U.S. DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

An interactive computer program to extrapolate the clay fraction distributions of truncated grain-size data

by

Poppe, L.J.[1] and Eliason, A.H.[2]

Open-File Report 99-27

[1] Coastal and Marine Geology Program, USGS, Quissett Laboratories, Woods Hole, MA 02543
[2] Eliason Data Services, 230 Meetinghouse Road, Mashpee, MA 02649

ABSTRACT

Since the mid-1960's, grain size analyses have been performed on marine sediment samples at the facilities of the Coastal and Marine Geology Program, U.S. Geological Survey in Woods Hole, Massachusetts. However, owing to analytical limitations, the procedures utilized commonly truncate the tails of the fine-fraction portion of the grain-size distributions. This report presents an interactive computer program that can be used to extrapolate incomplete grain-size distributions down to the colloidal clay fraction.

INTRODUCTION

Grain size is the most fundamental property of sediments. Geologists and sedimentologists use it to study trends in surface processes and sediment transport; engineers use it to study stability under load; geochemists use it to study kinetic reactions and the affinities of fine-grained particles and contaminants; and hydrologists use it when studying the movement of subsurface fluids (Blatt and others, 1972; McCave and Syvitski, 1991). Inasmuch as many geological observations consist of measurements made on a large number of specimens, the techniques and equipment used for particle-size analysis must be fast, accurate, and yield highly reproducible results.

The fine-grained fraction of a sediment is defined as silt (particles with diameters less than 62 microns down to 4 microns) plus clay (particles with diameters less than 4 microns, with colloidal clay being less than 0.1 microns). Classical techniques to analyze the fine-grained portion of grain-size distributions are dominated by sedimentation methods, such as the hydrometer and pipette methods (Krumbein and Pettijohn, 1938; Folk, 1974). These methods involve preparing a dispersed, homogeneous suspension of the fine fraction, dilution with a dispersant solution (usually sodium hexametaphosphate) to 1000 ml, and allowing the particles to settle in a graduated cylinder. As the settling of sediment particles continues according to Stoke's Law, samples are either withdrawn (pipette) or measurements made (hydrometer) of the suspension at preset time intervals. However, many sedimentation laboratories have discontinued or limited the use of pipette and hydrometer techniques because of inherent problems with settling (e.g. Brownian motion), thermal convection, irregular particle shape, mass settling in density currents, rounding errors due to large multiplication factors, and, especially, the time necessary to extend an analysis down into the clay range. For example, over 24 hours are required to extend a pipette analysis down to 11ϕ at 20°C, room temperature (Krumbein and Pettijohn, 1938).

Although originally designed to count and size blood cells (Coulter, 1957; Berg, 1958), electro-resistance multichannel particle-size analyzers have found other applications because of the short time and small amount of material required for analysis. These devices, which are currently manufactured by Coulter/Beckman (Coulter Counter) and Particle Data (Elzone), are also used in industry, in the biological sciences (Sheldon and Parsons, 1967), and in geology, where they are used to measure the grain size of sediments by measuring differences in electrical resistivity produced by sediment particles (McCave and Jarvis, 1973; Schiedler, 1976; Kranck and Milligan, 1979; Muerdter and others, 1981; Milligan and Kranck, 1991).

Because the aperture tubes used by electro-resistance multichannel particle-size analyzers can size particles of only 2-40% of the aperture diameter, at least two aperture tubes with overlapping ranges are required to determine the size distribution within the fine fraction of a marine sediment. For example, during a standard multi-aperture analysis using a Coulter Counter, a 200 micron aperture tube would be used to resolve the size distribution between 64 to 8 microns, and a 30 micron aperture tube would be needed to resolve the size distribution between 12 microns down to approximately 0.6 microns. These individual analyses or passes through the respective aperture tubes are then mathematically combined to produce the fine-fraction (silt plus clay) distribution. The fine fraction is, in turn, combined with the coarse fraction data to generate a complete grain size distribution.

An obvious limitation of this method is that electro-resistance multichannel particle-size analyzers, even if calibrated to resolve down to about 0.6 microns (10.75ϕ), can only resolve a portion of the clay size distribution. The fine clay in the 0.6 to 0.1 micron range (some of the 11ϕ and all of the 12ϕ and 13ϕ fractions that extend down to the colloidal clay boundary) is not detected. Although an analysis performed down to 0.6 microns is adequate for most freshwater, estuarine, and shelf environments, the sediments from many deeper water marine environments (e.g. rise and abyssal plain) may contain significant material in the fine clay fraction. By combining a double-point pipette analysis with the data from a electro-resistance multichannel particle-size analysis details of the fine-fraction can be extended through the fine clay range, but this solution is less desirable because of the problems with pipette analyses cited above and because the resultant data are no longer at whole-ϕ intervals.

Earlier efforts to extrapolate grain size data beyond the limits of the analysis used graphical methods (Schlee and Webster, 1967; Schlee, 1973). For example, if Schlee determined that more than 5% of the fine-grained sediment from a sample was less than 1 micron, additional data points were estimated for the finer sizes at whole-ϕ intervals by projecting the grain-size curve as plotted on probability paper and following the slope of the line. While this is a practical solution, it is much more labor intensive than the computer program presented herein.

SOFTWARE

Data Input

The required format of the input file and the data dictionary for the fields of the input file are supplied in Appendices A and B, respectively. The fields within the input file are comma delimited; records are delimited by dollar signs. No imbedded spaces or commas are allowed in any of the fields of the input file.

This particular format was chosen for the input file because it is the same format as that of the output file (a pre-database file) created by GSTAT, the main data processing and statistics program used in the Sedimentation Laboratory of the Woods Hole Field Center, Coastal and Marine Geology Program, U.S. Geological Survey (Poppe and others, 1985). Although many of the fields that describe the sample history are unnecessary and may be left blank, we recommend their inclusion because they provide a more useful record. The necessary fields include the lab number and the phi class and associated cumulative frequency percentages. The lab number is important for two reasons. First, it uniquely identifies the sample; any number or character string can be used. Second, it is the first field; without it the program will crash because the record delimiter will appear out of order. The absence of any identifiers (e.g. navigation, sampling device), attributes (e.g. verbal equivalent), or data (e.g. percent clay, statistics) in the remaining fields will not prevent the program from functioning..

Software Usage

The program CLAYEST, written in Microsoft QuickBasic, can be used to estimate the amount of clay present in a range between a user-defined upper size limit and 0.1 microns. It was designed to calculate that portion of the clay fraction not detected during a particle-size analysis by a Coulter Counter down to 13ϕ (Coulter Electronics, 1988). Documentation is supplied in Appendix C; the software code is provided in Appendix D. The program, when specified, will create a hardcopy (Fig. 1) and/or a digital data file (Fig. 2). The optional hardcopy will contain: the sample identifiers; the original (data supplied with the input file) and revised (data generated by the program) frequency percentages; the original and revised percentages of gravel, sand, silt, and clay; a revised sediment classification (verbal equivalent); and the revised Method of Moments statistics. All sample identifiers of with a length greater than 11 characters will be truncated. The optional data file will be comma delimited and contain: the sample identifiers; the revised percentages of sand, gravel, silt, and clay; the revised verbal equivalent; the revised statistics; and the revised frequency percentages (-5 to 13ϕ). The first line of this data file will contain a list of the field attributes.

The program CLAYEST may be run under DOS 3.0 or later. The program will also run in DOS under Windows 3.1/95/98, and directly under Windows by accessing it from the file manager or the Run command. The program is interactive and will prompt the operator for:

1.) A printer output on LPT1 (the default is no).
2.) Which to use: linear or exponential interpolation or the mean of both (the default is the mean). Linear interpolation may slightly over-estimate the amount of clay present; exponential interpolation may slightly under-estimate the amount of clay present.
3.) The smallest particle size in microns actually measured.
4.) The drive where the data file is located (the default is C:).
5.) The path name to the data file.
6.) What filename to read. If none is entered, the program will display a directory of the file names in the specified drive and path.
7.) The initial lab number. If no lab number is entered the program will search for the first lab number in the specified file.
8.) Whether to write the revised data to disk (the default is yes). The program will add an "E_", "L_", or "M_" depending on which interpolation is chosen to the beginning of the filename assigned to the output file. If the resultant filename is too long (over eight characters), the user will be warned and asked to enter a new file path and name.
9.) Whether the operator wishes to pause the scrolling screen after each record in order to examine the data.

After processing is completed, the operator will be prompted to quit or continue.. If the operator chooses to continue, the program will restart with the first prompt.

Figure 1. An example of a typical hardcopy of revised grain size data generated by the program CLAYEST. Each record is printed on a separate page.

USGS Sediment Size Analysis
Clay Fraction Estimation

Lab #	Field ID	Proj. ID	Cruise ID	Requestor	Latitude	Longitude
AM071	28-02	LIS	SEAX96017	HKNEBEL	41.02000	-73.05609

Device	Location	Depth	Top	Bottom	Sample Weight
VV	CT	36	0	2	28.9046

Phi Size	Original Freq. %	Revised Linearly Freq. %
-5	0.00	0.00
-4	0.00	0.00
-3	0.00	0.00
-2	0.00	0.00
-1	1.38	1.38
0	2.74	2.69
1	6.76	6.63
2	32.32	31.68
3	23.10	22.64
4	7.06	6.92
5	2.88	2.82
6	5.57	5.46
7	5.69	5.58
8	5.46	5.35
9	4.16	4.08
10	2.27	2.23
11	0.61	1.29
12	0.00	0.86
13	0.00	0.43
14	0.00	0.00

	Original		Revised
Gravel	%	1.38	1.35
Sand	%	71.98	70.56
Silt	%	19.61	19.21
Clay	%	7.04	8.88
		Total:	100.00

Revised Classification: SILTY SAND

Revised Statistics

Median	=	2.34
Mean	=	3.32
Standard Deviation	=	2.77
Skewness	=	0.55
Kurtosis	=	0.47

Figure 2. An example of a typical output file generated by the program CLAYEST. Each record contains the original sample identifiers and revised: fraction percentages, verbal equivalent, and statistics; extrapolation method; and revised frequency percents. The first record is a list of the field names; the remaining records are examples of typical data. The fields are delimited by commas and the records are delimited by dollar signs.

