STUDIA GEOBOTANICA
4: 63-99, 1984

SPAGHET: A COENOCLINE SIMULATOR USEFUL TO CALIBRATE SOFTWARE DETECTORS

M. LAGONEGRO

Keywords: calibration, simulation, software.

Abstract. A coenocline simulator is described, which allows one to calibrate programs or chains of programs before using them on survey data. An example is given, and the listings of two versions of the simulator, in BASIC and FORTRAN respectively.

Introduction

When a research branch grows beyond a certain threshold level, more and more facts drive the researches and the scholars toward the use of computer programs, often linked into sophisticated chains. The first is the horrendous amount of data collected in decades of field surveys or laboratory experiments, which makes it impossible to extricate sensible result by sheer visual inspection and clever intuition. One has to sort, select and reorganize the data sets into data banks, in order to achieve tables reflecting the logical structure of a targeted search through all the archives. This new table will be cleared of all the information not consistent with the intersection of parameters which have driven the search process.

Eventually the need arises to tray to extract would be relations among these parameters. They are indirectly represented in the table, as species counts or states. For this, as it is typical of a mature science, one uses indirect techniques, that is computer codes which make use of numerical indexes, which in turn reflect some assumptions on the underlying structure of the involved relations in the data.

At this point there is the danger of finding artifacts rather than the real structure. To reduce the chances of this actually happening, one has to follow the path of calibration of software detectors he is going to use. So let us see what we need to perform a reasonable calibration.

A calibrating program must allow the user to simulate a good many features of the actual working condition, under which data are usually collected. For this purpose some error generating routine must be provided, which allows to simulate errors usual in data collection. However, user control must be allowed at any level he choses. Finally, the program must be able to give the user some theoretical reference condition that could be thought of as the result of a perfect measurement. In this way one will know how near to or far from the perfect condition the software detector is working.

SPAGHET is one of these calibrating programs. Another for example is by Gauch (1976), SPAGHET assumes a certain underlying model of coenocline, in which a given number of characters reacts to a complex of conditioning factors, which interact with each other in some not always known way to give size to an X axis. This axis represents the field of definition of the response functions, whose shapes, intensity and types can be varied (Austin, 1976a and b).

SPAGHET has already been used to test the performance of program COCHIS (Feoli & Lagonegro, 1983) and in teaching. It is routinely employed when a program chain has to be tested before release to the users. Another example of its usefulness is given in this volume (Feoli & Lagonegro, 1984). SPAGHET has been written in FORTRAN IV for a CDC 170/730 CYBER, operating under NOS 2.1. Two simplified versions have been realized in BASIC for APPLE II plus and OLIVETTI M20 ST respectively. The differences will be pointed at in the following. In the appendix are reported the listings of an updated FORTRAN version, called NEWSPAG and of the OLIVETTI M20ST version. APPLE users who want to implement it on their computers may have only to change the OPEN and CLOSE instructions and substitute the LPRINT instruction with the usual DOS command which sends the output to the appropriate printer slot.

1. Technical description of SPAGHET

In Figure 1 is given a simple example of the coenocline model on which SPAGHET is based. In the example all the response functions tre Gaussian, since it gives a clearer picture, but other shapes are allowed, in order to attain a more realistic simulation.

1.1. Types of response functions

Four types of response functions are possible in Spaghet, and they are:

Gaussian $g(i, X) = h(i)^* \exp(-(X - m(i)) / s(i))\hat{\ }2)$

Bimodal $b(i, X) = g(i, X) - g(i, X + 4s(i))$

where $m(i)$ is the position of the mode, while $s(i)$ is the dispersion parameter sigma

Poisson $p(i, X) = h(i)^* (1/X!)^* m(i)\hat{\ }X^* \exp(-m(i))$

Mirror-poisson $p(i, X\,max - X)$

where $X\,max$ is described in Fig. 2 and $m(i)$ is the mean value, equal to $s(i)\hat{\ }2$. This sort of response function assumes non-zero values in an interval from 0 to 16, centered at the mean position.

Only the first type has been implemented on the APPLE II plus and OLIVETTI M20ST.

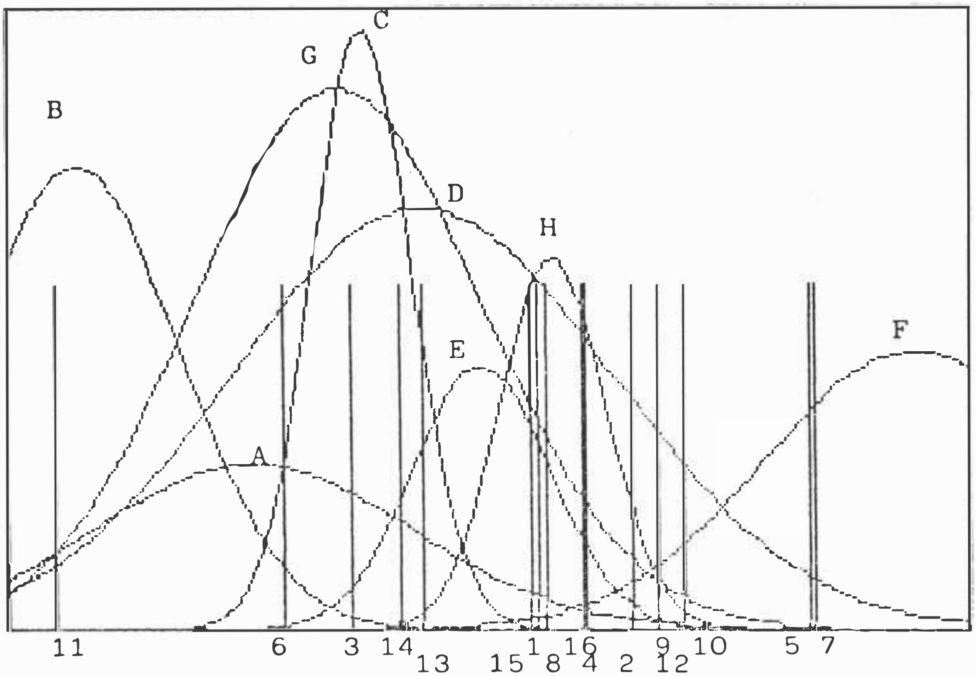

Fig. 1 — Plot of the 8 response functions and of the 16 sampling units reported in Table 1. Capital letters label the functions, while numbers label the units.

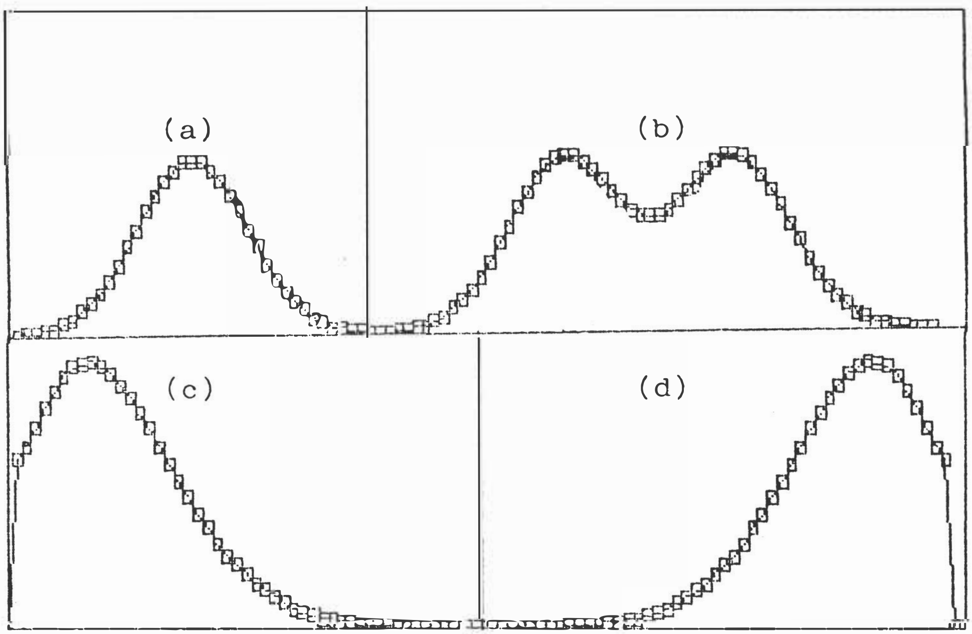

Fig. 2 — Types of response functions allowed in NEWSPAG: (a) gaussian, (b) bimodal, (c) poisson, (d) mirror-poisson. Only the first type is allowed in the versions for APPLE II and OLIVETTI M20.

1.2. Evaluation of function parameters and sampling unit positions

The type can be fixed from outside by the user, if he wants to test some ideas, or it can be randomly generated and assigned. The variable X ranges from 0 to a maximum value given in input, toghether with the number of the response functions and sampling units. The later are indicated in Fig. 1 by vertical bars. The value of the i-th response function in the j-th sampling unit is just the value assumed by the function at point $X = Xj$ (Fig. 3). Errors are introduced by allowing it to be randomly chosen inside an interval centered on the theoretical value, with amplitude allowed by a given "noise" factor, which is also an input quantity. Eventually, to simulate frequency counts, the value is given that of the nearest integer number.

The range depends on the desired average density of response function per arbitrary unit of the coenocline. In the case shown in Fig. 1, 8 functions have been generated, with 16 sampling units, in a range of value 4; so we have there an average density, both of response functions and sampling units, equal to 2 and 4 respectively. On this range value depend also the minimum and maximum values of the parameter *s (i)*, measuring the dispersion of the i-th function. The values are

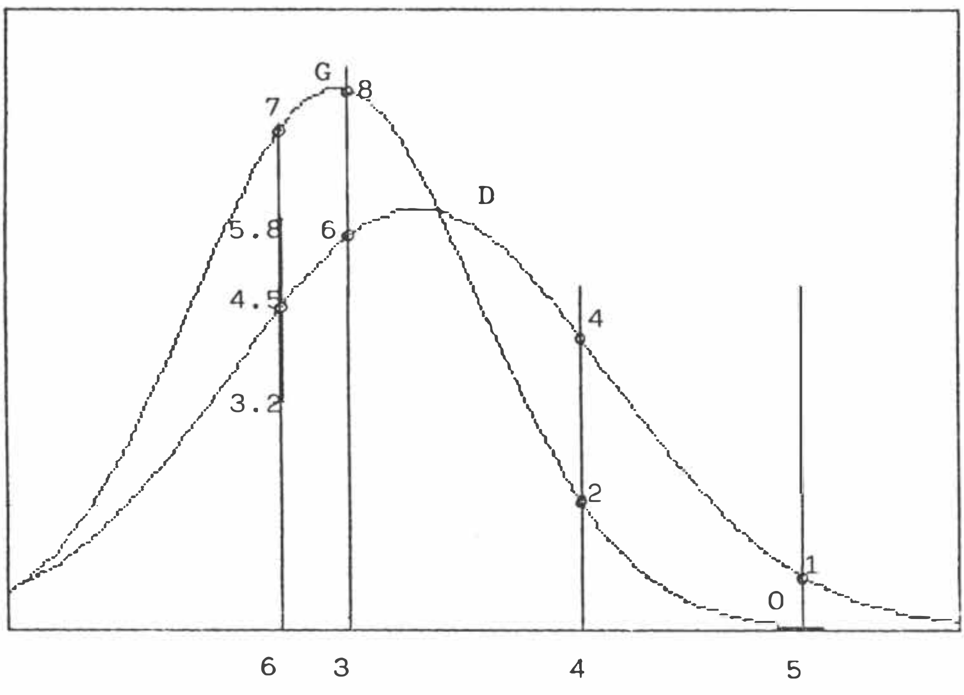

Fig. 3 — Two functions (D and G) and four sampling units (3, 4, 5, 6) from Fig. 1 are reported. In case of 0% noise, the values of the response functions in the four units would be those written near the circles. On unit 6 is reported an interval (from about 3 to about 6) into which would have fallen the simulated datum in case of 60% noise (+/− 30% around theoretical value, and then rounded to the nearest integer).

computed according to:

$$s\ (\min) = \text{Range value}/20$$
$$s\ (\max) = \text{Range value}/5$$

The functions and the sampling units are generated at random and the letter labels indicate the order of generation of the first, while the numbers are employed for the others.

The height $h\ (i)$ of the functions at the mode is given in input or randomly computed inside an interval depending on the maximum value which the user chooses. The interval limits are:

$$h\ (\min) = 1$$
$$h\ (\max) = \text{Max value}$$

The $m\ (i)$ are assigned by the user or randomly generated in the interval from 0 to the Range value.

