

CONFIDENTIEL

N-746

LECTURE AUTOMATIQUE DES
GELS D'ELECTROPHORESE

rapport technique

S. BEUCHER

FONTAINEBLEAU

FEVRIER 1982

INTRODUCTION

Le présent rapport a pour objet de décrire de façon détaillée l'algorithme de détection des taches d'électrophorèse, dont les résultats sont présentés dans le rapport de faisabilité intitulé "Analyse automatique de gels d'électrophorèse bidimensionnelle et Morphologie Mathématique". Ce programme a été écrit en FORTRAN pour l'analyseur de textures Leitz-TAS. On trouvera en annexe le listing du programme dans sa version actuelle, version qui ne comprend pas les améliorations qui seront mentionnées plus loin ainsi que la partie mesures.

ORDINOGRAMME DU PROGRAMME

Nous ne rappellerons pas ici les différents problèmes à résoudre. Une description détaillée en est donnée dans le rapport mentionné plus haut. La Figure 1 illustre l'aspect d'un gel d'électrophorèse.

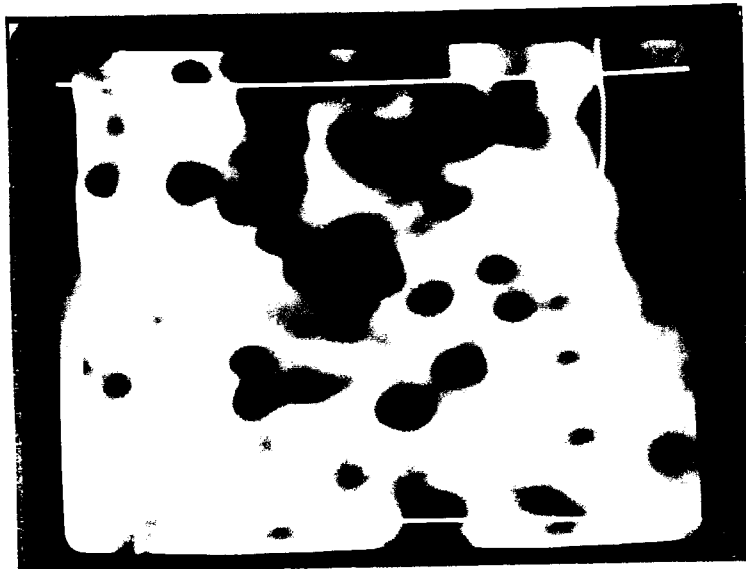


Figure 1 : Gel d'électrophorèse-image initiale

Nous désignerons par $f(x)$ la fonction de gris de l'image initiale au point x .

La lecture automatique des gels comprend deux parties : la détection automatique des taches et la mise en évidence de leur contour. On décrira les différentes étapes de ce traitement en donnant, s'il y a lieu, les améliorations possibles des algorithmes.

DETECTION DES TACHES

La détection des taches se fait de façon tout à fait classique par mise en évidence des minima de la fonction de gris. Cette méthode permet de séparer des taches qui ont tendance à se recouvrir, à condition que les minima restent bien marqués. En fait, les minima sont détectés sur la fonction fermée par un hexagone de taille i , $(f)^{iH}$, ceci afin de réduire considérablement le bruit impulsionnel. Cette façon de faire malheureusement déprécie la détection (Figure 2).



Figure 2 : Minima de la fonction.

Ces minima sont ensuite amincis par un élément structurant
. 0
1 1 0 de façon à faire apparaître les centroïdes (Figure 3).
. 0



Figure 3 : Centroïdes

Divers algorithmes ont été testés afin d'améliorer la détection des minima. L'un des procédés utilisés est le suivant. Il consiste, lors de la mise en évidence des minima à travailler non sur la fonction $(f)^{iH}$, mais sur la fonction dont les seuils successifs sont les zones d'influence des composantes connexes des érodés de taille i des seuils de la fonction $(f)^{iH}$. La figure 4 explicite la transformation.

