Woods Hole

WHOI-80-27

COPY 2

*Oceanographic
Institution*

WHOI
DOCUMENT
COLLECTION

PATS-1, A PACKAGE OF PROGRAMS FOR THE
ANALYSIS OF MARINE MICROPALaeONTOLOGICAL
DATA ON THE VAX 11/780 COMPUTER

by

G.P. Lohmann

June 1980

TECHNICAL REPORT

*Prepared for the National Science Foundation
under Grants OCE 78-19768, OCE 77-82149,
and OCE 79-18286.*

WOODS HOLE, MASSACHUSETTS 02543

WHOI-80-27

PATS-1, A PACKAGE OF PROGRAMS FOR THE
ANALYSIS OF MARINE MICROPALaeONTOLOGICAL
DATA ON THE VAX 11/780 COMPUTER

by

G.P. Lohmann

WOODS HOLE OCEANOGRAPHIC INSTITUTION
Woods Hole, Massachusetts 02543

June 1980

TECHNICAL REPORT

*Prepared for the National Science Foundation under
Grants OCE 78-19768, OCE 77-82149 and OCE 79-18286.*

*Reproduction in whole or in part is permitted for
any purpose of the United States Government. This
report should be cited as: Woods Hole Oceanographic
Institution Technical Report WHOI-80-27.*

Approved for Distribution:

John I. Ewing
John I. Ewing, Chairman
Department of Geology & Geophysics

Table of Contents

	<u>Page</u>
1. INTRODUCTION	1
2. PROGRAM DESIGN	1
3. IMPLEMENTATION	2
4. PROGRAMS	3
<u>Descriptions</u>	3
BARPLOT	
BENTHIC	
CHORUS	
CIRCLE	
CORREL	
COVAR	
CROSSPROD	
DATAPLOT	
DPCA	
EIGEN	
ELLIPSE	
FLIP	
HISTOGRAM	
ISOTOPE	
LINTERP	
MULTIPLY	
NANNO	
OMEGA	
PERCENT	
QPCA	
RECTERP	
REPERCENT	
RPCA	
SLICER	
SMOOTHER	
SQUARE	
TIMER	
WEATHER	
<u>Examples</u>	7
<u>Listings</u>	31

Table of Contents (cont.)

	<u>Page</u>
5. SUBROUTINES	87
<u>Descriptions</u>	87
ASCEND	
AXISAGE	
AXISDAGE	
AXISDEPTH	
AXISINTEGER	
AXISLOG10	
BOXLINE	
CERRORBAR	
CHECKDUP	
COEFF	
CONFIDAXIS	
CORREL	
COUNTERROR	
COVAR	
CROSSPROD	
CURVELINE	
DIAG	
DOT	
DSCORES	
ELLIPSE	
HATCHIT	
HOW	
MATINV	
MATPRINT	
MULTIPLY	
NORMALIZE	
OMEGA	
PERCENT	
POINTPLOT	
POLAR	
PRINCO	
QMATRIX	
QSCORES	
RATIOERROR	
RECTANG	
RERRORBAR	
ROTATE	
RSCORES	
SORTVECTOR	
SPACER	
SPLINE	
TIMER	

Table of Contents (cont).

	<u>Page</u>
TSPLLOT	
UNITVAR	
XYPLOT	
ZEROMEAN	
<u>Listings</u>	95
6. ACKNOWLEDGEMENTS	147
7. REFERENCES	148

1. INTRODUCTION

PATS-1 is the first version of a package of computer programs and subroutines written or acquired specifically for the archiving, analysis, and presentation of marine micropaleontological data. The aim of this report is to document the procedures and programs incorporated in PATS-1 and make them available to colleagues working on closely related research problems.

The program package described in this report is not intended to be either comprehensive or generally useful. Rather, the PATS-1 package was assembled for a specific purpose. It reflects the solutions to problems encountered in the course of my own work in micropaleontological oceanography. Some programs are so strictly tailored to my own research that they may have to be modified to be of any use to others. However, where convenient I have tried to generalize programs, especially in design of the subroutines incorporated into the PATS-1/LIBRARY.

This report provides program listings, documentation, examples and instructions for implementation and use of the PATS-1 package. Programs were written to be used interactively and at WHOI are run exclusively on the time-sharing system of the VAX 11/780 computer.

2. PROGRAM DESIGN

In general main programs are designed to solve a particular problem by eliciting the necessary information from the user, accessing the appropriate data, calling the required subroutines and then creating files, selected printings and plots of the results. Most of the actual computations are accomplished by subroutines.

There are essentially two kinds of main programs. "Elementary" programs accomplish single tasks such as normalizing vectors, transposing, inverting or multiplying matrices, computing correlation or dispersion matrices, or computing eigenvalues and eigenvectors. These elementary programs allow the user the freedom and flexibility to formulate and carry out his own procedures and analyses; however, they require a somewhat detailed understanding of the computational steps that are involved.

There are also "production" programs that are designed to accomplish entire analyses or predetermined procedures, such as Q or R-mode principal components analysis (p.c.a.), plotting results stratigraphically, or updating and archiving a particular suite of census variables. Although production programs allow relatively little flexibility, they require less understanding of the details of the computation and are much easier for routine analyses.

3. IMPLEMENTATION

Program listings always specify the required subroutines. However, all necessary subroutines are included in the PATS1/LIBRARY, so specification of individual subroutines is not necessary. Plotting programs also require the CALSPOOL/LIBRARY and preferably the SYMBOLDFT drafting letter set subroutine. Plots are spooled on a disk file PLOT.PLT.

Step 1. Compile program

\$FORTRAN program

Step 2. Link program with necessary subroutines

\$LINK program, PATS1/LIB

or for plotting programs,

\$LINK program, PATS1/LIB, SYMBOLDFT, CALSPOOL/LIB

Step 3. Execute program

\$RUN program

All necessary information is elicited from the user by the program.

All names for input and output files requested by the program must be entered within single quotes. Prefixes must be no larger than 9 characters, suffixes no longer than 3.

'CHARACTER.SUF'

I/O is usually list directed; except in a few cases, no formatting of data is assumed or necessary.

Samples, whatever their associated variables, are always identified by a depth (either stratigraphic or bathymetric) and an age. If age is unknown, usually set it equal to stratigraphic depth. There are no other sample identifiers, except for the name of the file in which the sample is stored.

Census data, in whatever transformation, also include the total number of individuals counted in addition to depth and age. Whenever depth, age, and total individuals are not required by a procedure, they are input as dummy variables, then ignored.

Exceptions to formatting conventions and associated variables are made clear in individual programs.

4. PROGRAMS

Descriptions

BARPLOT plots variables down-core against either depth or age. Census variables, expressed either as percents or ratios, can be plotted either as points or as interval estimates reflecting specified confidence levels. Data can be connected by either a straight, rectangular or smooth line.

A depth scale can be plotted according to the origin and scaling parameter specified for the time axis. This allows different records all to be plotted on the same time scale and still show their depth scales. Its limitation is that the age model must assume constant sedimentation rate throughout each downcore record. An initial depth and age and the sedimentation rate must be specified if this option (B) is selected. Two different data sets may be plotted on the same axes. This allows original data (points only) to be plotted over interpolated data that has been plotted and connected by a curved line (spline).

Requires subroutines COUNTERERROR, CERRORBAR, RATIOERROR, RERRORBAR, AXISPAGE, SORTVECTOR, BOXLINE, CURVELINE, CHECKDUP, AXISAGE, AXISDEPTH, ASCEND, HATCHIT, AXISINTEGER, AXISLOG10, SPACER, SPLINE, POINTPLOT, SYMBOLDFT, CALSPool/LIBRARY.

BENTHIC creates, updates and archives Quaternary benthic foram census files, computes a Q-mode p.c.a. and can estimate principal components in other samples if desired. It assumes that the taxonomy shown in table 1 has been followed.

Requires subroutines PERCENT, NORMALIZE, ASCEND, QMATRIX, HOW, QSCORES, PRINCO, XYPLOT, TSPLIT

CHORUS computes the similarity matrix that is diagonalized in correspondence analysis.

After J.E. Klován

CIRCLE plots depth or latitude vs. age distributions. Plots either samples alone, indicating their minimum and maximum estimated ages as interval bars; or plots associated data value as circles whose radii are proportional to the magnitude of the data.

Requires subroutines DOT, SYMBOLDFT, CALSPool/LIBRARY

CORREL computes a correlation matrix. Number of samples input is unlimited and resulting correlation matrix is stored in upper triangular form.

Requires subroutines CORREL and MATPRINT.

COVAR computes a dispersion (variance-covariance) matrix. Number of samples input is unlimited and resulting dispersion matrix is stored in upper triangular form.

Requires subroutines COVAR and MATPRINT.

CROSSPROD computes a cross-product (raw sums of squares and cross-products) matrix. Number of samples input is unlimited and resulting cross-product matrix is stored in upper triangular form.
Requires subroutines CROSSPROD and MATPRINT.

DATAPLOT presents a table of data in a form suitable for publication using the CALCOMP plotter.
Requires subroutines SYMBOLDFT and CALSPOOL/LIBRARY

DPCA computes a "D-mode" principal components analysis; i.e., principal components of a dispersion or variance-covariance matrix. DPCA computes loadings and scores.
Requires subroutines COVAR, HOW, DSCORES and PRINCO.

EIGEN computes the eigenvalues and eigenvectors of a real symmetric matrix. Matrix is input in upper triangular form.
Requires subroutines HOW and MATPRINT

ELLIPSE projects groups of samples as equal frequency or covariation ellipses on the principal components plane.
Requires subroutines DIAG, ELLIPSE, SYMBOLDFT and CALSPOOL/LIBRARY

FLIP transposes a matrix. The maximum product of rows times columns is 10,000.

HISTOGRAM plots a histogram.
Requires subroutines SYMBOLDFT and CALSPOOL/LIBRARY

ISOTOPE plots sets of carbon and isotope data and computes and plots their joint mean and associated covariation ellipse. Requires subroutines DIAG, CONFIDAXIS, ELLIPSE, SYMBOLDFT, and CALSPOOL/LIBRARY.

LINTERP equally spaces data by linear interpolation (after Davis, 1973).

MULTIPLY multiplies two matrices.
Requires subroutine MULTIPLY

NANNO creates or updates Miocene nannoplankton census data files and computes a Q-mode p.c.a. The taxonomy listed in table 2 is assumed. Requires subroutines PERCENT, NORMALIZE, ASCEND, QMATRIX, HOW, QSCORES, PRINCO, XYPLOT and TSPLIT.

OMEGA computes carbonate saturation from in situ temperature, salinity, depth, alkalinity and total CO₂.
Requires subroutine OMEGA

PERCENT converts census data to percent or normalizes sample vectors to unit length.
Requires subroutines PERCENT and NORMALIZE

Table 1. Census sheet for Quaternary benthic foraminifera showing taxonomy assumed for program BENTHIC.

WHOI QUATERNARY ABYSSAL BENTHIC CENSUS (1st level)

Core _____	Down-core depth _____
1. Primitive agglutinants	_____
2. Complex agglutinants	_____
3. Porcelaneous calcareous	_____
4. Perforate calcareous	_____
5. Nubeculariids (<i>Orthalimidium</i> , <i>Spiroloculina</i>)	_____
6. <i>Quinqueloculina</i>	_____
7. <i>Pyrgo</i>	_____
8. <i>Mumoloculina</i>	_____
9. Nodosariids (<i>Nodosaria</i> , <i>Dentalina</i> , <i>Lagera</i> , <i>Lenticulina</i>)	_____
10. Polymorphinids (<i>Polymorphina</i> , <i>Pyrulina</i>)	_____
11. Glandulinids (<i>Glandulina</i> , <i>Oolina</i> , <i>Fissurina</i>)	_____
12. <i>Sphaeroidina bulloides</i>	_____
13. <i>Bolivina</i>	_____
14. <i>Stilostomella</i>	_____
15. Buliminids (<i>Bulimina</i> , <i>Globobulimina</i>)	_____
16. <i>Uvigerina</i>	_____
17. <i>Epistominella exigua</i>	_____
18. <i>Laticarinina pauperata</i>	_____
19. <i>Mutallides umbonifera</i>	_____
20. <i>Rupertina</i>	_____
21. Cibicidids (<i>Planulina</i> , <i>Cibicides</i>)	_____
22. <i>Pleurostomella</i>	_____
23. <i>Ehrenbergina</i>	_____
24. <i>Favocassidulina</i>	_____
25. <i>Globocassidulina subglobosa</i>	_____
26. <i>Chilostomella</i>	_____
27. <i>Astrononion echolsi</i>	_____
28. <i>Fullenia</i>	_____
29. <i>Gyroldina</i>	_____
30. <i>Oridorsalis</i>	_____
31. <i>Osangularia</i>	_____
32. <i>Gyroldinoides</i>	_____
33. <i>Anomalina</i>	_____
34. <i>Cibicidoides</i> (<i>C. kullenbergi</i> , <i>C. bradyi</i> , <i>C. cicatricosus</i>)	_____
35. <i>Melonis</i> (<i>M. barleeanus</i> , <i>M. pompilioides</i>)	_____
36. <i>Hoeglundina elegans</i>	_____

Table 2. Census sheet for Miocene calcareous nannoplankton showing taxonomy assumed for program NANNO.

CENOP NANNO CENSUS (Miocene)

Sample _____ Sub-bottom depth _____

Preservation _____

X-nicols and light

1. R. pseudoumbilica-R. gelidus
2. R. haqii
3. C. minutulus
4. C. macintyreii
5. C. leptoporus
6. D. abisectus-R. hillae
7. C. floridanus
8. C. pelagicus-C. miopelagicus
9. Sphenoliths
10. Helicopontosphaera
11. C. rotula
12. Triquetrorhabdulus
13. Discoasters
14. Ceratoliths & Amauroliths
15. Rhabdoliths
16. Discoliths
17. Oolithotus
18. Scyphosphaera
19. Small elliptical placoliths
20. Small circular to subcircular placoliths
21. Dissolved or recrystallized nannos
22. Etc. & unidentified large nannos

phase

23. C. pataecus
24. C. radiatus
25. U. jafari

QPCA computes a Q-mode principal components analysis, both loadings and scores, a varimax rotation of the principal components and estimates principal and varimax factor loadings in other samples if desired. Requires subroutines NORMALIZE, QMATRIX, HOW, QSCORES, PRINCO, ROTATE, COEFF and MATINV

RECTERP equally spaces data by rectangular integration (after program EQSPR by Davis, 1973).

REPERCENT converts varimax factor loadings or scores to units of percent.

RPCA computes an R-mode principal components analysis, both loadings and scores, a varimax rotation of the principal components, and estimates principal and varimax factor scores in other samples if desired. Requires subroutines CORREL, HOW, RSCORES, PRINCO, ROTATE, COEFF and MATINV

SLICER strips synchronous elements from correlated time series. Time series must be identically spaced and begin at same time.

SMOOTHER smooths data sequence with m-term weighted moving average.

SQUARE computes $A'A$ and its inverse. Requires subroutine MATINV

TIMER converts stratigraphic depth to time, age estimated by linear interpolation between or, extrapolation beyond, known age levels. Requires subroutine TIMER.

WEATHER computes Miocene nanno climate and dissolution indices from five calcareous nannoplankton taxa (Lohmann, 1980).

Examples

This section is intended to illustrate the interactive aspects of the PATS-1 programs and show some of their results. Note that questions posed by the programs and intended to elicit required information from the user are indicated by the symbol .

The following data from Cooley and Lohnes (1971) are used in many of the program examples in this report and can be used to test those programs.

1	1	7	4	3
2	2	4	1	8
3	3	6	3	5
4	4	8	6	1
5	5	8	5	7
6	6	7	2	9
7	7	5	3	3
8	8	9	5	8
9	9	7	4	5
10	10	8	2	2

1. Convert census data to percent.

Input data, questions and answers appearing at the computer terminal, and a listing of the results is shown.

```
$ TYPE TESTDATA.DAT
```

```
1 1 7 4 3
2 2 4 1 8
3 3 6 3 5
4 4 8 6 1
5 5 8 5 7
6 6 7 2 9
7 7 5 3 3
8 8 9 5 8
9 9 7 4 5
10 10 8 2 2
```

```
$ RUN PERCENT
```

```
<PERCENT> OR <NORMALIZED> SAMPLE VECTORS
```

```
>COMPUTE PERCENT (P) OR UNIT (U) SAMPLE VECTORS?
```

```
P
```

```
>WHAT IS THE FILENAME OF YOUR DATA?
```

```
'TESTDATA.DAT'
```

```
>DATA FORMAT? (DEPTH, AGE, CENSUS DATA)
```

```
'(F2.0,F3.0,3F2.0)'
```

```
>NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR?
```

```
3
```

```
>GIVE RESULTS A FILENAME.
```

```
'TESTDATA.PNT'
```

```
10 SAMPLES COUNTED
```

```
FORTRAN STOP
```

```
$ TYPE TESTDATA.PNT
```

1.000000	1.000000	14.00000	50.00000
28.57143	21.42857		
2.000000	2.000000	13.00000	30.76923
7.692308	61.53846		
3.000000	3.000000	14.00000	42.85714
21.42857	35.71429		
4.000000	4.000000	15.00000	53.33334
40.00000	6.666667		
5.000000	5.000000	20.00000	40.00000
25.00000	35.00000		
6.000000	6.000000	18.00000	38.88889
11.11111	50.00000		
7.000000	7.000000	11.00000	45.45455
27.27273	27.27273		
8.000000	8.000000	22.00000	40.90909
22.72727	36.36364		
9.000000	9.000000	16.00000	43.75000
25.00000	31.25000		
10.00000	10.00000	12.00000	66.66667
16.66667	16.66667		

2. Normalize sample vectors to unit length.

<PERCENT> OR <NORMALIZE> SAMPLE VECTORS

>COMPUTE PERCENT (P) OR UNIT (U) SAMPLE VECTORS?

U

>WHAT'S THE FILENAME OF YOUR DATA?

'TEST.PNT'

♦-FORMAT ASSUMED ! (DEPTH, AGE, TOTAL, CENSUS DATA)

>NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR?

3

>GIVE RESULTS A FILENAME.

'TEST.UNIT'

10 SAMPLES COUNTED

FORTRAN STOP

1.	1.	61.	0.814	0.465	0.349
2.	2.	69.	0.444	0.111	0.889
3.	3.	60.	0.717	0.359	0.598
4.	4.	67.	0.796	0.597	0.100
5.	5.	59.	0.681	0.426	0.596
6.	6.	64.	0.605	0.173	0.777
7.	7.	60.	0.762	0.457	0.457
8.	8.	59.	0.690	0.383	0.614
9.	9.	59.	0.738	0.422	0.527
10.	10.	71.	0.943	0.236	0.236

3. Plot a table of census data in a form suitable for publication using the CALCOMP plotter.

'Number of times to overprint?' is used when plotting with ballpoint pen to insure dark, reproducible line. Input data and resulting CALCOMP plotted table is shown. Note that decimal numbers are truncated.

1	37.15	71	0	0	0	0	0	2	0	0	55	0	0	14	0	0	0
2	38.60	62	0	0	0	0	2	0	0	0	41	0	0	19	0	0	0
3	40.12	70	0	0	0	0	0	1	0	0	56	0	0	13	0	0	0
4	41.62	70	0	0	0	1	2	0	0	0	59	0	0	8	0	0	0
5	43.12	50	0	0	0	0	2	0	0	0	36	0	0	12	0	0	0
6	44.67	50	0	0	2	0	2	0	0	0	38	0	0	8	0	0	0
7	46.20	49	0	0	0	0	7	0	0	0	33	1	0	8	0	0	0
8	47.70	69	2	0	1	0	6	3	0	0	39	0	0	18	0	0	0
9	49.20	81	9	0	0	1	10	8	2	0	29	0	0	22	0	0	0
10	50.70	88	0	0	3	0	8	16	0	0	41	0	0	20	0	0	0
11	52.20	161	1	0	0	1	4	6	0	0	131	0	0	18	0	0	0
12	53.70	80	0	0	0	0	0	3	0	0	48	0	0	29	0	0	0

```

NUMBER OF ROWS, NUMBER OF COLUMNS
12 18
FILENAME?
'DSDP55.PUB'
NUMBER OF TIMES TO OVERPRINT?
2
> INPUT SCALE FACTOR
.5
FORTRAN STOP

```

1	37	71	0	0	0	0	0	2	0	0	55	0	0	14	0	0	0
2	39	62	0	0	0	0	2	0	0	0	41	0	0	19	0	0	0
3	40	70	0	0	0	0	0	1	0	0	56	0	0	13	0	0	0
4	42	70	0	0	0	1	2	0	0	0	59	0	0	8	0	0	0
5	43	50	0	0	0	0	2	0	0	0	36	0	0	12	0	0	0
6	45	50	0	0	2	0	2	0	0	0	38	0	0	8	0	0	0
7	46	49	0	0	0	0	7	0	0	0	33	1	0	8	0	0	0
8	46	69	2	0	1	0	6	3	0	0	39	0	0	18	0	0	0
9	49	81	9	0	0	1	10	8	2	0	29	0	0	22	0	0	0
10	51	88	0	0	3	0	8	16	0	0	41	0	0	20	0	0	0
11	52	161	1	0	0	1	4	6	0	0	131	0	0	18	0	0	0
12	54	80	0	0	0	0	0	3	0	0	48	0	0	29	0	0	0

4. Compute correlation matrix.

```
$ RUN CORREL
```

```
<CORREL> COMPUTES CORRELATION MATRIX !!
>FILENAME FOR INPUT DATA ? <ID: ID; ASSOCIATED VARIABLES>
'TESTDATA.DAT'
>HOW MANY VARIABLES? MAXIMUM IS 100
3
```

```
10 SAMPLES COUNTED
```

```
CORRELATION MATRIX FOR TESTDATA.DAT STORED UNDER FILE=CORREL.TMP
VECTOR OF RAW SUMS OF SQUARES STORED UNDER FILE=RAWSSQ.TMP
VECTOR OF MEANS (CENTROID VECTOR) STORED UNDER FILE=MEANS.TMP
VECTOR OF STANDARD DEVIATIONS STORED UNDER FILE=STDEVS.TMP
PRINTED RESULTS ALL SAVED ON FILE=PRINT.TMP
FORTRAN STOP
```

CORRELATION MATRIX

COMPUTED 1-DEC-80 FROM DATA UNDER FILENAME [TESTDATA.DAT]

CORRELATION MATRIX, THE MATRIX OF STANDARDIZED COVARIANCES (CORRELATION COEFFICIENTS)

SECTION 1

	1	2	3
1	0.1000E+01	0.6687E+00	-0.1013E+00
2	0.6687E+00	0.1000E+01	-0.2879E+00
3	-0.1013E+00	-0.2879E+00	0.1000E+01

RAW SUMS OF SQUARES ("VARIANCES") FROM PRINCIPAL DIAGONAL OF CROSS-PRODUCT MATRIX

COMPUTED 1-DEC-80 FOR DATA UNDER FILENAME [TESTDATA.DAT]

0.10000E+01	0.10000E+01	0.10000E+01
-------------	-------------	-------------

MEANS OR CENTROID VECTOR

COMPUTED 1-DEC-80 FOR DATA UNDER FILENAME [TESTDATA.DAT]

0.6900E+01	0.3500E+01	0.5100E+01
------------	------------	------------

5. Compute eigenvalues and eigenvectors of a real symmetric matrix.

```
$ RUN EIGEN
```

```
<EIGEN> COMPUTES EIGENVALUES & EIGENVECTORS OF A REAL SYMMETRIC MATRIX
```

```
>WHAT'S ITS FILENAME? (MATRIX MUST BE IN UPPER TRIANGULAR FORM !!)  
'CORREL.TMP'
```

```
>ORDER OF INPUT MATRIX? MAXIMUM IS 100  
3
```

```
TRACE = 3.000000
```

```
>PRINT INPUT MATRIX?
```

```
Y
```

```
>HOW MANY EIGENVECTORS DO YOU WANT EXTRACTED?  
(MAXIMUM IS THE ORDER OF YOUR INPUT MATRIX, WHICH IS 3
```

```
3
```

```
>WHAT'S THE SMALLEST EIGENVALUE YOU WANT LISTED?  
0
```

	EIGENVALUE	VARIANCE	CUM.VAR.
1	1.769	58.959	58.959
2	0.927	30.903	89.862
3	0.304	10.138	100.000

```
>HOW MANY EIGENVECTORS DO YOU WANT TO SAVE?  
3
```

```
EIGENVALUES AND EIGENVECTORS FOR DATA (CORREL.TMP 3 ARE  
STORED UNDER FILES "EIGENVAL.TMP" & "EIGENVEC.TMP"  
ALL PRINTED RESULTS IN "PRINT.TMP" FILE  
FORTRAN STOP
```

```
$ TYPE EIGENVAL.TMP
```

```
1.768774 0.9270758 0.3041499
```

```
$ TYPE EIGENVEC.TMP
```

```
0.6420043 0.3846720 -0.6632174
```

```
0.6863620 9.7130425E-02 0.7207451
```

```
-0.3416693 0.9179288 0.2016663
```

COMPUTATION OF EIGENVALUES AND EIGENVECTORS OF [CORREL.TMP 1 1-DEC-80

SYMMETRIC MATRIX INPUT TO SUBROUTINE <HOW>

SECTION 1

ROW	1	2	3
1	0.1000E+01	0.6687E+00	-0.1013E+00
2	0.6687E+00	0.1000E+01	-0.2879E+00
3	-0.1013E+00	-0.2879E+00	0.1000E+01

TRACE OF INPUT MATRIX = 3.000000

EIGENVALUES GREATER THAN 0.000000

COMPUTED 1-DEC-80 FOR DATA STORED UNDER FILENAME [CORREL.TMP 1

	EIGENVALUE	VARIANCE	CUM.VAR.
1	1.769	58.959	58.959
2	0.927	30.903	89.862
3	0.304	10.138	100.000

EIGENVECTORS

COMPUTED 1-DEC-80 FOR DATA STORED UNDER FILENAME [CORREL.TMP 1

MATRIX OF EIGENVECTORS

1	0.642	0.385	-0.663
2	0.686	0.097	0.721
3	-0.342	0.918	0.202

6. Compute a Q-mode principal components analysis (p.c.a.) and a varimax rotation of the principal components.

\$ RUN QPCA

THIS IS <QPCA> :

IT COMPUTES A Q-MODE P.C.A AND A VARIMAX ROTATION

>WHAT IS THE FILENAME FOR THE DATA TO BE ANALYZED?
"SAMPLE DEPTH(M); AGE(M.Y.); TOTAL COUNTED; CENSUS DATA AS PERCENT"

<TESTDATA.PNT'
>HOW MANY ELEMENTS IN EACH SAMPLE VECTOR?

3

CENSUS DATA HAS BEEN INPUT !
Q-MATRIX HAS BEEN COMPUTED !

TRACE = 10.0

>HOW MANY EIGENVECTORS DO YOU WANT EXTRACTED?

3

EIGENVALUES AND EIGENVECTORS HAVE BEEN COMPUTED !

	EIGENVALUE	% VARIANCE	CUM.VAR.
1	9.16	91.6	91.6
2	0.75	7.5	99.0
3	0.10	1.0	100.0

>HOW MANY EIGENVECTORS DO YOU WANT TO SAVE AND ROTATE?

3

Q-SCORES HAVE BEEN COMPUTED & ARE ON FILE=Q-SCORES.TMP !!
SCORES HAVE BEEN PRINTED !

EIGENVECTORS HAVE BEEN SCALED TO PRINCIPAL COMPONENTS !
THEY ARE STORED ON FILE=Q-LOADINGS.TMP !

VARIMAX ROTATION OF 3 FACTORS BASED ON 10 TESTS
PROPORTION OF VARIANCE ACCOUNTED FOR BY EACH FACTOR.

FACTOR 1 VAR. PROP. =0.519

FACTOR 2 VAR. PROP. =0.468

FACTOR 3 VAR. PROP. =0.012

PRINCIPAL COMPONENTS ROTATED TO VARIMAX FACTORS !

THEY ARE STORED ON FILE=Q-VMXLOAD.TMP !

ROTATED FACTOR SCORE COEFFICIENTS HAVE BEEN COMPUTED !

VARIMAX FACTOR SCORES STORED ON FILE=Q-VMXSCOR.TMP !

VARIMAX FACTOR SCORES HAVE BEEN PRINTED !

>ESTIMATE LOADINGS OF OTHER SAMPLES ON FACTOR AXES?

IF YES, INDICATE PRINCIPAL COMPONENTS(P) OR FACTORS(V)

ALL FINISHED !

PRINTED RESULTS ARE ON FILE=PRINT.TMP !

FORTRAN STOP

\$ TYPE QLOADINGS.TMP

1.000000	1.000000	0.9769468	0.2131994	1.0897570E-02
2.000000	2.000000	0.8540297	-0.5200269	-1.4311359E-02
3.000000	3.000000	0.9972669	-7.3458053E-02	8.0124522E-03
4.000000	4.000000	0.8801966	0.4649280	9.5381424E-02
5.000000	5.000000	0.9947332	-6.0165741E-02	8.2995504E-02
6.000000	6.000000	0.9383424	-0.3399414	-6.2886976E-02
7.000000	7.000000	0.9939168	9.9570811E-02	4.7086604E-02
8.000000	8.000000	0.9951194	-8.7644912E-02	4.5314085E-02
9.000000	9.000000	0.9990422	1.8012073E-02	3.9906722E-02
10.00000	10.00000	0.9259057	0.2688731	-0.2653393

\$ TYPE QSCORES.TMP

1.000000	1.000000	0.7524740	0.3956098	-0.5265694
2.000000	2.000000	0.3815344	0.3898588	0.8381172
3.000000	3.000000	0.5368553	-0.8315660	0.1424199

\$ TYPE QVMXLOAD.TMP

1.000000	1.000000	0.8570771	-0.5133446	-4.3523114E-02
2.000000	2.000000	0.2646604	-0.9632336	-4.6209533E-02
3.000000	3.000000	0.6755696	-0.7361513	-4.1091006E-02
4.000000	4.000000	0.9633511	-0.2652209	4.0178131E-02
5.000000	5.000000	0.6867372	-0.7261299	3.3611201E-02
6.000000	6.000000	0.4466714	-0.8887521	-0.1029851
7.000000	7.000000	0.7935636	-0.6084620	-5.7372525E-03
8.000000	8.000000	0.6662484	-0.7457204	-3.4234989E-03
9.000000	9.000000	0.7411125	-0.6712864	-1.1361140E-02
10.00000	10.00000	0.8436175	-0.4326693	-0.3179715

\$ TYPE QVMXSCOR.TMP

1.000000	1.000000	0.7905372	-0.2167191	-0.5727868
2.000000	2.000000	0.5880458	7.4213333E-03	0.8087917
3.000000	3.000000	-0.1710302	-0.9762053	0.1333072

\$

\$ TYPE RLOADINGS.TMP

1.000000	1.000000	0.8538353	0.3703806	-0.3657630
2.000000	2.000000	0.9128290	9.3521878E-02	0.3974894
3.000000	3.000000	-0.4544042	0.8838257	0.1112186

Q-MODE P.C.A. OF DATA STORED IN FILE (TESTDATA.PNT) COMPUTED 1-DEC-80

3 ELEMENTS OF CENSUS DATA SPECIFIED FOR EACH SAMPLE VECTOR
10 SAMPLE VECTORS COUNTED

NO.	DEPTH (M)	AGE (M.Y.)	TOTAL CENSUS DATA INPUT AS PERCENT FROM FILE (TESTDATA.PNT)			
1	1.000	1.000	14	50	29	21
2	2.000	2.000	13	31	8	62
3	3.000	3.000	14	43	21	36
4	4.000	4.000	15	53	40	7
5	5.000	5.000	20	40	25	35
6	6.000	6.000	18	39	11	50
7	7.000	7.000	11	45	27	27
8	8.000	8.000	22	41	23	36
9	9.000	9.000	16	44	25	31
10	10.000	10.000	12	67	17	17

Q-MODE PRINCIPAL COMPONENTS ANALYSIS OF DATA IN FILE (TESTDATA.PNT)

EIGENVALUES AND SCORES FOR Q-MODE EIGENVECTORS, COMPUTED 1-DEC-80

	EIGENVALUE	% VARIANCE	CUM.VAR.
1	9.16	91.6	91.6
2	0.75	7.5	99.0
3	0.10	1.0	100.0

SCORES OF CENSUS VARIABLES ON Q-MODE PRINCIPAL COMPONENTS
FOR DATA ON FILE (TESTDATA.PNT)

VAR.NO. PRINCIPAL COMPONENTS SCORES (* 100)

1	75.2	39.6	-52.7
2	38.2	39.0	83.8
3	53.7	-83.2	14.2

EIGENVECTORS OF Q-MATRIX SCALED BY EIGENVALUES TO Q-MODE PRINCIPAL COMPONENTS
FOR DATA ON FILE (TESTDATA.PNT) ON 1-DEC-80

NO.	DEPTH (M)	AGE (M.Y.)	% EXPLAINED	LOADINGS ON Q-MODE PRINCIPAL COMPONENTS (* 100)		
1	1.000	1.000	100	97.7	21.3	1.1
2	2.000	2.000	100	85.4	-52.0	-1.4
3	3.000	3.000	100	99.7	-7.3	0.8
4	4.000	4.000	100	88.0	46.5	9.5
5	5.000	5.000	100	99.5	-6.0	8.3
6	6.000	6.000	100	93.8	-34.0	-6.3
7	7.000	7.000	100	99.4	10.0	4.7
8	8.000	8.000	100	99.5	-8.8	4.5
9	9.000	9.000	100	99.9	1.8	4.0
10	10.000	10.000	100	92.6	26.9	-25.5

100% OF THE TOTAL OBSERVED VARIANCE ACCOUNTED FOR

Q-MODE PRINCIPAL COMPONENTS ROTATED TO VARIMAX FACTORS
FOR DATA ON FILE (TESTDATA.PNT) ON 1-DEC-80

NO.	DEPTH (M)	AGE (M.Y.)	% EXPLAINED	LOADINGS ON Q-MODE VARIMAX FACTORS (* 100)		
1	1.000	1.000	100	85.7	-51.3	-4.4
2	2.000	2.000	100	26.5	-96.3	-4.6
3	3.000	3.000	100	67.6	-73.6	-4.1
4	4.000	4.000	100	96.3	-26.5	4.0
5	5.000	5.000	100	68.7	-72.6	3.4
6	6.000	6.000	100	44.7	-88.9	-10.3
7	7.000	7.000	100	79.4	-60.8	-0.6
8	8.000	8.000	100	66.6	-74.6	-0.3
9	9.000	9.000	100	74.1	-57.1	-1.1
10	10.000	10.000	100	84.4	-43.3	-31.8

VARIANCE ACCOUNTED FOR

52%

47%

1%

SCORES OF CENSUS VARIABLES ON Q-MODE VARIMAX FACTORS
FOR DATA ON FILE (TESTDATA.PNT) ON 1-DEC-80

VAR.NO. VARIMAX FACTOR SCORES (* 100)

1	79.1	-21.7	-27.3
2	58.8	0.7	60.9
3	-17.1	-97.5	13.3

7. Q-mode p.c.a. of unlimited number of samples

In conventional Q-mode p.c.a. the matrix analyzed is the cosine theta matrix, Q , computed as the product $U'U$. U is the matrix of sample vectors, each normalized to unit length. The order of Q is n , the number of samples.

A Q-mode p.c.a. can also be computed from UU' (Pielou, 1969). The resulting matrix to be analyzed is the "pseudo cosine theta matrix" of Klován and Imbrie (1971). It is of order m , the number of variables.

When the number of samples is larger than the number of variables, it is more efficient to compute a Q-mode p.c.a. from UU' . Using this procedure and the PATS-1 package of programs the number of samples which can be analyzed is practically unlimited.

X is the m by n matrix of sample vectors (usually stored as n by m matrix X')

n is the number of samples

m is the number of variables; for the PATS-1 package m must not exceed 100.

U is the m by n matrix of sample vectors, each normalized to unit length

$U'U$ is the cosine theta matrix (theta is the angle between sample vectors and cosine theta is a measure of proportional similarity)

UU' is the "pseudo cosine theta matrix"

The eigenvalues of $U'U$ and UU' are the same (Pielou, 1969)

The eigenvectors of UU' are the "principal factor scores" of $U'U$ and are reported as unit vectors

The product of U' and the eigenvectors of UU' is the matrix of "principal factor loadings" of $U'U$

Step 1. Normalize sample vectors in X to unit length using option U in PERCENT. X and U are usually stored as n by m matrices X' and U' .

Step 2. Compute UU' using CROSSPROD. Since U is usually stored as U' , U' must be entered into CROSSPROD. Otherwise first transpose U to U' using FLIP.

Step 3. Compute the eigenvalues and eigenvectors of UU' using EIGEN. The eigenvalues are the same as those of $U'U$, a conventional Q-mode p.c.a. The eigenvectors are the "principal factor scores", the F -prime matrix of Imbrie and Kipp (1971).

Step 4. Compute the "Q-mode principal factor loadings", the product of U' and the eigenvectors of UU' . Use MULTIPLY. The result is the same as the eigenvectors of the cosine theta matrix scaled by the reciprocals of the square roots of the associated eigenvalues.

8. Compute an R-mode p.c.a. and a varimax rotation of the principal components.

```
$ RUN PPCA
```

```
THIS IS <PPCA> :
```

```
IT COMPUTES AN R-MODE P.C.A AND A VARIMAX ROTATION
```

```
>WHAT'S THE FILENAME FOR THE DATA TO BE ANALYZED?  
"SAMPLE DEPTH(M), AGE(M.Y.), ASSOCIATED DATA VARIABLES"  
'TESTDATA.DAT'
```

```
>HOW MANY VARIABLES IN EACH SAMPLE VECTOR?
```

```
3
```

```
DATA HAS BEEN INPUT :
```

```
10 SAMPLES COUNTED
```

```
R-MATRIX HAS BEEN COMPUTED :
```

```
TRACE = 3.0
```

```
>HOW MANY EIGENVECTORS DO YOU WANT EXTRACTED?
```

```
3
```

```
EIGENVALUES AND EIGENVECTORS HAVE BEEN COMPUTED :
```

	EIGENVALUE	% VARIANCE	CUM.VAR.
1	1.769	59.0	59.0
2	0.927	30.9	89.9
3	0.304	10.1	100.0

```
>HOW MANY EIGENVECTORS DO YOU WANT TO SAVE AND ROTATE?
```

```
3
```

```
R-SCORES HAVE BEEN STORED AS FILE "RSCORES.TMP" :
```

```
EIGENVECTORS HAVE BEEN SCALED TO PRINCIPAL COMPONENTS :
```

```
R-MODE PRINCIPAL COMPONENTS STORED IN "RLOADINGS.TMP"
```

```
VARIMAX ROTATION OF 3 FACTORS BASED ON 3 TESTS
```

```
PROPORTION OF VARIANCE ACCOUNTED FOR BY EACH FACTOR.
```

```
FACTOR 1 VAR. PROP. =0.339
```

```
FACTOR 2 VAR. PROP. =0.336
```

```
FACTOR 3 VAR. PROP. =0.325
```

```
PRINCIPAL COMPONENTS ROTATED TO VARIMAX FACTORS :
```

```
ROTATED FACTOR SCORE COEFFICIENTS HAVE BEEN COMPUTED :
```

```
VARIMAX FACTOR SCORES STORED IN "RVMXSCOR.TMP" :
```

```
>ESTIMATE SCORES OF OTHER SAMPLES ON FACTOR AXES?
```

```
IF YES, INDICATE PRINCIPAL COMPONENTS<P> OR FACTORS<V>
```

```
ALL FINISHED :
```

```
PRINTED RESULTS ARE ON FILE "PRINT.TMP" :
```

```
FORTRAN STOP
```

\$ TYPE RLOADINGS.TMP					
1.000000	1.000000	0.8538353	0.3703806	-0.3657630	
2.000000	2.000000	0.9128290	9.3521878E-02	0.3974894	
3.000000	3.000000	-0.4544042	0.8838257	0.1112186	
\$ TYPE RSCORES.TMP					
1.000000	1.000000	0.4080300	-0.6906171	6.4051211E-02	
2.000000	2.000000	-2.108268	6.8761885E-02	0.6324460	
3.000000	3.000000	-0.4628969	-0.3181450	0.2992928	
4.000000	4.000000	1.623010	-0.9958371	0.7000669	
5.000000	5.000000	0.7000642	1.085134	0.6527917	
6.000000	6.000000	-0.8589693	1.323102	-0.8544813	
7.000000	7.000000	-0.6038420	-1.310570	0.8564705	
8.000000	8.000000	0.9374913	1.719522	-4.1716188E-02	
9.000000	9.000000	0.2150657	2.5457419E-02	0.3387119	
10.00000	10.00000	0.1503149	-0.9068086	-2.647632	
\$ TYPE RVMXLOAD.TMP					
1.000000	1.000000	0.9375150	-2.8189948E-02	0.3467993	
2.000000	2.000000	0.3698687	-0.1600776	0.9151902	
3.000000	3.000000	-3.0589564E-02	0.9911891	-0.1288769	
\$ TYPE RVMXSCOR.TMP					
1.000000	1.000000	-4.0517587E-02	-0.7648407	0.2468183	
2.000000	2.000000	-1.761217	0.9211571	-0.9482086	
3.000000	3.000000	-0.6193513	-7.3486596E-02	-0.1267695	
4.000000	4.000000	0.2172525	-1.352152	1.496807	
5.000000	5.000000	0.4435015	0.8721924	1.065987	
6.000000	6.000000	0.5067953	1.355460	-1.060401	
7.000000	7.000000	-1.456039	-0.8303474	7.8988999E-02	
8.000000	8.000000	1.295992	1.240989	0.7859653	
9.000000	9.000000	-7.2122477E-02	1.1288386E-02	0.3953454	
10.00000	10.00000	1.485705	-1.380260	-1.934532	

R-MODE P.C.A. OF DATA STORED IN FILE [TESTDATA.DAT] COMPUTED 1-DEC-80

3 DATA VARIABLES SPECIFIED FOR EACH SAMPLE VECTOR

NO.	DEPTH (M)	AGE (M.Y.)	ASSOCIATED DATA VARIABLES INPUT FROM FILE [TESTDATA.DAT]		
1	1.000	1.000	7	4	3
2	2.000	2.000	4	1	8
3	3.000	3.000	6	3	5
4	4.000	4.000	8	6	1
5	5.000	5.000	8	5	7
6	6.000	6.000	7	2	9
7	7.000	7.000	5	3	3
8	8.000	8.000	9	5	8
9	9.000	9.000	7	4	5
10	10.000	10.000	8	2	2

STATISTICS FOR INPUT DATA FILE [TESTDATA.DAT]

NO.	MEAN	STD. DEV.
1	6.900	1.446
2	3.500	1.500
3	5.100	2.663

R-MODE PRINCIPAL COMPONENTS ANALYSIS OF DATA IN FILE [TESTDATA.DAT]
EIGENVALUES FOR R-MODE EIGENVECTORS COMPUTED 1-DEC-80

	EIGENVALUE	% VARIANCE	CUM. VAR.
1	1.709	59.0	59.0
2	0.927	30.9	89.9
3	0.304	10.1	100.0

SCORES OF SAMPLES ON R-MODE PRINCIPAL COMPONENTS
OF DATA FROM FILE [TESTDATA.DAT] ON 1-DEC-80

NO.	DEPTH(M)	AGE(M.Y.)	R-MODE PRINCIPAL COMPONENTS SCORES		
1	1.000	1.000	0.4	-0.7	0.1
2	2.000	2.000	-2.1	0.1	0.6
3	3.000	3.000	-0.5	-0.3	0.3
4	4.000	4.000	1.6	-1.0	0.7
5	5.000	5.000	0.7	1.1	0.7
6	6.000	6.000	-0.9	1.3	-0.9
7	7.000	7.000	-0.6	-1.3	0.9
8	8.000	8.000	0.9	1.7	0.0
9	9.000	9.000	0.2	0.0	0.3
10	10.000	10.000	0.2	-0.9	-2.6

EIGENVECTORS OF R-MATRIX SCALED BY EIGENVALUES TO R-MODE PRINCIPAL COMPONENTS
OF DATA IN FILE [TESTDATA.DAT]

VAR. NO. % EXPLAINED R-MODE PRINCIPAL COMPONENTS LOADINGS(*100)

1	100	85.4	37.0	-36.6
2	100	91.3	9.4	39.7
3	100	-45.4	88.4	11.1

100% OF THE TOTAL OBSERVED VARIANCE ACCOUNTED FOR

R-MODE PRINCIPAL COMPONENTS ROTATED TO VARIMAX FACTORS
FOR DATA IN FILE (TESTDATA.DAT) ON 1-DEC-80

NO. % EXPLAINED LOADINGS ON R-MODE VARIMAX FACTORS (* 100)

1	100	93.8	-2.8	34.7
2	100	37.0	-16.0	91.5
3	100	-3.1	99.1	-12.9

VARIANCE	34%	34%	32%
----------	-----	-----	-----

SCORES OF SAMPLES ON R-MODE VARIMAX FACTORS
FOR DATA IN FILE (TESTDATA.DAT) ON 1-DEC-80

NO. DEPTH(M) AGE(M.Y.) VARIMAX FACTOR SCORES

1	1.000	1.000	0.0	-0.3	0.2
2	2.000	2.000	-1.3	0.9	-0.9
3	3.000	3.000	-0.6	-0.1	-0.1
4	4.000	4.000	0.2	-1.4	1.5
5	5.000	5.000	0.4	0.9	1.1
6	6.000	6.000	0.5	1.4	-1.1
7	7.000	7.000	-1.5	-0.8	0.1
8	8.000	8.000	1.3	1.2	0.8
9	9.000	9.000	-0.1	0.0	0.4
10	10.000	10.000	1.5	-1.4	-1.9

9. Update and archive Miocene calcareous nannoplankton census data.

THIS IS <NANNO> !