1.3. Generation of tables

When all the functions and the sampling units are defined, a table is produced with the response functions as columns and the units as rows. The table, including noise and rounding effects, is then written to a proper file. In this form it is ready for input in a chain of software detectors.

In the case of Fig. 1, the maximum response value is 9 and the noise level is 30% (0.3). The simulated coenocline has the characteristics of Table 1.

The resulting data table (including a noise fluctuation of 30%, that is + − 15% around the theoretical value) is shown on Table 2.

Generating the same simulated experiment, but with 0% noise, gives a table with 23 non zero numbers differing by one or two units from the corresponding elements of the 30% table; this incorporates a good 18% of the whole table and a 37% of the non zero elements.

The simulation of the condition of ''perfect measurement'' is performed via a large number of equally spaced sampling units; it is clear that the larger the number the better the simulation. Anyhow, after a certain number of units the results become stable and one choses the least expensive in terms of time and computer resources. Such a perfect sample, with 100 units, is reported in Fig. 4. One could think that also a very dense random sampling could simulate this condition; may be, but one has to generate a very large number of data points, without being sure of having obtained the asymptotic condition of the theoretical measurement. To look at the effect of this technique a random sample of 100 units has also been generated. In the following paragraph the three tables will be processed via a well known procedure and the results will be compared to one another and with the content of Fig. 1.

Table 1 — Characteristics of the 8 response functions and positions of the 16 random generated sampling units.

h(min)=1 h(max)=9 s(min)=4/20=0.2 s(max)=4/5=0.8

Function number	h(i)	m(i)	s(i)
i=1(A)	2.39	1.02	0.67
2(B)	6.71	0.29	0.43
3(C)	8.69	1.49	0.21
4(D)	6.09	1.73	0.79
5(E)	3.80	1.98	0.28
6(F)	4.01	3.78	0.63
7(G)	7.84	1.39	0.60
8(H)	5.36	2.28	0.22

Sampling unit number j=	Xj=
1	2.22
2	2.60
3	1.44
4	2.41
5	3.34
6	1.15
7	3.36
8	2.25
9	2.70
10	2.82
11	0.20
12	2.71
13	1.74
14	1.64
15	2.19
16	2.40

2. Calibration of a software tool

Suppose now that we want to calibrate the well known procedure of eigenanalysis on a correlation coefficient matrix derived from the data table. What we want to see is if the procedure can work with a sampling error rate of up to 30%. We first compute the simmetric correlation matrix, which is given in Table 3.

Now we submit this matrix to an eigenanalysis in order to get the eigenvalues and the associated eigenvectors. After having written the last ones to a file, we call into action a plot routine. The results of the plot is displayed in Fig. 4. We note the appearance of the typical horseshoe trend in the scatter of the 8 response functions.

Table 2 — Data table resulting from computing the response function values at the 16 sampling units and allowing 30% noise, that is a 15% range of fluctuation both over and under the theoretical value; the values cannot anyhow be greater than the maximum allowed for the response functions.

		Response functions							
		A	B	C	D	E	F	G	H
Sampling units	1	.	.	.	4	2	.	3	5
	2	.	.	.	3	.	1	1	0
	3	2	.	6	5	0	.	6	.
	4	.	.	.	3	1	.	2	4
	5	.	.	.	1	.	3	.	.
	6	2	1	2	4	.	.	5	.
	7	.	.	.	1	.	2	.	.
	8	.	.	.	4	2	.	2	5
	9	.	.	.	2	.	1	1	1
	10	.	.	.	2	.	1	.	.
	11	1	6	.	1	.	.	1	.
	12	.	.	.	2	.	1	1	1
	13	1	.	3	5	2	.	6	.
	14	1	.	5	4	2	.	6	.
	15	.	.	.	4	3	.	3	5
	16	.	.	.	3	1	.	2	4

One can now compare sequence and groupings with the situation depicted in Fig. 1. We submit then the first three components (eigenvectors) to a three-dimensional Minimum Spanning Tree (MST) procedure, to obtain a more complete information about mutual relationships between pairs of response functions. The result is shown by te arrows in Fig. 4.

We impose this sequence as column order on the original table and rearrange it internally, by shifting the elements of each sampling unit after this order of functions. Then we reorder the unit positions (that is the row sequence) in the table by putting first that which has non zero valued elements corresponding to as many as possible of the response functions coming first in the column sequence, and so on. The results are shown in Table 4a. One can easily spot the characteristic trend of a coenocline. By comparing the sequence of sampling units with those in Fig. 1, one can see that the analysis has been quite successful, since the groups and their sequence have been properly identified.

A minimum has been found between functions G and C, thus suggesting the existence of two broad groups of functions, (H, D, G) and (C, A, B, E, F). The same grouping is suggested by average linkage clustering performed on Table 3, the

Table 3 — Matrix of correlation coefficients between pairs of response functions, computed based on the 16 random sampling units.

		Columns							
		1	2	3	4	5	6	7	8
Rows	1	1.	.733	.583	-.929	.503	.006	-.339	-.572
	2		1.	.143	-.716	.293	.006	-.447	-.378
	3			1.	-.491	.172	-.204	.260	-.719
	4				1.	-.639	-.104	.415	.401
	5					1.	-.191	-.472	.192
	6						1.	-.577	-.265
	7							1.	-.110
	8								1.

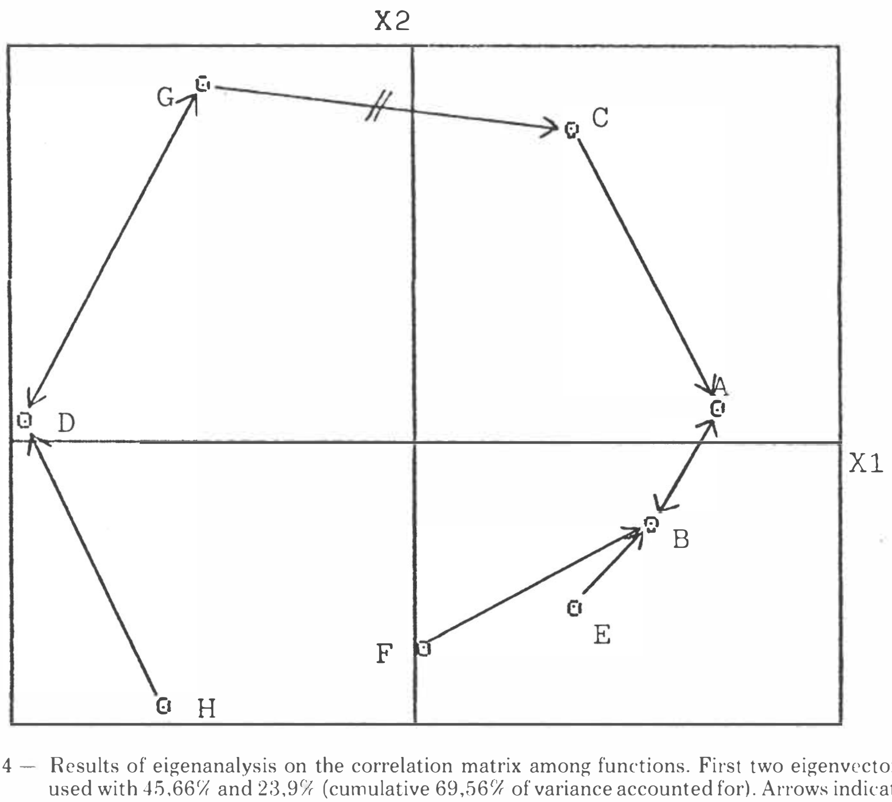

Fig. 4 — Results of eigenanalysis on the correlation matrix among functions. First two eigenvectors used with 45,66% and 23,9% (cumulative 69,56% of variance accounted for). Arrows indicate mutual M.S.T. relations suggested by a three-dimensional plot routine which uses three eigenvectors (cumulative 87,5%) and a distance matrix to give M.S.T. pairings (for its listing see in Appendix C). A minimum has been found between C and G, thus suggesting the existence of two broad groups (H D G) and (C A B E F). F has been joined to B as the last unassigned function left at the end of the clustering process.

Table 4 — Data table (Table 2) rearranged first by ordering columns according to MST based on response functions (a), then by performing the same but with order from MST based on sampling units (b).

		Response functions							
Sequence from MST		B	A	E	C	G	D	H	F
Sampling unit number	6	1	2	.	2	5	4	.	.
	11	6	1	.	.	1	1	.	.
	13	.	1	2	3	6	5	.	.
	14	.	1	2	5	6	4	.	.
	3	.	2	.	6	6	5	.	.
	1	.	.	2	.	3	4	5	.
	4	.	.	1	.	2	3	4	.
	8	.	.	2	.	2	4	5	.
	15	.	.	3	.	3	4	5	.
	16	.	.	1	.	2	3	4	.
	9	.	.	.	.	1	2	1	1
	12	.	.	.	.	1	2	1	1
	2	.	.	.	.	1	3	1	.
	5	.	.	.	.	.	1	.	3
	7	.	.	.	.	.	1	.	2
	10	.	.	.	.	.	2	.	1

(a)

		Sampling unit numbers															
Sequence from MST		5	7	10	9	12	2	16	15	8	4	1	13	14	6	3	11
Response functions	D	1	1	2	2	2	3	4	4	4	3	4	6	5	3	5	1
	F	3	3	1	1	1	1	.	.	.	.	.	.	.	.	.	.
	G	.	.	.	1	1	1	1	2	2	2	3	6	7	6	6	1
	H	.	.	.	1	1	2	3	4	5	4	5	.	.	.	.	.
	E	.	.	.	.	.	.	1	2	2	1	2	2	1	.	1	.
	A	.	.	.	.	.	.	.	.	.	.	.	1	1	2	2	1
	C	.	.	.	.	.	.	.	.	.	.	.	4	5	2	8	.
	B	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	5

(b)

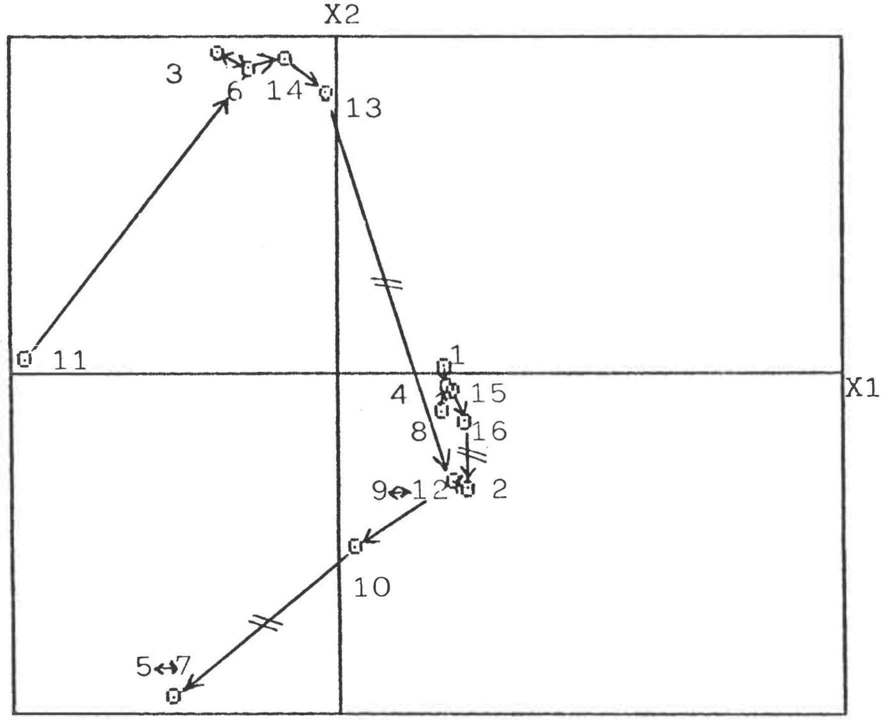

Fig. 5 — Results of eigenanalysis of correlation matrix for sampling units. First two eigenvectors account for 48,79% and 22,80% (cumulative 71,59% of the total variance). Arrows indicate suggestions of a three-dimensional M.S.T. procedure (cumulative 89,56%). Three minima are found, thus isolating four groups of units, (5 7) (10 9 12 2) (16 15 8 4 1) (13 4 6 3 11). This fits well with the structure shown in Fig. 1, all the lateral branches are very short and the sequence looks good indeed.

dendrogram of which is given in Fig. 6. If one examines Fig. 1, one sees the two function groups appear well isolated. The only exception is function *F*, which does not belong to the group it has been assigned to. The assignment is due to the fact that *F* was the last unassigned item and it was connected with the nearest of the other ones, no matter how feeble the bound was. Identical results were obtained by sum of squares agglomeration using the euclidean distance matrix based on Table 2.

