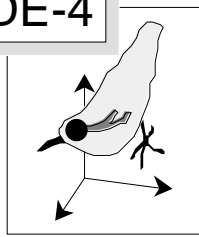


ADE-4



NGUtil

NGUtil : BinToGraph.....2
 NGUtil : Edge->Graph.....4
 NGUtil : Edit Graph.....6
 NGUtil : From_Cabri_Graph.....8
 NGUtil : From_Groups&Graphs.....11
 NGUtil : LinearGraph.....14
 NGUtil : Text->Graph.....16
 NGUtil : To_Cabri_Graph.....18
 NGUtil : To_Groups&Graphs.....21

Interfaces avec :

The image shows two overlapping software windows. The foreground window is titled "Groups & Graphs" and contains a complex graph structure with nodes and edges. Text within this window includes: "version 2.4 June, 96", "Designed and written by William Kocay and Christian Pantel, with help from Ben Li, Patric Niesink, Lara Winstone, Hamish Carr", "written in CodeWarrior Pascal, from Metrowerks.", "2.4 PPC", and a copyright notice for 1987-1996 by William Kocay at the University of Manitoba. The background window is titled "Cabri-graphe" and is described as "a Tool for Research and Teaching in Graph Theory". It lists authors: "D. Baudon, C. Benzaken, J. Bordier, Y. Carboneaux, P. Eades, I.Havel, J.M. Laborde, G. Lejeune, M. Mollard". It features a logo, "copyright LSD2/CNRS/UJF", "Version 3.2b1(PowerPC)", and "OK" and "More..." buttons.

NGUtil : BinToGraph



Utilitaire de manipulation de fichier.



Quand on possède une matrice carrée dans un fichier binaire, l'option vérifie que la matrice est carrée, symétrique, qu'elle contient des valeurs 0 ou 1 et que toutes les valeurs sur la diagonale sont nulles. Dans ce cas, elle crée une matrice de voisinage (voir [NGUtil : Edit Graph](#)) et les versions des fichiers ASCII contenant la même information dans la version de [NGUtil : Text->Graph](#) et celle de [NGUtil : Edge->Graph](#). Cette option n'est utile que pour la gestion des cartes de données.



L'option utilise une seule fenêtre de dialogue :

Nom du fichier binaire d'entrée.

Option de sauvegarde du graphe en ASCII.

Option de sauvegarde du graphe sous forme de liste d'arêtes.

Option de sauvegarde du graphe dans le format numéro du points, nombre de voisins, listes des voisins.



Utiliser la carte Sicile+1. Lire le graphe de voisinage ([NGUtil : Edge->Graph](#)) :

Passer en binaire le fichier W ([TextToBin : Char->Binary](#)) :

Utiliser l'option :

```
Input file: WBin
Row: 13 Col: 13
Symtric matrix with 0-1 values
```

La matrice est effectivement un graphe de voisinage.

```
Neighborhood graph in binary file: WBin_G
It contains graph matrix (LEBART's M) with 13 rows and columns
Neighborhood weights in binary file: WBin_G.gpl
It contains 13 rows and 1 column
```

Editer le contenu du fichier texte de sortie :

```
0110000000000
1010100000000
1101100000000
0010110000001
0111011000000
0001101110011
0000110100000
0000011010000
0000010100010
0000000000101
0000000001011
0000010010101
0001010001110
```

Input file: WBin
Row: 13 Col: 13
Output text file: WBin_G_edge

Editer ce fichier. On a la même information sous forme de liste d'arêtes :

```
1 2
1 3
2 3
2 5
3 4
3 5
4 5
4 6
4 13
5 6
5 7
6 7
6 8
6 9
6 12
6 13
7 8
8 9
9 12
10 11
10 13
11 12
11 13
12 13
```

Input file: WBin
Row: 13 Col: 13
Output text file: WBin_edit

Editer ce fichier. On a la même information sous forme de liste de sommets :

```
1 2 2 3
2 2 3 5
3 2 4 5
4 3 5 6 13
5 2 6 7
6 5 7 8 9 12 13
7 1 8
8 1 9
9 1 12
10 2 11 13
11 2 12 13
12 1 13
```

NGUtil : Edge->Graph



Utilitaire de saisie de données.



Le module assure la lecture d'un graphe de voisinage. Un graphe de voisinage sur n points est une matrice A à n lignes et n colonnes où A_{ij} vaut 1 si i est voisin de j et 0 sinon. Pour constituer une telle matrice, on peut utiliser un tableur pour saisir la liste des arêtes (couples de voisins) :

	0	1	2	3	4	5
1		2				
1		3				
2		3				
2		5				
3		4				
3		5				
4		5				
4		6				
4		13				
5		6				
5		7				
6		7				
6		8				
7		8				

Sauvegarder en texte seulement (Excel 4) ou en texte avec tabulation (Excel 5) ou, avec un éditeur de texte, séparer les nombres par des tabulations. Il suffit d'indiquer une seule des arêtes (i,j) ou (j,i).



L'option utilise une seule fenêtre de dialogue :

Edge->Graph

Text input file

Total point number

Binary output file

Quit Ok

Fichier texte d'entrée.

Nombre total de points (entier n).

Nom générique des fichiers de sortie.



Utiliser la carte Sicile+1. Lire le graphe de voisinage :

Edge->Graph

Text input file

Total point number

Binary output file

```
Neighborhood relation reading
Input text file: SicileG.edges
Neighborhood graph in text file: W_G
It contains graph matrix (LEBART's M) with 13 rows and columns
Neighborhood weights in binary file: W_G.gpl
It contains 13 rows and 1 column
```

Editer le graphe (NGUtil : Edit Graph) :

Edit Graph

Graph input file 13 1

```

Neighborhood relation
Input file: W
Neighborhood graph in text file: W
Symetric neighborhood matrix (. for 0):
-----
●11.....
1●1.1.....
11●11.....
..1●11....1
.111●11....
...11●111..11
....11●1....
.....11●1...
.....1.1●.1.
.....●1.1
.....1●11
.....1..1.1●1
.....1.1...111●
-----

```

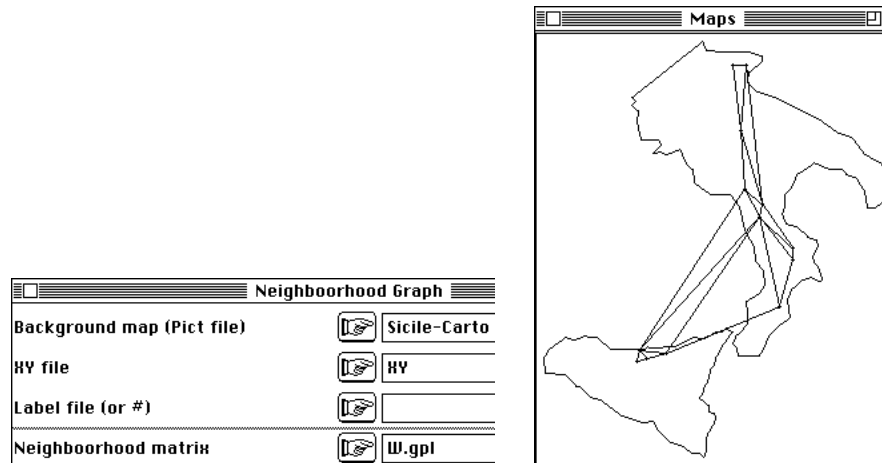
Les fichiers contenant un graphe de voisinage ont le suffixe `_G`. Il y a toujours un fichier de suffixe `_G.gpl` associé qui contient le poids de voisinage de chaque point c'est-à-dire le rapport du nombre de voisins de ce point sur le nombre total de couples de voisins.



Lebart, L. (1969) Analyse statistique de la contiguïté. *Publication de l'Institut de Statistiques de l'Université de Paris* : 28, 81-112.



On peut aussi saisir un graphe de voisinage par GNUtil : Texte->Graph. Pour vérifier la matrice de voisinage obtenue sur une carte, utiliser Maps : Neighborhood Graph :



NGUtil : Edit Graph



Utilitaire d'édition de fichier.



