

INVESTIGATIONS IN GEOSTATISTICAL SIMULATION

AS AN AID TO MINE PLANNING

A thesis submitted to the
Imperial College of Science and Technology
for the award of Doctor of Philosophy

by

Richard Phillip Bryan Durham B.Sc.(Eng) A.R.S.M.

January 1980

ABSTRACT

Richard Durham

Investigations in geostatistical simulation as an aid to mine planning

The geostatistical 'turning bands' simulation technique is examined. The simulations are conditioned, and the importance of the conditioning data points is shown. The correlation between the accuracy of a simulation before and after conditioning is examined.

The standard 'turning bands' three dimensional simulation technique involves the use of fifteen regularly orientated axes. A new technique for finding the fifteen co-ordinates of a point is developed. It is tested against other one dimensional axes configurations.

Conditional simulations are made of stopes from a lode within South Crofty tin mine. Non-stationarity amongst the data is found to be overcome by the conditioning process.

A block simulation method is developed which allows the same unconditional simulations to be used for several sites. It produces a large number of simulations at a relatively low cost.

CONTENTS

LIST OF TABLES	I
LIST OF FIGURES	II
ACKNOWLEDGEMENTS	1
INTRODUCTION	2
SECTION A : THE SIMULATION METHOD	
1.SELECTION OF THE SIMULATION MODEL	
1.1.Traditional methods	5
1.2.Turning bands method	7
2.ONE DIMENSIONAL SIMULATIONS	
2.1.The simulation technique	9
2.2.Determination of 'B'	12
2.3.Computer subroutine SIM	13
3.COMPOSITION OF THE ONE DIMENSIONAL SIMULATIONS INTO THREE DIMENSIONS	
3.1.The composition technique	15
3.2.Number and orientation of axes	17
3.3.Unconditional simulation example	18
3.4.The uses of unconditional simulations	19
4.CONDITIONING OF THE SIMULATIONS	
4.1.The conditioning technique	21
4.2.The importance of the number of data points	23

4.3.The importance of the position of the data points	25
4.4.Merits of conditioning	27
4.5.Simulations before and after conditioning	28
4.6.The cost of conditioning	31
5.ALTERNATIVE ONE DIMENSIONAL AXES CONFIGURATIONS	
5.1.The importance of the axes configuration	33
5.2.Ten regular axes	33
5.3.Fifteen regular axes	34
5.4.Randomly orientated axes	37
5.5.Comparison of one dimensional axes configurations	38
5.6.Selection of axes configuration	43

SECTION B : APPLICATIONS IN SOUTH CROFTY MINE

1.SOUTH CROFTY MINE	
1.1.General description	45
1.2.Available data	46
2.DATA ANALYSIS	
2.1.Data accuracy	49
2.2.Conversion of imperial measurements	53
2.3.Calculation of average assay values	55
2.4.Semi-variogram analysis	57
2.5.Preparation of data for simulation	63
2.6.Lode widths simulation	63
2.7.Lode assays simulation	71
2.8.Actual widths estimation	74
2.9.Actual assays estimation	78

3.STOPE SIMULATION	
3.1.General approach	82
3.2.Representation of the stope	82
3.3.Simulation programs	83
3.4.Conditional simulation results	84
3.5.Susceptibility of the simulations to the models	92
3.6.Costs of the simulations	95
3.7.Actual width and actual assay simulations	95
4.BLOCK SIMULATION	
4.1.General approach	101
4.2.Choice of block size	102
4.3.Choice of number of grid points	102
4.4.Production and storage of unconditional simulations	104
4.5.Choice of kriging pattern	106
4.6.Conditional simulation results	108
4.7.Sensitivity of the block simulation method	112
4.8.Costs of the simulations	118
4.9.Conditioning with grouped data	120
CONCLUSIONS	122
APPENDICES	128
REFERENCES	148

LIST OF TABLES

- 1 - Correlation between simulations
before and after conditioning
- 2 - Comparison of one dimensional axes configurations
- 3 - Comparison of one dimensional axes configurations
- 4 - Trimming the semi-variogram values
- 5 - Development ln lode widths trend surfaces
- 6 - Stope simulation results
- 7 - Accuracy of different kriging patterns
- 8 - Block simulation results
- 9 - Sensitivity of block simulation method
- 10 - Number of simulations required
- 11 - Conditioning with grouped data

LIST OF FIGURES

- 1 - Conditional simulation
- 2 - Spherical semi-variogram models
- 3 - One dimensional simulation axis
- 4 - One dimensional simulations
- 5 - Three dimensional unconditional simulation
- 6 - Simulations before conditioning
- 7 - Simulations after conditioning to points
at 20 metres intervals
- 8 - Simulations after conditioning to points
at 10 metres intervals
- 9 - Simulations after conditioning to points
at 5 metres intervals
- 10 - Simulations after conditioning to points
at 20,5,10 metres intervals
- 11 - Simulations after conditioning to points
at 10,5,20 metres intervals
- 12 - Different conditioning data points
- 13 - Scattergram of variance vs sill
- 14 - Scattergram of MS(model) vs |Sill-20|
- 15 - Scattergram of MS(data before) vs MS(data after)
Conditioning points at 10 metres intervals
- 16 - Scattergram of MS(data before) vs MS(data after)
Conditioning points at 20 metres intervals

- 17 - Scattergram of MS(data before) vs MS(data after)
Conditioning points at 30 metres intervals
- 18 - Scattergram of MS(data before) vs MS(data after)
Conditioning points at 40 metres intervals
- 19 - Influence of a one dimensional simulation axis
- 20 - Semi-variograms of unconditional simulations
- 21 - Semi-variogram of all conditioning data
- 22 - Different axes configurations
Conditioning points every 10 metres
- 23 - Different axes configurations
Conditioning points every 10 metres
- 24 - Different axes configurations
Conditioning points every 10 metres
- 25 - Different axes configurations
Conditioning points every 10 metres
- 26 - Different axes configurations
Conditioning points every 20 metres
- 27 - Different axes configurations
Conditioning points every 30 metres
- 28 - Different axes configurations
Conditioning points every 40 metres
- 29 - Histogram of figures after decimal point
for imperial lode widths
- 30 - Histogram of figures after decimal point
for metric lode widths
- 31 - Histogram of last digit of imperial lode assays
between 50 and 100 lbs/ton

- 32 - Histogram of last digit of metric lode assays
- 33 - Semi-variograms from every fifth sample
- 34 - Minimum/maximum envelope from semi-variograms
from 100 consecutive samples - simulation 2637
- 35 - Minimum/maximum envelope from semi-variograms
from 100 consecutive samples - lode widths
- 36 - Minimum/maximum envelope of semi-variograms
after random removal of samples - simulation 2637
- 37 - Minimum/maximum envelope of semi-variograms
after random removal of samples - lode widths
- 38 - Histogram of $[G(x)-G(x+h)]^2$ distributions
- 39 - Effect of trimming the semi-variogram values
- 40 - Semi-variograms in different directions
for stope lode widths
- 41 - Semi-variograms from different levels
for lode widths
- 42 - Histogram of imperial development lode widths
- 43 - Semi-variograms from different levels
for ln lode widths
- 44 - Semi-variogram of residuals of development
ln lode widths cubic trend surface
- 45 - Histogram of residuals of development
ln lode widths cubic trend
- 46 - Semi-variograms in different directions
for stope lode assays
- 47 - Semi-variograms from different levels
for lode assays

- 48 - Histogram of lode values
- 49 - Semi-variograms from different levels
for ln lode assays
- 50 - Semi-variogram of development ln lode assays
- 51 - Conditional simulations of lode widths
for Meredew stope
- 52 - Conditional simulations of lode assays
for Meredew stope
- 53 - Conditional simulations of lode widths
for Pisowocki stope
- 54 - Conditional simulations of lode assays
for Pisowocki stope
- 55 - The minimum/maximum envelope
for Meredew stope simulation no.1
- 56 - Conditional simulations of lode widths
for Meredew stope ignoring trend
- 57 - Conditional simulations of lode widths
for Pisowocki stope ignoring trend
- 58 - Conditional simulations of lode widths
for Meredew stope ignoring trend with different models
- 59 - Conditional simulations of lode assays
for Meredew stope with different models
- 60 - Conditional simulations of actual widths
for Meredew stope
- 61 - Conditional simulations of actual assays
for Meredew stope

- 62 - Conditional simulation of
two horizontally adjacent blocks
- 63 - Block conditional simulation layout

ACKNOWLEDGEMENTS

By definition the work involved in attaining a Ph.D. degree is original and unaccompanied. However, this thesis would not have been written without the help of several people.

Thanks are due to my supervisor, Dr. Isobel Clark, and to South Crofty Limited for permission to use their data.

Paul Bearman assisted with my vocabulary, and proof reading was carried out by Margaret Burr.

Grateful acknowledgement is made to the Science Research Council for financial support during the course of this research.

INTRODUCTION

A simulation is a representation of reality. It is a possible existing situation which exhibits imposed characteristics.

An unconditional simulation displays the known characteristics of the sampled data values. A conditional simulation goes one step further. It exhibits the same characteristics, and at the data points the simulated value equals the sampled real value.

The purpose of any simulation approach is to aid understanding, prediction, and control of a system. When simulating ore reserves this system is the geological genesis of the rocks themselves. Obviously, control of this process is impossible, and greater understanding and prediction are left as the aims of such simulations.

Simulation is not the same as estimation. A simulation produces a possible situation, whereas estimation derives the most likely situation. For any given set of data values, there exists an infinite number of simulations, but only one estimation. The average of all the simulations (possible situations) is equal to the estimation (probable situation).

Since a single simulation is only one of an infinite number of possible simulations, to create only one simulation and to draw conclusions from it can be extremely misleading. It is equivalent to examining a single value

from a distribution. A sufficiently large number of simulations must be created to give an idea of the distribution of features around the mean discovered by estimation. The possibility of taking decisions on the basis of a single exceptional simulation is thereby avoided.

The major use of simulating ore reserves is for the investigation of possible fluctuations. These fluctuations, for instance, can be between the grade of adjacent mining blocks, or the hoisted production from contiguous shifts.

Consider two blocks of ore next to each other. Estimation can inform as to whether each of them is below or above the cut off grade. But, it can not predict what is the probability of both of them being of the same classification (pay or non-pay). Simulation can do this. For instance, the results from several simulations might show that there is only a ten percent chance of the two mining blocks being of different classifications. Under these conditions, the mining block size could be doubled and the mining technique thereby simplified.

One of the problems with statistical simulation has always been that of cost. Due to the sheer number of figures which have to be manipulated, the use of a fairly large computer is essential. It is the cost of the computer time and storage which creates the expense of statistical simulation. This is due to the large number, rather than the complexity, of the calculations.

Throughout the course of this study the cost of the simulations was always considered. It is essential that this cost does not exceed the saving due to a simulation based decision.

All the computer programs used in this study are in the FORTRAN IV language, and were compiled by a University of Minnesota MNF5 compiler. The computer work was undertaken at the Imperial College computer complex. The machinery available consisted of a CDC Cyber 174 and a CDC 6500. These two computers share 250,000 60-bit words of extended memory space. The execution time of the computer programs used in this study are quoted in terms of central processor seconds (henceforth CP seconds).

This thesis has been written in two sections. The first consists of a detailed evaluation of the simulation method. The second shows how such a simulation technique has been applied to some mine data.

SECTION A : THE SIMULATION METHOD

A.1. SELECTION OF THE SIMULATION MODEL

A.1.1. Traditional methods :-

There are many different types of statistical simulation models. They broadly fall into four categories (Harbaugh and Bonham-Carter, 1970) as follows :-

- (a) Static - independent of time.
- (b) Dynamic - varying with time.
- (c) Deterministic - with no element of chance.
- (d) Probabilistic - involving a random component.

A simulation model is either static or dynamic, and either deterministic or probabilistic.

A model was required to produce simulations of geological variables. For the purposes of this study these variables were lode widths and lode assays. Both of these parameters do not change with time, and so a static model was needed. Also, neither of them is precisely known at all points, and thus the model had to be probabilistic.

The standard approach to a static probabilistic model is a Monte Carlo method. This involves taking samples at random from a known distribution. Thus, it is a relatively easy matter to produce simulated values which exhibit a known distribution.

However, if geological variables are to be realistically modelled, they must be spatially correlated

as well. In other words, they must exhibit an imposed auto-correlation function.

There are several methods for producing three dimensional, spatially correlated simulations (Jenkins and Watts,1968; Journel,1974). They are generally of two types :-

(a) Simulations of orthogonal random numbers which have a variance density equal to the spectrum density of the desired auto-correlation function.

(b) Moving averages, usually performed with a sphere, over a field with a Poisson distribution.

However, these simulation methods are extremely costly on computer space and time. Oertel and Walton (1967) in 1967 stated that their program 'exceeds the directly accessible memory space of present computers', and 'the use of indirect memory space.....would cause intolerably long running times'. Great advances in computer technology since 1967 no doubt mean that these simulation methods could now at least be handled, but they would still be relatively huge undertakings. They are generally considered to be prohibitively expensive.

However, these methods can be used to produce one dimensional simulations at a reasonable cost. It is only when they are asked to perform in three dimensions that they become expensive. It is the transfer from one to three dimensions which causes an 'explosion' of computer time and storage requirements.

Newton and Royle (1972) describe a method of producing a two dimensional simulation at a reasonable cost. It is based on a trigonometrical expression and could be expanded into three dimensions fairly simply and cheaply. However, it seems that the behaviour of the simulated values can only be found after the simulation has taken place. No desired function, either of auto-correlation or of distribution, can be imposed on the simulation.

A.1.2. Turning bands method :-

In 1973 Matheron (1973) first postulated the 'turning bands' simulation approach. Later Journel and Huijbregts (1978) presented the method in more detail, and more comprehensibly. The 'turning bands' method does not need enormous computer facilities to produce three dimensional simulations. The simulations have the required characteristics, that is of the imposed distribution and auto-correlation functions. The auto-correlation function used in the 'turning bands' method is the semi-variogram. This function is made up of a series of figures calculated from :-

$$\text{Gamma}(h) = \frac{1}{2 \cdot N} \sum_{i=1}^{i=N} [G(x) - G(x+h)]^2$$

where h = vector commonly called the lag

Gamma(h) = semi-variogram value at lag 'h'

G(x) = variable value at position 'x'

G(x+h) = variable value at position 'x+h'

N = number of pairs of values
separated by 'h'

Determination of this figure for various lag values produces the semi-variogram, usually presented as a graph of ' $\Gamma(h)$ ' against 'h'.