By submitting to the same analysis the sampling units, the results shown in Fig. 5 and Fig. 6 are obtained. Table 4b contains the transpose of Table 2, rearranged this time after the order of the sampling units. We can see that four roups of releves are put into evidence by the MST procedure, that is (5, 7) (10, 9, 12, 2) (16, 15, 8, 4, 1) (13, 14, 6, 3, 11). This fits very well with the evidence of Fig. 1; all the lateral branches are very short and the sequence looks pretty good. The clusters from an average linkage procedure are shown in Fig. 7, and give the same suggestion, with only unit 11 out of the groups. One can see the typical stairwise linking of groups,

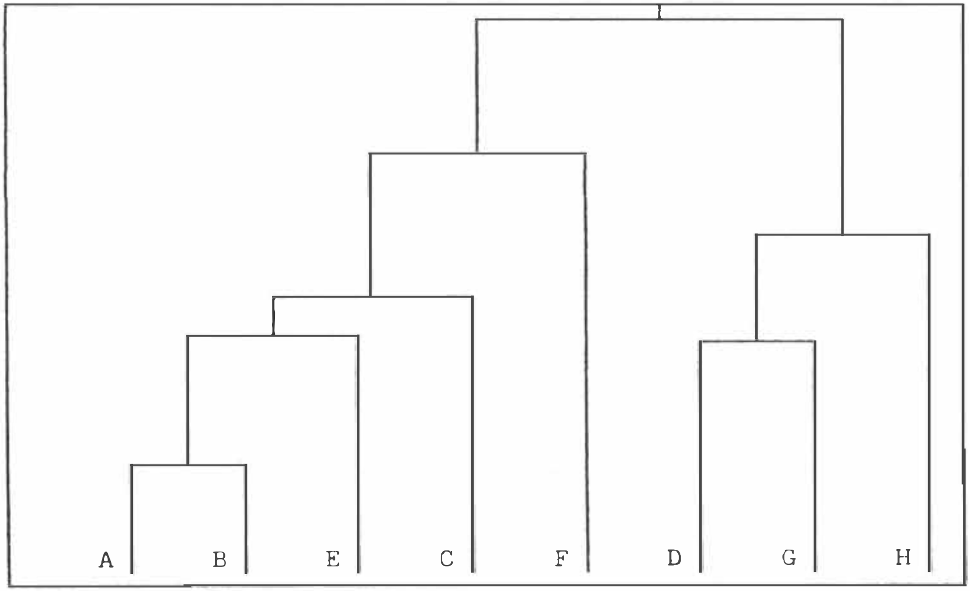

Fig. 6 — Cluster formed by average linkage clustering based on the correlation matrix among functions. Two groups are suggested (A B E C F) and (D G H).

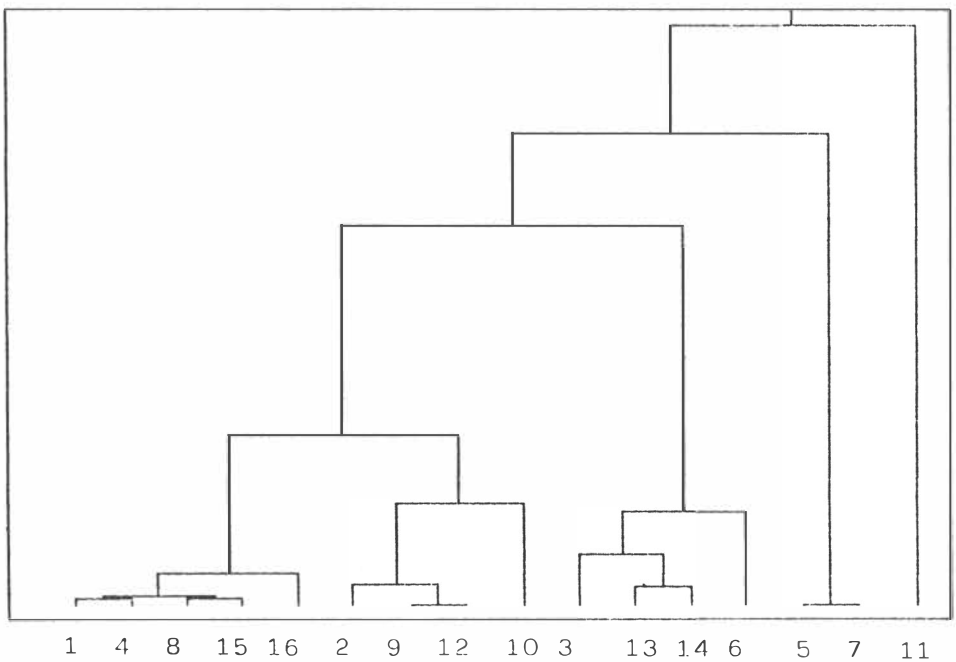

Fig. 7 — Cluster formed by average linkage clustering based on correlation matrix among sampling units. Five groups are suggested (1 4 8 15 16) (2 9 12 10) (3 13 14 6) (5 7) (11).

from left to right, which also hints the presence of a coenocline inside the data. A clustering procedure performed via sum of squares agglomeration adds elements 5.7.11 to another group, thus showing only three broad groups. The cross matrices are reported in Table 5, together with the level of significance of the comparisons. We see that the three classifications are statistically identical, so that we can use which of them we want to. By the way this comparison was not necessary in the case of the response functions, since the three suggestions were identical.

So much for the sampling units, since they have been classified and ordered quite satisfactorily and one can say that the praxis works very well on coenocline detection via their analysis.

Now it comes to do the same processing on the data matrix produced by the "perfect sampling", in order to verify the correspondence that the best experimental condition we could ever dream of (Fig. 8a) has with the preceeding case. In Fig. 8b we have the scattergram and the suggestions of MST: three groups can be identified, that is (*D, G*) (*C, A, B*) (*E, H, F*), with the last two closer each other than with respect to the first. The average linkage clustering shows (Fig. 9) the structure (A, B, C, E, H) (F) (D, G), with the first group exibiting the two substructures (A, B, C) and (E, H). The sum of squares procedure shows two broad groups, (A, C, E, H, F, B) and (D, G).

Table 5 — Matrices of comparison between pairs of classifications of the 16 sampling units randomly generated.

Classes from euclidean distance (rows) × Classes from MST (columns)

0	0	5	0
2	4	0	1
0	0	0	4

2I=29.3 for 6 degrees of freedom (sign.=0.000055%)

Classes from euclidean distance (rows) × Classes from average linkage (columns)

0	0	0	1	4
1	2	0	4	0
0	0	4	0	0

2I=29.3 for 8 degrees of freedom (sign.=0.00029%)

Classes from MST (rows) × Classes from average linkage (columns)

0	2	0	0	0
0	0	0	4	0
0	0	0	1	4
1	0	4	0	0

2I=37.7 for 12 degrees of freedom (sign.=0.00017%)

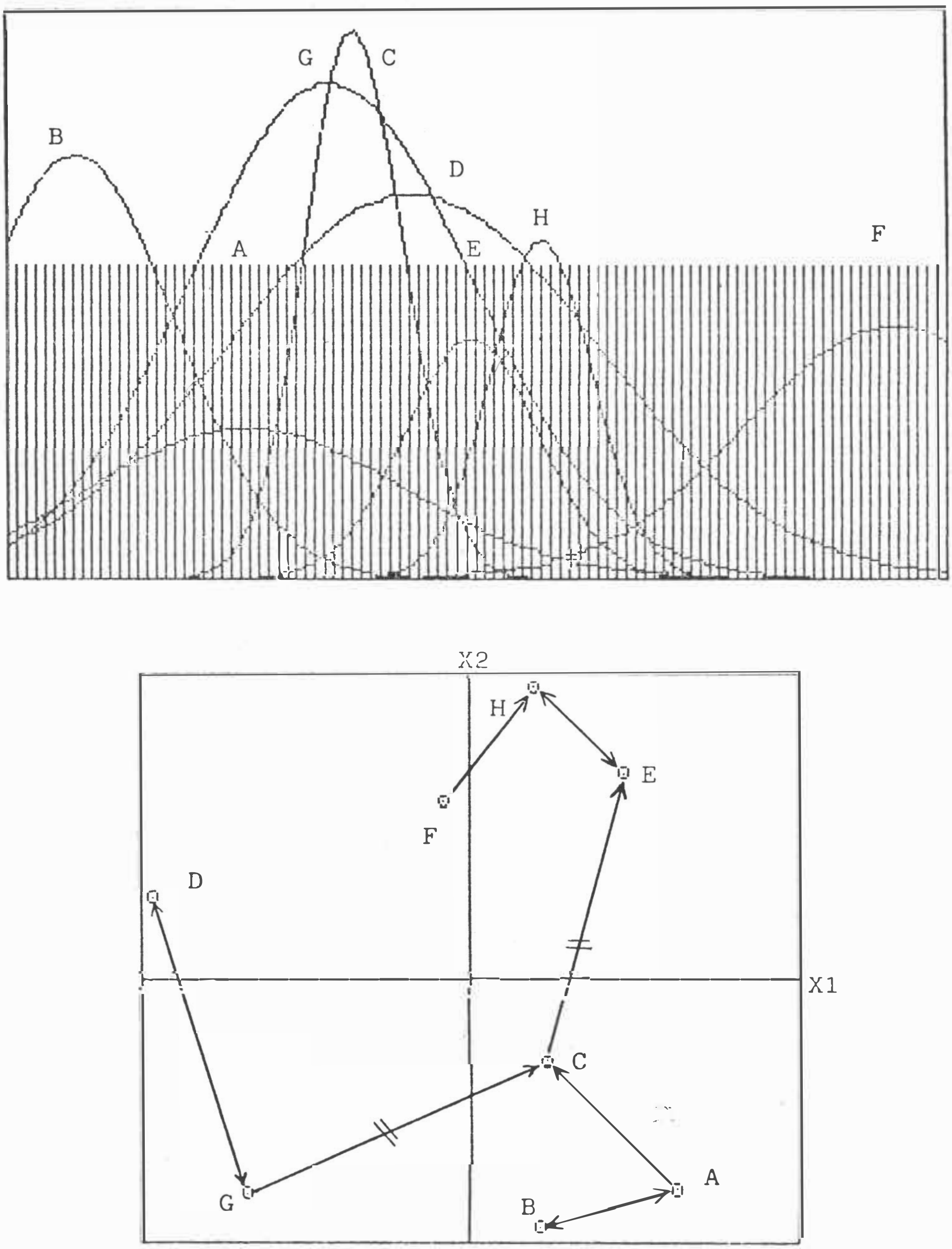

Fig. 8 — Perfect measurement sampling (a) and scattergram with M.S.T. suggestions. Two minima are found, between G and C and between E and C, thus suggesting three groups (F H E) (C A B) (G D).

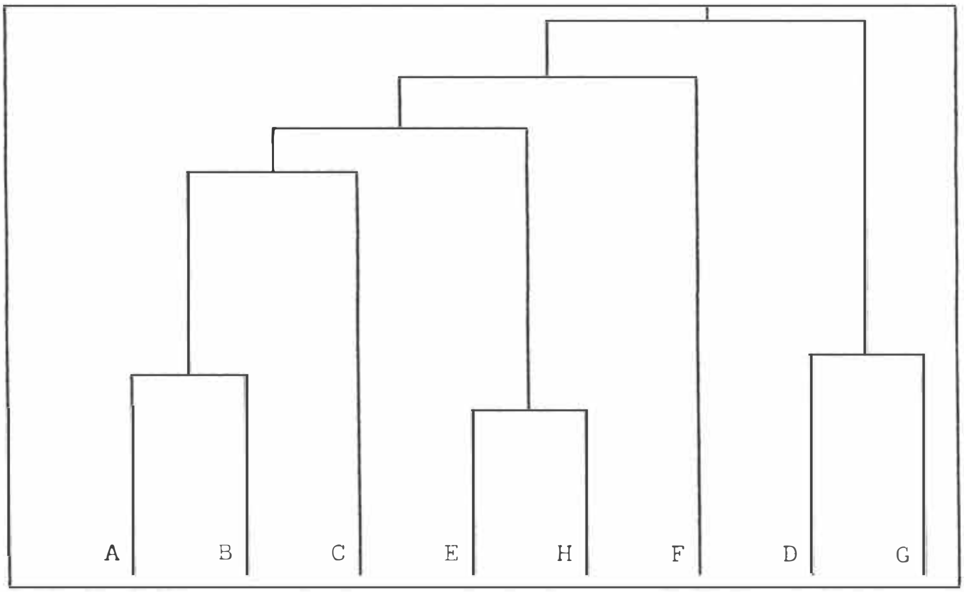

Fig. 9 — Cluster based on average linkage clustering based on correlation matrix among functions, computed using the perfect sampling condition. Three groups appear (A B C E H) (F) (D G), with the first one showing two substructures (A B C) and (E H).