<NANNO> CREATES OR UPDATES NANNOFOSSIL CENSUS DATA FILES
IT ALSO COMPUTES A Q-MODE PRINCIPAL COMPONENTS ANALYSIS

>BEFORE YOU INPUT NEW DATA, IS THERE ALREADY DATA ON FILE FOR THIS SITE?

ANSWER Y OR N

N

>THEN GIVE YOUR NEW DATA FILE A NAME (UP TO 9 "CHARACTER.PNT")
'DSDP289.DAT'

>TYPE IN YOUR NEW NANNO CENSUS DATA:

>FIRST, LABEL EACH SAMPLE WITH ITS SUBBOTTOM DEPTH (IN METERS)

>THEN, TYPE IN COUNTS FOR THE 25 NANNO TAXA, EACH COUNT SEPARATED BY A SPACE

205.505 24 16 0 2 5 0 0 9 6 3 1 0 0 0 0 0 0 119 2 42 1 2 1 0

>CHECK THE CENSUS DATA FOR THIS SAMPLE. IS IT CORRECT?

Y

>ANY MORE SAMPLES FOR THIS SITE?

Y

>THEN INPUT THE DEPTH AND COUNTS FOR YOUR NEXT SAMPLE

196.005 39 6 0 0 5 0 0 18 5 2 0 0 1 0 0 0 0 139 0 53 0 3 3 0

>CHECK THE CENSUS DATA FOR THIS SAMPLE. IS IT CORRECT?

Y

>ANY MORE SAMPLES FOR THIS SITE?

N

0 OLD SAMPLES;

2 NEW SAMPLES

SAMPLES HAVE BEEN RESORTED BY THEIR SUB-BOTTOM DEPTHS !

CENSUS DATA HAS BEEN OUTPUT TO A DISC FILE !

CENSUS DATA HAS BEEN PRINTED !

Q-MATRIX HAS BEEN COMPUTED !

EIGENVALUES & EIGENVECTORS HAVE BEEN COMPUTED !

Q-SCOPES HAVE BEEN COMPUTED !

RESULTS ARE NOW BEING PRINTED !

PRINCIPAL COMPONENTS HAVE BEEN COMPUTED !

RESULTS HAVE BEEN PLOTTED !

FINISHED !

CENOP CENSUS DATA FILE OF MIOCENE CALCAREOUS NANNOPLANKTON [DSDP317B.PNT] AS OF 15-DEC-79

UPDATED 15-DEC-79 AND STORED UNDER FILENAME [DSDP317B.PNT]

21 SAMPLES IN PREVIOUS DATA FILE; 4 NEW SAMPLES ADDED

NANNOPLANKTON CENSUS DATA (EXPRESSED AS PERCENT)

NO.	DEPTH	TOTAL	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	A	B	C
1	54.44✓	86	3	0	0	0	3	0	0	2	27	0	0	0	0	0	0	0	0	0	31	1	19	1	13	1	0
2	57.44✓	136	2	0	0	1	1	0	0	0	15	0	0	0	1	0	0	0	0	0	56	0	13	0	4	3	0
3	58.44✓	130	2	0	0	1	2	0	0	0	25	0	0	0	1	0	0	0	0	0	39	1	16	0	13	0	1
4	60.51✓	139	0	0	0	0	4	0	0	0	24	0	0	0	1	0	0	0	0	0	46	0	13	0	12	0	0
5	64.51✓	87	9	0	0	0	3	0	0	0	38	2	0	0	7	0	0	0	0	1	16	1	17	0	5	0	0
6	65.44✓	110	22	0	0	0	2	0	0	0	20	1	0	0	14	1	0	0	0	1	12	3	15	0	7	2	1
7	66.44✓	122	7	0	0	1	4	0	0	1	29	0	0	0	2	0	0	0	0	0	33	2	7	1	11	0	1
8	68.48✓	101	6	0	0	2	3	0	0	1	38	2	0	0	6	0	0	0	0	0	19	8	8	0	4	2	7
9	69.44✓	192	9	0	0	0	5	0	0	7	34	1	0	0	1	0	0	0	1	1	29	1	8	2	2	1	0
10	71.44✓	82	4	0	0	0	4	0	0	1	37	0	0	0	1	0	0	0	0	0	44	4	1	0	4	1	0
11	73.44✓	223	11	0	0	1	3	0	0	1	47	2	0	0	1	0	0	0	0	0	17	0	7	0	13	0	0
12	74.44✓	110	16	0	0	2	6	0	0	2	30	1	0	0	1	0	0	0	0	0	25	0	4	0	17	0	1
13	76.44✓	160	23	0	0	0	1	0	0	3	41	0	1	0	1	0	0	0	0	0	16	1	4	1	9	1	1
14	77.48✓	129	0	0	0	0	4	0	0	0	46	2	0	1	0	0	0	0	0	0	19	0	22	0	5	1	2
15	79.47✓	116	3	1	0	0	3	0	0	3	41	1	0	0	1	0	0	0	0	2	28	0	10	0	5	0	0
16	80.44✓	96	0	0	0	1	2	0	0	0	42	2	2	0	2	0	0	0	0	2	24	1	14	0	5	0	0
17	82.44✓	121	1	0	0	2	2	0	0	0	55	1	0	0	2	0	0	0	0	0	21	1	8	0	7	0	0
18	84.47✓	128	0	0	0	2	3	0	0	2	58	2	0	0	2	0	0	0	0	1	22	3	2	0	3	0	0
19	86.03✓	185	0	0	0	0	3	0	0	2	63	2	1	0	0	0	0	0	0	0	15	7	4	0	7	1	0
20	87.53✓	136	0	1	0	0	0	0	0	0	40	0	1	0	0	0	0	0	0	0	7	7	2	1	5	1	1
21	89.48✓	58	0	1	0	1	1	0	0	3	57	0	0	0	0	0	0	0	0	4	18	4	4	1	3	0	0
22	90.48✓	88	0	0	0	0	8	0	0	1	51	0	0	0	1	0	0	0	0	0	23	5	2	1	5	0	0
23	93.48✓	91	2	0	0	0	1	0	0	0	64	1	0	0	1	0	0	0	0	1	15	2	2	1	7	0	0
24	95.48✓	122	0	0	0	1	4	0	0	3	57	0	0	0	1	0	0	0	0	1	7	11	1	0	15	0	0
25	97.76✓	109	1	0	0	1	1	0	0	2	48	1	0	0	1	0	0	0	0	0	33	5	3	0	6	0	0

NANNOPLANKTON FLORAL COMPOSITIONS (SCORES) FOR THE 1ST & 2ND Q-MODE EIGENVECTORS, CENOP SITE [DSDP317B.PNT]

EIGENVECTOR	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	A	B	C
1ST (NORMALIZED)	11	0	0	1	6	0	0	3	82	2	0	0	4	0	0	0	0	1	50	5	17	1	18	1	1
1ST (AS PERCENT)	5%	0%	0%	1%	3%	0%	0%	1%	40%	1%	0%	0%	2%	0%	0%	0%	0%	1%	24%	2%	9%	0%	9%	1%	0%
2ND (NORMALIZED)	-7	0	0	0	-2	0	0	1	54	2	1	0	-2	0	0	0	0	2	-80	7	-24	0	-1	-2	0

(NOTE THAT ALL COMPOSITIONS GIVEN IN NORMALIZED FORM HAVE BEEN MULTIPLIED BY 100)

THE 1ST Q-MODE EIGENVECTOR REPRESENTS THE "AVERAGE" SAMPLE FOR THIS DATA;
IT ALONE DESCRIBES 85.8% OF THE OBSERVATIONS.

THE 2ND EIGENVECTOR DESCRIBES THE PRINCIPAL DIMENSION OF DEVIATIONS IN NANNOFLORAL COMPOSITIONS FROM THIS "AVERAGE";
IT ACCOUNTS FOR AN ADDITIONAL 7.6%

1ST & 2ND Q-MODE PRINCIPAL COMPONENTS OF MIOCENE NANNOPLANKTON FOR CENOP FILE [DSDP317B.PNT]

	DEPTH	1ST	2ND	% EXPLAINED
1	54.49M	0.93	-0.33	96%
2	57.49M	0.73	-0.67	98%
3	58.99M	0.89	-0.43	98%
4	60.51M	0.86	-0.48	98%
5	64.51M	0.95	0.05	92%
6	65.49M	0.76	-0.10	58%
7	66.99M	0.95	-0.28	99%
8	68.49M	0.98	-0.08	96%
9	69.99M	0.97	-0.15	96%
10	71.49M	0.93	-0.27	93%
11	73.49M	0.97	0.19	98%
12	74.99M	0.94	-0.14	90%
13	76.49M	0.92	0.13	87%
14	77.98M	0.95	0.08	91%
15	79.47M	0.99	-0.35	99%
16	80.99M	0.99	0.31	97%
17	82.99M	0.98	0.17	99%
18	84.47M	0.96	0.22	97%
19	85.03M	0.94	0.33	99%
20	87.53M	0.78	-0.23	69%
21	89.98M	0.95	0.27	97%
22	90.48M	0.97	-0.16	97%
23	93.99M	0.94	0.33	99%
24	95.48M	0.89	0.41	97%
25	97.76M	0.98	-0.02	96%

Q12 IS THE TOTAL OBSERVED VARIANCE ACCOUNTED FOR

PLT OF 1ST VS. 2ND Q-MODE PRINCIPAL COMPONENTS OF NANNOPLANKTON FOR CENOP FILE [DSOP3178.PNT]

2ND Q-MODE PRINCIPAL COMPONENT OF NANNOPLANKTON FOR CENOP FILE [DSOP3178.PNT] PLOTTED DOWN-CURE
(500-METER DEPTHS IN METERS LISTED ON THE RIGHT)

10. CIRCLE plots depth or latitude vs. age distributions of variables. Variables are plotted as circles whose radii are proportional to the magnitude of the variables. Redrafted results from Tjalsma and Lohmann (1980) are shown.

The first result shows the bathymetric distribution of Eocene samples; the bars reflect uncertainty of the age estimates. Input is depth (meters), minimum age (m.y.) and maximum age for each sample. The second result shows the distribution of a species of benthic foram in these samples. Circles are drawn proportional to relative abundance according to a specified scale.

11. ELLIPSE projects group of samples as equal frequency ellipses on the principal components plane. Samples are input a group at a time; each sample is described by its loadings on the 1st three Q-mode principal components. An ellipse is centered at the center of gravity (joint mean) of the loadings on the samples. The orientation and eccentricity of the ellipse reflect the nature and strength of the covariation between the loadings for each sample group. A result from Lohmann and Carlson (1980) is shown. ISOTOPE Plots the oxygen and carbon isotopic compositions of groups of samples, then summarizes them and describes their covariation in a similar way.

```

<ELLIPSE> PROJECTS SAMPLE GROUPS AS ELLIPSES
>PLOT DATA POINTS(P), ELLIPSES(E), OR BOTH(B)
B
>NAME OF FILE CONTAINING PRINCIPAL COMPONENTS LOADINGS?
'DSDP71.EPL'
( 0.11274E-01  -0.12127E-01 )  LAMDA 1= 0.79475E-01  U1= 0.1751  U2= -0.9846
( -0.12127E-01   0.77319E-01 )  LAMDA 2= 0.91180E-02  V1= 0.9846  V2= 0.1751
( 0.86192E-01  -0.40175E-01 )  LAMDA 1= 0.12217E+00  U1= 0.7449  U2= -0.6672
( -0.40175E-01   0.77319E-01 )  LAMDA 2= 0.41336E-01  V1= 0.6672  V2= 0.7449
>PLOT ANOTHER SET OF SAMPLES?
Y
>NAME OF FILE CONTAINING PRINCIPAL COMPONENTS LOADINGS?
'DSDP77B.EPL'
( 0.72156E-02  -0.51004E-02 )  LAMDA 1= 0.35349E-01  U1= 0.1784  U2= -0.9840
( -0.51004E-02   0.34424E-01 )  LAMDA 2= 0.62909E-02  V1= 0.9840  V2= 0.1784
( 0.77692E-01  -0.27680E-01 )  LAMDA 1= 0.91190E-01  U1= 0.8988  U2= -0.4383
( -0.27680E-01   0.34424E-01 )  LAMDA 2= 0.20927E-01  V1= 0.4383  V2= 0.8988
>PLOT ANOTHER SET OF SAMPLES?
N
>ALTER SIZE OF PLOT? INPUT SCALE FACTOR.
.5
FORTRAN STOP

```

SAMPLES FROM INPUT FILE [DSDP71.EPL]
 PLOTTED INDIVIDUALLY AND AS AN ELLIPSE ON PRINCIPAL COMPONENTS PLANE
 SET NO. 1 CONTAINS 30 SAMPLES

DISPERSION MATRIX	0.113E-01	-0.121E-01	
	-0.121E-01	0.773E-01	
ELLIPSE CENTERED AT		0.768	0.171
LENGTHS OF PRINCIPAL AXES		0.564	0.191
DIRECTION SIN & COS OF MAJOR AXIS		0.175	-0.985

DISPERSION MATRIX	0.862E-01	-0.402E-01	
	-0.402E-01	0.773E-01	
ELLIPSE CENTERED AT		0.110	0.171
LENGTHS OF PRINCIPAL AXES		0.699	0.407
DIRECTION SIN & COS OF MAJOR AXIS		0.745	-0.667

Figure . Covariation of 1st and 2nd Q-mode principal components and 2nd and 3rd Q-mode principal components of nannoplankton data at two DSDP sites described by program ELLIPSE. + is the joint mean, orientation of the ellipse indicates the source of variation, and the shape of the ellipse (eccentricity) indicates the strength of the covariation. See Sokal and Rolf (1969), p. 526-532.

12. BARPLOT plots variables down-core. Census data, expressed as either percents or ratios, can be plotted as either point or interval estimates. Interval estimates are computed and plotted at a specified confidence level (see Mosimann, 1965). If specified confidence level is 0, then only points are plotted. If negative, the program hatches in the series about a specified midline. Points can be connected by either straight, rectangular or curved lines. Curved lines are computed as cubic splines. Up to 50 variables or cores can be plotted. The list of questions and responses and examples of plots are shown in the figure.

```
$ RUN [231231]BARPLOT
```

```
<BARPLOT> PLOTS CENSUS VARIABLE DOWN-CORE AS POINT OR INTERVAL ESTIMATES !
```

```
>TITLE? (UP TO 80 CHARACTERS)
```

```
  RIO GRANDE RISE CORE CHAIN115-88
>PLOT AGAINST AGE(A), DEPTH(D), OR BOTH(B) ?
```

```
D
```

```
>NAME OF CORE OR VARIABLE? (UP TO 20 CHARACTERS)
  PERCENT CARBONATE
```

```
>WHAT'S ITS FILENAME ? "DEPTH, AGE, CENSUS DATA"
  '88CARB.DAT'
```

```
>HOW MANY VARIABLES ASSOCIATED WITH EACH SAMPLE?
```

```
1
```

```
>PLOT WHICH OF THESE VARIABLES? ENTER ITS SUBSCRIPT.
```

```
1
```

```
>PERCENTAGE (P) OR RATIO (R) DATA ?
```

```
P
```

```
>PROBABILITY LEVEL (C) FOR CONFIDENCE INTERVALS?
```

```
  IF PROBABILITY=0, ONLY POINTS ARE PLOTTED !
```

```
  IF PROBABILITY=-1, PLOT IS HATCHED IN !
```

```
-1
```

```
MINIMUM Y-VALUE = . 0.000
```

```
MAXIMUM Y-VALUE = 6.900
```

```
Y-AXIS (DEPTH OR AGE) ORIGIN ?
```

```
0 .5 14
```

```
>INCREMENT PER INCH (IN UNITS OF M OR M.Y.) ?
```

```
.5
```

```
>MAXIMUM VALUE FOR AXIS ?
```

```
7
```

```
MINIMUM X-VALUE = 66.800
```

```
MAXIMUM X-VALUE = 92.300
```

```
X-AXIS ORIGIN, INCREMENT PER INCH, & LENGTH IN INCHES?
```

```
65 5 6
```

```
>CONNECT POINTS WITH A LINE ?
```

```
Y
```

```
>STRAIGHT (S), RECTANGULAR (R), OR CURVED (C) ?
```

```
C
```

```
>SPECIFY DEPTH OR TIME WINDOW WITHIN WHICH SAMPLES
  ARE CONSIDERED DUPLICATES (USUALLY WIDTH = 0)
```

```
.01
```

```
>"ZERO" OR CENTER ABOUT WHICH LINE IS DRAWN ?
```

```
80
```

```
>ANY MORE CORES OR VARIABLES?
```

```
Y
```



```

>ON THE SAME AXIS ? (ONLY POINTS WILL BE PLOTTED)
N
>NAME OF CORE OR VARIABLE? (UP TO 20 CHARACTERS)
RIGHT/LEFT G.TRUNC.
>WHATS ITS FILENAME ? "DEPTH, AGE, CENSUS DATA"
'88LATRUNC.BNT'
>HOW MANY VARIABLES ASSOCIATED WITH EACH SAMPLE?
4
>PLOT WHICH OF THESE VARIABLES? ENTER ITS SUBSCRIPT.
4
>PERCENTAGE (P) OR RATIO (R) DATA ?
R
>PROBABILITY LEVEL (%) FOR CONFIDENCE INTERVALS?
IF PROBABILITY=0, ONLY POINTS ARE PLOTTED !
IF PROBABILITY=-1, PLOT IS HATCHED IN !
68
>WHICH DATA COLUMN CONTAINS ABUNDANCE OF SPIKE ?
3

MINIMUM Y-VALUE = 0.006
MAXIMUM Y-VALUE = 6.900
Y-AXIS (DEPTH OR AGE) ORIGIN ?
0
>INCREMENT PER INCH (IN UNITS OF M OR M.Y.) ?
.50
>MAXIMUM VALUE FOR AXIS ?
7.0

MINIMUM X-VALUE = 0.66E-02
MAXIMUM X-VALUE = 0.13E+02
X-AXIS ORIGIN ? (ENTER AS A POWER OF 10)
WHEN AN X-VALUE .EQ. 0 OR IS .LE. ORIGIN, ITS PLOTTED AT THE ORIGIN
-1
>INCREMENT ALONG X-AXIS ? (INCHES PER POWER OF 10)
4
>LENGTH OF X-AXIS ? (IN INCHES)
8
>CONNECT POINTS WITH A LINE ?
Y
>STRAIGHT (S), RECTANGULAR (R), OR CURVED (C) ?
C
>SPECIFY DEPTH OR TIME WINDOW WITHIN WHICH SAMPLES
ARE CONSIDERED DUPLICATES (USUALLY WIDTH = 0)
.01
>ANY MORE CORES OR VARIABLES?
N
>INPUT SCALE FACTOR FOR SIZE OF PLOT ?
.5
PLOT IS SPOOLED ON FILENAME=PLOT.PLT !!
FORTRAN STOP

```

Figure . Example of BARPLOT. The ratios of left and right coiling *G. truncatulinoides* are plotted on a log scale as interval estimates at the 68% confidence level.

RIO GRANDE RISE CORE CHAIN 115-88

Listings

BARPLOT
BENTHIC
CHORUS
CIRCLE
CORREL
COVAR
CROSSPROD
DATAPLOT
DPCA
EIGEN
ELLIPSE
FLIP
HISTOGRAM
ISOTOPE
LINTERP
MULTIPLY
NANNO
OMEGA
PERCENT
QPCA
RECTERP
REPERCENT
RPCA
SLICER
SMOOTHER
SQUARE
TIMER
WEATHER


```

1  C <BARPLOT> PLOTS DOWN-CORE DATA AS POINTS OR INTERVAL ESTIMATES.
2  C
3  C THIS VERSION HAS TWO SPECIAL CAPABILITIES:
4  C
5  C
6  C 1. A DEPTH SCALE IS PLOTTED ACCORDING TO ORIGIN AND SCALING
7  C    PARAMETERS FOR A TIME AXIS. THIS ALLOWS DIFFERENT RECORDS
8  C    ALL TO BE PLOTTED ON THE SAME TIME SCALE & AND STILL SHOW
9  C    THEIR DEPTH SCALES.
10 C    LIMITATION IS THAT AGE MODEL MUST ASSUME CONSTANT SEDIMENTATION
11 C    RATE.
12 C 2. TWO DIFFERENT DATA SETS MAY BE PLOTTED ON SAME AXES. THIS ALLOWS
13 C    ORIGINAL DATA (POINTS ONLY) TO BE PLOTTED OVER INTERPOLATED DATA
14 C    THAT HAS BE CONNECTED BY A CURVED LINE (SPLINE).
15 C
16 C REQUIRES SUBROUTINES COUNTERROR, CERRORBAR, RATIOERROR, RERRORBAR,
17 C    SORTVECTOR, BOXLINE, CURVELINE, CHECKDUP, AXISINTEGER,
18 C    ASCEND, AXISAGE, AXISDEPTH, AXISDAGE, HATCHIT,
19 C    AXISLOG10, SPACER, SPLINE, POINTPLOT, SYMBOLDFT, CALSPool/LIB
20 C
21 C PLOT IS DEFERRED BY SPOOLING ON FILENAME=PLOT.PLT
22 C
23     DIMENSION X(1000),Y(1000),T(1000),Z(50),IDX(50,5),IX(5),TITLE(20)
24     DIMENSION XD(50),XL(50),XI(50),YD(50),YL(50),YI(50),PROB(50)
25     DIMENSION NS(50),IVAR(50),JVAR(50),KVAR(50),WIDTH(50),XC(50)
26     DIMENSION LANS(50),MANS(50),NANS(50)
27     DIMENSION LVAR(50),MVAR(50),KANS(50),SEDRAT(50),AGE1(50)
28     DIMENSION DEPTH1(50)
29     PARAMETER NMAX=1000,NSETS=50
30     CHARACTER FILENAME(50)*13,FNAME(50)*13
31     CHARACTER FILE*13
32     TYPE 1
33 1 FORMAT(// ' <BARPLOT> PLOTS CENSUS VARIABLE DOWN-CORE AS POINT OR '
34 + 'INTERVAL ESTIMATES !'/'/' >TITLE? (UP TO 80 CHARACTERS)'/)
35     ACCEPT 2,TITLE
36 2 FORMAT(20A4)
37     TYPE *, '>PLOT AGAINST AGE(A), DEPTH(D), OR BOTH(B) ?'
38     ACCEPT 3, IANS
39 3 FORMAT(A1)
40     DO 16 NIN=1,NSETS
41     TYPE *, '>NAME OF CORE OR VARIABLE? (UP TO 20 CHARACTERS)'
42     ACCEPT 2, (IDX(NIN,I),I=1,5)
43     TYPE *, '>WHATS ITS FILENAME ? "DEPTH, AGE, CENSUS DATA"'
44     ACCEPT *,FILENAME(NIN)
45     OPEN(UNIT=1,NAME=FILENAME(NIN),TYPE='OLD')
46     TYPE *, '>HOW MANY VARIABLES ASSOCIATED WITH EACH SAMPLE?'
47     ACCEPT *,IVAR(NIN)
48     TYPE *, '>PLOT WHICH OF THESE VARIABLES? ENTER ITS SUBSCRIPT.'
49     ACCEPT *,JVAR(NIN)
50     TYPE *, '>PERCENTAGE (P) OR RATIO (R) DATA ?'
51     ACCEPT 3,LANS(NIN)
52     TYPE *, '>PROBABILITY LEVEL (%) FOR CONFIDENCE INTERVALS?'
53     TYPE *, ' IF PROBABILITY=0, ONLY POINTS ARE PLOTTED !'
54     TYPE *, ' IF PROBABILITY=-1, PLOT IS HATCHED IN !'
55     ACCEPT *,PROB(NIN)
56     IF(PROB(NIN).LE.0) GO TO 4
57     IF(LANS(NIN).EQ.1HR) THEN
58         TYPE *, '>WHICH DATA COLUMN CONTAINS ABUNDANCE OF SPIKE ?'
59     ELSE IF(LANS(NIN).EQ.1HP) THEN
60         TYPE *, '>WHICH DATA COLUMN CONTAINS TOTAL NUMBER COUNTED ?'

```



```

62      ACCEPT *,KVAR(NIN)
63      4 DO 5 J=1,NMAX
64      READ(1,*,END=5) D,A,{Z(I),I=1,IVAR(NIN)}
65      IF(IANS.EQ.1HJ) Y(J)=D
66      IF(IANS.EQ.1HA.OR.IANS.EQ.1HB) Y(J)=A
67      T(J)=Z(KVAR(NIN))
68      5 X(J)=Z(JVAR(NIN))
69      6 NS(NIN)=J-1
70      IF(IANS.EQ.1HJ) THEN
71      TYPE *, '>FILE WITH 1ST DEPTH, 1ST AGE, SED RATE ?'
72      ACCEPT *,FILE
73      OPEN(UNIT=2,NAME=FILE,TYPE='OLD')
74      READ(2,*) DEPTH1(NIN),AGE1(NIN),SEDRA1(NIN)
75      CLOSE(UNIT=2)
76      END IF
77      TYPE 7,Y(1),Y(NS(NIN))
78      7 FORMAT(/' MINIMUM Y-VALUE ='F10.3/' MAXIMUM Y-VALUE ='F10.3/
79      + '>Y-AXIS (DEPTH OR AGE) ORIGIN ?')
80      ACCEPT *,Y0(NIN)
81      TYPE *, '>INCREMENT PER INCH (IN UNITS OF M OR M.Y.) ?'
82      ACCEPT *,YI(NIN)
83      TYPE *, '>MAXIMUM VALUE FOR AXIS ?'
84      ACCEPT *,YL(NIN)
85      IF(IANS.NE.1HJ) YL(NIN)=FLOAT(NINT((YL(NIN)-Y0(NIN))/YI(NIN)))
86      K=1
87      8 IF(X(K).EQ.0.AND.K.LE.NS(NIN)) THEN
88      K=K+1
89      GO TO 8
90      END IF
91      XMIN=X(K)
92      XMAX=X(1)
93      DO 10 I=1,NS(NIN)
94      IF(LANS(NIN).EQ.1HR.AND.X(I).EQ.0) GO TO 9
95      IF(X(I).LT.XMIN) THEN
96      XMIN=X(I)
97      TXMIN=T(I)
98      END IF
99      9 IF(X(I).GT.XMAX) THEN
100     XMAX=X(I)
101     TXMAX=T(I)
102     END IF
103 10 CONTINUE
104     IF(LANS(NIN).EQ.1HP) THEN
105     IF(PROB(NIN).LE.0) GO TO 11
106     CALL COUNTERROR(TXMIN,XMIN,PROB(NIN),DUMMY,XMIN)
107     CALL COUNTERROR(TXMAX,XMAX,PROB(NIN),XMAX,DUMMY)
108 11     TYPE 12,XMIN,XMAX
109 12     FORMAT(/' MINIMUM X-VALUE ='F10.3/' MAXIMUM X-VALUE ='F10.3/
110     + '>X-AXIS ORIGIN, INCREMENT PER INCH, & LENGTH IN INCHES?')
111     ACCEPT *,X0(NIN),XI(NIN),XL(NIN)
112     END IF
113     IF(LANS(NIN).EQ.1HR) THEN
114     IF(PROB(NIN).LE.0) GO TO 13
115     IF(TXMIN.EQ.0) TXMIN=1
116     CALL RATIOERROR(TXMIN,XMIN,PROB(NIN),DUMMY,XMIN)
117     CALL RATIOERROR(TXMAX,XMAX,PROB(NIN),XMAX,DUMMY)
118 13     TYPE 14,XMIN,XMAX
119 14     FORMAT(/' MINIMUM X-VALUE ='E10.2/' MAXIMUM X-VALUE ='E10.2/
120     + '>X-AXIS ORIGIN ? (ENTER AS A POWER OF 10)'/' WHEN AN X-VALUE '
121     + '.EQ. 0 OR IS .LE. ORIGIN, ITS PLOTTED AT THE ORIGIN')

```



```

123     TYPE *, '>INCREMENT ALONG X-AXIS ? (INCHES PER POWER OF 10)'
124     ACCEPT *, XI(NIN)
125     TYPE *, '>LENGTH OF X-AXIS ? (IN INCHES)'
126     ACCEPT *, XL(NIN)
127     END IF
128     CLOSE(UNIT=1)
129     TYPE *, '>CONNECT POINTS WITH A LINE ?'
130     ACCEPT 3, MANS(NIN)
131     IF (MANS(NIN).EQ.14Y) THEN
132         TYPE *, '>STRAIGHT (S), RECTANGULAR (R), OR CURVED (C) ?'
133         ACCEPT 3, NANS(NIN)
134         IF (NANS(NIN).EQ.14C) THEN
135             TYPE *, '>SPECIFY DEPTH OR TIME WINDOW WITHIN WHICH SAMPLES '
136             TYPE *, 'ARE CONSIDERED DUPLICATES (USUALLY WIDTH = 0)'
137             ACCEPT *, WIDTH(NIN)
138             END IF
139             IF (NANS(NIN).EQ.14R.OR.NANS(NIN).EQ.14C.AND.PROB(NIN).LT.0) THEN
140                 TYPE *, '>"ZERO" OR CENTER ABOUT WHICH LINE IS DRAWN ?'
141                 ACCEPT *, XC(NIN)
142                 END IF
143             END IF
144     19 TYPE *, '>ANY MORE CORES OR VARIABLES?'
145     ACCEPT 3, JANS
146     IF (JANS.EQ.14N) THEN
147         GO TO 17
148     ELSE
149         TYPE *, '>ON THE SAME AXIS ? (ONLY POINTS WILL BE PLOTTED)'
150         ACCEPT 3, JANS
151         IF (JANS.EQ.14N) GO TO 16
152         KANS(NIN)=JANS
153         IF (KANS(NIN).EQ.14Y) THEN
154             TYPE *, '>FILENAME ?'
155             ACCEPT *, FNAME(NIN)
156             TYPE *, '>HOW MANY VARIABLES ASSOCIATED WITH EACH SAMPLE ?'
157             ACCEPT *, LVAR(NIN)
158             TYPE *, '>PLOT WHICH VARIABLE ?'
159             ACCEPT *, VVAR(NIN)
160             GO TO 19
161             END IF
162         END IF
163     16 CONTINUE
164     17 TYPE *, '>INPUT SCALE FACTOR FOR SIZE OF PLOT ?'
165     ACCEPT *, SIZE
166     C PLOT
167     CALL ASSIGN(9, 'PLOT.PLT')
168     CALL PLOTS(0,0,9)
169     IF (SIZE.GT.0.) CALL FACTOR(SIZE)
170     IF (IANS.EQ.143) THEN
171         YLM=FLOAT(NINT((YL(1)-Y0(1))/YI(1)))
172     ELSE
173         YLM=YL(1)
174         DO 18 I=1,NIN
175     18 IF (YL(I).GT.YLM) YLM=YL(I)
176         YL(1)=YLM
177     END IF
178     CALL PLOT(0.,YLM,-3)
179     CALL SYMBOL(0.,2.,.49,TITLE,0.,80)
180     IF (IANS.EQ.14A) THEN
181         IF (YI(1).GE..1) THEN
182             CALL SYMBOL(-.5,-(YLM/2)-2.1,.42,'AGE (M.Y.)'.90..10)

```

```

184      CALL SYMBOL(-.5,-(YLM/2)-2.1,.42,'AGE (K.Y.)',90.,10)
185      END IF
186  ELSE
187  IF(IANS.EQ.1HD.OR.IANS.EQ.1HB) THEN
188      DI=YI(1)
189      IF(IANS.EQ.1HB) DI=DI*SEDRAT(1)
190      IF(DI.GE.1) THEN
191          CALL SYMBOL(-.5,-(YLM/2)-2.94,.42,'DEPTH (METERS)',90.,14)
192      ELSE
193          CALL SYMBOL(-.5,-(YLM/2)-3.99,.42,'DEPTH (CENTIMETERS)',90.,19)
194      END IF
195  END IF
196  END IF
197  DO 27 J=1,NIN
198      IF(IANS.EQ.1HA) CALL AXISAGE(YO(J),YI(J),YL(J))
199      IF(IANS.EQ.1HD) CALL AXISDEPTH(YO(J),YI(J),YL(J))
200      IF(IANS.EQ.1HB) THEN
201          CALL AXISDAGE(YO(J),YL(J),YI(J),AGE1(J),DEPTH1(J),SEDRAT(J))
202      END IF
203      DO 22 I=1,5
204  22 IX(I)=IDX(J,I)
205      IF(IANS(J).EQ.1HP) CALL AXISINTEGER(0.,.25,IX,XL(J),XD(J),XI(J))
206      IF(IANS(J).EQ.1HR) CALL AXISLOG10(0.,.25,IX,XL(J),XD(J),XI(J))
207      OPEN(UNIT=1,NAME=FILENAME(J),TYPE='OL')
208      DO 23 I=1,NS(J)
209      READ(1,*) D,A,(Z(K),K=1,IVAR(J))
210      IF(IANS.EQ.1HD) Y(I)=D
211      IF(IANS.EQ.1HA.OR.IANS.EQ.1HB) Y(I)=A
212      T(I)=Z(KVAR(J))
213  23 X(I)=Z(JVAR(J))
214      CALL SORTVECTOR(Y,X,NS(J))
215      IF(PROB(J).LE.0) GO TO 24
216      IF(IANS(J).EQ.1HP) THEN
217          CALL CERRURBAR(Y,YO(J),YI(J),X,XD(J),XI(J),NS(J),T,PROB(J),XC(J))
218      ELSE
219          CALL KERRURBAR(Y,YO(J),YI(J),X,XD(J),XI(J),NS(J),T,PROB(J),XC(J))
220      END IF
221  24 IF(IANS(J).EQ.1HN) GO TO 26
222      IF(IANS(J).EQ.1HR) THEN
223          XI(J)=1/XI(J)
224          DO 25 K=1,NS(J)
225          IF(X(K).LE.(10.**XD(J)).OR.X(K).EQ.0) THEN
226              X(K)=XD(J)
227              GO TO 25
228          ELSE
229              X(K)=ALOG10(X(K))
230          END IF
231  25 CONTINUE
232      END IF
233      IF(PROB(J).LT.0) THEN
234          CALL HATCHIT(Y,YO(J),YI(J),X,XD(J),XI(J),NS(J),XC(J))
235      END IF
236      IF(IANS(J).EQ.1HS) THEN
237          CALL CHECKDUP(0.,Y,X,NS(J))
238          X(NS(J)+1)=XD(J)
239          X(NS(J)+2)=XI(J)
240          Y(NS(J)+1)=YO(J)
241          Y(NS(J)+2)=-YI(J)
242          CALL LINE(X,Y,NS(J),1,0,1)
243      END IF
244  26

```



```

245     IF (NANS(J).EQ.1HR) THEN
246         IF (PROB(J).GT.0) THEN
247             CALL BOXLINE(Y,YO(J),YI(J),X,XO(J),XI(J),XC(J),NS(J))
248         ELSE
249             CALL RECTANG(Y,YO(J),YI(J),X,XO(J),XI(J),XC(J),NS(J))
250         END IF
251     END IF
252     IF (NANS(J).EQ.1HC) THEN
253         CALL CURVELINE(Y,YO(J),YI(J),X,XJ(J),XI(J),NS(J),WIDTH(J))
254     END IF
255 26  CLOSE(UNIT=1)
256     IF (KANS(J).EQ.1HY) THEN
257         OPEN(UNIT=1,NAME=FNAME(J),TYPE='OLD')
258         I=0
259 28  READ(1,*,END=29) D,A,(Z(K),K=1,LVAR(J))
260         I=I+1
261         IF (IANS.EQ.1HD) Y(I)=D
262         IF (IANS.EQ.1HA.OR.IANS.EQ.1HB) Y(I)=A
263         X(I)=Z(MVAR(J))
264         IF (LANS(J).EQ.1HR) THEN
265             IF (X(I).LE.(10.**XO(J)).OR.X(I).EQ.0) THEN
266                 X(I)=XO(J)
267             ELSE
268                 X(I)=ALOG10(X(I))
269             END IF
270         END IF
271         GO TO 28
272 29  CALL POINTPLOT(Y,YO(J),YI(J),X,XO(J),XI(J),I)
273         CLOSE(UNIT=1)
274     END IF
275 27  CALL PLOT(XL(J)+.5,0.,-3)
276     CALL PLOT(0.,0.,999)
277     TYPE *, 'PLOT IS SPOOLED ON FILENAME=PLUT.PLT !!!'
278     STOP
279     END

```

```

1  C <BENTHIC> CREATES QUATERNARY BENTHIC FORAM CENSUS DATA FILES,
2  C      COMPUTES A Q-MODE P.C.A, AND CAN ESTIMATE PRINCIPAL
3  C      COMPONENTS IN OTHER SAMPLES IF DESIRED
4  C
5  C      REQUIRES SUBROUTINES PERCENT, NORMALIZE, ASCEND, QMATRIX, HOW,
6  C      QSCORES, PRINCO, XYPLT, TSPLT
7  C
8      DIMENSION X(100,36),Q(100,100),DEPTH(100),TOTAL(100),Z(36)
9      DIMENSION E(100),V(100,100),A(100),B(100),C(100),D(100)
10     DIMENSION F(32,100),Y(32),IX(36),IY(32),IZ(32),INDX(100),IW(32)
11     CHARACTER FILENAME*13,TODAY*9
12     PARAMETER NDIM=100
13     NS=0
14     TYPE 1
15     1 FORMAT(///' THIS IS <BENTHIC> !'///' <BENTHIC> CREATES OR '
16     + 'UPDATES BENTHIC CENSUS DATA FILES'//11X'COMPUTES A Q-MODE '
17     + 'PRINCIPAL COMPONENTS ANALYSIS'//11X'AND CAN ESTIMATE PRINCIPAL '
18     + 'COMPONENTS IN OTHER SAMPLES'///'>BEFORE YOU '
19     + 'INPUT NEW DATA, IS THERE ALREADY DATA ON FILE FOR '
20     + 'THIS AREA OR CORE?'///' ANSWER Y OR N')
21     ACCEPT 2,IANS
22     2 FORMAT(A1)
23     IF(IANS.EQ.1HY) THEN
24         TYPE *,>'WHATS ITS FILENAME?'
25         ACCEPT *,FILENAME
26         OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
27         DO 3 N=1,NDIM
28             3 READ(1,*,END=4) DEPTH(N),TOTAL(N),(X(N,I),I=1,36)
29             4 NS=N-1
30             TYPE *,>'WILL YOU BE ADDING ANY NEW DATA TO THIS FILE?'
31             ACCEPT 2,KANS
32             IF(KANS.EQ.1HN) THEN
33                 MS=NS
34                 GO TO 19
35             END IF
36             ELSE
37                 TYPE 5
38                 5 FORMAT(' >THEN GIVE YOUR NEW DATA FILE A NAME (UP TO 9 '
39                 + '"CHARACTER.PNT")')
40                 ACCEPT *,FILENAME
41                 N=1
42                 END IF
43     C INPUT CENSUS DATA
44     TYPE 6
45     6 FORMAT(/' >TYPE IN YOUR NEW BENTHIC CENSUS DATA:'//2X,>'FIRST, '
46     + ' LABEL EACH SAMPLE WITH ITS DEPTH (IN METERS)'//3X,>'THEN, '
47     + ' TYPE IN COUNTS FOR THE 36 BENTHIC TAXA, ALL COUNTS SEPARATED '
48     + 'BY SPACES'///)
49     DO 10 M=N,NDIM
50     7 ACCEPT *, DEPTH(M),Z
51     TYPE *,>'CHECK THE CENSUS DATA FOR THIS SAMPLE. IS IT CORRECT?'
52     ACCEPT 2, JANS
53     IF(JANS.EQ.1HN) GO TO 7
54     DO 8 I=1,32
55     8 Y(I)=Z(I+4)
56     CALL PERCENT(Y,32,T)
57     X(M,1)=Z(1)/T
58     X(M,2)=Z(2)/T
59     X(M,3)=Z(3)/T
60     X(M,4)=Z(4)/T

