
      GO TO 38
   66 IF(ABS(T(I)-B(I))-EPSI)54,54,43
   43 AU = 1.0
      UV(I) = UV(I)*C*AU
      GO TO 38
   41 IF(S(I)-S(I))45,45,44
   45 ALE(S(I)-B(I))55,55,56
   56 AU = 1.0
      UV(I) = UV(I)*D*AU
      GO TO 38
   55 AU = -1.0
      UV(I) = UV(I)*D*AU
      GO TO 38
   67 AL = (T(I)-B(I))*U(I,J+1)+(S(I)-B(I))*U(I,J)
      IF(AL) 55,68,68
   68 ALI = (T(I)-B(I))*U(I,J+1)
      IF(ABS(ALI)-EPSI)69,69,70
   69 AU = -1.
      JSK = JSK+1
      UV(I) = UV(I)*E*AU
      GO TO 38
   70 IF(ABS(T(I)-B(I))-EPSI)69,69,57
   57 AU = -1.0
      UV(I) = UV(I)*C*AU
   38 CONTINUE
      IF(JSK-M)59,61,59
   59 DO 49 I = 1,M
      S(I) = T(I)
      U(I,J) = U(I,J+1)
   49 U(I,J+1) = U(I,J)+UV(I)
      WRITE(3,9)((U(I,J),UV(I),J=1,M),I=1,M)
    9 FORMAT(10X,8(F8.4,5X)//)
   47 CONTINUE
      KK = 2
   61 CALL POW(X,P,Q,A,U,N,M,KK,MML,LIP,EXI,JRE
     1 EPS,EMSI,UX,REIN,REA,NBOR)
      WRITE(3,71)(UX(I),I=1,M)
   71 FORMAT(40X,F10.5///)
      DO 73 I=1,M
      ECART = W(I,UX)
      WRITE(3,74)  I, ECART
   74 FORMAT(40X,I2,5X,F10.5///)
   73 CONTINUE
      CALL EXIT
      END
// XEQ.
```