Table 6 — Comparisons of classifications obtained from data representing the perfect measurement.

	Classes from euclidean distance	
Classes from average linkage	5	0
	1	0
	0	1

2I=8.99 for 2 degrees of freedom (sign.=1.1%)

	Classes from MST		
Classes from average linkage	2	3	0
	1	0	0
	0	0	2

2I=10.58 for 4 degrees of freedom (sign.=3.2%)

	Classes from MST		
Classes from euclid. dist.	3	3	0
	0	0	2

2I=8.99 for 2 degrees of freedom (sign.=1.1%)

Table 7 — Data Table 2 restructured following the order of MST from (a) 100 equidistributed sampling units, (b) 100 randomly chosen sampling units. In both the coenocline trend is clearly seen. The ordering groups response functions and sampling units into homogeneous blocks.

(a)

Sampling unit N.	F	H	E	C	A	B	G	D
	Response functions							
9	1	1	.	.	.	.	1	2
12	1	1	.	.	.	.	1	2
5	3	.	.	.	.	.	.	1
7	2	.	.	.	.	.	.	1
10	1	.	.	.	.	.	.	2
1	.	5	2	.	.	.	3	4
4	.	4	1	.	.	.	2	3
8	.	5	2	.	.	.	2	4
15	.	5	3	.	.	.	3	4
16	.	4	1	.	.	.	2	3
2	.	1	.	.	.	.	1	3
13	.	.	2	3	1	.	6	5
14	.	.	2	5	1	.	6	4
6	.	.	.	2	2	1	5	4
3	.	.	.	6	2	.	6	5
11	.	.	.	.	1	6	1	1

(b)

Sampling unit N.	B	A	C	G	D	E	H	F
	Response functions							
6	1	2	2	5	4	.	.	.
11	6	1	.	1	1	.	.	.
13	.	1	3	6	5	2	.	.
14	.	1	5	6	4	2	.	.
3	.	2	6	6	5	.	.	.
1	.	.	.	3	4	2	5	.
4	.	.	.	2	3	1	4	.
8	.	.	.	2	4	2	5	.
15	.	.	.	3	4	3	5	.
16	.	.	.	2	3	1	4	.
9	.	.	.	1	2	.	1	1
12	.	.	.	1	2	.	1	1
2	.	.	.	1	3	.	1	.
5	.	.	.	.	1	.	.	3
7	.	.	.	.	1	.	.	2
10	.	.	.	.	2	.	.	1

In table 6 are reported the cross matrices of the comparisons and the related significativity level, which tell us they are statistically equivalent. In Table 7a we have the data Table 2 rearranged in the way already described before: the coenocline trend is evident, with the less important functions D and G put aside. The sampling units are well grouped and so happens in Table 7b, reulting from the already known procedure applied to a table of 100 randomly chosen sampling units. The clustering procedures give results which are identical to those of the theoretical case.

Now it comes to comparing the results obtained for the response functions from the analysis of the 16 units table and those of the perfect case and of the 100 randomly chosen units. While the latter two give the same results and are therefore completely equivalent, the first and the theoretical case have a similarity level of 6,1%, very high but not significant, good but not perfect.

3. Conclusions

Having done all this, what can we conclude? Same things may in my opinion be deduced:

a) the software procedure based on correlation coefficient is a tool capable of detecting a coenocline, also in presence of 30% of "noise", that is experimental random errors;
b) the parallel procedure employing the euclidean distance is a good one too;
c) since the 16 random sampling units are enough the characterize their groups, but not enough to completely sample the response functions, while the 100 randomly chosen do, it would be wise to retray all again with more units, for istance 20 or 30, anyhow a number placed between 16 and 100. Another trial could be made with a smaller noise, which in turn means a more careful sampling and therefore a better survey planning.
d) since on practical ground one hasn't any Fig. 1 which he can look at, but one's professional skill and experience, it is highly advisable to use parallel procedures, both previously calibrated with the same calibrating "source", so that the probability of getting out lovable artifacts both ways can be considered very low, when not negligible;
e) the calibration must be performed with different noise levels, in order to correctly evaluate the threshold level of a given procedure, that is the level over which all the method breaks down and starts to give unstable results or no results at all.
f) the ratio between the number of sampling units, which allow to reconstruct the structure of the simulated data, and the number of response functions should be considered a sort of lower limit indicator when the actual number of survey units must be planned. This means that, if 24 units work well when 8 functions are present, the number of planned survey sampling units should be at least three times the numer of characters or species taken into account.

4. Working structure of SPAGHET

While the APPLE II and OLIVETTI M20ST versions are completely manageable through the user menu they give, the FORTRAN version NEWSPAG needs some explaining. This will be given in this section in a flow-chard like fashion.

Before starting, an advice is in order. The program has been implemented on a CDC machine and therefore all the alphanumeric strings are thought of in blocks of 10 per CDC word. The computation label, for instance, may have up to 40 characters and the associated variable is dimensioned in 4 words. If a user has to use a machine with 6 bytes per word, for example, he has to dimension the variable in 7 words so that all the 40 characters can find room, or else maintain the old dimension and reduce the label length to 24 characters.

4.1. Data flow and steps

The parameters are presented to the program in the following sequence:

1) TITLE (up to 40 characters, format 4A10), that is the computation label.
   IF the first 10 charachers are all blank THEN the program stops ELSE is goes to read the second string of parameters.
2) these parameters are:

NC, NR, NBOH, MAXALT, IRANGE, IPLOT, IFIS, IFIR, NOISE, LSLP, JACCA (free format)

NC: number of response functions (up to 100);

NR: number of sampling units (up to 300);

NBOH: arbitrary number of unused random number generations before starting the actual computation. This allows a user to change pattern to the sequence of random numbers produced by function RANF (library). We shall use the word RANF in the following to indicate a random number generation. The numbers are generated with uniform distribution density in the 0-1 interval;

MAXALT: maximum value allowed for the modal value of the response functions;

IRANGE: range ov X (in undimensioned arbitrary units);

IPLOT: IF > = 0 THEN the response functions are printed on paper in noiseless values (JACCA values)
ELSE IF < 0 THEN the Minimum Spanning Tree groups are recognized through inspection of a similarity ratio matrix, computed based on JACCA equidistributed noiseless valued sampling units;

IFIS: IF > 0 THEN the parameters of the response functions are given in input by the user
ELSE IF < = 0 THEN they are randomly generated in the chosen field of definition;

IFIR: the same as before but for the sampling units;

NOISE: in %, it is the maximum percentage indeterminacy allowed in the simulated data;

LSLP: IF = −1 THEN the similarity ratio is computed with cover scores
ELSE IF = 0 THEN normalized-cover (normacover) scores are used
ELSE IF = 1 THEN presence-absence (binary) scores are used;

JACCA: this has been already explained. The number of equidistributed sampling units can be up to 3000. A similar option, but with less units, is present in the BASIC version.

3) IF IFIS > 0 THEN the parameters of the NC response curves must be given in input, for each curve; they include:
TYPE (format A10), PMOD, VMED, SIG, HMOD (free format)
where:

TYPE: GAUSS (gaussian function)
BIMODAL (two gaussian functions of the same standard deviation SIG but mean value VMED differing 4*SIG, in order to generate two maxima);
POISSON (poissonian profile, with the maximum on the left);
MIRRORPOIS: like POISSON, but with the maximum on the right.

PMOD: mode of the function;

VMED, SIG: already explained;

HMOD: modal value.

ELSE the parameters are randomly generated in the following way:

```
          generated          a = RANF
        IF 0     < = a    < = 0,25 THEN  TYPE = POISSON
ELSE  IF 0,25 <    a    < = 0,5   THEN    ”    = BIMODAL
ELSE  IF 0,5   <    a    <    0,75 THEN    ”    = GAUSS
ELSE  IF 0,75 < = a    < = 1      THEN    ”    = MIRRORPOIS
```

(only the GAUSS shape is provided by the APPLE II and M20ST versions)

```
          generated          b = RANF
THEN PMOD = b*IRANGE

          generated          c = RANF
THEN VMED = 2+c* (SIGMAX - SIGMIN) IF TYPE = POISSON or
       MIRRORPOIS
       ELSE    VMED = PMOD

          generated          d = RANF
THEN SIG = SIGMIN+d* (SIGMAX - SIGMIN) IF TYPE = GAUSS or
       BIMODAL
       ELSE    SIG = (VMED)^ 0,5

          generated          e = RANF
THEN HMOD = 1+e* (MAXALT - 1).
```

4) IF IFIR > 0 THEN the NR values of positions of the sampling units must be given in input (free format);
ELSE the NR sampling unit positions are randomly generated in the interval (0, IRANGE);

5) The simulated data table is written to file 2, with format (10X, 10F10. 0);

6) IF JACCA = 0 THEN the program goes back to step 1;
ELSE the table of JACCA equispaced noiseless sampling units is written to file 1 and printed (option IPLOT > = 0).

7) Similarity ratio matrix is computed, following options specified by the value of LSLP, on the theoretical table and the nearest neighbour sequence extracted.

8) IF IPLOT > 0 THEN the program goes back to step 1;
ELSE the MST groups and sequence are found and the MST sequence replaces the JACCA records of the equispaced sampling units table on file 1.

9) Then the program goes back to step 1.

Riassunto. Viene descritto un simulatore di cenoclini utile per valutare i rusultati dell'applicazione di metodi di ordinamento e classificazione a particolari strutture di dati. La simulazione di cenoclini avviene mediante il programma SPAGHET. Vengono presentati esempi di applicazione di metodi di largo uso sui dati provenienti dalla simulazione.

Acknowledgements. The work has been supported by a grant from M.P.I. 60%. The author is very grateful to prof. L. Orlóci for having revised this paper.

References

Austin M.P. (1976a) - *On non-linear species response models in ordination.* Vegetatio 33: 33-41.

Austin M.P. (1976b) - *Performance of four ordination technique assuming different non-linear species response models.* Vegetatio 33: 43-49.

Feoli E. & Lagonegro M. (1983) - *A resemblance function based on probability: Applications to field and simulated data.* Vegetatio 53: 3-9.

Gauch H.G. & Whittaker R.H. (1976) - *Simulation of community patterns.* Vegetatio 33: 13-16.

Mario Lagonegro
Dipartimento di Biologia dell'Università di Trieste
Sezione di Geobotanica ed Ecologia Vegetale