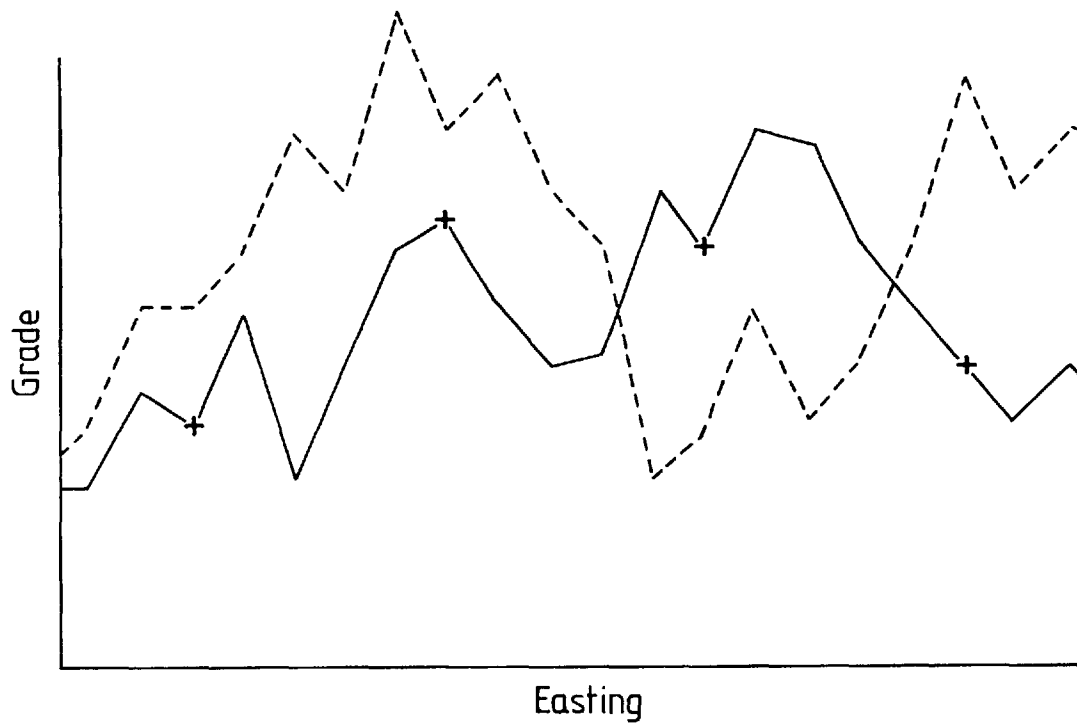
The 'turning bands' simulation method basically consists of two steps. The first is to produce simulations along one dimensional lines. The second is to rotate these lines in space and transfer their characteristics into three dimensions. It is the one to three dimensions technique which is the great originality, and cost saver, of the 'turning bands' approach.

Journel (1974) gives details of how a 'turning bands' simulation can be made conditional. That is, the simulated value at any sampled point equals the actual variable value at that point (see figure 1). The method uses the kriging estimation technique, as developed by Krige (1951), and described by Delfiner and Delhomme (1973), and Brooker (1979).

All the simulations used in this study were produced by the 'turning bands' technique. The next three sections describe and evaluate the method in detail.

FIGURE 1: CONDITIONAL SIMULATION

+ Real data values
----- Unconditional simulation
————— Conditional simulation



A.2.ONE DIMENSIONAL SIMULATIONS

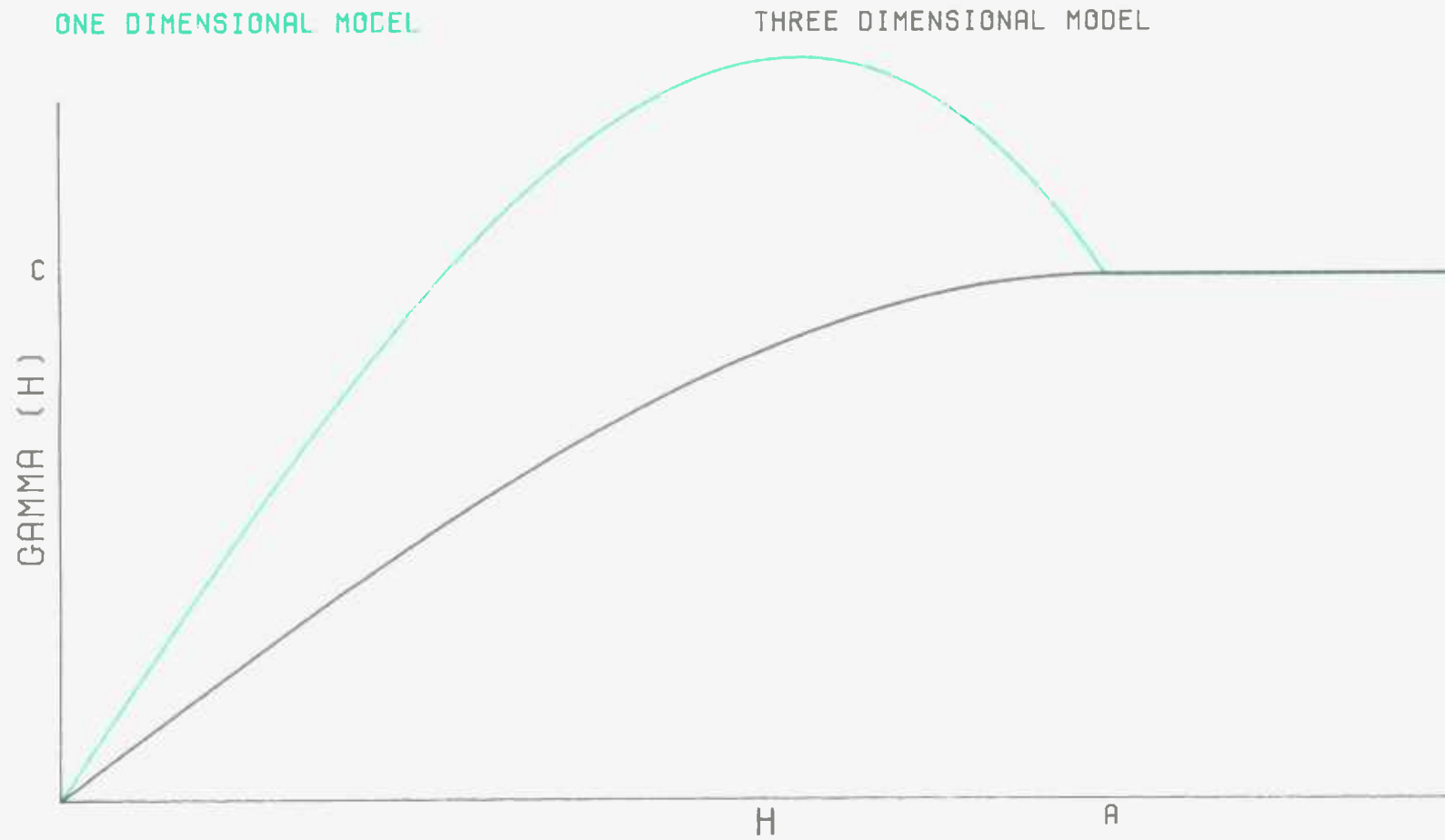
A.2.1.The simulation technique :-

The first step in the 'turning bands' simulation method is to produce one dimensional simulations. Each of these consists of a set of figures along a straight line. The values on the line must exhibit the imposed distribution and auto-correlation functions.

The 'turning bands' approach produces simulated values which follow a Normal or Gaussian distribution. Henceforth the notation $N(M,V)$ will be used to denote a Normal distribution with a mean of 'M' and a variance of 'V'.

As mentioned before the auto-correlation function used is the semi-variogram. As with distributions, there exist several standard models for the shape of a semi-variogram (Blais and Carlier,1968; Huijbregts,1973). The one used throughout this study is the spherical or Matheron model (see figure 2). Of all the semi-variogram models, the spherical scheme is by far the most frequently used. David and Blais (1972) state that 'In all the cases we have studied, we have been able to fit spherical schemes to our experimental variogram curves'. The equation for the spherical model is :-

FIGURE 2 : SPHERICAL SEMI-VARIOGRAM MODELS



$$= (C-E) * [1.5 * \frac{H}{A} - 0.5 * \frac{H^3}{A^3}] + E$$

where C = total sill

E = nugget effect

H = lag value

A = range of influence

The spherical model is defined by the three parameters of range, sill, and nugget effect. Henceforth the notation Spherical(A,C,E) will be used to indicate the precise shape of any spherical model.

There are many simple ways of simulating independent values which follow a certain distribution. The problem arises when the values have to be spatially correlated.

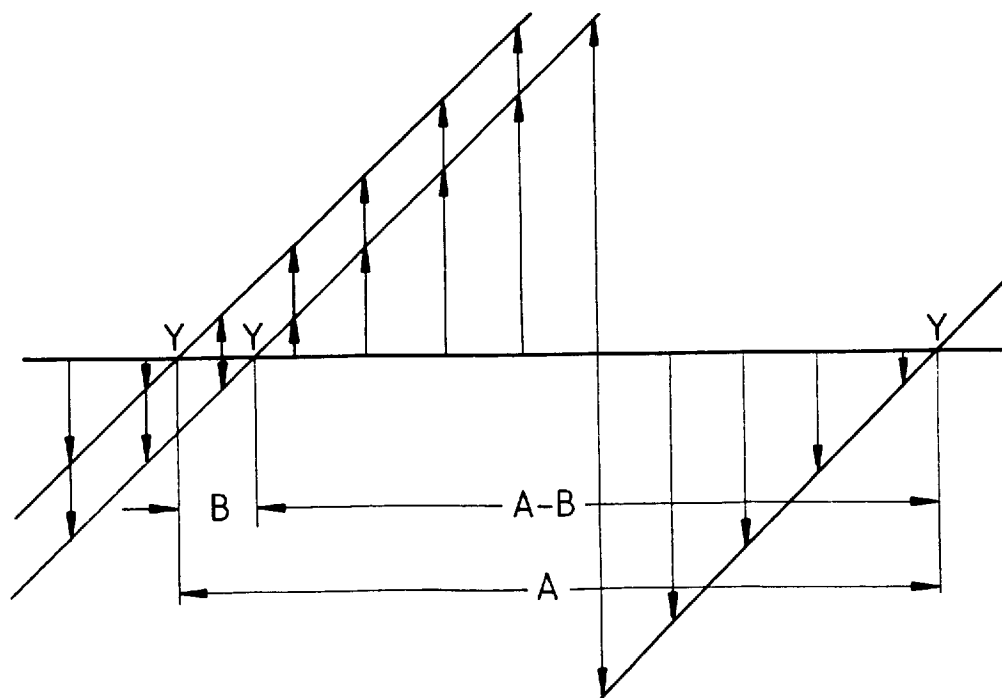
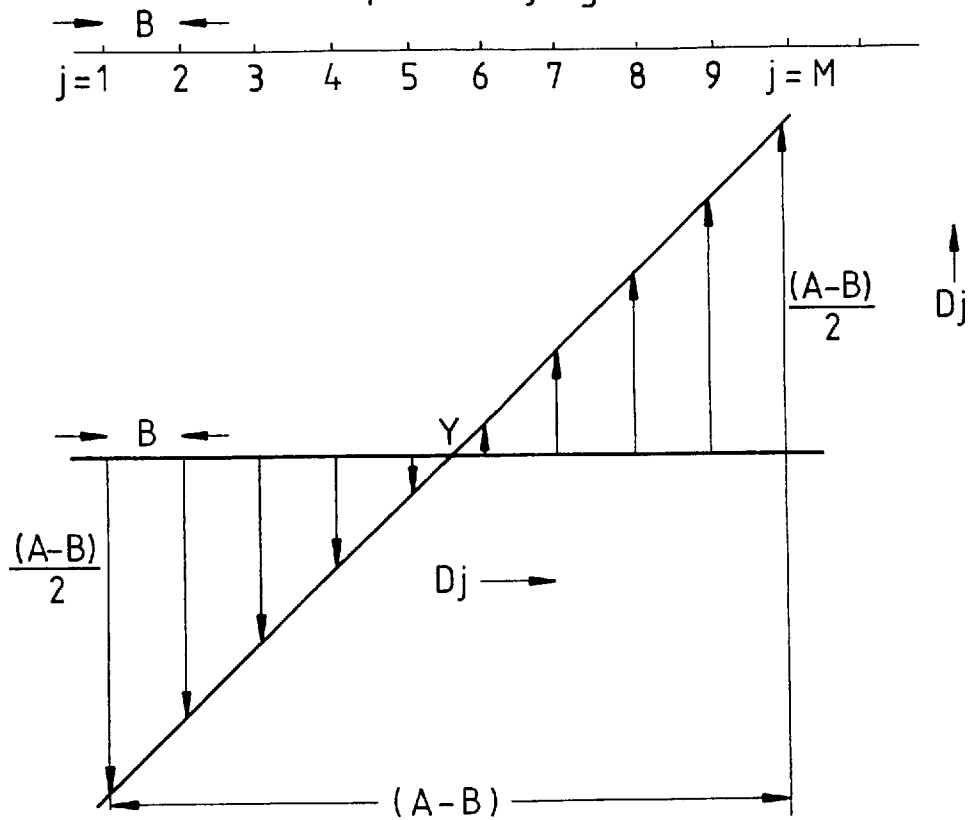
The initial step is to produce a series of independent values with a Normal distribution of $N(0,1)$. The computer available can produce random numbers from a uniform distribution between 0.0 and 1.0. The sum of twelve such numbers has a Normal distribution of $N(6.0,1.0)$. Subtraction of 6.0 produces the desired distribution shape.

The independent, or 'T', values are placed along a line at regular intervals of 'B'. A parallel line of 'Y' values is now considered. They are also regularly spaced at intervals of 'B' but are out of alignment with the 'T' values by ' $0.5*B$ '. Each 'Y' value is a function of the closest 'T' figures. In fact all the 'T' figures up to a distance of ' $(A-B)/2$ ' in both directions are included (see figure 3). By this means, two 'Y' values at a small

FIGURE 3

$A=10.0$ $B=1.0$

Independent T_j figures



distance apart have a large number of common 'T' figures, and are therefore heavily correlated. As the distance increases the correlation reduces to a minimum of zero at distances of 'A' or greater. Each 'Y' value is defined as :-

$$= WF * \sum_{j=1}^{j=M} T_j * D_j$$

where WF = weighting factor

M = number of 'T' values considered
in both directions

$$= \frac{2 * (A-B)}{B} + 1 = \frac{A}{B}$$

T_j = 'T' value at point 'J'

D_j = distance from point 'J' to the
position of 'Y'

$$= - \frac{(A-B)}{2} \text{ when } J=1$$

$$= + \frac{(A-B)}{2} \text{ when } J=M$$

Journel (1974) states that such 'Y' figures have a semi-variogram of the form :-

$$C * \left[3 * \frac{H}{A} - 2 * \frac{H^3}{A^3} \right]$$

He also shows that if the desired three dimensional semi-variogram is 'S(H)' then the required one dimensional semi-variogram equals :-

$$= \frac{d}{dH} H * S(H)$$

For a spherical model this becomes :-

$$\begin{aligned}
 &= \frac{d}{dH} H * C * [1.5 * \frac{H}{A} - 0.5 * \frac{H^3}{A^3}] * E \\
 &= C * [3 * \frac{H}{A} - 2 * \frac{H^3}{A^3}]
 \end{aligned}$$

So the required one dimensional semi-variogram has been produced (see figure 2). It can be shown that the sill of these 'Y' values is :-

$$= WF^2 * \frac{A * (A^2 + 11 * B^2)}{(12 * B)}$$

Therefore to produce a sill of 1.0 the weighting factor 'WF' is set to :-

$$= \sqrt{\frac{(12 * B)}{A * (A^2 + 11 * B^2)}}$$

Thus the 'Y' values exhibit the correct semi-variogram, and have a Normal distribution of $N(0,1)$.