Lab Number,Field ID,Project ID,CruiseID,Requestor,Month,Day, Year, Latitude, Longitude, Sample Device, Location, Depth, Top, Bottom, Sample Weight, % Sand, % Gravel, % Silt,% Clay,Verbal Equivalent, Median, Mean, StandardDeviation, Skewness, Kurtosis, Extrapolation Method, 13 Phi Fq. %, 12 Phi Fq. %, 11 Phi Fq. %, 10 Phi Fq. %, 9 Phi Fq. %, 8 Phi Fq. %, 7 Phi Fq. %, 6 Phi Fq. %, 5 Phi Fq. %, 4 Phi Fq. %, 3 Phi Fq. %, 2 Phi Fq. %, 1 Phi Fq. %, 0 Phi Fq. %,-1 Phi Fq. %,-2 Phi Fq. %,-3 Phi Fq. %,-4 Phi Fq. %,-5 Phi Fq. %,
AM307,ELIS-38,LIS,JD96-1,LPOPPE,8,27,96,41.28087,-72.22818,VV,CT,9.0,0,2,39.0394,93.52, 0.00, 4.68, 1.80, SAND, 2.75, 2.97, 1.27, 1.70, 18.20, LINEAR, 0.13, 0.25, 0.38, 0.49, 0.56, 0.54, 0.75, 1.29, 2.11, 28.34, 59.93, 2.90, 1.40, 0.93, 0.00, 0.00, 0.00, 0.00, 0.00,$

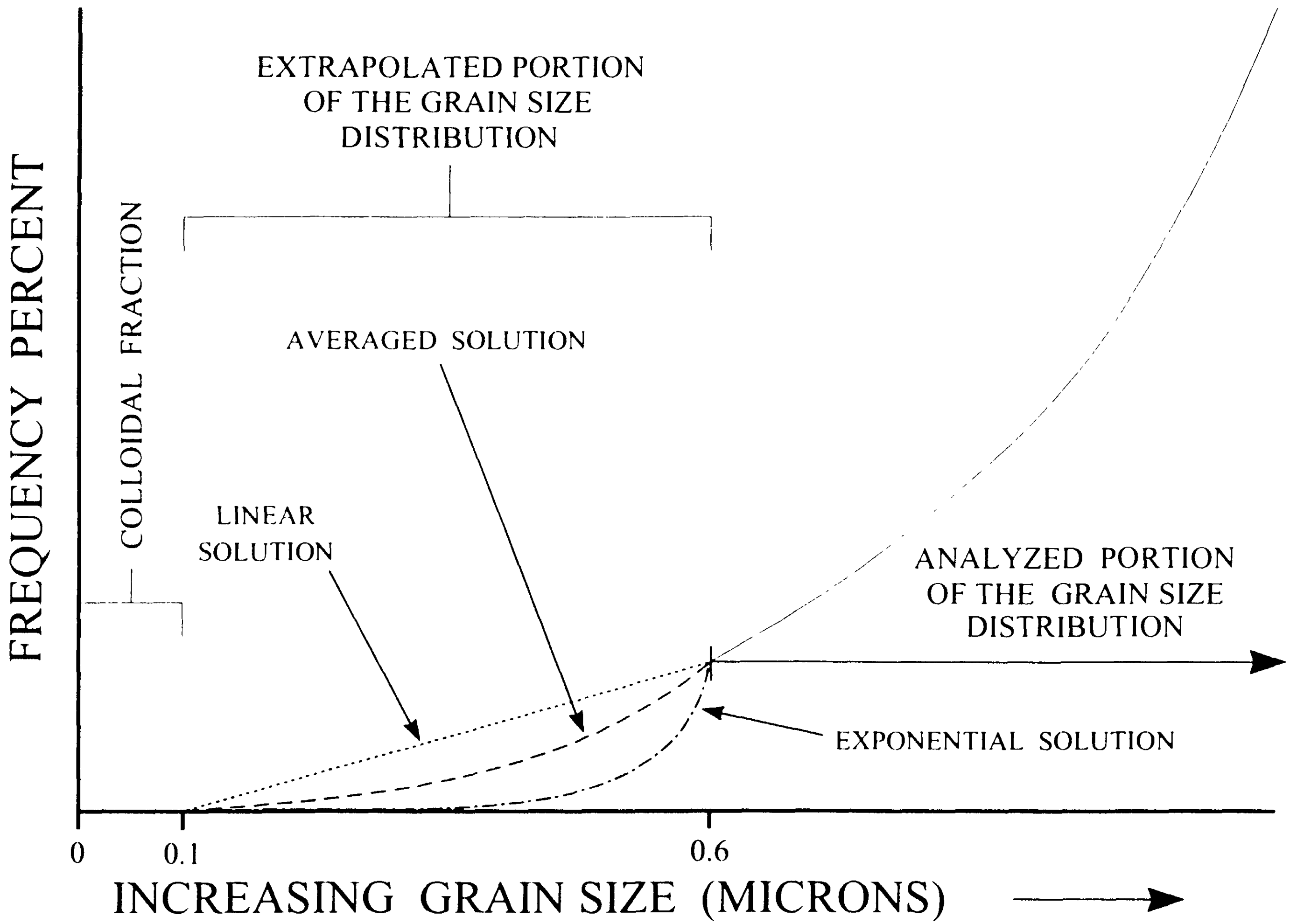

Figure 3. Stylized plot showing that portion of the clay fraction typically truncated by electro-resistance particle size analyzers. Also shown are examples of linear, exponential, and average of the linear and exponential extrapolations to 0.1 microns, the clay-colloidal boundary by the program CLAYEST. Errors inherent in the linear and exponential solutions have been exaggerated to show their effect.

Comments

Figure 3 presents a stylized plot showing that portion of the grain-size distribution typically truncated by electro-resistance particle size analyzers. Also shown in this figure are examples of linear, exponential, and averaged extrapolations of the clay fraction to 0.1 microns, the clay-colloidal clay boundary. Errors inherent in the linear and exponential extrapolations have been exaggerated within this figure to more clearly demonstrate their effect on the grain-size distribution. The linear extrapolation tends to slightly overestimate the amount of clay present in a typical distribution and may be used by operators to account for material within the colloidal fraction. The exponential extrapolation tends to slightly underestimate the amount of clay present in a typical distribution and may be used by operators who want data that represents the minimal amount of clay present. Although an operator may select linear extrapolation, exponential extrapolation, or the mean of both extrapolations, the mean usually provides the most accurate estimate and is, therefore, the recommended solution for most applications.

ACKNOWLEDGMENTS

Interested parties can obtain copies of the documentation and compiled and uncompiled versions of the software on 3.5" diskettes from the Woods Hole Field Center of the Coastal and Marine Geology Program, U.S. Geological Survey, Woods Hole, MA. We thank J. Commeau and R. Rendigs for reviewing this report.

REFERENCES

Berg, R.H., 1958, Electronic size analysis of sub-sieve particles by flowing through a small liquid resistor: ASTM Special Technical Publication 234, p. 245-258.

Blatt, Harvey, Middleton, Gerard, and Murray, Raymond, 1972, Origin of Sedimentary Rocks: Englewood Cliffs, New Jersey, Prentice-Hall, 634 p.

Coulter, W.H., 1957, High speed automatic blood cell counter and cell size analyzer: proceedings of the National Electronics Conference, p. 1034-1042.

Coulter Electronics, Inc., 1988, Fine particle application notes: Instruction manual, 72 p.

Folk, R.L., 1974, The petrology of sedimentary rocks: Hemphill Publishing Co., Austin, 182 p.

Kranck, K. and Milligan, T.G., 1979, The use of the Coulter Counter in studies of particle-size distributions in aquatic environments: Bedford Institute Report Series B1-R, v. 79, p. 7.

Krumbein, W.C., and Pettijohn, F.J., 1938, Manual of Sedimentary petrology, New York, Appleton, Century, and Crofts, 549 p.

McCave, I.N., and Jarvis, J., 1973, Use of the Model T Coulter Counter in size analysis of fine to coarse sand: Sedimentology, v. 20, p. 305-315.

McCave, I.N., and Syvitski, J.P.M., 1991, Principles and methods of particle size analysis, In J.P.M. Syvitski (ed.), Principles, Methods, and Applications of Particle Size Analysis, New York, Cambridge University Press, p. 3-21.

Milligan, T.G., and Kranck, Kate, 1991, Electro-resistance particle size analyzers, In J.P.M. Syvitski (ed.), Principles, Methods, and Applications of Particle Size Analysis, New York, Cambridge University Press, p. 109-118.

Muerdter, D.R., Dauphin, J.P., and steele, G., 1981, An interactive computerized system for grain size analysis of silt using electro-resistance: Journal of Sedimentary Petrology, v. 51, p. 647-650.

Poppe, L.J., Eliason, A.H., and Fredericks, J.J., 1985, APSAS - An automated particle size analysis system: U.S. Geological Survey Circular 963, 77 p.

Schlee, J., and Webster, J., 1967, A computer program for grain-size data: Sedimentology, v. 8, p. 45-54.

Schlee, J., 1973, Atlantic continental shelf and slope of the United States -- sediment texture of the northeastern part: U.S. Geological Survey Professional Paper 529-L, 64 p.

Sheldon, R.W., and Parsons, T.R., 1967, A practical manual on the use of the Coulter Counter in marine research: Toronto, Coulter Electronics, 66 p.

Shideler, G.L., 1976, A comparison of electronic particle counting and pipet techniques in routine mud analysis: Journal of Sedimentary Petrology, v. 42, p. 122-134.