L'option édite les matrices de graphe de voisinage.



L'option utilise une seule fenêtre de dialogue :



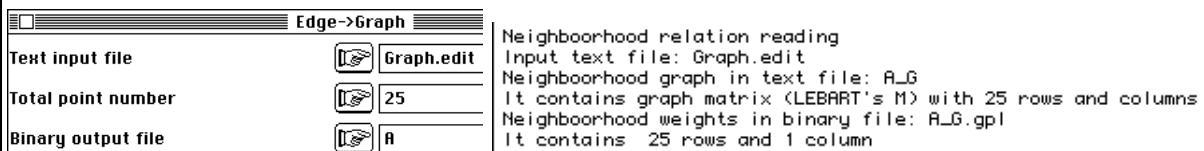
Nom du fichier binaire d'entrée de type ###.gpl (*n* lignes et une colonne) associé à un fichier TEXT contenant une matrice de valeurs 0-1 *n*-- symétrique.



Utiliser la carte Irlande :

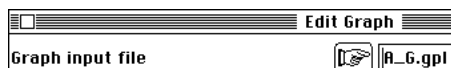
```
ADE-4•Data,Irlande,Graph.edit
1 5 8 9 10 24 25
2 5 11 13 23 16 17
3 4 12 21 6 7
4 4 7 12 21 22
5 1 11
6 5 15 19 18 21 3
7 3 4 12 3
8 5 16 18 10 1 25
9 5 1 10 21 22 24
10 5 18 21 9 1 8
11 5 5 20 19 13 2
12 4 3 21 4 7
13 4 11 19 23 2
14 2 16 17
15 3 20 19 6
16 6 14 17 2 23 18 8
17 3 2 16 14
18 7 21 10 8 16 23 19 6
19 7 6 18 23 13 11 20 15
20 3 15 19 11
21 8 6 3 12 4 22 9 10 18
22 4 4 21 9 24
23 5 13 19 18 16 2
24 4 22 9 1 25
25 3 24 1 8
```

Lire le fichier texte associé :



Lister le fichier créé :

```
Neighborhood relation
Input file: A_G
Neighborhood graph in text file: A_G
Symetric neighborhood matrix (. for 0):
.....1
..1.....1.1.....111
...1.111.....1.1..11..1.
..1.....11.....11...
1...1.....1.1..1.1..1..
...1...1...1...1.11.....
..1..1.....11.....1.1...
1.1.....11.....1.1...
1..1..1.....11.1.
1.....1.....1.....1.
...111.....1.....11.....
```



Utiliser la carte Voisinages :

```
ADE-4•Data,Voisinages,AtlasG.gra
1 2 2 3
2 4 3 4 6 5
3 1 4
4 2 6 9
5 5 6 9 10 8 7
6 1 9
```

Lire le fichier texte associé :

Edge->Graph	
Text input file	AtlasG.gra
Total point number	23
Binary output file	B

Neighborhood relation reading
 Input text file: AtlasG.gra
 Neighborhood graph in text file: B_G
 It contains graph matrix (LEBART's M) with 23 rows and columns
 Neighborhood weights in binary file: B_G.gpl
 It contains 23 rows and 1 column

Lister le fichier créé :

```
Neighborhood relation
Input file: B_G
Neighborhood graph in text file: B_G
Symetric neighborhood matrix (. for 0):
```

```
-----
•11..1..1.....11
1•11..1...1..1..1111...
11•1...1...1...1.....
..11•..1.....1.....
...•1..1.....
11..1•.....
.....•11.....
...11..1•.....
1...1..1..•1.....
.....1•1.....
..1.....1•..1.....
.....•1..1.....
..1.....1•1.....
..1.....1•..11.....
...1.....11...•1.....
.....•1...1..
..11.....1..1•.....
..1.....1...•1...1
..1.....1...1•...1
..1.....1...•1...1
.....1...•1...1
1.....1...•1...1
1.....11111•
```

Edit Graph	
Graph input file	B_G.gpl



Pour obtenir un graphe de voisinage on peut :

- créer un fichier texte dans le format lu par NGUtil : Edge->Graph ;
- créer un fichier texte dans le format lu par NGUtil : Text->Graph ;
- utiliser l'éditeur de graphe **Cabri-Graphe** et revenir par l'option NGUtil : From_Cabri_Graph ;
- utiliser l'éditeur de graphe **Groups & Graphs** et revenir par l'option NGUtil : From_Groups&Graphs ;
- entrer la matrice de voisinage directement, passer le fichier en binaire et revenir par l'option NGUtil : BinToGraph ;
- créer un graphe géométrique par NGUtil : LinearGraph ;
- créer un graphe entre unités surfacique par Areas : AreasToGraph ;
- créer un graphe entre placettes d'une grille par Lattices : LattiToGraph ;

Dans tous les cas, le graphe de voisinage est dans un fichier ASCII et on y accède par le fichier .gpl associé.

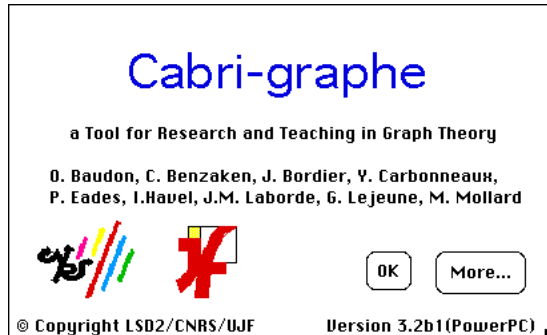
NGUtil : From_Cabri_Graph



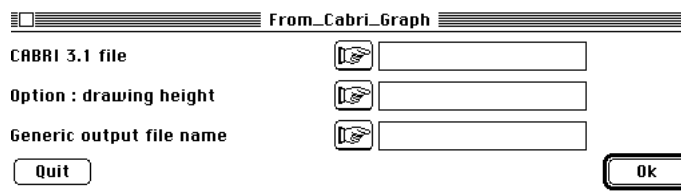
Utilitaire d'interface.





L'option lit les fichiers texte créés par Cabri-Graphe, logiciel freeware téléchargeable à :
http://www-cabri.imag.fr/CabriGraphes/cabri_graph.html



L'option utilise une seule fenêtre de dialogue :



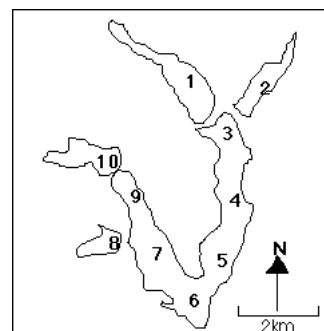
 Nom du fichier texte d'entrée créé par Cabri-Graph 3.1.

 Option d'entrée de la hauteur du graphique.

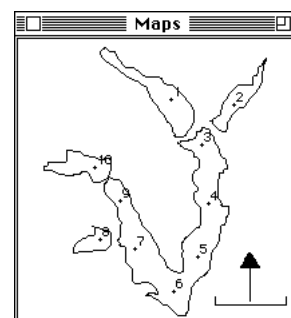
 Nom générique des fichiers de sortie (création).



