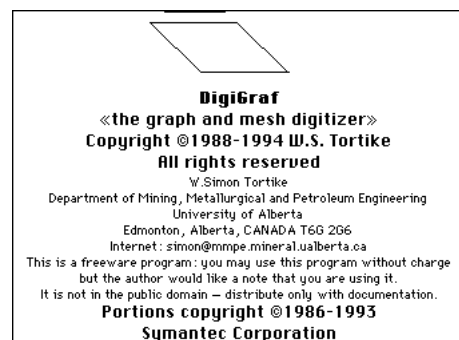


MapUtil

MapUtil : CornerScaling.....2
 MapUtil : GetPictSize.....6
 MapUtil : XYUtil->QuadratLattice.....10

Interfaces avec :



MapUtil : CornerScaling



Utilitaire de transformation de fichiers de coordonnées.



On suppose ici qu'on connaît les coordonnées de n points dans l'espace dans un fichier binaire (système utilisateur). On possède un fond de carte contenant deux points de référence dont on connaît les coordonnées dans l'espace dans un fichier binaire (système utilisateur). L'option donne un fichier de coordonnées des n points adaptées au fond de carte (système écran).



L'option utilise une seule fenêtre de dialogue :



Nom du fichier de fond de carte.



Nom du fichier binaire de digitalisation des deux points de référence.



Nom du fichier binaire des coordonnées des deux points de référence.



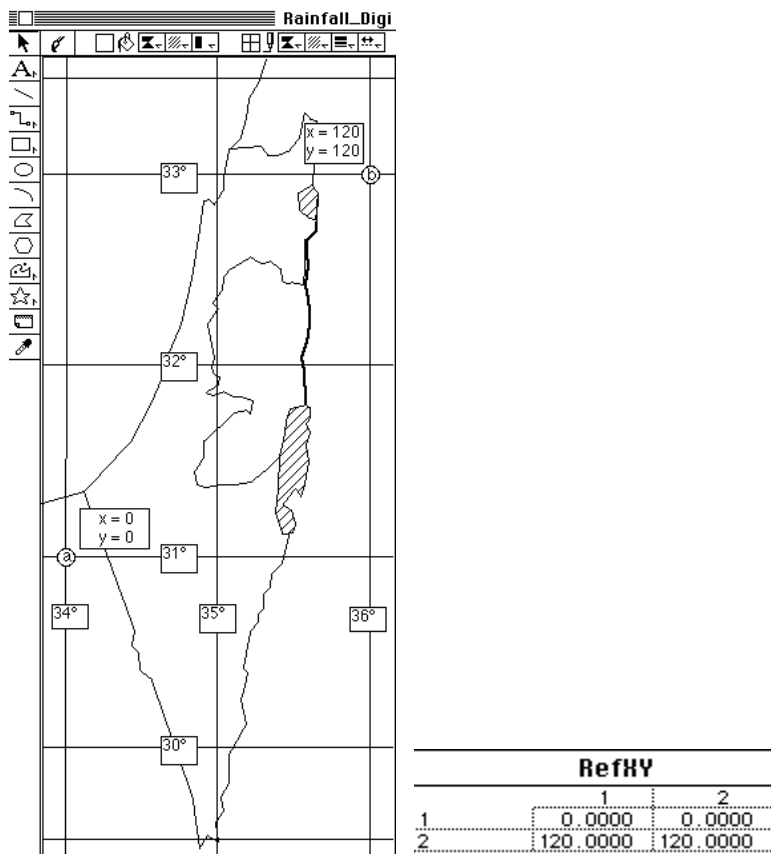
Nom du fichier binaire des coordonnées des n points à transformer.



Utiliser le dossier de travail créé par la carte Rainfall de la pile ADE-4•Data¹. L'article de K.R. Gabriel donne les coordonnées (longitude, latitude) de 55 stations météorologiques. Pour obtenir facilement une carte de ces stations on utilise la latitude en minutes (à partir de 34° Est) et le longitude en minutes (à partir de 31° Nord) comme coordonnées dans l'espace.