APPENDIX A - INPUT FILE FORMAT FOR AN INDIVIDUAL RECORD

LABNUMBER,FIELDID,PROJECTID,CRUISEID,REQUESTOR,MONTH,DAY,YEAR,LATITUDE,LONGITUDE,DEVICE,LOCATION,DEPTH,TOP,BOTTOM,SAMPLE,WEIGHT,%SAND,%GRAVEL,%SILT,%CLAY,SEDCLASS,MEDIAN,MEAN,STDDEV,SKEWNESS,KURTOSIS,MODE1,M1F%,MODE2,,M2F%,MODE3,M3F%,#MODES,11ϕ,CFP11,10,CFP10,9ϕ,CFP9,8ϕ,CFP8,7ϕ,CFP7,6ϕ,CFP6,5ϕ,CFP5,4ϕ,CFP4,3ϕ,CFP3,2ϕ,CFP2,1ϕ,CFP1,0ϕ,CFP0,-1ϕ,CFP-1,-2ϕ,CFP-2,-3ϕ,CFP-3.-4ϕ,CFP-4,-5ϕ,CFP-5,$

APPENDIX B-DATA DICTIONARY

The data dictionary contains an index of, and definitions for, the abbreviations and the sediment fields names used in the input file (Appendix A).

Short Field Name	Full-Length Field Name	Field Description
LABNUMBER	Unique sample identifier	Unique identification number (consisting of 2 characters and 3 numerics) generated in the sedimentation laboratory of the USGS's Woods Hole Field Center for the Atlantic Margin Texture Database
FIELDID	Field identifier	Sample name or number as assigned in the field
PROJECTID	Project name	Project under which samples were taken, or data generated (i.e. NOS Digital Database); sometimes indicates more specific area
CRUISEID	Cruise identifier	Cruise on which samples were colloected
REQUESTOR	Requestor	Name of Principle Investigator or designee requesting analysis from lab
MONTH	Month	Month sample collected/analyzed
DAY	Day	Day sample collected/analyzed
YEAR	Year	Calender year sample collected/analyzed
LATITUDE	Latitude	Latitude in decimal-degrees (south latitudes are depicted by negative values)
LONGITUDE	Longitude	Longitude in decimal-degrees (west longitudes are depicted by negative values)
DEVICE	Sampling device	Device used to collect the sample
LOCATION	Area	General geographic area (general enough to easily locate on a state map)
DEPTH	Depth or sounding in meters	Measured depth of water overlying sediment at sample time, not corrected for tides, in meters
TOP	Top depth	Top depth of the sample below the sediment-water interface, in cm
BOTTOM	Bottom depth	Bottom depth of the sample below the sediment-water interface, in cm
WEIGHT	Sample weight	Weight of sample in grams
%GRAVEL	Percent gravel	Gravel content in percent dry weight of the sample (particles with nominal diameters greater than 2 mm; -1ϕ and larger)
%SAND	Percent sand	Sand content in percent dry weight of the sample (particles with nominal diameters less than 2 mm, but greater than or equal to 0.0625 mm ; 0ϕ through 4ϕ)
%SILT	Percent silt	Silt content in percent dry weight of the sample (particles with nominal diameters less than 0.0625 mm, but greater than or equal to 0.004 mm; 5ϕ through 8ϕ, inclusive)
%CLAY	Percent clay	Clay content in percent dry weight of the sample (particles with nominal diameters less than 0.004 mm; 9ϕ and smaller)
SEDCLASS	Lithology	General lithologic description
MEDIAN	Median	Middle point in the grain size distribution in phi units
MEAN	Mean	Average value in the grain size distribution in phi units

STDDEV	Standard deviation (sorting)	Standard deviation (root mean square of the deviations) of the grain size distribution in phi units
SKEWNESS	Skewness	Skewness (deviation from symmetrical form) of the grain size distribution in phi units
KURTOSIS	Kurtosis	Kurtosis (degree of curvature near the mode) of the grain size distribution in phi units
MODE1	Mode 1 class	First mode (particle size that occurs the most number of times) in phi units
M1F%	Mode 1 strength	Modal strength of the first mode in percent
MODE2	Mode 2 class	Second mode in phi units
M2F%	Mode 2 strength	Modal strength of the second mode in percent
MODE3	Mode 3 class	Third mode in phi units
M3F%	Mode 3 strength	Modal strength of the third mode in percent
#MODE	Number of modes	Number of modes
11ϕ	11 phi	11 phi fraction
CFP11	Cfp11	Cumulative frequency percent of the 11ϕ fraction and coarser (less than 0.001 mm)
10ϕ	10 phi	10 phi fraction
CFP10	Cfp10	Cumulative frequency percent of the 10ϕ fraction and coarser (less than 0.002 mm)
9ϕ	9 phi	9 phi fraction
CFP09	Cfp09	Cumulative frequency percent of the 9ϕ fraction and coarser (less than 0.004 mm)
8ϕ	8 phi	8 phi fraction
CFP08	Cfp08	Cumulative frequency percent of the 8ϕ fraction and coarser (less than 0.008 mm)
7ϕ	7 phi	7 phi fraction
CFP07	Cfp07	Cumulative frequency percent of the 7ϕ fraction and coarser (less than 0.0016 mm)
6ϕ	6 phi	6 phi fraction
CFP06	Cfp06	Cumulative frequency percent of the 6ϕ fraction and coarser (less than 0.031 mm)
5ϕ	5 phi	5 phi fraction
CFP05	Cfp05	Cumulative frequency percent of the 5ϕ fraction and coarser (less than 0.062 mm)
4ϕ	4 phi	4 phi fraction
CFP04	Cfp04	Cumulative frequency percent of the 4ϕ fraction and coarser (less than 0.125 mm)
3ϕ	3 phi	3 phi fraction
CFP03	Cfp03	Cumulative frequency percent of the 3ϕ fraction and coarser (less than 0.250 mm)
2ϕ	2 phi	2 phi fraction
CFP02	Cfp02	Cumulative frequency percent of the 2ϕ fraction and coarser (less than 0.5 mm)
1ϕ	1 phi	1 phi fraction
CFP01	Cfp01	Cumulative frequency percent of the 1ϕ fraction and coarser (less than 0.1 mm)
0ϕ	0 phi	0 phi fraction
CFP00	Cfp00	Cumulative frequency percent of the 0ϕ fraction and coarser (less than 2 mm)
-1ϕ	-1 phi	-1 phi fraction
CFP-1	Cfp-1	Cumulative frequency percent of the-1ϕ fraction and coarser (less than 4 mm)
-2ϕ	-2 phi	-2 phi fraction
CFP-2	Cfp-2	Cumulative frequency percent of the-2ϕ fraction and coarser (less than 8 mm)

-3ϕ	-3 phi	-3 phi fraction
CFP-3	Cfp-3	Cumulative frequency percent -3ϕ fraction and coarser (less than 16 mm)
-4ϕ	-4 phi	-4 phi fraction
CFP-4	Cfp-4	Cumulative frequency percent -4ϕ fraction and coarser (less than 32 mm)
-5ϕ	-5 phi	-5 phi fraction
CFP-5	Cfp-5	Cumulative frequency percent -5ϕ fraction and coarser

APPENDIX C - PROGRAM DOCUMENTATION

Program Name: CLAYEST.EXE

Type: Main program

Purpose: This program will calculate that portion of the clay fraction not detected during a particle-size analysis by a Coulter Counter (down to 13.0 phi). The operator may select linear or exponential extrapolation, or the mean of both.

Operating System: DOS version 3.0 or later, or DOS under Windows 3.1 or Windows 95

Source Language: Microsoft QuickBASIC version 4.0 or later
Library-standard

Source Code: CLAYEST.BAS

Program Category: Data processing

Input: The pre-database file created by GSTAT, the main data processing and statistics program used in the Sedimentation Laboratory of the Coastal and Marine Geology Program, Woods Hole Section. The format for this pre-database file and an example of a typical file is shown in Appendix A.

Output: When specified, this program will generate a hardcopy (Fig. 1) and/or a data file (Fig. 2).

The optional hardcopy will contain: the sample identifiers; the original and revised frequency percents; the original and revised percentages of gravel, sand, silt, and clay; a revised sediment classification (verbal equivalent); and the revised statistics. All sample identifiers of length greater than 11 characters will be truncated.

The optional data file will be comma delimited and contain: the sample identifiers; the revised percentages of sand, gravel, silt, and clay; the revised verbal equivalent; the revised statistics; and the revised frequency percentages (-5 to 13 phi). The first line of this data file will contain a list of the field attributes.

Usage: To use the program from DOS, change to the working directory and type: CLAYEST
To use the program from Windows 95/98, access it from the file manager or the Run command.

The program CLAYEST is interactive and will prompt the operator for:

1.) A printer output on LPT1 (the default is no).

2.) Which to use: linear or exponential interpolation or the mean of both (the default is the mean). Linear interpolation may slightly over-estimate the amount of clay present; exponential interpolation may slightly under-estimate the amount of clay present.
3.) The smallest particle size in microns actually measured.
4.) The drive where the data file is located (the default is C:).
5.) The path name to the data file.
6.) What filename to read. If none is entered, the program will display a directory of the file names in the specified drive and path.
7.) The initial Lab Number. If no lab number is entered the program will search for the first lab number in the specified file.
8.) Whether to write the revised data to disk (the default is yes). The program will add an "E_", "L_", or "M_" depending on which interpolation is chosen to the beginning of the filename assigned to the output file. If the resultant filename is too long (over eight characters), the user will be warned and asked to enter a new file path and name.
9.) Whether the operator wishes to pause the scrolling screen after each record in order to examine the data.