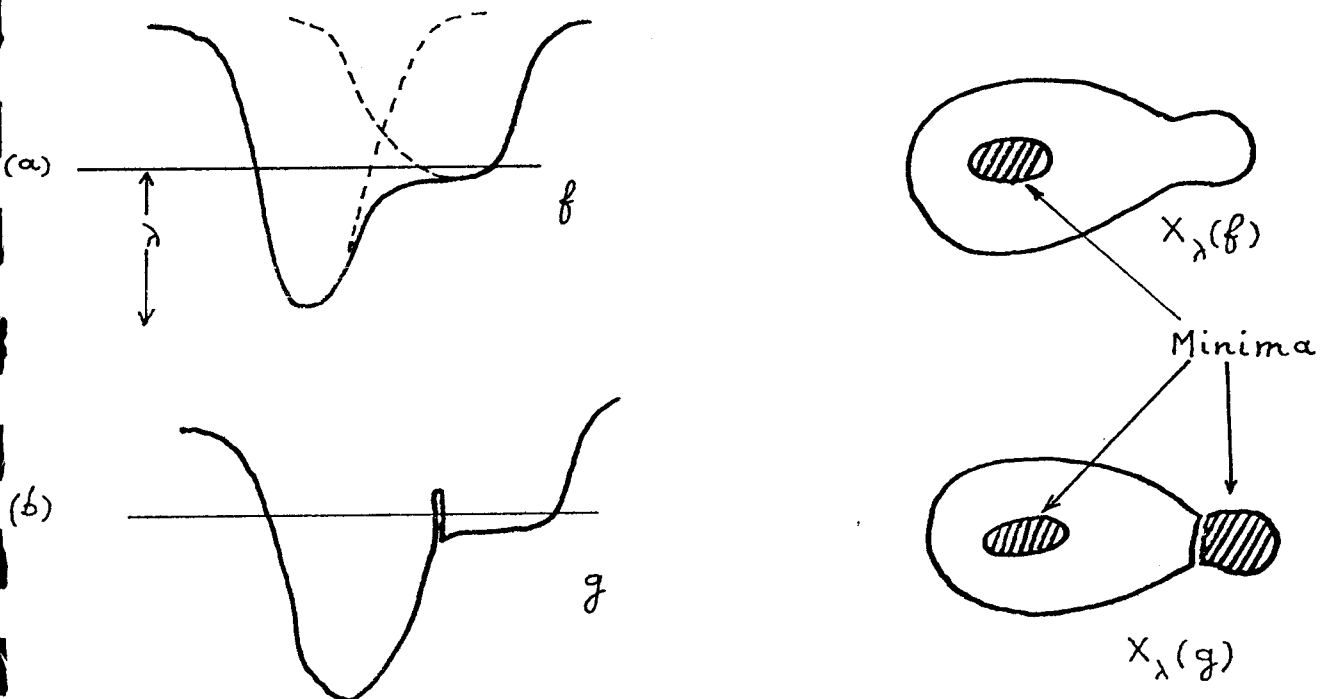


Figure 4 : Détection des Minima :
a) sur la fonction $(f)^{iH}$
b) sur la fonction g

En fait, l'expérience a montré que ces procédés n'étaient pas significativement meilleurs que la détection classique en ce sens que, s'ils font bien apparaître davantage de minima, ces derniers ne sont pas toujours exacts.

Pour cette raison, l'algorithme classique a été conservé. Une meilleure détection serait certainement obtenue à partir d'images filtrées par sommation.

DETECTION DES CONTOURS

Désormais nous désignerons par g la fonction de gris filtrée. Cette fonction sera appelée fonction initiale.

La détection des contours utilise une version sophistiquée de l'algorithme de lignes de partage des eaux de la fonction gradient. Donnons d'abord la définition de ce que nous entendons par contour des taches : Ce contour est constituée de l'ensemble des points d'inflexion de la fonction initiale (Figure 5).