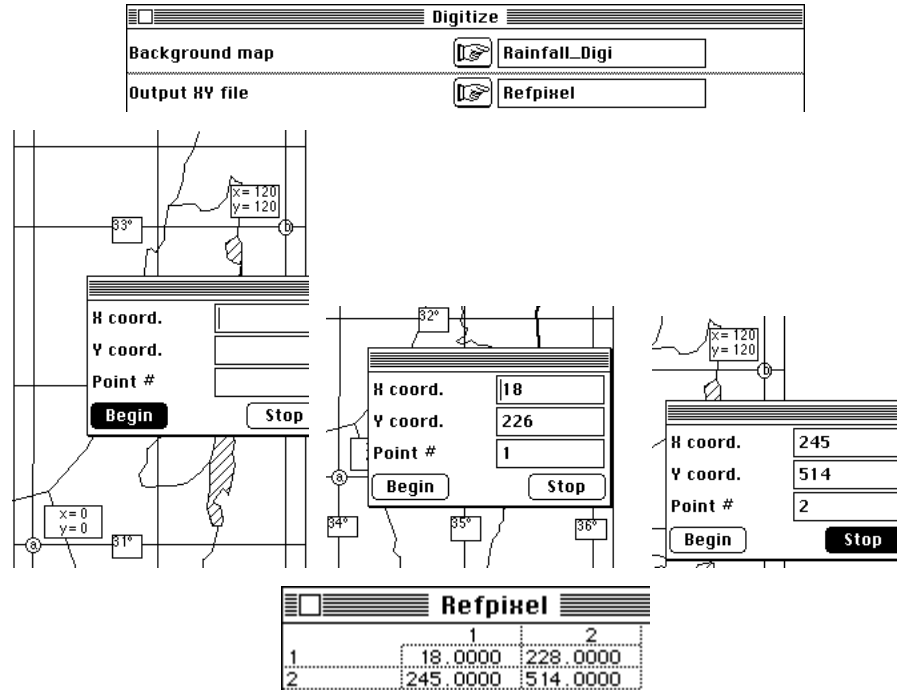
Un fond de carte est implanté dans un fichier Pict "Rainfall_Digi :

LatiLongi		
	1	2
1	66.0000	125.0000
2	66.0000	116.0000
3	59.0000	109.0000
4	59.0000	108.0000
5	65.0000	105.0000
6	56.0000	102.0000
7	57.0000	94.0000
8	55.0000	86.0000
9	46.0000	66.0000
10	54.0000	60.0000
11	50.0000	51.0000
12	44.0000	44.0000
13	36.0000	35.0000
14	28.0000	30.0000
15	18.0000	21.0000
16	94.0000	135.0000
17	89.0000	123.0000
18	76.0000	121.0000
19	90.0000	119.0000
20	90.0000	107.0000
21	78.0000	103.0000
22	84.0000	102.0000
23	71.0000	100.0000
24	68.0000	97.0000
25	77.0000	96.0000
26	84.0000	93.0000
27	86.0000	91.0000
28	an nnnn	an nnnn

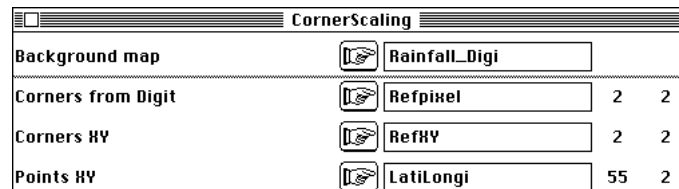


Pour caler les stations sur ce fond élémentaire, deux points repères sont placés. Le premier correspond à 34° Est et 31° Nord soit (0,0). Le second correspond à 36° Est et 33° Nord soit en minutes (120, 120). Ces valeurs sont consignées dans le fichier binaire RefXY. Le fichier LatiLongi contient les coordonnées des 55 stations, le fichier RefXY contient les coordonnées de deux points de référence.

Il suffit de digitaliser les deux points de référence pour faire le transfert d'échelle (Digit : Digitalize) :



Exécuter le changement d'échelle :



Le fichier LatiLongi_XY est créé :

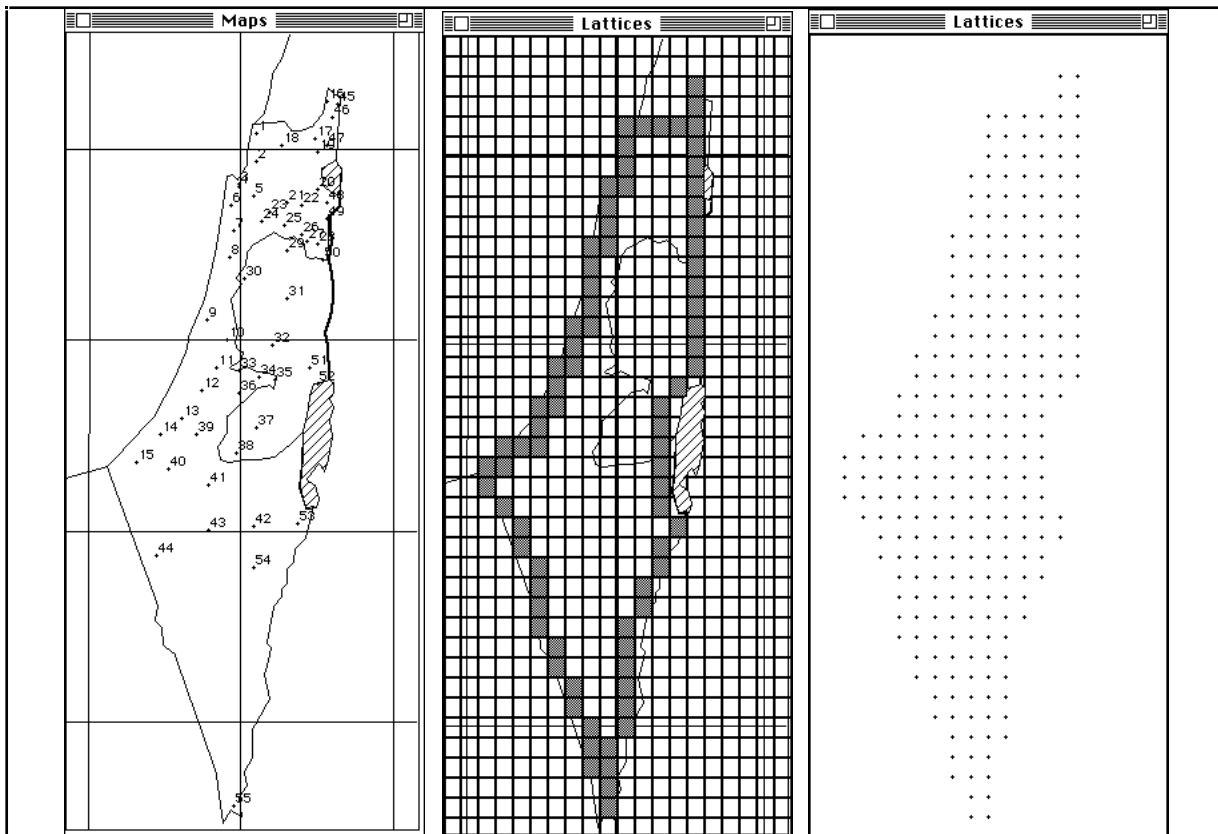
	1	2
1	142.8500	525.9167
2	142.8500	504.4667
3	129.6083	487.7833
4	129.6083	485.4000
5	140.9583	478.2500
6	123.9333	471.1000
7	129.8250	452.0333
8	122.0417	432.9667
9	105.0167	385.3000
10	120.1500	371.0000