Author: Eliason Data Services, Mashpee, MA 02649 (508) 477-1400

Maintenance: L. Poppe, USGS, Woods Hole, MA 02543

APPENDIX D - PROGRAM SOFTWARE

```
' *************************************************************
' ** Name:       CLAYEST97.BAS                          **
' ** Type:       MAIN [ X ]  FUNCTION [  ]  SUB [  ]   **                  **
' ** Language:   Microsoft QuickBASIC 4.00 or above    **
' ** Library :   Standard                     **
' ** Arguments:  None  - Main Module                   **
' ** Programmer: A. Eliason- Eliason Data- (508)477-1400 **
' ** Date:       Version # in main module includes date  **
' **             as: MMDDYYYY                         **
' ** Description: Please see comments within this module  **
' **                                         **
' ** Usage:      If no info here please see the line     **
' **             comments.                   **
' *************************************************************

' Clay Extrapolation Program - CLAYES97.BAS
' Last rev 97-06-24 AHE
DEFINT I-N                              'use Fortran conventions
DEFDBL A-H, O-Z

DIM PHIP(32), FP(32), FPcor(32), FPcor100(32), Iden(4), F(2002)
DIM XFit(4), YFit(4)
DIM s(32), index(32)
DIM Itext$(1000)
DIM Lable$(70)

DECLARE SUB CPI (PHIP(), FP(), N, DP, F(), IP, DPHI)
DECLARE SUB BPLOT (TMIN, TMAX, X(), N)            'EDS sub for FOPLOT
DECLARE SUB PressAnyKey ()
DECLARE SUB CSCOEF (N, X(), Y(), s(), index())
DECLARE SUB CSFIT1 (N, X(), Y(), s(), index(), X1, sx)
DECLARE SUB CSFIT2 (N, X1(), Y1(), YP1, YPN, X, Y)
DECLARE SUB FFIT (N, X(), Y(), IType, A, B)
DECLARE FUNCTION U2Phi (US)

Version$ = "062497.01"
Diag% = 0
FileWrite% = 1
Cont% = 0
InterpFlag = 1
ISWFlag = 0      'no Schlee-Webster
```

```
Lprnt = 0
PhiNudge = .045   ' this factor is added to conversions from micron dia to Phi
          ' in order to adjust for the offset between whole micron
          ' diamaters and phi boundries eg: 1 micron actually= 9.965..Phi
          ' so we add .045 to Phi in order to cross the 10 Phi boundry.
NextFile:
CLS

PRINT
PRINT "                    USGS   Sediment Size Analysis"
PRINT "               Clay Fraction Estimation    Version "; Version$
PRINT
PRINT "OPTIONS:"
INPUT "     Do you want printed output (Y/N)?         [N] ", A$
IF UCASE$(LEFT$(A$, 1)) = "Y" THEN Lprnt = 1
PRINT
ExType = 4                              ' default to MEAN
InterpFlag = 1
' lets not kill this yet  rev 6-18-97
'INPUT "      Do you want to use Extrapolation (Y/N)?   [Y] ", A$
'IF UCASE$(LEFT$(A$, 1)) = "N" THEN InterpFlag = 0
   ' If NO is selected then only the original data are printed/output
IF InterpFlag THEN

   PRINT "Please READ the following carefully:"
   PRINT
   PRINT "  You may elect to estimate clay content from the smallest measured size to"
   PRINT "  13 phi. This may be done by LINEAR or EXPONENTIAL extrapolation, or the "
   PRINT "  MEAN of both sets of results. "
   PRINT "  Linear extrapolation may tend to over-estimate the clay present, while"
   PRINT "  exponential may under-estimate."
   PRINT "  It is up to the operator to determine the best method based"
   PRINT "  upon the original data."
   PRINT

   INPUT "  Do you wish to use LINEAR, EXPONENTIAL, or MEAN (L/E/M)?  [M] ", A$
   IF UCASE$(LEFT$(A$, 1)) = "L" THEN ExType = 1
   IF UCASE$(LEFT$(A$, 1)) = "E" THEN ExType = 3
   ' else ExType = the default, MEAN, = 4
   PRINT
   PRINT "  The program needs to know the smallest measured size based upon"
   PRINT "  the Coulter callibration. Normally this will be a value between"
   PRINT "  0.5 and 0.8 microns. You MUST enter a value."
Size:
```

```
    PRINT
    DO
       INPUT "      Smallest measured micron diameter"; UD
    LOOP WHILE UD <= 0
    IF UD < .5 OR UD > .8 THEN
       PRINT USING "   You have entered a value which corresponds to ###.## Phi"; U2Phi(UD)
       INPUT "   Is this correct (Y/N)? ", A$
       IF UCASE$(LEFT$(A$, 1)) <> "Y" THEN GOTO Size:
    END IF
    PhiMin = U2Phi(UD)          ' actulal phi size
    PhiMin = PhiMin + PhiNudge   ' correct to USGS boundry
    IntPhi = INT(PhiMin)         ' Integer portion of smallest measured
    IntPhiFrac = IntPhi + 1      ' Fraction containing the smallest meas.
    NdxPhiCor = IntPhiFrac + 6   ' index to Fraction Array
    PhiCorFac = PhiMin - IntPhi   ' divisor for correction
    PctPhiCor = 100 - INT(PhiCorFac * 10000) / 100   ' just for information

    PRINT USING "   The value you entered corresponds to ##.## Phi"; PhiMin
    PRINT USING "   The ## Phi Fraction will be increased in order to account"; IntPhiFrac
    PRINT USING "   for the ##.## percent perceived to be undetected in this fraction."; PctPhiCor
    IF Daig% THEN PRINT "   Value will be divided by "; PhiCorFac
    PRINT


END IF
PressAnyKey
' Read the input field lables
FOR I = 1 TO 68
   READ Lable$(I)
   IF Diag% = 1 THEN PRINT Lable$(I); " ~ ";
NEXT I
PRINT

' new file read routine  9-16-96 ############  AHE
FilePath:
INPUT "Enter Drive Letter for data file - [C:] ", Drv$
IF Drv$ = "" THEN Drv$ = "C:"
IF LEN(Drv$) = 1 THEN Drv$ = Drv$ + ":"
IF LEN(Drv$) > 2 THEN GOTO FilePath
PRINT
PRINT "You will now be asked for a PATH then a FILENAME. If you don't know, leave"
PRINT "one or both blank and you will be shown a Directory screen."
INPUT "File PATH to read?  - eg '\DATA\' [\] ", FPN$
IF FPN$ = "" THEN FPN$ = "\"
```

```
IF LEFT$(FPN$, 1) <> "\" THEN FPN$ = "\" + FPN$
IF RIGHT$(FPN$, 1) <> "\" THEN FPN$ = FPN$ + "\"
PRINT "Data Path =        "; Drv$ + FPN$
INPUT "FILENAME to read?  "; FLN$

IF FLN$ = "" THEN                        ' Guess s/he doesn't know
   FILES Drv$ + FPN$
   PRINT "Please start again. When asked, enter a FILENAME from the list above."
   PRINT "You MUST include the extension (eg  AAAAAA.PG)"      '6-97 AHE
   PRINT "This program WILL NOT accept FILENAMEs of more than 6 characters nor ext-"
   PRINT "tions greater than 3 characters"
   GOTO FilePath
END IF

LabNo:
INPUT "Initial LAB NUMBER eg 'AHE001'       ? ", LabNo$
LabNo$ = UCASE$(LabNo$)
LnLen = LEN(LabNo$)
FoundFlag% = 0
Drv$ = UCASE$(Drv$)
FPN$ = UCASE$(FPN$)
FLN$ = UCASE$(FLN$)


PRINT "Attempting to open "; Drv$ + FPN$ + FLN$

OPEN Drv$ + FPN$ + FLN$ FOR INPUT AS #1

DO WHILE NOT EOF(1)
   INPUT #1, A$
   IF (UCASE$(LEFT$(A$, LnLen)) = LabNo$) OR (UCASE$(MID$(A$, 2, LnLen)) = LabNo$)
THEN
      IF LEFT$(A$, 1) = "$" THEN A$ = MID$(A$, 2, LnLen)
      FoundFlag% = 1
      EXIT DO
   END IF

   IF Diag% THEN PRINT A$; "  ";
LOOP

IF FoundFlag% THEN
   PRINT
   PRINT " Found "; A$
   LN$ = A$
```

```
ELSE
   BEEP
   PRINT
   PRINT " Did NOT find "; LabNo$
   PRINT " Please try again with another Lab Number."
END IF
IF FoundFlag% = 0 THEN
   CLOSE #1
   PRINT
   GOTO LabNo
END IF
INPUT "Do you wish to write extrapolated data to Disk? [Y]"; A$
IF UCASE$(LEFT$(A$, 1)) = "N" THEN FileWrite% = 0
IF FileWrite% = 1 THEN
   RevType$ = "M_"                    'default- mean of both
   IF ExType = 1 THEN RevType$ = "L_"     'linear
   IF ExType = 3 THEN RevType$ = "E_"     'exponential

   OutFile$ = Drv$ + FPN$ + RevType$ + FLN$
      IF LEN(FLN$) > 10 THEN
         PRINT
         PRINT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
         PRINT "CAN NOT Write output data to "; OutFile$