```



```

52      DO 9 I=1,32
53      9 X(M,I+4)=Y(I)
54      TYPE *, '>ANY MORE SAMPLES FOR THIS DATA FILE?'
55      ACCEPT 2,JANS
56      IF(JANS.EQ.1HN) GO TO 11
57      10 TYPE *, '>THEN INPUT THE DEPTH AND COUNTS FOR YOUR NEXT SAMPLE'
58      11 MS=M
59      TYPE 15, NS,MS-NS
70      15 FORMAT(///110,' OLD SAMPLES;',I10,' NEW SAMPLES'///)
71      CLOSE(UNIT=1)
72      CALL DATE(TODAY)
73      C SORT SAMPLES BY DEPTH, THEN OUTPUT TO DISK FILE
74      19 CALL ASCEND(DEPTH,INDX,MS)
75      DO 16 I=1,MS
76      A(I)=DEPTH(INDX(I))
77      B(I)=TOTAL(INDX(I))
78      DO 16 J=1,36
79      16 Q(I,J)=X(INDX(I),J)
80      DO 17 I=1,MS
81      DEPTH(I)=A(I)
82      TOTAL(I)=B(I)
83      DO 17 J=1,36
84      17 X(I,J)=Q(I,J)
85      TYPE *, 'SAMPLES HAVE BEEN RESORTED BY THEIR DEPTHS !'
86      OPEN(UNIT=2,NAME=FILENAME)
87      DO 18 I=1,MS
88      18 WRITE(2,*) DEPTH(I),TOTAL(I),(X(I,J),J=1,36)
89      TYPE *, 'CENSUS DATA HAS BEEN OUTPUT TO A DISC FILE !'
90      C PRINT CENSUS DATA EXPRESSED AS PERCENT
91      OPEN(UNIT=3,NAME='PRINT.TMP',DISPOSE='PRINT')
92      WRITE(3,20) FILENAME
93      20 FORMAT(1H1//6X,'CENSUS DATA FILE OF QUATERNARY BENTHIC FORAMIN'
94      + ' IFERA ON FILE ['',A13,'']'///)
95      IF(KANS.EQ.1HN) GO TO 23
96      IF(IANS.EQ.1HY) WRITE(3,21) TODAY,FILENAME,NS,MS-NS
97      IF(IANS.EQ.1HN) WRITE(3,22) TODAY,FILENAME,MS-NS
98      21 FORMAT(6X,'UPDATED ',A9,' AND STORED UNDER FILENAME ['',A13,'']'//
99      + I10,' SAMPLES IN PREVIOUS DATA FILE;',I5,' NEW SAMPLES ADDED'///)
100     22 FORMAT(6X,'CREATED ',A9,' AND STORED UNDER FILENAME ['',A13,'']'//
101     + 6X,'THIS NEW DATA FILE CREATED WITH',I10,' SAMPLES'///)
102     23 WRITE(3,24) FILENAME,(I,I=1,36)
103     24 FORMAT(6X,'QUATERNARY BENTHIC FORAM CENSUS DATA (EXPRESSED AS '
104     + 'PERCENT) FOR ['',A13,'']'///4X'NO. DEPTH TOTAL '36I3/)
105     DO 26 K=1,MS
106     IT=NINT(TOTAL(K))
107     DO 25 L=1,36
108     25 IX(L)=NINT(X(K,L))
109     26 WRITE(3,27) K,DEPTH(K),IT,IX
110     27 FORMAT(2X,14,F9.2,15,2X,36I3)
111     TYPE *, 'CENSUS DATA HAS BEEN PRINTED !'
112     C COMPUTE Q-MATRIX, EIGENVALUES, EIGENVECTORS, & Q-SCORES
113     DO 28 I=1,MS
114     DO 28 J=1,32
115     28 X(I,J)=X(I,J+4)
116     CALL QMATRIX(X,Q,MS,32,NDIM,36)
117     TYPE *, 'Q-MATRIX HAS BEEN COMPUTED !'
118     CALL HOW(MS,NDIM,3,Q,E,V,A,B,C,D)
119     TYPE *, 'EIGENVALUES & EIGENVECTORS HAVE BEEN COMPUTED !'
120     CALL QSCORES(X,V,E,MS,32,NDIM,32,F)
121     TYPE *, 'Q-SCORES HAVE BEEN COMPUTED !'
122     DO 30 I=1,32

```



```

123      Y(I)=F(I,1)
124      30 IX(I)=NINT(Y(I)*100.)
125      CALL PERCENT(Y,32,T)
126      DO 31 I=1,32
127      IY(I)=NINT(Y(I))
128      IW(I)=NINT(F(I,2)*100.)
129      31 IZ(I)=NINT(F(I,3)*100.)
130      E1=(E(1)/MS)*100.
131      E2=(E(2)/MS)*100.
132      E3=(E(3)/MS)*100.
133      TYPE *, 'RESULTS ARE NOW BEING PRINTED !'
134      WRITE(3,32) FILENAME,(1,I=5,36),(IX(I),I=1,32),IY,IW,IZ,E1,E2,E3
135      32 FORMAT(1H1//6X, 'BENTHIC FORAMINIFERAL COMPOSITIONS (SCORES) FOR '
136      + ' THE 1ST 3 Q-MODE EIGENVECTORS FOR DATA FILE [' ,A13,']'////3X,
137      + ' 32I4//2X, '1',32I4//4X,32(I3,'%')//2X, '2',32I4//2X, '3',32I4////
138      + ' 6X, '(NOTE THAT ALL COMPOSITIONS GIVEN IN NORMALIZED FORM HAVE '
139      + ' BEEN MULTIPLIED BY 100)'////6X, 'THE 1ST Q-MODE EIGENVECTOR '
140      + ' REPRESENTS THE "AVERAGE" SAMPLE FOR THIS DATA;'//6X 'IT ALONE '
141      + ' DESCRIBES ',F5.1,'% OF THE OBSERVATIONS.'////6X, 'THE 2ND EIGEN'
142      + ' VECTOR DESCRIBES THE PRINCIPAL DIMENSION OF DEVIATIONS IN '
143      + ' BENTHIC FORAMINIFERAL COMPOSITIONS FROM THIS "AVERAGE";'//6X,
144      + ' IT ACCOUNTS FOR AN ADDITIONAL ',F5.1,'% '////6X, 'THE 3RD '
145      + ' EIGENVECTOR ACCOUNTS FOR ',F5.1,'%')
146      C SCALE EIGENVECTORS BY SQRT(EIGENVALUES) TO GET PRINCIPAL COMPONENTS
147      CALL PRINCO(V,E,MS,NDIM)
148      WRITE(3,33) FILENAME
149      33 FORMAT(1H1//6X, '1ST 3 Q-MODE PRINCIPAL COMPONENTS OF QUATERNARY '
150      + ' BENTHIC FORAMS FOR DATA FILE [' ,A13,']'////11X, 'DEPTH',7X, '1ST'
151      + ',5X, '2ND',5X, '3RD',3X, '% EXPLAINED'//)
152      DO 34 I=1,MS
153      ICOMM=NINT((V(I,1)**2+V(I,2)**2+V(I,3)**2)*100.)
154      34 WRITE(3,35) I,DEPTH(I),V(I,1),V(I,2),V(I,3),ICOMM
155      35 FORMAT(I5,3X,F8.2, 'M',F9.2,2F8.2,I10, '%')
156      TYPE *, 'PRINCIPAL COMPONENTS HAVE BEEN COMPUTED !'
157      C PLOT RESULTS
158      TYPE *, '>PLOT RESULTS?'
159      ACCEPT 2, IANS
160      IF(IANS.EQ.1HN) GO TO 47
161      WRITE(3,40) FILENAME
162      40 FORMAT(1H1//6X, 'PLOT OF 1ST VS. 2ND Q-MODE PRINCIPAL COMPONENTS '
163      + ' OF BENTHIC FORAMS FOR DATA FILE [' ,A13,']'////)
164      DO 41 I=1,MS
165      A(I)=V(I,1)
166      41 B(I)=V(I,2)
167      CALL XYPLJT(B,A,MS,3)
168      WRITE(3,42) FILENAME
169      42 FORMAT(/39X, '2ND Q-MODE PRINCIPAL COMPONENT'//1H1//6X, ' 2ND Q-MODE'
170      + ' PRINCIPAL COMPONENT OF BENTHIC FORAMS FOR DATA FILE [' ,A13,']'
171      + ' PLOTTED DOWN-DEPTH'//7X, '(DEPTHS IN METERS LISTED ON'
172      + ' THE RIGHT)'//39X, '2ND Q-MODE PRINCIPAL COMPONENT')
173      CALL TSPLJT(DEPTH,B,MS,2,3)
174      WRITE(3,43) FILENAME
175      43 FORMAT(1H1//6X, 'PLOT OF 2ND VS. 3RD Q-MODE PRINCIPAL COMPONENTS '
176      + ' OF BENTHIC FORAMS FOR DATA FILE [' ,A13,']'////)
177      DO 44 I=1,MS
178      A(I)=V(I,2)
179      44 B(I)=V(I,3)
180      CALL XYPLJT(B,A,MS,3)
181      WRITE(3,45) FILENAME
182      45 FORMAT(/39X, '3RD Q-MODE PRINCIPAL COMPONENT'//1H1//6X, ' 3RD Q-MODE'

```



```

184 + ' PLOTTED DOWN-DEPTH'//7X'(DEPTHS IN METERS LISTED ON THE RIGHT)'
185 + '///39X','3RD Q-MODE PRINCIPAL COMPONENT')
186 CALL TSPLLOT(DEPTH,8,MS,2,3)
187 TYPE *, 'RESULTS HAVE BEEN PLOTTED !'
188 47 TYPE *, '>ESTIMATE PRINCIPAL COMPONENTS IN OTHER SAMPLES?'
189 ACCEPT 2, IANS
190 IF(IANS.EQ.1H4) STOP
191 48 TYPE *, '>FILENAME FOR SET OF SAMPLE VECTORS?'
192 ACCEPT *, FILENAME
193 WRITE(3,49) FILENAME
194 49 FORMAT(1H1//6X,'1ST THREE Q-MODE PRINCIPAL COMPONENTS ESTIMATED '
195 + 'IN SAMPLES FROM INPUT FILE ['A13,']'//)
196 OPEN(UNIT=4, NAME=FILENAME, TYPE='OLD')
197 TYPE *, '>GIVE ESTIMATED LOADINGS A FILENAME.'
198 ACCEPT *, FILENAME
199 OPEN(UNIT=5, NAME=FILENAME)
200 WRITE(3,50) FILENAME
201 50 FORMAT(6X,'ESTIMATED PRINCIPAL COMPONENTS LOADINGS STORED UNDER '
202 + 'FILENAME ['A13,']'//6X'STORED AS "DEPTH(METERS), TOTAL COUNTED, '
203 + 'LOADINGS FOR PRINCIPAL COMPONENTS 1,2,3"'//11X'DEPTH'7X'1ST'5X
204 + '2ND'5X'3RD'3X'% EXPLAINED'//)
205 ISUM=0
206 NS=0
207 51 READ(4,*,END=55) DPTH,TOTL,Z
208 NS=NS+1
209 DO 54 I=1,32
210 54 Y(I)=Z(I+4)
211 CALL NORMALIZE(Y,32,T)
212 DO 52 J=1,3
213 E(J)=0.
214 DO 52 I=1,32
215 52 E(J)=E(J)+Y(I)*F(I,J)
216 A(NS)=DPTH
217 B(NS)=E(2)
218 C(NS)=E(3)
219 ICUMM=NINT((E(1)**2+E(2)**2+E(3)**2)*100.)
220 ISUM=ISUM+ICUMM
221 WRITE(3,53) NS,DPTH,E(1),E(2),E(3),ICUMM
222 53 FURMAT(15,3X,F8.2,'M',F9.2,2F8.2,110,'%')
223 WRITE(5,57) DPTH,TOTL,E(1),E(2),E(3)
224 57 FURMAT(2F10.3,3F10.5)
225 GO TO 51
226 55 WRITE(3,56) ISUM/NS
227 56 FURMAT(//5X,15,'% OF THE TOTAL OBSERVED VARIANCE ACCOJNTED FOR')
228 CLOSE(UNIT=4)
229 CLOSE(UNIT=5)
230 C PLOT ESTIMATED PRINCIPAL COMPONENTS
231 WRITE(3,60) FILENAME
232 60 FURMAT(1H1//6X,'PLOT OF ESTIMATED 2ND VS.3RD Q-MODE PRINCIPAL '
233 + 'COMPONENTS STORED IN ['A13,']'//)
234 CALL XYPLLOT(C,B,NS,3)
235 WRITE(3,61) FILENAME
236 61 FURMAT(39X'2ND Q-MODE PRINCIPAL COMPONENT'//1H1//6X,' ESTIMATED '
237 + '2ND Q-MODE PRINCIPAL COMPONENT STORED IN ['A13,'] PLOTTED DOWN-'
238 + 'DEPTH'//7X'(DEPTHS IN METERS LISTED ON THE RIGHT)'//39X
239 + '2ND Q-MODE PRINCIPAL COMPONENT')
240 CALL TSPLLOT(A,B,NS,2,3)
241 TYPE *, '>ESTIMATE PRINCIPAL COMPONENTS IN MORE SAMPLES?'
242 ACCEPT 2, IANS
243 IF(IANS.EQ.1H4) GO TO 48
244 TYPE *, 'FINISHED !'

```

245
246

STOP
END


```

1  C <CHORUS> COMPUTES THE SIMILARITY MATRIX FOR CORRESPONDENCE ANALYSIS
2  C
3  C AFTER A PROGRAM PROBABLY WRITTEN BY J.E.KLOVAN WHILE HE WAS IN UPPSALA
4  C
5      DIMENSION RSUM(1000),CSUM(50),XM(50),S(50,50)
6      CHARACTER FILENAME*13,FMT*80
7      TYPE 1
8      1 FORMAT(' THIS HERE IS <CHORUS>, A PROGRAM OF QUESTIONABLE ORIGIN!'
9      + '//' BJORN SAYS KLOVAN WROTE IT WHILE HE WAS IN UPPSALA. IT '
10     + 'SHOULD'//' COMPUTE A SIMILARITY MATRIX FOR CORRESPONDANCE'
11     + ' ANALYSIS.'//' >DATA FORMAT ? NO SAMPLE I.D. ')
12     ACCEPT *,FMT
13     TYPE *, '>NUMBER OF VARIABLES ?'
14     ACCEPT *,NCOL
15     TYPE *, '>DATA FILENAME ?'
16     ACCEPT *,FILENAME
17     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
18     OPEN(UNIT=2,NAME='CHORUS.TMP')
19     I=0
20     2 READ(1,FMT,END=4) (XM(J),J=1,NCOL)
21     I=I+1
22     RSUM(I)=0
23     DO 3 J=1,NCOL
24         CSUM(J)=CSUM(J)+XM(J)
25         TSUM=TSUM+XM(J)
26     3 RSUM(I)=RSUM(I)+XM(J)
27     GO TO 2
28     4 CLOSE(UNIT=1)
29     NR0W=1
30     TYPE 5,NR0W
31     5 FORMAT('6 SAMPLES COUNTED')
32     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
33     TRACE=-1.
34     DO 6 I=1,NCOL
35     DO 6 J=1,NCOL
36     6 S(I,J)=0.
37     DO 9 I=1,NR0W
38     READ(1,FMT) (XM(J),J=1,NCOL)
39     DO 7 J=1,NCOL
40     IF((RSUM(I)*CSUM(J)).LE.0) TYPE *,I,J
41     7 XM(J)=XM(J)/SQRT(RSUM(I)*CSUM(J))
42     DO 8 K=1,NCOL
43     DO 8 L=K,NCOL
44     8 S(K,L)=(XM(K)*XM(L))+S(K,L)
45     9 CONTINUE
46     DO 11 K=1,NCOL
47     DO 10 L=K,NCOL
48     10 S(L,K)=S(K,L)
49     TRACE=TRACE+S(K,K)
50     11 WRITE(2,*) (S(K,L),L=K,NCOL)
51     TYPE 12,TRACE
52     12 FORMAT(' TRACE = 'F13.6)
53     TRACE=TRACE+TSUM
54     TYPE 13,TRACE
55     13 FORMAT(' MODIFIED TRACE = 'F13.6)
56     STOP
57     END

```

```

1  C <CIRCLE> PLOTS DEPTH OR LATITUDE VS. AGE DISTRIBUTIONS AS CIRCLES
2  C
3  C PLOTS EITHER SAMPLES ONLY (MIN TO MAX AGE AS INTERVALS) OR
4  C PLOTS ASSOCIATED DATA VALUE AS CIRCLES WHOSE RADII ARE PROPORTIONAL
5  C TO THE MAGNITUDE OF THE DATA
6  C
7  C REQUIRES SUBROUTINES OUT, SYMBOLDEF, CALSPOOL/LIBRARY
8  C
9  C PLOT IS DEFERRED BY SPOOLING ON FILE=PLOT.PLT
10 C
11     DIMENSION Z(1000),TITLE(20)
12     CHARACTER FILENAME*13
13     TYPE *, 'THIS IS <CIRCLE>'
14     TYPE *, 'IT PLOTS DEPTH(M) OR LATITUDE VS. AGE(M.Y.) DISTRIBUTIONS'
15     TYPE *, '>PLOT SAMPLE DISTRIBUTION ONLY (S)'
16     TYPE *, ' OR PLOT ASSOCIATED DATA AS CIRCLES (C) ??'
17     ACCEPT 1, ITYPE
18 1   FORMAT(A1)
19     TYPE *, '>TITLE ?'
20     ACCEPT 9, TITLE
21 9   FORMAT(20A4)
22     TYPE *, '>PLOT VS. DEPTH (D) OR LATITUDE (L)??'
23     ACCEPT 1, JTYPE
24     IF (ITYPE.EQ.140) GO TO 5
25     TYPE *, '>FILENAME FOR INPUT?'
26     TYPE *, '"DEPTH(M) OR LATITUDE(-S,+N), MIN.AGE(M.Y.), MAX.AGE"'
27     ACCEPT *, FILENAME
28     OPEN(UNIT=1, NAME=FILENAME, TYPE='OLD')
29     NS=1
30 2   READ(1,*,END=3) D,TMIN,TMAX
31     IF(NS.EQ.1) THEN
32         DMIN=D
33         DMAX=D
34         TMN=TMIN
35         TMX=TMAX
36     ELSE
37         IF(D.LT.DMIN) DMIN=D
38         IF(D.GT.DMAX) DMAX=D
39         IF(TMIN.LT.TMN) TMN=TMIN
40         IF(TMAX.GT.TMX) TMX=TMAX
41     END IF
42     NS=NS+1
43     GO TO 2
44 3   NS=NS-1
45     GO TO 10
46 5   TYPE *, '>FILENAME FOR INPUT?'
47     TYPE *, '"DEPTH(M) OR LATITUDE(-S,+N), AGE(M.Y.), DATA VALUES"'
48     ACCEPT *, FILENAME
49     OPEN(UNIT=1, NAME=FILENAME, TYPE='OLD')
50     TYPE *, '>HOW MANY DATA VALUES ASSOCIATED WITH EACH SAMPLE?'
51     ACCEPT *, IVAR
52     TYPE *, '>PLOT WHICH OF THESE? ENTER ITS SUBSCRIPT.'
53     ACCEPT *, JVAR
54     NS=1
55 6   READ(1,*,END=7) D,AGE,(Z(K),K=1,IVAR)
56     IF(NS.EQ.1) THEN
57         DMIN=D
58         DMAX=D
59         TMN=AGE
60         TMX=AGE
61         ZMIN=Z(JVAR)

```



```

52     ZMAX=Z(JVAR)
53     ELSE
54     IF(D.LT.DMIN) DMIN=D
55     IF(D.GT.DMAX) DMAX=D
56     IF(AGE.LT.TMN) TMN=AGE
57     IF(AGE.GT.TMX) TMX=AGE
58     IF(Z(JVAR).LT.ZMIN) ZMIN=Z(JVAR)
59     IF(Z(JVAR).GT.ZMAX) ZMAX=Z(JVAR)
70     END IF
71     NS=NS+1
72     GO TO 6
73 7 NS=NS-1
74     TYPE 8,ZMIN,ZMAX
75 8 FORMAT(' MINIMUM DATA VALUE ='F10.3'; MAXIMUM ='F10.3/' INPUT '
76 + 'DATA-VALUE INCREMENT PER INCH OF RADIUS')
77     ACCEPT *,RSCALE
78 10 CLOSE(UNIT=1)
79     IF(JTYPE.EQ.140) THEN
80         TYPE 11,NS,DMIN,DMAX,TMN,TMX
81         TYPE *, '>MIN. DEPTH, MAX. DEPTH, & DEPTH/INCH FOR PLOT AXIS ??'
82     END IF
83     IF(JTYPE.EQ.141) THEN
84         TYPE 12,NS,DMIN,DMAX,TMN,TMX
85         TYPE *, '>MIN. & MAX. LATITUDES & DEGREES/INCH FOR PLOT AXIS ??'
86     END IF
87 11 FORMAT('15' SAMPLES COUNTED'/' MINIMUM DEPTH ='F8.0' M, MAXIMUM '
88 + 'DEPTH ='F8.0' M/' MINIMUM AGE ='F5.1' M.Y., MAXIMUM AGE ='
89 + 'F5.1' M.Y. '/')
90 12 FORMAT('15' SAMPLES COUNTED'/' MINIMUM LATITUDE ='F6.1' DEG, '
91 + 'MAXIMUM LATITUDE ='F6.1' DEG/' MINIMUM AGE ='F5.1' M.Y., '
92 + 'MAXIMUM AGE ='F5.1' M.Y. '/')
93     ACCEPT *,YMIN,YMAX,YSCALE
94     TYPE *, '>MINIMUM AGE, MAXIMUM AGE, & M.Y./INCH FOR PLOT AXIS ??'
95     ACCEPT *,XMIN,XMAX,XSCALE
96     TYPE *, '>INPUT SCALE FACTOR FOR SIZE OF PLOT !'
97     ACCEPT *,SIZE
98 C PLOT
99     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
100     CALL ASSIGN(9,'PLOT.PLT')
101     CALL PLOTS(0,0,9)
102     IF(SIZE.GT.0.) CALL FACTOR(SIZE)
103     CALL PLOT(0.,0.,-3)
104     X1=0.
105     X2=(XMAX-XMIN)/XSCALE
106     XL=FLUAT(INT(X2+.5))
107     Y1=(YMIN-YMAX)/-YSCALE
108     Y2=0.
109     YL=FLUAT(INT(Y1+.5))
110     CALL SYMBOL(0.,YL+.5,.35,TITLE,0.,80)
111     CALL PLOT(XL,YL,3)
112     DO 20 I=1,2*INT(XL)
113     CALL PLOT(XL-I*.5,YL,2)
114     CALL PLOT(XL-I*.5,YL-.1,2)
115 20 CALL PLOT(XL-I*.5,YL,2)
116     DO 21 I=1,2*INT(YL)
117     CALL PLOT(0.,YL-I*.5,2)
118     CALL PLOT(.1,YL-I*.5,2)
119 21 CALL PLOT(0.,YL-I*.5,2)
120     DO 22 I=1,2*INT(XL)
121     CALL PLOT(I*.5,0.,2)
122     CALL PLOT(I*.5,.1,2)

```

```

123 22 CALL PLOT(I*.5,0.,2)
124    DO 23 I=1,2*INT(YL)
125      CALL PLOT(XL,I*.5,2)
126      CALL PLOT(XL-.1,I*.5,2)
127 23 CALL PLOT(XL,I*.5,2)
128    IF(JTYPE.EQ.140) MINT=1000/YSCALE
129    IF(JTYPE.EQ.141) MINT=20/YSCALE
130    DO 24 I=1,INT(YL)+1,MINT
131 24 CALL NUMBER(XL+.1,YL-.1-(I-1),.21,YMIN+(I-1)*YSCALE,0.,-1)
132    IF(JTYPE.EQ.140) THEN
133      CALL SYMBOL(XL+1.23,(YL/2)-2.,.21,'PALEODEPTH (METERS)',90.,19)
134    END IF
135    IF(JTYPE.EQ.141) THEN
136      CALL SYMBOL(XL+.81,(YL/2)+1.36,.21,'PALEOLATITUDE',-90.,13)
137    END IF
138    DO 25 I=1,INT(XL)+1,INT(1/XSCALE)
139 25 CALL NUMBER(-.21+(I-1),-.3,.21,XMAX-(I-1)*XSCALE,0.,-1)
140    CALL SYMBOL((XL/2)-1.05,-.6,.21,'AGE (M.Y.)',0.,10)
141    IF(ITYPE.EQ.140) GO TO 30
142    DO 26 I=1,NS
143      READ(1,*) D,TMIN,TMAX
144      T1=(TMIN-XMAX)/-XSCALE
145      T2=(TMAX-XMAX)/-XSCALE
146      T=(T1+T2)/2
147      Y=(D-YMAX)/-YSCALE
148      CALL SYMBOL(T1,Y,.07,13,0.,-1)
149      CALL PLOT(T+.1,Y,2)
150      CALL NUMBER(T-.07,Y-.035,.07,FLOAT(I),0.,-1)
151      CALL PLOT(T-.1,Y,3)
152 26 CALL SYMBOL(T2,Y,.07,13,0.,-2)
153    GO TO 40
154 30 DO 31 I=1,NS
155      READ(1,*) D,AGE,(Z(K),K=1,IVAR)
156      AGE=(AGE-XMAX)/-XSCALE
157      Y=(D-YMAX)/-YSCALE
158      CALL NUMBER(AGE-.1,Y-.035,.07,Z(JVAR)*100.,0.,-1)
159 31 CALL DOT(AGE,Y,Z(JVAR)/RSCALE)
160 40 CALL PLOT(0.,0.,999)
161    TYPE *, 'PLOT IS SPOOLED ON FILE=PLOT.PLT !!!'
162    STOP
163  END

```



```

1  C <CORREL> COMPUTES CORRELATION MATRIX
2  C
3  C INPUT - RAW SAMPLE VECTORS
4  C OUTPUT - MEANS OR CENTROID VECTOR
5  C          - STANDARD DEVIATIONS VECTOR
6  C          - RAW SUMS OF SQUARES
7  C          - CORRELATION MATRIX
8  C
9  C REQUIRES SUBROUTINE CORREL, MATPRINT
10 C
11     DIMENSION A(100,100),XN(100),SD(100)
12     CHARACTER FILENAME*13,TODAY*9
13     PARAMETER NDI=100
14     TYPE 1
15     1 FORMAT(///' <CORREL> COMPUTES CORRELATION MATRIX !!!')
16     TYPE *, '>FILENAME FOR INPUT DATA ? (ID, ID, ASSOCIATED VARIABLES)'
17     ACCEPT *, FILENAME
18     TYPE 2, NDI
19     2 FORMAT(' >HOW MANY VARIABLES? MAXIMUM IS ',I4)
20     ACCEPT *, M
21     CALL CORREL(FILENAME,M,XN,SD,A,NDI)
22     OPEN(UNIT=2,NAME='CORREL.TMP')
23     DO 5 J=1,M
24     5 WRITE(2,*) (A(J,K),K=J,M)
25     OPEN(UNIT=5,NAME='PRINT.TMP')
26     CALL DATE(TODAY)
27     WRITE(5,7) TODAY,FILENAME
28     7 FORMAT(1H1/' CORRELATION MATRIX'///' COMPUTED 'A9
29     + ' FROM DATA UNDER FILENAME ['A13']')
30     + '////' CORRELATION MATRIX, THE MATRIX OF STANDARDIZED ',
31     + ' COVARIANCES (CORRELATION COEFFICIENTS)')
32     CALL MATPRINT(A,M,NDI,5)
33     OPEN(UNIT=4,NAME='RAWSSQ.TMP')
34     DO 9 I=1,M
35     9 SD(I)=A(I,I)
36     WRITE(4,*) (SD(J),J=1,M)
37     CLOSE(UNIT=4)
38     CALL DATE(TODAY)
39     WRITE(5,10) TODAY,FILENAME,(SD(J),J=1,M)
40     10 FORMAT(////' RAW SUMS OF SQUARES ("VARIANCES") FROM PRINCIPAL '
41     + ' DIAGONAL OF CROSS-PRODUCT MATRIX'///' COMPUTED 'A9' FOR '
42     + ' DATA UNDER FILENAME ['A13']'//(2X,10E13.5))
43     OPEN(UNIT=3,NAME='MEANS.TMP')
44     WRITE(3,*) (XN(J),J=1,M)
45     WRITE(5,12) TODAY, FILENAME, (XN(J),J=1,M)
46     12 FORMAT(////' MEANS OR CENTROID VECTOR'///' COMPUTED 'A9' FOR '
47     + ' DATA UNDER FILENAME ['A13']'//(2X,10E12.4))
48     OPEN(UNIT=4,NAME='STDEVS.TMP')
49     WRITE(4,*) (SD(J),J=1,M)
50     IF(JANS.EQ.14Y) WRITE(5,13) TODAY, FILENAME, (SD(J),J=1,M)
51     13 FORMAT(////' VECTOR OF STANDARD DEVIATIONS'///' COMPUTED 'A9
52     + ' FOR DATA UNDER FILENAME ['A13']'//(2X,10E12.4))
53     TYPE 14,FILENAME
54     14 FORMAT(//' CORRELATION MATRIX FOR 'A13' STORED UNDER '
55     + ' FILE=CORREL.TMP'/' VECTOR OF RAW SUMS OF SQUARES STORED UNDER '
56     + ' FILE=RAWSSQ.TMP'/' VECTOR OF MEANS (CENTROID VECTOR) STORED '
57     + ' UNDER FILE=MEANS.TMP'/' VECTOR OF STANDARD DEVIATIONS STORED '
58     + ' UNDER FILE=STDEVS.TMP'/' PRINTED RESULTS ALL SAVED ON FILE='
59     + ' PRINT.TMP')
60     STOP
61     END

```

```

1  C <COVAR> COMPUTES VARIANCE-COVARIANCE (DISPERSION) MATRIX
2  C
3  C INPUT - RAW SAMPLE VECTORS
4  C OUTPUT - MEANS OR CENTROID VECTOR
5  C          - STANDARD DEVIATIONS VECTOR
6  C          - RAW SUMS OF SQUARES
7  C          - VARIANCE-COVARIANCE (DISPERSION) MATRIX
8  C
9  C REQUIRES SUBROUTINE COVAR, MATPRINT
10 C
11     DIMENSION A(100,100),XN(100),SD(100)
12     CHARACTER FILENAME*13,TODAY*9
13     PARAMETER NDIM=100
14     TYPE 1
15     1 FORMAT(///' <COVAR> COMPUTES VARIANCE-COVARIANCE MATRIX !!!')
16     TYPE *, '>FILENAME FOR INPUT DATA ? (I), ID, ASSOCIATED VARIABLES)'
17     ACCEPT *, FILENAME
18     TYPE 2, NDIM
19     2 FORMAT(' >HOW MANY VARIABLES? MAXIMUM IS ',I4)
20     ACCEPT *, M
21     CALL COVAR(FILENAME,M,XN,SD,A,NDIM)
22     OPEN(UNIT=2,NAME='COVAR.TMP')
23     DO 5 J=1,M
24     5 WRITE(2,*) (A(J,K),K=J,M)
25     OPEN(UNIT=3,NAME='PRINT.TMP')
26     CALL DATE(TODAY)
27     WRITE(3,7) TODAY,FILENAME
28     7 FORMAT(141/' DISPERSION MATRIX'///' COMPUTED 'A9
29     + ' FROM DATA UNDER FILENAME ['A13']'
30     + ' ///' DISPERSION MATRIX, THE MATRIX OF VARIANCES AND ',
31     + ' COVARIANCES')
32     CALL MATPRINT(A,M,NDIM,5)
33     OPEN(UNIT=4,NAME='RAWSSQ.TMP')
34     DO 9 I=1,M
35     9 SD(I)=A(I,I)
36     WRITE(4,*) (SD(J),J=1,M)
37     CLOSE(UNIT=4)
38     CALL DATE(TODAY)
39     WRITE(5,10) TODAY,FILENAME,(SD(J),J=1,M)
40     10 FORMAT(////' RAW SUMS OF SQUARES ("VARIANCES") FROM PRINCIPAL '
41     + ' DIAGONAL OF CROSS-PRODUCT MATRIX'///' COMPUTED 'A9' FOR '
42     + ' DATA UNDER FILENAME ['A13']'//(2X,10E13.5))
43     OPEN(UNIT=3,NAME='MEANS.TMP')
44     WRITE(3,*) (XN(J),J=1,M)
45     WRITE(5,12) TODAY, FILENAME, (XN(J),J=1,M)
46     12 FORMAT(////' MEANS OR CENTROID VECTOR'///' COMPUTED 'A9' FOR '
47     + ' DATA UNDER FILENAME ['A13']'//(2X,10E12.4))
48     OPEN(UNIT=4,NAME='STDEVS.TMP')
49     WRITE(4,*) (SD(J),J=1,M)
50     IF(JANS.EQ.1HY) WRITE(5,13) TODAY, FILENAME, (SD(J),J=1,M)
51     13 FORMAT(////' VECTOR OF STANDARD DEVIATIONS'///' COMPUTED 'A9
52     + ' FOR DATA UNDER FILENAME ['A13']'//(2X,10E12.4))
53     TYPE 14,FILENAME
54     14 FORMAT(///' VARIANCE-COVARIANCE MATRIX FOR 'A13' STORED UNDER '
55     + ' FILE=COVAR.TMP'/' VECTOR OF RAW SUMS OF SQUARES STORED UNDER '
56     + ' FILE=RAWSSQ.TMP'/' VECTOR OF MEANS (CENTROID VECTOR) STORED '
57     + ' UNDER FILE=MEANS.TMP'/' VECTOR OF STANDARD DEVIATIONS STORED '
58     + ' UNDER FILE=STDEVS.TMP'/' PRINTED RESULTS ALL SAVED ON FILE='
59     + ' PRINT.TMP')
60     STOP

```



```

1  C <CROSSPROD> COMPUTES CROSS-PRODUCT MATRIX
2  C
3  C INPUT   - RAW SAMPLE VECTORS
4  C OUTPUT  - RAW SUMS OF SQUARES
5  C          - RAW SUMS
6  C          - MATRIX OF RAW SUMS OF SQUARES AND CROSS-PRODUCTS
7  C
8  C REQUIRES SUBROUTINE CROSSPROD, MATPRINT
9  C
10     DIMENSION A(100,100),SD(100),XN(100)
11     CHARACTER FILENAME*13,TODAY*9
12     PARAMETER MDIM=100
13     TYPE 1
14     1 FORMAT(///' <CROSSPROD> COMPUTES CROSS-PRODUCT MATRIX'//)
15     TYPE *, '>FILENAME FOR INPUT DATA ? (ID, ID, ASSOCIATED VARIABLES)'
16     ACCEPT *, FILENAME
17     TYPE 2, MDIM
18     2 FORMAT(' >HOW MANY VARIABLES? MAXIMUM IS ',I4)
19     ACCEPT *, M
20     CALL CROSSPROD(FILENAME,M,XN,A,MDIM)
21     OPEN(UNIT=2,NAME='CROSSPROD.TMP')
22     DO 5 J=1,M
23     5 WRITE(2,*) (A(J,K),K=J,M)
24     OPEN(UNIT=5,NAME='PRINT.TMP')
25     CALL DATE(TODAY)
26     CALL MATPRINT(A,M,MDIM,5)
27     6 FORMAT(1H1/' CROSS-PRODUCT MATRIX'//' COMPUTED 'A9
28     + ' FROM DATA IN FILE ['A13']')
29     + '////' CROSS-PRODUCT MATRIX, THE MATRIX OF RAW SUMS OF SQUARES '
30     + 'AND CROSS PRODUCTS')
31     OPEN(UNIT=4,NAME='RAWSSQ.TMP')
32     DO 9 I=1,M
33     9 SD(I)=A(I,I)
34     WRITE(4,*) (SD(J),J=1,M)
35     CALL DATE(TODAY)
36     WRITE(5,10) TODAY,FILENAME,(SD(J),J=1,M)
37     10 FORMAT(/////' RAW SUMS OF SQUARES ("VARIANCES") FROM PRINCIPAL '
38     + 'DIAGONAL OF CROSS-PRODUCT MATRIX'//' COMPUTED 'A9' FOR '
39     + 'DATA UNDER FILENAME ['A13']'//(2X,10E13.5))
40     TYPE 11,FILENAME
41     11 FORMAT(//' CROSS-PRODUCT MATRIX FOR ['A13'] STORED UNDER '
42     + 'FILE=CROSSPROD.TMP !!'//' VECTOR OF RAW SUMS OF SQUARES STORED '
43     + 'UNDER FILE=RAWSSQ.TMP !!'//' ALL PRINTED RESULTS SAVED ON FILE='
44     + 'PRINT.TMP')
45     STOP
46     END

```

```

1  C <DATAPLOT> PRESENTS DATA TABLE ON CALCJMP PLOTTER
2  C
3  C   REQUIRES SYMBOLDFT,CALSPPOOL/LIB
4  C
5      DIMENSION X(100,100),MAX(100)
6      CHARACTER FILENAME*13
7      TYPE *, 'NUMBER OF ROWS, NUMBER OF COLUMNS'
8      ACCEPT *,NR,NC
9      TYPE *, 'FILENAME?'
10     ACCEPT *,FILENAME
11     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
12     DO 1 I=1,NR
13     1 READ(1,*) (X(I,J),J=1,NC)
14     DO 3 J=1,NC
15     XMAX=X(I,J)
16     DO 2 I=2,NR
17     2 IF(XMAX.LT.X(I,J)) XMAX=X(I,J)
18     IF(XMAX.LT.10.) MAX(J)=2
19     IF(XMAX.GE.10.AND.XMAX.LT.100.) MAX(J)=3
20     3 IF(XMAX.GE.100.) MAX(J)=4
21     TYPE *, 'NUMBER OF TIMES TO OVERPRINT?'
22     ACCEPT *,NTIMES
23     TYPE *, '>INPUT SCALE FACTOR'
24     ACCEPT *,SIZE
25  C PLOT
26     CALL ASSIGN(9,'PLOT.PLT')
27     CALL PLOTS(0,0,9)
28     IF(SIZE.GT.0.) CALL FACTOR(SIZE)
29     XAXIS=0.
30     Y=0.
31     DO 6 J=1,NC
32     DO 5 I=1,NR
33     XAXIS=XAXIS+.21
34     DO 5 K=1,NTIMES
35     5 CALL NUMBER(XAXIS,Y,.14,X(I,J),90.,-1)
36     Y=Y+.14*MAX(J)
37     6 XAXIS=0.
38     CALL PLOT(0.,0.,999)
39     TYPE *, 'PLOT SPOOLED ON FILE=PLOT.PLT !!!'
40     STOP
41     END

```



```

1  C <DPCA> COMPUTES A "D-MODE" P.C.A. (PRINCIPAL COMPONENTS OF A
2  C     VARIANCE-COVARIANCE MATRIX). BOTH LOADINGS AND SCORES ARE
3  C     COMPUTED.
4  C
5  C REQUIRES SUBROUTINES COVAR, HOW, DSCORES, PRINCO
6  C
7  C     DIMENSION X(100),R(100,100),XM(100),XS(100)
8  C     DIMENSION E(100),V(100,100),A(100),B(100),C(100),D(100),F(100)
9  C     CHARACTER FILENAME*13,TODAY*9
10 C     PARAMETER MDIM=100
11 C     TYPE 1
12 1 FORMAT(///' THIS IS <DPCA> !'///' IT COMPUTES A D-MODE P.C.A'//
13 + ' >FILENAME FOR DATA TO BE ANALYZED ? "DEPTH(M), AGE(M.Y.), '
14 + ' ASSOCIATED VARIABLES"'')
15 ACCEPT *,FILENAME
16 OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
17 OPEN(UNIT=3,NAME='PRINT.TMP')
18 CALL DATE(TODAY)
19 TYPE *, '>HOW MANY VARIABLES IN EACH SAMPLE VECTOR?'
20 ACCEPT *,NV
21 WRITE(3,2) FILENAME,TODAY,NV,FILENAME
22 2 FORMAT(1H1'D-MODE P.C.A OF DATA STORED IN FILE ['A13'] '
23 + ' COMPUTED 'A9//110' DATA VARIABLES SPECIFIED FOR EACH SAMPLE'//
24 + ' NO. DEPTH (M) AGE (M.Y.) TOTAL ASSOCIATED DATA '
25 + ' VARIABLES INPUT FROM FILE ['A13']'//)
26 NS=0
27 3 READ(1,*,END=5) XID,AGE,(X(J),J=1,NV)
28 NS=NS+1
29 WRITE(3,4) NS,XID,AGE,(X(J),J=1,NV)
30 4 FORMAT(16,2F11.3,110,2X,10F8.3/(30X,10F8.3))
31 GO TO 3
32 5 TYPE *, 'DATA HAS BEEN INPUT !'
33 CLOSE(UNIT=1)
34 CALL COVAR(FILENAME,NV,XM,XS,R,MDIM)
35 TYPE *, 'D-MATRIX HAS BEEN COMPUTED !'
36 WRITE(3,8) FILENAME
37 8 FORMAT(1H1'STATISTICS FOR INPUT DATA FILE ['A13']'//8X'NO.'8X
38 + 'MEAN'6X'STD. DEV.'//)
39 TRACE=0.
40 DO 10 J=1,NV
41 WRITE(3,9) J,XM(J),XS(J)
42 9 FORMAT(110,2F13.3)
43 10 TRACE=TRACE+R(J,J)
44 TYPE 11,TRACE
45 11 FORMAT(///' TRACE = 'F10.1//)
46 TYPE *, '>HOW MANY EIGENVECTORS DO YOU WANT EXTRACTED?'
47 ACCEPT *,NE
48 CALL HOW(NV,MDIM,NE,R,E,V,A,B,C,D)
49 TYPE *, 'EIGENVALUES AND EIGENVECTORS HAVE BEEN COMPUTED !'
50 WRITE(3,12) FILENAME,TODAY
51 12 FORMAT(1H1'D-MODE PRINCIPAL COMPONENTS ANALYSIS OF DATA IN FILE '
52 + '['A13']'//' EIGENVALUES FOR D-MODE EIGENVECTORS '
53 + ' COMPUTED 'A9///12X'EIGENVALUE % VARIANCE CUM.VAR.'//)
54 TYPE *, ' EIGENVALUE % VARIANCE CUM.VAR.'
55 CUM=0.
56 DO 13 I=1,NE
57 VAR=(E(I)/TRACE)*100.
58 CUM=CUM+VAR
59 WRITE(3,14) I,E(I),VAR,CUM
60 13 TYPE 14,I,E(I),VAR,CUM
61 14 FORMAT(17,1E12,2,2E12,1)