Appendix A — FORTRAN listing, sample case and output of NEWSPAG.

```
      PROGRAM NEWSPAG(INPUT=100,OUTPUT=100,TAPE1=514,TAPE2=
     1514,TAPE6=OUTPUT)
      DIMENSION ITIP(100),PMOD(100),VMED(100),SIG(100),HMOD(100)
     1,ITITLE(4)
C   (M.LAGONEGRO-1984)
C---SIMULATOR OF COENOCLINES-TO CALIBRATE SOFTWARE DETECTORS
C---ITITLE:TITLE OF THE RUN(MAX.A40/CHAR.)
C---NC,NR:N.OF RESPONSE FUNCTIONS/MAX100/N.OF RELEVES/MAX.300/
C---      RELEVES TO TAPE 2 WITH FORMAT(10X,10F10.0)
C---NBOH:ANY INTEGER FIVE DIGITS NUMBER TO INITIALIZE  RANF
C---MAXALT:MAX.VALUE FOR RESPONSE FUNCTIONS
C---IRANGE:UPPER LIMIT OF THE INTERVAL OF DEFINITION FOR
C---      RESPONSE FUNCTIONS AND GENERATED RELEVES
C---IPLOT:IF GT.0,RESP.FUNCT. WRITTEN/JACCA VALUES/
C---      TO TAPE1 (FORMAT:10X,10G10.3);IF LT.0,NO PRINTING IN SIMRAT.
C---      IN THIS CASE MIN. SP. TREE PRODUCED BY MISPAT
C---JACCA: N. OF RELEVES FOR COMPUTATION OF LIMITING VALUE OF SIMRATIO.
C---      IF LE.0,NO COMPUTATION.ZERO NOISE ASSUMED IN SIMRAT.
C---LSLP : GT.0,BINARY SCORES IN SIMRATIO,EQ.0.NORMACOVER,LT.0.COVER.
C---IFIS:IF LE.0,PARAMETERS OF FUNCTIONS RANDOMLY PRODUCED
C---      ELSE GIVEN IN INPUT/IN MAIN:ITIP,PMOD,VMED,SIG,
C---      HMOD/
C---IFIR:SAME FOR RELEVE POSITION/SUB.GENRIL.VECT.PRIL/
C---NOISE:IN %,IS THE NOISE SUPPOSED IN THE SIMULATED-DATA TABLE
C---A BLANK CARD MUST TERMINATE THE INPUT CARDS
C
      PRINT 9999
9999  FORMAT(1HT)
      REWIND 1
      REWIND 2
      NCEN=0
3     PRINT 102
102   FORMAT('OIN HEADLINE-4A10')
      READ 1,ITITLE
1     FORMAT(4A10)
      IF(ITITLE(1).EQ.1H          ) GO TO 100
      PRINT 103
103   FORMAT('OIN NS,NREL,NBOH,MAXALT,IRANGE,IPLOT,IFIS,IFIR,NOISE'/
     1' LSLP AND N. OF RELEVES FOR THEOR.VALUE OF SIMRATIO COMPUTATION')
      READ *,NC,NR,NBOH,MAXALT,IRANGE,IPLOT,IFIS,IFIR,NOISE,LSLP,JACCA
      PRINT 2,ITITLE,NC,NR,NBOH,MAXALT,IRANGE,IPLOT,IFIS,IFIR,NOISE
     1,JACCA,LSLP
2     FORMAT(1H1,4A10/' FUNCT.N.',I5/' RELEVES N.',I5/' INITIAL RANF N
     1 ',I5/' MAX.ALT.CURVE ',I5/' RANGE COENOCLINE ',I5/' IFPLOT ',I5/
     2' OPTION ON FUNCTIONS',I5/' OPTION ON RELEVES ',I5/
     3' NOISE LEVEL(%)',I4/' JACCA NUMBER',I5/' LASOLP',I3/)
      FNOIS=FLOAT(NOISE)/100.
      DO 8 I=1,NBOH
8     CALL RANF()
      NCEN=NCEN+1
      RANGE=IRANGE
      IF(IFIS.LE.0) GO TO 4
```

```
C
C---READS DATA OF RESPONSE FUNCTIONS(IF IFIS GT.0)
C---ITIP CAN BE:GAUSS    /POISSON   /MIRRORPOIS/BIMODAL   /
C
       DO 5 I=1,NC
       PRINT 204.I
204   FORMAT('OIN TYPE OF RESP.CURVE',I4,':GAUSS/BIMODAL/POISSON/'/
     1' MIRRORPOIS')
      READ 6,ITIP(I)
      PRINT 205
205   FORMAT('OIN PMOD,VMED.SIG,HMOD')
      READ *,PMOD(I),VMED(I),SIG(I),HMOD(I)
5     PRINT 11,I,ITIP(I),PMOD(I),VMED(I),SIG(I),HMOD(I)
6     FORMAT(A10)
11    FORMAT(' FUNCT.',I5,' TYPE ',A10,4F10.2)
      GO TO 7
C
C---GENERATES RANDOMLY THE NC FUNCTIONS(USES LIBRARY  RANF)
C
4     CALL GENCURV(NC,ITIP,PMOD,VMED,SIG,HMOD,MAXALT,RANGE)
7     CALL GENRIL(NC,NR,RANGE,ITIP,PMOD,VMED,SIG,HMOD,IFIR,FNOIS)
      IF(JACCA.EQ.0) GO TO 3
C
C---IF REQUESTED,TABULATES FUNCTIONS FOR PLOT ON PRINTER(IPLOT GT.0)
C---IF IPLOT LT.0,MINIMUM SPANNING TREE PRODUCED
C
      CALL SIMRAT(NC,IRANGE,ITIP,PMOD,VMED,SIG,HMOD,JACCA,IPLOT,LSLP)
      GO TO 3
100   PRINT 101,NCEN
101   FORMAT(//////' ',I5,' COENOCLINES READY TO COOK')
      STOP
      END
      SUBROUTINE GENCURV(NC,ITIP,PMOD,VMED,SIG,HMOD,MAXALT,RANGE)
      DIMENSION ITIP(1),PMOD(1),VMED(1),SIG(1),HMOD(1)
      SIGMAX=RANGE/5.
      SIGMIN=RANGE/20.
      ALT=MAXALT
      IF(NC.GT.100) NC=100
      PRINT 3
3     FORMAT(15X,'TYPE',8X,' MODA ',5X,'MEAN V.',4X,'SIGMA',6X,'MAX.VAL.
     1'/)
      DO 1 I=1,NC
      RAT=RANF()
      ITIP(I)=10HMIRRORPOIS
      IF(RAT.LE.0.25) ITIP(I)=7HPOISSON
      IF(RAT.GE.0.75) ITIP(I)=5HGAUSS
      IF(RAT.GT.0.25.AND.RAT.LE.0.5) ITIP(I)=8HBIMODAL
      PMOD(I)=RANF()*RANGE
      IM1=I-1
      DO 4 KK=1,IM1
      RAP=(PMOD(I)-PMOD(KK))/PMOD(I)
      IF(ABS(RAP).LT.0.01) PMOD(I)=RANF()*RANGE
4     CONTINUE
      VMED(I)=2.+RANF()*(SIGMAX-SIGMIN)
      IF(ITIP(I).EQ.5HGAUSS.OR.ITIP(I).EQ.8HBIMODAL ) VMED(I)=PMOD(I)
      SIG(I)=   RANF()*(SIGMAX-SIGMIN)+SIGMIN
      IF(ITIP(I).EQ.7HPOISSON.OR.ITIP(I).EQ.10HMIRRORPOIS)
```

```
     1SIG(I)=VMED(I)
      HMOD(I)=1.+RANF()*(ALT-1.)
1     PRINT 2,I,ITIP(I),PMOD(I),VMED(I),SIG(I),HMOD(I)
2     FORMAT(' FUNCT.N.:',I5,1X,A10,4G11.4)
      RETURN
      END
      SUBROUTINE GENRIL(NC,NR,RANGE,ITIP,PMOD,VMED,SIG,HMOD,IFI,RUM)
      DIMENSION ITIP(1),PMOD(1),VMED(1),SIG(1),HMOD(1),PRIL(300)
     1,QUADR(100)
      IF(IFI.LE.0) GO TO 8
      PRINT 11,NR
11    FORMAT('OIN',I5,' VALUES OF RELEVES POSITIONS')
      READ *,(PRIL(I),I=1,NR)
8     DO 1 I=1,NR
      IF(IFI.GT.0) GO TO 10
      RIL=RANF()*RANGE
      IF(I.EQ.1) GO TO 5
      IM1=I-1
      DO 4 K=1,IM1
      RAP=(RIL-PRIL(K))/RIL
      IF(ABS(RAP).LE.0.01)RIL=RANF()*RANGE
4     CONTINUE
5     PRIL(I)=RIL
10    DO 2 J=1,NC
      IF(ITIP(J).NE.5HGAUSS.AND.ITIP(J).NE.8HBIMODAL ) GO TO 3
      QUADR(J)=AINT(STAGAU(SIG(J),PRIL(I),VMED(J),HMOD(J),RUM))
      IF(ITIP(J).NE.8HBIMODAL ) GO TO 2
      SECMOD=VMED(J)+4.*SIG(J)
      IF(SECMOD.GT.RANGE)SECMOD=VMED(J)-4.*SIG(J)
      QUADR(J)=QUADR(J)+AINT(STAGAU(SIG(J),PRIL(I),SECMOD,HMOD(J),RUM))
      IF(QUADR(J).GT.HMOD(J)) QUADR(J)=HMOD(J)
      GO TO 2
3     RI=PMOD(J)-PRIL(I)-0.5
      R=AINT(VMED(J)-RI)
      IF(ITIP(J).EQ.10HMIRRORPOIS) R=AINT(VMED(J)+RI)
      QUADR(J)=AINT(POISS(R,VMED(J),HMOD(J),RUM))
2     CONTINUE
      WRITE(2,6)(QUADR(KK),KK=1,NC)
6     FORMAT(10X,10F10.0)
1     PRINT 7,I,PRIL(I),(QUADR(KK),KK=1,NC)
7     FORMAT('OREL. ',I5,' X =',F10.1/(1X,10F10.0))
      RETURN
      END
      FUNCTION POISS(AR,EM,H,RUM)
      POISS=0.
      IF(AR.LT.0.OR.AR.GT.16.) RETURN
      IAR=AR
      IF(IAR.LT.1) IAR=1
      IF(IAR.GT.16) IAR=16
      FACT=1.
      DO 1 I=1,IAR
1     FACT=FACT*FLOAT(I)
      HI=3.694528049*H*(1.+RUM*(RANF()-0.5))
      FORM=EXP(-EM)*(EM)**AR/FACT
      POISS=HI*FORM+0.5
      RETURN
      END
```

```
      FUNCTION STAGAU(SIG,AR,EM,H,RUM)
      STAGAU=0.
      DIST=ABS(AR-EM)
      RAM=3.*SIG
      IF(DIST.GT.RAM) RETURN
      HI=H*(1.+RUM*(RANF()-0.5))
      DIST=DIST/SIG
      STAGAU=EXP(-(DIST**2/2.))*HI+0.5
      IF(STAGAU.GT.H) STAGAU=H
      RETURN
      END
      SUBROUTINE SIMRAT(NC,IRANGE,ITIP,PMOD,VMED,SIG,HMOD,NP,IPLOT,LSLP)
      DIMENSION ITIP(1),PMOD(1),VMED(1),SIG(1),HMOD(1),RIG(3000)
     1,SIM(4950)
      REAL MASPE(100)
      RANGE=IRANGE
      TRAPA=0.
      IF(NP.GT.3000) NP=3000
      PASSO=RANGE/FLOAT(NP)
      DO 4 J=1,NC
      DO 1 I=1,NP
      VAL=FLOAT(I-1)*PASSO
      IF(ITIP(J).NE.5HGAUSS.AND.ITIP(J).NE.8HBIMODAL ) GO TO 3
      RIG(I)=STAGAU(SIG(J),VAL,VMED(J),HMOD(J),TRAPA)
      IF(ITIP(J).NE.8HBIMODAL ) GO TO 1
      SECMOD=VMED(J)+4.*SIG(J)
      IF(SECMOD.GT.RANGE)SECMOD=VMED(J)-4.*SIG(J)
      RIG(I)=RIG(I)+STAGAU(SIG(J),VAL,SECMOD,HMOD(J),TRAPA)
      GO TO 1
3     RI=PMOD(J)-VAL-0.5
      R=AINT(VMED(J)-RI)
      IF(ITIP(J).NE.7HPOISSON) R=AINT(VMED(J)+RI)
      RIG(I)=POISS(R,VMED(J),HMOD(J),TRAPA)
1     CONTINUE
      WRITE(1)(RIG(K),K=1,NP)
      CALL GRAND(RIG,NP,NC,MASPE,J)
      IF(IPLOT.LT.0) GO TO 4
      PRINT 6,J,(RIG(K),K=1,NP)
4     CONTINUE
6     FORMAT(' FUNCT.',I5/(1X,10G10.3))
      CALL CPSIM(NC,NP,LSLP,SIM,RIG,MASPE)
      IPLOT=-IPLOT
      CALL ORDSPEC(NC,NP,SIM,RIG,IPLOT)
      RETURN
      END
      FUNCTION ILPOCU(I,K)
      ILPOCU=0
      IF(K.EQ.I)RETURN
      IF(K.GT.I) GO TO 1
      ILPOCU=(I-2)*(I-1)/2+K
      RETURN
1     ILPOCU=(K-2)*(K-1)/2+I
      RETURN
      END
      SUBROUTINE CPSIM(NSPEC,NQUAD,IE,SIM,RIG,MASPE)
      DIMENSION SIM(1),RIG(1),ROG(3000)
      REAL MASPE(100)
```

```
      APARK=0.
      APORK=0.
      DO 3 I=1,4950
3     SIM(I)=0.
      DO 1 I=2,NSPEC
      KONT=0
      REWIND 1
11    READ(1)(RIG(IL),IL=1,NQUAD)
      KONT=KONT+1
      IF(KONT.LT.I)GO TO 11
       REWIND 1
      IM1=I-1
      DO 1 K=1,IM1
      READ(1)(ROG(IL),IL=1,NQUAD)
      PRSC=0.
      PNORMI=0.
      PNORMK=0.
      DO 2 J=1,NQUAD
      APARK=RIG(J)
      APORK=ROG(J)
      IF(IE.LT.0) GO TO 4
      IF(IE.EQ.0) GO TO 8
      IF(APARK.GT.1.)APARK=1.
      IF(APORK.GT.1.)APORK=1.
      GO TO 4
8      APARK=100.*APARK/(MASPE(I))
       APORK=100.*APORK/(MASPE(K))
4     DPRSC=APARK*APORK
      DNORMI=APARK**2
      DNORMK=APORK**2
      PRSC=PRSC+DPRSC
      PNORMI=PNORMI+DNORMI
    2 PNORMK=PNORMK+DNORMK
      DENOM=PNORMI+PNORMK-PRSC
      IF(DENOM.LE.0.) DENOM=-1.E-6
      LPOS=ILPOCU(I,K)
      SIM(LPOS)=PRSC/DENOM
    1 CONTINUE
      RETURN
      END
      SUBROUTINE ORDSPEC(NSPEC,NQUAD,SIM,RIG,ITREE)
      DIMENSION SIM(1),RIG(1),IBUF(100),ISPEC(100)
      DIMENSION IORD(100),IMXM(100),KMXM(100),VMXM(100)
      NPROG=0
      INTRUFL=9999
      DO 15 I=1,NSPEC
15    ISPEC(I)=I
14    IACEGAT=0
       IMAX=0
       KMAX=0
       SMAX=-1.E-6
      IEFATTO=NSPEC*(NSPEC-1)/2
      IORD(1)=10H.
      DO 12 I=2,10
12    IORD(I)=10H
      WRITE(6,200)
200   FORMAT(1X////4X,16HDENDROGRAM TABLE,9X,1H1,8X,2H.9,8X,2H.8,8X,2H.7