A.2.2. Determination of 'B' :-

It will have been noted that 'B' has not yet been specified. As usual a balance has to be reached between accuracy and cost. If 'B' is very small the simulations will be highly accurate, but expensive to produce.

The spacing of the 'Y' values does not have to be equal to 'B', in other words that of the 'T' figures. It is a simple matter to make the 'Y' values' spacing a whole number multiple of 'B'.

A one dimensional simulation was made of a line of one thousand 'Y' values. The 'Y' values were regularly spaced at an interval of 1 metre. The range of influence of the

semi-variogram was set at 25 metres. The spacing of the independent 'T' figures was 1 metre. In all, twenty five simulations were made under these conditions.

The accuracy of each simulation semi-variogram was estimated by finding its closeness to the one dimensional semi-variogram model. This was represented by the root mean square of the residuals (henceforth called the RMS). This was defined as :-

$$= \sqrt{\frac{1}{N} \sum_{h=1}^{h=N} (O_h - E_h)^2}$$

where N = number of points of the semi-variogram considered

O_h = observed semi-variogram value at lag 'h'

E_h = expected or model semi-variogram value at lag 'h'

The average RMS term for all twenty five simulations was found to be 0.23.

Similarly twenty five simulations were produced using 'T' figures spaced at 0.5, 0.25, 0.125, and 0.0625 metres. The average RMS terms for these values of 'B' were 0.18, 0.15, 0.15, and 0.14 respectively. The law of diminishing returns applies and the results with a 'B' value of 'A/100' seem acceptable. Clark and White (1976) agree with this 'B' value of one hundredth of the range.

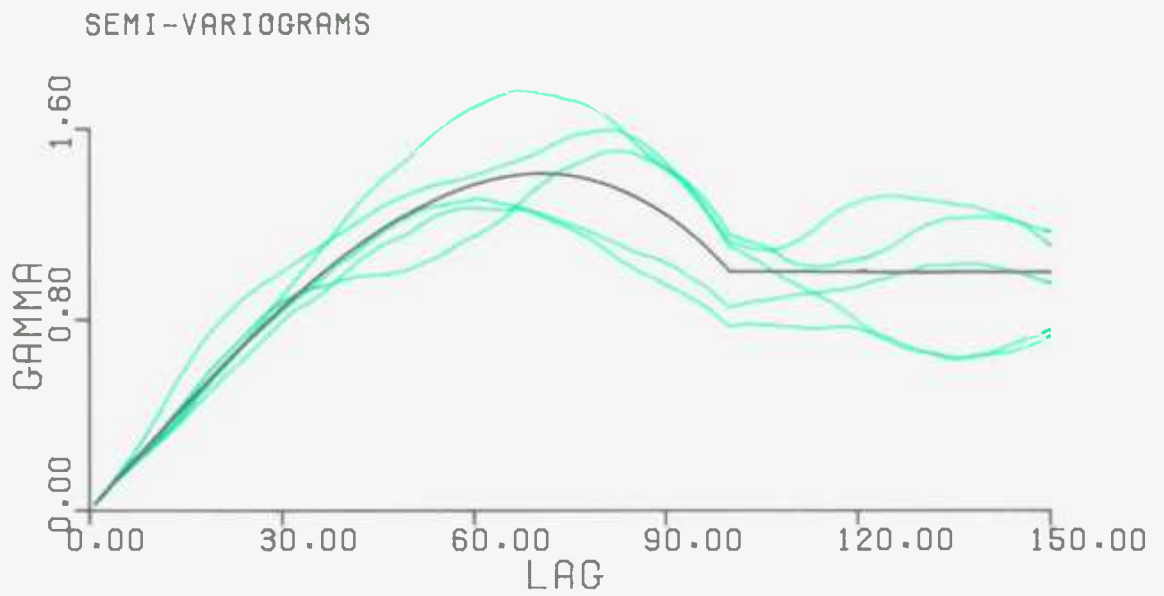
Throughout this study the one dimensional simulated values have been produced at a regular interval of one hundredth of the range. This may not prove acceptable under

different conditions. For instance, with a range of 200.0 metres the value for 'A/100' is 2.0 metres. If the three dimensional simulation requires points 1.0 metre apart, this will be inadequate, and 'B' will have to be reduced.

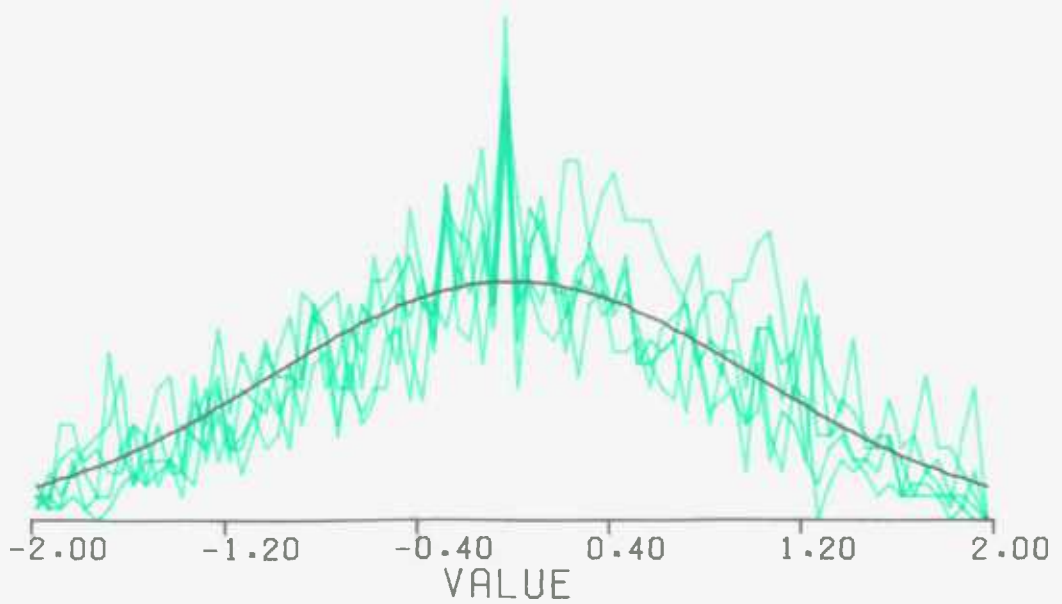
A.2.3. Computer subroutine SIM :-

The one dimensional simulation process has been incorporated in a computer subroutine SIM (see Appendix 1). Using this subroutine five one dimensional simulations were produced. Figure 4 shows the resulting semi-variograms and distribution frequencies. Each simulation was of a line of one thousand points, spaced at regular intervals of 1 metre. The two models, of auto-correlation and distribution, are also displayed.

FIGURE 4 : ONE DIMENSIONAL SIMULATIONS



DISTRIBUTIONS



A.3.COMPOSITION OF THE ONE DIMENSIONAL SIMULATIONS INTO THREE DIMENSIONS

A.3.1.The composition technique :-

Having produced the simulations along one dimensional lines, the next step is to transfer their features into three dimensions.

Consider one of these lines as an axis in three dimensional space. Points at regular intervals ('B') along the axis have each been assigned a simulated value ('Y'). Each point has a co-ordinate on the axis, and all points in space with the same co-ordinate form a plane at right angles to the axis. All the points on this plane are assigned the same simulated value, that is the value of the point where the plane cuts the axis.

Repetition of this process for all simulated values along the axis produces a family of planes. The planes are parallel and spaced at regular intervals of 'B'. Each plane has a simulated 'Y' value associated with it. If the axis is considered to be infinitely long, its family of planes covers all of space.

Each two dimensional plane is expanded, in the third dimension, to a thickness of 'B'. The family now consists of bands (hence the term 'turning bands'), rather than planes. Each band has assigned to it the 'Y' value where its central plane cuts the axis. The family of bands occupies all of three dimensional space.

So, any point in space can now be given a simulated value, from the band in which it lies. If several, say 'N', axes are considered, then any point in space has 'N' simulated values assigned to it. Summing these 'N' values produces the desired three dimensional simulation value.

As has been stated before, the simulated one dimensional values, along the axes, follow a semi-variogram of :-

$$1.0 * \left[3 \frac{H}{A} - 2 \frac{H^3}{A^3} \right]$$

and a Normal distribution of $N(0,1)$.

The summation of such 'Y' values from 'N' axes gives simulated three dimensional 'Z' values which are :-

$$\sum_{i=1}^{i=N} Y_i$$

where Y_i = one dimensional value attributed to
'turning band' from axis number 'i'

The 'Z' values have a semi-variogram of :-

$$N * \left[1.5 \frac{H}{A} - 0.5 \frac{H^3}{A^3} \right]$$

and a Normal distribution of $N(0,N)$.

The variance, and therefore also the sill, of these 'Z' values is now altered. It is changed by multiplication of each value by the term :-

$$\frac{C_0}{N}$$

where C_0 = difference between the sill and
the nugget effect
= $C - E$

The variance and sill of the simulated values thus become
' C_0 '.

The nugget effect is a purely random factor which increases the variance but does not affect the average. So it can be treated as having a Normal distribution of $N(0, E)$. A different random number from such a distribution is added to each of the simulated 'Z' values. This changes them to exhibit a semi-variogram of :-

$$C_0 * [1.5 * \frac{H}{A} - 0.5 * \frac{H^3}{A^3}] + E$$

and a Normal distribution of $N(0, C_0 + E) = N(0, C)$

The desired mean 'M' is now added to all the 'Z' values. This has no effect on the semi-variogram but alters the distribution to $N(M, C)$.

A.3.2. Number and orientation of axes :-

Obviously in an ideal world the number of axes, 'N', used in the summation to produce the three dimensional simulation, would be infinity. So, the question provoked is how many axes will be a sufficiently accurate representation of infinity? Journel (1974) states that fifteen axes, regularly positioned in space, will suffice.

The orientation of each of the fifteen regular axes is

derived from an icosahedron. This regular polyhedron has twenty faces, and thirty edges. The centre of the icosahedron is defined as the origin. The fifteen lines joining the mid-points of opposite edges, all passing through the origin, give the orientation of the fifteen axes.

Journal (1974) determined a three by three matrix, 'R', which can be used to derive the co-ordinate of a point on each axis. The point's position is defined by the co-ordinates on three orthogonal axes. Multiplying these co-ordinates by 'R' produces three co-ordinates of another set of orthogonal axes. In turn, these co-ordinates, when multiplied by 'R', give another set of three orthogonal co-ordinates. Two more multiplications complete the required fifteen co-ordinates.

If the final set of three co-ordinates is multiplied by 'R', the initial three co-ordinates are achieved. This acts as evidence of the regular spacing of the axes.

A subroutine (COORDS1) has been produced which performs the task of deriving the fifteen co-ordinates (see Appendix 2).

A.3.3. Unconditional simulation example :-

A computer program (SIM3D) has been written (see Appendix 3), which produces a simulation of a cuboid of any shape. The required input to the program is :-

- (a) The spherical semi-variogram parameters,

that is the range, sill, and nugget effect.

(b) The Normal distribution average. The variance since it is equal to the sill, has already been specified.

(c) The cuboid dimensions.

The program produces a simulation consisting of a value at every three dimensional grid point. These are at intervals of one unit length apart. Also generated are a histogram of all the simulated values, and the semi-variogram of the values in the three principal directions.