         PRINT "!!!!!!!!!! File Name too Long !!!!!!!!!!"
         PRINT "You must specify a new output file Path/Name!"
         BEEP
      ELSE
         PRINT "Output data will be written to the following Path/File,"
         PRINT "unless you enter another complete pathname."
      END IF
   PRINT "Output Filename  [ "; OutFile$; " ] ";
   INPUT "", A$
   IF UCASE$(A$) = "Y" THEN A$ = ""
   IF A$ <> "" THEN OutFile$ = A$
   PRINT "Attempting to open "; OutFile$; " for output."
   OPEN OutFile$ FOR OUTPUT AS #2
   PRINT "Open sucessful."
   ' Write the attribute names to disk
   FOR I = 1 TO 26
      PRINT #2, Lable$(I); ",";
   NEXT I
   ' attribute name # 27 is type of extrapolation
   PRINT #2, "Extrapolation Method,";
```

```
    ' kill this LJP 10-24-96 PRINT #2, "Mode 1 Fq.%, Mode 2 Fq.%, Mode 3 Fq.%,Number of
Modes,";

   FOR I = 13 TO -5 STEP -1              ' -4 to -5 10/96 AHE
      PRINT #2, I; " Phi Fq. %,";
   NEXT I
   PRINT #2, ""

END IF
PRINT
Cont% = 0                          ' Correct error AHE 6-24-97
INPUT "Do you wish to pause and examine the data for each record? [Y]", A$
IF UCASE$(LEFT$(A$, 1)) = "N" THEN Cont% = 1      'True


'PressAnyKey
NextSamp:

Itext$(1) = LN$
I = 2
DO
   A$ = "": B$ = ""
   DO
      DO                            ' ignore CR & LF
         B$ = INPUT$(1, #1)
      LOOP WHILE B$ = CHR$(10) OR B$ = CHR$(13)
      IF B$ = "," THEN EXIT DO          ' end of field - exit
      A$ = A$ + B$                      ' concatenate
      IF B$ = "$" THEN EXIT DO          ' because $ is not comma delimited
   LOOP
   Itext$(I) = A$
   I = I + 1
   IF Diag% THEN PRINT A$; "~";
LOOP UNTIL A$ = "$"

IF Diag% THEN
   PRINT : PRINT
   FOR I = 1 TO 68
      PRINT USING "\                 \&"; Lable$(I); Itext$(I)
      IF (I MOD (22) = 0) THEN PressAnyKey
   NEXT I
   PressAnyKey
END IF
```

```
' check for proper field placement
IF Itext$(68) <> "$" THEN
    PRINT : PRINT
    PRINT "End of Record Character ($) NOT IN PROPER LOCATION"
    PRINT "Aborting Program"
    BEEP: BEEP: BEEP
    PRINT : PRINT
    END
END IF


' Now convert Cumulative Fq% to Fq%

FOR j = 1 TO 32
    PHIP(j) = j - 6: FP(j) = 0: FPcor(j) = 0: FPcor100(j) = 0 'init the arrays
NEXT j

Cumul = 0: FqPct = 0: j = 1
FOR I = 67 TO 35 STEP -2          ' step from largest to smallest size
    FqPct = VAL(Itext$(I)) - Cumul
    FqPct = INT(FqPct * 100 + .499999999#) / 100' round to two places
    Cumul = Cumul + FqPct
    PHIP(j) = VAL(Itext$(I - 1))  ' Phi Size from input data
    FP(j) = FqPct               ' Actual phi percentage
    IF Diag% THEN PRINT USING "## \     \ ##.## ###.##"; PHIP(j); Itext$(I); FP(j); Cumul
    j = j + 1
    '*** PHIP(j) now contains -5phi at 1, -4 at 2,... 11phi at 17 ***
NEXT I
PRINT


N = 20                    ' -5 to 14 phi FRACTION inclusive
DPHI = .2
XSand = VAL(Itext$(17))
Grav = VAL(Itext$(18))
Silt = VAL(Itext$(19))
clay = VAL(Itext$(20))


' test for total percent = 100
CFP = 0
FOR j = 1 TO N
    CFP = CFP + FP(j)
NEXT j
```

```
IF CFP > 101 OR CFP < 99 THEN

   PRINT "Cumulative Frequency Percent is not 100 +/- 1"
   PRINT
   IF Lprnt THEN
      LPRINT "Cumulative Frequency Percent is not 100 +/- 1"
      LPRINT
   END IF
   PressAnyKey
   GOTO NextSamp
END IF

' test if Phi intervals are even
DP = PHIP(2) - PHIP(1)              '=.5
FOR j = 3 TO N
   Test2 = PHIP(j) - PHIP(j - 1)     '=.5
   Test3 = ABS(DP - Test2)           '=0
   IF Diag% THEN PRINT DP, Test2, Test3: PRINT
   IF (Test3 > .001) THEN
      PRINT Iden$; " Phi intervals are not even"
      PRINT
      IF Lprnt THEN
         PRINT Iden$; " Phi intervals are not even"
         PRINT
      END IF
      PressAnyKey
      GOTO NextSamp
   END IF
   DP = Test2
NEXT j

'PressAnyKey

RealStuff:
IF InterpFlag = 0 THEN GOTO PrintOut
PRINT
PRINT "Doing the initial Extrapolation"
FOR I = 1 TO N             ' copy the array
   FPcor(I) = FP(I)
NEXT I

FOR I = N TO 1 STEP -1       ' find the smallest non-zero fraction
   IF FP(I) > 0 THEN
      NZ = I            ' NZ= index to smallest FRACTION with data
```

```
        EXIT FOR
    END IF
NEXT I
' Make the percentage correction to the fraction containg the smallest
' measurable data
imin = NdxPhiCor
IF FP(imin) > 0 THEN
    FPcor(imin) = FP(imin) / PhiCorFac
    PRINT USING "The smallest measurable fraction is ## Phi with a value of ##.##%"; IntPhiFrac;
FP(imin)
    PRINT USING "##.## has been changed to ##.##"; FP(imin); FPcor(imin)
    PRINT
ELSE
    PRINT USING "The ## Phi fraction was zero so it needed no change."; IntPhiFrac
    PRINT
END IF

' Now do Extrapolation from the smallest non zero fraction to 14 FRACTION
' which is zero. Don't forget the 13 FRACTION contains the 12.001 to 13 data

PRINT USING "The smallest non-zero fraction is ## Phi with a value of ##.##%"; PHIP(NZ);
FPcor(NZ)
IF ExType = 1 THEN PRINT "Linear ";
IF ExType = 3 THEN PRINT "Exponential ";
IF ExType = 4 THEN PRINT "Mean of L./E. ";
IF ExType <> 4 AND ExType <> 3 AND ExType <> 1 THEN PRINT "error at ExType. Program
halted.": STOP
PRINT "Extrapolation will now be performed to 13 Phi.(14 Phi Fraction=0)"
PRINT
' Set up for calls to FFIT (Linear and Exponential)
index = NZ
XFit(1) = index
YFit(1) = FPcor(index)

XFit(2) = index
YFit(2) = FPcor(index)

XFit(3) = 20
YFit(3) = .0000001

CALL FFIT(3, XFit(), YFit(), 1, Alin, Blin)
CALL FFIT(3, XFit(), YFit(), 3, Aexp, Bexp)
```

```
index = NZ
'Npoints = 20 - NZ
'CLinFac = FPcor(index) / Npoints
FOR I = index + 1 TO 19
   IF ExType = 1 THEN FPcor(I) = Alin + Blin * I         ' linear
   IF ExType = 3 THEN FPcor(I) = Aexp * EXP(Bexp * I)     ' logarithmic
   IF ExType = 4 THEN FPcor(I) = ((Alin + Blin * I) + (Aexp * EXP(Bexp * I))) / 2
NEXT I

' Re-normalize to 100 percent
TotalPct = 0
FOR I = 1 TO 20                        ' N?
   TotalPct = TotalPct + FPcor(I)
NEXT I
FOR I = 1 TO 20
   FPcor100(I) = 100 * FPcor(I) / TotalPct
NEXT I



PrintOut:
PRINT
IF Cont% = 0 THEN PressAnyKey

'

' Print the header data
CLS
PRINT "                  USGS   Sediment Size Analysis"
PRINT "              Clay Fraction Estimation   Version "; Version$
PRINT
PRINT "Lab #     Field ID   Proj. ID   Cruise ID  Requestor  Latitude   Longitude"
PRINT
PRINT USING "\       \"; Itext$(1); Itext$(2); Itext$(3); Itext$(4); Itext$(5); Itext$(9); Itext$(10)
PRINT
PRINT
PRINT "Device    Location   Depth     Top      Bottom    Samp. Weight"
PRINT
PRINT USING "\       \"; Itext$(11); Itext$(12); Itext$(13); Itext$(14); Itext$(15); Itext$(16)
PRINT
IF Cont% = 0 THEN PressAnyKey
PRINT
PRINT "             Original    Revised";
IF ExType = 1 THEN PRINT " Linearly"
```

```
IF ExType = 3 THEN PRINT " Exponentially"
IF ExType = 4 THEN PRINT " by Mean of Linear and Exponential"


PRINT "  Phi Size    Freq. %     Freq. %"
FOR j = 1 TO N
   PRINT USING "     ##       ##.##       ##.##"; PHIP(j); FP(j); FPcor100(j)
NEXT j
PRINT
IF Lprnt THEN
   ' Print the original data to LPT1

   LPRINT "                    USGS   Sediment Size Analysis"
   LPRINT "              Clay Fraction Estimation     Version "; Version$; ""
   LPRINT
   LPRINT "   Lab #     Field ID   Proj. ID   Cruise ID  Requestor  Latitude   Longitude"
   LPRINT "    ";