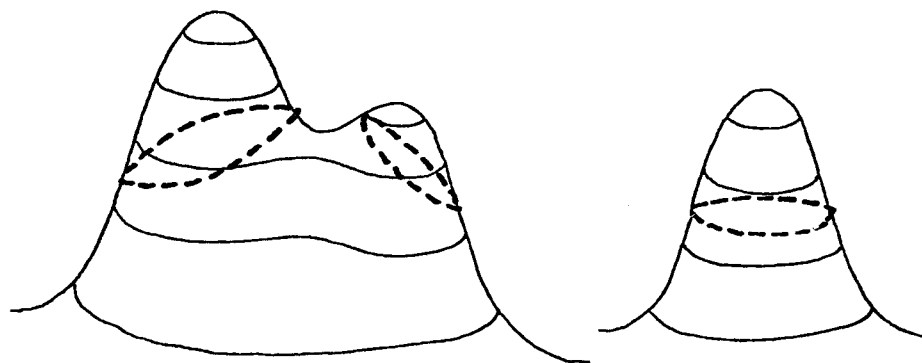


Figure 5 : Contours des taches (en pointillés)

Cette définition implique qu'à chaque tache lui corresponde un contour fermé et que tous les contours soient séparés, même lorsque les taches se recouvrent partiellement.

Pour mettre en évidence ces contours, nous allons utiliser l'information contenue dans la fonction de gris initiale, en la résumant.

Ce résumé de l'information disponible se fera par l'intermédiaire de deux transformations : Les Minima de la fonction, transformation qui nous a servi à mettre en évidence les taches, et la ligne de partage des eaux de cette fonction initiale (Figure 6).

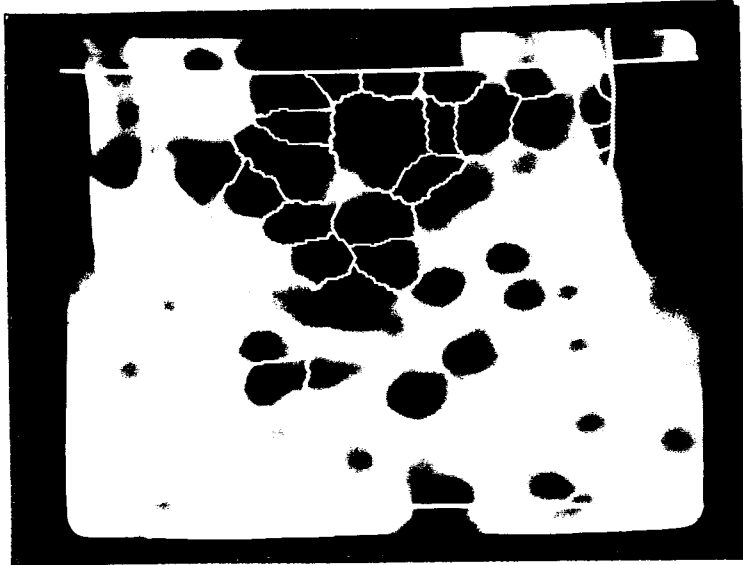


Figure 6 : Ligne de partage des eaux de la fonction initiale.

Nous désignerons par $M(g)$, l'ensemble des minima de la fonction et par $LPE(g)$, les lignes de partage des eaux de g . La figure 7 représente ces différents ensembles remarquables du graphe de la fonction g (celle-ci a été inversée, de telle sorte que les minima sont devenus des maxima, et les lignes de partage des eaux, des lignes de cols). Il apparaît clairement sur cette figure que les lignes de partage des eaux segmentent le graphe en différentes composantes, chacune d'elle consistant en une tache ou encore un dôme de la surface topographique dessinée par le graphe de g .