```

```
     1,8X,2H.6,8X,2H.5,8X,2H.4,8X,2H.3,8X,2H.2,8X,2H.1,8X,2H.0/1X)
      DO 6 I=1,NSPEC
6     IBUF(I)=0
C
C  FINDS FIRST MAXIMUM
C
      DO 1 I=1,NSPEC
      DO 1 K=1,NSPEC
      IF(K.EQ.I) GO TO 1
      IPO=ILPOCU(I,K)
      IF(SIM(IPO).LE.SMAX) GO TO 1
      SMAX=SIM(IPO)
      IMAX=I
      KMAX=K
1     CONTINUE
      IBUF(1)=IMAX
      IBUF(2)=KMAX
      IFOUND=1
      IACEGAT=IACEGAT+1
      IPOX=ILPOCU(IMAX,KMAX)
      VALMAX=SIM(IPOX)
      WRITE(6,100) IMAX,KMAX,VALMAX,ISPEC(IMAX),ISPEC(KMAX),(IORD(IP),IP
     1=1,10)
100   FORMAT(1X,2I4,G12.5,2I4,10A10)
      SIM(IPOX)=-SIM(IPOX)
      IMXM(IFOUND)=IMAX
      KMXM(IFOUND)=KMAX
      VMXM(IFOUND)=VALMAX
C
C  SCANS THE ROWS OF ALREADY FOUND MAXIMA TO FIND
C    THE NEXT MAXIMUM
5     SMAX=-1.E-6
      IMAX=0
       KMAX=0
      DO 2 M=1,NSPEC
      IF(IBUF(M).LE.0) GO TO 3
      IRIG=IBUF(M)
      DO 2 K=1,NSPEC
      IF(K.EQ.IRIG) GO TO 2
      IPOG=ILPOCU(IRIG,K)
      IF(SIM(IPOG).LE.SMAX) GO TO 2
      SMAX=SIM(IPOG)
      IMAX=IRIG
      KMAX=K
2     CONTINUE
3     IPOX=ILPOCU(IMAX,KMAX)
      VALMAX=SIM(IPOX)
      IACEGAT=IACEGAT+1
      SIM(IPOX)=-SIM(IPOX)
      IF(IACEGAT.GE.IEFATTO) GO TO 11
      JU=0
       JE=0
      DO 4 J=1,NSPEC
      IF(IBUF(J).EQ.IMAX) JU=1
      IF(IBUF(J).EQ.KMAX) JE=1
4     CONTINUE
      IF(JU.EQ.1.AND.JE.EQ.1) GO TO 8
```

```
C
C  STORES NEWLY FOUND MAXIMUM
C
      DO 7 M=1,NSPEC
      IF(IBUF(M).GT.0) GO TO 7
      IF(JU.EQ.1) IBUF(M)=KMAX
      IF(JE.EQ.1) IBUF(M)=IMAX
      IFOUND=IFOUND+1
      IMXM(IFOUND)=IMAX
      KMXM(IFOUND)=KMAX
      VMXM(IFOUND)=VALMAX
      IF(MOD(IFOUND,10).EQ.0) GO TO 13
      WRITE(6,100) IMAX,KMAX,VALMAX,ISPEC(IMAX),ISPEC(KMAX)
      GO TO 8
13    WRITE(6,100) IMAX,KMAX,VALMAX,ISPEC(IMAX),ISPEC(KMAX),(IORD(IP),IP
     1=1,10)
      GO TO 8
7     CONTINUE
8     IF(IBUF(NSPEC).EQ.0) GO TO 5
11    CONTINUE
      IF(ITREE.LT.0)RETURN
      CALL MISPAT(IMXM,KMXM,NSPEC,VMXM,IBUF)
      PRINT 600,(IBUF(IUK),IUK=1,NSPEC)
600   FORMAT(' MST-SEQUENCE'/(1X,20I4))
      RETURN
      END
      SUBROUTINE GRAND(VEC,NQUAD,NSPECM,MASPE,I)
      DIMENSION VEC(1)
      REAL MASPE(100)
      XMI=0.
      XMA=0.
      DO 1 J=1,NQUAD
      XMI=VEC(J)
      IF(XMI.GT.XMA)XMA=XMI
1     CONTINUE
      IF(XMA.EQ.0.)XMA=-100.
      MASPE(I)=XMA
      RETURN
      END
      SUBROUTINE MISPAT(IMX,KMX,NS,VMX,IBUF)
      DIMENSION IMX(100),KMX(100),VMX(100),LEPRIM(100),NUMG(100),
     1LMIN(100),IBUF(100),ILEAF(100),IPIVOT(100,3),MST(100,4),
     2IPUN(100)
C---CLEARS ARRAYS
      DO 1 I=1,NS
      LEPRIM(I)=0
      NUMG(I)=0
      LMIN(I)=0
      ILEAF(I)=0
      MST(I,4)=10H
      DO 1 J=1,3
      IPIVOT(I,J)=0
1     MST(I,J)=0
C---SHIFTS ONE PLACE DOWNWARDS VECTORS OF DENDROGRAM TABLE
      NS1=NS-1
      DO 22 I=1,NS1
      IMX(NS-I+1)=IMX(NS-I)
```

```
      KMX(NS-I+1)=KMX(NS-I)
 22    VMX(NS-I+1)=VMX(NS-I)
      KMX(1)=IMX(2)
      IMX(1)=0
      VMX(1)=0.
C---LOOKS FOR LEAF-LIKE ELEMENTS
      PRINT*,'OLEAF-LIKE SPECIES JOIN ONLY ONCE IN DENDR.TABLE'
      NLEAF=0
      DO 3 I=2,NS
      IACI=KMX(I-1)
      NONCE=1
      DO 4 J=I,NS
      IF(J.EQ.2)GO TO 4
      IF(IMX(J).NE.IACI) GOTO 4
      NONCE=0
 4    CONTINUE
      IF(NONCE.EQ.0)GOTO 3
      NLEAF=NLEAF+1
      PRINT 9002,NLEAF,IACI
9002  FORMAT(10X,' LEAF N.',I4,5X,I4)
      ILEAF(NLEAF)=IACI
 3    CONTINUE
      NLEAF=NLEAF+1
      ILEAF(NLEAF)=KMX(NS)
      PRINT 9002,NLEAF,ILEAF(NLEAF)
C---LOOKS FOR PIVOTAL PAIRS
      PRINT*,'OPIVOTAL PAIRS SIGNAL GROUPS - IF FINAL N.OF'
      PRINT*,' GROUPS IS LOWER THAN THAT OF PIV.PAIRS, THEN'
      PRINT*,' SOME CONTIGUOUS GROUPS HAVE BEEN GROUPED INTO'
      PRINT*,' A LARGER ONE - THIS SHOULD NOT AFFECT THE FI-'
      PRINT*,' NAL TABLE, SINCE THE GROUPS ARE HOMOGENEOUS'
      IPIVOT(1,1)=IMX(2)
      IPIVOT(1,2)=VMX(2)*1000.+.5
      IPIVOT(1,3)=KMX(2)
      NPIVOT=1
      PRINT 7,NPIVOT,(IPIVOT(1,JK),JK=1,3)
      NS2=NS-2
      DO 5 I=2,NS2
      SIM1=VMX(I)
      SIM2=VMX(I+1)
      SIM3=VMX(I+2)
      IF(SIM1.LT.SIM2.AND.SIM3.LT.SIM2)GOTO 6
      GOTO 5
 6    NPIVOT=NPIVOT+1
      IPIVOT(NPIVOT,1)=IMX(I+1)
      IPIVOT(NPIVOT,2)=SIM2*1000.+.5
      IPIVOT(NPIVOT,3)=KMX(I+1)
      PRINT 7,NPIVOT,(IPIVOT(NPIVOT,JK),JK=1,3)
 7    FORMAT(' PIV.PAIR N.',I4,' : ',I4,'(--',I4,'--)',I4)
 5    CONTINUE
      IF(VMX(NS-1).GT.VMX(NS))GO TO 8
      NPIVOT=NPIVOT+1
      IPIVOT(NPIVOT,1)=IMX(NS)
      IPIVOT(NPIVOT,2)=VMX(NS)*1000.+0.5
      IPIVOT(NPIVOT,3)=KMX(NS)
      PRINT 7,NPIVOT,(IPIVOT(NPIVOT,JK),JK=1,3)
C---SCANS THE DENDROGRAM TABLE TO FIND OUT M.S.T. GROUPS
```

```
8     KOG=2
      MST(1,1)=IMX(2)
      MST(1,2)=VMX(2)*1000.+.5
      MST(1,3)=KMX(2)
      MILEG=3000
      KG=1
      NG=1
      LEPRIM(NG)=IMX(2)
      IPUN(NG)=1
      I=1
      K=I
      KAFEL=0
      KIRK=0
      IRK=0
      N=MST(K,1)
      K=K-1
      GO TO 70
30    N=MST(K,3)
      IF(N.EQ.0)GOTO 60
      IF(N.LT.0)GOTO 50
      IF(MST(K,4).EQ.10H      LEAF)GOTO 50
70    DO 10 J=2,NS
      IF(K.EQ.0.AND.J.EQ.2)GOTO 10
      IF(N.NE.IMX(J))GOTO 10
      I=I+1
      IF(N.NE.IRK)GO TO 9
      IF(KIRK.EQ.1)GO TO 11
      KAFEL=1
      GO TO 11
9     IRK=0
      KAFEL=0
11    KOG=KOG+1
      KIRK=KIRK-1
      MST(I,1)=IMX(J)
      MST(I,2)=VMX(J)*1000+.5
      MST(I,3)=KMX(J)
      LE=LEAF(NLEAF,KMX(J),ILEAF)
      IF(LE.NE.0)MST(I,4)=10H      LEAF
      N1=IMX(J)
      N2=KMX(J)
      MI=MINUSC(NS,N1,N2,IMX,KMX,VMX,J)
      IF(KAFEL.GT.0)MI=LATG(NS,N1,N2,IMX,KMX,VMX,J)
      IZI=MPIV(N1,N2,NPIVOT,IPIVOT)
      IF(MI.EQ.0.AND.IZI.EQ.0)GOTO 10
      NG=NG+1
      LEPRIM(NG)=MST(I,3)
      LMIN(NG)=MST(I,2)
      MST(I,3)=-MST(I,3)
      IF(MST(I,2).LT.MILEG)MILEG=MST(I,2)
      KOG=KOG-1
      MST(I,4)=10H B-CARRIER
10    CONTINUE
50    K=K+1
      IF(K.GE.NS)GOTO 60
      GOTO 30
60    KG=KG+1
      NUMG(KG-1)=KOG
```

```
      KOG=1
      N=LEPRIM(KG)
      IRK=N
      KIRK=1
      IPUN(KG)=K
      LMIN(KG-1)=MILEG
      MILEG=LMIN(KG)
      IF(K.GE.NS)GOTO 40
      K=K-1
      GOTO 70
C---EXIT FROM THE MAZE
 40   CONTINUE
      REWIND 1
      NUR=0
      NTOT=0
      DO 80 I=1,NG
      PRINT 100,I
 100  FORMAT('OGROUP N.',I4)
      IUK=IPUN(I)
      IAK=IPUN(I+1)-1
      IF(I.EQ.NG)IAK=NS1
      IACI=0
      DO 90 J=IUK,IAK
      MOST=MST(J,1)
      MEST=IABS(MST(J,3))
      IOK=IPIV(MOST,MEST,NPIVOT,IPIVOT)
      IF(IOK.NE.0.AND.IACI.EQ.0) IACI=IOK
      PRINT 110,I,(MST(J,JJ),JJ=1,4)
 110  FORMAT(5X,2I4,'(--.',I4,'---',I4,5X,A10)
      WRITE(1)I,(MST(J,JJ),JJ=1,4)
      NUR=NUR+1
 90   CONTINUE
      NTOT=NTOT+NUMG(I)
 80   PRINT 120,NUMG(I),LMIN(I),(IPIVOT(IACI,II),II=1,3),NTOT
 120  FORMAT('OGR.COMP.',I4,' MIN.OUT-LINK ',I4,' PIVOTAL PAIR',I4,
     1'(--',I4,'--)',I4,' ADDED VARIABLES N.',I4)
C
C---FINDS OUT KIND OF ALL SPECIES
C
      KONT=0
      DO 2 I=1,NG
      REWIND 1
      PRINT 100,I
      ILGR=0
      DO 21 J=1,NUR
      ITOP=1H
      ITIP=7HNUCLEAR
      READ(1)IG,IACI,LEG,IOCI
      IF(IG.NE.I)GO TO 21
      ILGR=ILGR+1
      IF(IOCI.LT.0)ITIP=9HB-CARRIER
      IF(LEG.LT.LMIN(I))ITIP=7HORBITAL
      IF(IOCI.LT.0)GO TO 222
      IF(ILGR.NE.1)GO TO 7782
      KONT=KONT+1
      IBUF(KONT)=IACI
      IF(IACI.NE.LEPRIM(1))ITOP=9HB-CARRIER
```

```
7782  KONT=KONT+1
      IBUF(KONT)=IOCI
222   IAK=IOCI
      IF(IOCI.LT.0)IAK=IACI
      IF(ILGR.EQ.1)PRINT 7778,IACI,ITOP
      PRINT 7778,IAK,ITIP
21    CONTINUE
7778  FORMAT(' SPECIES',I4,' IS ',A10)
2     CONTINUE
      RETURN
      END
      FUNCTION LEAF(NS,N,ILEAF)
      DIMENSION ILEAF(1)
      LEAF=0
      DO 1 I=1,NS
      IF(N.EQ.ILEAF(I))GOTO 2
 1    CONTINUE
      RETURN
 2    LEAF=N
      RETURN
      END
      FUNCTION IPIV(N1,N2,NS,IPIVOT)
      DIMENSION IPIVOT(100,3)
      IPIV=0
      DO 1 I=1,NS
      IF(N1.EQ.IPIVOT(I,1).AND.N2.EQ.IPIVOT(I,3))GOTO 2
 1    CONTINUE
      RETURN
 2    IPIV=I
      RETURN
      END
      FUNCTION MPIV(N1,N2,NS,IPIVOT)
      DIMENSION IPIVOT(100,3)
      MPIV=0
      DO 1 I=1,NS
      IF(N1.EQ.IPIVOT(I,1).OR.N1.EQ.IPIVOT(I,3))GOTO 2
 1    CONTINUE
      RETURN
 2    K=I
      DO 3 I=1,NS
      IF(N2.EQ.IPIVOT(I,1).OR.N2.EQ.IPIVOT(I,3))GOTO 4
 3    CONTINUE
      RETURN
 4    IF(K.NE.I)MPIV=1
      RETURN
      END
      FUNCTION MINUSC(NS,N1,N2,IMX,KMX,VMX,I)
      DIMENSION IMX(1),KMX(1),VMX(1)
      MINUSC=0
      K=I
 4    K=K-1
      IF(K.LT.1)RETURN
      IF(N1.EQ.KMX(K))GOTO 30
      GOTO 4
 30   L=I
 7    L=L+1
      IF(L.GT.NS)RETURN
```

```
      IF(N2.EQ.IMX(L))GOTO 5
      GOTO 7
5     SIM1=VMX(K)
      SIM2=VMX(I)
      SIM3=VMX(L)
      IF(SIM2.LT.SIM1.AND.SIM2.LT.SIM3)MINUSC=N2
      RETURN
      END
      FUNCTION LATG(NS,N1,N2,IMX,KMX,VMX,I)
      DIMENSION IMX(1),KMX(1),VMX(1)
      LATG=0
      K=I
4     K=K-1
       IF(K.LT.1)RETURN
        IF(N1.EQ.IMX(K))GO TO 30
         GO TO 4
30    L=I
7     L=L+1
       IF(L.GT.NS)RETURN
        IF(N2.EQ.IMX(L))GO TO 5
         GO TO 7
5     SIM1=VMX(K)
       SIM2=VMX(I)
        SIM3=VMX(L)
         IF(SIM2.LT.SIM1.AND.SIM2.LT.SIM3)LATG=N2
      RETURN
      END
```