Vérifier avec un carte par étiquettes (Maps : Labels) :



Implanter une grille par (Lattices : Create_Bkgnd) :

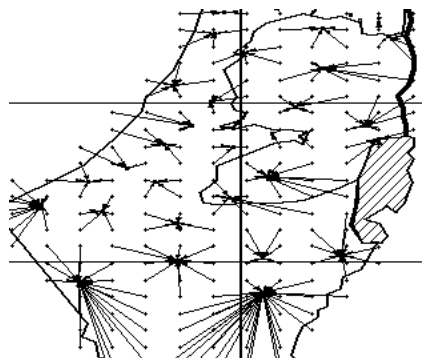




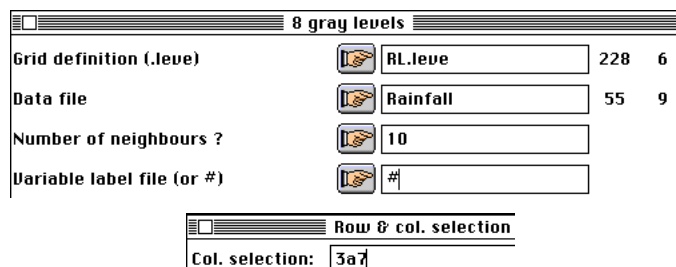
Préparer le passage au module de cartographie (Lattices : LattiToLevel) :