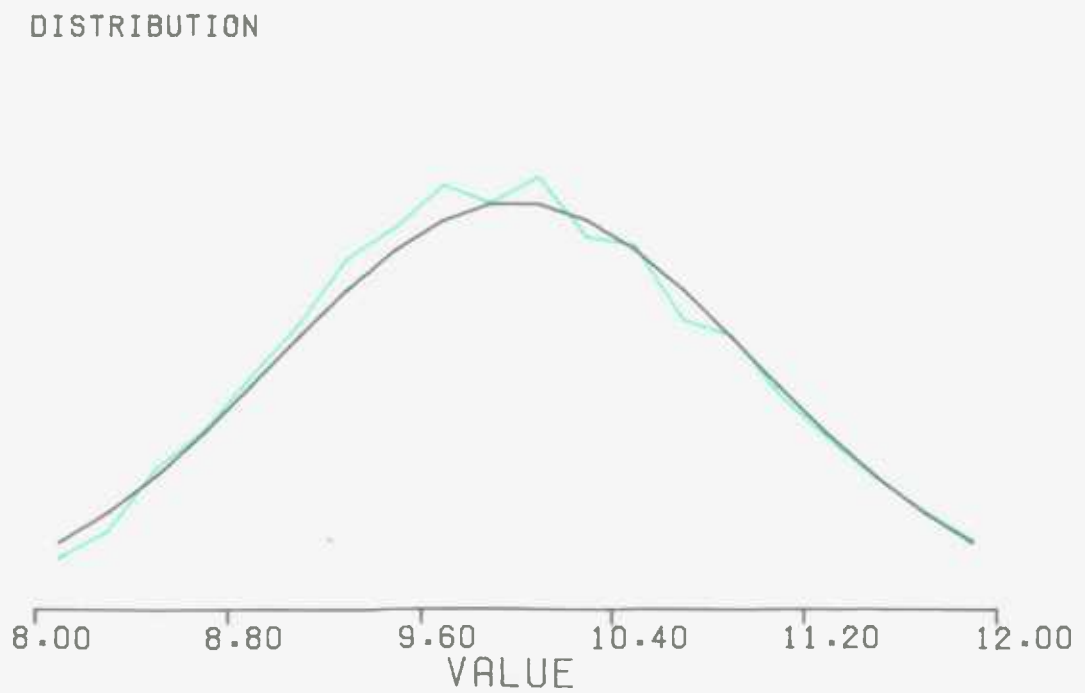
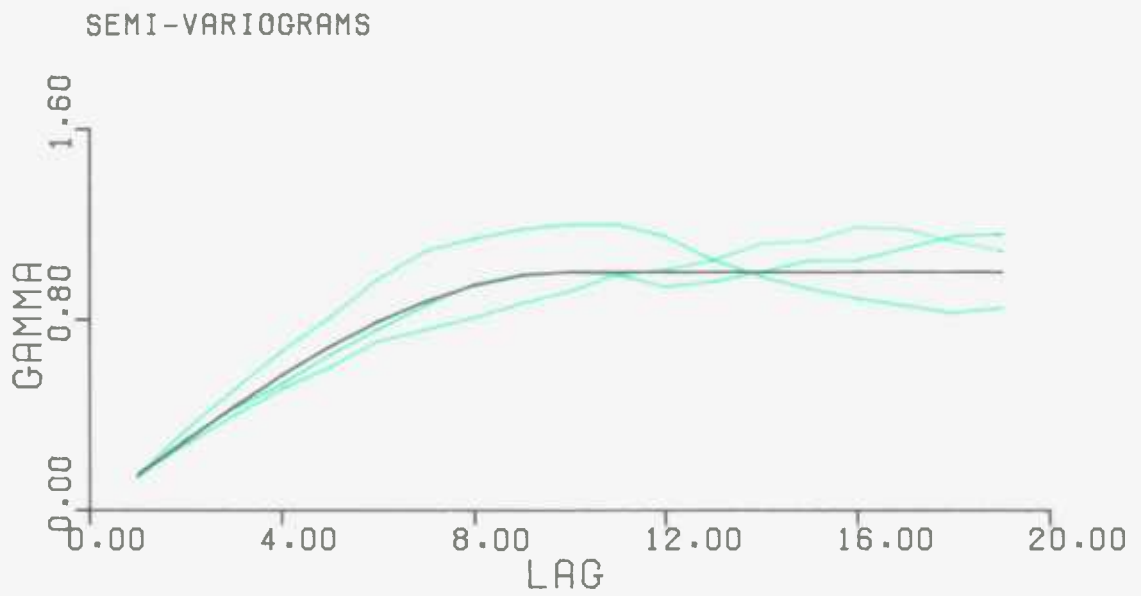
An example of results from the program is given in figure 5. The simulation was of a $20 \times 20 \times 20$ block with the values originating from a semi-variogram of Spherical(10,1,0), and with a Normal distribution of $N(10,1)$. Even with eight thousand simulated figures the deviations from the models are clearly visible.

So, unconditional simulations can now be produced. They exhibit the desired semi-variogram and distribution features.

A.3.4. The uses of unconditional simulations :-

Obviously the initial step in any simulation approach is to determine the models to be used. To do this there has to be some known data. Since data exists, it seems sensible to use it to improve the simulations by making them conditional. In other words, there are few occasions when

FIGURE 5 : THREE DIMENSIONAL
UNCONDITIONAL SIMULATION



only an unconditional simulation can be produced. It is usually possible, and desirable, to proceed to the greater applicability of a conditional simulation.

Unconditional simulations have limited applications. They may be of use for global simulations, early in the history of an orebody. Little or no sampling will have been carried out in the new orebody, and the simulation models will be taken from geologically analagous situations elsewhere.

Similarly, unconditional simulations can be useful for examining the behaviour of an unexplored region within an orebody. In this case, the models would be obtained from the well sampled areas of the orebody.

A.4.CONDITIONING OF THE SIMULATIONS

A.4.1.The conditioning technique :-

The process of conditioning the simulations is a fairly simple one. The object is to make the simulated values equal the actual values at any data points.

The result of an unconditional simulation is a set of values which follow a certain semi-variogram and distribution. A simulated value is known at every grid point of the block under consideration. A sample will have been taken at some grid points, and so the real value is known there. With these data points a kriged estimate of the real value at any grid point can be calculated. Using the simulated values at the same positions (and therefore the same kriging weights) a simulated value estimate may be found. So, for all grid points there are three known figures :-

- (1) 'S' - value from unconditional simulation.
- (2) 'Rk' - kriged estimate from real values at data points.
- (3) 'Sk' - kriged estimate from simulated values at data points.

To make the simulation conditional all the simulated values are replaced by the term :-

$$S - S_k + R_k$$

If the grid point considered is also a data point then the following is true :-

$$S_k = S$$

$$R_k = R$$

and the conditional simulation value becomes :-

$$= S_k - S_k + R$$

$$= R$$

Thus the values of the conditioned simulation exactly agree with the known sampled values at all the data points.

Since kriging is an unbiased estimation technique, the average of the kriged estimates equals that of the samples used in the kriging systems. Therefore the mean of the 'Sk' values is the same as that of the 'S' values. Similarly for the 'Rk' and 'R' figures. The average of the conditioned simulation which :-

$$= \text{average 'S'} - \text{average 'Sk'} + \text{average 'Rk'}$$

thus becomes :-

$$= \text{average 'R'}$$

So, the conditional simulation has the same average as that of the data points.

It can be shown that the process of conditioning a simulation does not affect the semi-variogram of the values (Journel, 1974). They will still follow the model specified for the unconditional simulation.

So, a conditional simulation has the following important properties :-

- (a) It coincides with the real values at data points.
- (b) It has an average equal to the average of the real values.
- (c) It has a semi-variogram of the desired form.

The conditioning process can be summarised as involving the transfer of the term :-

$$(S - S_k)$$

A possible difference between a value and its kriged estimate is simulated. It is then added to the kriged estimate of the real value.

It can be seen that of the two parameters defining the distribution, only the variance survives the conditioning process. Whatever the average of an unconditional simulation, it will become that of the data points after conditioning. The distribution variance is calculated using a standard statistical formula, which assumes that all the values are independent. This is not true for pairs of points less than the range of influence ('A') apart. Because of this the sill is generally a better estimate of the true variance of the data. Therefore the only reason for finding the distribution frequency of the data is to check that it is a Normal distribution.

A.4.2. The importance of the number of data points :-

Five simulations were made of a line of one thousand

points, regularly spaced at 1 metre intervals. The values came from a semi-variogram of Spherical(50,20,0) and a Normal distribution of N(100,20). The semi-variograms and distribution frequencies of each simulation are shown in figure 6.

The simulations were conditioned to data points regularly spaced along the line. The data points themselves originated from a similar unconditional simulation, and had an average of 101.34 and a variance of 17.40. Simple alteration of all the one thousand data values to become :-

$$[(S-101.34)*\frac{4.47}{4.17}] + 100.00$$

changed their average and variance to 100.00 and 20.00 respectively. The simulation then fitted the models closely (see figure 6).

The conditioning of the simulations was carried out by a computer program, CON (see Appendix 4). CON reads in five unconditional simulations and one data simulation, from previously formed computer storage files. It produces a kriging system for all points of the unconditional simulation between the third and antepenultimate data points. The weights are then calculated for the four nearest data points. If any weight is less than -0.003 or greater than 0.6, the kriging system is altered and new weights calculated. In the former case the point with the negative weighting is eliminated from the kriging system. With a 0.6 or larger weight, an extra point beyond it is

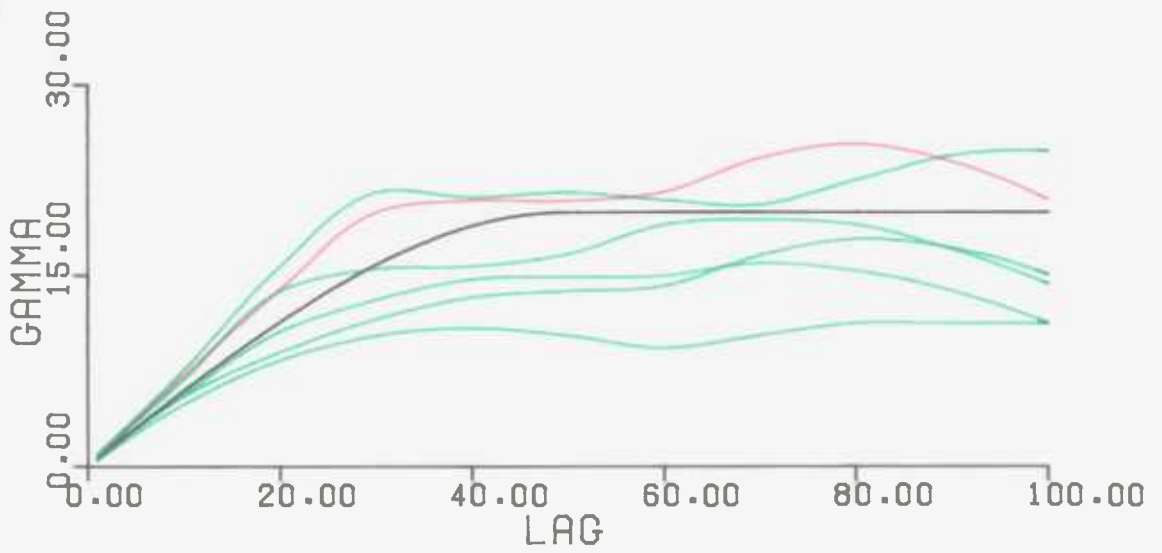
FIGURE 6 : SIMULATIONS BEFORE CONDITIONING

MODELS

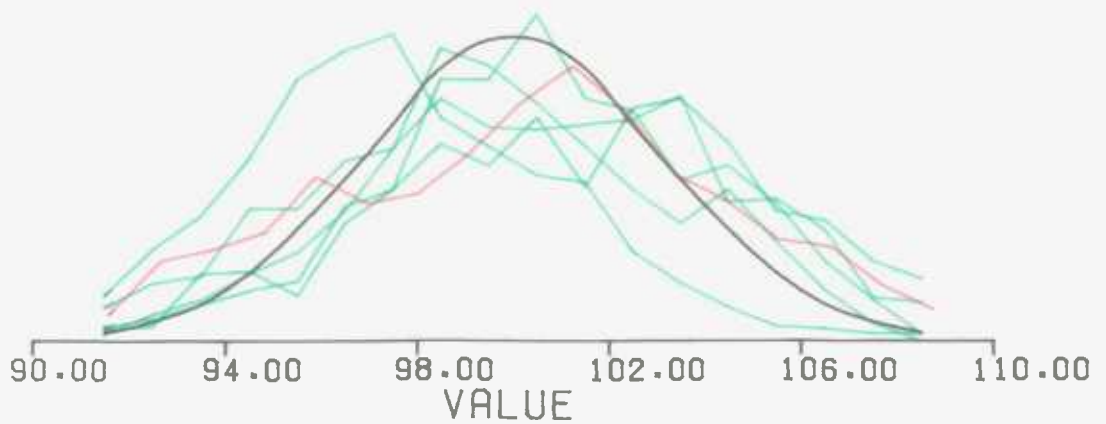
DATA SIMULATION

UNCONDITIONAL SIMULATIONS

SEMI-VARIOGRAMS



DISTRIBUTIONS



added to the system. The output of the program includes the semi-variogram and histogram of the conditioned values.

The number of data points to which the five simulations were conditioned was varied between :-

(a) Fifty points at regular intervals of 20 metres. With this arrangement of conditioning points a kriging system was found for every point, between the third and forty eighth data points. Using these kriging systems the average kriging estimation variance of the kriged estimates was 4.07, or 20.4 percent of the sill. The semi-variograms and histograms of the simulations after being conditioned are shown in figure 7.

(b) One hundred points at regular intervals of 10 metres. The average estimation variance of the kriged estimates was 10.0 percent of the sill. The five semi-variograms and histograms after conditioning are seen in figure 8.

(c) Two hundred points at regular intervals of 5 metres. The average kriging variance was 6.0 percent. The results of this data pattern are shown in figure 9.

As expected, with a greater number of data points the conditional simulations are closer in behaviour to the data values. The more data points used, the better are the conditional simulations. So, an effort should always be made to include as many data points as possible in the