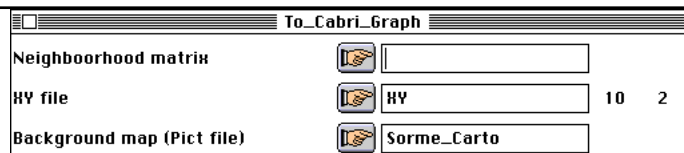
Utiliser le dossier créé par la carte Somme de la pile ADE-4•Data. Digitaliser les 10 stations représentées sur le fond de carte Somme_Digi (Digit : Digitize) :



Vérifier (Maps : Labels) :



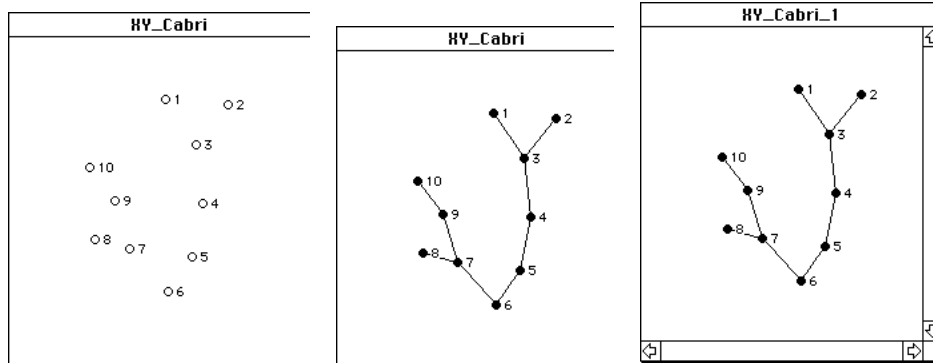
On voudrait implanter un graphe de voisinage entre stations "à la souris". On peut le faire par l'éditeur de graphe Cabri-Graphe. Envoyer l'information au logiciel par :



On obtient :

Input file (neighborhood graph):
 Input file (coordinates): XY
 Output file for Cabri Graph 3.1: XY_Cabri

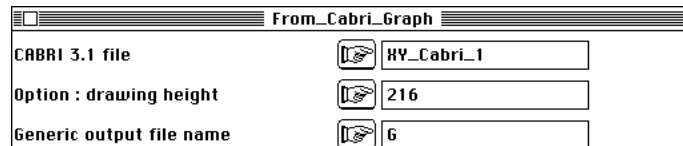
Ouvrir le fichier par Cabri-Graphe qui affiche :



Avec la touche commande enfoncée qui crée un curseur en forme de crayon; tracer des traits entre les sommets du graphe pour obtenir le graphe de voisinage désiré. Sauvegarder en texte. Pour revenir dans ADE-4, utiliser MapUtil : GetPictSize pour obtenir la hauteur de la figure des fonds de carte :



Input file (Background-PICT): Sorme_Digi
 Width = 214
 Height = 216



```

vertice: 1 degree: 1
vertice: 2 degree: 1
vertice: 3 degree: 3
vertice: 4 degree: 2
vertice: 5 degree: 2
vertice: 6 degree: 2
vertice: 7 degree: 3
vertice: 8 degree: 1
vertice: 9 degree: 2
vertice: 10 degree: 1
Input file (From cabriGraph): XY_Cabri_1
Neighborhood graph in text file: G_G
It contains graph matrix (LEBART's M) with 10 rows and columns
Neighborhood weights in binary file: G_G.gpl
It contains 10 rows and 1 column
Vertice coordinates in file: G_XY
It contains 10 rows and 2 columns
  
```

Editer (NGUtil : Edit Graph) :



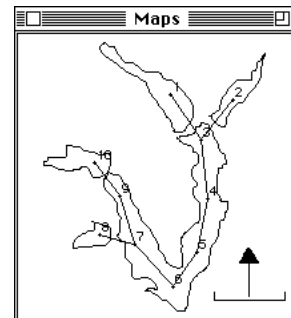
```

Neighborhood relation
Input file: G_G
Neighborhood graph in text file: G_G
Symetric neighborhood matrix (< for 0):
-----
● 1.....
● 1.....
11●1.....
...1●1....
... 1●1...
... 1●1...
... 1●11.
... 1●...
... 1.●1
... 1●
-----

```

Vérifier (Maps : Neighborhood Graph) :

Neighborhood Graph	
Background map (Pict file)	<input type="text" value="Sorme_Carto"/>
HV file	<input type="text" value="GS_HV"/> 10 2
Label file (or #)	<input type="text" value="#"/>
Neighborhood matrix	<input type="text" value="GS_\$G"/> 10 10



Cabri-Grappe accepte des graphes de voisinage sans arête et peut donc être utilisé comme éditeur de graphes de voisinage. Il permet évidemment bien d'autres fonctions (Boîte à outil, Générateur aléatoire, Transformation, Opération, Calcul d'invariants, Visualisation, Représentation, ...



Guido Y. (1984) Cabri: un cahier de brouillon informatisé pour l'étude de la théorie des graphes, thèse 3^o cycle, université de Montpellier, France.

Dao M., Habib M., Richard J.P. & Tallot D. (1986) "Cabri" an interactive system for graph manipulation, Graph Theoric Concepts in Computer Science, Proceeding of the International Workshop WG'86, Bernried FRG, june 1986, Lecture Notes in Computer Science 246, Springer Verlag, pp. 58-67.

Laborde J-M. (1987), "Cabri" a tool for research and teaching in graph theory, Colloque "La combinatoire et l'informatique", Montreal, Canada.

Baudon Olivier (1990), CABRI-Graphes, un CAHIER de BRouillon Interactif pour la Théorie des Graphes, These de 3-ième cycle, Grenoble, France.

Y. Carbonneaux , J-M. Laborde & M. Madani, "Cabri-Graph: a Tool for Research and Teaching in Graph Theory", Springer-Verlag, GRAPH DRAWING'95 (à paraître en janvier 1996).

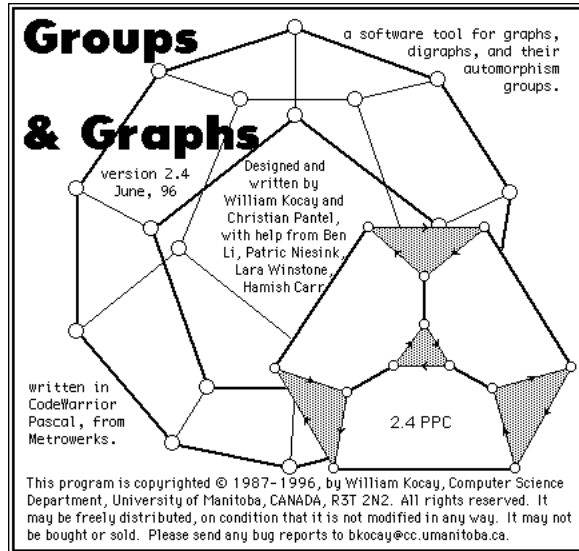
NGUtil : From_Groups&Graphs



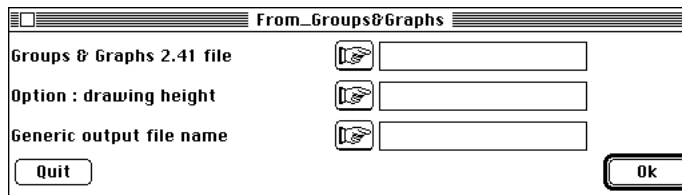
Utilitaire d'interface.




L'option lit les fichiers texte écrits par Groups & Graphs, logiciel freeware téléchargeable à <http://130.179.24.217/G&G/G&G.html>.




L'option utilise une seule fenêtre de dialogue :



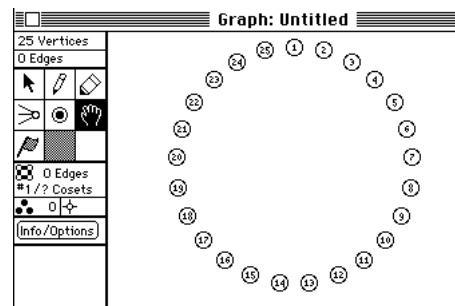
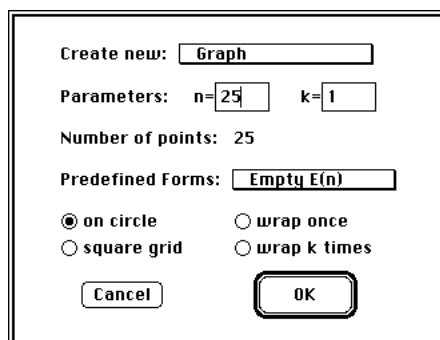
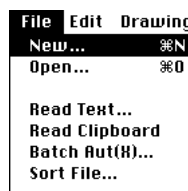
 Nom du fichier texte d'entrée créé par Groups & Graphs.

 Option d'entrée de la hauteur du dessin.