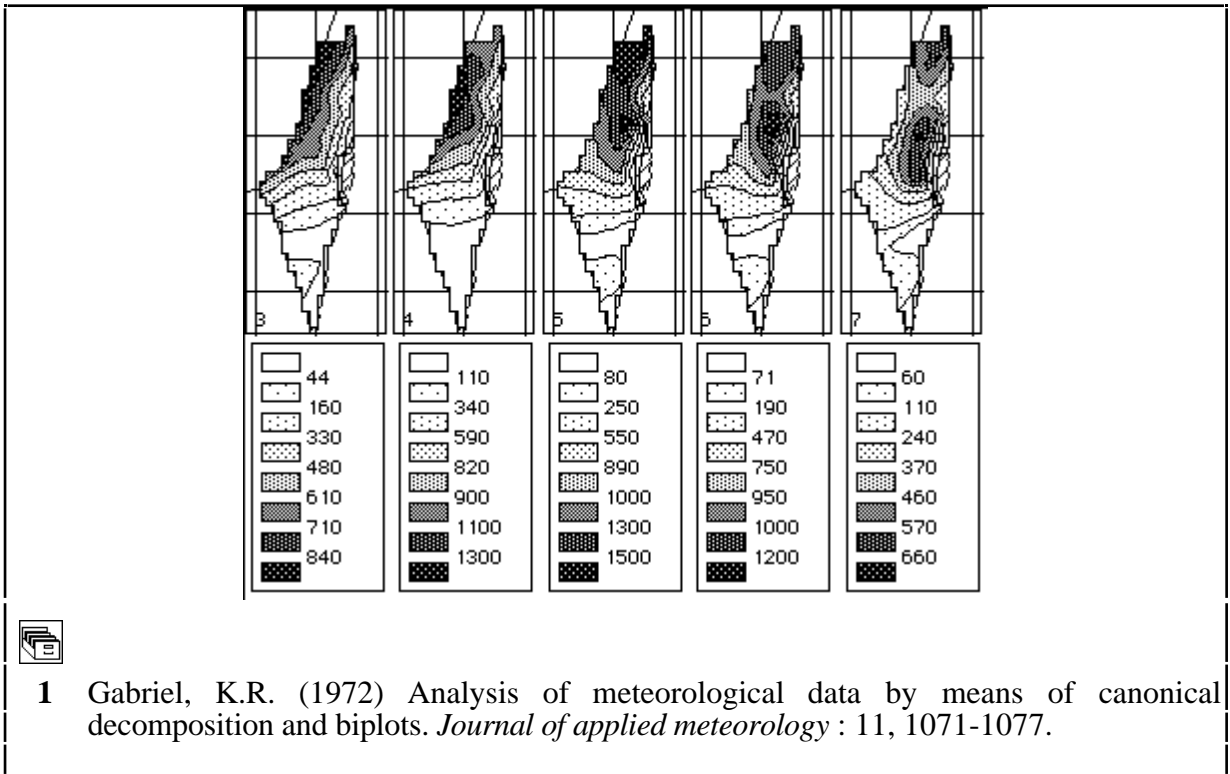


Connecter coordonnées des stations et grille de quadrats (Levels : Prepare) :



Cartographier les données (Levels : 8 gray levels) :





MapUtil : GetPictSize



Utilitaire de lecture d'une figure dans un fichier PICT.



Les tracés sur fond de carte (Maps : Labels, Maps : Values, Maps : Neighborhood Graph, Levels : 8 gray levels) utilisent un fichier de coordonnées des points de référence qui sont acquis par la fonction de digitalisation. Digit : Digitalize assure cette fonction mais on trouve des logiciels freeware qui font cela très efficacement. Pour assurer l'interface, la seule difficulté concerne les échelles. La fenêtre physique (tracé à l'écran) et la fenêtre logique (échelle des abscisses et des ordonnées) doivent se correspondre. L'option donne les valeurs utiles.



L'option utilise une seule fenêtre de dialogue :



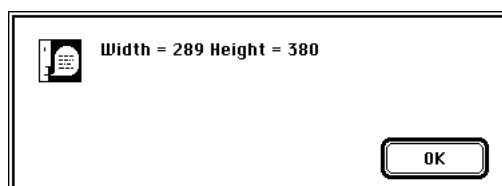
 Nom du fichier PICT d'entrée.



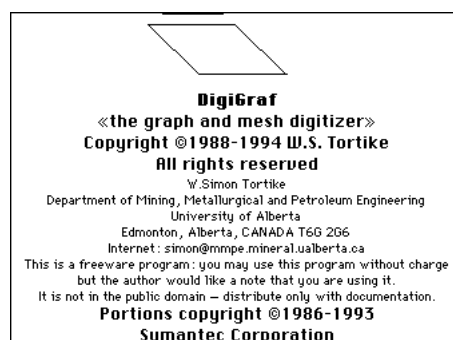
Utiliser l'exemple décrit dans la page de Digit : Digitalize :



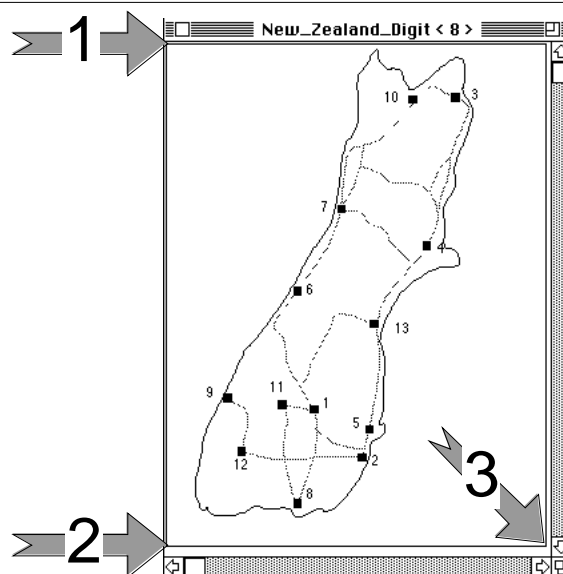
L'option affiche la figure et annonce simplement la largeur (289 pixels) et la hauteur (380 pixels) de la carte.