FIGURE 7 : SIMULATIONS AFTER CONDITIONING
TO DATA POINTS AT 20 METRES INTERVALS

50 DATA POINTS

SIMULATIONS

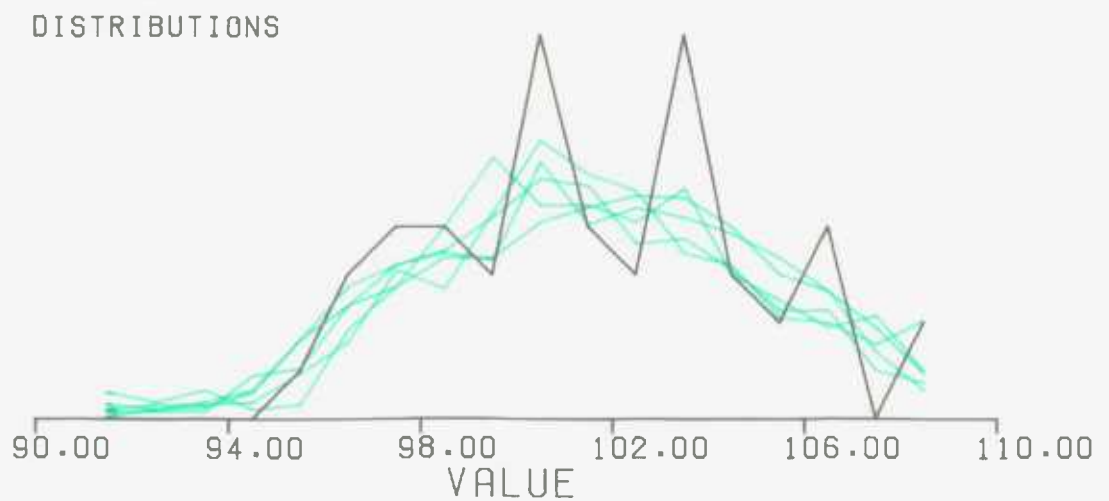
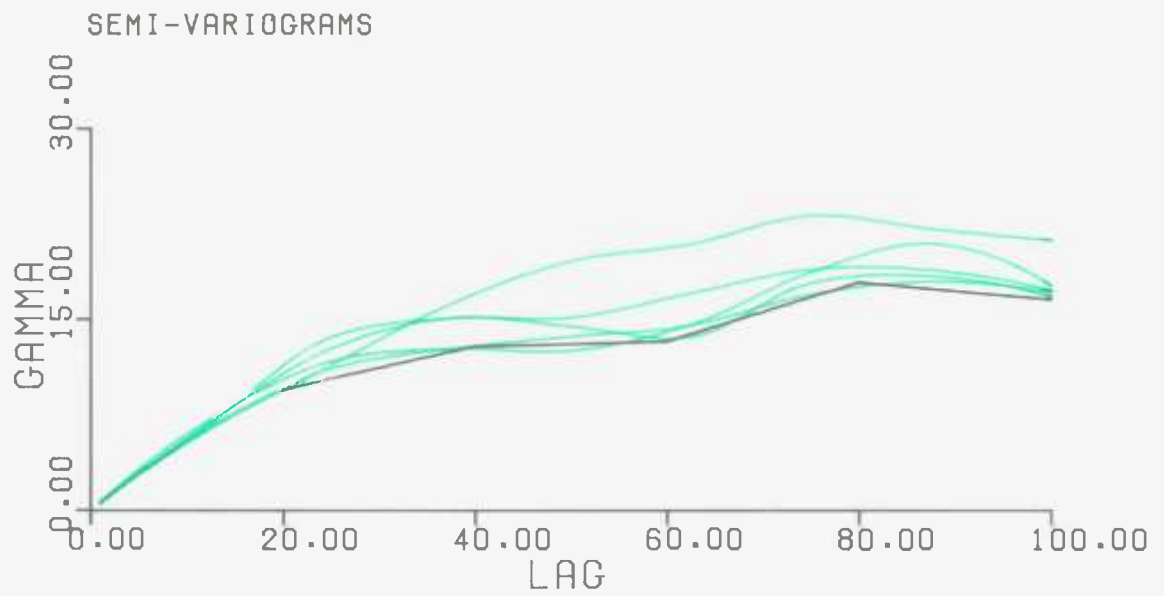
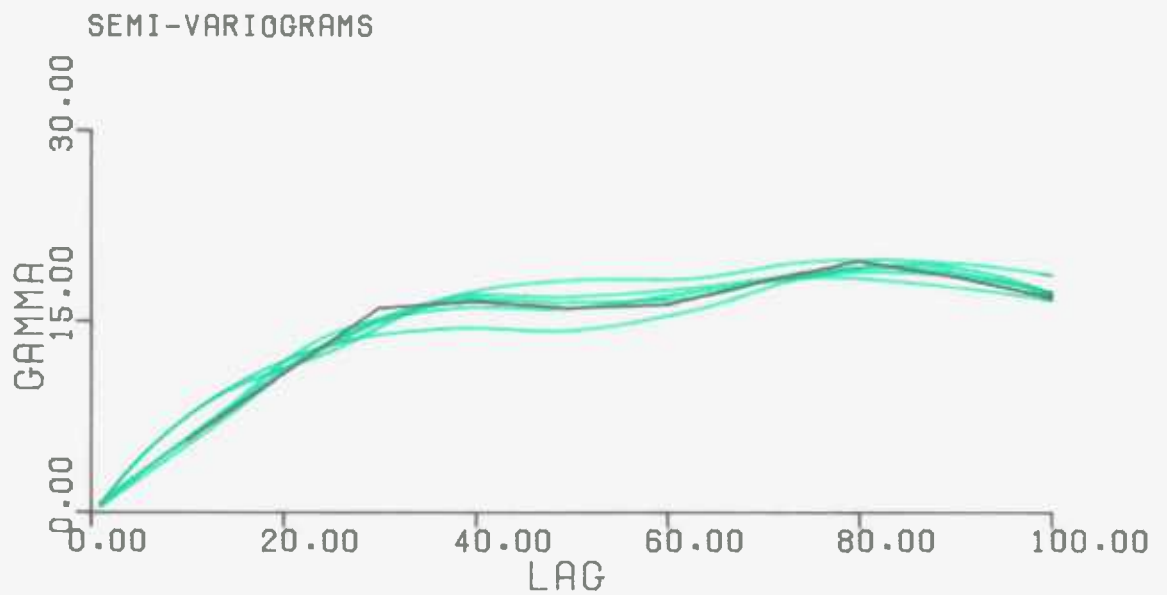


FIGURE 8 : SIMULATIONS AFTER CONDITIONING
TO DATA POINTS AT 10 METRES INTERVALS

100 DATA POINTS

SIMULATIONS



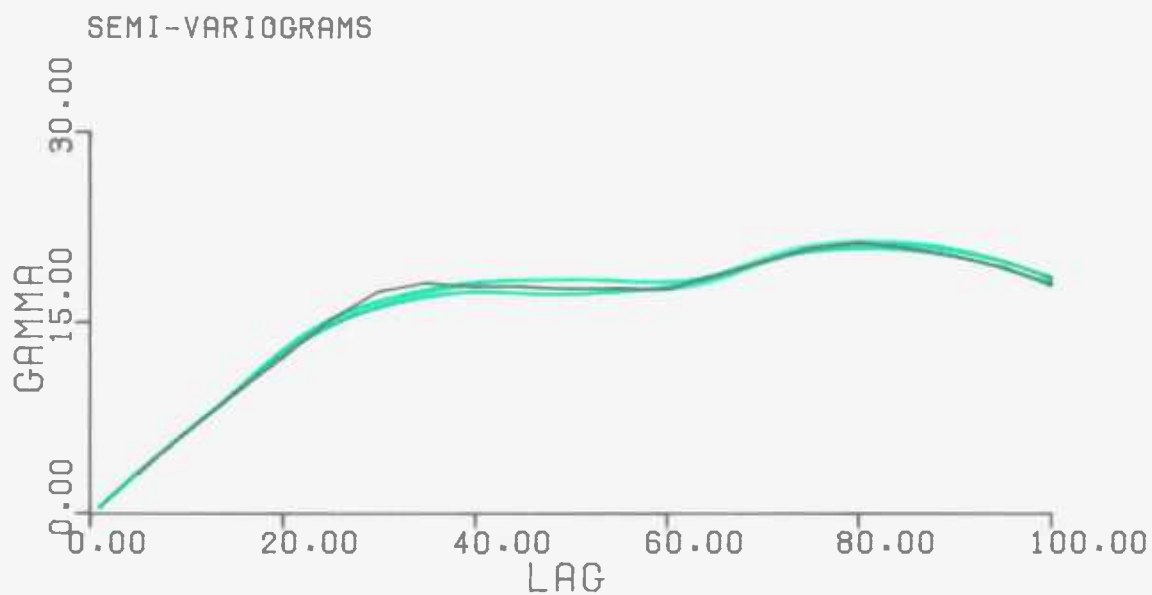
DISTRIBUTIONS



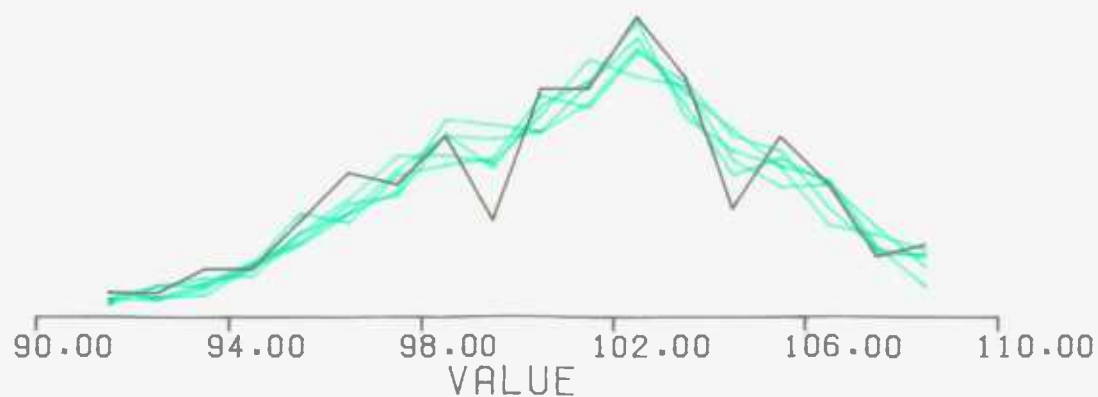
FIGURE 9 : SIMULATIONS AFTER CONDITIONING
TO DATA POINTS AT 5 METRES INTERVALS

200 DATA POINTS

SIMULATIONS



DISTRIBUTIONS



conditioning process.

A.4.3. The importance of the position of the data points :-

The behaviour of the conditional simulations is not due entirely to the number of data points. The arrangement of the conditioning data points is also important. This is shown by conditioning the five simulations to one hundred data points in the following patterns :-

(a) Twenty points at 20 metres intervals, next to forty points at 5 metres intervals, next to forty points at 10 metres intervals. With this arrangement the average kriging variance was 10.5 percent. The results are shown in figure 10.

(b) Forty points at 10 metres, forty points at 5 metres, and twenty points at 20 metres. Obviously this is merely the last pattern reversed. Figure 11 shows the results.

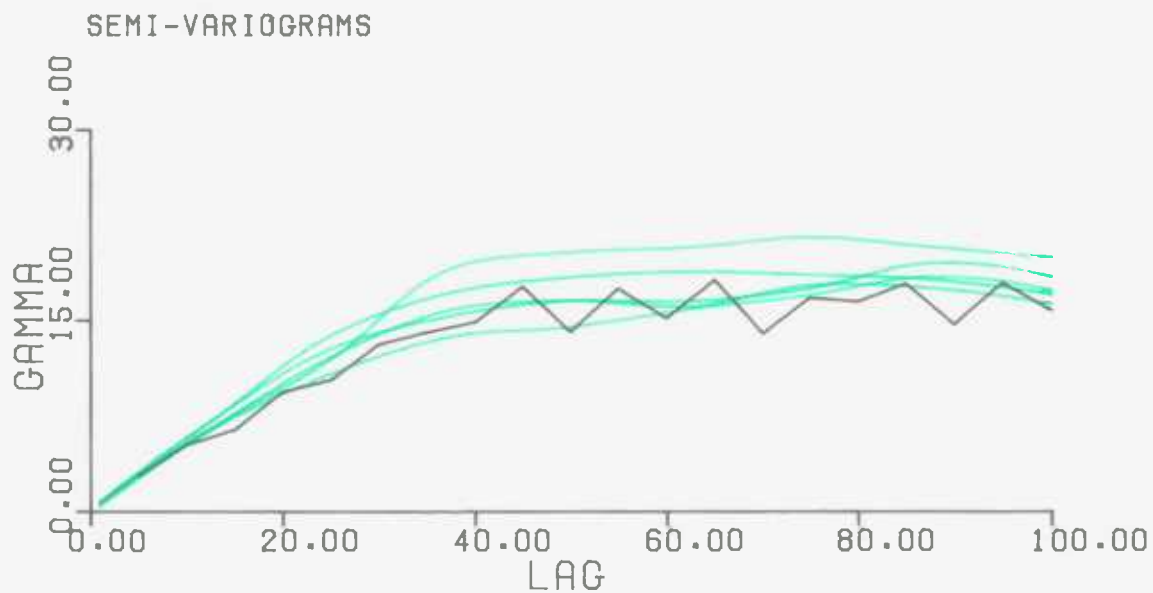
Comparison of the results using these two data point arrangements, and those with one hundred uniformly spaced points (see figure 8), shows the importance of the conditioning data points. The arrangement of the data points, and their actual values, have a large influence on the behaviour of the conditional simulation.

The importance of the conditioning data points arrangement is further emphasized by the semi-variograms shown in figure 12. Two similar (that is using the same models) unconditional simulations of a line of two hundred