```

```

62     TYPE *, '>HOW MANY EIGENVECTORS DO YOU WANT TO SAVE ?'
63     ACCEPT *, ME
64     OPEN(UNIT=2, NAME='DSCORES.TMP')
65     WRITE(3,15) FILENAME, TODAY
66 15  FORMAT(1H1'SCORES OF SAMPLES ON D-MODE PRINCIPAL COMPONENTS '
67     + 'FOR DATA FROM FILE ['A13'] ON 'A9//5X'NO.    DEPTH(M)  AGE(M.Y.)'
68     + ' D-MODE PRINCIPAL COMPONENTS SCORES'//)
69     OPEN(UNIT=1, NAME=FILENAME, TYPE='OLD')
70     DO 20 I=1, NV
71     READ(1,*) XID, AGE, (X(J), J=1, NV)
72     CALL DSCORES(X, XM, V, E, 1, NV, 1, MDIM, F)
73     WRITE(2,*) XID, AGE, (F(J), J=1, ME)
74 20  WRITE(3,21) I, XID, AGE, (F(J), J=1, ME)
75 21  FORMAT(I7,2F11.3,10F10.2/(29X,10F10.2))
76     CLOSE(UNIT=1)
77     TYPE *, 'D-SCORES COMPUTED AND STORED ON FILE=DSCORES.TMP !!'
78     DO 24 I=1, NV
79     DO 24 J=1, ME
80 24  R(I,J)=V(I,J)
81     CALL PRINCO(R, E, NV, MDIM)
82     TYPE *, 'EIGENVECTORS HAVE BEEN SCALED TO PRINCIPAL COMPONENTS !'
83     OPEN(UNIT=4, NAME='DLOADINGS.TMP')
84     WRITE(3,30) FILENAME
85 30  FORMAT(1H1'EIGENVECTORS OF D-MATRIX SCALED BY EIGENVALUES TO '
86     + 'D-MODE PRINCIPAL COMPONENTS'/' FOR DATA ON FILE ['A13']'
87     + '//2X' VAR.NO.  XEXPLAINED  D-MODE PRINCIPAL COMPONENTS LOADINGS'
88     + '(*100)'//)
89     TOTVAR=0.
90     DO 32 I=1, NV
91     COMM=0.
92     DO 31 J=1, ME
93 31  COMM=COMM+R(I,J)*R(I,J)
94     TOTVAR=TOTVAR+COMM
95     WRITE(4,*) FLOAT(I), FLOAT(1), (R(I,J), J=1, ME)
96     ICOMM=NINT(COMM*100.)
97 32  WRITE(3,33) I, ICOMM, ((R(I,J)*100), J=1, ME)
98 33  FORMAT(I6,113,10F10.1/(19X,10F10.1))
99     WRITE(3,34) NINT((TOTVAR*100.)/NV)
100 34  FORMAT('//3X,I5'% OF THE TOTAL OBSERVED VARIANCE ACCOUNTED FOR')
101     TYPE *, 'PRINTED RESULTS STORED ON FILE=PRINT.TMP !'
102     STOP
103     END

```



```

1  C <EIGEN> - COMPUTES EIGENVALUES & EIGENVECTORS BY CALLING ROUTINE <HOW>
2  C
3  C   INPUT  - REAL SYMMETRIC MATRIX
4  C   OUTPUT - EIGENVALUES
5  C           - EIGENVECTORS
6  C   REQUIRES SUBROUTINES HOW, MATPRINT
7  C
8      DIMENSION R(100,100),E(100),V(100,100),A(100),B(100),C(100),D(100)
9      CHARACTER FILENAME*13,TODAY*9
10     PARAMETER NDIM=100
11     TYPE 1
12 1 FORMAT(///' <EIGEN> COMPUTES EIGENVALUES & EIGENVECTORS OF '
13 + 'A REAL SYMMETRIC MATRIX'///'>WHAT'S ITS FILENAME? (MATRIX MUST '
14 + 'BE IN UPPER TRIANGULAR FORM !!!)')
15     ACCEPT *,FILENAME
16     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
17     TYPE 2,NDIM
18 2 FORMAT(' >ORDER OF INPUT MATRIX?  MAXIMUM IS'14)
19     ACCEPT *,NVAR
20     DO 3 J=1,NVAR
21 3 READ(1,*) (R(J,I),I=J,NVAR)
22     TRACE=0.
23     DO 4 J=1,NVAR
24     TRACE=TRACE+R(J,J)
25     DO 4 I=1,NVAR
26 4 R(I,J)=R(J,I)
27     TYPE 5,TRACE
28 5 FORMAT(///' TRACE = 'F13.6//)
29     OPEN(UNIT=4,NAME='PRINT.TMP')
30     CALL DATE(TODAY)
31     WRITE(4,7) FILENAME,TODAY
32     TYPE *, '>PRINT INPUT MATRIX?'
33     ACCEPT 6,IANS
34 6 FORMAT(A1)
35     IF(IANS.EQ.1HY) THEN
36         WRITE(4,8)
37         CALL MATPRINT(R,NVAR,NDIM,4)
38     END IF
39 7 FORMAT(1H1/' COMPUTATION OF EIGENVALUES AND EIGENVECTORS OF ['A13
40 + ']' 'A9'////)
41 8 FORMAT(' SYMMETRIC MATRIX INPUT TO SUBROUTINE <HOW>'//)
42     WRITE(4,9) TRACE
43 9 FORMAT(///' TRACE OF INPUT MATRIX = ' F13.6//)
44     TYPE 10, NVAR
45 10 FORMAT(' >HOW MANY EIGENVECTORS DO YOU WANT EXTRACTED?'/
46 + ' ' (MAXIMUM IS THE ORDER OF YOUR INPUT MATRIX, WHICH IS '13)
47     ACCEPT *,NVECT
48     CALL HOW(NVAR,NDIM,NVECT,R,E,V,A,B,C,D)
49     OPEN(UNIT=2,NAME='EIGENVAL.TMP')
50     WRITE(2,*) (E(I),I=1,NVAR)
51     TYPE *, '>WHAT'S THE SMALLEST EIGENVALUE YOU WANT LISTED?'
52     ACCEPT *,CUT
53     WRITE(4,11) CUT,TODAY,FILENAME
54 11 FORMAT(1H1///' EIGENVALUES GREATER THAN'F13.6//
55 + ' ' COMPUTED 'A9' FOR DATA STORED UNDER FILENAME ['A13']'///
56 + ' 12X'EIGENVALUE  VARIANCE  CUM.VAR.'//)
57     TYPE 12
58 12 FORMAT(///12X'EIGENVALUE  VARIANCE' CUM.VAR.'//)
59     CUM=0.
60     DO 15 I=1,NVAR

```

```

52     VAR=(E(I)/TRACE)*100.
53     CUM=CUM+VAR
54     TYPE 13,I,E(I),VAR,CUM
55 13  FORMAT(I7,4X,3F10.3)
56     WRITE(4,13) I,E(I),VAR,CUM
57     IF(E(I).LT.CUT) GO TO 16
58 15  CONTINUE
59 16  TYPE *, '>HOW MANY EIGENVECTORS TO YOU WANT TO SAVE?'
70     ACCEPT *,MVECT
71     OPEN(UNIT=3,NAME='EIGENVEC.TMP')
72     IF(MVECT.GT.NVECT) MVECT=NVECT
73     DO 17 M=1,NVAR
74 17  WRITE(3,*) (V(M,N),N=1,MVECT)
75     WRITE(4,18) TODAY,FILENAME
76 18  FORMAT(1H1///' EIGENVECTORS'///' COMPUTED 'A9' FOR DATA STORED '
77     + ' UNDER FILENAME ['A13']'///' MATRIX OF EIGENVECTORS'//)
78     DO 19 M=1,NVAR
79 19  WRITE(4,20) M,(V(M,N),N=1,MVECT)
80 20  FORMAT(I5,15F8.3/(5X,15F8.3))
81     TYPE 21,FILENAME
82 21  FORMAT(///' EIGENVALUES AND EIGENVECTORS FOR DATA ['A13'] STORED '
83     + ' UNDER FILES "EIGENVAL.TMP" & "EIGENVEC.TMP"'/' ALL PRINTED '
84     + ' RESULTS IN "PRINT.TMP" FILE')
85     STOP
86     END

```



```

1  C <ELLIPSE> PROJECTS SAMPLE GROUPS AS EQUAL FREQUENCY ELLIPSES ON THE
2  C      PLANE OF THE PRINCIPAL COMPONENTS
3  C
4  C THIS VERSION DEFERS PLOTTING BY SPOOLING ON TO A DISK FILE
5  C
6  C REQ. SUBROUTINES  DIAG, ELLIPSE, SYMBOLDFT, CALSPOOL/LIBRARY
7  C
8      DIMENSION D(4),V(2),U(4),SIT(2,50),COT(2,50)
9      DIMENSION XO(2,50),YO(2,50),AA(2,50),BB(2,50)
10     DIMENSION X(100,50),Y(100,50),Z(100,50),MS(50)
11     CHARACTER FILENAME*13
12     PARAMETER XMIN=-1.,YMIN=-1.,SCALE=.2,XMAX=1.,YMAX=1.
13     PARAMETER NDIM=100,NSET=50
14     TYPE *, '<ELLIPSE> PROJECTS SAMPLE GROUPS AS ELLIPSES'
15     OPEN(UNIT=2,NAME='PRINT.TMP')
16     TYPE *, '>PLOT DATA POINTS(P), ELLIPSES(E), OR BOTH(B)'
17 1  ACCEPT 2,JANS
18 2  FORMAT(A1)
19     IF(JANS.NE.1HP.AND.JANS.NE.1HE.AND.JANS.NE.1HB) GO TO 1
20     IF(JANS.EQ.1HP.OR.JANS.EQ.1HB) THEN
21         TYPE *, '>PLOT POINTS AS INTEGERS REFLECTING DATA SET NO. ?'
22         ACCEPT 2,NUMB
23     END IF
24     DO 14 NS=1,NSET
25         TYPE *, '>NAME OF FILE CONTAINING PRINCIPAL COMPONENTS LOADINGS?'
26         ACCEPT *,FILENAME
27         IF(JANS.EQ.1HP) WRITE(2,3) FILENAME
28         IF(JANS.EQ.1HE) WRITE(2,4) FILENAME
29         IF(JANS.EQ.1HB) WRITE(2,5) FILENAME
30 3  FORMAT(///' SAMPLES FROM INPUT FILE ['A13']/' PLOTTED '
31  + ' INDIVIDUALLY ON PRINCIPAL COMPONENTS PLANE')
32 4  FORMAT(///' SAMPLES FROM INPUT FILE ['A13']/' PLOTTED AS AN '
33  + ' ELLIPSE ON PRINCIPAL COMPONENTS PLANE')
34 5  FORMAT(///' SAMPLES FROM INPUT FILE ['A13']/' PLOTTED INDIVIDUA
35  + ' LLY AND AS AN ELLIPSE ON PRINCIPAL COMPONENTS PLANE')
36     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
37     DO 6 M=1,NDIM
38 6  READ(1,*,END=7) DUM,DUM,X(M,NS),Y(M,NS),Z(M,NS)
39 7  M=M-1
40     MS(NS)=M
41     CLOSE(UNIT=1)
42     WRITE(2,8) NS,M
43 8  FORMAT(' SET NO.'14' CONTAINS'14' SAMPLES')
44     IF(JANS.EQ.1HE.OR.JANS.EQ.1HB) THEN
45         DO 11 N=1,2
46             ASUM=0.
47             BSUM=0.
48             D(1)=0.
49             D(2)=0.
50             D(3)=0.
51             D(4)=0.
52             DO 10 I=1,M
53             B=Y(I,NS)
54             IF(N.EQ.1) A=X(I,NS)
55             IF(N.EQ.2) A=Z(I,NS)
56             ASUM=ASUM+A
57             BSUM=BSUM+B
58             D(1)=D(1)+A*A
59             D(2)=D(2)+A*B
60             D(3)=D(3)+B*B

```

```

52      XO(N,NS)=ASUM/M
53      YO(N,NS)=BSUM/M
54      D(1)=(D(1)-ASUM*ASUM/M)/M
55      D(2)=(D(2)-ASUM*BSUM/M)/M
56      D(3)=(D(3)-BSUM*ASUM/M)/M
57      D(4)=(D(4)-BSUM*BSUM/M)/M
58      CALL DIAG(D,V,U)
59      AA(N,NS)=SQRT(V(1))
60      BB(N,NS)=SQRT(V(2))
61      COT(N,NS)=U(1)
62      SIT(N,NS)=U(2)
63      11 WRITE(2,12) D,XO(N,NS),YO(N,NS),2*AA(N,NS),2*BB(N,NS),U(1),U(2)
64      12 FORMAT(/6X'DISPERSION MATRIX'
65      + 6X,2E12.3/29X,2E12.3/6X'ELLIPSE CENTERED AT'15X,2F12.3/6X
66      + 'LENGTHS OF PRINCIPAL AXES'9X,2F12.3/6X'DIRECTION SIN & COS '
67      + 'OF MAJOR AXIS '2F12.3)
68      END IF
69      TYPE #,'>PLOT ANOTHER SET OF SAMPLES?'
70      13 ACCEPT 2, IANS
71      IF(IANS.NE.1HY.AND.IANS.NE.1HY) GO TO 13
72      14 IF(IANS.EQ.1HN) GO TO 18
73      18 TYPE #,'>ALTER SIZE OF PLOT? INPUT SCALE FACTOR.'
74      ACCEPT #,SIZE
75      C PLOT
76      CALL ASSIGN(9,'PLOT.PLT')
77      CALL PLOTS(0,0,9)
78      IF(SIZE.GT.0.) CALL FACTOR(SIZE)
79      CALL PLOT(0.,0.,-3)
80      X1=(XMAX-XMIN)/(2*SCALE)
81      X2=(XMAX-XMIN)/SCALE
82      Y1=0.
83      Y2=(YMAX-YMIN)/SCALE
84      CALL PLOT(X1,Y1,3)
85      DO 21 I=1,10
86      CALL PLOT(X1+I*.5,Y1,2)
87      CALL PLOT(X1+I*.5,Y1+.1,2)
88      21 CALL PLOT(X1+I*.5,Y1,2)
89      DO 22 I=1,20
90      CALL SYMBOL(X2,Y1+I*.5,.21,13,90.,-2)
91      DO 23 I=1,10
92      CALL PLOT(X2-I*.5,Y2,2)
93      CALL PLOT(X2-I*.5,Y2-.1,2)
94      23 CALL PLOT(X2-I*.5,Y2,2)
95      DO 24 I=1,20
96      CALL PLOT(X1,Y2-I*.5,2)
97      CALL PLOT(X1+.1,Y2-I*.5,2)
98      24 CALL PLOT(X1,Y2-I*.5,2)
99      CALL NUMBER(X1,Y1-.1,.21,0.,-90.,-1)
100     CALL NUMBER(X1+2.4,Y1-.1,.21,.5,-90.,1)
101     CALL NUMBER(X1+4.9,Y1-.1,.21,1.,-90.,1)
102     CALL SYMBOL(X1-.44,Y1-1.2,.28,'PRINCIPAL COMPONENT 1',0.,21)
103     CALL NUMBER(X1-.3,Y1+.45,.21,-1.,-90.,1)
104     CALL NUMBER(X1-.3,Y1+2.9,.21,-.5,-90.,1)
105     CALL NUMBER(X1-.3,Y1+5.1,.21,0.,-90.,-1)
106     CALL NUMBER(X1-.3,Y1+7.8,.21,.5,-90.,1)
107     CALL NUMBER(X1-.3,Y1+10.3,.21,1.,-90.,1)
108     CALL SYMBOL(X1-.8,Y1+7.9,.28,'PRINCIPAL COMPONENT 2',-90.,21)
109     DO 26 I=1,NS
110     IF(JANS.EQ.14P.OR.JANS.EQ.1HB) THEN
111     DO 25 J=1,4S(I)

```



```

123      YY=(Y(J,I)-YMIN)/SCALE
124      IF(NUMB.EQ.1HN) THEN
125      CALL SYMBOL(XX,YY,.07,1,0.,-1)
126      ELSE
127      IF(NS.EQ.1) CALL NUMBER(XX,YY,.07,FLOAT(J),-90.,-1)
128      IF(NS.GT.1) CALL NUMBER(XX,YY,.07,FLOAT(I),-90.,-1)
129      END IF
130 25    CONTINUE
131      END IF
132 26 IF(JANS.EQ.1HE.OR.JANS.EQ.1HB)
133 +   CALL ELLIPSE(XD(1,I),YD(1,I),AA(1,I),BB(1,I),COT(1,I),SIT(1,I),
134 +   XMIN,YMIN,SCALE,XMAX,YMAX)
135      CALL PLOT(10.,0.,-3)
136      X1=0.
137      DO 30 I=1,20
138      CALL PLOT(X1+I*.5,Y1,2)
139      CALL PLOT(X1+I*.5,Y1+.1,2)
140 30    CALL PLOT(X1+I*.5,Y1,2)
141      DO 31 I=1,20
142      CALL PLOT(X2,Y1+I*.5,2)
143      CALL PLOT(X2-.1,Y1+I*.5,2)
144 31    CALL PLOT(X2,Y1+I*.5,2)
145      DO 32 I=1,20
146      CALL PLOT(X2-I*.5,Y2,2)
147      CALL PLOT(X2-I*.5,Y2-.1,2)
148 32    CALL PLOT(X2-I*.5,Y2,2)
149      CALL NUMBER(0.,Y1+10.9,.21,-1.,-90.,1)
150      CALL NUMBER(2.5,Y1+10.9,.21,-.5,-90.,1)
151      CALL NUMBER(5.,Y1+10.4,.21,0.,-90.,-1)
152      CALL NUMBER(7.5,Y1+10.7,.21,.5,-90.,1)
153      CALL NUMBER(10.,Y1+10.7,.21,1.,-90.,1)
154      CALL SYMBOL(2.4,Y1+11.1,.28,'PRINCIPAL COMPONENT 3',0.,21)
155      DO 34 I=1,NS
156      IF(JANS.EQ.1HP.OR.JANS.EQ.1HB) THEN
157      DO 33 J=1,MS(I)
158      ZZ=(Z(J,I)-XMIN)/SCALE
159      YY=(Y(J,I)-YMIN)/SCALE
160      IF(NUMB.EQ.1HN) THEN
161      CALL SYMBOL(ZZ,YY,.07,1,0.,-1)
162      ELSE
163      IF(NS.EQ.1) CALL NUMBER(ZZ,YY,.07,FLOAT(J),-90.,-1)
164      IF(NS.GT.1) CALL NUMBER(ZZ,YY,.07,FLOAT(I),-90.,-1)
165      END IF
166 33    CONTINUE
167      END IF
168 34 IF(JANS.EQ.1HE.OR.JANS.EQ.1HB)
169 +   CALL ELLIPSE(XD(2,I),YD(2,I),AA(2,I),BB(2,I),COT(2,I),SIT(2,I),
170 +   XMIN,YMIN,SCALE,XMAX,YMAX)
171      CALL PLOT(0.,0.,999)
172      TYPE *, 'PLOT SPOOLED ON FILENAME=PLOT.PLT !!!'
173      STOP
174      END

```

```

1  C <FLIP> TRANSPOSES A MATRIX
2  C
3  C MAXIMUM ROWS * COLUMNS = 10,000
4  C
5      DIMENSION X(10000)
6      CHARACTER FILENAME*13
7      TYPE 1
8      1 FORMAT(// ' <FLIP> TRANSPOSES A MATRIX (MAX. ROWS * '
9      + ' COLUMNS = 10,000'// ' >FILENAME FOR INPUT MATRIX ?')
10     ACCEPT *,FILENAME
11     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
12     TYPE *, ' >NUMBER OF COLUMNS IN MATRIX ?'
13     ACCEPT *,NC
14     NR=0
15     2 IC=NC*NR
16     READ(1,*,END=3) (X(I),I=IC+1,IC+NC)
17     NR=NR+1
18     GO TO 2
19     3 TYPE 4,NR
20     4 FORMAT(/I5 ' ROWS IN INPUT MATRIX'//)
21     OPEN(UNIT=2,NAME='FLIP.TMP')
22     DO 5 J=1,NC
23     5 WRITE(2,*) (X(I),I=J,NC*NR,NC)
24     TYPE *, 'RESULTS STORED IN FILE=FLIP.TMP !!!'
25     STOP
26     END

```



```

1  C <HISTOGRAM>
2  C
3  C REQUIRES SUBROUTINES SYMBOLDFT, CALSPDOL/LIB
4  C
5      DIMENSION TITLE1(5),TITLE2(5),X(5000),Y(50),Z(50)
6      PARAMETER NMAX=5000
7      CHARACTER FILENAME*13
8      TYPE 1
9      1 FORMAT(// ' <HISTOGRAM> IS HERE !!'// ' >LABEL FOR VARIABLE AXIS ?'
10     + ' (UP TO 20 CHARACTERS)')
11     ACCEPT 2, TITLE1
12     2 FORMAT(5A4)
13     TYPE *, '>LABEL FOR SAMPLE SET ?'
14     ACCEPT 2, TITLE2
15     TYPE *, '>FILENAME FOR INPUT ? "DEPTH, AGE, DATA"'
16     ACCEPT *, FILENAME
17     OPEN(UNIT=1, NAME=FILENAME, TYPE='OLD')
18     TYPE *, '>HOW MANY VARIABLES ASSOCIATED WITH EACH SAMPLE ?'
19     ACCEPT *, IVAR
20     TYPE *, '>PLOT WHICH OF THESE ? ENTER ITS SUBSCRIPT ?'
21     ACCEPT *, JVAR
22     DO 5 J=1, NMAX
23     READ(1, *, END=5) D, A, (Z(I), I=1, IVAR)
24     5 X(J)=Z(JVAR)
25     6 NS=J-1
26     TYPE 7, NS
27     7 FORMAT(/I6 ' VALUES COUNTED !'//)
28     CLOSE(UNIT=1)
29     XMIN=X(1)
30     XMAX=X(1)
31     DO 10 I=1, NS
32     IF(X(I).LT.XMIN) XMIN=X(I)
33     10 IF(X(I).GT.XMAX) XMAX=X(I)
34     TYPE 12, XMIN, XMAX
35     12 FORMAT(/ ' MINIMUM VALUE = 'F10.3/ ' MAXIMUM VALUE = 'F10.3/
36     + ' >ORIGIN, CLASS INTERVAL (PER INCH), & MAXIMUM UPPER LIMIT ?')
37     ACCEPT *, XD, XI, XM
38     TYPE *, '>NUMBER OF COUNTS PER INCH OF PLOT ?'
39     ACCEPT *, YI
40     NC=INT(((XM-XD)/XI)+.5)
41     DO 13 I=1, NS
42     DO 13 J=1, NC
43     IF(XI.GE.0) THEN
44         IF(X(I).GT.(XD+(J-1)*XI).AND.X(I).LT.(XD+J*XI)) Y(J)=Y(J)+1
45     ELSE
46         IF(X(I).LT.(XD+(J-1)*XI).AND.X(I).GT.(XD+J*XI)) Y(J)=Y(J)+1
47     END IF
48     13 IF(X(I).EQ.(XD+(J-1)*XI).OR.X(I).EQ.(XD+J*XI)) Y(J)=Y(J)+.5
49     YMAX=Y(1)
50     DO 14 J=2, NC
51     14 IF(Y(J).GT.YMAX) YMAX=Y(J)
52     TYPE *, '>ALTER SIZE OF PLOT ? INPUT SCALE FACTOR'
53     ACCEPT *, SIZE
54     C PLOT
55     CALL ASSIGN(9, 'PLOT.PLT')
56     CALL PLOTS(0,0,9)
57     IF(SIZE.GT.0.) CALL FACTOR(SIZE)
58     CALL SYMBOL(NC/2-2.1,0.,.21,TITLE1,0.,20)
59     DO 20 I=1, NC+1
60     IF(ABS(XI).LT.1) THEN

```

```
52      ELSE
53      CALL NUMBER(-.07+(I-1),.31,.14,X0+(I-1)*XI,0.,-1)
54      END IF
55  20 CONTINUE
56      CALL PLOT(0.,.5,-3)
57      DO 21 I=1,NC
58      CALL PLOT(FLOAT(I-1),0.,3)
59      CALL PLOT(FLOAT(I-1),Y(I)/YI,2)
70      CALL PLOT(FLOAT(I),Y(I)/YI,2)
71  21 CALL PLOT(FLOAT(I),0.,2)
72      CALL PLOT(0.,0.,2)
73      CALL SYMBOL(0.,(YMAX/YI)+2.,.21,TITLE2,0.,20)
74      CALL SYMBOL(0.,(YMAX/YI)+1.5,.21,'N =',0.,3)
75      CALL NUMBER(.84,(YMAX/YI)+1.5,.21,FLOAT(NS),0.,-1)
76      CALL PLOT(0.,0.,999)
77      TYPE *, 'PLOT IS SPOOLED ON FILE=PLT.PLT !!'
78      STOP
79      END
```



```

1  C <ISOTOPE> COMPUTES AND PLOTS EQUAL FREQUENCY ELLIPSES DESCRIBING THE
2  C   COVARIATION BETWEEN OXYGEN AND CARBON ISOTOPES
3  C
4  C   REQ. SUBROUTINES  DIAG, CONFIDAXIS, ELLIPSE, SYMBOLDFT, CALSPPOOL/LIB
5  C
6      DIMENSION D(4),V(2),U(4),SIT(50),COT(50)
7      DIMENSION XD(50),YD(50),AA(50),BB(50)
8      DIMENSION X(100,50),Y(100,50),MS(50)
9      CHARACTER FILENAME*13
10     PARAMETER XMIN=-2.,YMIN=-2.,SCALE=.4,XMAX=5.,YMAX=2.
11     PARAMETER NDIM=100,NSET=50
12     TYPE *, '<ISOTOPE> PLOTS COVARIATION ELLIPSE FOR C AND O ISOTOPES'
13     OPEN(UNIT=2,NAME='PRINT.TMP')
14     TYPE *, '>PLOT DATA POINTS(P), ELLIPSES(E), OR BOTH(B)'
15 1  ACCEPT 2,JANS
16 2  FORMAT(A1)
17     IF(JANS.NE.1HP.AND.JANS.NE.1HE.AND.JANS.NE.1HB) GO TO 1
18     DD 14 NS=1,NSET
19     TYPE *, '>NAME OF FILE CONTAINING OXYGEN AND CARBON ISOTOPE DATA ?'
20     ACCEPT *,FILENAME
21     IF(JANS.EQ.1HP) WRITE(2,3) FILENAME
22     IF(JANS.EQ.1HE) WRITE(2,4) FILENAME
23     IF(JANS.EQ.1HB) WRITE(2,5) FILENAME
24 3  FORMAT(///' SAMPLES FROM INPUT FILE ['A13']/' PLOTTED '
25     + ' INDIVIDUALLY')
26 4  FORMAT(///' SAMPLES FROM INPUT FILE ['A13']/' PLOTTED AS AN '
27     + ' ELLIPSE')
28 5  FORMAT(///' SAMPLES FROM INPUT FILE ['A13']/' PLOTTED INDIVIDUA
29     + ' LLY AND AS AN ELLIPSE')
30     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
31     DD 6 M=1,NDIM
32 6  READ(1,*,END=7) DUM,DUM,X(M,NS),Y(M,NS)
33 7  M=M+1
34     MS(NS)=M
35     CLOSE(UNIT=1)
36     WRITE(2,8) NS,M
37 8  FORMAT(' SET NO. 'I4' CONTAINS 'I4' SAMPLES')
38     IF(JANS.EQ.1HE.OR.JANS.EQ.1HB) THEN
39         TYPE *, 'CONFIDENCE LEVEL FOR RANGE SLOPE OF MAJOR AXIS (%) ?'
40         ACCEPT *,CL
41         ASUM=0.
42         BSUM=0.
43         D(1)=0.
44         D(2)=0.
45         D(3)=0.
46         D(4)=0.
47         DD 10 I=1,M
48         B=Y(I,NS)
49         A=X(I,NS)
50         ASUM=ASUM+A
51         BSUM=BSUM+B
52         D(1)=D(1)+A*A
53         D(2)=D(2)+A*B
54         D(3)=D(3)+B*A
55 10     D(4)=D(4)+B*B
56         XD(NS)=ASUM/M
57         YD(NS)=BSUM/M
58         D(1)=(D(1)-ASUM*ASUM/M)/M
59         D(2)=(D(2)-ASUM*BSUM/M)/M
60         D(3)=(D(3)-BSUM*ASUM/M)/M

```

```

52      CALL DIAG(D,V,U)
53      AA(NS)=SQRT(V(1))
54      BB(NS)=SQRT(V(2))
55      COT(NS)=U(1)
56      SIT(NS)=U(2)
57      CALL CONFDAXIS(M,2*AA(NS),2*BB(NS),U(1),U(2),CL,SN,S, SX)
58      WRITE(2,12) U,XJ(NS),YQ(NS),2*AA(NS),2*BB(NS),U(1),U(2),
59      *      S,INT(CL),SN,SX
70      12  FORMAT(/5X'DISPERSION MATRIX'
71      *      6X,2E12.3/29X,2E12.3/6X'ELLIPSE CENTERED AT'15X,2F12.3/6X
72      *      'LENGT-HS OF PRINCIPAL AXES'9X,2F12.3/6X'DIRECTION SIN & COS '
73      *      'OF MAJOR AXIS '2F12.3/6X'SLOPE'29X,F12.3/
74      *      6X'MIN & MAX SLOPES ('13'% CONF.LEV.) '2F12.3)
75      END IF
76      TYPE #,'>PLOT ANOTHER SET OF SAMPLES?'
77      13  ACCEPT 2, IANS
78      IF(IANS.NE.1HY.AND.IANS.NE.1HY) GO TO 13
79      14  IF(IANS.EQ.14N) GO TO 20
80      20  TYPE #,'>INPUT SCALE FACTOR TO ALTER SIZE OF PLOT (USJALLY=1) !'
81      ACCEPT *,SIZE
82      C PLOT
83      CALL ASSIGN(9,'PLDT.PLT')
84      CALL PLOTS(0,0,9)
85      IF(SIZE.GT.0.) CALL FACTOR(SIZE)
86      CALL PLOT(0.,0.,-3)
87      X2=(XMAX-XMIN)/SCALE
88      Y2=(YMAX-YMIN)/SCALE
89      DO 30 I=1,20
90      CALL PLOT(I*.5,Y1,2)
91      CALL PLOT(I*.5,Y1+.1,2)
92      30  CALL PLOT(I*.5,Y1,2)
93      DO 31 I=1,20
94      CALL PLOT(X2,I*.5,2)
95      CALL PLOT(X2-.1,I*.5,2)
96      31  CALL PLOT(X2,I*.5,2)
97      DO 32 I=1,20
98      CALL PLOT(X2-I*.5,Y2,2)
99      CALL PLOT(X2-I*.5,Y2-.1,2)
100     32  CALL PLOT(X2-I*.5,Y2,2)
101     DO 24 I=1,20
102     CALL PLOT(0.,Y2-I*.5,2)
103     CALL PLOT(.1,Y2-I*.5,2)
104     24  CALL PLOT(0.,Y2-I*.5,2)
105     CALL NUMBER(0.,10.7,.21,2.,-90.,-1)
106     CALL NUMBER(2.5,10.5,.21,3.,-90.,-1)
107     CALL NUMBER(5.,10.4,.21,4.,-90.,-1)
108     CALL NUMBER(7.5,10.4,.21,5.,-90.,-1)
109     CALL NUMBER(10.,10.5,.21,6.,-90.,-1)
110     CALL SYMBOL(2.8,10.6,.21,'OXYGEN-18/OXYGEN-16',0.,19)
111     CALL NUMBER(-.3,.25,.21,-2.,-90.,-1)
112     CALL NUMBER(-.3,2.7,.21,-1.,-90.,-1)
113     CALL NUMBER(-.3,5.1,.21,0.,-90.,-1)
114     CALL NUMBER(-.3,7.6,.21,1.,-90.,-1)
115     CALL NUMBER(-.3,10.25,.21,2.,-90.,-1)
116     CALL SYMBOL(-.6,7.2,.21,'CARBON-13/CARBON-12',-90.,19)
117     DO 34 I=1,NS
118     IF(JANS.EQ.1HP.OR.JANS.EQ.1HB) THEN
119     DO 33 J=1,MS(I)
120     XX=(X(J,I)-XMIN)/SCALE
121     YY=(Y(J,I)-YMIN)/SCALE

```



```
123      33  IF(NS.GT.1) CALL NUMBER(XX,YY,.07,FLOAT(I),-90.,-1)
124          END IF
125      34  IF(JANS.EQ.1HE.OR.JANS.EQ.1H8)
126          +  CALL ELLIPSE(XO(I),YO(I),AA(I),BB(I),COT(I),SIT(I),XMIN,YMIN,
127          +  SCALE,XMAX,YMAX)
128          CALL PLOT(0.,0.,999)
129          TYPE *, 'PLOT IS SPOOLED ON FILE=PLT.PLT !!'
130          STOP
131          END
```

```

1  C <LINTERP> LINEAR INTERPOLATOR (AFTER DAVIS, 1973)
2  C
3  C   EQUALLY SPACES DATA BY LINEAR INTERPOLATION. DATA SEQUENCE XIN IS
4  C   CONVERTED TO POINTS EQUALLY SPACED XI APART, FROM INITIAL POINT XB
5  C   TO FINAL POINT XL. THE EQUALLY SPACED DATA SEQUENCE IS XOUT.
6  C
7  C   XIN(JS,1) & XIN(JS,2) ARE X & Y COORDINATES IN INPUT ARRAY
8  C   XOUT(IS,1) & XOUT(IS,2) ARE X & Y COORDINATES IN OUTPUT ARRAY
9  C   X IS THE X-COORDINATE OF THE NEXT DATA POINT TO BE INTERPOLATED. THIS
10 C   VALUE IS INCREMENTED BY XI.
11 C
12   DIMENSION XIN(1000,2),DATA(100),XOUT(3000,2)
13   CHARACTER FILENAME*13
14   TYPE 12
15   12 FORMAT(//' <LINTERP> EQUALLY SPACES DATA BY LINEAR INTERPOLATION'/
16   + '/' >FILENAME FOR INPUT DATA (DUMMY, AGE, ASSOCIATED VARIABLES)?')
17   ACCEPT *,FILENAME
18   OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
19   TYPE *, '>INPUT INITIAL-X, FINAL-X, & INCREMENT FOR SPACED DATA'
20   ACCEPT *,XB,XL,XI
21   TYPE *, '>NUMBER OF ASSOCIATED VARIABLES ?'
22   ACCEPT *,NVAR
23   TYPE *, '>SUBSCRIPT OF ASSOCIATED VARIABLE TO BE INTERPOLATED ?'
24   ACCEPT *,NSUB
25   NIN=1
26   13 READ(1,*,END=14) DUMMY,XIN(NIN,1),(DATA(J),J=1,NVAR)
27   XIN(NIN,2)=DATA(NSUB)
28   NIN=NIN+1
29   GO TO 13
30   14 NIN=NIN-1
31  C INITIALIZE INDEXING PARAMETERS AND X-COORDINATE OF FIRST DATA POINT
32  C TO INTERPOLATE
33   IS=0
34   JS=1
35   X=XB-XI
36   1 X=X+XI
37  C IF THE VARIABLE X IS GREATER THAN THE FINAL POINT XL, STOP
38   IF(X-XL) 2,2,99
39   2 IS=IS+1
40   XOUT(IS,1)=X
41  C IF THE VARIABLE X IS LESS THAN THE FIRST X VALUE IN INPUT DATA, THE
42  C FIRST Y VALUE IN XIN IS ASSIGNED TO XOUT(IS,2)
43   IF(X-XIN(1,1)) 3,3,4
44   3 XOUT(IS,2)=XIN(1,2)
45   GO TO 1
46  C IF THE VARIABLE X IS GREATER THAN THE LAST Y VALUE IN THE INPUT DATA,
47  C THAT LAST Y VALUE IS ASSIGNED TO XOUT(IS,2)
48   4 IF(X-XIN(NIN,1)) 6,5,5
49   5 XOUT(IS,2)=XIN(NIN,2)
50   GO TO 1
51  C FIND THE TWO DATA POINTS OF XIN SUCH THAT XOUT(IS,1) LIES IN THE
52  C INTERVAL (XIN(JS,1),XIN(JS+1,1)). USE THE Y VALUES OF THESE TWO
53  C DATA POINTS TO INTERPOLATE THE VALUE OF XOUT(IS,2)
54   6 IF(X-XIN(JS,1)) 9,8,7
55   7 IF(X-XIN(JS+1,1)) 8,8,9
56   8 XOUT(IS,2)=XIN(JS,2)+(XIN(JS+1,2)-XIN(JS,2))*(X-XIN(JS,1))/
57   + (XIN(JS+1,1)-XIN(JS,1))
58   GO TO 1
59   9 JS=JS+1
60   IF(JS-NIN) 6,10,10

```



```
52      GO TO 5
53  C OUTPUT THE EQUALLY SPACED POINTS
54      99 OPEN(UNIT=2,NAME='LINTERP.TMP')
55      DO 11 I=1,IS
56      11 WRITE(2,*) XOUT(I,1),XOUT(I,1),XOUT(I,2)
57      TYPE *,IS,' EQUALLY SPACED POINTS IN INTERPOLATED DATA SEQUENCE !'
58      TYPE *,'RESULTS ARE ON FILE=LINTERP.TMP !!'
59      STOP
70      END
```

```

1  C <MULTIPLY> - <MULTIPLY> CALLING ROUTINE
2  C
3  C     REQUIRES SUBR. MULTIPLY
4  C
5      DOUBLE PRECISION IDCOL1,IDROW1,IDCOL2,IDROW2
6      CHARACTER FILENAME*13
7      DIMENSION X(100),Y(100,20),Z(20)
8      PARAMETER IDIM=1,JDIM=100,KDIM=10
9      TYPE 1
10     1 FORMAT(// ' <MULTIPLY> TWO MATRICES' //)
11     TYPE *, '>FILENAME FOR YOUR FIRST MATRIX?'
12     ACCEPT *,FILENAME
13     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
14     TYPE *, '>COLUMNS ARE THE...? (A8)'
15     ACCEPT 2,IDCOL1
16     2 FORMAT(A8)
17     TYPE *, '>HOW MANY COLUMNS? (MAX IS 100)'
18     ACCEPT *,NCOL1
19     TYPE *, '>ROWS ARE THE...? (A8)'
20     ACCEPT 2,IDROW1
21     TYPE *, '>ARE ROWS OF 1ST MATRIX ASSOCIATED WITH DEPTH,AGE,TOTAL ?'
22     ACCEPT 3, IANS
23     3 FORMAT(A1)
24     TYPE *, '>FILENAME FOR YOUR 2ND MATRIX?'
25     ACCEPT *,FILENAME
26     OPEN(UNIT=2,NAME=FILENAME,TYPE='OLD')
27     TYPE *, '>COLUMNS ARE THE...? (A8)'
28     ACCEPT 2,IDCOL2
29     TYPE *, '>HOW MANY COLUMNS? (MAX IS 20)'
30     ACCEPT *,NCOL2
31     TYPE *, '>ROWS ARE THE...? (A8)'
32     ACCEPT 2,IDROW2
33     N2=1
34     4 READ(2,*,END=5) (Y(N2,M),M=1,NCOL2)
35     N2=N2+1
36     GO TO 4
37     5 N2=N2-1
38     TYPE 6,N2,IDROW2
39     6 FORMAT(// ' THERE ARE',I4,1X,A8,' OR ROWS IN THE 2ND MATRIX' //)
40     IF(N2.NE.NCOL1) TYPE 7,IDCOL1,IDROW2
41     7 FORMAT(// ' NO. OF ',A8,' IN MATRIX 1 .NE. TO ',A8,' IN MATRIX 2')
42     TYPE 9,IDROW1,IDCOL2
43     9 FORMAT(// ' PRODUCT OF TWO MATRICES' //
44     + ' ROWS = ',A8,' COLUMNS = ',A8//)
45     OPEN(UNIT=3,NAME='MULT.TMP')
46     N1=0
47     10 IF(IANS.EQ.1H4) READ(1,*,END=11) (X(J),J=1,NCOL1)
48     IF(IANS.EQ.1HY) READ(1,*,END=11) A,B,C,(X(J),J=1,NCOL1)
49     N1=N1+1
50     CALL MULTIPLY(X,1,1,NCOL1,JDIM,Y,NCOL2,KDIM,Z)
51     IF(IANS.EQ.1HY) WRITE(3,*) A,B,C,(Z(K),K=1,NCOL2)
52     IF(IANS.EQ.1HY) WRITE(3,*) (Z(K),K=1,NCOL2)
53     GO TO 10
54     11 TYPE 12, N1,IDROW1
55     12 FORMAT(// 'THERE ARE',I5,1X,A8,' OR ROWS IN THE PRODUCT MATRIX' //
56     + ' RESULTS ARE ON FILE=MULT.TMP !!')
57     STOP
58     END

```



```

1  C <NANNO> CREATES MIOCENE NANNOPLANKTON CENSUS DATA FILES &
2  C      COMPUTES THE 1ST & 2ND Q-MODE PRINCIPAL COMPONENTS
3  C
4  C  REQUIRES SUBROUTINES PERCENT, NORMALIZE, ASCEND, QMATRIX, HOW,
5  C      QSCORES, PRINCO, XYPLOT, TSPLIT
6  C
7      DIMENSION X(100,25),Q(100,100),DEPTH(100),TOTAL(100)
8      DIMENSION E(100),V(100,100),A(100),B(100),C(100),D(100)
9      DIMENSION F(25,100),Y(25),IX(25),IY(25),IZ(25),INDX(100)
10     CHARACTER FILENAME*13,TODAY*9
11     PARAMETER NDIM=100,MDIM=25
12     CALL DATE(TODAY)
13     NS=0
14     TYPE 1
15     1 FORMAT(///' THIS IS <NANNO> !'///' <NANNO> CREATES OR UPDATES'
16     + '  NANNOFOSSIL CENSUS DATA FILES'//11X'IT ALSO COMPUTES 1ST & 2ND'
17     + '  Q-MODE PRINCIPAL COMPONENTS '///// ' >BEFORE YOU '
18     + ' INPUT NEW DATA, IS THERE ALREADY DATA ON FILE FOR '
19     + ' THIS SITE?'///' ANSWER Y OR N')
20     ACCEPT 2,IANS
21     2 FORMAT(A1)
22     IF(IANS.EQ.14Y) THEN
23         TYPE *, '>WHATS ITS FILENAME?'
24         ACCEPT *,FILENAME
25         OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
26         DO 3 N=1,NDIM
27     3   READ(1,*,END=4) DEPTH(N),DUM,TOTAL(N),(X(N,I),I=1,MDIM)
28     4   NS=N-1
29         TYPE *, '>WILL YOU BE ADDING ANY NEW DATA TO THIS FILE?'
30         ACCEPT 2,KANS
31         IF(KANS.EQ.14N) THEN
32             MS=NS
33             GO TO 19
34         END IF
35     ELSE
36         TYPE 5
37     5   FORMAT(' >THEN GIVE YOUR NEW DATA FILE A NAME (UP TO 9 '
38     + '  "CHARACTER.PNT")')
39         ACCEPT *,FILENAME
40         N=1
41         END IF
42  C INPUT CENSUS DATA
43     TYPE 6
44     6 FORMAT(/' >TYPE IN YOUR NEW NANNO CENSUS DATA:'//5X'>FIRST, LABEL'
45     + '  EACH SAMPLE WITH ITS SUBBOTTOM DEPTH (IN METERS)'//5X'>THEN,'
46     + '  TYPE IN COUNTS FOR THE 25 NANNO TAXA, EACH COUNT SEPARATED '
47     + '  BY A SPACE'///)
48     DO 9 M=N,NDIM
49     7   READ(5,*) DEPTH(M),Y
50         TYPE *, '>CHECK THE CENSUS DATA FOR THIS SAMPLE. IS IT CORRECT?'
51         ACCEPT 2,JANS
52         IF(JANS.EQ.14N) GO TO 7
53         CALL PERCENT(Y,MDIM,T)
54         TOTAL(M)=T
55         DO 8 I=1,MDIM
56     8   X(M,I)=Y(I)
57         TYPE *, '>ANY MORE SAMPLES FOR THIS SITE?'
58         ACCEPT 2,JANS
59         IF(JANS.EQ.14N) GO TO 10
60     9   TYPE *, '>THEN INPUT THE DEPTH AND COUNTS FOR YOUR NEXT SAMPLE'