On peut se servir de ce renseignements pour la digitalisation de la carte par **DigiGraf** 1.4.4 de Simon Tortike. **DigiGraf** est officiellement disponible par FTP anonyme à l'adresse [mmpe.mineral.ualberta.ca](ftp://mmpe.mineral.ualberta.ca) dans le répertoire `ftp/pub/digigraf144.hqx`. Ce programme fonctionne sur deux fenêtre distinctes. Ouvrir d'abord le fond de carte avec un programme quelconque, ajuster la fenêtre à l'écran puis lancer **DigiGraf** lui-même :



La fenêtre de **DigiGraf** et la fenêtre du fond de carte sont indépendantes. Utiliser l'option Define axes... du menu **Data** et cliquer, comme il est indiqué au maximum de l'axe des Y, à l'origine puis au maximum de l'axe des X :



La fenêtre physique étant enregistrée, utiliser l'option Axes Scales du même menu et passer dans la fenêtre de dialogue les paramètres x min = 0, x max = 289, y min = 0 et y max = 380. La fenêtre de DigiGraph affiche alors :

```
/* Put your comments between these delimiters */

x-scale ranges from 0 to 289
x-scale is linear
y-scale ranges from 0 to 380
y-scale is linear
```

Utiliser l'option Add data... du menu Data, cliquer sur chaque point puis après l'enregistrement du dernier taper Command-Maj-point pour terminer. La fenêtre affiche alors :

Point no.	scaled x	scaled y
1	111.54	104.10
2	149.06	66.70
3	220.05	339.57
4	197.74	226.38
5	155.15	87.93
6	98.36	193.03
7	132.84	254.68
8	99.38	32.34
9	45.63	113.19
10	187.60	338.56
11	89.24	107.13
12	56.79	70.74
13	158.19	168.78

Sauvegarder le texte du résultat, l'ouvrir avec BBEdit pour enlever la partie non numérique et passer le résultat en binaire pour obtenir le fichier de coordonnées. Les résultats entre la digitalisation ADE-4 et celle de **DigiGraf** ne sont pas identiques mais les nuances ne dépassent pas quelques pixels.



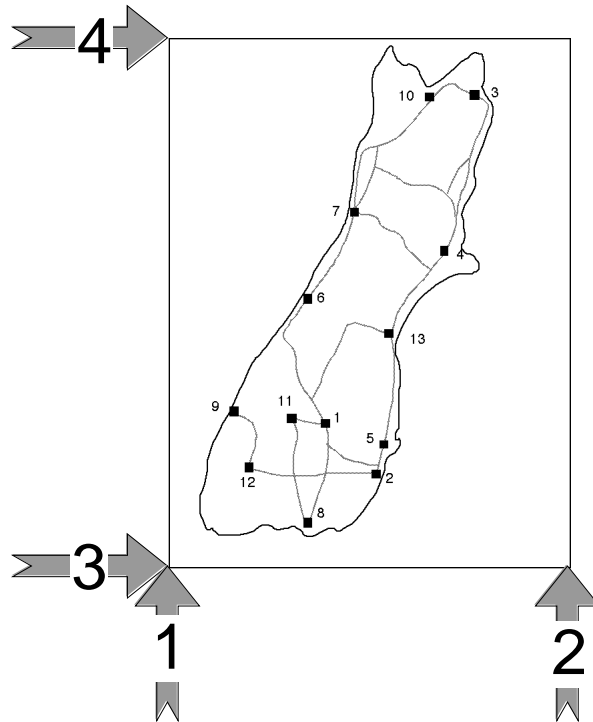
On peut se servir pour la digitalisation de la carte de **dataThief** 2.0 de Kees Huyser et Jan van der Loan. Ce programme est disponible sur Internet à l'adresse

<http://www.nikhefk.nikhef.nl/~keeshu/dataThief.sit.hqx>

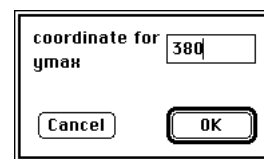
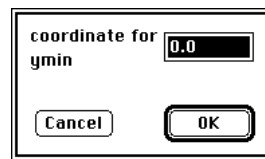
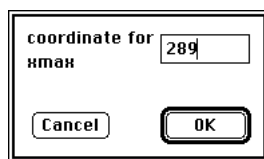
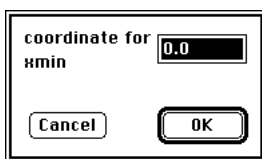
Ouvrir le programme :



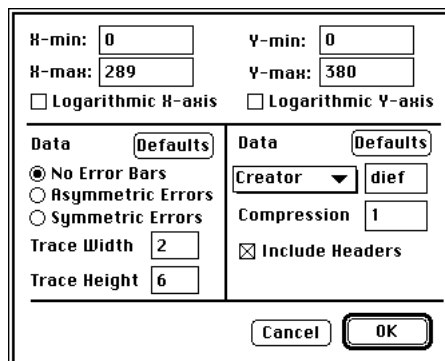
Ouvrir le fichier “fond de cartes”. Le programme affiche en pleine page le dessin. Passer les quatre paramètres en cliquant sur les bornes du dessin dans l’ordre :



Passer à chaque fois la valeur de la fenêtre logique correspondante :



Vérifier dans l’option options... du menu Option :



Cliquer alors sur chacun des point, une petite fenêtre affichant les coordonnées en permanence :

```

current
x: 1.38e+02
y: 1.37e+02
clicked
x: 1.13e+02
y: 1.04e+02