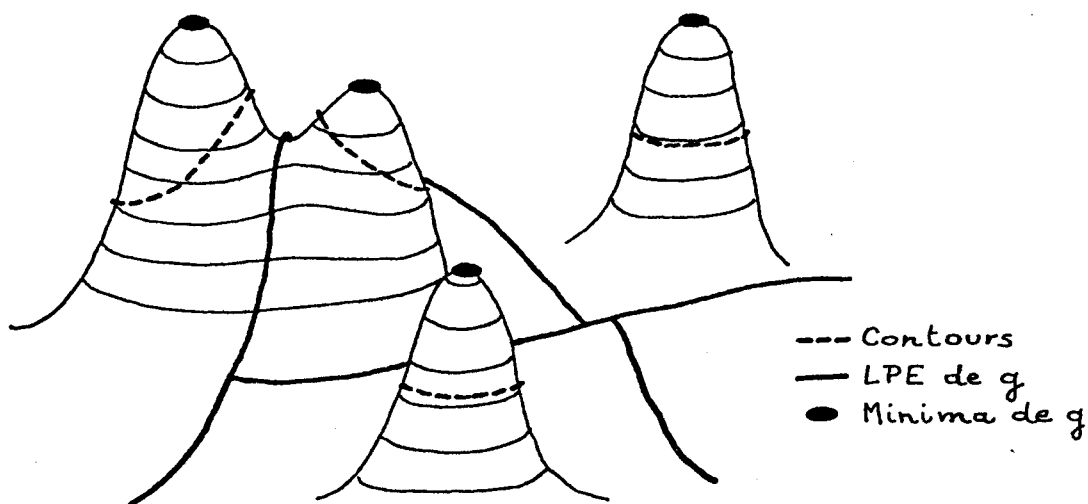


Figure 7 : lignes remarquables du graphe de $-g$

A chaque composante connexe, ou bassin versant, correspond donc une tache et une seule, et par là même un seul contour. De plus ce contour est toujours situé entre le minimum correspondant à la tache et la ligne de partage des eaux qui l'entoure (Figure 8).



Figure 8 : Image initiale, minima et LPE (la présence du contour entre le minimum et la ligne de partage des eaux est particulièrement nette pour les taches isolées)

Le contour est donc un lacet inclus dans le complémentaire de l'ensemble $(M(g) \cup LPE(g))$. Ce contour et l'ensemble $(M(g) \cup LPE(g))^c$ en particulier ont mêmes relations d'homotopie. L'algorithme de mise

en évidence du contour doit donc préserver l'homotopie de l'ensemble $(M(g) \cup LPE(g))^c$. Nous savons donc que le contour appartient à cet ensemble. Il nous reste à le positionner plus exactement. Pour ce faire, le module du gradient de g va nous être très utile.

Le contour correspond en effet aux zones à fort gradient. La transformation utilisée sera un amincissement géodésique de l'ensemble $(M(g) \cup LPE(g))^c$. Les éléments structurants utilisés seront le couple $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ de la transformation ligne de partage des eaux. Cette ligne de partage des eaux du gradient est un peu particulière en ce sens qu'on n'autorise qu'un nombre restreint de minima à "inonder" le graphe de $|\vec{\text{grad}} g|$. (En l'occurrence, les minima du gradient autorisés sont les minima de la fonction g d'une part et un minimum connexe représenté par la ligne de partage des eaux de g).

En désignant par :

$$X_i(|\vec{\text{grad}} g|) = \{ x : |\vec{\text{grad}} g|(x) \geq i \}$$

les seuils successifs du gradient, l'algorithme de détection du contour peut s'écrire :

$$Y_0 = (M(g) \cup LPE(g))^c$$

$$Y_{i+1} = [Y_i \circ \{T\} ; X_i(|\vec{\text{grad}} g|)]$$

Les ensembles Y_i constituent une suite décroissante qui converge vers le contour des taches. La Figure 9 explicite l'algorithme, tandis que la Figure 10 illustre le résultat obtenu sur une image d'électrophorèse.