The following sample produces 10 functions and 20 releves randomly generated. Besides, 1000 sampling units are generated for the perfect measurement simulation. The 10x20 table has a 20% level noise. The meaning of the remaining parameters can be easily deduced by looking at the sample output.

```
SAMPLE CASE FOR SPAGHET
10,20,45,9,12,-1,0,0,20,0,1000
          <—LAST CARD,FIRST 10 CHAR.BLANK
```

```
OIN HEADLINE-4A10
OIN NS,NREL,NCOH,MAXALT,IRANGE,IPLOT,IFIS,IFIR,NOISE
 LSLP AND N. OF RELEVES FOR THEOR.VALUE OF SIMRATIO COMPUTATION
1SAMPLE CASE FOR SPAGHET
 FUNCT.N.   10
 RELEVES N.   20
 INITIAL RANF N      45
 MAX.ALT.CURVE     9
 RANGE COENOCLINE    12
 IFPLOT    -1
 OPTION ON FUNCTIONS    0
 OPTION ON RELEVES     0
 NOISE LEVEL(%)  20
 JACCA NUMBER 1000
 LASOLP  0
```

	TYPE	MODA	MEAN V.	SIGMA	MAX.VAL.
FUNCT.N.	1 POISSON	7.094	2.799	1.673	8.374
FUNCT.N.	2 POISSON	9.984	2.275	1.508	6.423
FUNCT.N.	3 GAUSS	4.338	4.338	1.106	4.027
FUNCT.N.	4 POISSON	3.752	3.199	1.789	5.082
FUNCT.N.	5 POISSON	4.638	2.209	1.486	6.681
FUNCT.N.	6 MIRRORPOIS	7.697	2.028	1.424	8.631
FUNCT.N.	7 BIMODAL	7.251	7.251	.7375	5.609
FUNCT.N.	8 GAUSS	6.865	6.865	2.012	5.712
FUNCT.N.	9 BIMODAL	5.431	5.431	1.723	7.435
FUNCT.N.	10 POISSON	6.195	3.140	1.772	6.616

OREL.	1 X =	6.1								
	8.	0.	1.	1.	3.	6.	2.	5.	7.	6.
OREL.	2 X =	10.2								
	1.	5.	0.	0.	0.	0.	5.	1.	0.	1.
OREL.	3 X =	9.8								
	3.	6.	0.	0.	0.	4.	5.	2.	0.	1.
OREL.	4 X =	9.2								
	3.	5.	0.	0.	0.	5.	2.	3.	1.	2.
OREL.	.5 X =	11.0								
	1.	5.	0.	0.	0.	0.	3.	1.	0.	0.
OREL.	6 X =	5.5								
	5.	0.	2.	2.	5.	6.	0.	4.	7.	5.
OREL.	7 X =	8.9								
	3.	6.	0.	0.	0.	4.	1.	4.	1.	2.
OREL.	8 X =	11.5								
	1.	3.	0.	0.	0.	0.	1.	0.	0.	0.
OREL.	9 X =	5.3								
	5.	0.	3.	2.	5.	6.	0.	5.	7.	5.
OREL.	10 X =	3.7								
	2.	0.	3.	4.	6.	1.	0.	2.	4.	3.
OREL.	11 X =	2.6								
	0.	0.	1.	4.	3.	0.	0.	1.	2.	1.
OREL.	12 X =	4.9								
	5.	0.	4.	3.	7.	3.	0.	4.	7.	5.
OREL.	13 X =	7.3								
	7.	3.	0.	1.	1.	8.	5.	5.	4.	4.
OREL.	14 X =	2.2								
	0.	0.	1.	4.	3.	0.	0.	0.	2.	1.
OREL.	15 X =	3.3								
	2.	0.	2.	4.	6.	1.	0.	1.	3.	1.
OREL.	16 X =	2.5								
	0.	0.	1.	4.	3.	0.	0.	0.	3.	1.
OREL.	17 X =	9.9								
	1.	7.	0.	0.	0.	4.	5.	2.	0.	1.
OREL.	18 X =	5.0								
	5.	0.	3.	3.	5.	3.	0.	4.	7.	5.
OREL.	19 X =	11.3								
	1.	3.	0.	0.	0.	0.	2.	0.	0.	0.
OREL.	20 X =	7.0								
	7.	3.	0.	1.	1.	8.	5.	5.	5.	4.

DENDROGRAM TABLE				1	.9	.8	.7	.6	.5	.4	.3	.2
1	8	.94534	1	8.	.	.	.	.	.	.	.	.

```
   1   6  .91189       1   6
   8  10  .88999       8  10
  10   9  .81234      10   9
   9   5  .77325       9   5
   5   3  .91681       5   3
   5   4  .88641       5   4
   6   7  .64572       6   7
   7   2  .79717       7   2
 GROUP-LEADING SPEC.AND EXT.LINK
   1    646
   5    773
   7    646
OGROUP N.   1
       1<-- 945---   8
       1<-- 912---   6
       8<-- 890---  10
       6<-- 646---  -7
      10<-- 812---   9
       9<-- 773---  -5
OGR.COMP.   5 MIN.OUT-LINK  646 MAX.LINK  945 BETWEEN   1<--->   8
 TOT.SP.N.   5
OGROUP N.   2
       5<-- 917---   3
       5<-- 886---   4
OGR.COMP.   3 MIN.OUT-LINK  773 MAX.LINK  917 BETWEEN   5<--->   3
 TOT.SP.N.   8
OGROUP N.   3
       7<-- 797---   2
OGR.COMP.   2 MIN.OUT-LINK  646 MAX.LINK  797 BETWEEN   7<--->   2
 TOT.SP.N.  10
OGROUP N.   1
 SPECIES   1 IS
 SPECIES   8 IS NUCLEAR
 SPECIES   6 IS NUCLEAR
 SPECIES  10 IS NUCLEAR
 SPECIES   6 IS B-CARRIER
 SPECIES   9 IS NUCLEAR
 SPECIES   9 IS B-CARRIER
OGROUP N.   2
 SPECIES   5 IS B-@ARRIER
 SPECIES   3 IS NUCLEAR
 SPECIES   4 IS NUCLEAR
OGROUP N.   3
 SPECIES   7 IS B-CARRIER
 SPECIES   2 IS NUCLEAR
 MST-SEQUENCE
   1   8   6  10   9   5   3   4   7   2
OIN HEADLINE-4A10



   1 COENOCLINES READY TO COOK
```