```

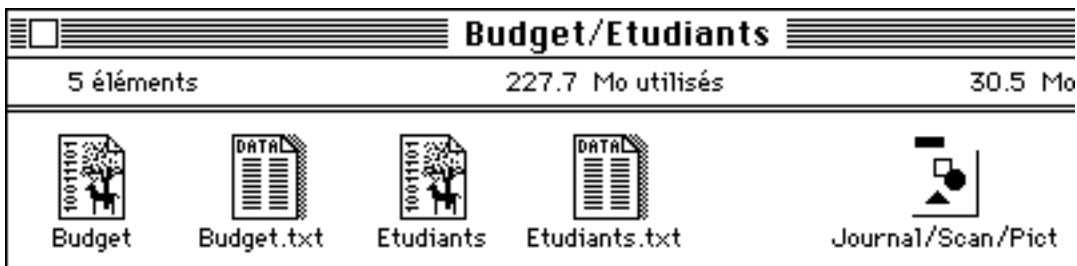
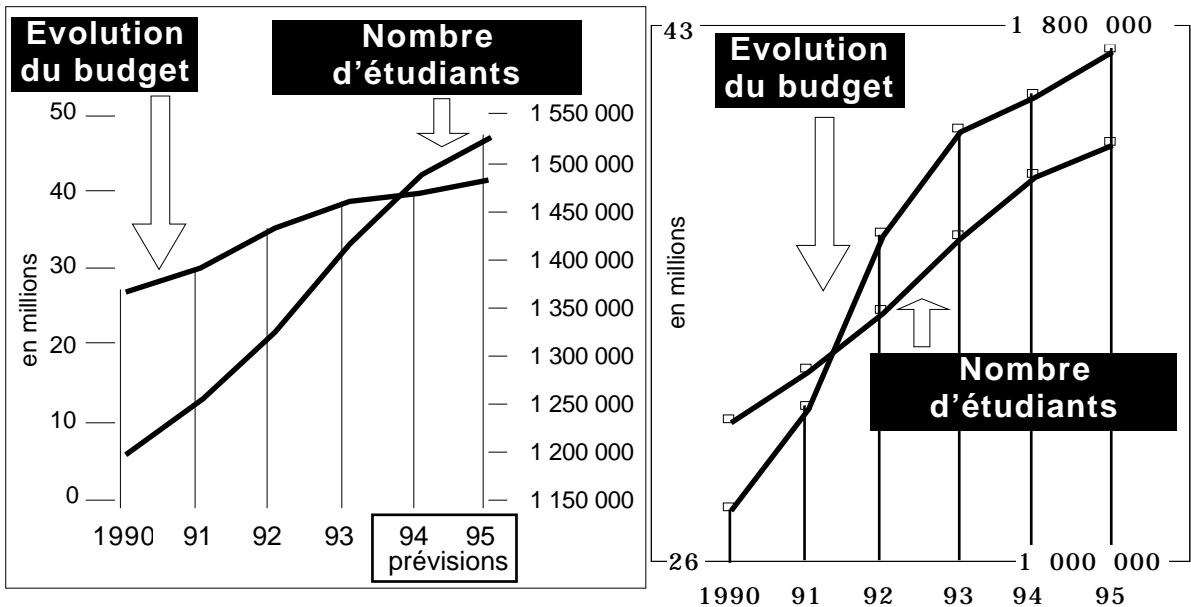
Enregistrer le résultat et vérifier le contenu du fichier de sortie :

Data stolen from file: New_Zealand_Digit
 The X-axis is linear from 0 to 289
 The Y-axis is linear from 0 to 380

X-value	Y-value	Delta Y[+	Delta Y-]
1.127e+02	1.037e+02		
1.498e+02	6.628e+01		
2.211e+02	3.394e+02		
...			
1.874e+02	3.381e+02		
8.863e+01	1.075e+02		
5.732e+01	7.217e+01		
1.585e+02	1.682e+02		

Utiliser BBEdit pour enlever la partie non numérique et passer le résultat en binaire pour obtenir le fichier de coordonnées. Ces outils puissants sont recommandés pour les digitalisations complexes. Le programme **dataThief** est particulièrement simple et efficace. Il a été écrit pour récupérer des valeurs sur des courbes scannées. Il permet un exercice amusant. Faire un dossier de travail avec la carte Budget/Étudiants de la pile ADE-4•Data.

Un journal sérieux a publié le graphique scanné dans le fichier Journal/Scan/Pict. Utiliser **dataThief** pour enregistrer les valeurs des deux courbes puis Scatters : Labels pour retracer le dessin :



Commenter sans excès.

MapUtil : XYUtil->QuadratLattice



Utilitaire de comptage de points par quadrat.



On suppose ici qu'on connaît les coordonnées de n points dans l'espace dans un fichier binaire (système utilisateur). On possède une grille de quadrats dont on connaît les coordonnées du rectangle de contour dans un fichier binaire (système utilisateur). L'option donne pour chaque point le numéro du quadrat auquel il appartient et pour chaque quadrat le nombre de points qu'il contient.