FIGURE 10 : SIMULATIONS AFTER CONDITIONING
TO DATA POINTS AT 20,5,10 METRES INTERVALS

100 DATA POINTS

SIMULATIONS



DISTRIBUTIONS

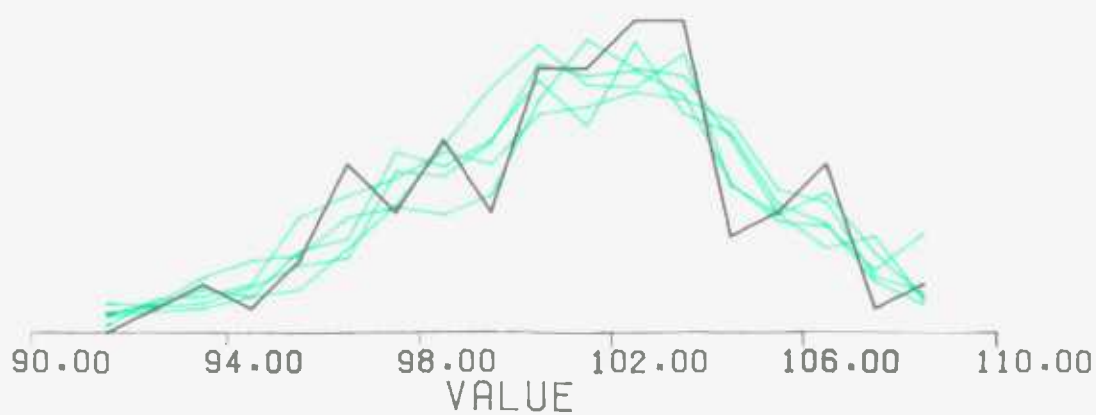


FIGURE 11 : SIMULATIONS AFTER CONDITIONING
TO DATA POINTS AT 10,5,20 METRES INTERVALS

100 DATA POINTS

SIMULATIONS

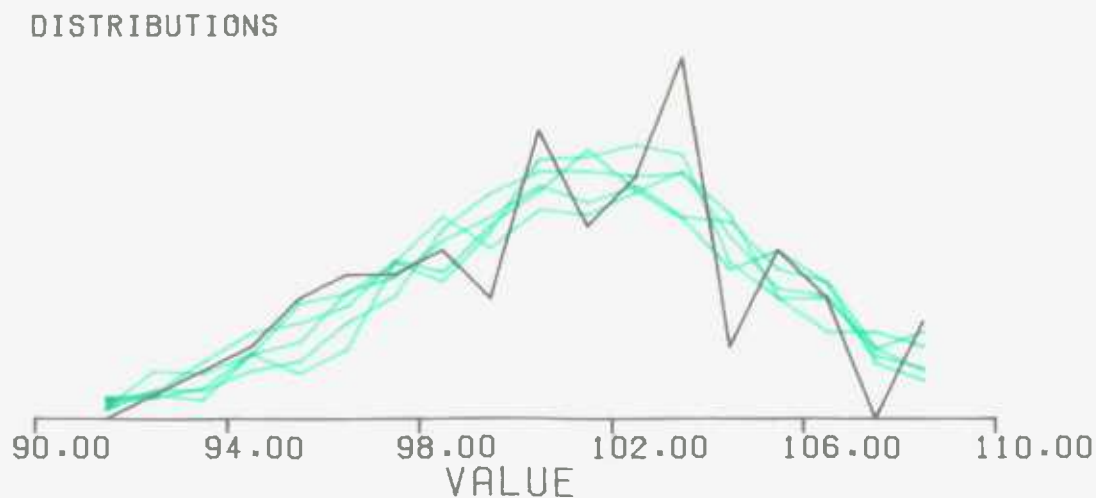
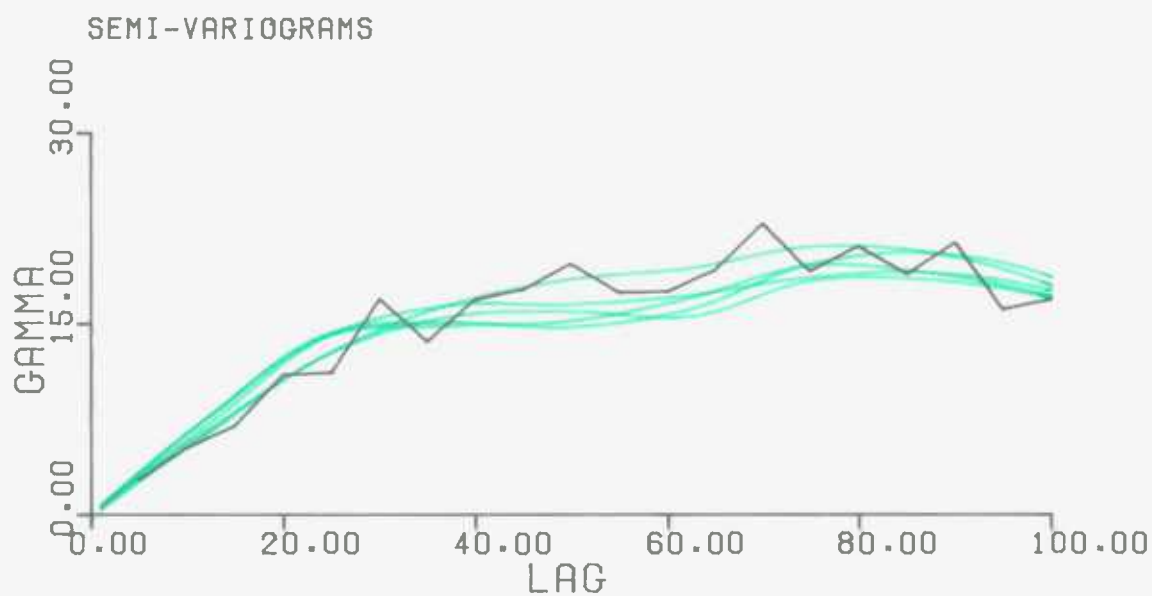
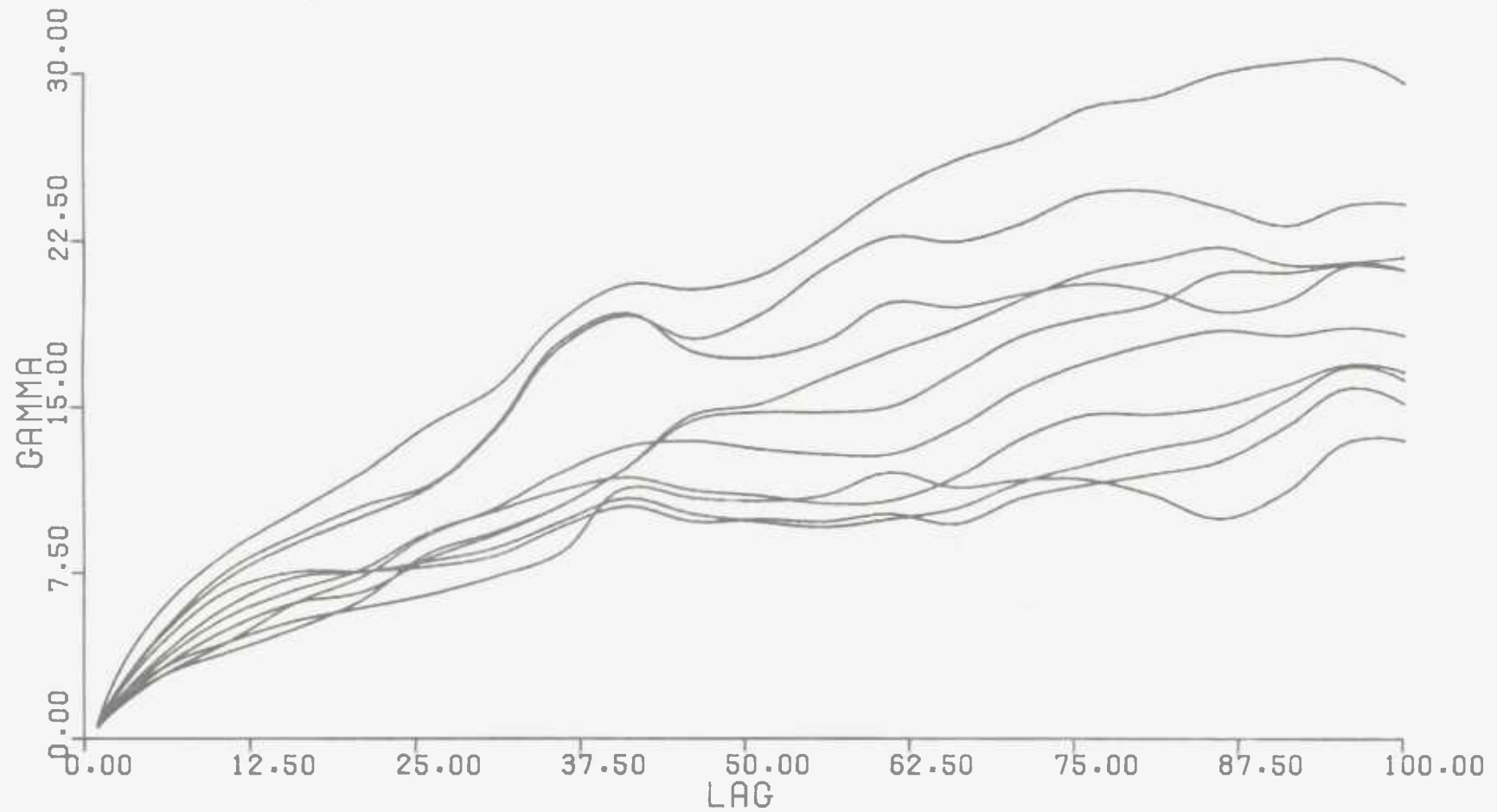


FIGURE 12 : DIFFERENT CONDITIONING DATA POINTS

CONDITIONING POINTS EVERY 10 METRES



points were produced. The points were regularly spaced at 1 metre intervals. Using an adaptation of the program CON, one simulation was conditioned to data values from the other simulation. The data points were regularly spaced at intervals of 10 metres. The differences between the semi-variograms shown are due to the different positions of the data points. The first conditional simulation had its first conditioning point at position number 10, its second at number 20, and so on. The second simulation started at number 11, and again continued in steps of 10 metres. Similarly, eight other conditional simulations were produced with the data points moved one position each time. Thus the tenth simulation had its first conditioning point at position number 19.

A.4.4. Merits of conditioning :-

The number, and position, of the conditioning data points have a great effect on the conditional simulation values. The conditioning process is the most influential step in the production of a conditional simulation. It forces the simulations to agree with any known actual values. This is particularly of use if a small area within a much larger one is being simulated, since any local small scale idiosyncrasies will be coped with by the conditioned simulation. This is the great advantage of conditional 'turning bands' simulations over other, unconditional simulation techniques.

Whilst conditioning is the great strength of the 'turning bands' approach, it is also the great weakness. This is because it attributes one hundred percent accuracy to the data. The conditional simulation must pass exactly through the data values without allowance for small sampling inaccuracies. As has been shown, the values at the data points have a great influence on the behaviour of the conditional simulation. So, if any of the sampled data values is incorrect its error may have large repercussions.

A.4.5. Simulations before and after conditioning :-

If no data points are used in the conditioning process, the unconditional simulations exactly equal the conditional simulations. So, the simulation which is most accurate (with respect to the models) before conditioning is also the most accurate after the conditioning. The correlation between the accuracy of the simulations before and after the conditioning operation is total.

Similarly, at the other end of the scale, if an infinite number of data points are used, all the conditioned simulations are the same. The unconditional simulations are totally irrelevant. The correlation between the accuracy of the simulations before and after conditioning is zero.

It was decided to find at what number of data points the correlation, between a simulation's accuracy before and after conditioning, becomes significant.

A parameter was needed to act as a measure of the proximity of a simulation to the models. One of the models, the distribution, is defined by the average and the variance. As has been stated, the average of an unconditional simulation has no effect on the average of the conditioned values. So, the accuracy of the unconditional simulation average before conditioning is irrelevant. The variance of the distribution also finds expression in the semi-variogram. Therefore, a measure of the closeness of the semi-variogram to the model also checks the one important parameter of the histogram.

The deviation of a semi-variogram from the model was expressed by the mean square of the residuals (henceforth called the MS). This was defined as :-

$$= \frac{1}{N} * \sum_{h=1}^{h=N} (O_h - E_h)^2$$

where N = number of points of the semi-variogram considered

O_h = observed semi-variogram value at lag 'h'

E_h = expected semi-variogram value at lag 'h'

Twenty five simulations were produced of a line of one thousand points regularly spaced at 1 metre intervals. The values originated from a Normal distribution of $N(100, 20)$, and a semi-variogram of Spherical(50, 20, 0). For each unconditional simulation the semi-variogram was calculated up to a lag of 60 metres. The MS of the semi-variogram was found and also the sill, as estimated by the average

semi-variogram value between lags of 50 and 60 metres. The variance of the unconditioned values was also calculated using a classical statistics' formula. The twenty-five simulations were conditioned (using program CON) to one hundred points from the altered data simulation used previously (see figure 6). The points were regularly spaced at 10 metres intervals. The same three parameters (the MS and sill of the semi-variogram, and the variance) were calculated for the values after they had been conditioned.

The scattergram of the twenty five pairs of sill and variance values before conditioning shows great correlation. This is also true after conditioning, and all fifty pairs are plotted in figure 13. The correlation coefficient is 0.65, which, as expected, is highly significant.

Figure 14 shows the correlation, for both the unconditional and conditional simulation semi-variograms, between the MS term and the deviation of the sill from the model (that is the modulus of the term 'sill-20.00'). The correlation coefficient is 0.84 which is significantly high.

So, the MS of a semi-variogram is proportional to the accuracy of the sill. Therefore, it is also proportional to the accuracy of the variance. It would seem to give a good idea of a simulation's proximity to the two models of distribution and auto-correlation.

The relationship between the MS term for the

FIGURE 13 : SCATTERGRAM OF
VARIANCE VS. SILL

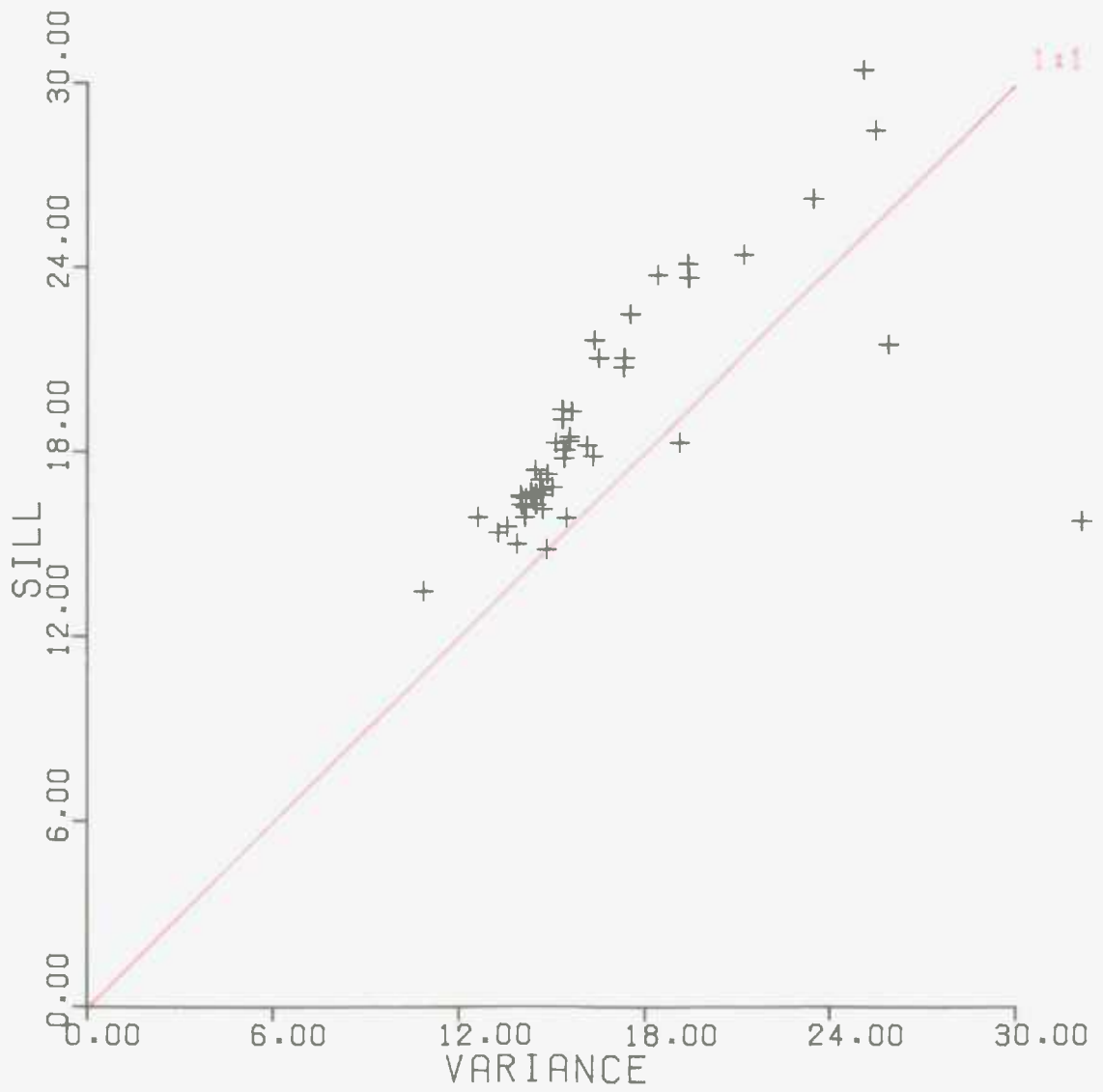
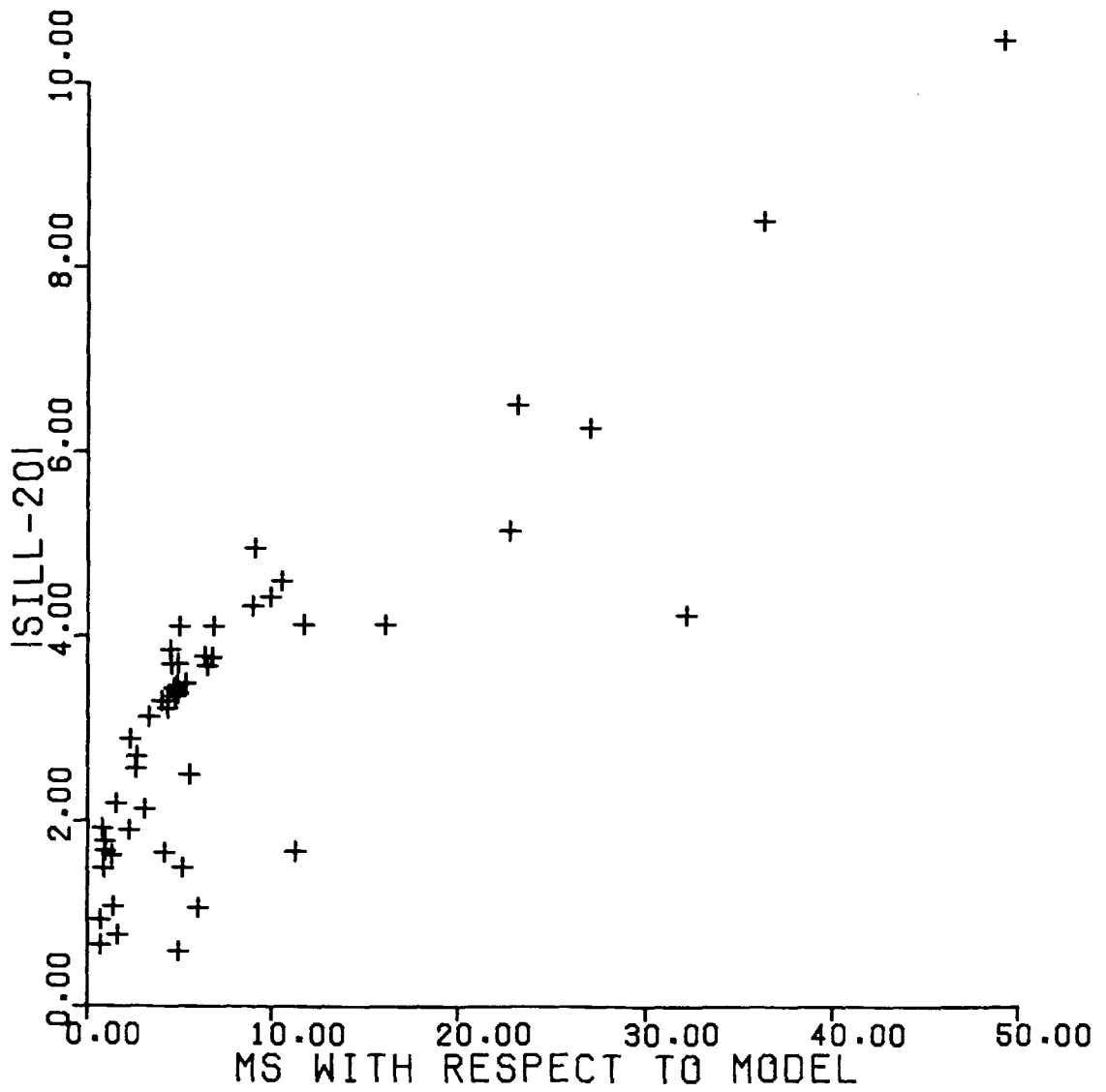