```

```

52      TYPE 15, NS, MS-NS
53      15 FORMAT(///110' OLD SAMPLES;'110' NEW SAMPLES'///)
54      C SORT SAMPLES BY DEPTH, THEN OUTPUT TO DISK FILE
55      19 CALL ASCEND(DEPTH, INDX, MS)
56      DO 16 I=1, MS
57      A(I)=DEPTH(INDX(I))
58      B(I)=TOTAL(INDX(I))
59      DO 16 J=1, MDIM
60      16 Q(I, J)=X(INDX(I), J)
61      DO 17 I=1, MS
62      DEPTH(I)=A(I)
63      TOTAL(I)=B(I)
64      DO 17 J=1, MDIM
65      17 X(I, J)=Q(I, J)
66      TYPE *, 'SAMPLES HAVE BEEN RESORTED BY THEIR SUB-BOTTOM DEPTHS !'
67      OPEN(UNIT=2, NAME=FILENAME)
68      DO 18 I=1, MS
69      18 WRITE(2, *) DEPTH(I), DEPTH(I), TOTAL(I), (X(I, J), J=1, MDIM)
70      TYPE *, 'CENSUS DATA HAS BEEN OUTPUT TO A DISC FILE !'
71      C PRINT CENSUS DATA EXPRESSED AS PERCENT
72      OPEN(UNIT=3, NAME='PRINT.TMP', DISPOSE='PRINT')
73      WRITE(3, 20) FILENAME, TODAY
74      20 FORMAT(1H1//6X'CENSUS DATA FILE OF MIOCENE CALCAREOUS '
75      + 'NANNOPLANKTON ['A13'] AS OF 'A9'///)
76      IF(KANS.EQ.14N) GO TO 23
77      IF(IANS.EQ.14Y) WRITE(3, 21) TODAY, FILENAME, NS, MS-NS
78      IF(IANS.EQ.14N) WRITE(3, 22) TODAY, FILENAME, MS-NS
79      21 FORMAT(6X'UPDATED 'A9' AND STORED UNDER FILENAME ['A13']'//
80      + '110' SAMPLES IN PREVIOUS DATA FILE;'15' NEW SAMPLES ADDED'///)
81      22 FORMAT(6X'CREATED 'A9' AND STORED UNDER FILENAME ['A13']'//
82      + '6X'THIS NEW DATA FILE CREATED WITH'110' SAMPLES'///)
83      23 WRITE(3, 24) (I, I=1, 22)
84      24 FORMAT(52X'NANNOPLANKTON CENSUS DATA (EXPRESSED AS PERCENT) '
85      + '//6X'NO. DEPTH TOTAL '22I4' A B C'//)
86      DO 26 K=1, MS
87      IT=NINT(TOTAL(K))
88      DO 25 L=1, MDIM
89      25 IX(L)=NINT(X(K, L))
90      26 WRITE(3, 27) K, DEPTH(K), IT, IX
91      27 FORMAT(5X, I4, F8.2, I7, 3X, 25I4)
92      TYPE *, 'CENSUS DATA HAS BEEN PRINTED !'
93      C COMPUTE Q-MATRIX, EIGENVALUES & EIGENVECTORS, Q-SCORES
94      CALL QMATRIX(X, Q, MS, MDIM, MDIM, MDIM)
95      TYPE *, 'Q-MATRIX HAS BEEN COMPUTED !'
96      CALL HOW(MS, MDIM, 2, Q, E, V, A, B, C, D)
97      TYPE *, 'EIGENVALUES & EIGENVECTORS HAVE BEEN COMPUTED !'
98      CALL QSCORES(X, V, E, MS, MDIM, MDIM, MDIM, =)
99      TYPE *, 'Q-SCORES HAVE BEEN COMPUTED !'
100     DO 30 I=1, MDIM
101     Y(I)=F(I, 1)
102     30 IX(I)=NINT(Y(I)*100.)
103     CALL PERCENT(Y, MDIM, T)
104     DO 31 I=1, MDIM
105     IY(I)=NINT(Y(I))
106     Y(I)=F(I, 2)
107     31 IZ(I)=NINT(Y(I)*100.)
108     E1=(E(1)/MS)*100.
109     E2=(E(2)/MS)*100.
110     TYPE *, 'RESULTS ARE NOW BEING PRINTED !'
111     WRITE(3, 32) FILENAME, (I, I=1, 22), IX, IY, IZ, E1, E2
112

```



```

123      + 'THE 1ST & 2ND Q-MODE EIGENVECTORS, CENOP SITE ['A13']'////8X
124      + 'EIGENVECTOR'4X,22I4'  A  B  C'//6X'1ST (NORMALIZED) ',
125      + 25I4/6X'1ST (AS PERCENT)  '25(I3'%')//6X'2ND (NORMALIZED) '
126      + 25I4//
127      + 25X'(NOTE THAT ALL COMPOSITIONS GIVEN IN NORMALIZED FORM HAVE '
128      + 'BEEN MULTIPLIED BY 100)'////6X'THE 1ST Q-MODE EIGENVECTOR '
129      + 'REPRESENTS THE "AVERAGE" SAMPLE FOR THIS DATA;'//6X'IT ALONE '
130      + 'DESCRIBES 'F5.1'% OF THE OBSERVATIONS.'//6X'THE 2ND EIGEN'
131      + 'VECTOR DESCRIBES THE PRINCIPAL DIMENSION OF DEVIATIONS IN '
132      + 'NANNOFLORAL COMPOSITIONS FROM THIS "AVERAGE";'//6X'IT ACCOUNTS'
133      + ' FOR AN ADDITIONAL 'F5.1'%')
134      C SCALE EIGENVECTORS BY SQRT(EIGENVALUES) TO GET PRINCIPAL COMPONENTS
135      CALL PRINCO(V,E,MS,NDIM)
136      WRITE(3,33) FILENAME
137      33 FORMAT(14I1//6X'1ST & 2ND Q-MODE PRINCIPAL COMPONENTS OF MIOCENE '
138      + 'NANNOPLANKTON FOR CENOP FILE ['A13']'////11X'DEPTH'7X'1ST'5X
139      + '2ND'3X'% EXPLAINED'//)
140      ISUM=0
141      DO 34 I=1,MS
142      ICOMM=NINT((V(I,1)**2+V(I,2)**2)*100.)
143      ISUM=ISUM+ICOMM
144      34 WRITE(3,35) I,DEPTH(I),V(I,1),V(I,2),ICOMM
145      35 FORMAT(I5,3X,=8.2'M'F9.2,F8.2,I10'%')
146      WRITE(3,36) ISUM/MS
147      36 FORMAT(////6X,I5'% OF THE TOTAL OBSERVED VARIANCE ACCOUNTED FOR')
148      TYPE *, 'PRINCIPAL COMPONENTS HAVE BEEN COMPUTED !'
149      C PLOT RESULTS
150      WRITE(3,40) FILENAME
151      40 FORMAT(14I1//6X'PLOT OF 1ST VS. 2ND Q-MODE PRINCIPAL COMPONENTS '
152      + 'OF NANNOPLANKTON FOR CENOP FILE ['A13']'////)
153      DO 41 I=1,MS
154      A(I)=V(I,1)
155      41 B(I)=V(I,2)
156      CALL XYPLOT(3,A,MS,3)
157      WRITE(3,42) FILENAME
158      42 FORMAT(/39X'2ND Q-MODE PRINCIPAL COMPONENT'/14I1//6X' 2ND Q-MODE '
159      + ' PRINCIPAL COMPONENT OF NANNOPLANKTON FOR CENOP FILE ['A13']'
160      + ' PLOTTED DOWN-CORE'//7X'(SUB-BOTTOM DEPTHS IN METERS LISTED ON'
161      + ' THE RIGHT)'//39X'2ND Q-MODE PRINCIPAL COMPONENT')
162      CALL TSPLIT(DEPTH,B,MS,2,3)
163      TYPE *, 'RESULTS HAVE BEEN PLOTTED !'
164      TYPE *, 'FINISHED !'
165      STOP
166      END

```

```

1  C <OMEGA> COMPUTES CARBONATE SATURATION
2  C
3  C AFTER A PROGRAM WRITTEN IN BASIC BY ROBIN KEIR
4  C
5      OPEN(UNIT=1,NAME='OMEGA.TMP')
6      TYPE *, 'THIS IS <OMEGA> !!'
7      1 TYPE 2
8      2 FORMAT(// ' >INPUT T, S, DEPTH(KM), ALK, TCO2 ')
9      ACCEPT *, T, C, Z, X, Y
10     CALL OMEGA(T, C, Z, X, Y, H, OM)
11     TYPE 4, Z, H, OM
12     4 FORMAT(/3X, 'AT', F10.3, 'KM AND', F10.3, 'MMOLE/L 1-OMEGA=', F10.3/)
13     WRITE(1, *) Z, H, OM
14     TYPE *, '>MORE ?'
15     ACCEPT S, IANS
16     5 FORMAT(A1)
17     IF (IANS.EQ.14Y) GO TO 1
18     TYPE *, 'RESULTS ARE ON FILE=OMEGA.TMP !!!'
19     STOP
20     END

```



```

1  C <PERCENT> - <PERCENT> & <NORMALIZE> CALLING ROUTINE
2  C
3  C INPUT  = RAW SAMPLE VECTORS
4  C OUTPUT = PERCENT SAMPLE VECTORS & TOTAL NUMBER INDIVIDUALS COUNTED
5  C          OR, UNIT SAMPLE VECTORS & ORIGINAL VECTOR LENGTHS
6  C
7  C REQUIRES SUBROUTINES PERCENT, NORMALIZE
8  C
9      DIMENSION FMT(20),X(1000)
10     CHARACTER FILENAME*13
11     TYPE 1
12     1 FORMAT(///' <PERCENT> OR <NORMALIZE> SAMPLE VECTORS'///
13     + ' >COMPUTE PERCENT (P) OR UNIT (U) SAMPLE VECTORS?')
14     ACCEPT 2, IANS
15     2 FORMAT(A1)
16     TYPE *, '>WHATS THE FILENAME OF YOUR DATA?'
17     ACCEPT *, FILENAME
18     OPEN(UNIT=1, NAME=FILENAME, TYPE='OLD')
19     IF(IANS.EQ.1HJ) THEN
20         TYPE *, ' *-FORMAT ASSUMED : (DEPTH, AGE, TOTAL, CENSUS DATA)'
21         ELSE
22         TYPE *, '>DATA FORMAT? (DEPTH, AGE, CENSUS DATA) '
23         ACCEPT 3, FMT
24     3  FORMAT(20A4)
25     END IF
26     TYPE *, '>NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR?'
27     ACCEPT *, NV
28     TYPE *, '>GIVE RESULTS A FILENAME.'
29     ACCEPT *, FILENAME
30     OPEN(UNIT=2, NAME=FILENAME)
31     N=0
32     4  IF(IANS.EQ.1HP) READ(1, FMT, END=5) D, A, (X(I), I=1, NV)
33     IF(IANS.EQ.1HJ) READ(1, *, END=5) D, A, T, (X(I), I=1, NV)
34     N=N+1
35     IF(IANS.EQ.1HP) CALL PERCENT(X, NV, T)
36     IF(IANS.EQ.1HJ) CALL NORMALIZE(X, NV, T)
37     WRITE(2, *) D, A, T, (X(I), I=1, NV)
38     GO TO 4
39     5  TYPE 6, N
40     6  FORMAT(///16'  SAMPLES COUNTED'///)
41     STOP
42     END

```

```

1  C <QPCA> COMPUTES A Q-MODE PRINCIPAL COMPONENTS ANALYSIS (LOADINGS &
2  C      SCORES) AND COMPUTES A VARIMAX ROTATION
3  C
4  C REQUIRES SUBROUTINES NORMALIZE, QMATRIX, HOW, QSCORES, PRINCO, ROTATE,
5  C      CDEFF, MATINV
6  C
7      DIMENSION X(100,100),Q(100,100),XID(100),AGE(100),TOTAL(100)
8      DIMENSION E(100),V(100,100),A(100),B(100),C(100),D(100),F(100,100)
9      CHARACTER FILENAME*13,TODAY*9
10     PARAMETER NDIM=100,MDIM=100
11     TYPE 1
12     1 FORMAT(///' THIS IS <QPCA> !'///' IT COMPUTES A Q-MODE P.C.A. AND A
13     + ' VARIMAX ROTATION'///' >WHAT'S THE FILENAME FOR THE DATA TO BE '
14     + ' ANALYZED?'/' "SAMPLE DEPTH(M), AGE(M.Y.), TOTAL COUNTED, '
15     + ' CENSUS DATA AS PERCENT"'')
16     ACCEPT *,FILENAME
17     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
18     OPEN(UNIT=3,NAME='PRINT.TMP')
19     CALL DATE(TODAY)
20     TYPE *, '>HOW MANY ELEMENTS IN EACH SAMPLE VECTOR?'
21     ACCEPT *,NV
22     WRITE(3,2) FILENAME,TODAY,NV
23     2 FORMAT(1H1'Q-MODE P.C.A. OF DATA STORED IN FILE ['A13'] '
24     + ' COMPUTED 'A9//I10' ELEMENTS OF CENSUS DATA SPECIFIED FOR EACH '
25     + ' SAMPLE VECTOR')
26     DO 3 N=1,NDIM
27     3 READ(1,*,END=4) XID(N),AGE(N),TOTAL(N),(X(N,J),J=1,NV)
28     4 NS=N-1
29     TYPE *, 'CENSUS DATA HAS BEEN INPUT !'
30     WRITE(3,5) NS,FILENAME
31     5 FORMAT(110' SAMPLE VECTORS COUNTED'//4X
32     + ' NO. DEPTH (M)   AGE (M.Y.)   TOTAL CENSUS DATA INPUT AS '
33     + ' PERCENT FROM FILE ['A13']'//)
34     DO 6 N=1,NS
35     6 WRITE(3,7) N,XID(N),AGE(N),NINT(TOTAL(N)),(NINT(X(N,J)),J=1,NV)
36     7 FORMAT(16,2F11.3,110,2X,20I4/(40X,20I4))
37     CALL QMATRIX(X,Q,NS,NV,NDIM,MDIM)
38     TYPE *, 'Q-MATRIX HAS BEEN COMPUTED !'
39     TRACE=0.
40     DO 10 J=1,NS
41     10 TRACE=TRACE+Q(J,J)
42     TYPE 11,TRACE
43     11 FORMAT(///' TRACE = 'F10.1//)
44     TYPE *, '>HOW MANY EIGENVECTORS DO YOU WANT EXTRACTED?'
45     ACCEPT *,NE
46     CALL HOW(NS,NDIM,NE,Q,E,V,A,B,C,D)
47     TYPE *, 'EIGENVALUES AND EIGENVECTORS HAVE BEEN COMPUTED !'
48     WRITE(3,12) FILENAME,TODAY
49     12 FORMAT(1H1'Q-MODE PRINCIPAL COMPONENTS ANALYSIS OF DATA IN FILE '
50     + ' ['A13']'///' EIGENVALUES AND SCORES FOR Q-MODE EIGENVECTORS, '
51     + ' COMPUTED 'A9//I10' EIGENVALUE % VARIANCE   CUM.VAR.'//)
52     TYPE *, '
53     EIGENVALUE % VARIANCE   CUM.VAR.'
54     CUM=0.
55     DO 13 I=1,NE
56     13 VAR=(E(I)/TRACE)*100.
57     CUM=CUM+VAR
58     WRITE(3,14) I,E(I),VAR,CUM
59     14 FORMAT(17,F12.2,2F12.1)
60     TYPE *, '>HOW MANY EIGENVECTORS DO YOU WANT TO SAVE AND ROTATE?'

```



```

52 CALL QSCORES(X,V,E,NS,NV,NDIM,MDIM,F)
53 TYPE *, 'QSCORES HAVE BEEN COMPUTED & ARE ON FILE=QSCORES.TMP !!'
54 OPEN(UNIT=2,NAME='QSCORES.TMP')
55 WRITE(3,20) FILENAME
56 20 FORMAT(///' SCORES OF CENSUS VARIABLES ON Q-MODE PRINCIPAL '
57 * 'COMPONENTS'/' FOR DATA ON FILE ['A13']'//2X
58 * 'VAR.NO. PRINCIPAL COMPONENTS SCORES (* 100)'/)
59 DO 21 I=1,NV
60 WRITE(2,*) FLOAT(I),FLOAT(I),(F(I,J),J=1,ME)
61 21 WRITE(3,23) I,((F(I,J)*100.),J=1,ME)
62 23 FORMAT(I7,10F10.1/(7X,10F10.1))
63 TYPE *, 'SCORES HAVE BEEN PRINTED !'
64 CALL PRINCU(V,E,NS,NDIM)
65 TYPE *, 'EIGENVECTORS HAVE BEEN SCALED TO PRINCIPAL COMPONENTS !'
66 TYPE *, ' THEY ARE STORED ON FILE=QLOADINGS.TMP !'
67 OPEN(UNIT=4,NAME='QLOADINGS.TMP')
68 WRITE(3,30) FILENAME,TODAY
69 30 FORMAT(141'EIGENVECTORS OF Q-MATRIX SCALED BY EIGENVALUES TO '
70 * 'Q-MODE PRINCIPAL COMPONENTS'/' FOR DATA ON FILE ['A13
71 * ']' ON 'A9'/' NO. DEPTH (M) AGE (M.Y.) % EXPLAINED '
72 * 'LOADINGS ON Q-MODE PRINCIPAL COMPONENTS (* 100)'/)
73 TOTVAR=0.
74 DO 32 I=1,NS
75 COMM=0.
76 DO 31 J=1,ME
77 31 COMM=COMM+V(I,J)*V(I,J)
78 TOTVAR=TOTVAR+COMM
79 WRITE(4,*) XID(I),AGE(I),(V(I,J),J=1,ME)
80 ICOMM=NINT(COMM*100.)
81 32 WRITE(3,33) I,XID(I),AGE(I),ICOMM,((V(I,J)*100.),J=1,ME)
82 33 FORMAT(I6,2F11.3,113,10F10.1/(41X,10F10.1))
83 WRITE(3,34) NINT((TOTVAR*100.)/NS)
84 34 FORMAT(//3X,I5'% OF THE TOTAL OBSERVED VARIANCE ACCOUNTED FOR')
85 C ROTATE PRINCIPAL COMPONENTS
86 CALL ROTATE(ME,NS,0,V,NDIM)
87 TYPE *, 'PRINCIPAL COMPONENTS ROTATED TO VARIMAX FACTORS !'
88 TYPE *, ' THEY ARE STORED ON FILE=QVMXLOAD.TMP !'
89 OPEN(UNIT=5,NAME='QVMXLOAD.TMP')
90 WRITE(3,39) FILENAME,TODAY
91 39 FORMAT(141'Q-MODE PRINCIPAL COMPONENTS ROTATED TO VARIMAX FACTORS'
92 * '/' FOR DATA ON FILE ['A13'] ON 'A9'/' NO. DEPTH (M) AGE '
93 * '(M.Y.) % EXPLAINED LOADINGS ON Q-MODE VARIMAX FACTORS '
94 * '(* 100)'/)
95 DO 40 I=1,ME
96 E(I)=0.
97 DO 42 I=1,NS
98 COMM=0.
99 DO 41 J=1,ME
100 E(J)=E(J)+V(I,J)*V(I,J)
101 41 COMM=COMM+V(I,J)*V(I,J)
102 WRITE(5,*) XID(I),AGE(I),(V(I,J),J=1,ME)
103 ICOMM=NINT(COMM*100.)
104 42 WRITE(3,33) I,XID(I),AGE(I),ICOMM,((V(I,J)*100.),J=1,ME)
105 WRITE(3,43) (NINT((E(I)/NS)*100.),I=1,ME)
106 43 FORMAT(//' VARIANCE ACCOUNTED FOR'11X,10(I9'%'))
107 C COMPUTE FACTOR SCORE COEFFICIENTS (B) FOR ROTATED FACTORS
108 CALL COEFF(V,NS,ME,NDIM,MDIM)
109 TYPE *, ' ROTATED FACTOR SCORE COEFFICIENTS HAVE BEEN COMPUTED !'
110 C NORMALIZE ORIGINAL DATA
111 DO 51 I=1,NS
112

```

```

123      50 A(J)=X(I,J)
124      CALL NORMALIZE(A,NV,T)
125      DO 51 J=1,NV
126      51 X(1,J)=A(J)
127 C COMPUTE FACTOR SCORES BY F=U*B
128      DO 52 I=1,NV
129      DO 52 J=1,ME
130      Q(I,J)=0.
131      DO 52 K=1,NS
132      52 Q(I,J)=Q(I,J)+X(K,I)*V(K,J)
133      TYPE *, ' VARIMAX FACTOR SCORES STORED ON FILE=QVMXSCOR.TMP !'
134      OPEN(UNIT=6,NAME='QVMXSCOR.TMP')
135      WRITE(3,53) FILENAME,TODAY
136      53 FORMAT(1H1'SCORES OF CENSUS VARIABLES ON Q-MODE VARIMAX FACTORS '
137      + /' FOR DATA ON FILE ['A13'] ON 'A9//2X'VAR.NO.    VARIMAX '
138      + 'FACTOR SCORES (* 100)'/)
139      DO 54 I=1,NV
140      WRITE(6,*) FLOAT(I),FLOAT(I),(Q(I,J),J=1,ME)
141      54 WRITE(3,23) I,((Q(I,J)*100.),J=1,ME)
142      TYPE *, ' VARIMAX FACTOR SCORES HAVE BEEN PRINTED !'
143 C ESTIMATE LOADINGS OF VARIMAX FACTORS ON PRINCIPAL COMPONENTS
144      WRITE(3,55)
145      55 FORMAT(1H1'ESTIMATED LOADINGS OF VARIMAX FACTORS ON PRINCIPAL '
146      + 'COMPONENTS'///' FACTOR NO.  LOADINGS ON PRINCIPAL COMPONENTS '
147      + '(* 100)'/)
148      DO 57 I=1,ME
149      DO 56 J=1,ME
150      A(J)=0.
151      DO 56 K=1,NV
152      56 A(J)=A(J)+Q(K,I)*F(K,J)
153      57 WRITE(3,58) I,(A(J)*100.,J=1,ME)
154      58 FORMAT(I7,1X,10F10.1/(8X,10F10.1))
155 C ESTIMATE LOADINGS OF OTHER SAMPLES ON FACTOR AXES
156      TYPE *, '>ESTIMATE LOADINGS OF OTHER SAMPLES ON FACTOR AXES?'
157      59 TYPE *, ' IF YES, INDICATE PRINCIPAL COMPONENTS(P) OR FACTORS(V)'
158      ACCEPT 60, IANS
159      60 FORMAT(A1)
160      IF(IANS.NE.1HV.AND.IANS.NE.1HP) GO TO 90
161      DO 61 I=1,NV
162      DO 61 J=1,ME
163      IF(IANS.EQ.1HV) V(I,J)=Q(I,J)
164      61 IF(IANS.EQ.1HP) V(I,J)=F(I,J)
165      62 TYPE *, '>FILENAME FOR SET OF SAMPLE VECTORS?'
166      ACCEPT *, FILENAME
167      OPEN(UNIT=7,NAME=FILENAME,TYPE='OLD')
168      IF(IANS.EQ.1HP) WRITE(3,63) FILENAME
169      63 FORMAT(1H1'Q-MODE PRINCIPAL COMPONENTS ESTIMATED IN SAMPLES FROM '
170      + 'INPUT FILE ['A13']')
171      IF(IANS.EQ.1HV) WRITE(3,64) FILENAME
172      64 FORMAT(1H1'Q-MODE VARIMAX FACTORS ESTIMATED IN SAMPLES FROM INPUT'
173      + ' FILE ['A13']')
174      TYPE *, '>GIVE ESTIMATED LOADINGS A FILENAME.'
175      ACCEPT *, FILENAME
176      OPEN(UNIT=8,NAME=FILENAME)
177      WRITE(3,65) FILENAME,TODAY
178      55 FORMAT(' ESTIMATED LOADINGS STORED UNDER FILENAME ['A13'] ON '
179      + 'A9///' NO.  DEPTH (M)    AGE (M.Y.)    % EXPLAINED  ESTIMATED '
180      + 'LOADINGS (* 100)'/)
181      SUM=0.
182      MS=0

```



```

1  C <RECTERP> EQUALLY SPACES DATA BY RECTANGULAR INTEGRATION
2  C
3  C DATA SEQUENCE XIN IS CONVERTED TO POINTS EQUALLY SPACED XI UNITS APART
4  C FROM INITIAL POINT XB TO FINAL POINT XL. THE EQUALLY SPACED DATA
5  C SEQUENCE IS XOUT.
6  C
7  C XIN(JS,1) & XIN(JS,2) ARE X & Y COORDINATES OF A POINT IN INPUT ARRAY
8  C XOUT(IS,1) & XOUT(IS,2) ARE X & Y COORDINATES OF POINT IN OUTPUT ARRAY
9  C THE VARIABLE X CONTAINS THE X COORDINATE OF THE NEXT DATA POINT TO BE
10 C INTERPOLATED. THIS VALUE IS INCREMENTED BY XI.
11 C
12 C (AFTER PROGRAM EQSPR BY DAVIS, 1973)
13 C
14     DIMENSION XIN(1000,2),DATA(100),XOUT(3000,2)
15     CHARACTER FILENAME*13
16     TYPE J1
17     31 FORMAT(// ' <RECTERP> EQUALLY SPACES DATA BY RECTANGULAR INTEGRAT
18 + 'ION'// ' >FILENAME FOR INPUT DATA (DUMMY, AGE, ASSOCIATED '
19 + 'VARIABLES) ?' )
20     ACCEPT *,FILENAME
21     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
22     TYPE *, '>INITIAL X-VALUE, FINAL X & INCREMENT FOR SPACED DATA ?'
23     ACCEPT *,XB,XL,XI
24     TYPE *, '>NUMBER OF ASSOCIATED VARIABLES ?'
25     ACCEPT *,NVAR
26     TYPE *, '>SUBSCRIPT OF ASSOCIATED VARIABLE TO BE INTERPOLATED ?'
27     ACCEPT *,NSUB
28     NIN=1
29     33 READ(1,*,END=34) DUMMY,XIN(NIN,1),(DATA(J),J=1,NVAR)
30     XIN(NIN,2)=DATA(NSUB)
31     NIN=NIN+1
32     GO TO 33
33     34 NIN=NIN-1
34 C INITIALIZE INDEXING PARAMETERS AND X COORDINATES OF 1ST OUTPUT INTERVAL
35     XXB=XB
36     XXE=XB+XI
37     IG=1
38     IS=0
39     SL=0.
40     DO 100 I=1,NIN
41     IF(I-NIN) 1,45,45
42     1 DX=XIN(I+1,1)-XIN(I,1)
43     GO TO (11,12,13),IG
44 C DETERMINE WHETHER X COORDINATE OF 1ST OUTPUT INTERVAL IS LESS THAN OR
45 C GREATER THAN X COORDINATE OF 1ST INPUT DATA POINT
46     11 IF(XIN(1,1)-XXB) 7,8,3
47 C X COORDINATE OF BEGINNING OF OUTPUT INTERVAL IS LESS THAN X COORDINATE
48 C OF 1ST INPUT DATA POINT. DETERMINE IF 1ST POINT OF INPUT DATA IS IN
49 C CURRENT OUTPUT INTERVAL.
50     3 IF(XIN(1,1)-XXB) 6,6,4
51     4 IF(XIN(1,1)-XXE) 5,5,5
52 C 1ST INPUT DATA POINT IS NOT IN CURRENT INTERVAL. SET VALUE AT
53 C BEGINNING OF OUTPUT INTERVAL TO 1ST INPUT DATA VALUE AND MOVE TO
54 C NEXT OUTPUT INTERVAL.
55     5 IS=IS+1
56     XOUT(IS,1)=XXB
57     XOUT(IS,2)=XIN(1,2)
58     XXB=XXE
59     XXE=XXE+XI
60     IF(XXB-XL) 3,3,99

```



```

52  C START INTEGRATION
53      5 XXB=XIN(I,1)
54      IG=3
55      GO TO 13
56  C X COORDINATE OF 1ST OUTPUT INTERVAL IS GREATER THAN XCOORDINATE OF 1ST
57  C INPUT DATA POINT. IN THIS CASE FIND 1ST INPUT DATA POINT WHOSE X
58  C COORDINATE IS GREATER THAN BEGINNING X COORDINATE OF 1ST OUTPUT
59  C INTERVAL. THEN INITIALIZE AND START INTEGRATION.
60      7 IG=2
61      12 IF(XIN(I+1,1)-XXB) 100,8,8
62      8 IG=3
63      DX=XIN(I+1,1)-XXB
64  C DETERMINE IF INPUT INTERVAL IS TOTALLY WITHIN CURRENT OUTPUT INTERVAL
65  C OR IF IT COVERS MORE THAN ONE OUTPUT INTERVAL.
66      13 IF((XXB+DX)-XXE) 25,21,21
67  C INPUT INTERVAL COVERS MORE THAN ONE OUTPUT INTERVAL. ADD AREA OF INPUT
68  C INTERVAL WHICH IS WITHIN OUTPUT INTERVAL TO TOTAL AREA OF OUTPUT
69  C INTERVAL. CALCULATE VALUE AT BEGINNING OF OUTPUT INTERVAL AND STORE.
70  C ADJUST INPUT INTERVAL LENGTH FOR THE AREA ALREADY USED. MOVE TO NEXT
71  C OUTPUT INTERVAL.
72      21 DIF=XXE-XXB
73      XXB=XXE
74      SL=SL+DIF*XIN(I,2)
75      SL=SL/XI
76      IS=IS+1
77      XOUT(IS,1)=XXE-XI
78      XOUT(IS,2)=SL
79      DX=DX-DIF
80      SL=0.
81      XXE=XXE+XI
82      IF((XXE-XI)-XL) 13,13,99
83  C INPUT INTERVAL TOTALLY WITHIN CURRENT OUTPUT INTERVAL. ADD AREA OF
84  C INPUT INTERVAL TO AREA OF CURRENT OUTPUT INTERVAL, THEN MOVE TO NEXT
85  C INPUT INTERVAL
86      25 SL=SL+DX*XIN(I,2)
87      XXB=XXB+DX
88      GO TO 100
89  C FOR ALL DATA POINTS IN OUTPUT SEQUENCE WHICH ARE GREATER THAN THE LAST
90  C INPUT DATA POINT, SET OUTPUT VALUE TO LAST VALUE OF INPUT DATA. THIS
91  C IS DONE BY SETTING LENGTH OF LAST INPUT INTERVAL TO A VERY LARGE
92  C VALUE
93      45 DX=1.0E 15
94      GO TO 13
95  100 CONTINUE
96  C OUTPUT THE EQUALLY SPACED POINTS
97      99 OPEN(UNIT=2,NAME='RECTERP.TMP')
98      DO 39 I=1,IS
99      39 WRITE(2,*) XOUT(I,1),XOUT(I,1),XOUT(I,2)
100      TYPE *,IS,' EQUALLY SPACED POINTS STORED IN FILE=RECTERP.TMP !'
101      STOP
102      END

```

```

1  C <REPERCENT> CONVERTS VARIMAX FACTOR LOADINGS AND SCORES TO PERCENT
2  C
3      DIMENSION VL(20),IVL(20),VS(100),IVS(100)
4      CHARACTER FILENAME*13
5      TYPE 1
6      1 FORMAT(' <REPERCENT> CONVERTS VARIMAX FACTORS TO PERCENT !'
7      + '// ' >FILENAME FOR FACTOR LOADINGS ?')
8      ACCEPT *,FILENAME
9      OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
10     TYPE *, '>FILENAME FOR SCORES ?'
11     ACCEPT *,FILENAME
12     OPEN(UNIT=2,NAME=FILENAME,TYPE='OLD')
13     OPEN(UNIT=3,NAME='REPERCENT.TMP')
14     TYPE *, '>NUMBER OF FACTORS ?'
15     ACCEPT *,NF
16     2 READ(1,*,END=5) D,D,(VL(I),I=1,NF)
17     N=N+1
18     C=0
19     S=0
20     DO 3 J=1,NF
21     S=S+ABS(VL(J))
22     3 C=C+VL(J)**2
23     C=SQR(1-C)
24     S=S+C
25     DO 4 J=1,NF
26     4 IVL(J)=NINT((VL(J)/S)*100)
27     WRITE(3,5) N,NINT(D),(IVL(I),I=1,NF)
28     5 FORMAT(20I4)
29     GO TO 2
30     6 TYPE *,N,' SAMPLES COUNTED; RESULTS IN FILE=REPERCENT.TMP !'
31     STOP
32     END

```



```

1  C <RPCA> COMPUTES AN R-MODE PRINCIPAL COMPONENTS ANALYSIS (LOADINGS &
2  C      SCORES) AND COMPUTES A VARIMAX ROTATION
3  C
4  C REQ. SUBR. CORREL, HOW, RSCORES, PRINCO, ROTATE, COEFF, MATINV
5  C
6      DIMENSION X(100),R(100,100),XM(100),XS(100)
7      DIMENSION E(100),V(100,100),A(100),B(100),C(100),D(100),F(100)
8      CHARACTER FILENAME*13,TODAY*9
9      PARAMETER MDIM=100
10     TYPE 1
11     1 FORMAT(///' THIS IS <RPCA> !'///' IT COMPUTES AN R-MODE P.C.A AND '
12     + 'A VARIMAX ROTATION'///' >WHAT'S THE FILENAME FOR THE DATA TO BE '
13     + 'ANALYZED?'/' 'SAMPLE DEPTH(M), AGE(M.Y.), ASSOCIATED '
14     + 'DATA VARIABLES'')
15     ACCEPT *,FILENAME
16     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
17     OPEN(UNIT=3,NAME='PRINT.TMP')
18     CALL DATE(TODAY)
19     TYPE *, '>HOW MANY VARIABLES IN EACH SAMPLE VECTOR?'
20     ACCEPT *,NV
21     WRITE(3,2) FILENAME,TODAY,NV,FILENAME
22     2 FORMAT(1H1'R-MODE P.C.A. OF DATA STORED IN FILE ['A13'] '
23     + 'COMPUTED 'A9//I10' DATA VARIABLES SPECIFIED FOR EACH '
24     + 'SAMPLE VECTOR'//4X'NO. DEPTH (M) AGE (M.Y.) ASSOCIATED'
25     + 'DATA VARIABLES INPUT FROM FILE ['A13']'//)
26     NS=0
27     3 READ(1,*,END=5) XID,AGE,(X(J),J=1,NV)
28     NS=NS+1
29     WRITE(3,4) NS,XID,AGE,(NINT(X(J)),J=1,NV)
30     4 FORMAT(I6,2F11.3,2X,20I4/(30X,20I4))
31     GO TO 3
32     5 TYPE *, 'DATA HAS BEEN INPUT !'
33     CLOSE(UNIT=1)
34     CALL CORREL(FILENAME,NV,XM,XS,R,MDIM)
35     TYPE *, 'R-MATRIX HAS BEEN COMPUTED !'
36     WRITE(3,8) FILENAME
37     8 FORMAT(1H1'STATISTICS FOR INPUT DATA =FILE ['A13']'//8X'NO.'8X
38     + 'MEAN'6X'STD. DEV.'//)
39     TRACE=0.
40     DO 10 J=1,NV
41     WRITE(3,9) J,XM(J),XS(J)
42     9 FORMAT(I10,2F13.3)
43     10 TRACE=TRACE+R(J,J)
44     TYPE 11,TRACE
45     11 FORMAT(///' TRACE = 'F10.1//)
46     TYPE *, '>HOW MANY EIGENVECTORS DO YOU WANT EXTRACTED?'
47     ACCEPT *,NE
48     CALL HOW(NV,MDIM,NE,R,E,V,A,B,C,D)
49     TYPE *, 'EIGENVALUES AND EIGENVECTORS HAVE BEEN COMPUTED !'
50     WRITE(3,12) FILENAME,TODAY
51     12 FORMAT(1H1'R-MODE PRINCIPAL COMPONENTS ANALYSIS OF DATA IN FILE '
52     + '['A13']'/' EIGENVALUES FOR R-MODE EIGENVECTORS '
53     + 'COMPUTED 'A9///12X'EIGENVALUE % VARIANCE CUM.VAR.'//)
54     TYPE *, ' EIGENVALUE % VARIANCE CUM.VAR.'
55     CUM=0.
56     DO 13 I=1,NE
57     VAR=(E(I)/TRACE)*100.
58     CUM=CUM+VAR
59     WRITE(3,14) I,E(I),VAR,CUM
60     13 TYPE 14,I,E(I),VAR,CUM

```

```

52      TYPE *, '>HOW MANY EIGENVECTORS DO YOU WANT TO SAVE AND ROTATE?'
53      ACCEPT *, ME
54      OPEN(UNIT=2, NAME='RSCORES.TMP')
55      WRITE(3,15) FILENAME, TODAY
56 15  FORMAT(1H1'SCORES OF SAMPLES ON R-MODE PRINCIPAL COMPONENTS'
57      + '/' OF DATA FROM FILE ['A13'] ON 'A9//5X
58      + 'NO.    DEPTH(M)    AGE(M.Y.)    '
59      + 'R-MODE PRINCIPAL COMPONENTS SCORES'//)
60      OPEN(UNIT=1, NAME=FILENAME, TYPE='OLD')
61      DO 20 I=1, NS
62      READ(1,*) XID, AGE, (X(J), J=1, NV)
63      CALL RSCORES(X, XM, XS, V, E, 1, NV, 1, MDIM, F)
64      WRITE(2,*) XID, AGE, (F(J), J=1, ME)
65 20  WRITE(3,21) I, XID, AGE, (F(J), J=1, ME)
66 21  FORMAT(17,2F11.3,10F10.1/(29X,10F10.1))
67      CLOSE(UNIT=1)
68      TYPE *, 'R-SCORES HAVE BEEN STORED AS FILE "RSCORES.TMP" !'
69      DO 24 I=1, NV
70      DO 24 J=1, ME
71 24  R(I,J)=V(I,J)
72      CALL PRINCO(R, E, NV, MDIM)
73      TYPE *, 'EIGENVECTORS HAVE BEEN SCALED TO PRINCIPAL COMPONENTS !'
74      OPEN(UNIT=4, NAME='RLOADINGS.TMP')
75      WRITE(3,30) FILENAME
76 30  FORMAT(1H1'EIGENVECTORS OF R-MATRIX SCALED BY EIGENVALUES TO '
77      + 'R-MODE PRINCIPAL COMPONENTS'/' FOR DATA IN FILE ['A13']'
78      + '//2X' VAR.NO.  %EXPLAINED  R-MODE PRINCIPAL COMPONENTS LOADINGS'
79      + '(*100)'//)
80      TOTVAR=0.
81      DO 32 I=1, NV
82      COMM=0.
83      DO 31 J=1, ME
84 31  COMM=COMM+R(I,J)*R(I,J)
85      TOTVAR=TOTVAR+COMM
86      WRITE(4,*) FLDAT(I), FLDAT(I), (R(I,J), J=1, ME)
87      ICCOM=NINT(COMM*100.)
88 32  WRITE(3,33) I, ICCOM, ((R(I,J)*100), J=1, ME)
89 33  FORMAT(16,113,10F10.1/(19X,10F10.1))
90      WRITE(3,34) NINT((TOTVAR*100.)/NV)
91 34  FORMAT(//3X,15'% OF THE TOTAL OBSERVED VARIANCE ACCOUNTED FOR')
92      TYPE *, 'R-MODE PRINCIPAL COMPONENTS STORED IN "RLOADINGS.TMP"'
93  C ROTATE PRINCIPAL COMPONENTS
94      CALL ROTATE(ME, NV, 0, R, MDIM)
95      TYPE *, 'PRINCIPAL COMPONENTS ROTATED TO VARIMAX FACTORS !'
96      OPEN(UNIT=5, NAME='RVMXLOAD.TMP')
97      WRITE(3,39) FILENAME, TODAY
98 39  FORMAT(1H1'R-MODE PRINCIPAL COMPONENTS ROTATED TO VARIMAX FACTORS'
99      + '/' FOR DATA IN FILE ['A13'] ON 'A9//'    NO.    '
100     + '    % EXPLAINED  LOADINGS ON R-MODE VARIMAX FACTORS '
101     + '(* 100)'//)
102     DO 40 I=1, ME
103     E(I)=0.
104     DO 42 I=1, NV
105     COMM=0.
106     DO 41 J=1, ME
107     E(J)=E(J)+R(I,J)*R(I,J)
108 41  COMM=COMM+R(I,J)*R(I,J)
109     WRITE(5,*) FLDAT(I), FLDAT(I), (R(I,J), J=1, ME)
110     ICCOM=NINT(COMM*100.)
111 42  WRITE(3,33) I, ICCOM, ((R(I,J)*100), J=1, ME)