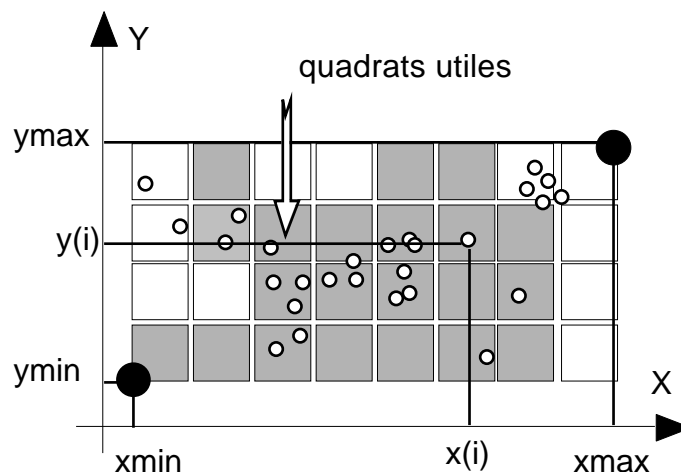


L'option utilise une seule fenêtre de dialogue :

.lat type file		<input type="text"/>
XY Corners		<input type="text"/>
XY Points		<input type="text"/>
Output file name		<input type="text"/>

Nom du fichier .lat de la grille de quadrats (voir la fiche de [Lattices](#)).

Nom du fichier binaire des coordonnées des deux points de référence. Ce fichier contient sur la première ligne les coordonnées x_{min} et y_{min} du sommet en bas et à gauche du quadrat en bas et à gauche de la grille. Il contient sur la deuxième ligne les coordonnées du sommet en haut et à droite du quadrat en haut et à droite de la grille.



Nom du fichier binaire des coordonnées des points à compter.

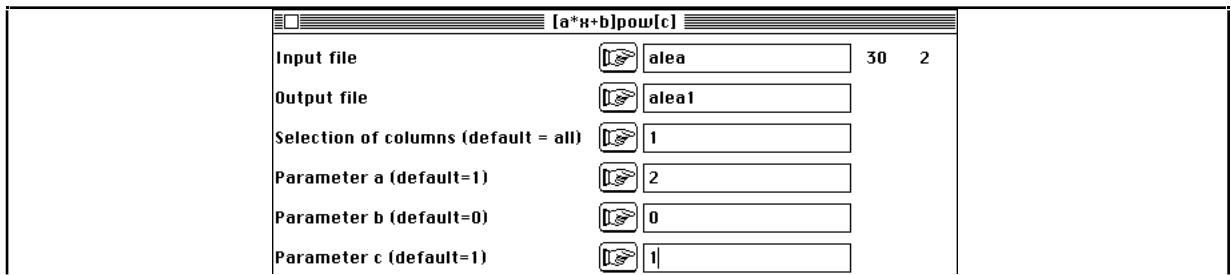
Nom générique des fichiers de sortie.



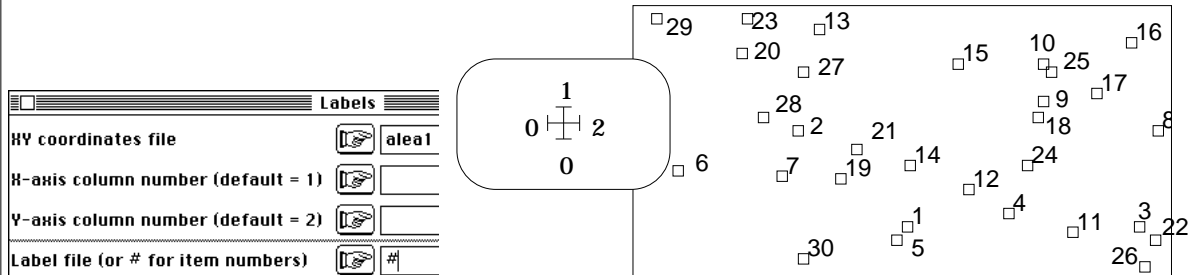
Utiliser le dossier de travail en cours. Générer un fichier de nombres aléatoires à 30 lignes et deux colonnes (TextToBin : CreateRandom) :

Output file name		<input type="text" value="alea"/>
Row number		<input type="text" value="30"/>
Column number		<input type="text" value="2"/>

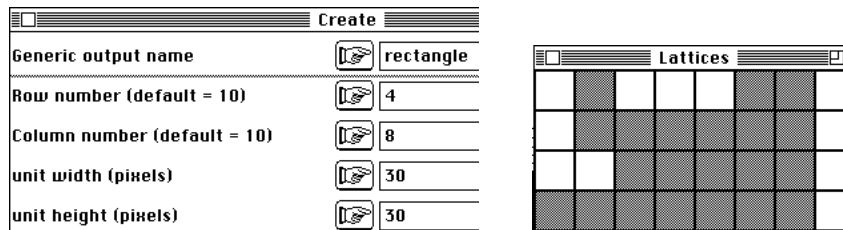
Multiplier par 2 la première coordonnée (Bin->Bin : $[a*x+b]pow[c]$) :