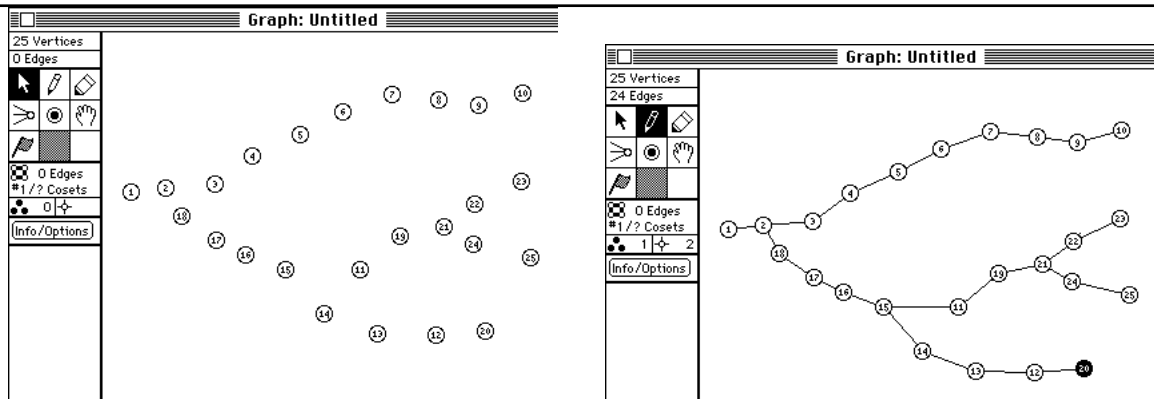
 Nom générique des fichiers de sortie.



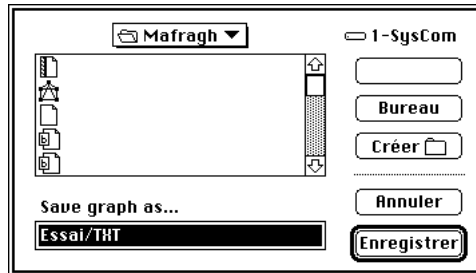
Utiliser le dossier de travail en cours et ouvrir Groups & Graphs :



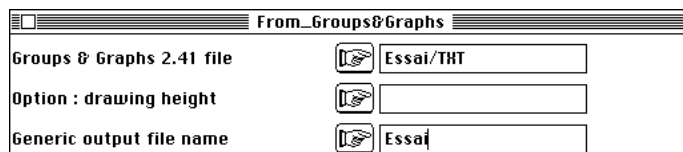
L'interface utilisateur est remarquablement efficace. Déplacer les points avec l'outil flèche et tracer les arêtes avec l'outil crayon. Déplacer la figure avec l'outil main :



Sauvegarder en texte avec l'option Write text du menu file :



Importer dans ADE-4 :



Input file: Essai/TXT
Text file with 51 rows
Neighborhood graph in binary file: Essai_G
It contains graph matrix (LEBART's M) with 25 rows and columns
Neighborhood weights in binary file: Essai_G.gpl
It contains 25 rows and 1 column
Vertice coordinates in file: Essai_XY

Editer (NGUtil : Edit Graph) :

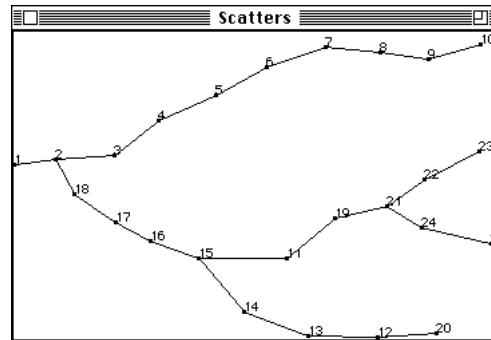
```

•1.....
1•1.....1.....
.1•1.....
..1•1.....
...1•1.....
....1•1.....
.....1•1.....
.....1•1.....
.....1•1.....
.....1.....
.....•..1..1.....
.....•1.....1.....
.....1•1.....
.....1•1.....
.....1..1•1.....
.....1•1.....
.....1•1.....
.....1•1.....
.....1.....
.....1.....•1.....
.....1.....•.....
.....1..1•1.....
.....1•1..
.....1•..
.....1..•1
.....1•

```

Vérifier la pertinence des informations transmises (Scatters : Neighborhood graphs) :

Neighborhood graphs	
XY coordinates file	Essai_HY
X-axis column number (default = 1)	
Y-axis column number (default = 2)	
Label file (or # for item numbers)	#
Draw points (no = 2)	
Neighborhood matrix	Essai_\$G



Groups & Graphs n'accepte pas des graphes de voisinage sans arête mais c'est un éditeur de graphes extrêmement efficace. Il permet évidemment bien d'autres fonctions complexes :

- Visual graph/digraph editor;
- Automorphism group computation;
- Graph certificate which identifies a graph uniquely up to isomorphism;
- Hamiltonian cycles;
- Planarity test and planar layout;
- Line graphs, neighbour graphs, bipartite doubles;
- High quality PostScript output of graphs and digraphs;
- Display of orbits, generators, and elements of permutation groups;
- Block systems, commutator subgroups, stabilisers.



Les utilitaires d'interface sont sujets à des modifications issues des versions ultérieures des deux logiciels. Prévenir les auteurs en cas de problèmes.

NGUtil : LinearGraph



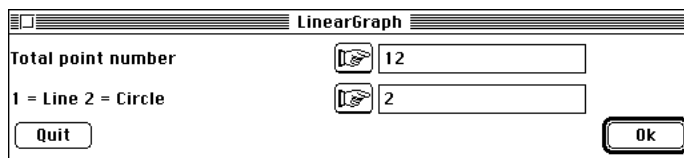
Utilitaire de création de fichier.



L'option assure la création d'un graphe de voisinage. Un graphe de voisinage sur n points est une matrice A à n lignes et n colonnes où A_{ij} vaut 1 si i est voisin de j et 0 sinon. On manipule ici la relation de voisinage linéaire (les n points sont sur une droite : 1 est voisin de 2, 2 est voisin de 1 et 3, j est voisin de $j - 1$ et $j + 1$, ..., n est voisin de $n - 1$) ou la relation de voisinage circulaire (es n points sont sur un cercle : 1 est voisin de 2 et de n , 2 est voisin de 1 et 3, j est voisin de $j - 1$ et $j + 1$, ..., n est voisin de $n - 1$ et de 1).



L'option utilise une seule fenêtre de dialogue :



Nombre de points total (entier n).

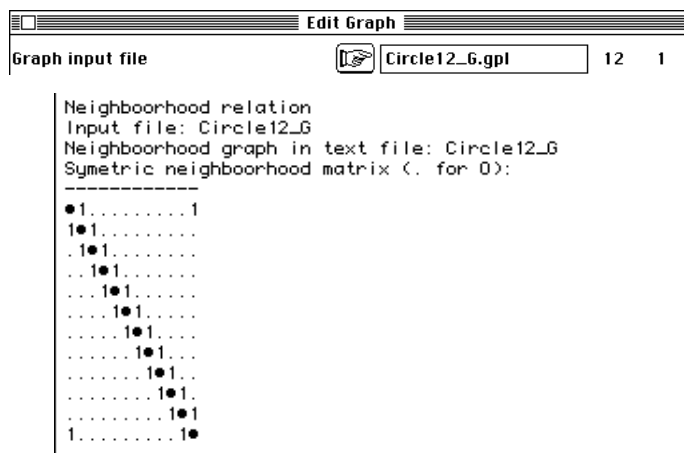
Option relation linéaire (taper 1) ou relation circulaire (taper 2).