```



```

123      43 FORMAT(//11X'VARIANCE '10(I9'Z'))
124      C COMPUTE FACTOR SCORES COEFFICIENTS (B) FOR ROTATED FACTORS
125          IF(ME.GT.10) ME=10
126          CALL COEFF(R,V,ME,MDIM,MDIM)
127          TYPE *, ' ROTATED FACTOR SCORE COEFFICIENTS HAVE BEEN COMPUTED !'
128      C COMPUTE FACTOR SCORES BY F=Z*B
129          OPEN(UNIT=6,NAME='RVMXSCOR.TMP')
130          WRITE(3,52) FILENAME,TODAY
131      52 FORMAT(1H1'SCORES OF SAMPLES ON R-MODE VARIMAX FACTORS '
132          + /' FOR DATA IN FILE ['A13'] ON 'A9//4X'ND.    DEPTH(1) '
133          + 'AGE(M.Y.)    VARIMAX FACTOR SCORES'//)
134          OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
135          DO 54 I=1,NS
136          READ(1,*) XID,AGE,(X(L),L=1,NV)
137          DO 53 J=1,ME
138              F(J)=0.
139          DO 53 K=1,NV
140      53 F(J)=F(J)+((X(K)-XM(K))/XS(K))*R(K,J)
141          WRITE(6,*) XID,AGE,(F(J),J=1,ME)
142      54 WRITE(3,21) I,XID,AGE,(F(J),J=1,ME)
143          CLOSE(UNIT=1)
144          TYPE *, ' VARIMAX FACTOR SCORES STORED IN "RVMXSCOR.TMP" !'
145      C ESTIMATE SCORES OF OTHER SAMPLES ON FACTOR AXES
146          TYPE *, '>ESTIMATE SCORES OF OTHER SAMPLES ON FACTOR AXES?'
147      59 TYPE *, ' IF YES, INDICATE PRINCIPAL COMPONENTS(P) OR FACTORS(V)'
148          ACCEPT 60, IANS
149      60 FORMAT(A1)
150          IF(IANS.NE.1HV.AN).IANS.NE.1HP) GO TO 90
151      52 TYPE *, '>FILENAME FOR SET OF SAMPLE VECTORS?'
152          ACCEPT *, FNAME
153          OPEN(UNIT=7,NAME=FNAME,TYPE='OLD')
154          IF(IANS.EQ.1HP) WRITE(3,63) FNAME
155      63 FORMAT(1H1'R-MODE PRINCIPAL COMPONENTS SCORES ESTIMATED IN '
156          + 'SAMPLES FROM INPUT FILE ['A13']')
157          IF(IANS.EQ.1HV) WRITE(3,64) FNAME
158      54 FORMAT(1H1'R-MODE VARIMAX FACTORS SCORES ESTIMATED IN SAMPLES '
159          + 'FROM INPUT FILE ['A13']')
160          TYPE *, '>GIVE ESTIMATED SCORES A FILENAME.'
161          ACCEPT *, FNAME
162          OPEN(UNIT=8,NAME=FNAME)
163          WRITE(3,65) FNAME,TODAY
164      65 FORMAT(' ESTIMATED SCORES STORED UNDER FILENAME ['A13'] ON '
165          + 'A9///' ND.    DEPTH (M)    AGE (M.Y.)    ESTIMATED '
166          + 'SCORES(*10)'//)
167          MS=0
168      67 READ(7,*,END=70) DPTH,AG,TOTL,(X(K),K=1,NV)
169          MS=MS+1
170          IF(IANS.EQ.1HP) THEN
171              CALL RSCORES(X,XM,XS,V,E,1,NV,1,MDIM,F)
172              DO 68 I=1,ME
173      68 A(I)=F(I)
174              ELSE
175              DO 69 J=1,ME
176              A(J)=0.
177              DO 69 K=1,NV
178      69 A(J)=A(J)+((X(K)-XM(K))/XS(K))*R(K,J)
179              END IF
180              WRITE(3,21) MS,DPTH,AG,((A(J)*10),J=1,ME)
181              WRITE(8,*) DPTH,AG,(A(J),J=1,ME)
182              GO TO 67

```

```
184      CLOSE(UNIT=8)
185      TYPE *, '>ESTIMATE SCORES OF MORE SAMPLES ON FACTOR AXES?'
186      GO TO 59
187 90 TYPE *, 'ALL FINISHED !'
188      TYPE *, 'PRINTED RESULTS ARE ON FILE "PRINT.TMP" !'
189      STOP
190      END
```



```

1  C <SLICER> STRIPS SYNCHRONOUS ELEMENTS FROM CORRELATED TIME SERIES
2  C
3      DIMENSION X(500),Y(500,50),Z(100)
4      CHARACTER FILENAME*13
5      NSERIES=1
6      TYPE 1
7      1 FORMAT(// ' THIS IS <SLICER> !!'// ' >FILENAME FOR 1ST TIME SERIES '
8      + ' (DUMMY, AGE, ASSOCIATED VARIABLES) ?' )
9      TYPE *, '>NUMBER OF ASSOCIATED VARIABLES ? (SAME FOR ALL SERIES)'
10     ACCEPT *,NVAR
11     TYPE *, '>SUBSCRIPT OF ASSOCIATED VARIABLE TO BE SELECTED ?'
12     ACCEPT *,NSUB
13     2 ACCEPT *,FILENAME
14     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
15     NX=1
16     3 READ(1,*,END=4) DUMMY,X(NX),(Z(I),I=1,NVAR)
17     Y(NX,NSERIES)=Z(NSUB)
18     NX=NX+1
19     GO TO 3
20     4 NX=NX-1
21     TYPE *,NX,' ELEMENTS IN SERIES ',NSERIES
22     TYPE *, '>ANYMORE TIME SERIES ? '
23     ACCEPT 5, IANS
24     5 FORMAT(A1)
25     IF(IANS.EQ.1HY) THEN
26         NSERIES=NSERIES+1
27         CLOSE(UNIT=1)
28         TYPE *, '>FILENAME FOR NEXT SERIES ?'
29         GO TO 2
30     END IF
31     OPEN(UNIT=2,NAME='SLICER.TMP')
32     DO 6 N=1,NX
33     6 WRITE(2,*) X(N),X(N),(Y(N,J),J=1,NSERIES)
34     TYPE *,NX,' ROWS OF ',NSERIES,' SERIES STORED IN FILE=SLICER.TMP!'
35     STOP
36     END

```

```

1  C <SMOOTHER> SMOOTHS DATA SEQUENCE WITH M-TERM WEIGHTED MOVING AVERAGE
2  C
3      DIMENSION D(1000),A(1000),X(1000),Y(1000),Z(21)
4      CHARACTER FILENAME*13
5      TYPE *, 'NUMBER OF TERMS IN SMOOTHING FUNCTION ? (ODD NUMBER)'
6      ACCEPT *,NT
7      TYPE *, 'WEIGHTS FOR TERMS IN SMOOTHING FUNCTION ? (DEPTH, AGE, X)'
8      ACCEPT *,(Z(J),J=1,NT)
9      TYPE *, 'FILENAME FOR SEQUENCE TO BE SMOOTHED ?'
10     ACCEPT *,FILENAME
11     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
12     N=1
13     1 READ(1,*,END=2) D(N),A(N),X(N)
14     N=N+1
15     GO TO 1
16     2 NS=N-1
17     M=NT/2
18     W=0
19     DO 3 I=1,NT
20     3 W=W+Z(I)
21     DO 4 I=1,NS
22     4 Y(I)=X(I)
23     DO 5 I=M,NS-2*M
24     S=0
25     DO 5 J=1,NT
26     5 S=S+(X(I+J-M)*Z(J))
27     Y(I+M)=S/W
28     TYPE *, 'FILENAME FOR OUTPUT ? (DEPTH, AGE, X, SMOOTHED X)'
29     ACCEPT *,FILENAME
30     OPEN(UNIT=2,NAME=FILENAME)
31     DO 7 N=1,NS
32     7 WRITE(2,*) D(N),A(N),X(N),Y(N)
33     STOP
34     END

```



```

1  C <SQUARE> COMPUTES A*A & ITS INVERSE
2  C
3  C REQ. SUBR. MATINV
4  C
5      DIMENSION A(100,10),B(100),F(10,10)
6      CHARACTER FILENAME*13
7      TYPE *,*>FILENAME FOR INPUT?
8      ACCEPT *,FILENAME
9      OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
10     NC=3
11     DO 1 I=1,100
12     1 READ(1,*,END=2) (A(I,J),J=1,NC)
13     2 NR=1-1
14     TYPE *,NR
15     DO 4 I=1,NC
16     DO 3 K=1,NR
17     3 B(K)=A(K,I)
18     DO 4 J=1,NC
19     F(I,J)=0.
20     DO 4 K=1,NR
21     4 F(I,J)=A(K,J)*B(K)+F(I,J)
22     DO 5 J=1,NC
23     DO 5 I=J,NC
24     5 F(I,J)=F(J,I)
25     OPEN(UNIT=2,NAME='SQUARE.TMP')
26     DO 6 I=1,NC
27     6 WRITE(2,*) (F(I,J),J=1,NC)
28     CALL MATINV(F,NC,10,DET)
29     OPEN(UNIT=3,NAME='INVERSE.TMP')
30     DO 7 I=1,NC
31     7 WRITE(3,*) (F(I,J),J=1,NC)
32     TYPE *,'RESULTS ARE ON FILES "SQUARE.TMP" & "INVERSE.TMP" !'
33     STOP
34     END

```

```

1  C <TIMER> - SUBROUTINE <TIMER> CALLING ROUTINE
2  C
3  C INPUT - ARRAY OF SUB-BOTTOM DEPTHS & ASSOCIATED DATA VARIABLES
4  C        - AGE AND DEPTH OF DATED LEVELS
5  C OUTPUT - ARRAY OF ESTIMATED TIME WITH ASSOCIATED DATA VARIABLES
6  C
7  C MAXIMUM SAMPLES N = 1000, MAXIMUM ASSOCIATED VARIABLES M = 100
8  C
9  C REQUIRES SUBROUTINE TIMER
10 C
11     DIMENSION DEPTH(1000),TIME(1000),DATA(100)
12     CHARACTER FILENAME*13
13     TYPE 1
14     1 FORMAT(//' <TIMER> CONVERTS STRATIGRAPHIC INTERVALS TO TIME !'//
15     + ' >FILENAME FOR DATA ? (DEPTH,DUMMY,ASSOCIATED DATA VARIABLES)')
16     ACCEPT *,FILENAME
17     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
18     TYPE *, '>HOW MANY ASSOCIATED DATA VARIABLES ?'
19     ACCEPT *,NV
20     NS=0
21     2 NS=NS+1
22     READ(1,*,END=3) DEPTH(NS),DUMMY,(DATA(I),I=1,NV)
23     GO TO 2
24     3 NS=NS-1
25     CLOSE(UNIT=1)
26     TYPE 4,NS
27     4 FORMAT(/15' SAMPLES COUNTED !'//)
28     CALL TIMER(DEPTH,TIME,NS)
29     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
30     OPEN(UNIT=2,NAME='TIMER.TMP')
31     I=0
32     5 I=I+1
33     READ(1,*,END=5) DEPTH(I),DUMMY,(DATA(J),J=1,NV)
34     WRITE(2,*) DEPTH(I),TIME(I),(DATA(J),J=1,NV)
35     GO TO 5
36     6 TYPE *, 'RESULTS STORED UNDER FILENAME=TIMER.TMP !'
37     STOP
38     END

```



```

1  C <WEATHER> COMPUTES MIOCENE NANNO CLIMATE INDEX FROM 5 TAXA
2  C      COMPUTES DISSOLUTION INDEX FROM 3 TAXA
3  C
4  C      X = SUBBOTTOM DEPTH
5  C      A = DISCOASTERS
6  C      B = C. LEPTOPORUS
7  C      C = R. PSEUDUMBILICA
8  C      D = C. PELAGICUS
9  C      E = DISSOLVED LARGE ELLIPTICAL PLACOLITHS
10 C
11      CHARACTER FILENAME*13
12      DIMENSION X(500),AX(500),Z(500,5),AZ(500,5),INDX(500)
13      TYPE *, '>FILENAME ?'
14      ACCEPT *, FILENAME
15      OPEN(UNIT=1, NAME=FILENAME, TYPE='OLD')
16      OPEN(UNIT=2, NAME='WEATHER.TMP')
17      OPEN(UNIT=3, NAME=FILENAME)
18      N=1
19      1 READ(1,*,END=2) X(N),(Z(N,M),M=1,5)
20      N=N+1
21      GO TO 1
22      2 N=N-1
23      TYPE *, N, ' SAMPLES COUNTED !'
24  C SORT SAMPLES BY DEPTH
25      CALL ASCEND(X, INDX, N)
26      DO 3 I=1, N
27          AX(I)=X(INDX(I))
28          DO 3 J=1, 5
29              3 AZ(I,J)=Z(INDX(I),J)
30              DO 4 I=1, N
31                  X(I)=AX(I)
32                  DO 4 J=1, 5
33                      4 Z(I,J)=AZ(I,J)
34              TYPE *, 'SAMPLES HAVE BEEN RESORTED BY THEIR SUB-BOTTOM DEPTH !'
35              DO 5 I=1, N
36                  A=Z(I,1)
37                  B=Z(I,2)
38                  C=Z(I,3)
39                  D=Z(I,4)
40                  E=Z(I,5)
41                  T=A+B
42                  S1=C+D+E
43                  R=C
44                  S2=C+D
45                  IF(S1.GT.0) T1=T/S1
46                  IF(S2.GT.0) R=E/S2
47                  WRITE(2,*) X(I),X(I),T1,S1,R,S2
48      5 WRITE(3,6) X(I),INT(A),INT(B),INT(C),INT(D),INT(E)
49      6 FORMAT(F8.3,5I3)
50      TYPE *, ' RESULTS STORED IN FILENAME=WEATHER.TMP'
51      STOP
52      END

```

5. SUBROUTINES

DescriptionsSUBROUTINE ASCEND (V, INDEX, N)

Send vector V of length N to be sorted along with a pointer vector INDEX of equal length. The subroutine changes the order of the pointer vector entries so that when sorted the pointer vector entries taken 1 through N give the positions of the minimum through the maximum values in the original sample vector. The sample vector is not touched, only the pointer vector is modified.
After W.R. Allen, 1973.

SUBROUTINE AXISAGE (YO,YI,YL)

Draws age axis incremented by millions of years or thousands of years. YO is value at the origin. YI is increment along axis in m.y. per inch. YL is axis length in inches.

SUBROUTINE AXISDAGE (YO,YLAST,YSCALE,AGE1,DEPTH1,SEDRAT)

Plots depth axis incremented in m.y./meter. YO is value at origin. YLAST is age at end of depth scale. YSCALE is increment in m.y./inch. AGE1 and DEPTH1 are age and depth of first sample (in meters and m.y.). SEDRAT is average sedimentation rate for entire record in m/m.y.

SUBROUTINE AXISDEPTH (YO,YI,YL)

Draws depth axis incremented by meters or centimeters. YO is value at origin. YI is increment in meters per inch. YL is axis length in inches.

SUBROUTINE AXISINTEGER (X,Y,ID,AL,AO,AI)

Draws axis for real data. Fractional data are truncated when increment AI is greater than 1. X and Y specify coordinates for beginning of axis in inches. Axis increment AI is in units per inch. Axis length is in inches. AO is the value at the origin.

SUBROUTINE AXISLOG10 (X,Y,ID,AXLENG,AXORIG,AXINCR)

Draws log-scale axis incremented by powers of 10. X and Y specify coordinates for beginning of axis in inches. Axis increment AXINCR is in inches per power of 10. Axis length AXLENG is in inches. Values at the axis origin AXORIG is a power of 10.

SUBROUTINE BOXLINE (Y,YO,YI,X,XO,XI,XC,N)

Plots vector of X-values of length N as a rectangular line against increasing Y-values. Edges of boxes are drawn halfway between adjacent Y's. XC specifies the "zero" or center from which boxes originate. XO and YO specify the values of X and Y at the origin. XI and YI are X and Y increments per inch.

Requires subroutine ASCEND.

SUBROUTINE CERRORBAR (Y,YORIG,YINC,X,XORIG,XINC,NS,TOTAL,PROB,XC)

Plots interval estimates of census variables expressed as percent. X is vector of percentages of length NS. X and Y increments XI, YI are specified in units per inch. TOTAL is a vector of total number of individuals counted. XC is value assigned XMIN and XMAX, the interval estimates, if TOTAL is a null vector.

Requires subroutine COUNTERROR

SUBROUTINE CHECKDUP (WIDTH,X,Y,N)

Checks vector of X's of length N for adjacent duplicates. If found, average of associated Y's is assigned to the average X-value. WIDTH is the range about X for which adjacent X's are considered duplicates.

SUBROUTINE COEFF (A,N,M,NDIM,MDIM)

Computes factor score coefficients matrix B for rotated orthogonal factors by $B=S(S'S)^{-1}$, see Cooley and Lohnes (1971). A is the N by M factor pattern matrix. N is the number of elements in each factor in A. M is the number of factors in A. NDIM and MDIM are the maximum N and M of A as dimensioned in calling routine. B is the factor coefficients matrix; replaces A. Requires subroutine MATINV.

SUBROUTINE CONFIDAXIS (N,XL1,XL2,XSIN,XCOS,PROB,SMIN,S,SMAX)

Calculates minimum and maximum slopes of major axis of covariation ellipse (see Sokal and Rohlf, 1969, p. 529). N is number of samples. XL1, XL2 are lengths of major and minor axes. XSIN, XCOS are direction sine and cosine of major axis. CL specifies confidence level in percent. S is slope of major axis. SMIN, SMAX are minimum and maximum slopes for major axis at specified CL.

SUBROUTINE CORREL (FILENAME,M,Y,X,A,MDIM)

Computes correlation matrix R (after Cooley and Lohnes, 1971). FILENAME is the file name for input data (ID, ID, associated variables). M is number of elements in each sample vector. X reports standard deviations; Y reports means. A reports correlation matrix. NDIM is the maximum dimension of X, Y and A in the calling routine.

SUBROUTINE COUNTERROR (TOTAL,PERCNT,PROB,PERMAX,PERMIN)

Computer interval estimates at specified confidence levels for percentage data. Formulae after Mosimann (1965, p. 642-643). TOTAL number of individuals counted. PERCNT is a percentage value. PROB is specified confidence level in percent. Interval estimate is between PERMAX and PERMIN.

SUBROUTINE COVAR (FILENAME,M,Y,X,A,NDIM)

Computes variance-covariance or dispersion (D) matrix (after Cooley and Lohnes, 1971). FILENAME is the file name for input data (ID, ID, associated variables). M is number of elements in each sample vector. X reports standard deviations; Y reports means. A reports dispersion matrix. NDIM is maximum dimension of X, Y, and A in calling routine.

SUBROUTINE CROSSPROD (FILENAME,M,Y,A,NDIM)

Computes matrix of raw sums of squares and cross-products (cross-product matrix). FILENAME is the file name for input data (ID, ID, associated variables). M is the number of elements in each sample vector. A reports cross-product matrix. Y reports raw sums. NDIM is maximum dimensions of X, Y and A in calling program.

SUBROUTINE CURVELINE (Y,YO,YI,X,XO,XI,N,WIDTH)

Plots a curved line (spline) through N X-values against increasing Y. YO and XO are X and Y values at the origin. XI and YI are X and Y increments per inch. WIDTH is range about each Y-value within which other Y's are considered duplicate; usually WIDTH=0. Requires subroutines SPACER, CHECKDUP, and SPLINE

SUBROUTINE DIAG (A,V,UTILD)

Diagonalizes a 2x2 symmetric matrix, applying a quadratic solution for its eigenvalues and eigenvectors. A is the matrix. V are the eigenvalues. UTILD are the eigenvectors. After a program used by Dagbert and David (1974).

SUBROUTINE DOT (XC,YC,RAD)

Plots a circle given its radius RAD and X,Y coordinates of its center XC,YC (all in inches). After Mary Hunt, 1975.

SUBROUTINE DSCORES (X,XM,V,E,N,M,NDIM,MDIM,F)

Computes "D-mode" principal component scores from results of diagonalizing a variance-covariance (dispersion) matrix. X is the matrix of raw sample vectors $X(N,M)$. XM is the vector of means $XM(M)$. E is the vector of eigenvalues $E(M)$. V is the matrix of unit eigenvectors of the D-matrix, $V(N,M)$. N is the number of sample vectors in X, M the number of elements in each sample vector. NDIM and MDIM are the numbers of rows and columns dimensioned for X in the calling routine. F is the matrix of "D-mode" factor scores.

SUBROUTINE ELLIPSE (XO,YO,AA,BB,COTET,SITET,XMIN,YMIN,S,XMAX,YMAX)

Plots an ellipse given coordinates of its center XO,YO, its major and minor axes AA, BB, and the direction sines and cosines of its major axis COTET, SITET. XMIN,YMIN,S,XMAX,YMAX are plot dimensions and scaling parameters.

After a program used by Dagbert and David (1974).

SUBROUTINE HATCHIT (Y,YORIG,YINC,X,XORIG,XINC,NS,XC)

Hatches in a plotted sequence. X is vector of variable plotted. XINC and YINC are increments specified in units/inch. XORIG and YORIG are values at origin. NS is number of Y,X pairs. XC is middle of plot about which it is hatched.

SUBROUTINE HOW (MVAR,MDIM,NVECT,R,E,V,A,B,C,D)

Computes eigenvalues and eigenvectors of a symmetric matrix. Algorithm by Householder, Ortega and Wilkinson. MVAR is the order of the input matrix R. MDIM is the dimensioned size of R, E, V, A, B, C, and D in the calling program. NVECT is the number of eigenvectors to be computed. E is the vector in which eigenvalues are returned. V is the matrix in which the eigenvectors are returned. The eigenvectors are stored as columns in V. A,B,C, and D are workspace vectors. From Cooley and Lohnes (1971), modified to run on the VAX 11/780.

SUBROUTINE MATINV (A,M,MDIM,DET)

Computes inverse and determinant of $A(M,M)$ by the Gauss-Jordan method. M is the order of the square matrix A. MDIM is the maximum M for A as dimensioned in the calling routine. A inverse replaces A. Determinant of A is placed in DET. From Cooley and Lohnes (1971).

SUBROUTINE MATPRINT (R,M,NDIM,IOUT)

Prints a square matrix. $R(M,M)$ is the matrix to be printed, of order M. NDIM is the dimension of R in the calling program. IOUT is the unit number for printed output.

SUBROUTINE MULTIPLY (X,IROWS,IDIM,JCOLS,JDIM,Y,KCOLS,KDIM,Z)

Multiplies two matrices. X is the first matrix. I is the number of rows in the X and resulting Z matrix. IDIM is the maximum number of rows in X and in Z as dimensioned in calling routine. J is the number of columns in X and rows in Y. JDIM is the maximum number of columns in X and rows in Y as dimensioned in the calling routine. Y is the second matrix. K is the number of columns in the Y and Z matrices. KDIM is the maximum number of columns in the Y and Z matrices as dimensioned in the calling routine. Z is the resulting product matrix.

SUBROUTINE NORMALIZE (X,M,XL)

Normalizes a sample vector X to unit length. M is the number of elements in X. XL is the length of the raw sample vector before adjustment.

SUBROUTINE OMEGA (T,S,DEPTH,ALK,TCO2,W,OM)

Computes carbonate saturation OM from in situ temperature T, salinity S, water DEPTH in km, alkalinity ALK and total CO_2 TCO_2 . After a program in BASIC by Robin Keir.

SUBROUTINE PERCENT (X,M,T)

Converts raw sample to percent. X is the sample vector with M elements. T is the total number of individuals counted.

SUBROUTINE POINTPLOT (Y,YORIG,YINC,X,XORIG,XINC,NS)

Plots X-values as points against increasing Y. YORIG and XORIG are X and Y values at the origin. YINC and XINC are X and Y increments in units per inch. NS is the number of elements in X and Y.

SUBROUTINE POLAR (X,Y,THETA,R)

Converts X, Y rectangular coordinates into polar coordinates THETA and R.

SUBROUTINE PRINCO (V,E,N,NDIM)

Scales eigenvectors to principal components. V is the matrix of unit eigenvectors. E is the vector of eigenvalues. N is the number of elements in each eigenvector. NDIM is the order of V as dimensioned in the calling routine. Principal components = eigenvectors * squareroot of their eigenvalues.

SUBROUTINE QMATRIX (X,Q,N,M,NDIM,MDIM)

Computes cosine theta matrix Q. X is the matrix of raw sample vectors. Q is the cosine theta matrix. N is the number of sample vectors in X and M is the number of elements in each sample vector of X. NDIM and MDIM are numbers of rows and columns dimensioned for X in the calling routine.

Requires subroutine NORMALIZE.

SUBROUTINE QSCORES (X,V,E,N,M,NDIM,MDIM,F)

Computes Q-mode principal factor scores from results of diagonalizing a cosine-theta (Q) matrix. X is the matrix of raw sample vectors X(N,M). E is the vector of eigenvalues E(N). V is the matrix of unit eigenvectors of Q, V(N,N). N is the number of sample vectors in X, M the number of elements in each sample vector of X. NDIM and MDIM are the numbers of rows and columns dimensioned for X in the calling routine. F is the matrix of "factor scores" F(M,N). L is the maximum number of non-zero eigenvalues (minimum of M or N). Q scores = transposed normalized data * eigenvectors * 1/squareroot of the eigenvalues.

SUBROUTINE RATIOERROR (SPIKE,RATIO,PROB,RATMAX,RATMIN)

Computes interval estimate at specified confidence level for ratio of two different counting categories. Formulae after Mosimann (1965), p. 659-660. SPIKE is the denominator of RATIO. PROB is the specified confidence level in percent. The interval estimate is between RATMAX and RATMIN.

SUBROUTINE RECTANG (Y,YO,YI,X,XO,XI,XC,N)

Plots X-values as rectangles against increasing Y-values. N is the number of X and Y values. XO and YO are the X and Y values at the origin. XI and YI are X and Y increments in units per inch. XC specifies the "zero" or center from which the rectangles originate.

SUBROUTINE RERRORBAR (Y,YORIG,YINC,X,XORIG,XINC,NS,SPIKE,PROB,XC)

Plots interval estimates of census variables expressed as ratios. X is the vector of ratios. XORIG is the value of X at the origin, as a power of 10. When X is less than XORIG, X is plotted at the origin. The X increment XINC is in inches per power of 10. XC is the value to be assigned XMIN and XMAX if SPIKE is a null vector.

Requires subroutine RATIOERROR

SUBROUTINE ROTATE (N,M,METHOD,A,NDIM)

Varimax (or quartimax) rotation of orthogonal factors (after Cooley and Lohnes, 1971). N is the number of factors, M is the number of tests (elements in each factor). METHOD is 0 for varimax, 1 for quartimax. A is the factor pattern matrix A(M,N). NDIM is the maximum dimension of A in the calling program.

SUBROUTINE RSCORES (X,KM,XS,V,E,N,M,NDIM,MDIM,F)

Computes R-mode principal component scores from results of diagonalizing a correlation (R) matrix. X is the matrix of raw sample vectors X(N,M). XM is the vector of means XM(M), XS the vector of standard deviations XS(M). E is the vector of eigenvalues E(M). V is the matrix of unit eigenvectors of the R matrix, V(N,M). N is the number of sample vectors in X, M the number of elements in each sample vector. NDIM and MDIM are the numbers of rows and columns dimensioned for X in the calling routine. F is the matrix of "factor scores" F(N,M).

SUBROUTINE SORTVECTOR (S,X,N)

Sorts a vector X of length N according to increasing values in vector S.
Requires subroutine ASCEND

SUBROUTINE SPACER (N,M,X,Y,Z,SPACE,START,WIDTH)

Interpolates using a spline. X and Y are vectors of length N. Z is a vector of Y values interpolated at equal X increments of SPACE; there are M values in Z. START is the initial X-value and must not be less than X(1). WIDTH is the range about X for which adjacent X's are considered duplicates and are averaged.
Requires subroutines CHECKDUP and SPLINE

SUBROUTINE SPLINE (NPTS,X,Y,B,C,D)

Computes the coefficients of a cubic natural spline. X and Y are arrays with the input data, each containing NPTS points. B,C, and D are arrays with the coefficients of the cubic spline. To interpolate data between X(I) and X(I+1) at value XX use

$$S(XX) = ((D(I)*T + C(I))*T + B(I))*T + Y(I), \text{ where } T = XX - X(I)$$

After Carnahan, Luther and Wilkes (1969)

SUBROUTINE TIMER (DEPTH,TIME,N)

Converts stratigraphic DEPTH to TIME, estimated by linear interpolation between, or extrapolation beyond, known age levels. N is number of depths to be converted.

SUBROUTINE TSLOT (Y,X,N,ITYPE,LP)

Plots a series on the line printer. X is the vector array to be plotted. Y is the vector array associated with X that is to be printed but not plotted. N is the number of Y, X elements. If ITYPE=1 data plotted must have range -1 to +1; ITYPE=2 any data may be plotted; ITYPE=3 LOG10 of the X array will be plotted (original X array will not be destroyed). LP is the unit number for printed output.
After Davis (1973)

SUBROUTINE UNITVAR (X,N,XM,XV)

Standardizes elements of a variable vector to zeromean and unit variance. X is a vector of length N containing the raw input and standardized output. XM and XV are the computed mean and variance of X.

SUBROUTINE XYLOT (X,Y,N,LP)

Plots X,Y pairs on a line printer. X and Y are vector arrays of length N to be plotted. LP is the unit number for printed output.
After Davis (1973)

SUBROUTINE ZEROMEAN (X,N,XM)

Standardizes elements of a vector array to zeromean. X is the array of raw input and standardized output of length N. XM is the computed mean of X.

Listings

All subroutines are accessible from PATS1/LIBRARY

ASCEND
AXISAGE
AXISDAGE
AXISDEPTH
AXISINTEGER
AXISLOGIO
BOXLINE
CERRORBAR
CHECKDUP
COEFF
CONFIDAXIS
CORREL
COUNTERROR
COVAR
CROSSPROD
CURVELINE
DIAG
DOT
DScores
ELLIPSE
HATCHIT
HOW
MATINV
MATPRINT
MULTIPLY
NORMALIZE
OMEGA
PERCENT
POINTPLOT
POLAR
PRINCO
QMATRIX
QScores
RATIOERROR
RECTANG
RERRORBAR
ROTATE
RScores
SORTVECTOR
SPACER
SPLINE
TIMER
TSLOT
UNITVAR
XYPLOT
ZEROMEAN


```

1      SUBROUTINE ASCEND(V,INDEX,N)
2      C
3      C BUBBLE SORT IN ASCENDING VALUE
4      C
5      C SEND VECTOR OF LENGTH N TO BE SORTED ALONG WITH A
6      C POINTER VECTOR OF EQUAL LENGTH. THE SUBROUTINE CHANGES THE
7      C ORDER OF THE POINTER VECTOR ENTRIES SO THAT WHEN SORTED
8      C THE POINTER VECTOR ENTRIES TAKEN IN ORDER 1 THROUGH N
9      C GIVE THE POSITIONS OF THE MINIMUM THROUGH THE MAXIMUM
10     C VALUES IN THE ORIGINAL SAMPLE VECTOR. THE SAMPLE VECTOR
11     C ENTRIES ARE NOT TOUCHED, ONLY THE POINTER VECTOR IS MODIFIED.
12     C
13     C AFTER W. RICHARD ALLEN, NOVEMBER 9, 1973
14     C
15     DIMENSION V(1),INDEX(1)
16     DO 1 I=1,N
17     1 INDEX(I)=I
18     DO 4 I=2,N
19     IF(V(INDEX(I-1)).LE.V(INDEX(I))) GO TO 4
20     IT=INDEX(I)
21     J=I
22     2 J=J-1
23     C SHIFT
24     INDEX(J+1)=INDEX(J)
25     IF(J.LT.2) GO TO 3
26     IF(V(INDEX(J-1)).LE.V(IT)) GO TO 3
27     GO TO 2
28     C INSERT
29     3 INDEX(J)=IT
30     C END OF SORT
31     4 CONTINUE
32     RETURN
33     END

```

```

1      SUBROUTINE AXISINTEGER(X,Y,ID,AL,AJ,AI)
2      C
3      C DRAWS AXIS FOR REAL DATA, TRUNCATED AND LABELLED AS INTEGER VALUES
4      C
5      C X AND Y SPECIFY BEGINNING OF AXIS IN INCHES
6      C AXIS-INCREMENT IS IN PERCENT PER INCH
7      C AXIS-LENGTH IS IN INCHES
8      C
9      DIMENSION ID(5)
10     CALL PLOT(X,Y,3)
11     CALL PLOT(X,Y+.14,2)
12     N=NINT(AL)
13     DO 1 I=1,N
14     CALL PLOT(X+I,Y+.14,2)
15     CALL PLOT(X+I,Y,2)
16 1    CALL PLOT(X+I,Y+.14,2)
17     DO 2 I=1,(N/2)+1
18     IF (ABS(AI).LT.1) THEN
19         IF ((10*AI-INT(10*AI)).GT..1) THEN
20             CALL NUMBER(X-.32+2*(I-1),Y+.35,.28,AJ+(I-1)*AI*2,0.,2)
21             ELSE
22             CALL NUMBER(X-.32+2*(I-1),Y+.35,.28,AJ+(I-1)*AI*2,0.,1)
23             END IF
24         ELSE
25             CALL NUMBER(X-.13+2*(I-1),Y+.35,.28,AJ+(I-1)*AI*2,0.,-1)
26             END IF
27 2    CONTINUE
28     CALL SYMBOL(A_/2-3.5,Y+.85,.35,ID,0.,20)
29     RETURN
30     END

```



```

1      SUBROUTINE AXISAGE(YO,YI,YL)
2      C
3      C PLOTS AGE AXIS INCREMENT IN M.Y. IF YI .GE. .1
4      C                               IN K.Y. IF YI .LT. .1
5      C
6      C      YO = VALUE AT ORIGIN
7      C      YI = INCREMENT IN M.Y. PER INCH
8      C      YL = AXIS LENGTH IN INCHES
9      C
10     C
11     MIN=NINT(YL)
12     CALL NEWPEN(2)
13     IF(YI.LT..1) GO TO 2
14     YSPACE=YI*10
15     DO 1 I=0,MIN-1
16     DO 1 K=0,IFIX(YSPACE)
17     YN=YO+I*YI+FLOAT(K)/10
18     IF((YN-IFIX(YN)).LT..01) THEN
19     CALL NEWPEN(1)
20     CALL NUMBER(0.,-((I*YI+FLOAT(K)/10)/YI)-.17,.35,YN,0.,-1)
21     CALL SYMBOL(.92,-((I*YI+FLOAT(K)/10)/YI),.28,13,90.,-1)
22     CALL SYMBOL(.92,-((I*YI+FLOAT(K)/10)/YI),.28,13,90.,-2)
23     CALL NEWPEN(2)
24     END IF
25     CALL SYMBOL(.92,-FLOAT(I)-K/YSPACE,.14,13,90.,-1)
26     1 CALL SYMBOL(.92,-FLOAT(I)-K/YSPACE,.14,13,90.,-2)
27     YN=YO+MIN*YI+FLOAT(K)/10
28     IF((YN-IFIX(YN)).LT..01) THEN
29     CALL NEWPEN(1)
30     CALL NUMBER(0.,-((MIN*YI+FLOAT(K)/10)/YI)-.17,.35,YN,0.,-1)
31     CALL SYMBOL(.92,-((MIN*YI+FLOAT(K)/10)/YI),.28,13,90.,-1)
32     CALL SYMBOL(.92,-((MIN*YI+FLOAT(K)/10)/YI),.28,13,90.,-2)
33     END IF
34     CALL NEWPEN(1)
35     CALL PLOT(1.05,0.,-3)
36     RETURN
37     2 YSPACE=YI*100
38     DO 3 I=0,MIN-1
39     DO 3 K=0,IFIX(YSPACE)
40     YN=(YO+I*YI+FLOAT(K)/100)*10
41     IF((YN-IFIX(YN)).LT..01) THEN
42     CALL NEWPEN(1)
43     CALL NUMBER(0.,-((I*YI+FLOAT(K)/100)/YI)-.14,.28,YN*100,0.,-1)
44     CALL SYMBOL(.92,-((I*YI+FLOAT(K)/100)/YI),.28,13,90.,-1)
45     CALL SYMBOL(.92,-((I*YI+FLOAT(K)/100)/YI),.28,13,90.,-2)
46     CALL NEWPEN(2)
47     END IF
48     CALL SYMBOL(.92,-FLOAT(I)-K/YSPACE,.14,13,90.,-1)
49     3 CALL SYMBOL(.92,-FLOAT(I)-K/YSPACE,.14,13,90.,-2)
50     YN=(YO+MIN*YI+FLOAT(K)/100)*10
51     IF((YN-IFIX(YN)).LT..01) THEN
52     CALL NEWPEN(1)
53     CALL NUMBER(0.,-((MIN*YI+FLOAT(K)/100)/YI)-.14,.28,YN*100,0.,-1)
54     CALL SYMBOL(.92,-((MIN*YI+FLOAT(K)/100)/YI),.28,13,90.,-1)
55     CALL SYMBOL(.92,-((MIN*YI+FLOAT(K)/100)/YI),.28,13,90.,-2)
56     END IF
57     CALL NEWPEN(1)
58     CALL PLOT(1.05,0.,-3)
59     RETURN
60     END

```

```

1      SUBROUTINE AXISDAGE(YO,YLAST,YSCALE,AGE1,DEPTH1,SED RAT)
2      C
3      C PLOTS DEPTH AXIS INCREMENTED IN M.Y./METER
4      C
5      C   YO = VALUE AT ORIGIN
6      C   YLAST = AGE AT END OF DEPTH SCALE
7      C   YSCALE = INCREMENT IN M.Y. PER INCH
8      C   AGE1 & DEPTH1 = AGE & DEPTH OF 1ST SAMPLE (M.Y. & METERS)
9      C   SED RAT = SEDIMENTATION RATE (METERS/M.Y.)
10     C
11     C
12     C DETERMINE NEXT WHOLE METER AND ITS AGE
13         DEPTH=FLOAT(INT(DEPTH1+.999))
14         AGEDEPTH=AGE1+(DEPTH-DEPTH1)/SED RAT
15     C PLOT AND INCREMENT BY A METER
16         1 YY=-(AGEDEPTH-YO)/YSCALE
17         IF((DEPTH-10*INT(DEPTH/10)).LT..9) THEN
18             CALL NEWPEN(1)
19             CALL NUMBER(0.,YY,.14,.28,DEPTH,0.,-1)
20             CALL SYMBOL(1.06,YY,.28,13,90.,-1)
21             CALL SYMBOL(1.06,YY,.28,13,90.,-2)
22         ELSE
23             CALL NEWPEN(2)
24             CALL SYMBOL(1.06,YY,.14,13,90.,-1)
25             CALL SYMBOL(1.06,YY,.14,13,90.,-2)
26         END IF
27         DEPTH=DEPTH+1
28         AGEDEPTH=AGEDEPTH+1/SED RAT
29         IF(AGEDEPTH.LE.YLAST) GO TO 1
30         CALL NEWPEN(1)
31         CALL PLOT(1.2,0.,-3)
32         RETURN
33     END

```



```

1      SUBROUTINE AXISDEPTH(YD,YI,YL)
2      C
3      C   PLOTS DEPTH AXIS INCREMENTED IN METERS IF YI .GE. 1
4      C                                   IN CENTIMETERS IF YI .LT. 1
5      C
6      C   YD = VALUE AT ORIGIN
7      C   YI = INCREMENT IN METERS PER INCH
8      C   YL = AXIS LENGTH IN INCHES
9      C
10     C
11     MIN=NINT(YL)
12     IF(YI.LT.1) GO TO 3
13     DO 2 I=0,MIN-1
14     CALL NUMBER(0.,-FLOAT(I)-.14,.28,YD+I*YI,0.,-1)
15     CALL SYMBOL(1.06,-FLOAT(I),.28,13,90.,-1)
16     CALL SYMBOL(1.06,-FLOAT(I),.28,13,90.,-2)
17     CALL NEWPEN(2)
18     DO 1 K=0,IFIX(YI)
19     CALL SYMBOL(1.06,-FLOAT(I)-K/YI,.14,13,90.,-1)
20     1 CALL SYMBOL(1.06,-FLOAT(I)-K/YI,.14,13,90.,-2)
21     CALL NEWPEN(1)
22     2 CONTINUE
23     CALL NUMBER(0.,-FLOAT(MIN)-.14,.28,YD+MIN*YI,0.,-1)
24     CALL SYMBOL(1.06,-FLOAT(MIN),.28,13,90.,-1)
25     CALL SYMBOL(1.06,-FLOAT(MIN),.28,13,90.,-2)
26     CALL PLOT(1.2,0.,-3)
27     RETURN
28     3 DO 5 I=0,MIN-1
29     CALL NUMBER(0.,-FLOAT(I)-.14,.28,(YD+I*YI)*100,0.,-1)
30     CALL SYMBOL(1.36,-FLOAT(I),.28,13,90.,-1)
31     CALL SYMBOL(1.36,-FLOAT(I),.28,13,90.,-2)
32     CALL NEWPEN(2)
33     DO 4 K=0,IFIX(YI*10)
34     CALL SYMBOL(1.36,-FLOAT(I)-K/(YI*10),.14,13,90.,-1)
35     4 CALL SYMBOL(1.36,-FLOAT(I)-K/(YI*10),.14,13,90.,-2)
36     CALL NEWPEN(1)
37     5 CONTINUE
38     CALL NUMBER(0.,-FLOAT(MIN)-.14,.28,(YD+MIN*YI)*100,0.,-1)
39     CALL SYMBOL(1.36,-FLOAT(MIN),.28,13,90.,-1)
40     CALL SYMBOL(1.36,-FLOAT(MIN),.28,13,90.,-2)
41     CALL PLOT(1.5,0.,-3)
42     RETURN
43     END

```

```

1      SUBROUTINE AXISLOG10(X,Y,ID,AXLENG,AXORIG,AXINCR)
2      C
3      C DRAWS LOG-SCALE AXIS INCREMENTED BY POWERS OF 10
4      C
5      C X AND Y SPECIFY BEGINNING OF AXIS IN INCHES
6      C AXIS-INCREMENT IS IN INCHES PER POWER OF 10
7      C AXIS-LENGTH IS IN INCHES
8      C
9      DIMENSION ID(5)
10     CALL PLOT(X,Y,3)
11     CALL PLOT(X,Y+.14,2)
12     N=NINT(AXLENG/AXINCR)
13     DO 1 I=1,N
14     DO 1 J=2,10
15     CALL PLOT(X+ALOG10(FLOAT(J))*AXINCR+AXINCR*(I-1),Y+.14,2)
16     CALL PLOT(X+ALOG10(FLOAT(J))*AXINCR+AXINCR*(I-1),Y,2)
17     1 CALL PLOT(X+ALOG10(FLOAT(J))*AXINCR+AXINCR*(I-1),Y+.14,2)
18     DO 2 I=1,N+1
19     2 CALL NUMBER(X-.14+(I-1)*AXINCR,Y+.35,.28,AXORIG+FLOAT(I-1),0.,-1)
20     CALL SYMBOL(AXLENG/2-3.5,Y+.85,.35,ID,0.,20)
21     RETURN
22     END