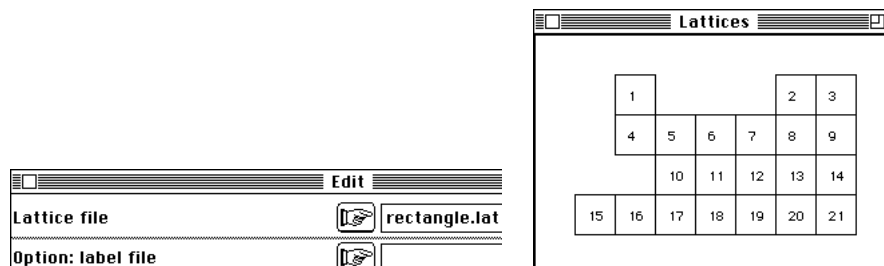
Dessiner le nuage des 30 points (Scatters : Labels) :



Implanter une grille à 4 lignes et 8 colonnes (Lattices : Create) :



Editer le résultat (Lattices : Edit) :



Noter le caractère très ennuyeux de la situation : la grille saisie à la souris est définie par des coordonnées en pixels et le nuage des points est connu en coordonnées utilisateurs. Pour ajuster les deux systèmes, vérifier le contenu des fichiers associés à la grille. Dans rectangle.lat on lit

```
6 (nombre de lignes de la matrice)
9 (nombre de colonnes de la matrice)
30.000000 (largeur d'un quadrat en pixels)
30.000000 (hauteur d'un quadrat en pixels)
```

Dans rectangle on trouve :

```
000000000
001000110
001111110
000111110
011111110
000000000
```

Il y a une bande de quadrats non sélectionnés tout autour de la figure utile. On peut les enlever ou ce qui va plus vite donner au sommet en bas à gauche l'abscisse -0.25 et l'ordonnée -0.25. En haut à droite, le sommet en haut à droite a les coordonnées 2.25 et 1.25. Mettre ces valeurs dans un fichier binaire (ADEBin : ADEBin : List BIN File) :

coin		
	1	2
1	-0.2500	-0.2500
2	2.2500	1.2500

Utiliser la présente option :

HVUtil->QuadratLattice			
.lat type file		rectangle.lat	
HV Corners		coin	2 2
HV Points		aleal	30 2
Output file name		essai	

Input file (lattice definition): rectangle
 -> Grid with 6 rows and 9 columns
 -> Selected quadrats: 21
 Input file (Corner coordinates): coin
 -> Bottom left x = -2.5000e-01, y = -2.5000e-01
 -> Top right x = 2.2500e+00, y = 1.2500e+00
 Input file (Point coordinates): aleal
 -> 30 rows and 2 columns

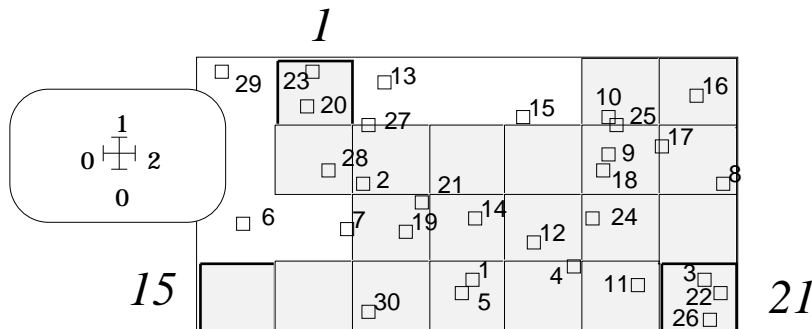
File :aleal

Col.	Mini	Maxi
1	8.679e-02	1.967e+00
2	2.808e-02	9.585e-01

Output file essai_qua
 -> 30 rows and 1 column
 -> Quadrat number in which stay each point
 -> 0 if the point is outside
 -> Outside point number: 6

Output file essai_cou
 -> 21 rows and 1 column
 -> Point number inside each quadrat

Vérifier en superposant la grille et le nuage dans le grapheur :



Il y a 17 quadrats utiles et 30 points. Les quadrats sont numérotés ligne par ligne de gauche à droite : on compte le nombre de points par quadrats et on associe à chaque point le numéro du quadrat où il se trouve. Vérifier que le point 3 est dans le quadrat 21, que le point 7 n'est pas dans un quadrat utilisé (enregistrement 0), que le quadrat 1 contient 2 points et que le 7 est vide ...

essai_qua	
	1
1	18.0000
2	5.0000
3	21.0000
4	19.0000
5	18.0000
6	0.0000
7	0.0000
8	9.0000

r19	10.0000
t20	1.0000
t21	10.0000
t22	21.0000
x23	1.0000
x24	13.0000
x25	2.0000
y26	21.0000

essai_cou	
	1
1	2.0000
2	2.0000
3	1.0000
4	1.0000
5	1.0000
6	0.0000
7	0.0000
8	2.0000

⚠ Remarque : les points tombant sur la frontière des quadrats sont attribués de la manière suivante :