Si on veut la relation circulaire entre les 12 mois de l'année, utiliser les paramètres ci-dessus :







```
Geometric neighborhood relation
12 points on a circle
Neighborhood graph in text file: Circle12_G
It contains graph matrix (LEBART's M) with 12 rows and columns
Neighborhood weights in binary file: Circle12_G.gpl
It contains 12 rows and 1 column
Point coordinate on a circle are in: Circle12_GXY
It contains 12 rows and 2 columns
```

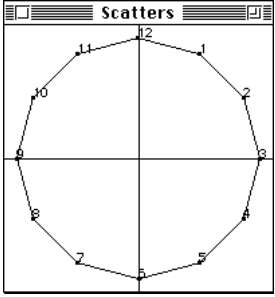
Editer (NGUtil : Edit Graph) :




Les fichiers contenant un graphe de voisinage ont le suffixe `_G`. Le nom est imposé (Line pour relation linéaire et Circle pour relation circulaire suivis du nombre de points). Il y a toujours un fichier de suffixe `_G.gpl` associé qui contient le poids de voisinage de chaque point c'est-à-dire le rapport du nombre de voisins de ce point sur le nombre total d'arêtes (couples de voisin).

Le fichier de suffixe `_GXY` donne un système de coordonnées sur le cercle unité pour des représentations graphiques variées dont la plus simple est obtenue par `Scatters : Neighborhood graphs` :

Neighborhood graphs			
XY coordinates file	 Circle12_GHY	12	2
X-axis column number (default = 1)	 <input type="text"/>		
Y-axis column number (default = 2)	 <input type="text"/>		
Label file (or # for item numbers)	 #		
Draw points (no = 2)	 <input type="text"/>		
Neighborhood matrix	 Circle12_G.gpl	12	1



 Méot, A., Chessel, D. & Sabatier, R. (1993) Opérateurs de voisinage et analyse des données spatio-temporelles. In : *Biométrie et Environnement*. Lebreton, J.D. & Asselain, B. (Eds.) Masson, Paris. 45-72.

NGUtil : Text->Graph



Utilitaire de saisie de données.



Le module assure la lecture d'un graphe de voisinage. Un graphe de voisinage sur n points est une matrice A à n lignes et n colonnes où A_{ij} vaut 1 si i est voisin de j et 0 sinon. Pour constituer une telle matrice, on peut utiliser un tableur pour saisir l'information sous la forme :

	A	B	C	D	E	F	G	t
1	1	2	2	3				
2	2	4	3	4	6	5		
3	3	1	4					
4	4	2	6	9				
5	5	5	6	9	10	8	7	
6	6	1	9					
7	7	3	8	13	12			
8	8	4	10	11	14	13		
9	9	1	10					
10	10	2	11	15				
11	11	2	14	15				
12	12	3	13	17	16			
13	13	3	14	17	18			
14	14	4	15	19	18	20		
15	15	2	20	21				
16	16	3	17	23	22			
17	17	2	18	23				
18	18	2	19	23				
19	19	2	20	23				
20	20	2	21	23				
21	22	1	23					
??								

Dans la première colonne, on indique le numéro du point enregistré (entier compris entre 1 et n). Dans la seconde on indique le nombre de ses voisins (entier compris entre 1 et n). Si m est le nombre de voisins indiqué, les m cellules suivantes sur la même ligne contiennent les numéros des voisins (entiers compris entre 1 et n). Une arête (i,j) peut être enregistrée une fois par i voisin de j ou une fois par j voisin de i ou les deux fois. Sauvegarder en texte seulement (Excel 4) ou en texte avec tabulation (Excel 5) ou, avec un éditeur de texte, séparer les nombres par des tabulations.



L'option utilise une seule fenêtre de dialogue :

Text->Graph

Text input file

Total point number

Output file name

Fichier texte d'entrée.

Nombre total de points (entier n).

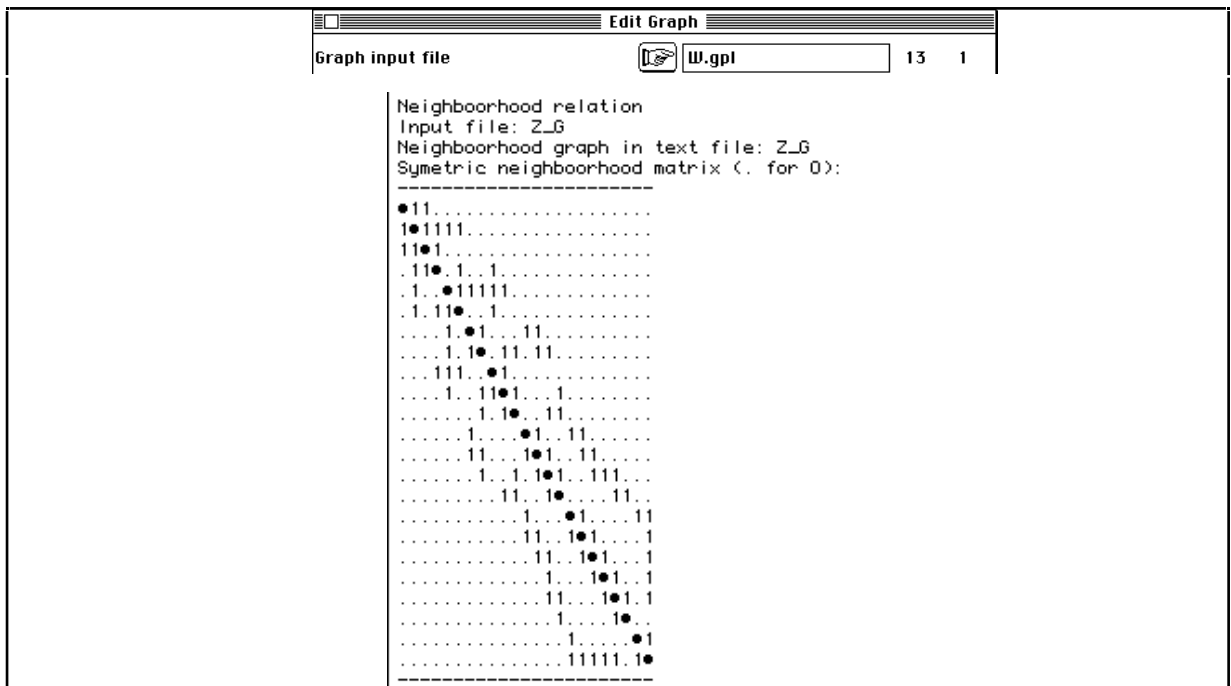
Nom générique des fichiers de sortie.



Utiliser la carte Voisinages de la pile ADE-4•Data. Obtenir avec le champ de gauche, le fichier texte AtlasG.gra. Vérifier le format en lisant le fichier avec un tableur et exécuter l'option :

```
Neighborhood relation
Input file: AtlasG.gra
Neighborhood graph in text file: Z_G
It contains graph matrix (LEBART's M) with 23 rows and columns
Neighborhood weights in binary file: Z_G.gpl
It contains 23 rows and 1 column
```

Editer le résultat :



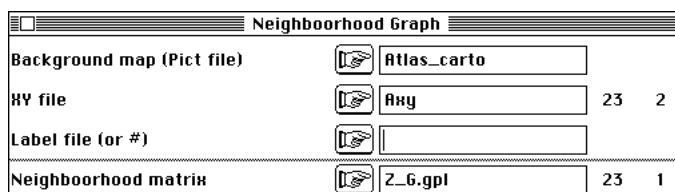
Les fichiers contenant un graphe de voisinage ont le suffixe `_G`. Il y a toujours un fichier de suffixe `_G.gpl` associé qui contient le poids de voisinage de chaque point c'est-à-dire le rapport du nombre de voisins de ce point sur le nombre total de couples de voisins.



Lebart, L. (1969) Analyse statistique de la contiguïté. *Publication de l'Institut de Statistiques de l'Université de Paris* : 28, 81-112.



On peut aussi saisir un graphe de voisinage par `NGUtil : Edge->Graph`. Pour vérifier la matrice de voisinage obtenue sur une carte, utiliser `Maps : Neighborhood Graph`.