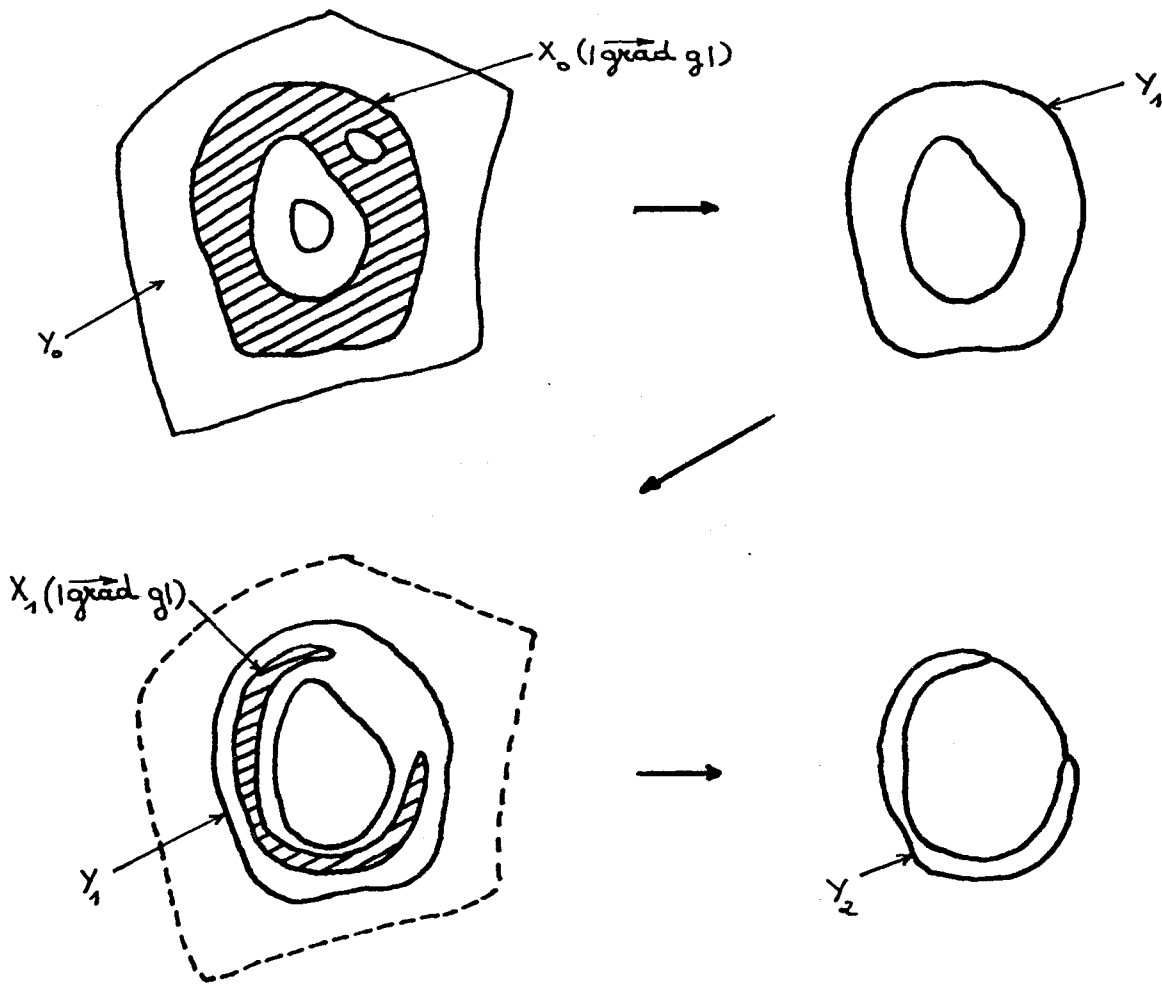


Figure 9 : Algorithme de détection de contour

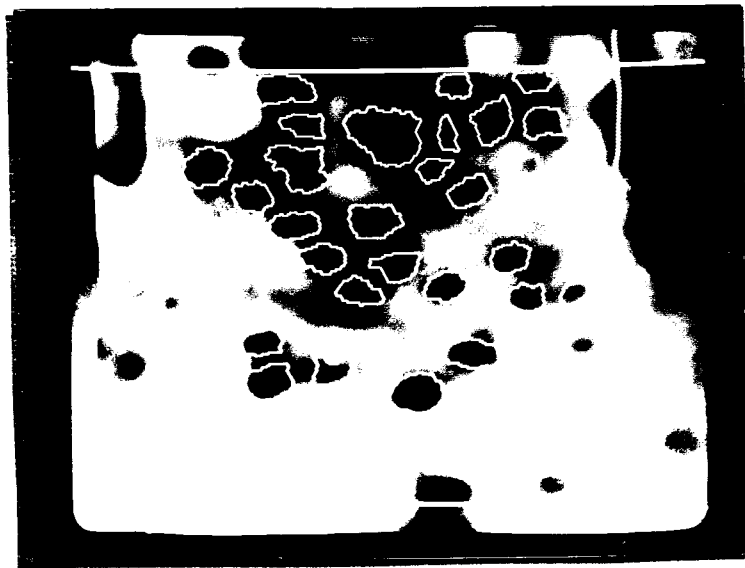


Figure 10 : Contours des taches sur un gel d'électrophorèse

Les contours définis précédemment délimitent pour chaque tache un masque de mesure trop petit. Les mesures densitométriques des taches risquent d'y être légèrement biaisées. Il est néanmoins possible d'étendre le masque sur la périphérie de la tache, à l'aide d'une procédure prenant en compte les valeurs du gradient le long du contour. Cette procédure consisterait à étendre le masque autour de chaque point du contour en raison inverse du module du gradient en ce point. (Figure 11).