```



```

1      SUBROUTINE BOXLINE(Y,Y0,Y1,X,X0,XI,XC,N)
2      C
3      C PLOTS X-VALUES AS RECTANGULAR LINE AGAINST INCREASING Y-VALUES
4      C
5      C XC SPECIFIES THE "ZERO" OR CENTER FROM WHICH BOXES ORIGINATE
6      C
7      C REQUIRES SUBROUTINE ASCEND
8      C
9      DIMENSION X(1),Y(1),INDEX(1000)
10     CALL ASCEND(Y,INDEX,N)
11     YY1=(Y(INDEX(1))-Y0)/Y1
12     X1=(XC-X0)/XI
13     CALL PLOT(X1,-YY1,3)
14     CALL PLOT((X(INDEX(1))-X0)/XI,-YY1,2)
15     DO 1 I=2,N
16     YY=(Y(INDEX(I-1))+Y(INDEX(I))-2*Y0)/(2*Y1)
17     CALL PLOT((X(INDEX(I-1))-X0)/XI,-YY,2)
18     1 CALL PLOT((X(INDEX(I))-X0)/XI,-YY,2)
19     YY2=(Y(INDEX(N))-Y0)/Y1
20     CALL PLOT((X(INDEX(N))-X0)/XI,-YY2,2)
21     CALL PLOT(X1,-YY2,2)
22     CALL PLOT(X1,-YY1,2)
23     RETURN
24     END

```

```

1      SUBROUTINE CERRORBAR(Y,YORIG,YINC,X,XORIG,XINC,NS,TOTAL,PROB,XC)
2      C
3      C PLOTS INTERVAL ESTIMATES OF CENSUS VARIABLES EXPRESSED AS PERCENT
4      C
5      C X IS VECTOR OF PERCENTS
6      C X AND Y-INCREMENTS ARE SPECIFIED IN UNITS PER INCH
7      C XC IS VALUE ASSIGNED XMIN & XMAX IF "TOTAL" IS A NULL VECTOR
8      C
9      C REQ. SUBROUTINE COUNTERROR
10     C
11     DIMENSION X(1),Y(1),TOTAL(1)
12     CALL NEWPEN(2)
13     DO 1 I=1,NS
14     CALL COUNTERROR(TOTAL(I),X(I),PROB,XMAX,XMIN)
15     YY=(Y(I)-YORIG)/YINC
16     XX=(X(I)-XORIG)/XINC
17     IF(TOTAL(I).EQ.0) THEN
18         XMAX=(XC-XORIG)/XINC
19         XMIN=(XC-XORIG)/XINC
20     ELSE
21         XMAX=(XMAX-XORIG)/XINC
22         XMIN=(XMIN-XORIG)/XINC
23     END IF
24     CALL SYMBOL(XMIN,-YY,.14,13,0.,-1)
25     CALL SYMBOL(XMIN,-YY,.14,13,0.,-2)
26     CALL SYMBOL(XX,-YY,.07,1,0.,-2)
27     CALL SYMBOL(XX,-YY,.07,1,0.,-2)
28     CALL SYMBOL(XMAX,-YY,.14,13,0.,-2)
29     CALL SYMBOL(XMAX,-YY,.14,13,0.,-2)
30     I CALL PLOT(XMIN,-YY,2)
31     CALL NEWPEN(1)
32     RETURN
33     END

```



```

1      SUBROUTINE CHECKDUP(WIDTH,X,Y,N)
2      C
3      C CHECKS VECTOR OF X'S FOR ADJACENT DUPLICATES; IF FOUND, AVERAGE OF
4      C ASSOCIATED Y'S IS ASSIGNED TO THE AVERAGE X-VALUE
5      C
6      C WIDTH = RANGE ABOUT X FOR WHICH ADJACENT X'S ARE CONSIDERED DUPLICATE
7      C
8      DIMENSION X(1),Y(1),A(1000),B(1000)
9      I=1
10     J=2
11     A(I)=X(I)
12     B(I)=Y(I)
13     1 K=1
14     2 IF((X(J)-WIDTH).LE.X(J-1)) THEN
15         A(I)=A(I)+X(J)
16         B(I)=B(I)+Y(J)
17         K=K+1
18         IF(J.EQ.N) GO TO 3
19         J=J+1
20         GO TO 2
21     END IF
22     3 A(I)=A(I)/K
23     B(I)=B(I)/K
24     I=I+1
25     A(I)=X(J)
26     B(I)=Y(J)
27     IF(J.EQ.N) GO TO 4
28     J=J+1
29     GO TO 1
30     4 N=I
31     DO 5 I=1,N
32     X(I)=A(I)
33     5 Y(I)=B(I)
34     RETURN
35     END

```

```
1      SUBROUTINE COEFF(A,N,M,NDIM,MDIM)
```

```
2      C
```

```
3      C COMPUTES FACTOR SCORE COEFFICIENTS MATRIX B FOR ROTATED
```

```
4      C ORTHOGONAL FACTORS BY  $B=S*(S'S)^{-1}$  INVERSE
```

```
5      C
```

```
6      C      A      = FACTOR PATTERN MATRIX A(N,M)
```

```
7      C      N      = NUMBER OF ELEMENTS IN EACH FACTOR IN A
```

```
8      C      M      = NUMBER OF FACTORS IN A
```

```
9      C      NDIM,MDIM = MAXIMUM NUMBER OF N,M AS DIMENSIONED IN CALLING ROUTINE
```

```
10     C      B      = FACTOR COEFFICIENTS MATRIX, REPLACES A-MATRIX
```

```
11     C
```

```
12     C REQUIRES SUBROUTINE MATINV
```

```
13     C
```

```
14     DIMENSION A(NDIM,MDIM),B(100),F(100,100)
```

```
15     DO 2 I=1,M
```

```
16     DO 1 K=1,N
```

```
17     1 B(K)=A(K,I)
```

```
18     DO 2 J=1,M
```

```
19     F(I,J)=0.
```

```
20     DO 2 K=1,N
```

```
21     2 F(I,J)=A(K,J)*B(K)+F(I,J)
```

```
22     DO 3 J=1,M
```

```
23     DO 3 I=J,M
```

```
24     3 F(I,J)=F(J,I)
```

```
25     CALL MATINV(F,M,100,DET)
```

```
26     DO 5 I=1,N
```

```
27     DO 4 K=1,M
```

```
28     B(K)=0.
```

```
29     DO 4 J=1,M
```

```
30     4 B(K)=A(I,J)*F(J,K)+B(K)
```

```
31     DO 5 J=1,M
```

```
32     5 A(I,J)=B(J)
```

```
33     RETURN
```

```
34     END
```



```

1      SUBROUTINE CONFIDAXIS(N,XL1,XL2,XSIN,XCOS,PROB,SMIN,S,SMAX)
2      C
3      C      CALCULATES MINIMUM AND MAXIMUM SLOPES OF MAJOR AXIS
4      C      OF COVARIATION ELLIPSE (SOKAL AND ROHLF, 1969, P. 529)
5      C
6      C
7      C      N = NUMBER OF SAMPLES
8      C      XL1,XL2 = LENGTH OF MAJOR AND MINOR AXES
9      C      XSIN,XCOS = DIRECTION SINES AND COSINES
10     C      CL = CONFIDENCE LEVEL IN PERCENT
11     C      S = SLOPE OF MAJOR AXIS
12     C      SMIN,SMAX = MINIMUM AND MAXIMUM S
13     C
14     CHI=7.88
15     IF(PROB .LE. 99.)CHI=6.63
16     IF(PROB .LE. 97.5)CHI=5.02
17     IF(PROB .LE. 95.)CHI=3.84
18     IF(PROB .LE. 90.)CHI=2.71
19     IF(PROB .LE. 75.)CHI=1.32
20     IF(PROB .LE. 50.)CHI=0.455
21     EV1=(XL1/2.)**2
22     EV2=(XL2/2.)**2
23     H=CHI/(N*((EV1/EV2)+(EV2/EV1)-2.))
24     A=SQRT(H/(1.-H))
25     S=XSIN/XCOS
26     SMAX=(S-A)/(1.+S*A)
27     SMIN=(S+A)/(1.-S*A)
28     RETURN
29     END

```

```

1      SUBROUTINE CORREL(FILENAME,M,Y,X,A,NDIM)
2      C
3      C COMPUTES CORRELATION MATRIX (AFTER COOLEY & LOHNES, 1971)
4      C
5      C FILENAME = FILENAME FOR INPUT
6      C M = NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR
7      C X = SAMPLE VECTOR (INPUT); STANDARD DEVIATIONS (OUTPUT)
8      C Y = MEANS OR CENTROID VECTOR
9      C A = CORRELATION MATRIX
10     C
11     DIMENSION X(NDIM),Y(NDIM),A(NDIM,NDIM)
12     CHARACTER FILENAME*13
13     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
14     DO 1 J=1,M
15     Y(J)=0.
16     DO 1 K=1,M
17     1 A(J,K)=0.
18     N=0
19     2 READ(1,*,END=4) T,T,(X(J),J=1,M)
20     N=N+1
21     DO 3 J=1,M
22     Y(J)=Y(J)+X(J)
23     DO 3 K=1,M
24     3 A(J,K)=A(J,K)+X(J)*X(K)
25     GO TO 2
26     4 CLOSE(UNIT=1)
27     TYPE 5, N
28     5 FORMAT(///I6,' SAMPLES COUNTED'///)
29     C Y(J) NOW CONTAINS RAW SUMS; A(J,K) CONTAINS RAW S.S.C.P.
30     DO 6 J=1,M
31     DO 6 K=1,M
32     6 A(J,K)=(A(J,K)-Y(J)*Y(K)/N)/N
33     DO 7 J=1,M
34     Y(J)=Y(J)/N
35     7 X(J)=SQRT(A(J,J))
36     DO 8 J=1,M
37     DO 8 K=1,M
38     IF(X(J)*X(K).EQ.0) THEN
39     A(J,K)=0.
40     ELSE,
41     A(J,K)=A(J,K)/(X(J)*X(K))
42     END IF
43     8 CONTINUE
44     RETURN
45     END

```



```

1      SUBROUTINE COUNTERROR(TOTAL,PERCNT,PROB,PERMAX,PERMIN)
2      C
3      C COMPUTES INTERVAL ESTIMATES OF SPECIFIED CONFIDENCE FOR PROPORTION
4      C OF ONE COUNTING CATEGORY IN A SAMPLE. FORMULAE AFTER MOSIMANN(1965),
5      C P. 642-643.
6      C
7      IF(PERCNT.EQ.0.OR.TOTAL.EQ.0) THEN
8          PERMAX=0.
9          PERMIN=0.
10         RETURN
11     END IF
12     SIGN=PERCNT/ABS(PERCNT)
13     P=ABS(PERCNT)/100.
14     IF(PROB.LE.99) A=2.58
15     IF(PROB.LE.95) A=1.96
16     IF(PROB.LE.90) A=1.64
17     IF(PROB.LE.85) A=1.44
18     IF(PROB.LE.80) A=1.28
19     IF(PROB.LE.75) A=1.15
20     IF(PROB.LE.68) A=1.00
21     IF(PROB.LE.50) A=.674
22     T=TOTAL
23     PERMAX=P+(A*A)/(2*T)+A*SQRT(P*(1.-P)/T+(A*A)/(4*T*T))
24     PERMAX=PERMAX/(1+(A*A)/T)
25     PERMAX=PERMAX*SIGN*100.
26     PERMIN=P+(A*A)/(2*T)-A*SQRT(P*(1.-P)/T+(A*A)/(4*T*T))
27     PERMIN=PERMIN/(1+(A*A)/T)
28     PERMIN=PERMIN*SIGN*100.
29     RETURN
30     END

```

```

1      SUBROUTINE COVAR(FILENAME,M,Y,X,A,NDIM)
2      C
3      C COMPUTES VARIANCE-COVARIANCE MATRIX (AFTER COOLEY & LOHNES, 1971)
4      C
5      C FILENAME = FILENAME FOR INPUT
6      C M      = NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR
7      C X      = SAMPLE VECTOR (INPUT); STANDARD DEVIATIONS (OUTPUT)
8      C Y      = MEANS OR CENTROID VECTOR
9      C A      = VARIANCE-COVARIANCE (DISPERSION) MATRIX
10     C
11     DIMENSION X(NDIM),Y(NDIM),A(NDIM,NDIM)
12     CHARACTER FILENAME*13
13     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
14     DO 1 J=1,M
15     Y(J)=0.
16     DO 1 K=1,M
17     1 A(J,K)=0.
18     N=0
19     2 READ(1,*,END=4) T,T,(X(J),J=1,M)
20     N=N+1
21     DO 3 J=1,M
22     Y(J)=Y(J)+X(J)
23     DO 3 K=1,M
24     3 A(J,K)=A(J,K)+X(J)*X(K)
25     GO TO 2
26     4 CLOSE(UNIT=1)
27     TYPE 2, N
28     5 FORMAT('///16,0  SAMPLES COUNTED'///)
29     C Y(J) NOW CONTAINS RAW SUMS; A(J,K) CONTAINS RAW S.S.C.P.
30     DO 6 J=1,M
31     DO 6 K=1,M
32     6 A(J,K)=(A(J,K)-Y(J)*Y(K)/N)/N
33     DO 7 J=1,M
34     Y(J)=Y(J)/N
35     7 X(J)=SQRT(A(J,J))
36     RETURN
37     END

```



```

1      SUBROUTINE CROSSPROD(FILENAME,M,Y,A,MDIM)
2      C
3      C COMPUTES MATRIX OF RAW SUMS OF SQUARES AND CROSS-PRODUCTS
4      C (AFTER COBLEY & LOHNES 1971)
5      C
6      C FILENAME = FILENAME FOR INPUT
7      C M      = NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR
8      C X      = SAMPLE VECTOR (INPUT)
9      C Y      = VECTOR OF RAW SUMS
10     C A      = CROSS-PRODUCT MATRIX
11     C
12     DIMENSION X(1),Y(MDIM),A(MDIM,MDIM)
13     CHARACTER FILENAME*13
14     OPEN(UNIT=1,NAME=FILENAME,TYPE='OLD')
15     DO 1 J=1,M
16     Y(J)=0.
17     DO 1 K=1,M
18     1 A(J,K)=0.
19     N=0
20     2 READ(1,*,END=4) D,T,(X(J),J=1,M)
21     N=N+1
22     DO 3 J=1,M
23     Y(J)=Y(J)+X(J)
24     DO 3 K=1,M
25     3 A(J,K)=A(J,K)+X(J)*X(K)
26     GO TO 2
27     4 CLOSE(UNIT=1)
28     TYPE 5, N
29     5 FORMAT('///16,' SAMPLES COUNTED'///)
30     C Y(J) NOW CONTAINS RAW SUMS; A(J,K) CONTAINS RAW S.S.C.P.
31     RETURN
32     END

```

```

1      SUBROUTINE CURVELINE(Y,YO,YI,X,XO,XI,N,WIDTH)
2      C
3      C PLOTS CURVED LINE (SPLINE) THROUGH X-VALUES AGAINST INCREASING Y
4      C
5      C REQUIRES SUBROUTINES SPACER, CHECKDUP, SP_LINE
6      C
7      DIMENSION X(1),Y(1),Z(5000)
8      START=Y(1)
9      SPACE=(Y(N)-Y(1))/(N*10)
10     CALL SPACER(N,M,Y,X,Z,SPACE,START,WIDTH)
11     CALL PLOT((Z(1)-XO)/XI,-(Y(1)-YO)/YI,3)
12     DO 1 I=1,M
13     IF(XI/ABS(XI).LT.0.AND.Z(I).GT.XO) Z(I)=XO
14     1 IF(XI/ABS(XI).GT.0.AND.Z(I).LT.XO) Z(I)=XO
15     DO 2 I=2,M
16     2 CALL PLOT((Z(I)-XO)/XI,-(Y(1)+SPACE*(I-1)-YO)/YI,2)
17     RETURN
18     END

```

```

1      SUBROUTINE DIAG(A,V,UTILD)
2      C
3      C DIAGONALIZES A 2X2 SYMMETRIC MATRIX, APPLYING A QUADRATIC SOLUTION
4      C FOR ITS EIGENVALUES AND EIGENVECTORS
5      C
6      DIMENSION A(4),V(2),UTILD(4)
7      B=A(1)+A(4)
8      C=A(1)*A(4)-A(2)*A(3)
9      V(1)=(B+SQRT(B**2-4.*C))/2.
10     V(2)=(B-SQRT(B**2-4.*C))/2.
11     IF(V(1).NE.V(2)) GO TO 5
12     UTILD(1)=0.
13     UTILD(2)=0.
14     UTILD(3)=0.
15     UTILD(4)=0.
16     TYPE *, ' ELLIPSE CIRCULAIRE...PLAN PROPRE !!'
17     GO TO 6
18     5 UTILD(1)=-A(2)/SQRT((A(2)**2)+((A(1)-V(1))**2))
19     UTILD(2)=(A(1)-V(1))/SQRT((A(2)**2)+((A(1)-V(1))**2))
20     UTILD(3)=(A(4)-V(2))/SQRT((A(3)**2)+((A(4)-V(2))**2))
21     UTILD(4)=-A(3)/SQRT((A(3)**2)+((A(4)-V(2))**2))
22     6 CONTINUE
23     TYPE 10, A(1),A(2),V(1),UTILD(1),UTILD(2)
24     10 FORMAT(2X,'( ',E12.5,3X,E12.5,' )',2X,'LAMBDA 1=',E12.5,2X,'U1=',
25     + F8.4,2X,'U2=',F8.4)
26     TYPE 12, A(3),A(4),V(2),UTILD(3),UTILD(4)
27     12 FORMAT(2X,'( ',E12.5,3X,E12.5,' )',2X,'LAMBDA 2=',E12.5,2X,'V1=',
28     + F8.4,2X,'V2=',F8.4)
29     RETURN
30     END

```



```

1      SUBROUTINE DOT(XC,YC,RAD)
2      C
3      C PLOTS A CIRCLE GIVEN ITS RADIUS AND THE X,Y COORDINATES OF ITS
4      C CENTER (ALL IN INCHES)
5      C
6          R = ABS(RAD)
7          X = XC + R
8          Y = YC
9          CALL PLOT(X,Y,3)
10         TU = 0.0
11         C = X - R
12         DO 10 I=1,32
13             TU = TU + 0.19635
14             X = R*COS(TU) + C
15             Y = R*SIN(TU) + YC
16             CALL PLOT(X,Y,2)
17         10 CONTINUE
18         RETURN
19         END

```

```

1      SUBROUTINE DSCORES(X,XM,V,E,N,M,NDIM,MDIM,F)
2      C
3      C COMPUTES D-MODE PRINCIPAL COMPONENT SCORES FROM RESULTS OF
4      C   DIAGONALIZING A VARIANCE-COVARIANCE OR DISERSION (D) MATRIX
5      C
6      C   X      = MATRIX OF RAW SAMPLE VECTORS X(N,M)
7      C   XM     = VECTOR OF MEANS XM(M)
8      C   E      = VECTOR OF EIGENVALUES E(M)
9      C   V      = MATRIX OF UNIT EIGENVECTORS OF D-MATRIX V(M,M)
10     C   N      = NUMBER OF SAMPLE VECTORS IN X
11     C   M      = NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR OF X
12     C   NDIM   = NUMBER OF ROWS DIMENSIONED FOR X IN CALLING ROUTINE
13     C   MDIM   = NUMBER OF COLUMNS DIMENSIONED FOR X IN CALLING ROUTINE
14     C   F      = MATRIX OF "FACTOR SCORES" F(N,M)
15     C
16         DIMENSION X(NDIM,MDIM),V(MDIM,MDIM),E(MDIM),F(NDIM,MDIM)
17         DIMENSION XM(MDIM)
18     C STANDARDIZE VARIABLES TO ZERO MEAN
19         DO 1 I=1,N
20             DO 1 J=1,M
21                 1 X(I,J)=X(I,J)-XM(J)
22     C D-SCORES=STANDARDIZED VARIABLES*EIGENVECTORS*1/SQRT(EIGENVALUES)
23         DO 3 I=1,N
24             DO 3 J=1,M
25                 F(I,J)=0.
26                 DO 3 K=1,M
27                     IF(E(J).LE.0.) GO TO 3
28                     F(I,J)= F(I,J)+X(I,K)*V(K,J)/SQRT(E(J))
29             3 CONTINUE
30         RETURN
31     END

```

```

1      SUBROUTINE ELIPSE(XO,YO,AA,BB,COTET,SITET,XMIN,YMIN,S,XMAX,YMAX)
2      C
3      C PLOTS AN ELLIPSE
4      C
5      C      XO, YO          COORDINATES OF THE THE CENTER
6      C      AA, BB         MAJOR AND MINOR AXES
7      C      COTET, SITET   DIRECTION SINES AND COSINES
8      C      XMIN, YMIN
9      C      S
10     C      XMAX, YMAX      PLOT DIMENSIONS AND SCALING PARAMETERS
11     C
12     SCALE=S
13     TETA=1.5708
14     IF(COTET.NE.0.) TETA=ATAN(SITET/COTET)
15     XD=(XO-XMIN)/SCALE
16     YD=(YO-YMIN)/SCALE
17     AA=AA/SCALE
18     BB=BB/SCALE
19     X1=XO+AA*COS(TETA)
20     Y1=YO+AA*SIN(TETA)
21     XXMAX=(XMAX-XMIN)/SCALE
22     YYMAX=(YMAX-YMIN)/SCALE
23     IF(X1.GT.XXMAX) X1=XXMAX
24     IF(Y1.GT.YYMAX) Y1=YYMAX
25     IF(X1.LT.0.) X1=0.
26     IF(Y1.LT.0.) Y1=0.
27     CALL PLOT(X1,Y1,3)
28     DO 1 I=1,361,4
29     XE=AA*COS(FLOAT(I)*3.1416/180.)
30     YE=BB*SIN(FLOAT(I)*3.1416/180.)
31     XE1=XD+XE*COS(TETA)-YE*SIN(TETA)
32     YE1=YD+XE*SIN(TETA)+YE*COS(TETA)
33     IF(XE1.GT.XXMAX) XE1=XXMAX
34     IF(XE1.LT.0.) XE1=0.
35     IF(YE1.GT.YYMAX) YE1=YYMAX
36     IF(YE1.LT.0.) YE1=0.
37     1 CALL PLOT(XE1,YE1,2)
38     T=(TETA/3.1416)*180.
39     CALL SYMBOL(XD,YD,.1,3,T,-1)
40     RETURN
41     END

```



```

1      SUBROUTINE HATCHIT(Y,YORIG,YINC,X,XORIG,XINC,NS,XC)
2      C
3      C HATCHES IN PLOT
4      C
5      C X IS VECTOR OF VARIABLE PLOTTED
6      C X AND Y-INCREMENTS ARE SPECIFIED IN UNITS PER INCH
7      C XC IS MIDDLE OF PLOT ABOUT WHICH IT IS HATCHED
8      C
9      C
10     DIMENSION X(1),Y(1)
11     CALL NEWPEN(2)
12     DO 1 I=1,NS
13     YY=(Y(I)-YORIG)/YINC
14     XX=(X(I)-XORIG)/XINC
15     XM=(XC-XORIG)/XINC
16     CALL SYMBOL(XM,-YY,.14,13,0.,-1)
17     CALL PLOT(XX,-YY,2)
18     1 CALL SYMBOL(XM,-YY,.14,13,0.,-2)
19     CALL NEWPEN(1)
20     RETURN
21     END

```

```

1  SUBROUTINE HDN (MVAR, MDIM, NVECT, R, E, V, A, B, C, D)
2  C  EIGENVALUES AND EIGENVECTORS OF A SYMMETRIC MATRIX, ALGORITHM
3  C  BY HOUSEHOLDER, ORTEGA, AND WILKINSON. ORIGINAL PROGRAM BY
4  C  DAVID W. MATULA UNDER THE DIRECTION OF WILLIAM MEREDITH,
5  C  OF CALIFORNIA, BERKELEY, 1962.
6  C  MODIFIED BY P. R. LOHNES, PROJECT TALENT, 1966.
7  C
8  C  M IS THE ORDER OF THE INPUT MATRIX, R.
9  C  MD IS THE DIMENSIONED SIZE OF R IN THE MAIN PROGRAM.
10 C  NV IS THE NUMBER OF EIGENVECTORS TO BE COMPUTED.
11 C  E IS THE VECTOR IN WHICH THE EIGENVALUES ARE RETURNED.
12 C  V IS THE MATRIX IN WHICH THE EIGENVECTORS ARE RETURNED.
13 C  THE EIGENVECTORS ARE STORED AS COLUMNS IN V.
14 C  A, B, C, AND D ARE WORKSPACE VECTORS.
15 C
16 C  DIMENSION R(1), E(1), V(1), A(1), B(1), C(1), D(1)
17 C
18 C  M = MVAR
19 C  MD = MDIM
20 C  NV = NVECT
21 C  IF (M - 1) 100, 97, 96
22 96 M1 = M - 1
23 C  TRI-DIAGONALIZE THE MATRIX.
24 C  M2 = M1 * MD + M
25 C  M3 = M2 - MD
26 C  M4 = MD + 1
27 C  L = 0
28 C  DO 1 I = 1, M2, M4
29 C  L = L + 1
30 1 A(L) = R(1)
31 B(1) = 0.0
32 C  IF (M - 2) 13, 2, 3
33 3 KK = 0
34 C  DO 15 K = 2, M1
35 C  KL = KK + K
36 C  KU = KK + M
37 C  KJ = K + 1
38 C  SUM = 0.0
39 C  DO 4 J = KL, KU
40 4 SUM = SUM + R(J) **2
41 C  S = SQRT(SUM)
42 C  Z = R(KL)
43 C  B(K) = SIGN (S, -Z)
44 C  S = 1.0 / S
45 C  C(K) = SQRT (ABS (Z) * S + 1.0)
46 C  X = SIGN (S / C(K), Z)
47 C  R(KL) = C(K)
48 C  DO 5 I = KJ, M
49 C  JJ = I + KK
50 C  C(I) = X * R(JJ)
51 5 R(JJ) = C(I)
52 C  DO 8 J = K, M
53 C  JJ = J + 1
54 C  D(J) = 0.0
55 C  L = KK + J
56 C  DO 6 I = K, J
57 C  L = L + MD
58 6 D(J) = D(J) + R(L) * C(I)
59 C  IF (JJ - M) 7, 7, 9
60 7 DO 8 I = JJ, M

```

```

52      8      D(J) = D(J) + R(L) * C(I)
53      9      X = 0.0
54      DO 10 J = K, M
55      10      X = X + C(J) * D(J)
56      X = .50 * X
57      DO 11 I = K, M
58      11      D(I) = X * C(I) - D(I)
59      LL = KK
60      KK = KK + MD
61      DO 15 I = K, M
62      15      LL = LL + MD
63      DO 15 J = I, M
64      L = LL + J
65      15      R(L) = R(L) + D(I) * C(J) + D(J) * C(I)
66      L = L
67      DO 12 I = 1, M
68      X = A(I)
69      A(I) = R(L)
70      R(L) = X
71      12      L = L + M4
72      2      B(M) = R(M3)
73      C      COMPUTE EIGENVALUES.
74      13      BD = ABS (A(1))
75      DO 14 I = 2, M
76      14      BD = AMAX1(BD, ABS (A(I)) + B(I)**2)
77      BD = BD + 1.0
78      DO 16 I = 1, M
79      A(I) = A(I) / BD
80      B(I) = B(I) / BD
81      D(I) = 1.0
82      16      E(I) = -1.0
83      DO 37 K = 1, M
84      17      IF ((D(K)-E(K))/AMAX1(ABS (D(K)),ABS (E(K)),1.0E-9)-1.0E-5)37,37,18
85      18      X = (D(K) + E(K)) * .50
86      IS2 = 1
87      C(I) = A(I) - X
88      IF (C(I)) 19, 20, 20
89      19      IS1 = -1
90      N = 0
91      GO TO 21
92      20      IS1 = 1
93      N = 1
94      21      DO 31 I = 2, M
95      IF (B(I)) 22, 26, 22
96      22      IF (B(I-1)) 23, 27, 23
97      23      IF (ABS (C(I-1))+ABS (C(I-2))-1.0E-15) 24, 25, 25
98      24      C(I-1) = C(I-1) * 1.0E15
99      C(I-2) = C(I-2) * 1.0E15
100     25      C(I) = (A(I)-X)*C(I-1)-B(I)**2 * C(I-2)
101     GO TO 28
102     26      C(I) = (A(I)-X) * IS1
103     GO TO 28
104     27      C(I) = (A(I)-X) * C(I-1) - SIGN (B(I)**2, FLOAT(IS2))
105     28      IS2 = IS1
106     IF (C(I)) 29, 30, 29
107     29      IS1 = IABS(IS1)
108     IF (C(I).LT.0.) IS1=-IS1
109     IF (IS2+IS1) 30, 31, 30
110     30      N = N + 1
111     31      CONTINUE

```



```

123      IF (N - K) 34, 32, 32
124      32  DO 33  J = K, N
125      33  D(J) = X
126      34  N = N + 1
127      IF (M - N) 17, 35, 35
128      35  DO 36  J = N, M
129      IF (X - E(J)) 17, 17, 36
130      36  E(J) = X
131      GO TO 17
132      37  CONTINUE
133      DO 38  I = 1, M
134      A(I) = A(I) * BD
135      B(I) = B(I) * BD
136      38  C(I) = (D(I) + E(I)) * BD * .50
137      M1 = M
138      K = 1
139      39  I = 1
140      40  DO 43  J = 1, M1
141      IF (I - J) 41, 43, 41
142      41  IF (C(I) - C(J)) 43, 43, 42
143      42  I = J
144      GO TO 40
145      43  CONTINUE
146      E(K) = C(I)
147      K = K + 1
148      M1 = M1 - 1
149      IF (I - M1 - 1) 44, 46, 46
150      44  DO 45  M2 = 1, M1
151      45  C(M2) = C(M2+1)
152      46  IF (M1 - 1) 47, 47, 39
153      47  E(K) = C(1)
154      IF (ISIGN (1, NV)) 79, 76, 76
155      76  DO 77  I = 1, M
156      77  C(I) = E(1)
157      J = M
158      DO 78  I = 1, M
159      E(I) = C(J)
160      78  J = J - 1
161      79  CONTINUE
162  C
163  C  DECIDE WHETHER TO COMPUTE EIGENVECTORS, AND IF SO, HOW MANY
164  C
165      IF (NV) 48, 99, 48
166      48  KX = IABS(NV)
167      J = 1
168      DO 53  INV = 1, KX
169      X = A(1) - E(INV)
170      Y = B(2)
171      M1 = M - 1
172      DO 54  I = 1, M1
173      IJ = J + I - 1
174      IF (ABS (X) - ABS (B(I+1))) 49, 51, 53
175      49  C(I) = B(I+1)
176      D(I) = A(I+1) - E(INV)
177      V(IJ) = B(I+2)
178      Z = -X / C(I)
179      X = Z * D(I) + Y
180      IF (M1 - I) 50, 54, 50
181      50  Y = Z * V(IJ)
182      GO TO 54

```

```

134      52      X = 1.0E-10
135      53      C(I) = X
136              D(I) = Y
137              V(IJ) = 0.0
138              X = A(I+1) - (B(I+1) / X * Y + E(INV))
139              Y = B(I+2)
140      54      CONTINUE
141              MJ = M + J - 1
142              IF (X) 56, 50, 56
143      56      V(MJ) = 1.0 / X
144      57      I = M1
145              IJ = J + I - 1
146              V(IJ) = (1.0 - D(I) * V(MJ)) / C(I)
147              X = V(MJ)**2 + V(IJ)**2
148      58      I = I - 1
149              IJ = J + I - 1
150              IF (I) 59, 61, 59
151      59      V(IJ) = (1.0 - (D(I) * V(IJ+1) + V(IJ) * V(IJ+2))) / C(I)
152              X = X + V(IJ)**2
153              GO TO 58
154      60      V(MJ) = 1.0E10
155              GO TO 57
156      61      X = SQRT(X)
157              DO 62 I = 1, M
158              IJ = J + I - 1
159      62      V(IJ) = V(IJ) / X
160              J1 = M1 * MD - MD
161              K = M
162              GO TO 66
163      63      K = K - 1
164              J1 = J1 - MD
165              Y = 0.0
166              DO 64 I = K, M
167              IJ = J + I - 1
168              L = J1 + I
169      64      Y = Y + V(IJ) * R(L)
170              DO 65 I = K, M
171              IJ = J + I - 1
172              L = J1 + I
173      65      V(IJ) = V(IJ) - Y * R(L)
174      66      IF (J1) 63, 67, 63
175      67      NPLUS = 0
176              NMIN = 0
177              DO 70 I = 1, M
178              IJ = J + I - 1
179              IF (V(IJ)) 68, 69, 69
180      68      NMIN = NMIN + 1
181              GO TO 70
182      69      NPLUS = NPLUS + 1
183      70      CONTINUE
184              IF (NPLUS - NMIN) 71, 73, 73
185      71      DO 72 I = 1, M
186              IJ = J + I - 1
187      72      V(IJ) = -V(IJ)
188      73      CONTINUE
189      98      J = J + MD
190      C      RESTORE THE INPUT MATRIX.
191      99      MD1 = MD + 1
192              JJ = MD1
193              M1 = M * MJ

```

```
245      K = I
246      DO 74 J = JJ, M1, MD
247      R(K) = R(J)
248      74  K = K + 1
249      75  JJ = JJ + MD1
250      GO TO 100
251      97  E(1) = R(1)
252      V(1) = 1.0
253      100 RETURN
254      END
```



```

1      SUBROUTINE MATINV(A,M,MDIM,DET)
2  C   INVERSE AND DETERMINANT OF A(M,M) BY THE GAUSS-JORDAN METHOD,
3  C   M IS THE ORDER OF THE SQUARE MATRIX, A.
4  C   MDIM IS THE MAXIMUM M FOR A AS DIMENSIONED IN CALLING ROUTINE.
5  C   A-INVERSE REPLACES A
6  C   DETERMINANT OF A IS PLACED IN DET.
7  C   P.R. LUHNES, PROJECT TALENT, 1966 (COOLEY & LUHNES, 1973)
8  C
9      DIMENSION A(MDIM,MDIM),IPVT(100),PVT(100),IND(100,2)
10     DET=1.
11     DO 1 J=1,M
12     1 IPVT(J)=0
13     DO 10 I=1,M
14  C   SEARCH FOR THE PIVOT ELEMENT
15     AMAX=0.
16     DO 5 J=1,M
17     IF(IPVT(J)-1) 2,5,2
18     2 DO 5 K=1,M
19     IF(IPVT(K)-1) 3,5,20
20     3 IF(ABS(AMAX)-ABS(A(J,K))) 4,5,5
21     4 IROW=J
22     ICOL=K
23     AMAX=A(J,K)
24     5 CONTINUE
25     IPVT(ICOL)=IPVT(ICOL)+1
26  C   INTERCHANGE THE ROWS TO PUT THE PIVOT ELEMENT ON THE DIAGONAL
27     IF(IROW-ICOL) 6,8,6
28     6 DET=-DET
29     DO 7 L=1,M
30     SWAP=A(IROW,L)
31     A(IROW,L)=A(ICOL,L)
32     7 A(ICOL,L)=SWAP
33     8 IND(I,1)=IROW
34     IND(I,2)=ICOL
35     PVT(I)=A(ICOL,ICOL)
36     DET=DET*PVT(I)
37  C   DIVIDE THE PIVOT ROW BY THE PIVOT ELEMENT
38     A(ICOL,ICOL)=1.
39     DO 9 L=1,M
40     9 A(ICOL,L)=A(ICOL,L)/PVT(I)
41  C   REDUCE NON-PIVOT ROWS
42     DO 10 L1=1,M
43     IF(L1-ICOL) 11,10,11
44     11 SWAP=A(L1,ICOL)
45     A(L1,ICOL)=0.
46     DO 12 L=1,M
47     12 A(L1,L)=A(L1,L)-A(ICOL,L)*SWAP
48     10 CONTINUE
49  C   INTERCHANGE THE COLUMNS
50     DO 20 I=1,M
51     L=M+1-I
52     IF(IND(L,1)-IND(L,2)) 13,20,13
53     13 IROW=IND(L,1)
54     ICOL=IND(L,2)
55     DO 20 K=1,M
56     SWAP=A(K,IROW)
57     A(K,IROW)=A(K,ICOL)
58     A(K,ICOL)=SWAP
59     20 CONTINUE
60     RETURN

```

```
1      SUBROUTINE MATPRINT(R,M,NDIM,IOUT)
```

```
2      C
3      C PRINTS A SQUARE MATRIX
4      C
5      C R(I,J) = MATRIX TO BE PRINTED
6      C M      = ORDER OF MATRIX
7      C NDIM   = DIMENSION OF R IN CALLING PROGRAM
8      C IOUT   = UNIT NUMBER FOR PRINTED OUTPUT
9      C
10     C AFTER COOLEY & LOHNS (1973)
11     C
12     DIMENSION R(NDIM,NDIM),J(100)
13     PARAMETER L1=9,M=10
14     J1=0
15     J2=0
16     JSEC=0
17     DO 1 I=1,M
18     1 J(I)=I
19     2 J1=J2+1
20     J2=J1+L1
21     IF(J2-M) 4,4,3
22     3 J2=M
23     4 JSEC=JSEC+1
24     WRITE(IOUT,5) JSEC
25     5 FORMAT(/' SECTION',I4/)
26     WRITE(IOUT,6) (J(I),I=J1,J2)
27     6 FORMAT(' ROW',10I12)
28     DO 7 I=1,M
29     7 WRITE(IOUT,8) I,(R(I,K),K=J1,J2)
30     8 FORMAT(1X,I3,10E12.4)
31     IF(J2-M) 2,9,9
32     9 RETURN
33     END
```

```

1      SUBROUTINE MULTIPLY(X, IROWS, IDIM, JCOLS, JDIM, Y, KCOLS, KDIM, Z)
2  C <MULTIPLY> TWO MATRICES
3  C   X = 1ST MATRIX (IN)
4  C   I = ROWS IN X AND Z MATRICES
5  C   IDIM = MAX. ROWS IN X & IN Y AS DIMENSIONED IN CALLING ROUTINE (IN)
6  C   J = COLUMNS IN X MATRIX, ROWS IN Y MATRIX
7  C   JDIM = MAX. COLUMNS IN X & MAX. ROWS IN Y IN CALLING ROUTINE (IN)
8  C   Y = 2ND MATRIX (IN)
9  C   K = COLUMNS IN Y AND Z MATRICES
10 C   KDIM = MAX. COLUMNS IN Y & Z AS DIMENSIONED IN CALLING ROUTINE (IN)
11 C
12     DIMENSION X(IDIM,JDIM),Y(JDIM,KDIM),Z(IDIM,KDIM)
13     DO 1 I=1,IROWS
14     DO 1 K=1,KCOLS
15     Z(I,K)=0.
16     DO 1 J=1,JCOLS
17     1 Z(I,K)=X(I,J)*Y(J,K)+Z(I,K)
18     RETURN
19     END

```



```

1      SUBROUTINE NORMALIZE(X,M,XL)
2
3      C  NORMALIZES SAMPLE VECTOR TO UNIT LENGTH
4      C
5      C  X = SAMPLE VECTOR
6      C  M = NUMBER OF ELEMENTS IN SAMPLE VECTOR
7      C  XL = LENGTH OF RAW SAMPLE VECTOR BEFORE ADJUSTMENT
8      C
9      DIMENSION X(1)
10     SSQ=0.
11     DO 1 I=1,M
12 1    SSQ=SSQ+X(I)*X(I)
13     XL=SQRT(SSQ)
14     IF(SSQ.LE.0.) GO TO 3
15     DO 2 I=1,M
16 2    X(I)=X(I)/XL
17     RETURN
18 3    DO 4 I=1,M
19 4    X(I)=0.
20     RETURN
21     END

```

```

1      SUBROUTINE OMEGA(T,S,DEPTH,ALK,TCO2,H,OM)
2      C <OMEGA> COMPUTES CARBONATE SATURATION
3      C
4      C T      = IN SITU TEMPERATURE IN DEGREES CENTIGRADE
5      C S      = SALINITY
6      C DEPTH  = WATER DEPTH IN KM
7      C ALK    = ALKALINITY
8      C TCO2   = TOTAL CO2
9      C H      = CARBONATE ION CONCENTRATION IN MMOLE/L
10     C OM     = CARBONATE SATURATION (1-OMEGA)
11     C
12     C AFTER A PROGRAM WRITTEN IN BASIC BY ROBIN KEIR
13     C PRESSURE COEFFICIENT IS SPECIFIED AS -36 CC/MOLE
14     C
15     REAL L,K1,K
16     L=2.303
17     P1=3.44-.00553*C-.004*T*(1.+.02*T)
18     P2=.52-.00221*C-.002*T
19     B3=EXP(-L*P1)
20     B4=EXP(-L*P2)
21     B1=B3*EXP((-7.8-.045*T)*Z/(.8314*(273.+T)))
22     B2=B4*EXP((-11.1-.055*T)*Z/(.8314*(273.+T)))
23     K1=(36.+.2*T)*Z/(.8314*(273+T))
24     K=(.561-.0032*T)*EXP(K1)
25     B=.01176*C
26     C1=.296*C
27     P=1.-B2-4*B1-31/B2+4*B1*B2
28     QJ=1.-4*B1*B2-B1/B2
29     Q1=B*(1.-31/B2-4*B1)
30     RD=1.-4*B1
31     R1=B1*(1.-B2)
32     R3=B*B1
33     SU=B1*B2
34     Q=QJ*Y-P*X+Q1
35     R=-B2*(RD*X-Y)*Y-R1*X*X+R3*(X+2*Y)
36     S=-SU*X*X*Y
37     W=0.
38     DO 1 I=1,2
39     U=R-3*P*W*W
40     V=U+3*P*W
41     1 W=(-U+SQRT(U*U-4*V*(S+P*W*W*3)))/(2*V)
42     OM=1.-C1*W/K
43     RETURN
44     END

```

```

1      SUBROUTINE PERCENT(X,M,T)
2      C
3      C   CONVERTS RAW SAMPLE TO PERCENT; RETURNS TOTAL NUMBER OF INDIVIDUALS
4      C
5      C   X = RAW SAMPLE VECTOR AND SAMPLE VECTOR EXPRESSED AS PERCENT
6      C   M = NUMBER OF ELEMENTS IN THE SAMPLE VECTOR
7      C   T = TOTAL NUMBER OF INDIVIDUALS COUNTED IN SAMPLE VECTOR
8      C
9      DIMENSION X(1)
10     T=0.
11     DO 1 I=1,M
12     1 T=T+ABS(X(I))
13     IF(T.LE.0.) GO TO 3
14     DO 2 I=1,M
15     2 X(I)=(X(I)/T)*100.
16     RETURN
17     3 DO 4 I=1,M
18     4 X(I)=0.
19     RETURN
20     END

```



```

1      SUBROUTINE POINTPLOT(Y,YORIG,YINC,X,XORIG,XINC,NS)
2      C
3      C PLOTS X-VALUES AS POINTS AGAINST INCREASING Y
4      C
5      C X AND Y-INCREMENTS ARE SPECIFIED IN UNITS PER INCH
6      C
7      DIMENSION X(1),Y(1)
8      DO 1 I=1,NS
9      YY=(Y(I)-YORIG)/YINC
10     XX=(X(I)-XORIG)/XINC
11     CALL SYMBOL(X,-YY,.07,1,0.,-1)
12     CALL SYMBOL(X,-YY,.07,1,0.,-2)
13     CALL SYMBOL(X,-YY,.07,1,0.,-2)
14     1 CALL SYMBOL(X,-YY,.07,1,0.,-2)
15     RETURN
16     END