FIGURE 14 : SCATTERGRAM OF
MS WITH RESPECT TO MODEL VS.
ISILL-201



semi-variograms before and after conditioning is shown in figure 15. It should be noted that the expected semi-variogram used in the calculation of MS was not the model semi-variogram. It was the semi-variogram of the data values used in the conditioning operation. This allows for any deviations the selected values may have from the models. The semi-variogram of each simulation, both before and after conditioning, only used the values between the third and antepenultimate data points. To avoid making the program CON too complicated, the positions outside these two limits were not conditioned. The correlation coefficient for figure 15 is -0.20 which is not significantly different from zero. In other words, an inaccurate simulation before conditioning, is not necessarily an inaccurate simulation after conditioning. The one hundred evenly spaced data points are too many for this to be true. The number of data points has to be moved further towards zero for significant correlation to exist.

The twenty five simulations were conditioned to data points regularly spaced at intervals of 20, 30, and 40 metres. The scattergrams between the semi-variogram MS terms before and after conditioning are shown in figures 16, 17, and 18 respectively. The results are summarised in table 1. It shows that the correlation becomes significant with twenty five data points. This is when the average kriging variance used in the conditioning process is 44 percent. When it is 32 percent the correlation is only

FIGURE 15 : SCATTERGRAM OF
MS WITH RESPECT TO DATA BEFORE CONDITIONING VS.
MS WITH RESPECT TO DATA AFTER CONDITIONING
CONDITIONING POINTS AT 10 METRES INTERVALS

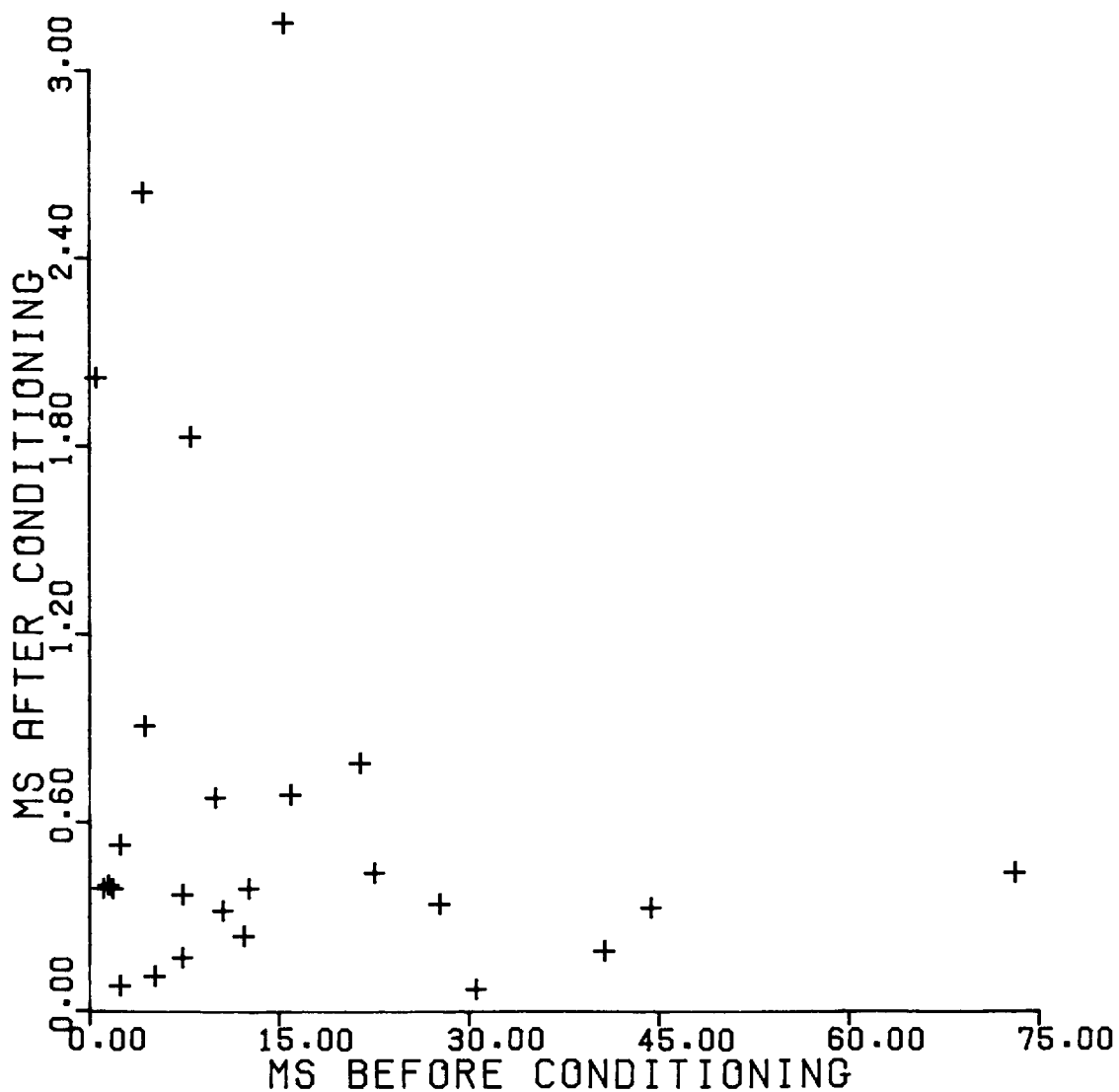


FIGURE 16 : SCATTERGRAM OF

MS WITH RESPECT TO DATA BEFORE CONDITIONING VS.

MS WITH RESPECT TO DATA AFTER CONDITIONING

CONDITIONING POINTS AT 20 METRES INTERVALS

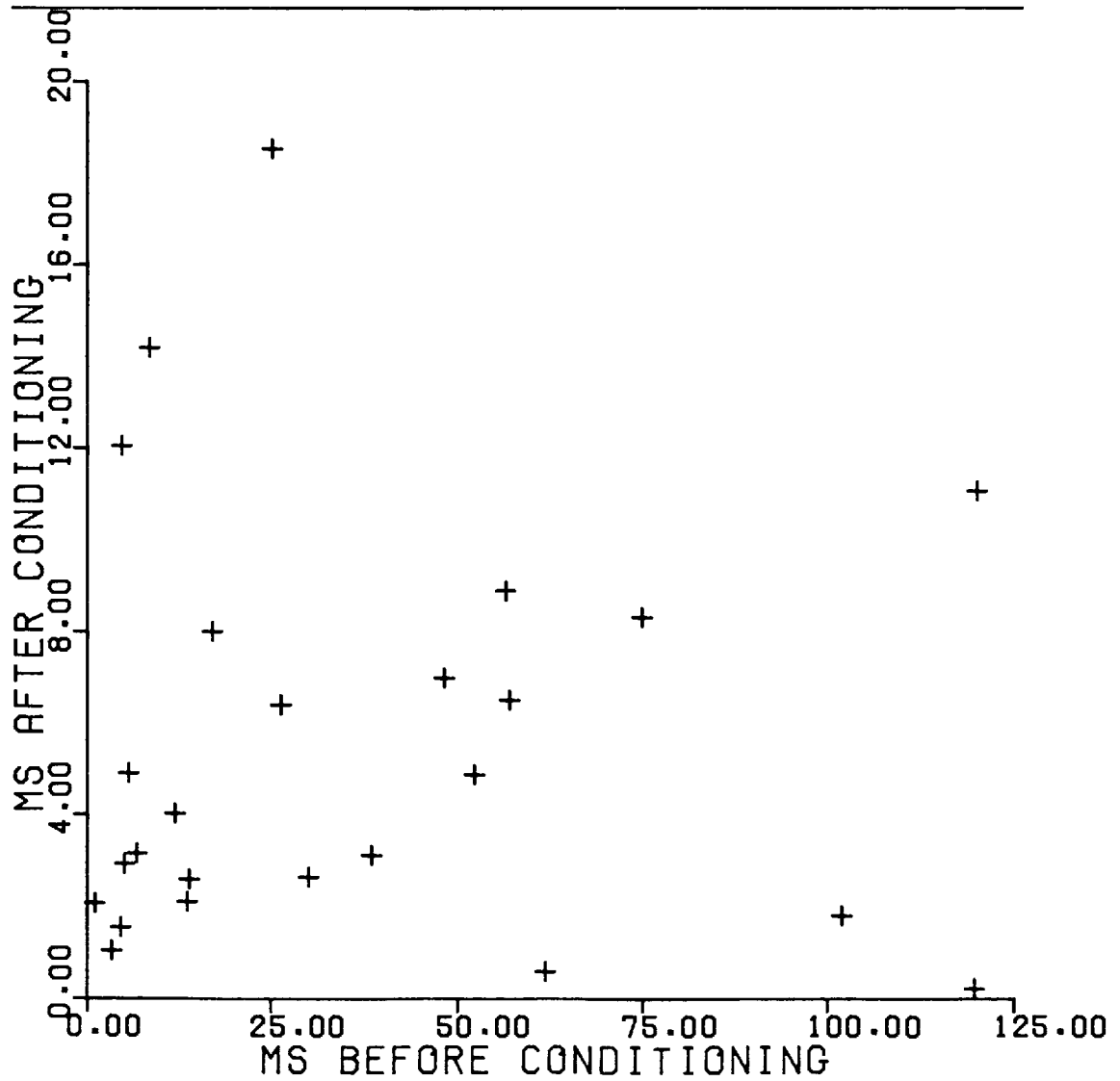


FIGURE 17 : SCATTERGRAM OF
MS WITH RESPECT TO DATA BEFORE CONDITIONING VS.
MS WITH RESPECT TO DATA AFTER CONDITIONING
CONDITIONING POINTS AT 30 METRES INTERVALS