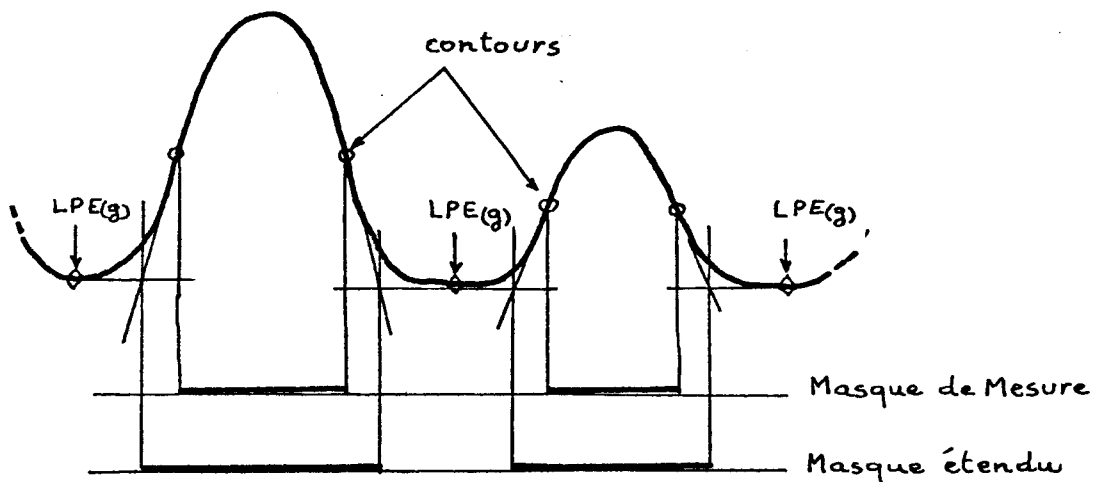


Figure 11 : Schéma de principe de l'extension du masque de mesure à la totalité de la tache.

Faute de temps, cet algorithme n'a pu être testé.

CONCLUSIONS PROVISOIRES

Cette première version du programme est loin d'être complète. De nombreuses procédures doivent être ajoutées et des améliorations importantes méritent d'être apportées aux routines existantes. Le lecteur est prié de se référer à l'étude citée en introduction pour avoir une revue plus complète des problèmes restant à résoudre. Néanmoins, ce programme constitue une base de traitement assez intéressante car il apporte des solutions prometteuses au problème le plus important à résoudre dans une première étape : la mise en évidence des taches de protéines.

Cette pré-étude va même plus loin, puisqu'elle a permis d'élaborer un algorithme de détection de contours et de segmentation d'image très général, ouvrant la voie à de nombreuses applications.

REFERENCE

*Ce rapport accompagne le rapport intitulé :
"Analyse automatique de gels d'électrophorèse bidimensionnelle et
Morphologie Mathématique".*