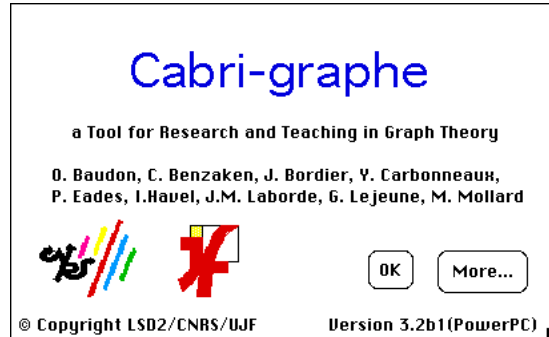
NGUtil : To_Cabri_Graph



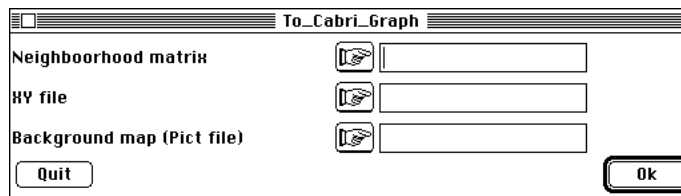
Utilitaire d'interface.




L'option écrit les fichiers texte lisibles par Cabri-Graphe, logiciel freeware téléchargeable à http://www-cabri.imag.fr/CabriGraphes/cabri_graph.html.



L'option utilise une seule fenêtre de dialogue :



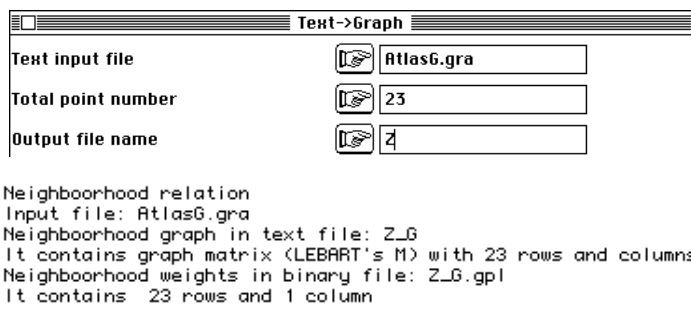
 Nom du fichier binaire d'entrée (graphe de voisinage).

 Nom du fichier binaire des coordonnées des sommets.

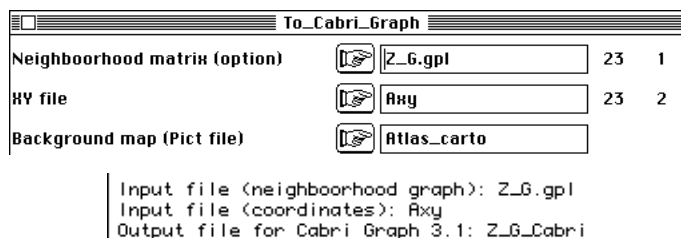
 Nom du fichier PICT de fond de carte.



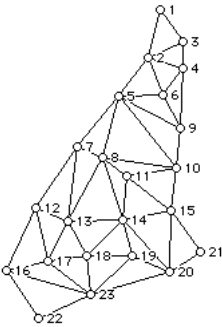
Utiliser la carte Voisinages de la pile ADE•Data. Obtenir avec le champ de gauche, le fichier texte AtlasG.gra. Lire le fichier :



Envoyer le graphe à Cabri-Graphe :



Ouvrir le fichier avec Cabri-Graphe et utiliser l'option "Place on a circumference" du menu "Transform". La structure spatiale est détruite. Sauvegarder en format texte dans un fichier Z_G_Cercle et relire ce fichier par MapUtil : From_Cabri_Graph :

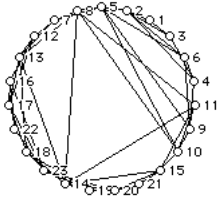


Transform Functions Interval

Do a Clique %K
 Complement
 Delete Induced Subgraph

Delete Edges %D
 Line Graph
 Product by K2
 Product by Clipboard...
 Subdivide...
 Selected Distances...

Horizontal Mirror
 Vertical Mirror
 Rotate
 Blow
Place on a Circumference
 Spring



Save as...

Format GRAF Format TERT (a bit longer)

From_Cabri_Graph

CABRI 3.1 file

Option : drawing height

Generic output file name

```

vertex: 18 degree: 3
vertex: 19 degree: 4
vertex: 20 degree: 5
vertex: 21 degree: 2
vertex: 22 degree: 2
vertex: 23 degree: 6
Input file (From cabriGraph): Z_G_Cercle
Neighborhood graph in text file: ZZ_G
It contains graph matrix (LEBART's M) with 23 rows and columns
Neighborhood weights in binary file: ZZ_G.gpl
It contains 23 rows and 1 column
Vertex coordinates in file: ZZ_XY
It contains 23 rows and 2 columns
        
```

Vérifier que le nouveau graphe dans ADE est celui qui était affiché dans Cabri-Graphe (Scatters : Neighborhood graphs) :

Neighborhood graphs

XY coordinates file

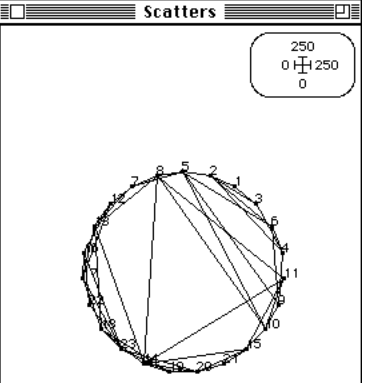
X-axis column number (default = 1)

Y-axis column number (default = 2)

Label file (or # for item numbers)

Draw points (no = 2)

Neighborhood matrix



Diagonaliser la matrice du graphe (GraphMat : Moran EigenVectors) :

Moran EigenVectors

Graph input file 23 1

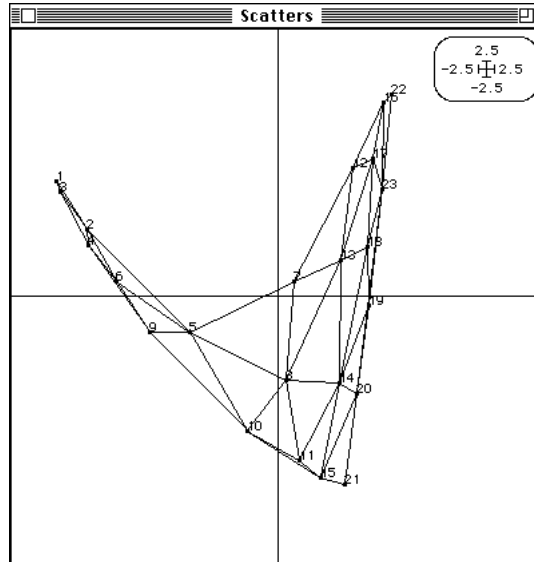
Garder au moins deux vecteurs propres et redessiner le graphe (Scatters : Neighborhood graphs) :

Moran operator diagonalization
Neighborhood graph: ZZ_G.gpl

```
-----  
Num. Eigenval. | Num. Eigenval. | Num. Eigenval. | Num. Eigenval. |  
001 9.211e-01 | 002 7.421e-01 | 003 6.402e-01 | 004 4.736e-01 |  
005 3.741e-01 | 006 2.964e-01 | 007 1.725e-01 | 008 4.518e-02 |  
009 -4.470e-08 | 010 -4.525e-02 | 011 -7.866e-02 | 012 -1.531e-01 |  
013 -1.933e-01 | 014 -2.646e-01 | 015 -2.990e-01 | 016 -3.147e-01 |  
017 -3.945e-01 | 018 -4.170e-01 | 019 -4.299e-01 | 020 -4.720e-01 |  
021 -4.916e-01 | 022 -5.279e-01 | 023 -5.838e-01 |
```

File ZZ_G.gvp contains the eigenvalues
--- It has 23 rows and 1 column
File ZZ_G.gax contains eigenvectors (norm = 1 for neighborhood weights)
--- It has 23 rows and 5 columns

Neighborhood graphs	
XY coordinates file	<input type="button" value="..."/> WW_\$Gax
X-axis column number (default = 1)	<input type="button" value="..."/> <input type="text"/>
Y-axis column number (default = 2)	<input type="button" value="..."/> <input type="text"/>
Label file (or # for item numbers)	<input type="button" value="..."/> #
Draw points (no = 2)	<input type="button" value="..."/> <input type="text"/>
Neighborhood matrix	<input type="button" value="..."/> WW_\$G



On ne retrouve pas tout à fait la structure de départ.