   LPRINT USING "\        \"; Itext$(1); Itext$(2); Itext$(3); Itext$(4); Itext$(5); Itext$(9); Itext$(10)
   LPRINT
   LPRINT "   Device     Location   Depth     Top       Bottom     Samp. Weight"
   LPRINT "    ";
   LPRINT USING "\        \"; Itext$(11); Itext$(12); Itext$(13); Itext$(14); Itext$(15); Itext$(16)
   LPRINT
   LPRINT "                 Original    Revised";
   IF ExType = 1 THEN LPRINT " Linearly"
   IF ExType = 3 THEN LPRINT " Exponentially"
   IF ExType = 4 THEN LPRINT " by Mean of Linear and Exponential"

   LPRINT "     Phi Size    Freq. %     Freq. %"
   LPRINT
   FOR j = 1 TO N
      LPRINT USING "        ##       ##.##       ##.##"; PHIP(j); FP(j); FPcor100(j)
   NEXT j
   LPRINT
END IF



' test for total percent = 100
CFP = 0
FOR j = 1 TO N
   CFP = CFP + FP(j)
NEXT j
IF CFP > 101 OR CFP < 99 THEN
```

```
    PRINT "Cumulative Frequency Percent is not 100 +/- 1"
    PRINT
    IF Lprnt THEN
        LPRINT "Cumulative Frequency Percent is not 100 +/- 1"
        LPRINT
    END IF
    PressAnyKey
    GOTO NextSamp
END IF

' test if Phi intervals are even
DP = PHIP(2) - PHIP(1)                  '=.5
FOR j = 3 TO N
    Test2 = PHIP(j) - PHIP(j - 1)       '=.5
    Test3 = ABS(DP - Test2)             '=0
    IF Diag% THEN PRINT DP, Test2, Test3: PRINT
    IF (Test3 > .001) THEN
        PRINT Iden$; " Phi intervals are not even"
        PRINT
        IF Lprnt THEN
            PRINT Iden$; " Phi intervals are not even"
            PRINT
        END IF
        PressAnyKey
        GOTO NextSamp
    END IF
    DP = Test2
NEXT j

IF Cont% = 0 THEN PressAnyKey
Emp1 = PHIP(1) - .5 * DP              'Find center of 1st class interval
                                      ' to be used at end for statistics.
IP = (PHIP(N) - PHIP(1)) / DPHI + 1.5  'Calculate # of Extrapolated points
                                      '(Span/Interp_Interval)+1.5 = # points
IF ISWFlag = 0 THEN
    'PRINT "No S-W Extrapolation Requested"
    'PRINT
    'IF Lprnt THEN
    '   LPRINT "No S-W Extrapolation Requested"
    '   LPRINT
    'END IF
    IP = N                        ' Extrapolated Points = Original
    DPHI = DP                        ' Delta Phi (interp) = Original
    FOR I = 1 TO N
```

```
      F(I) = FPcor100(I)              ' Extrapolated Data = Original
                           'changed FP(i) to FPcor(i) 10-24-96 AHE
      FP(I) = FPcor100(I)       'reload FP()            10-25-96
   NEXT I
ELSE
   PRINT "S-W Extrapolated Values will be used"
   PRINT
   IF Lprnt THEN
      LPRINT "S-W Extrapolated Values were used"
      LPRINT
   END IF
   FOR I = 1 TO N
      FP(I) = FPcor100(I)
   NEXT I

GOTO SchleeWebster
' Test CSCOEF *************************************************
   PRINT "Running CSCOEF ...";
   CALL CSCOEF(N, PHIP(), FPcor(), s(), index())
   PRINT " DONE"
      FOR I = 1 TO N
      PRINT s(I)
      NEXT I
   X1 = PHIP(1)
   FOR I = 1 TO IP
      'CALL CSFIT1(N, PHIP(), FPcor(), s(), index(), X1, SX)
      'CALL CSFIT2(N, PHIP(), FPcor(), 0, 0, X1, SX)
      F(I) = sx
      X1 = X1 + DPHI
   NEXT I
   ' Re-normalize to Phi Percentages
   DPI = DPHI / DP
   FOR I = 1 TO IP
      IF F(I) < 0 THEN F(I) = 0
      F(I) = F(I) * DPI
      IF Diag% THEN
         PRINT I, F(I)
         IF (I MOD (15) = 0) THEN PressAnyKey
      END IF
   NEXT I
   'Scale the data for plotting
   Ymax = 0
   FOR I = 1 TO IP
      IF F(I) > Ymax THEN Ymax = F(I)
```

```
   NEXT I
   IF Lprnt THEN LPRINT "Plot Using CSFIT1 - Natural Cubic Spline"
   'IF Lprnt THEN LPRINT "Plot Using CSFIT2 - Clamped Cubic Spline"
   CALL BPLOT(0!, Ymax + 1.5, F(), IP)
   IF Lprnt THEN LPRINT CHR$(12)
   ' Ends test ***********************************************************
SchleeWebster:


   ' move all the original values up by 1 in the arrays but leave first
   ' element as it was.
   FOR I = 1 TO N
      'IF Diag% THEN PRINT FP(J + 1), PHIP(J + 1)
      j = N - I
      FP(j + 2) = FP(j + 1)
      PHIP(j + 2) = PHIP(j + 1)
   NEXT I


   PRINT
   PRINT USING "### Points at ##.## Phi will be fit to ### points at #.### Phi"; N; DP; IP; DPHI
   IF Lprnt THEN
      LPRINT USING "### Points at ##.## Phi will be fit to ### points at #.### Phi"; N; DP; IP;
DPHI
      LPRINT
   END IF

   CALL CPI(PHIP(), FP(), N, DP, F(), IP, DPHI)
   ' Re-normalize to Phi Percentages
   DPI = DPHI / DP
   FOR I = 1 TO IP
      IF F(I) < 0 THEN F(I) = 0
      F(I) = F(I) * DPI
      IF Diag% THEN
         PRINT I, F(I)
         IF (I MOD (15) = 0) THEN PressAnyKey
      END IF

   NEXT I
   PRINT
END IF


'Here is a cut at the Lloyd Breslaus Sediment Class Program
```

```
'IF Cont% = 0 THEN PressAnyKey          'REM'd out to elininate double PAC 6-97
PRINT " Calculating Descriptive Classification Percentages"
PRINT
' Get the new percentages
Gravel = 0
Sand = 0
Silt = 0
clay = 0

FOR I = 1 TO 5              'Gravel
   Gravel = Gravel + FP(I)
NEXT I
FOR I = 6 TO 10             'Sand
   Sand = Sand + FP(I)
NEXT I
FOR I = 11 TO 14            'Silt
   Silt = Silt + FP(I)
NEXT I
FOR I = 15 TO 20            'Clay
   clay = clay + FP(I)
NEXT I

IF Sand = 0 THEN Sand = .001
IF Gravel = 0 THEN Gravel = .001
IF Silt = 0 THEN Silt = .001
IF clay = 0 THEN clay = .001
   DO
      IF Gravel >= 50 THEN C% = 1: EXIT DO
      IF Gravel >= 10 THEN C% = 2: EXIT DO
      XSand = Sand
      Sand = Sand + Gravel
      IF Sand >= 75 THEN C% = 3: EXIT DO
      IF Silt >= 75 THEN C% = 4: EXIT DO
      IF clay >= 75 THEN C% = 5: EXIT DO

      ' Sand, Silt, & Clay are all less than 75%
      SanSil = Sand / Silt
      ClySnd = clay / Sand
      SilCly = Silt / clay

      IF Sand <= 20 THEN
         IF SanSil > 1 THEN C% = 6: EXIT DO
         IF SilCly < 1 THEN C% = 7: EXIT DO
         IF ClySnd > 1 THEN
```

```
            C% = 8: EXIT DO
        ELSE C% = 9: EXIT DO
        END IF
    ELSE                    ' Sand> 20%

        IF clay <= 20 THEN
            IF SanSil < 1 THEN C% = 9: EXIT DO
            IF SilCly > 1 THEN C% = 10: EXIT DO
            IF ClySnd > 1 THEN
                C% = 6: EXIT DO
            ELSE
                C% = 11: EXIT DO
            END IF
        ELSE                    ' Clay > 20%

            IF Silt > 20 THEN C% = 12: EXIT DO
            IF ClySnd > 1 THEN
                C% = 6: EXIT DO
            ELSE
                C% = 11: EXIT DO
            END IF
        END IF
    END IF
    PRINT "ERROR IN BRESLAU ROUTINE": BEEP: END  ' ya can't get heah from theya
  LOOP
SELECT CASE C%
  CASE 1
    C$ = "GRAVEL"
  CASE 2
    C$ = "GRAVELLY SEDIMENT"
  CASE 3
    C$ = "SAND"
  CASE 4
    C$ = "SILT"
  CASE 5
    C$ = "CLAY"
  CASE 6
    C$ = "SANDY CLAY"
  CASE 7
    C$ = "SILTY CLAY"
  CASE 8
    C$ = "CLAYEY SILT"
  CASE 9
    C$ = "SANDY SILT"
```

```
   CASE 10
      C$ = "SILTY SAND"
   CASE 11
      C$ = "CLAYEY SAND"
   CASE 12
      C$ = "SAND SILT CLAY"
   CASE ELSE
      C$ = "Error in classification routine"
END SELECT

PRINT "     Original     Revised"
PRINT USING "Gravel %     ##.##        ##.##"; VAL(Itext$(18)); Gravel
PRINT USING "Sand  %     ##.##        ##.##"; VAL(Itext$(17)); XSand
PRINT USING "Silt  %     ##.##        ##.##"; VAL(Itext$(19)); Silt
PRINT USING "Clay  %     ##.##        ##.##"; VAL(Itext$(20)); clay
   Total = Gravel + XSand + Silt + clay