A N N E X E

Listing FORTRAN du programme PHORCO, V 1.0

```
0001      PROGRAM PHORCO
C
C      *****
C      * LECTURE AUTOMATIQUE D'ELECTROPHORESES *
C      * BI-DIMENSIONNELLES.                  *
C      *                                       *
C      *   VERSION 1.0 - FEVRIER 1982       *
C      *****
C
C      CE PROGRAMME DETECTE LES TACHES DE PROTEINES,
C      DETERMINE LEUR CENTRE ET LEUR CONTOUR.
C
0002      DIMENSION ISEUIL(10)
C      ####INITIALISATIONS
0003      CALL INITAS
0004      CALL HRDWR("106")
0005      CALL RESET
0006      CALL CTAS
0007      CALL CLMEM
C      ####PARAMETRE DU FILTRAGE
0008      TYPE 100
0009      100  FORMAT(1H#, 'FILTRE-OUVERTURE ?')
0010      ACCEPT 11, ISIZE
0011      11  FORMAT(I2)
C      ####VALEURS DE SEUIL DU GRADIENT
0012      DO 23 I=1,4
0013      ISEUIL(I)=I
0014      23  CONTINUE
0015      DO 27 I=5,10
0016      ISEUIL(I)=2*I-4
0017      27  CONTINUE
C      ####LIGNE DE PARTAGE DES EAUX DE LA FONCTION
C      ET RECHERCHE DES MINIMA.
0018      DO 15 I=1,100
0019      CALL SYNCR(0,0)
0020      CALL ALUSG(2,2,2)
0021      J=I
0022      CALL DEFG(7,3,2,ISIZE)
0023      CALL DNSA(0,J)
0024      CALL MOVE(0,1,1,0)
0025      CALL CLDE
0026      CALL IAU(3,2,1)
0027      15  CONTINUE
0028      CALL ALUSG(7,2,5)
0029      CALL MOVE(0,1,1,0)
C      ####CENTROIDES
0030      CALL CMEMR(4)
0031      CALL THIR(7,4,0,"35","34")
0032      CALL ALUSG(3,3,4)
0033      CALL MOVE(-1,1,1,0)
0034      CALL CMEMR(3)
C      ####VISUALISATIONS :
C      -MEMOIRE 5 : MINIMA
C      -MEMOIRE 4 : LPE FONCTION;
```

```
C      -MEMOIRE 7 : CENTROIDES
0035 CALL DPLSG(5,2,0)
0036 PAUSE 'MINIMA'
0037 CALL DPLSG(7,2,0)
0038 PAUSE 'CENTROIDES'
0039 CALL DPLSG(4,2,0)
0040 PAUSE 'ZONES INFLUENCE'
0041 CALL DEFO(7,1,3,1)
0042 CALL ALUSG(4,2,6)
0043 CALL MOVE(0,1,1,0)
0044 CALL CLDE
0045 CALL ALUCX(5,6,5,3)
0046 CALL MOVE(0,1,1,0)
0047 CALL DPLCX(5,4,4,0)
0048 PAUSE 'LPE FUNCTION + MINIMA'
C      ####ELABORATION MASQUE
0049 CALL ALUCX(5,4,3,5)
0050 CALL MOVE(0,1,1,0)
C      ####AMINCISSEMENT GEODESIQUE
0051 DO 30 I=1,10
0052     ITT=ISEUIL(I)
0053     IF (ITT-16)44,55,55
0054 44    CALL GRAD(1,0,2,ITT)
0055     GO TO 66
0056 55    CALL CMEMR(2)
0057 66    CALL ALUCX(2,3,2,0)
0058     CALL MOVE(0,1,1,0)
0059     CALL THIR(3,2,0,27,3)
0060     CALL THIR(3,2,0,15,15)
0061 30    CONTINUE
C      ####VISUALISATION CONTOURS (MEMOIRE 3)
0062 CALL DPLSG(3,2,0)
0063 PAUSE 'CONTOURS'
0064 STOP
0065 END
```

```
0001      SUBROUTINE IAU(IW,IM,IT)
C
C
C *****
C * SUBROUTINE LIGNE DE PARTAGE DES *
C * EAUX *
C *****
C
C      CETTE ROUTINE EFFECTUE LA LIGNE DE PARTAGE
C      DES EAUX DE LA FONCTION DONT LES SEUILS SONT
C      STOCKES DANS IM.IW CONTIENT LA LPE EN
C      FORMATION.IT EST UNE MEMOIRE DE TRAVAIL.
C      IM ET IW DOIVENT ETRE CONNECTABLES,AINSI
C      QUE IM ET IT.
C
0002      JRW=23
0003      JRB=7
0004      CALL THIR(IW,IM,1,JRW,JRB)
0005      JRW="17
0006      JRB=0
0007      CALL THIR(IW,IM,1,JRW,JRB)
0008      CALL MOVSD(IW,IT)
0009      CALL BLOW1(IM,IW)
0010      CALL ALUCX(IM,IW,IM,3)
0011      CALL MOVE(-1,1,1,0)
0012      CALL ALUSG(IM,2,6)
0013      CALL MOVE(0,1,1,0)
0014      CALL ALUCX(6,7,7,4)
0015      CALL MOVE(0,1,1,0)
0016      CALL ALUCX(IM,IT,IW,4)
0017      CALL MOVE(-1,1,1,0)
0018      RETURN
0019      END
```

```
0001      SUBROUTINE THIR(IW, IM, ITY, JRW, JRB)
C
C
C      *****
C      * SUBROUTINE AMINCISSEMENT/EPAISSISSEMENT *
C      * GEODESIQUE                               *
C      *****
C
C      CETTE ROUTINE EFFECTUE L'EPAISSISSEMENT (ITY=1)
C      OU L'AMINCISSEMENT (ITY=0) DE LA MEMOIRE IW
C      A L'INTERIEUR (OU L'EXTERIEUR) DU MASQUE IM.
C      LA CONFIGURATION EST CODEE PAR JRW ET JRB.
C      IW ET IM DOIVENT ETRE CONNECTABLES.
C
0002      I1="600
0003      I2=4
0004      IRW=JRW
0005      IRB=JRB
0006      IF(ITY-1)2,1,2
0007 1      I1="200
0008      I2=0
0009 2      CALL ALUSG(IW,2,IW)
0010      IF(IM)3,4,4
0011 4      CALL ALUCX(IW,IM,IW,I2)
0012      CALL MELO(0,1,R1)
0013      CALL DEFO(7,3,0,1)
0014 5      R2=R1
0015      DO 6 I=1,5
0016 6      CALL ROT(IRW,IRB,I1)
0017      CALL ROT(IRW,IRB,I1)
0018 3      CALL AREA(R1)
0019      IF(R1-R2)5,7,5
0020 7      CALL CLDE
0021      CALL CLRI
0022      CALL MOVE(-1,1,1,0)
0023      RETURN
0024      END
```



```
001      SUBROUTINE ROT(IRW,IRB,I1)
C
C
C      *****
C      * ROUTINE ROTATION *
C      *****
C          ROUTINE DE SERVICE PERMETTANT
C          DE FAIRE TOURNER AUTOMATIQUEMENT LES
C          CONFIGURATIONS UTILISEES DANS LA
C          LOGIQUE DE CONTOUR.
C
002      CALL RING(IRW,IRB,0,I1)
003      CALL MOVE(0,1,1,0)
004      IRW=IRW+IRW
005      IF(IRW-64)2,1,1
006      1  IRW=IRW-63
007      2  IRB=IRB+IRB
008      IF(IRB-64)3,4,4
009      4  IRB=IRB-63
010      3  RETURN
011      END
```