Les utilitaires d'interface sont sujets à des modifications issues des versions ultérieures des deux logiciels. Prévenir les auteurs en cas de problèmes.

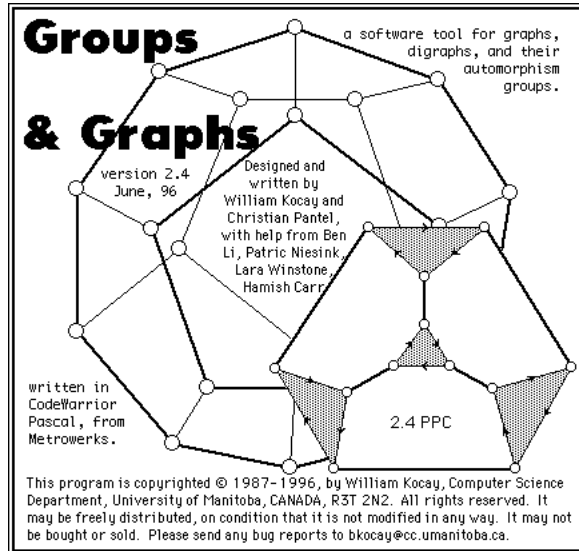
NGUtil : To_Groups&Graphs



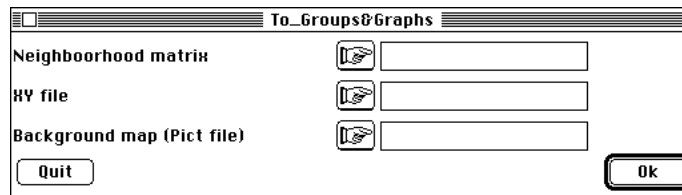
Utilitaire d'interface.





L'option écrit les fichiers texte lisibles par Groups & Graphs, logiciel de Bill Cokay, freeware téléchargeable à <http://130.179.24.217/G&G/G&G.html>.



L'option utilise une seule fenêtre de dialogue :



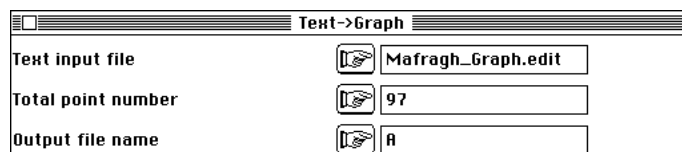
 Nom du fichier binaire d'entrée (graphe de voisinage).

 Nom du fichier binaire des coordonnées des points.

 Nom du fichier PICT de fond de carte.



Utiliser la carte Mafragh et le fichier de données créé par la carte Mafragh+5. Lire le graphe de voisinage en format texte (NGUtil : Text->Graph) :

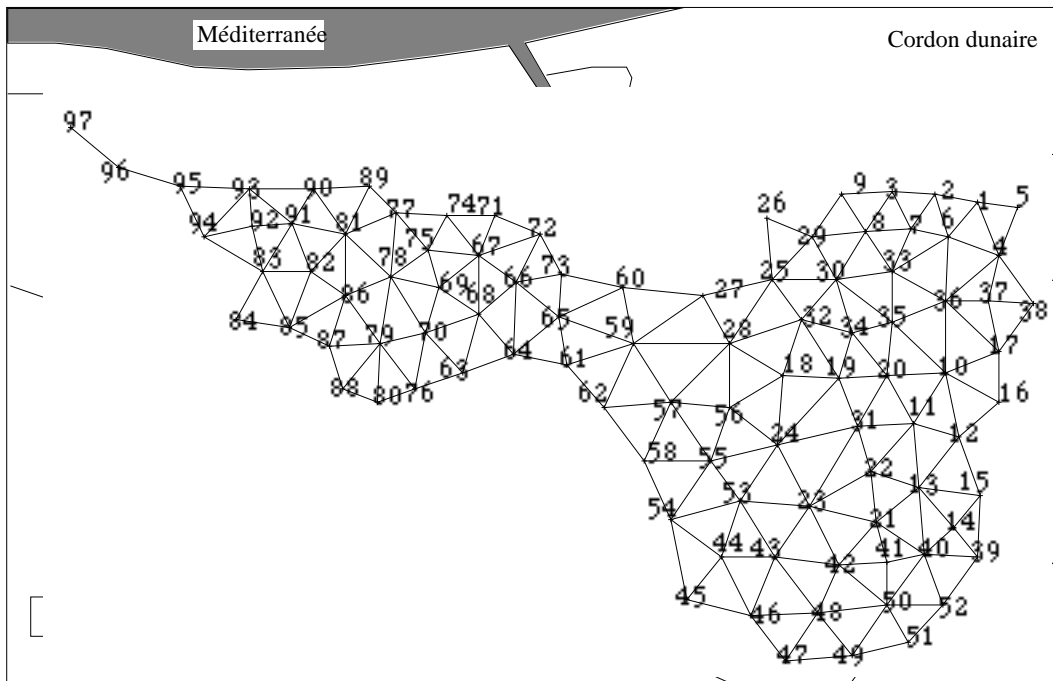


```
Neighborhood relation
Input file: Mafragh_Graph.edit
Neighborhood graph in text file: A_G
It contains graph matrix (LEBART's M) with 97 rows and columns
Neighborhood weights in binary file: A_G.gpl
It contains 97 rows and 1 column
```

```
•1.111.....
1•1..11.....
.1•...111.....
1..•11.....111.....
1..1•.....
11.1.1.....1.1.....
.11..1•1.....1.....
.1...1•.....11..1.....
.1...1•.....1.....
.....•11..11.1.....11.....
.....1•11.....1.1.....1.....
.....11•1.11.....
.....11•11.....11.....1.....
.....1•1.....11.....
```

Cartographier le graphe :

Neighborhood Graph			
Background map (Pict file)	<input type="button" value="Browse"/>	Mafragh_Digi	
XY file	<input type="button" value="Browse"/>	Mafragh_XY	97 2
Label file (or #)	<input type="button" value="Browse"/>		
Neighborhood matrix	<input type="button" value="Browse"/>	A_G.gpl	97 1



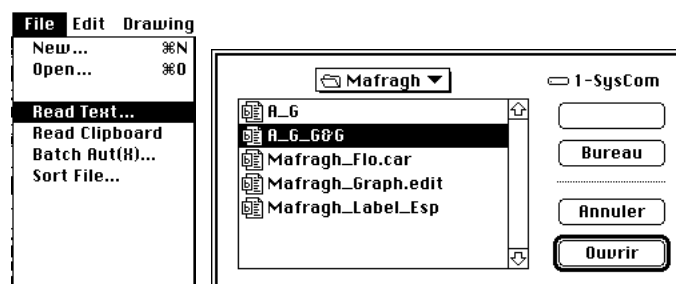
Envoyer le graphe à Groups & Graphs :