   PRINT USING "            Total = ###.##"; Total
   PRINT
PRINT "Revised Classification:  "; C$
Verbal$ = C$
PRINT
   IF Lprnt THEN
   LPRINT "         Original     Revised"
   LPRINT USING "    Gravel % ##.##       ##.##"; VAL(Itext$(18)); Gravel
   LPRINT USING "    Sand   % ##.##       ##.##"; VAL(Itext$(17)); XSand
   LPRINT USING "    Silt   % ##.##       ##.##"; VAL(Itext$(19)); Silt
   LPRINT USING "    Clay   % ##.##       ##.##"; VAL(Itext$(20)); clay

   LPRINT USING "              Total = ###.##"; Total
   LPRINT
   LPRINT "    Revised Classification:  "; C$
   LPRINT

END IF

IF Cont% = 0 THEN PressAnyKey


'Scale the data for plotting
Ymax = 0
FOR I = 1 TO IP
   IF F(I) > Ymax THEN Ymax = F(I)
NEXT I
```

```
CALL BPLOT(0!, Ymax + 1.5, F(), IP)


STATS:
' now do the statistics via Schlee/Webster 1966

' Start with MODE
'IP = N        ' need if not using schlee-webster
Del1 = -.1
PRINT "Revised Statistics"
IF Lprnt THEN
   LPRINT
   LPRINT "    Revised Statistics"
END IF


FOR I = 2 TO IP
   Del2 = F(I) - F(I - 1)
   IF ((Del2 * Del1) < 0) AND (Del2 < 0) AND ((F(I - 1) - 5! * DPHI) > 0) THEN
      Xi = I - 2
      Xem = Emp1 + Xi * DPHI
      Xf = F(I - 1)
      I = I + 1
      PRINT "  Mode      "; Xem, Xf
      IF Lprnt THEN
         'LPRINT
         'LPRINT "  Mode      "; Xem, Xf
      END IF
   END IF
   Del1 = Del2
NEXT I

' MEDIAN
Sum = 0
FOR I = 1 TO IP
   Sum = F(I) + Sum
   IF (Sum - 50!) >= 0 THEN EXIT FOR
NEXT I
Xi = I - 1
Ct = Emp1 + Xi * DPHI
CtMed = Ct - (Sum - 50!) * DPHI / F(I) + .5 * DPHI
Fmt$ = "     \             \ =#####.##"
PRINT USING Fmt$; "Median"; CtMed
IF Lprnt THEN LPRINT USING Fmt$; "Median"; CtMed
```

```
' Moments about the Mean
Sum = 0: Sum1 = 0: Sum2 = 0: Sum3 = 0: Sum4 = 0
FOR I = 1 TO IP
   Xi = I - 1
   Ct = Emp1 + Xi * DPHI
   Sum = Sum + F(I)
   Sum1 = Sum1 + F(I) * Ct
   Sum2 = Sum2 + F(I) * Ct ^ 2
   Sum3 = Sum3 + F(I) * Ct ^ 3
   Sum4 = Sum4 + F(I) * Ct ^ 4
NEXT I
En1 = Sum1 / Sum
En2 = Sum2 / Sum
En3 = Sum3 / Sum
En4 = Sum4 / Sum
Zm2 = En2 - En1 ^ 2
Zm3 = En3 - 3! * En2 * En1 + 2! * En1 ^ 3
Zm4 = En4 + En1 * (-4 * En3 + 6! * En1 * En2 - 3! * En1 ^ 3)
DPHI2 = DPHI * DPHI
Zm4 = Zm4 - .5 * DPHI2 * Zm2 + .02916667# * DPHI2 * DPHI2
Zm2 = Zm2 - DPHI2 / 12!
'PRINT En1; En2; En3; En4; Zm2; Zm3; Zm4

' Standard Deviation
Sigma = SQR(Zm2)
' Skewness
Skew = .5 * Zm3 / (Sigma * Zm2)
' Kurtosis
ZKurt = Zm4 / Zm2 ^ 2 - 3!
' Print those puppies

PRINT USING Fmt$; "Mean"; En1
PRINT USING Fmt$; "Standard Deviation"; Sigma
PRINT USING Fmt$; "Skewness"; Skew
PRINT USING Fmt$; "Kurtosis"; ZKurt
PRINT

IF Lprnt THEN
   LPRINT USING Fmt$; "Mean"; En1
   LPRINT USING Fmt$; "Standard Deviation"; Sigma
   LPRINT USING Fmt$; "Skewness"; Skew
   LPRINT USING Fmt$; "Kurtosis"; ZKurt
   LPRINT CHR$(12)                    ' Form Feed
END IF
```

```
' write the corrected data to disk
IF FileWrite% THEN
   FOR I = 1 TO 16
      PRINT #2, Itext$(I); ",";          ' Lab# - Samp Wt.
   NEXT I
   PRINT #2, USING "###.##,"; Sand; Gravel; Silt; clay;
   PRINT #2, Verbal$; ",";
   PRINT #2, USING "###.##,"; CtMed; En1; Sigma; Skew; ZKurt;
   'Print #2, "9999,9999,9999,0,"  ' NEEDS WORK AHE - Killed LJP 10/96
   IF ExType = 1 THEN PRINT #2, "LINEAR,";
   IF ExType = 3 THEN PRINT #2, "EXPONENTIAL,";
   IF ExType = 4 THEN PRINT #2, "MEAN_L_E,";

   FOR I = 19 TO 1 STEP -1
      PRINT #2, USING "###.##,"; FPcor100(I);  'round to 2 dp'
      NEXT I
   PRINT #2, "$"
END IF



IF Cont% = 0 THEN PRINT "For next Sample - "; : PressAnyKey
' See if the file is at end of data

DO WHILE NOT EOF(1)
   INPUT #1, LN$
   IF LN$ <> "" THEN GOTO NextSamp        ' start next record
LOOP
CLS
CLOSE #1
CLOSE #2
PRINT
PRINT "End of File"
INPUT "Enter 'C' to Continue or 'Q' to quit program. [Q] ", A$
IF UCASE$(LEFT$(A$, 1)) = "C" THEN RESTORE: GOTO NextFile

END

' Label Data
DATA Lab Number,Field ID,Project ID,Cruise ID,Requestor,Month,Day,Year
DATA Latitude,Longitude,Sample Device,Location,Depth,Top,Bottom,Sample Weight
DATA % Sand,% Gravel,% Silt,% Clay,Verbal Equivalent,Median,Mean
DATA Standard Deviation,Skewness,Kurtosis,Mode 1,Mode 1 Frequency %
DATA Mode 2,Mode 2 Frequency %,Mode 3,Mode 3 Frequency %,Number of Modes
```

```
DATA "11 Phi","11 Phi Cumulative Fq %","10 Phi","10 Phi Cummulative Fq %"
DATA " 9 Phi"," 9 Phi Cumulative Fq %"," 8 Phi"," 8 Phi Cummulative Fq %"
DATA " 7 Phi"," 7 Phi Cumulative Fq %"," 6 Phi"," 6 Phi Cummulative Fq %"
DATA " 5 Phi"," 5 Phi Cumulative Fq %"," 4 Phi"," 4 Phi Cummulative Fq %"
DATA " 3 Phi"," 3 Phi Cumulative Fq %"," 2 Phi"," 2 Phi Cummulative Fq %"
DATA " 1 Phi"," 1 Phi Cumulative Fq %"," 0 Phi"," 0 Phi Cummulative Fq %"
DATA "-1 Phi","-1 Phi Cumulative Fq %","-2 Phi","-2 Phi Cummulative Fq %"
DATA "-3 Phi","-3 Phi Cumulative Fq %","-4 Phi","-4 Phi Cummulative Fq %"
DATA "-5 Phi","-5 Phi Cumulative Fq %",End Flag
DATA ""

    '**********************************************************
    '** Name:        BPLOT                      **
    '** Type:        MAIN [  ] FUNCTION [  ] SUB [ X ]  **                    **
    '** Language:    Microsoft QuickBASIC 4.00 or above     **
    '** Library :    Standard                   **
    '** Arguments:                  **
    '** Programmer:  A. Eliason- Eliason Data- (508)477-1400 **
    '** Date:        Version # in main module includes date  **
    '**              as:  MMDDYYVV                 **
    '** Description:  Please see comments within this module  **
    '**                                **
    '** Usage:       If no info here please see the line     **
    '**              comments.                  **
    '**********************************************************
    '
    '
    '
    '
    '
    '**********************************************************
    '** Name:        BPLOT                      **
    '** Type:        MAIN [  ] FUNCTION [  ] SUB [ X ]  **                    **
    '** Language:    Microsoft QuickBASIC 4.00 or above     **
    '** Programmer:  A. Eliason- Eliason Data- (508)477-1400 **
    '** Date:        Version # in main module includes date  **
    '**              as:  MMDDYYVV                 **
    '** Description:  Please see comments within this routine **
    '**                                **
    '** Usage:       If no info here please see the line     **
    '**              comments.                  **
    '**********************************************************
SUB BPLOT (TMIN, TMAX, X(), N)
SHARED Lprnt, Cont%
```

```
    M = 0
    SPAN = TMAX - TMIN
    UNIT = SPAN / 70
    FOR I = 1 TO N
        M = M + 1
        IF M = 20 THEN M = 0: IF Cont% = 0 THEN PressAnyKey
        ISPC = (X(I) - TMIN) / UNIT
        Fract = (I - 1) / 5 - 5
        IFract = Fract - .4999
        'PRINT USING "## ##.##### "; IFract; X(i);
        PRINT USING "##.##### "; X(I);
        PRINT SPC(ISPC); "*"    ';         ' SPC(70 - ISPC);
        'IF Lprnt THEN
        '    LPRINT USING "## ##.##### "; IFract; X(i);
        '    LPRINT SPC(ISPC); "*"    ';         ' SPC(70 - ISPC);
        'END IF
    NEXT I
    PRINT
END SUB

SUB CPI (H(), Q(), N, DELH, F(), M, DELHC)
SHARED Diag%

Q(1) = 0
Q(N + 2) = 0
    FOR j = 2 TO N
        DELQ = (Q(j + 1) - Q(j)) / DELH
        Y = 3! * Q(j) - 3! * Q(j + 1) + Q(j + 2)