0001 SUBROUTINE GRAD(IM1,IM2,IM3,IVAL)

C
C
C
C
C
C
C
C
C
C
C
C

* SUBROUTINE GRADIENT *

CETTE ROUTINE EFFECTUE L'ALGORITHME
CLASSIQUE. IM1 ET IM2 SONT LES MEMOIRES
CONNECTABLES DE STOCKAGE DES SEUILS.
IM3 (CONNECTABLE AVEC IM1) CONTIENT
LE SEUIL IVAL DE GRADIENT.

0002 CALL CMEMR(IM3)
0003 CALL SYNCR(0,0)
0004 ITH=99-IVAL
0005 DO 20 I=1,ITH
0006 IPRIM=I
0007 CALL DNSA(0,IPRIM)
0008 CALL LGEX(IM1)
0009 CALL SYNCR(0,0)
0010 IPRIM=I+IVAL
0011 CALL DNSA(IPRIM,0)
0012 CALL LGEX(IM2)
0013 CALL DILME(IM1)
0014 CALL DILME(IM2)
0015 CALL ALUCX(IM1,IM2,IM1,0)
0016 CALL MOVE(-1,1,1,0)
0017 CALL ALUCX(IM1,IM3,IM3,4)
0018 CALL MOVE(-1,1,0,0)
0019 20 CONTINUE
0020 CALL SYNCR(1,0)
0021 RETURN
0022 END

C
C

#####FIN PROGRAMME