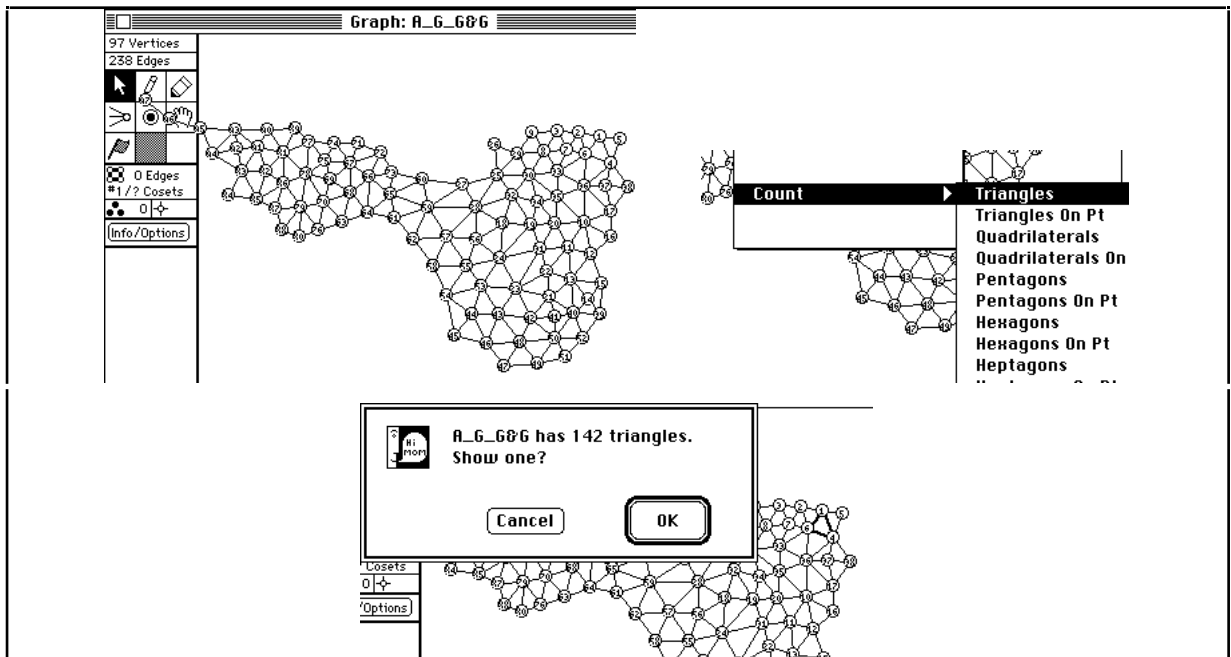
To_Groups&Graphs			
Neighborhood matrix	<input type="button" value="Browse"/>	A_G.gpl	97 1
XY file	<input type="button" value="Browse"/>	Mafragh_XY	97 2
Background map (Pict file)	<input type="button" value="Browse"/>	Mafragh_Carto	

```

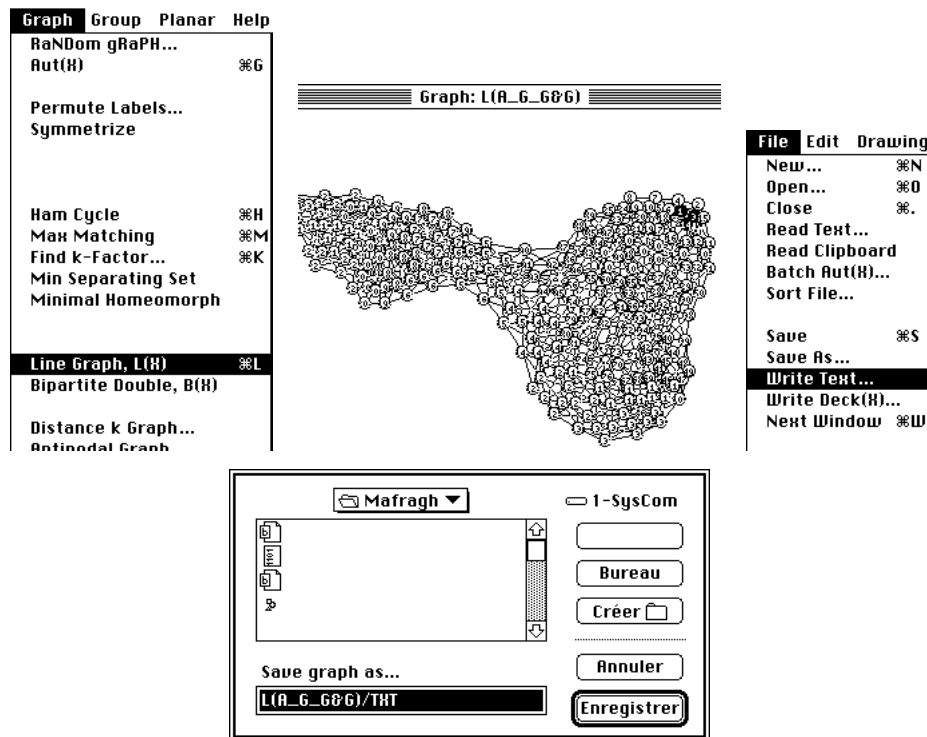
Input file (neighborhood graph): A_G.gpl
Input file (coordinates): Mafragh_XY
Output file for Groups&Graphs 2.4: A_G_G&G
  
```

Ouvrir Groups & Graphs et utiliser l'option Read Text du menu File :

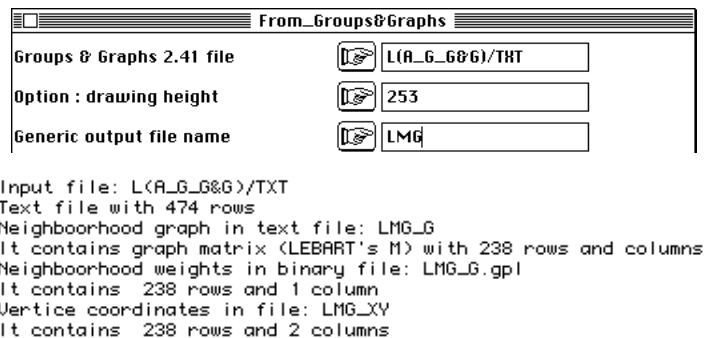








Reprendre le graphe initial (sans le déplacer). Construire le graphe L(X) et le sauvegarder en texte :

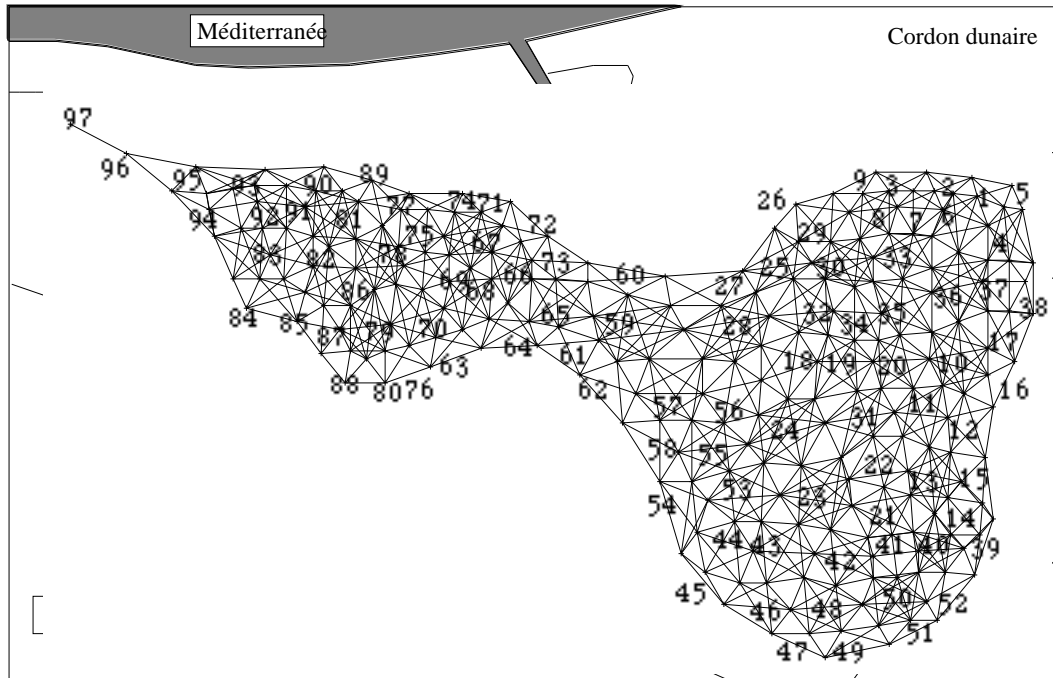


Importer le nouveau graphe dans ADE-4 :



Cartographier le nouveau graphe :

Neighborhood Graph			
Background map (Pict file)	 Mafragh_Digi		
HY file	 LM6_HY	238	2
Label file (or #)			
Neighborhood matrix	 LM6_6.gpl	238	1



Deviner la définition de $L(X)$. Vérifier qu'il y a 97 points (sommets) et 476 couples de voisins ($2 \cdot 238$ arêtes) dans l'ancien graphe et qu'il y a 238 sommets dans le nouveau graphe.