```

```
1      SUBROUTINE POLAR(X,Y,THETA,R)
2      C
3      C CONVERTS X,Y RECTANGULAR COORDINATES INTO POLAR COORDINATES
4      C
5      PI=3.141592654
6      R=SQRT(X*X+Y*Y)
7      THETA=ATAN(Y/X)
8      IF(THETA.GT.0.AND.X.LT.0)THEN
9          THETA=THETA+PI
10         END IF
11      IF(THETA.LT.0.AND.X.LT.0)THEN
12          THETA=THETA+PI
13      RETURN
14      END IF
15      IF(THETA.LT.0.AND.X.GT.0) THETA=THETA+2*PI
16      RETURN
17      END
```

```

1      SUBROUTINE PRINCO(V,E,N,NDIM)
2      C
3      C SCALES EIGENVECTORS TO PRINCIPAL COMPONENTS
4      C
5      C V      = MATRIX OF EIGENVECTORS V(N,N)
6      C E      = VECTOR OF EIGENVALUES E(N)
7      C N      = NUMBER OF ELEMENTS IN EACH EIGENVECTOR
8      C NDIM = ORDER OF V AS DIMENSIONED IN CALLING ROUTINE
9      C
10     DIMENSION V(NDIM,NDIM),E(NDIM)
11     C PRINCIPAL COMPONENTS = EIGENVECTORS * SQRT(EIGENVALUES)
12     DO 1 J=1,N
13     DO 1 I=1,N
14     IF(E(J).LE.0.) V(I,J)=0.
15     1 IF(E(J).GT.0.) V(I,J)=V(I,J)*SQRT(E(J))
16     RETURN
17     END

```



```

1      SUBROUTINE QMATRIX(X,Q,N,M,NDIM,MDIM)
2      C
3      C COMPUTE COSINE-THETA (Q) MATRIX
4      C
5      C   X       = MATRIX OF RAW SAMPLE VECTORS
6      C   Q       = COSINE-THETA MATRIX Q(N,N)
7      C   N       = NUMBER OF SAMPLE VECTORS IN X
8      C   M       = NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR OF X
9      C   NDIM    = NUMBER OF ROWS DIMENSIONED FOR X IN CALLING ROUTINE
10     C   MDIM    = NUMBER OF COLUMNS DIMENSIONED FOR X IN CALLING ROUTINE
11     C
12     C REQUIRES SUBROUTINE NORMALIZE
13     C
14     DIMENSION X(NDIM,MDIM),Q(NDIM,NDIM),Y(1000)
15     C ADJUST RAW SAMPLE VECTORS TO UNIT LENGTH
16     DO 2 I=1,N
17     DO 1 J=1,M
18     1 Y(J)=X(I,J)
19     CALL NORMALIZE(Y,M,T)
20     DO 2 J=1,M
21     2 X(I,J)=Y(J)
22     C COMPUTE Q
23     DO 4 I=1,N
24     DO 3 K=1,M
25     3 Y(K)=X(I,K)
26     DO 4 J=1,N
27     Q(I,J)=0.
28     DO 4 K=1,M
29     4 Q(I,J)=X(J,K)*Y(K)+Q(I,J)
30     DO 5 J=1,N
31     DO 5 I=J,N
32     5 Q(I,J)=Q(J,I)
33     RETURN
34     END

```

```

1      SUBROUTINE QSCORES(X,V,E,N,M,NDIM,MDIM,F)
2      C
3      C COMPUTES Q-MODE PRINCIPAL FACTOR SCORES FROM RESULTS OF DIAGONALIZING
4      C   A COSINE-THETA (Q) MATRIX
5      C
6      C   X      = MATRIX OF RAW SAMPLE VECTORS X(N,M)
7      C   E      = VECTOR OF EIGENVALUES E(N)
8      C   V      = MATRIX OF UNIT EIGENVECTORS OF Q-MATRIX V(N,N)
9      C   N      = NUMBER OF SAMPLE VECTORS IN X
10     C   M      = NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR OF X
11     C   NDIM   = NUMBER OF ROWS DIMENSIONED FOR X IN CALLING ROUTINE
12     C   MDIM   = NUMBER OF COLUMNS DIMENSIONED FOR X IN CALLING ROUTINE
13     C   F      = MATRIX OF "FACTOR SCORES" F(M,N)
14     C   L      = MAXIMUM NUMBER OF NON-ZERO EIGENVALUES (MINIMUM OF M OR N)
15     C
16     C   REQUIRES SUBROUTINE NORMALIZE
17     C
18     C   DIMENSION X(NDIM,MDIM),Y(1000),V(NDIM,NDIM),E(NDIM),F(MDIM,NDIM)
19     C   ADJUST RAW SAMPLE VECTORS TO UNIT LENGTH
20     C   DO 2 I=1,N
21     C   DO 1 J=1,M
22     C   1 Y(J)=X(I,J)
23     C   CALL NORMALIZE(Y,M,T)
24     C   DO 2 J=1,M
25     C   2 X(I,J)=Y(J)
26     C   Q-SCORES=NORMALIZED DATA(TRANPOSED)*EIGENVECTORS*1/SQRT(EIGENVALUES)
27     C   L=M
28     C   IF(N.LT.M) L=N
29     C   DO 3 I=1,M
30     C   DO 3 J=1,_
31     C   F(I,J)=0.
32     C   DO 3 K=1,N
33     C   IF(E(J).LE.0.) GO TO 3
34     C   F(I,J)= F(I,J)+X(K,I)*V(K,J)/SQRT(E(J))
35     C   3 CONTINUE
36     C   RETURN
37     C   END

```

```

1      SUBROUTINE RECTANG(Y,YO,YI,X,XO,XI,XC,N)
2      C
3      C PLOTS X-VALUES AS RECTANGLES AGAINST INCREASING Y-VALUES
4      C
5      C XC SPECIFIES THE "ZERO" OR CENTER FROM WHICH BOXES ORIGINATE
6      C
7      C REQUIRES SUBROUTINE ASCEND
8      C
9      DIMENSION X(1),Y(1),INDEX(1000)
10     CALL ASCEND(Y,INDEX,N)
11     YY1=(Y(INDEX(1))-YO)/YI
12     X1=(XC-XO)/XI
13     CALL PLOT(X1,-YY1,3)
14     CALL PLOT((X(INDEX(1))-XO)/XI,-YY1,2)
15     DO 1 I=2,N
16     SX=(X(INDEX(I-1))-XO)/XI
17     SY=(Y(INDEX(I-1))-YO)/YI
18     CALL SYMBOL(SX,-SY,.07,1,0.,-2)
19     CALL SYMBOL(SX,-SY,.07,1,0.,-2)
20     YY=(Y(INDEX(I-1))+Y(INDEX(I))-2*YO)/(2*YI)
21     CALL PLOT((X(INDEX(I-1))-XO)/XI,-YY,2)
22     CALL PLOT(X1,-YY,2)
23     1 CALL PLOT((X(INDEX(I))-XO)/XI,-YY,2)
24     YY2=(Y(INDEX(N))-YO)/YI
25     CALL PLOT((X(INDEX(N))-XO)/XI,-YY2,2)
26     CALL PLOT(X1,-YY2,2)
27     CALL PLOT(X1,-YY1,2)
28     RETURN
29     END

```



```

1      SUBROUTINE RERRORBAR(Y,YORIG,YINC,X,XORIG,XINC,NS,SPIKE,PROB,XC)
2      C
3      C PLOTS INTERVAL ESTIMATES OF CENSUS VARIABLES EXPRESSED AS RATIOS
4      C
5      C X IS THE VECTOR OF RATIOS
6      C X-ORIGIN IS A POWER OF 10; WHEN X .LE. X-ORIGIN, X IS PLOTTED AT
7      C THE ORIGIN.
8      C X-INCREMENT IS IN INCHES PER POWER OF 10
9      C XC IS VALUE ASSIGNED XMIN & XMAX IF "SPIKE" IS A NULL VECTOR
10     C
11     C REQ. SUBROUTINE RATIDERROR
12     C
13     DIMENSION X(1),Y(1),SPIKE(1)
14     CALL NEWPEN(2)
15     DO 2 I=1,NS
16     CALL RATIDERROR(SPIKE(I),X(I),PROB,XMAX,XMIN)
17     YY=(Y(1)-YORIG)/YINC
18     IF(SPIKE(I).EQ.0) THEN
19     IF(PROB.GE.0) THEN
20     XX=0.
21     XMAX=0.
22     XMIN=0.
23     GO TO 1
24     END IF
25     XMAX=(XC-XORIG)/XINC
26     XMIN=XMAX
27     GO TO 1
28     END IF
29     IF(X(I).EQ.0) THEN
30     XX=0.
31     XMAX=0.
32     XMIN=0.
33     GO TO 1
34     END IF
35     IF(ALOG10(X(I)).LE.XORIG) THEN
36     XX=0.
37     ELSE
38     XX=(ALOG10(X(I))-XORIG)/(1/XINC)
39     END IF
40     IF(ALOG10(XMAX).LE.XORIG) THEN
41     XMAX=0.
42     ELSE
43     XMAX=(ALOG10(XMAX)-XORIG)/(1/XINC)
44     END IF
45     IF(ALOG10(XMIN).LE.XORIG) THEN
46     XMIN=0.
47     ELSE
48     XMIN=(ALOG10(XMIN)-XORIG)/(1/XINC)
49     END IF
50     1 CALL SYMBOL(XMIN,-YY,.14,13,0.,-1)
51     CALL SYMBOL(XMIN,-YY,.14,13,0.,-2)
52     CALL SYMBOL(XX,-YY,.07,1,0.,-2)
53     CALL SYMBOL(XX,-YY,.07,1,0.,-2)
54     CALL SYMBOL(XMAX,-YY,.14,13,0.,-2)
55     CALL SYMBOL(XMAX,-YY,.14,13,0.,-2)
56     2 CALL PLOT(XMIN,-YY,2)
57     CALL NEWPEN(1)
58     RETURN
59     END

```

```

1      SUBROUTINE RATIOERROR(SPIKE,RATIO,PROB,RATMAX,RATMIN)
2      C
3      C COMPUTES INTERVAL ESTIMATES OF SPECIFIED CONFIDENCE FOR RATIO OF
4      C TWO COUNTING CATEGORIES. FORMULAE AFTER MOSIMANN(1965), P.659-660.
5      C
6      IF(RATIO.EQ.0) THEN
7          RATMAX=0.
8          RATMIN=0.
9          RETURN
10         END IF
11         IF(SPIKE.EQ.0) SPIKE=1.
12         SIGN=RATIO/ABS(RATIO)
13         P=ABS(RATIO)
14         IF(PROB.LE.99) A=2.58
15         IF(PROB.LE.95) A=1.96
16         IF(PROB.LE.90) A=1.64
17         IF(PROB.LE.85) A=1.44
18         IF(PROB.LE.80) A=1.28
19         IF(PROB.LE.75) A=1.15
20         IF(PROB.LE.68) A=1.00
21         IF(PROB.LE.50) A=.674
22         T=SPIKE
23         IF((A*A)/T.EQ.1) THEN
24             RATMAX=RATIO
25             RATMIN=RATIO
26         ELSE
27             RATMAX=P+(A*A)/(2*T)+A*SQRT(P*(1.+P)/T+(A*A)/(4*T*T))
28             RATMAX=RATMAX/(1-(A*A)/T)
29             RATMAX=RATMAX*SIGN
30             RATMIN=P+(A*A)/(2*T)-A*SQRT(P*(1.+P)/T+(A*A)/(4*T*T))
31             RATMIN=RATMIN/(1-(A*A)/T)
32             RATMIN=RATMIN*SIGN
33         END IF
34         RETURN
35     END

```

```

1      SUBROUTINE ROTATE(N,M,METHOD,A,NDIM)
2      C VARIMAX (OR QUARTIMAX) ROTATION OF ORTHOGONAL FACTORS.
3      C (AFTER COOLEY & LOHNES, 1973)
4      C
5      C THIS SUBROUTINE PERFORMS EITHER A VARIMAX OR A QUARTIMAX ROTATION OF
6      C AN ORTHOGONAL FACTOR PATTERN MATRIX, BY METHODS DESCRIBED BY H.F.
7      C KAISER(1958).
8      C TEST COMMUNALITIES ARE NOT ALTERED BY THE ROTATION. PROPORTIONS OF
9      C VARIANCE EXTRACTED BY THE ROTATED FACTORS ARE REPORTED.
10     C
11     C   N =      NUMBER OF FACTORS
12     C   M =      NUMBER OF TESTS
13     C   METHOD = 0 FOR VARIMAX, 1 FOR QUARTIMAX
14     C   A =      FACTOR PATTERN MATRIX A(M,N)
15     C   NDIM =   MAXIMUM DIMENSION OF A(NDIM,NDIM) IN CALLING PROGRAM
16     C
17     DIMENSION A(NDIM,NDIM),C(100)
18     C DO NOT TRY TO ROTATE ONE FACTOR. MINIMUM N IS 2.
19     EP=.00116
20     IF(METHOD) 6,5,8
21     6 TYPE 7,N,M
22     7 FORMAT(' VARIMAX ROTATION OF '13' FACTORS BASED ON '13' TESTS')
23     GO TO 10
24     8 TYPE 9,N,M
25     9 FORMAT(' QUARTIMAX ROTATION OF '13' FACTORS BASED ON '13' TESTS')
26     10 DO 13 J=1,M
27     C(J)=0.
28     DO 11 K=1,N
29     11 C(J)=C(J)+A(J,K)**2
30     C(J)=SQRT(C(J))
31     DO 13 K=1,N
32     13 A(J,K)=A(J,K)/C(J)
33     C NORMALIZES ROWS OF A
34     N1=N-1
35     14 NR=0
36     DO 23 I=1,N1
37     I1=I+1
38     DO 23 J=I1,N
39     A1=0.
40     B1=0.
41     C1=0.
42     D1=0.
43     DO 15 K=1,M
44     U=A(K,I)**2-A(K,J)**2
45     V=A(K,I)*A(K,J)*2.
46     A1=A1+U
47     B1=B1+V
48     C1=C1+U**2-V**2
49     15 D1=D1+U*V*2.
50     IF(METHOD) 16,16,17
51     16 EM=M
52     QN=D1-2.*A1*B1/EM
53     QD=C1-(A1**2-B1**2)/EM
54     GO TO 18
55     17 QN=D1
56     QD=C1
57     18 IF(ABS(QN)+ABS(QD)) 20,20,19
58     19 IF(ABS(QN)-ABS(QD)) 21,22,35
59     21 EM=ABS(QN/QD)
60     IF(EM-EP) 25,24,24

```



```

52      SN=SIN(ATAN(EM))
53      GO TO 26
54      25 IF(QD) 27,20,20
55      27 SP=.70710678
56      CP=SP
57      GO TO 29
58      30 EM=ABS(QD/QN)
59      IF(EM-EP) 31,30,30
70      30 SN=1/SQRT(1.+EM**2)
71      CS=SN*EM
72      GO TO 26
73      31 CS=0.
74      SN=1.
75      GO TO 26
76      22 CS=.70710678
77      SN=CS
78      20 EM=SQRT((1.+CS)*.5)
79      CS1=SQRT((1.+EM)*.5)
80      SN1=SN/(4.*CS1*EM)
81      IF(QD) 32,33,33
82      32 CP=.70710678*(CS1+SN1)
83      SP=.70710678*(CS1-SN1)
84      GO TO 34
85      33 CP=CS1
86      SP=SN1
87      34 IF(QN) 28,29,29
88      28 SP=-SP
89      GO TO 29
90      20 NR=NR+1
91      GO TO 23
92      29 DO 36 K=1,M
93      EM=A(K,I)*CP+A(K,J)*SP
94      A(K,J)=A(K,J)*CP-A(K,I)*SP
95      30 A(K,I)=EM
96      23 CONTINUE
97      IF(NR-(N*N1)/2) 14,37,14
98      37 DO 38 K=1,M
99      DO 38 L=1,N
100     38 A(K,L)=A(K,L)*C(K)
101  C DENORMALIZES ROWS OF A
102     DO 44 J=1,N
103     C(J)=0.
104     DO 44 K=1,M
105     44 C(J)=C(J)+A(K,J)**2
106     EM=M
107     DO 45 J=1,N
108     45 C(J)=C(J)/EM
109     TYPE *, ' PROPORTION OF VARIANCE ACCOUNTED FOR BY EACH FACTOR.'
110     DO 47 J=1,N
111     47 TYPE 40,J,C(J)
112     48 FORMAT(' FACTOR',I3, ' VAR. PROP. =',F5.3)
113     RETURN
114     END

```

```

1      SUBROUTINE RSCORES(X,XM,XS,V,E,N,M,NDIM,MDIM,F)
2      C
3      C COMPUTES R-MODE PRINCIPAL COMPONENT SCORES FROM RESULTS OF
4      C   DIAGONALIZING A CORRELATION (R) MATRIX
5      C
6      C   X      = MATRIX OF RAW SAMPLE VECTORS X(N,M)
7      C   XM     = VECTOR OF MEANS XM(M)
8      C   XS     = VECTOR OF STANDARD DEVIATIONS XS(M)
9      C   E      = VECTOR OF EIGENVALUES E(M)
10     C   V      = MATRIX OF UNIT EIGENVECTORS OF R-MATRIX V(M,M)
11     C   N      = NUMBER OF SAMPLE VECTORS IN X
12     C   M      = NUMBER OF ELEMENTS IN EACH SAMPLE VECTOR OF X
13     C   NDIM   = NUMBER OF ROWS DIMENSIONED FOR X IN CALLING ROUTINE
14     C   MDIM   = NUMBER OF COLUMNS DIMENSIONED FOR X IN CALLING ROUTINE
15     C   F      = MATRIX OF "FACTOR SCORES" F(N,M)
16     C
17     C   DIMENSION X(NDIM,MDIM),V(MDIM,MDIM),E(MDIM),F(NDIM,MDIM)
18     C   DIMENSION XM(MDIM),XS(MDIM)
19     C   STANDARDIZE VARIABLES TO ZERO MEAN AND UNIT VARIANCE
20     C   DO 1 I=1,N
21     C   DO 1 J=1,M
22     C   1 X(I,J)=(X(I,J)-XM(J))/XS(J)
23     C   R-SCORES=STANDARDIZED VARIABLES*EIGENVECTORS*1/SQRT(EIGENVALUES)
24     C   DO 3 I=1,N
25     C   DO 3 J=1,M
26     C   F(I,J)=0.
27     C   DO 3 K=1,M
28     C   IF(E(J).LE.0.) GO TO 3
29     C   F(I,J)= F(I,J)+X(I,K)*V(K,J)/SQRT(E(J))
30     C   3 CONTINUE
31     C   RETURN
32     C   END

```

```

1      SUBROUTINE SORTVECTOR(S,X,N)
2      C
3      C SORTS A VECTOR X ACCORDING TO INCREASING VALUES IN VECTOR S
4      C
5      C REQUIRES SUBROUTINE ASCEND
6      C
7      DIMENSION S(1),X(1),J(1000),A(1000)
8      CALL ASCEND(S,J,N)
9      DO 1 I=1,N
10     1 A(I)=X(J(I))
11     DO 2 I=1,N
12     2 X(I)=A(I)
13     DO 3 I=1,N
14     3 A(I)=S(J(I))
15     DO 4 I=1,N
16     4 S(I)=A(I)
17     RETURN
18     END

```



```

1      SUBROUTINE SPACER(N,M,X,Y,Z,SPACE,START,WIDTH)
2      C
3      C INTERPOLATES USING A SPLINE
4      C
5      C   X, Y   VECTORS OF LENGTH N
6      C   Z     VECTOR OF Y VALUES INTERPOLATED AT EQUAL INCREMENTS OF
7      C           SPACE; ITS LENGTH IS M
8      C   SPACE Y IS INTERPOLATED AT EQUAL INCREMENTS OF SPACE
9      C   START  INITIAL X-VALUE (MUST BE .GE. X(1))
10     C   WIDTH  RANGE ABOUT X FOR WHICH ADJACENT X'S ARE DUPLICATES
11     C
12     C REQUIRES SUBROUTINE CHECKDUP, SPLINE
13     C
14     REAL X(1),Y(1),Z(1),B(1000),C(1000),D(1000)
15     CALL CHECKDUP(WIDTH,X,Y,N)
16     CALL SPLINE(N,X,Y,B,C,D)
17     T=START-X(1)
18     Z(1)=((D(1)*T+C(1))*T+B(1))*T+Y(1)
19     M=2
20     XV=START+SPACE
21     DO 2 I=1,N-1
22     1 CONTINUE
23     IF((XV.LE.X(I+1)).AND.(XV.GE.X(I))) GO TO 3
24     2 CONTINUE
25     GO TO 4
26     3 T=XV-X(I)
27     Z(M)=((D(I)*T+C(I))*T+B(I))*T+Y(I)
28     M=M+1
29     XV=XV+SPACE
30     GO TO 1
31     4 M=M-1
32     RETURN
33     END

```

```

1      SUBROUTINE SP_LINE(NPTS,X,Y,B,C,D)
2      C
3      C COMPUTATION OF THE COEFFICIENTS OF A CUBIC NATURAL SPLINE.
4      C X,Y ARE ARRAYS WITH THE INPUT DATA.
5      C NPTS IS THE NUMBER OF POINTS IN X AND Y
6      C B,C,D ARE ARRAYS WITH THE COEFFICIENTS OF THE CUBIC SPLINE.
7      C TO INTERPOLATE DATA BETWEEN X(I) AND X(I+1) AT VALUE XX
8      C USE  $S(XX) = (C(I)*T + C(I+1))*T + B(I)*T + Y(I)$ , WHERE  $T = XX - X(I)$ .
9      C
10     C AFTER CARNAHAN, LUTHER & WILKES (1969)
11     C
12     DIMENSION X(1),Y(1),C(1),B(1),D(1)
13     M2=NPTS-1
14     S=0
15     DO 1 K=1,M2
16     D(K)=X(K+1)-X(K)
17     R=(Y(K+1)-Y(K))/D(K)
18     C(K)=K-S
19     1 S=R
20     R=0
21     S=0
22     C(1)=0.0
23     C(NPTS)=0.0
24     IF(M2.LT.2) GO TO 4
25     DO 2 K=2,M2
26     C(K)=C(K)+R*C(K-1)
27     B(K)=(X(K-1)-X(K+1))*2.0-R*S
28     S=D(K)
29     2 R=S/B(K)
30     DO 3 K=2,M2
31     KK=M2-K+2
32     3 C(KK)=(D(KK)*C(KK+1)-C(KK))/B(KK)
33     4 DO 5 K=1,M2
34     B(K)=(Y(K+1)-Y(K))/D(K)-(2.0*C(K)+C(K+1))*D(K)
35     D(K)=(C(K+1)-C(K))/D(K)
36     5 C(K)=3.0*C(K)
37     RETURN
38     END

```

```

1      SUBROUTINE TIMER(DEPTH,TIME,N)
2      C
3      C  CONVERTS STRATIGRAPHIC DEPTH TO TIME, ESTIMATED BY LINEAR
4      C  INTERPOLATION BETWEEN, OR EXTRAPOLATION BEYOND, KNOWN AGE LEVELS
5      C
6      DIMENSION DEPTH(1),TIME(1)
7      TYPE *, '>DEPTH AND AGE OF FIRST DATED LEVEL ?'
8      ACCEPT *,SAVDEP,SAVAGE
9      RATE=0.
10     I=1
11     1 TYPE *, '>DEPTH AND AGE OF NEXT DATED LEVEL ?'
12     ACCEPT *,DEEP,AGE
13     IF(AGE.LE.SAVAGE) GO TO 8
14     DELAGE=AGE-SAVAGE
15     DELDEP=DEEP-SAVDEP
16     RATE=DELDEP/DELAGE
17     TYPE 2,RATE
18     2 FORMAT(/' ACCUMULATION RATE = 'F10.3)
19     5 TIME(1)=(DEPTH(1)-SAVDEP)/RATE+SAVAGE
20     TYPE 6,DEPTH(1),TIME(1)
21     6 FORMAT(' DEPTH = 'F10.3'   TIME = 'F10.3)
22     I=I+1
23     IF(I.GT.N) RETURN
24     IF(DEPTH(1)-DEEP) 5,7,7
25     7 SAVAGE=AGE
26     SAVDEP=DEEP
27     GO TO 1
28     8 TIME(1)=(DEPTH(1)-SAVDEP)/RATE+SAVAGE
29     IF(IANS.EQ.1HY) TYPE 6,DEPTH(1),TIME(1)
30     IF(DEPTH(N)-DEPTH(1)) 10,10,9
31     9 I=I+1
32     IF(I.GT.N) RETURN
33     GO TO 8
34     10 RETURN
35     END

```



```

1      SUBROUTINE TSPLIT(Y,X,N,ITYPE,LP)
2      C
3      C      X      = THE VECTOR ARRAY TO BE PLOTTED
4      C      Y      = THE VECTOR ARRAY TO BE PRINTED WITH X, BUT NOT PLOTTED
5      C      N      = NUMBER OF ELEMENTS IN X THAT ARE TO BE PLOTTED
6      C
7      C      IF ITYPE=1, THE DATA TO BE PLOTTED WILL HAVE RANGE (-1,1)
8      C      IF ITYPE=2, ANY DATA MAY BE PLOTTED
9      C      IF ITYPE=3, LOG10 OF THE X ARRAY WILL BE PLOTTED
10     C      (THE ORIGINAL X-ARRAY WILL NOT BE DESTROYED)
11     C      LP      = UNIT NUMBER FOR PRINTED OUTPUT
12     C
13     DIMENSION X(1),Y(1),IOUT(61),XX(13)
14     DATA II,ISTAR,IBLNK/'I','*',' ' /
15     IF (ITYPE .NE. 1) GO TO 11
16     XMIN=-1.0
17     XMAX=+1.0
18     GO TO 12
19     11 XMIN=X(1)
20     XMAX=XMIN
21     DO 100 I=1,N
22     IF (X(I) .LT. XMIN) XMIN=X(I)
23     IF (X(I) .GT. XMAX) XMAX=X(I)
24     100 CONTINUE
25     IF (ITYPE .NE. 3) GO TO 12
26     XMIN=ALOG10(XMIN)
27     XMAX=ALOG10(XMAX)
28     12 DX=XMAX-XMIN
29     XXX=XMIN
30     DO 101 I=1,13
31     XX(I)=XXX
32     IF (ITYPE .EQ. 3) XX(I)=10.0**XXX
33     XXX=XXX+DX/12.0
34     101 CONTINUE
35     WRITE (LP,2002)
36     DO 102 I=1,N
37     DO 103 J=1,61
38     IOUT(J)=IBLNK
39     103 CONTINUE
40     DO 104 J=1,61,10
41     IOUT(J)=II
42     104 CONTINUE
43     XXX=X(1)
44     IF (ITYPE .EQ. 3) XXX=ALOG10(XXX)
45     IX=IFIX((XXX-XMIN)*60.0/DX)+1
46     IOUT(IX)=ISTAR
47     WRITE (LP,2003) Y(I),IOUT,X(I)
48     102 CONTINUE
49     WRITE (LP,2002)
50     WRITE (LP,2001) (XX(I),I=1,13,2)
51     WRITE (LP,2000) (XX(I),I=2,12,2)
52     RETURN
53     2000 FORMAT (23X,5F10.3)
54     2001 FORMAT (18X,7F10.3)
55     2002 FORMAT (23X,'+',12('-----'))
56     2003 FORMAT (11X,F10.2,2X,61A1,F10.3)
57     END

```

```

1      SUBROUTINE UNITVAR(X,N,XM,XV)
2      C
3      C STANDARDIZES ELEMENTS OF A VARIABLE VECTOR TO
4      C ZERO MEAN AND UNIT VARIANCE
5      C
6      C X = RAW VARIABLE VECTOR (IN) AND STANDARDIZED (OUT)
7      C N = NUMBER OF ELEMENTS IN VARIABLE VECTOR (IN)
8      C XM = MEAN OF X
9      C XV = VARIANCE OF X
10     C
11     DIMENSION X(1)
12     SUM=0.
13     SSQ=0.
14     DO 1 I=1,N
15     SUM=SUM+X(I)
16     1 SSQ=SSQ+X(I)*X(I)
17     XM=SUM/N
18     XV=SSQ/N-SUM*SUM
19     DO 2 I=1,N
20     2 X(I)=(X(I)-XM)/SQRT(XV)
21     RETURN
22     END

```

```

1      SUBROUTINE XYPLOT(X,Y,N,LP)
2      C
3      C   LP = UNIT NUMBER FOR PRINTED OUTPUT
4      C
5      C   AFTER J.C. DAVIS (1973)
6      C
7      DIMENSION X(1),Y(1),IDUT(61),XX(13)
8      DATA IBLNK,II,IPLUS,IMINUS,ISTAR /' ', 'I', '+', '-', '*', /
9      C FIND MINIMA AND MAXIMA OF DATA
10     XMIN=X(1)
11     XMAX=XMIN
12     YMIN=Y(1)
13     YMAX=YMIN
14     DO 1 I=1,N
15     IF(X(I).LT.XMIN) XMIN=X(I)
16     IF(X(I).GT.XMAX) XMAX=X(I)
17     IF(Y(I).LT.YMIN) YMIN=Y(I)
18     1 IF(Y(I).GT.YMAX) YMAX=Y(I)
19     DX=(XMAX-XMIN)/60.
20     DY=(YMAX-YMIN)/40.
21     C
22     DO 10 I=1,41
23     DO 2 J=1,61
24     IDUT(J)=IBLNK
25     2 IF(MOD(J-1,10).EQ.0) IDUT(J)=II
26     GO TO 5
27     3 DO 4 J=1,61
28     IDUT(J)=IMINUS
29     4 IF(MOD(J-1,5).EQ.0) IDUT(J)=IPLUS
30     C PLOT DATA
31     5 DO 6 J=1,N
32     IY=IFIX((Y(J)-YMIN)/DY+.005)+1
33     IF((42-I).NE.IY) GO TO 6
34     IX=IFIX((X(J)-XMIN)/DX+.005)+1
35     IDUT(IX)=ISTAR
36     6 CONTINUE
37     C PRINT LINE OF PLOT
38     IF(MOD(I-1,5).NE.0) GO TO 8
39     WRITE(LP,7) YMAX,IDUT
40     7 FORMAT(13X,F10.3,1X,61A1)
41     YMAX=YMAX-5.*DY
42     GO TO 10
43     8 WRITE(LP,9) IDUT
44     9 FORMAT(24X,61A1)
45     10 CONTINUE
46     C PRINT LEGEND ACROSS BOTTOM OF GRAPH
47     XXX=XMIN
48     DO 11 I=1,13
49     XX(I)=XXX
50     11 XXX=XXX+5.*DX
51     WRITE(LP,12) (XX(I),I=1,13,2)
52     12 FORMAT(20X,7(=10.3))
53     WRITE(LP,13) (XX(I),I=2,12,2)
54     13 FORMAT(25X,6(=10.3))
55     RETURN
56     END

```



```
1      SUBROUTINE ZEROMEAN(X,N,XM)
2      C
3      C STANDARDIZES ELEMENTS OF A VARIABLE VECTOR TO ZERO MEAN
4      C
5      C      X = RAW VARIABLE VECTOR (IN) AND ZERO-MEANED (OUT)
6      C      N = NUMBER OF ELEMENTS IN VARIABLE VECTOR (IN)
7      C      XM = MEAN OF X (OUT)
8      C
9      DIMENSION X(1)
10     SUM=0.
11     DO 1 I=1,N
12     1 SUM=SUM+X(I)
13     XM=SUM/N
14     DO 2 I=1,N
15     2 X(I)=X(I)-XM
16     RETURN
17     END
```

ACKNOWLEDGEMENTS

I always try to acknowledge the original sources of all programs within the documentation of the programs themselves. If I have not, it is certainly not intentional.

I would like to thank the program consultants at WHOI, especially Warren Sass, Mary Hunt, and Roger Goldsmith, who somehow always manage to find my unfindable programming errors. I especially appreciate the contributions of two graduate students, Mary Bremer and Alan Chave. Mary initiated my interest in interactive computer analysis, generated the momentum that eventually got a useable package of programs together, and initiated their adaptation to the VAX11/780. Without Alan's familiarity with the VAX and his willingness to share his extensive knowledge of numerical analytical techniques, I could not have assembled the package of programs in this report in the relatively short time it took. We spent many sunny weekends in front of a computer terminal.

This work was supported by National Science Foundation Grants OCE78-19768, OCE77-82149, and OCE79-18286.

REFERENCES

- Carnahan, B., Luther, H.A., and Wilkes, J.O., 1969. Applied Numerical Methods. John Wiley and Sons, Inc.. 604 pp.
- Cooley, W.S. and Lohnes, P.R., 1971. Multivariate Data Analysis. John Wiley and Sons, New York. 364 pp.
- Dagbert, M. and David, M., 1974. Pattern recognition and geochemical data: an application to Monterey Hills. Can. J. Earth Sci. 11, 1577-1585.
- Davis, J.C., 1973. Statistics and Data Analysis in Geology. John Wiley and Sons, New York. 550 pp.
- Imbrie, J. and Kipp, N.G., 1971. A new micropaleontological method for quantitative paleoclimatology: Application to a late Pleistocene Caribbean core, in K. Turekian (ed.), Late Cenozoic Glacial Ages. Yale University Press, p. 71-181.
- Klovan, J.E. and Imbrie, J., 1971. An algorithm and FORTRAN IV program for large-scale Q-mode factor analyses and calculation of factor scores. Math. Geol. 3, 61-77.
- Mosimann, J.E., 1965. Statistical methods for the pollen analyst: Multinomial and negative multinomial techniques, in B. Kummel and D. Raup (eds.), Handbook of Paleontological Techniques. W.H. Freeman, San Francisco, p. 36-673.
- Pielou, E.C., 1969. An Introduction to Mathematical Ecology. John Wiley and Sons, New York. 286 pp.
- Sokal, P.R., and Rohlf, J.F., 1969. Biometry, the Principals and Practice of Statistics in Biological Research. W.H. Freeman, San Francisco. 776 p.

REPORT DOCUMENTATION PAGE		1. REPORT NO. WHOI-80-27	2.	3. Recipient's Accession No.
4. Title and Subtitle PATS-1, A PACKAGE OF PROGRAMS FOR THE ANALYSIS OF MARINE MICRO-PALEONTOLOGICAL DATA ON THE VAX 11/780 COMPUTER				5. Report Date June 1980
7. Author(s) G.P. Lohmann				6.
9. Performing Organization Name and Address Woods Hole Oceanographic Institution Woods Hole, MA 02543				8. Performing Organization Rept. No.
12. Sponsoring Organization Name and Address National Science Foundation				10. Project/Task/Work Unit No.
				11. Contract(C) or Grant(G) No. (C) OCE 78-19768, OCE 77- (G) 82149, OCE 79-18286
				13. Type of Report & Period Covered Technical
				14.
15. Supplementary Notes				
16. Abstract (Limit: 200 words) <p>PATS-1 is the first version of a package of computer programs and subroutines written or acquired specifically for the archiving, analysis, and presentation of marine micro-paleontological data. The aim of this report is to document the procedures and programs incorporated in PATS-1 and make them available to colleagues working on closely related research problems.</p> <p>The program package described in this report is not intended to be either comprehensive or generally useful. Rather, the PATS-1 package was assembled for a specific purpose. It reflects the solutions to problems encountered in the course of my own work in micropaleontological oceanography. Some programs are so strictly tailored to my own research that they may have to be modified to be of any use to others. However, where convenient I have tried to generalize programs, especially in design of the subroutines incorporated into the PATS-1/LIBRARY.</p> <p>This report provides program listings, documentation, examples and instructions for implementation and use of the PATS-1 package. Programs were written to be used interactively and at WHOI are run exclusively on the time-sharing system of the VAX 11/780 computer.</p>				
17. Document Analysis a. Descriptors				
1. Micropaleontology 2. Computer programs 3. Quantitative Paleontology				
b. Identifiers/Open-Ended Terms				
c. COSATI Field/Group				
18. Availability Statement		19. Security Class (This Report)		21. No. of Pages 148
		20. Security Class (This Page)		22. Price

<p>Woods Hole Oceanographic Institution WHOI-80-27</p> <p>PATS-1, A PACKAGE OF PROGRAMS FOR THE ANALYSIS OF MARINE MICROPALEONTOLOGICAL DATA ON THE VAX 11/780 COMPUTER by G. P. Lohmann. 129 pages. June 1980. Prepared for the National Science Foundation under Grants OCE 78-19768, OCE 77-82149 and OCE 79-18286.</p> <p>PATS-1 is the first version of a package of computer programs and subroutines written or acquired specifically for the archiving, analysis, and presentation of marine micropaleontological data. The aim of this report is to document the procedures and programs incorporated in PATS-1 and make them available to colleagues working on closely related research problems.</p> <p>The program package described in this report is not intended to be either comprehensive or generally useful. Rather, the PATS-1 package was assembled for a specific purpose. It reflects the solutions to problems encountered in the course of my own work in micropaleontological oceanography. Some programs are so strictly tailored to my own research that they may have to be modified to be of any use to others. However, where convenient I have tried to generalize programs, especially in design of the subroutines incorporated into the PATS-1/LIBRARY.</p> <p>This report provides program listings, documentation, examples and instructions for implementation and use of the PATS-1 package. Programs were written to be used interactively and at WHOI are run exclusively on the time-sharing system of the VAX 11/780 computer.</p>	<p>1. Micropaleontology</p> <p>2. Computer programs</p> <p>3. Quantitative paleontology</p> <p>I. Lohmann, G.P.</p> <p>II. OCE 78-19768</p> <p>III. OCE 77-82149</p> <p>IV. OCE 79-18286</p> <p>This card is UNCLASSIFIED</p>	<p>Woods Hole Oceanographic Institution WHOI-80-27</p> <p>PATS-1, A PACKAGE OF PROGRAMS FOR THE ANALYSIS OF MARINE MICROPALEONTOLOGICAL DATA ON THE VAX 11/780 COMPUTER by G. P. Lohmann. 129 pages. June 1980. Prepared for the National Science Foundation under Grants OCE 78-19768, OCE 77-82149 and OCE 79-18286.</p> <p>PATS-1 is the first version of a package of computer programs and subroutines written or acquired specifically for the archiving, analysis, and presentation of marine micropaleontological data. The aim of this report is to document the procedures and programs incorporated in PATS-1 and make them available to colleagues working on closely related research problems.</p> <p>The program package described in this report is not intended to be either comprehensive or generally useful. Rather, the PATS-1 package was assembled for a specific purpose. It reflects the solutions to problems encountered in the course of my own work in micropaleontological oceanography. Some programs are so strictly tailored to my own research that they may have to be modified to be of any use to others. However, where convenient I have tried to generalize programs, especially in design of the subroutines incorporated into the PATS-1/LIBRARY.</p> <p>This report provides program listings, documentation, examples and instructions for implementation and use of the PATS-1 package. Programs were written to be used interactively and at WHOI are run exclusively on the time-sharing system of the VAX 11/780 computer.</p>	<p>1. Micropaleontology</p> <p>2. Computer programs</p> <p>3. Quantitative paleontology</p> <p>I. Lohmann, G.P.</p> <p>II. OCE 78-19768</p> <p>III. OCE 77-82149</p> <p>IV. OCE 79-18286</p> <p>This card is UNCLASSIFIED</p>	<p>Woods Hole Oceanographic Institution WHOI-80-27</p> <p>PATS-1, A PACKAGE OF PROGRAMS FOR THE ANALYSIS OF MARINE MICROPALEONTOLOGICAL DATA ON THE VAX 11/780 COMPUTER by G. P. Lohmann. 129 pages. June 1980. Prepared for the National Science Foundation under Grants OCE 78-19768, OCE 77-82149 and OCE 79-18286.</p> <p>PATS-1 is the first version of a package of computer programs and subroutines written or acquired specifically for the archiving, analysis, and presentation of marine micropaleontological data. The aim of this report is to document the procedures and programs incorporated in PATS-1 and make them available to colleagues working on closely related research problems.</p> <p>The program package described in this report is not intended to be either comprehensive or generally useful. Rather, the PATS-1 package was assembled for a specific purpose. It reflects the solutions to problems encountered in the course of my own work in micropaleontological oceanography. Some programs are so strictly tailored to my own research that they may have to be modified to be of any use to others. However, where convenient I have tried to generalize programs, especially in design of the subroutines incorporated into the PATS-1/LIBRARY.</p> <p>This report provides program listings, documentation, examples and instructions for implementation and use of the PATS-1 package. Programs were written to be used interactively and at WHOI are run exclusively on the time-sharing system of the VAX 11/780 computer.</p>	<p>1. Micropaleontology</p> <p>2. Computer programs</p> <p>3. Quantitative paleontology</p> <p>I. Lohmann, G.P.</p> <p>II. OCE 78-19768</p> <p>III. OCE 77-82149</p> <p>IV. OCE 79-18286</p> <p>This card is UNCLASSIFIED</p>	<p>Woods Hole Oceanographic Institution WHOI-80-27</p> <p>PATS-1, A PACKAGE OF PROGRAMS FOR THE ANALYSIS OF MARINE MICROPALEONTOLOGICAL DATA ON THE VAX 11/780 COMPUTER by G. P. Lohmann. 129 pages. June 1980. Prepared for the National Science Foundation under Grants OCE 78-19768, OCE 77-82149 and OCE 79-18286.</p> <p>PATS-1 is the first version of a package of computer programs and subroutines written or acquired specifically for the archiving, analysis, and presentation of marine micropaleontological data. The aim of this report is to document the procedures and programs incorporated in PATS-1 and make them available to colleagues working on closely related research problems.</p> <p>The program package described in this report is not intended to be either comprehensive or generally useful. Rather, the PATS-1 package was assembled for a specific purpose. It reflects the solutions to problems encountered in the course of my own work in micropaleontological oceanography. Some programs are so strictly tailored to my own research that they may have to be modified to be of any use to others. However, where convenient I have tried to generalize programs, especially in design of the subroutines incorporated into the PATS-1/LIBRARY.</p> <p>This report provides program listings, documentation, examples and instructions for implementation and use of the PATS-1 package. Programs were written to be used interactively and at WHOI are run exclusively on the time-sharing system of the VAX 11/780 computer.</p>	<p>1. Micropaleontology</p> <p>2. Computer programs</p> <p>3. Quantitative paleontology</p> <p>I. Lohmann, G.P.</p> <p>II. OCE 78-19768</p> <p>III. OCE 77-82149</p> <p>IV. OCE 79-18286</p> <p>This card is UNCLASSIFIED</p>
--	---	--	---	--	---	--	---