        Z = H(j)
        IMAX = DELH / DELHC  '= number of Extrapolated points per interval
        FOR I = 1 TO IMAX
            JP = (Z - H(2)) / DELHC + 1.4999   ' Prevent rounding error
            'PRINT "*********** JP= "; JP; " ": INPUT A$
            XK = (Z - H(j)) / DELH
            X = XK + 1!
            A = Q(j - 1) + XK * (Y - Q(j - 1))
            FC = A + X * (Q(j) - A + .5 * (Q(j + 1) - 2! * Q(j) + A) * XK)
            IF (j - 2) = 0 THEN GOTO Cont12
            IF (j - N) <> 0 THEN GOTO Cont5
Cont12:
                ' Test for Extrapolated value less than zero
                'PRINT FC
                IF FC < 0 THEN
                Z = H(j) + DELHC
```

```
            IF (j - 2) = 0 THEN
                Q(j) = .01
                PRINT "Exponential Extrapolation at Beginning"
                GOTO Cont7
            END IF

            IF (j - N) = 0 THEN
                Q(j + 1) = .01
                PRINT "Exponential Extrapolation at End"
            ELSE
                GOTO Cont5
            END IF
Cont7:
            IF Q(j + 1) = 0 THEN Q(j + 1) = .000001 'prevent div by zero
            IF Q(j) = 0 THEN Q(j) = .000001
            SLPE = LOG(Q(j) / Q(j + 1)) / DELH
            FOR II = 2 TO IMAX
                JP = (Z - H(2)) / DELHC + 1.4999
                ANTI = SLPE * (Z - H(j))
                F(JP) = Q(j) / EXP(ANTI)
                Z = Z + DELHC
            NEXT II
            GOTO Cont1:
          END IF
Cont5:      F(JP) = FC

Cont2:  Z = Z + DELHC
      NEXT I

Cont1:
    NEXT j
    F(JP + 1) = 0
IF Diag% THEN PRINT "CPI WAS RUN. EXITING CPI."




END SUB

SUB CSCOEF (N, X(), Y(), s(), index()) STATIC
' Copyright (c) 1993 Crescent Software
  ' Cublic spline coefficients subroutine
```

```
' Input

' N   = number of X and Y data points
' X() = array of X data points (N rows by 1 column)
' Y() = array of Y data points (N rows by 1 column)

' Output

' S()     = array of cubic spline coefficients
'           (N rows by 1 column)
' INDEX() = array of indices (N rows by 1 column)

REDIM RHO(N), TAU(N)

NM1 = N - 1

FOR I = 1 TO N
   index(I) = I
NEXT I

' ascending order data sort

FOR I = 1 TO NM1
   IP1 = I + 1
   FOR j = IP1 TO N
      II = index(I)
      IJ = index(j)
      IF (X(II) > X(IJ)) THEN
        ITEMP = index(I)
        index(I) = index(j)
        index(j) = ITEMP
      END IF
   NEXT j
NEXT I

NM2 = N - 2

RHO(2) = 0#
TAU(2) = 0#

FOR I = 2 TO NM1
   IIM1 = index(I - 1)
   II = index(I)
   IIP1 = index(I + 1)
```

```
    HIM1 = X(II) - X(IIM1)
    HI = X(IIP1) - X(II)
    Temp = (HIM1 / HI) * (RHO(I) + 2#) + 2#
    RHO(I + 1) = -1# / Temp
    D = 6# * ((Y(IIP1) - Y(II)) / HI - (Y(II) - Y(IIM1)) / HIM1) / HI
    TAU(I + 1) = (D - HIM1 * TAU(I) / HI) / Temp
  NEXT I

  s(1) = 0#
  s(N) = 0#

  ' compute cubic spline coefficients

  FOR I = 1 TO NM2
    IB = N - I
    s(IB) = RHO(IB + 1) * s(IB + 1) + TAU(IB + 1)
  NEXT I

  ERASE RHO, TAU


END SUB

SUB CSFIT1 (N, X(), Y(), s(), index(), X1, sx) STATIC
' Copyright (c) 1993 Crescent Software

  ' Cublic spline Extrapolation subroutine

  ' Input

  '  N      = number of X and Y data points
  '  X()    = array of X data points (N rows by 1 column)
  '  Y()    = array of Y data points (N rows by 1 column)
  '  S()    = array of cubic spline coefficients
  '           (N rows by 1 column)
  '  INDEX() = array of indices (N rows by 1 column)
  '  X1     = X data value to fit

  ' Output

  '  SX = cubic spline Extrapolated value for X

  FOR I = 2 TO N
    II = index(I)
```

```
    IF (X1 <= X(II)) THEN EXIT FOR
  NEXT I

  L = I - 1
  IL = index(L)
  ILP1 = index(L + 1)

  A = X(ILP1) - X1
  B = X1 - X(IL)

  HL = X(ILP1) - X(IL)

  sx = A * s(L) * (A * A / HL - HL) / 6# + B * s(L + 1) * (B * B / HL - HL) / 6# + (A * Y(IL) + B
* Y(ILP1)) / HL

END SUB

SUB CSFIT2 (N, X1(), Y1(), YP1, YPN, X, Y) STATIC
' Copyright (c) 1989, 1990 Crescent Software

  ' Clamped cubic spline Extrapolation subroutine

  ' Input

  '  N   = number of X and Y data points
  '  X1() = vector of X data points ( N rows )
  '  Y1() = vector of Y data points ( N rows )
  '  YP1 = derivative at data point 1
  '  YPN = derivative at data point N
  '  X   = X data point to fit

  ' Output

  '  Y = Extrapolated Y data point

  ' NOTE: X1(1) < X1(2) < X1(3) < ... < X1(N)

  REDIM U(N), YPP(N)

  YPP(1) = -.5
  U(1) = (3# / (X1(2) - X1(1))) * ((Y1(2) - Y1(1)) / (X1(2) - X1(1)) - YP1)

  FOR I = 2 TO N - 1
    SIG = (X1(I) - X1(I - 1)) / (X1(I + 1) - X1(I - 1))
```

```
    P = SIG * YPP(I - 1) + 2#
    YPP(I) = (SIG - 1#) / P
    U(I) = (6# * ((Y1(I + 1) - Y1(I)) / (X1(I + 1) - X1(I)) - (Y1(I) - Y1(I - 1)) / (X1(I) - X1(I - 1)))
/ (X1(I + 1) - X1(I - 1)) - SIG * U(I - 1)) / P
  NEXT I

  QN = .5
  UN = (3# / (X1(N) - X1(N - 1))) * (YPN - (Y1(N) - Y1(N - 1)) / (X1(N) - X1(N - 1)))

  YPP(N) = (UN - QN * U(N - 1)) / (QN * YPP(N - 1) + 1#)

  FOR K = N - 1 TO 1 STEP -1
    YPP(K) = YPP(K) * YPP(K + 1) + U(K)
  NEXT K

  ' Extrapolate

  KLO = 1
  KHI = N

  WHILE (KHI - KLO > 1)
    K = (KHI + KLO) / 2#
    IF (X1(K) > X) THEN
      KHI = K
    ELSE
      KLO = K
    END IF
  WEND

  H = X1(KHI) - X1(KLO)

  A = (X1(KHI) - X) / H
  B = (X - X1(KLO)) / H

  Y = A * Y1(KLO) + B * Y1(KHI) + ((A ^ 3 - A) * YPP(KLO) + (B ^ 3 - B) * YPP(KHI)) * (H
^ 2) / 6#

  ERASE U, YPP

END SUB

SUB FFIT (N, X(), Y(), IType, A, B) STATIC

  ' Function curve fit subroutine
```

```
' Input

'  N   = number of data points (N >= 3)
'  X() = vector of X data points ( N rows )
'  Y() = vector of Y data points ( N rows )
'  ITYPE = type of curve fit
'     1 = linear       Y = A + B * X
'     2 = logarithmic  Y = A + B * LOG(X)
'     3 = exponential  Y = A * EXP(B * X)

' Output

'  A, B = coefficients of curve fit

X1 = 0#
Y1 = 0#
Z = 0#
X2 = 0#
Y2 = 0#

SELECT CASE IType
CASE 1
   ' Linear
   FOR I = 1 TO N
      X1 = X1 + X(I)
      Y1 = Y1 + Y(I)
      X2 = X2 + X(I) * X(I)
      Y2 = Y2 + Y(I) * Y(I)
      Z = Z + X(I) * Y(I)
   NEXT I

   X1 = X1 / N
   Y1 = Y1 / N
   B = (Z - N * X1 * Y1) / (X2 - N * X1 * X1)
   A = Y1 - B * X1
CASE 2
   ' Logarithmic
   FOR I = 1 TO N
      XLOGX = LOG(X(I))
      X1 = X1 + XLOGX
      Y1 = Y1 + Y(I)
      X2 = X2 + XLOGX * XLOGX
      Y2 = Y2 + Y(I) * Y(I)
      Z = Z + XLOGX * Y(I)
```

```
      NEXT I

      X1 = X1 / N
      Y1 = Y1 / N

      B = (Z - N * X1 * Y1) / (X2 - N * X1 * X1)
      A = Y1 - B * X1
   CASE 3
      ' Exponential
      FOR I = 1 TO N
         XLOGY = LOG(Y(I))
         X1 = X1 + X(I)
         X2 = X2 + X(I) * X(I)
         Y1 = Y1 + XLOGY
         Y2 = Y2 + XLOGY * XLOGY
         Z = Z + X(I) * XLOGY
      NEXT I

      X1 = X1 / N
      Y1 = Y1 / N

      B = (Z - N * X1 * Y1) / (X2 - N * X1 * X1)
      A = EXP(Y1 - B * X1)
   END SELECT

END SUB

DEFSNG A-H, O-Z
SUB PressAnyKey
X% = POS(0)
Y% = CSRLIN
PRINT "Press any key to continue.";
DO
LOOP WHILE INKEY$ = ""
LOCATE Y%, X%
PRINT "                          ";
LOCATE Y%, X%
END SUB

DEFDBL A-H, O-Z
FUNCTION U2Phi (Usize)
' convert micron size to phi

   phi = -2.12720466241656D-16 + (-1.44269504088896#) * LOG(Usize / 1000)
```

```
    'PRINT USING "###.######### mm = "; Usize;
    'PRINT USING "###.### phi"; Phi
    U2Phi = phi
END FUNCTION
```