Appendix B — BASIC listing for OLIVETTI M20ST (160 kbyte-PCOS 2,0).

```
10 REM-oli20spaghet
20 ON ERROR GOTO 990
30 OPTION BASE 1
40 INPUT"give n.species,n.releves ";NS,NR
50 DIM H(NS),XM(NS),SG(NS),RP(NR),QUAD(NR,NS)
60 INPUT"give range for x ";RX
70 INPUT"give max.resp.funct.value,noise(%)";MALT,NOPC
80 NOPC=NOPC/100
90 INPUT"random gen.-y/n ";Y$
100 IF Y$="y" THEN 150
110FOR I=1 TO NS : PRINT"sp.n.";I;" give h,mode,sigma"
120 INPUT H(I),XM(I),SG(I)
130 NEXT I
140 GOTO 220
150 MXS=RX/5 : MNS=RX/20
160 DMNXS=MXS-MNS
170 FOR I=1 TO NS
180 H(I)=MALT*(.2+.8*RND(1))
190 XM(I)=RX*RND(1)
200 SG(I)=MNS+RND(1)*DMNXS
210 NEXT I
220 IF Y$="y" THEN 270
230 FOR I=1 TO NR
240 INPUT"give a pos.for rel. ";RP(I)
250 NEXT I
260 GOTO 310
270 INPUT"theocoenoc.-y/n ";TEO$ : IF TEO$="n" THEN 290
280 FOR I=1 TO NR : RP(I)=RX/NR*I : NEXT I : GOTO 310
290 FOR I=1 TO NR : RP(I)=RND(1)*RX
300 NEXT I
310 REM-printout of cenoc.characteristics
320 FOR I=1 TO NS : LPRINT "sp.";I;" h=";H(I);" mean=";XM(I);" sigma=";SG(I)
330 NEXT I
340 NAPC=NOPC*100 :LPRINT "releves table-noise(%)=";NAPC
350 FOR I=1 TO NR : LPRINT"rel.n.";I;" rel.x=";RP(I) : NEXT I
360 REM-table to file
370 INPUT "table to file-y/n ";YY$
380 FOR I=1 TO NR : FOR J-1 TO NS
390 QUAD(I,J)=0
400 DX=ABS(RP(I)-XM(J))
410 IF DX > 3*SG(J) THEN 460
420 DUS=2*SG(J)^2 : D2X=DX^2
430 DES=-D2X/DUS
440 QUAD(I,J)=H(J)*(1+NOPC*(RND(1)-1))*EXP(DES)
450 QUAD(I,J)=INT(QUAD(I,J)+.5)
460 NEXT J : NEXT I
470 REM-print table
480 FOR I=1 TO NR : FOR J=1 TO NS
490 LPRINT QUAD(I,J);" "; : NEXT J
500 LPRINT : NEXT I
510 REM-file option
520 IF YY$="n" THEN 610
530 INPUT"full filename ";N$
540 INPUT"does it already exist-y/n ";Z$
```

```
550 IF Z¤="n" THEN 570
560 KILL N¤
570 OPEN "O",#1,N¤
580 FOR I=1 TO NR : FOR J=1 TO NS
590 PRINT#1,QUAD(I,J) : NEXT J : NEXT I
600 CLOSE#1
610 CLS : COLOR 1,0
620 FX=400/RX : FY=200/MALT
630 LPRINT "fx=";FX;" fy=";FY : LPRINT
640 PSET(0,200)
650 LINE (0,200)-(0,0) : LINE (0,0)-(400,0)
660 LINE (400,0)-(400,200) : LINE (400,200)-(0,200)
670 X0=0 : Y0=0 : PSET(X0,Y0)
680 X1=0 : Y1=0
690 FOR J=1 TO NS
700 X0=0 : Y0=0
710 Y=0 : SW=0
720 FOR K=0 TO 400 STEP 5
730 KX=K/FX
740 DX=ABS(KX-XM(J))
750 TS=4*SG(J)
760 IF DX ) TS THEN 870
770 TS=TS/4
780 T2S=TS^2 : ES=-.5*(DX^2)/T2S
790 Y=H(J)*EXP(ES)*FY
800 X=K
810 IF Y ) 200 OR Y ( 0 THEN 880
820 IF X ) 400 OR X ( 0 THEN 880
830 IF SW = 0 THEN PSET(X1,Y1),1
840 PSET(X,Y),0 : LINE (X0,Y0)-(X,Y) : X0=X : Y0=Y
850 SW=1
860 GOTO 880
870 SW=0
880 NEXT K
890 PSET(0,0)
900 NEXT J
910 FOR I=1 TO NR : PUF=RP(I)*FX
920 PSET(PUF,0) : LINE (PUF,0)-(PUF,110)
930 NEXT I
940 INPUT "plot to printer-y/n ";Y¤
950 IF Y¤="n" THEN 970
960 EXEC"sp
970 CLS
980 END
990 PRINT "kapelmeister" : GOTO 980
```

Appendix C — BASIC Listing of OLIMISPAT (Olivetti Minimum Spanning Tree).

```
10 REM-OLIMISPAT(M.Lagonegro-1984)
30 INPUT"n.of data points ";C
40 INPUT"max.n.of comp.";NC
50 DIM MIN(NC),MAX(NC),C(NC,C),Y1(C),X1(C),SEQ(C),XQ(C),Z1(C),D(C,C)
60 INPUT"data file ";NA¤
70 INPUT"print coordinates-y/n ";AX¤
80 INPUT"print distance-y/n ";CO¤
90 OPEN "I",#1,NA¤
100 FOR I=1 TO NC : TT=0 : FOR J=1 TO C
110 INPUT#1,C(I,J) : TT=TT+C(I,J) : NEXT J
120 FOR J=1 TO C : C(I,J)=C(I,J)-TT/C : NEXT J
130 NEXT I
140 CLOSE#1
150 FOR I=1 TO NC : MIN(I)=C(I,1) : MAX(I)=C(I,1)
160 FOR J=1 TO C-1 : IF MIN(I) < C(I,J+1) THEN 180
170 MIN(I)=C(I,J+1)
180 IF MAX(I) > C(I,J+1) THEN 200
190 MAX(I)=C(I,J+1)
200 NEXT J
210 IF MIN(I) > 0 THEN MIN(I)=.000001
220 NEXT I
230 INPUT"first,second and third axes tag-to stop give 0,0,0 ";X1,Y1,Z1
240 IF X1*Y1*Z1=0 THEN END
250 ALFA=15 : REM-x axis perspective angle
260 LPRINT "axes x=X";X1;"    y=X";Y1;"    z=X";Z1
270 ALFR=ALFA*1.74533E-02
280 FOR I=1 TO C : X1(I)=C(X1,I) : Y1(I)=C(Y1,I)
290 Z1(I)=C(Z1,I)
300 NEXT I
310 GOSUB 1160
320 IF CO¤="n" THEN 350
330 FOR I=1 TO C : LPRINT"row ";I : FOR J=1 TO C
340 LPRINT D(I,J);" "; : NEXT J : LPRINT : NEXT I
350 FX=(MAX(X1)-MIN(X1))/350 : FY=(MAX(Y1)-MIN(Y1))/350 : FZ=(MAX(Z1)-MIN(Z1))/190
360 FOR I=1 TO C : SEQ(I)=I : XQ(I)=X1(I) :NEXT I
370 NSCMB=0 : FOR I=1 TO C-1 : IF XQ(I) <= XQ(I+1) THEN 400
380 BUF=XQ(I) : XQ(I)=XQ(I+1) : XQ(I+1)=BUF
390 BUF=SEQ(I) : SEQ(I)=SEQ(I+1) : SEQ(I+1)=BUF : NSCMB=1
400 NEXT I
410 IF NSCMB < > 0 THEN 370
420 LPRINT"horiz.sequence-x axis,from neg.to pos." : FOR I=1 TO C : LPRINT SEQ(I);" ";
430 NEXT I : LPRINT
440 CLS : COLOR 1,0
450 IF FX < FY THEN FX=FY
460 IF FY < FX THEN FY=FX
470 IF FX>FZ THEN 490
480 FX=FZ : FY=FZ : GOTO 500
490 FZ=FX
500 REM-starts drawirq axes
510 X0=200 : Y0=110 : PSET(0,Y0) : LINE (0,Y0)-(400,Y0) : PSET(X0,0) : LINE (X0,0)-(X0,220)
520 PSET(0,Y0-X0*TAN(ALFR)) : LINE (0,Y0-X0*TAN(ALFR))-(400,Y0+X0*TAN(ALFR))
530 FOR I=1 TO C : X=X1(I)/FX : Y=Y1(I)/FY : Z=Z1(I)/FZ
540 IF AX¤="y" THEN LPRINT"x";I;"=";X1(I);"  y";I;"=";Y1(I);"  z";I;"=";Z1(I)
550 X1(I)=X : Y1(I)=Y : Z1(I)=Z
560 XDIS=X0-X*COS(ALFR)+Y : YDIS=Y0-X*SIN(ALFR)+Z : XQ(I)=XDIS : SEQ(I)=YDIS : NEXT I
570 MIX=XQ(1) : MAX=MIX : MIY=SEQ(1) : MAY=MIY
580 FOR J=2 TO C
590 IF XQ(J)<MIX THEN MIX=XQ(J)
600 IF XQ(J)>MAX THEN MAX=XQ(J)
610 IF SEQ(J)<MIY THEN MIY=SEQ(J)
```

```
620 IF SEQ(J))MAY THEN MAY=SEQ(J)
630 NEXT J : IF MIX)=0 THEN MIX=.001
640 IF MIY)=0 THEN MIY=.001
650 FX=(MAX-MIX)/400 : FY=(MAY-MIY)/220
660 IF FX)FY THEN FY=FX
670 IF FY)FX THEN FX=FY
680 FOR I=1 TO C : X=X1(I)/FX : Y=Y1(I)/FX : Z=Z1(I)/FX
690 XDIS=XO-X*COS(ALFR)+Y : YDIS=YO-X*SIN(ALFR)+Z
700 PSET(XDIS,YO-X*SIN(ALFR))
710 LINE (XDIS,YO-X*SIN(ALFR))-(XDIS,YDIS)
720 XB=XO-X*COS(ALFR) : YB=YO-X*SIN(ALFR)
730 PSET(XDIS,YB)
740 LINE (XDIS,YB)-(XB,YB)
750 X1(I)=X : Y1(I)=Y : Z1(I)=Z
760 PSET(XDIS,YDIS) : CIRCLE(XDIS,YDIS),2 : NEXT I
770 X¤="la 'X',1,41" : Y¤="la 'Y',390,100" : Z¤="la 'Z',205,210" : EXEC X¤ : EXEC Y¤ : EXEC Z¤
780 NT=C : FOR I=1 TO NT : XQ(I,=I : NEXT I : KUI=1 : GOSUB 1020
790 XQ(1)=IM : SEQ(1)=JM : XQ(0)=JM
800 FOR I=2 TO C : NT=I-1
810 KUI=0 : GOSUB 1020
820 NT=NT+1 : XQ(NT)=IM : FOR I2=0 TO NT-1 : IF XQ(I2)()IM THEN 840
830 GOTO 860
840 NEXT I2
850 GOTO 900
860 XQ(NT)=JM : FOR I2=0 TO NT-1 : IF XQ(I2)()JM  THEN 880
870 NT=NT-1 : GOTO 810
880 NEXT I2
890 SEQ(NT)=IM : GOTO 910
900 SEQ(NT)=JM
910 IF NT)=C-1 THEN 914
912 NEXT I
914 LPRINT"nearest neighb. sequence"
920 FOR I=0 TO C-1 : LPRINT XQ(I);" "; : NEXT I : LPRINT : LPRINT
930 FOR I=1 TO C-1 : IM=XQ(I) : JM=SEQ(I) : GOSUB 1120
940 LPRINT"dist.=";ABS(D(IM,JM));"  pair ";IM;" ";JM : NEXT I
950 LPRINT : LPRINT
960 INPUT"plot to printer-y/n ";P¤ : IF P¤="n" THEN 990
970 EXEC"sp
980 LPRINT : LPRINT
990 CLS
1000 GOTO 230
1010 PRINT"kapelmeister" : LPRINT "kapelshuler" : END
1020 REM-finds min.dist.in residual matrix
1030 MIND=1E+30 : FOR IUK=KUI TO NT : FOR JUK=1 TO C
1040 II=XQ(IUK) : IF JUK=II  THEN 1090
1050 IF D(II,JUK)(0 THEN 1090
1060 IF D(II,JUK))=MIND THEN 1090
1070 MIND=D(II,JUK) : IM=II : JM=JUK
1080 IF D(II,JUK)=0 THEN D(II,JUK)=1E-09
1090 NEXT JUK : NEXT IUK
1100 D(IM,JM)=-D(IM,JM) : D(JM,IM)=D(IM,JM)
1110 RETURN
1120 REM-draws M.S.T. branch
1130 XA=XO-X1(IM)*COS(ALFR)+Y1(IM) : YA=YO-X1(IM)*SIN(ALFR)+Z1(IM)
1140 XC=XO-X1(JM)*COS(ALFR)+Y1(JM) : YC=YO-X1(JM)*SIN(ALFR)+Z1(JM)
1150 PSET(XA,YA) : LINE (XA,YA)-(XC,YC) : RETURN
1160 REM-computes euc.dist. for data
1170 FOR IAK=1 TO C : D(IAK,IAK)=0 : FOR JAK=IAK+1 TO C
1180 DX2=(X1(IAK)-X1(JAK))^2
1190 DY2=(Y1(IAK)-Y1(JAK))^2
1200 DZ2=(Z1(IAK)-Z1(JAK))^2
1210 D(IAK,JAK)=SQR(DX2+DY2+DZ2) : D(JAK,IAK)=D(IAK,JAK)
1220 NEXT JAK : NEXT IAK : RETURN
```