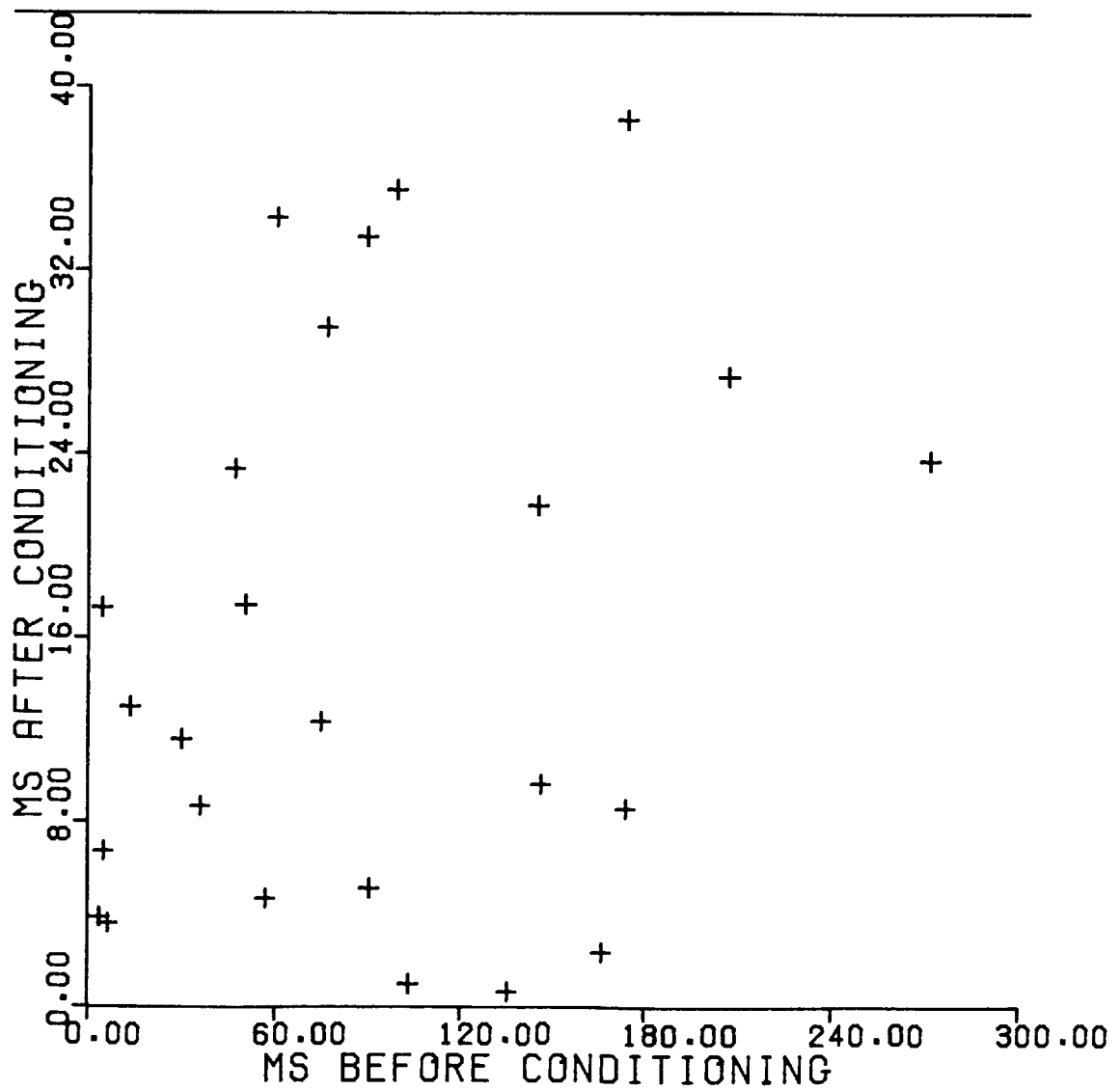


FIGURE 18 : SCATTERGRAM OF
MS WITH RESPECT TO DATA BEFORE CONDITIONING VS.
MS WITH RESPECT TO DATA AFTER CONDITIONING
CONDITIONING POINTS AT 40 METRES INTERVALS

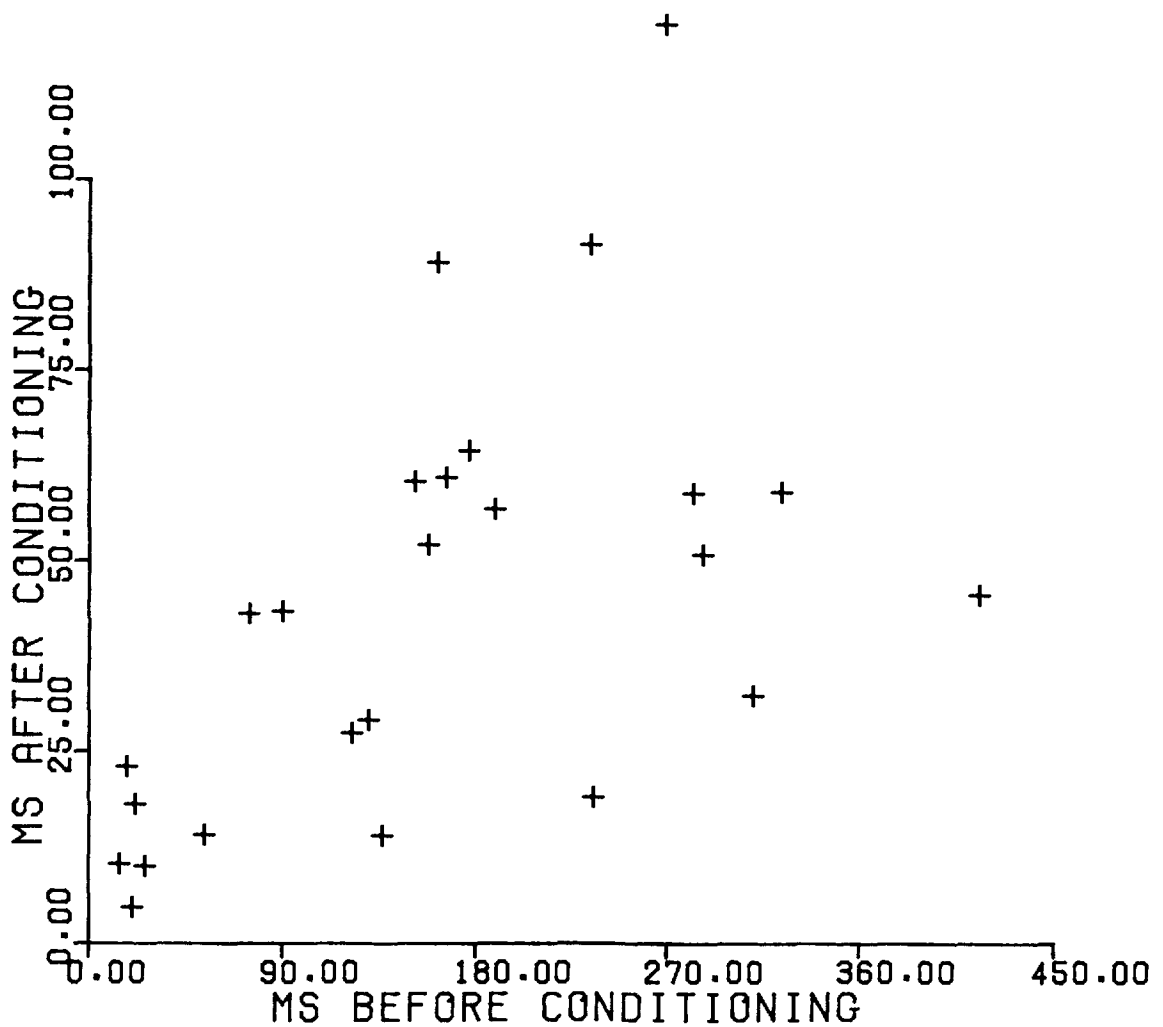


TABLE 1

Conditioning data points		Average kriging variance during conditioning % of total sill	Conditioned simulations	
Number	Spacing		Characteristics	Correlation between MS before and after conditioning
∞	0	0%	local	R= 0.00 T= 0.00
100	10	10%	↓	-0.20 0.96
50	20	21%		-0.01 0.04
33	30	32%		0.25 1.25
25	40	44%		0.54 3.06
0	∞	200%		global

significant at the 25 percent level. So, it seems that the correlation is significant if the average kriging variance is more than one third of the sill.

A.4.6. The cost of conditioning :-

Although the conditioning operation is a relatively straightforward one, it can be time consuming. This is because it requires the derivation of a kriging system for every simulated point. The kriging system used in program CON is a moderately simple one. It involves between two and six points in any one kriging estimation. It could have been made much more complex, but the increase in accuracy of the estimates (as measured by the average kriging variance) would have been insignificant.

The unconditional simulations of one thousand points each used 63.8 CP seconds to be created. To condition all twenty five of them to one hundred regularly spaced data points took another 192.5 CP seconds. With a pattern of twenty five evenly positioned data points the conditioning process took 171.1 CP seconds. An idea of the extra cost of conditioning a simulation is given by these increases in execution time of 12 and 11 percent.

A.5.ALTERNATIVE ONE DIMENSIONAL AXES CONFIGURATIONS

A.5.1.The importance of the axes configuration :-

As has been stated before, the major drawback to three dimensional simulation is the excessive use of computer time and storage space. The 'turning bands' method does, to a large extent, alleviate this problem.

However, the aim should always be to reduce cost to a minimum. The composition of the one dimensional simulations into three dimensions is an operation normally responsible for an explosion of computer cost. Journal (1974) gives no reason for his choice of fifteen regular axes. So, it was decided that the configuration of the one dimensional axes deserved closer scrutiny.

The icosahedron is the regular polyhedron with the largest number of faces. Its thirty edges are used to determine the orientations of the fifteen axes. The same results could be achieved using the thirty edges of the twelve sided dodecahedron. Presumably the configuration of fifteen regular axes was selected because no more complex regular polyhedron exists, from which to deduce the orientations of, for instance, twenty regular axes.

So, the question remains; are fifteen axes too few, or indeed, too many?

A.5.2.Ten regular axes :-

If fifteen axes are considered to be too many, then a smaller number is required. The next largest number of

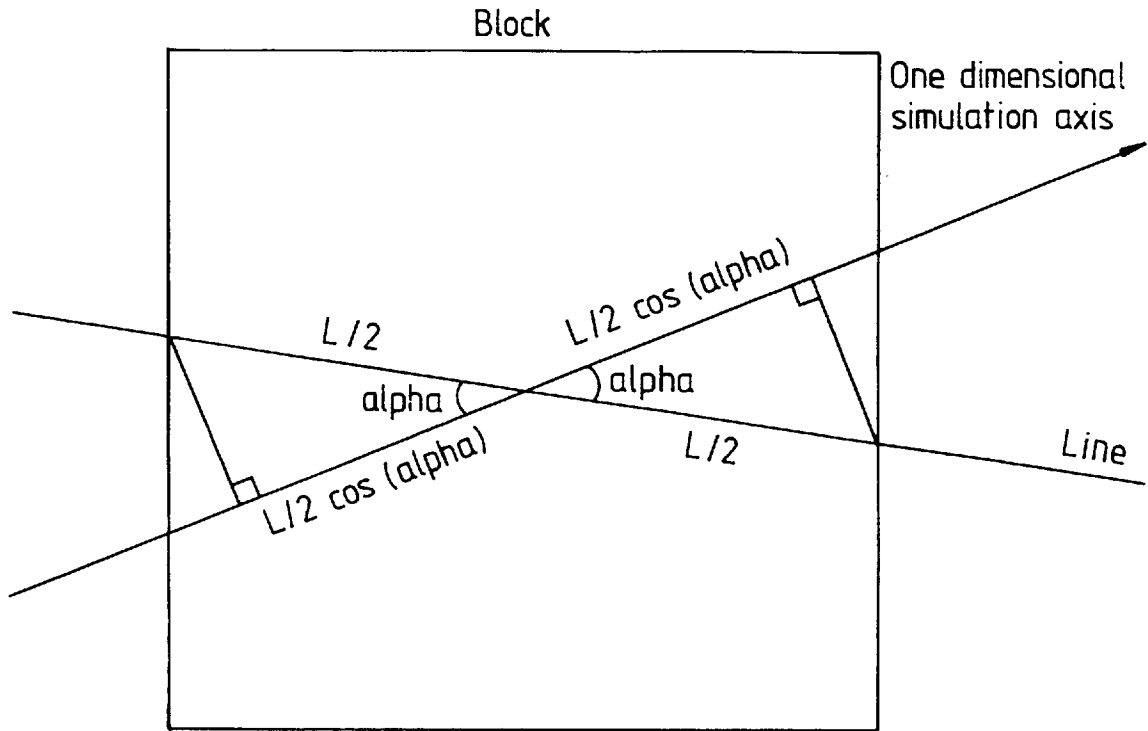
regular axes, possible to obtain with a polyhedron, is ten. The orientations of these axes can be calculated from the twenty vertices of a dodecahedron, or the twenty faces of an icosahedron. Using the geometry of a dodecahedron, the ten co-ordinates of any point in space can be derived (see Appendix 5). The process has been incorporated in a computer subroutine, COORDS2 (see Appendix 6). It differs from that used by Journal for fifteen axes in that no rotation matrix is involved. The three initial position defining co-ordinates are not included in the output of ten co-ordinates. In other words, the three standard orthogonal grid directions are not the same as any of the ten regular axes.

A.5.3. Fifteen regular axes :-

A similar method has been evolved for calculating the co-ordinates with the fifteen regular axes configuration (see Appendix 7). No rotation matrix is involved and the initial three orthogonal co-ordinates are used to find the fifteen extra co-ordinates. It is found that the orientation of the fifteen axes differs from that of Journal. Only one of the standard grid axes is included in the fifteen axes. With Journal's method the orthogonal x, y, and z directions are equal to axes numbers 1, 2, and 3. With the new approach the y direction is the same as axis number 12.

Consider, as shown in figure 19, a line of

FIGURE 19



length 'L', in any direction within a simulated block. The angle between it and one of the one dimensional simulation axes is 'alpha'. The origin is at the centre of the block. Therefore the co-ordinates on the one dimensional axis, of the line as it enters and leaves the block, are ' $-L\cos(\alpha)/2$ ' and ' $+L\cos(\alpha)/2$ '. It can be seen that the number of points, on the one dimensional axis, which contribute to the simulated values on the line, is proportional to ' $\cos(\alpha)$ '. This is true for all the fifteen axes. Therefore, the number of points used to produce the simulated values is proportional to the sum of the fifteen ' $\cos(\alpha)$ ' figures.

It is obvious that the accuracy (that is similarity to the model) of the semi-variogram along the line is in proportion to the number of one dimensional points contributing to it. It is therefore proportional to the sum of the ' $\cos(\alpha)$ ' terms.

With Journel's method the fifteen axes are orientated such that the sum of the ' $\cos(\alpha)$ ' values is 7.47 in each of the x, y, and z directions (see Appendix 8). With the new technique the ' $\cos(\alpha)$ ' figures for the x, y, z directions total 8.67, 7.47, and 6.87 respectively (see Appendix 7).

So, with the new approach there exists anisotropy, with more points contributing in the x direction. The semi-variogram in that direction is probably more accurate than in the other two principal directions. Also, it is

likely to be more accurate than that of the x direction under Journel's axes orientation.

When simulating two or three dimensions this accuracy anisotropy is irrelevant, since there is an infinite number of directions possible. Semi-variograms are usually given for the three principal directions merely for ease of calculation. Other directions with a different sum of 'cos(alpha)' values are just as valid.

However, with one dimensional simulations the anisotropy is important and can be used to advantage. A simulation of a 1000*1*1 block is likely to be more accurate than one of a 1*1000*1 or 1*1*1000 block. It will also be better than a one dimensional simulation in any of the three principal directions with Journel's method. Under his fifteen axes orientation a direction with a sum of 'cos(alpha)' value of 8.67 could be found. However, the simulation of this line would be intimidatingly complex.

The new method of orientating the fifteen axes configuration was incorporated in a computer subroutine, COORDS3 (see Appendix 9). Comparison of computer running time was made with the similar subroutine (COORDS1) representing Journel's rotation matrix method. It was found that for X, Y, Z input co-ordinates of between -5.0 and +5.0, Journel's technique took six percent longer to calculate the fifteen co-ordinates.

It was decided that, if a simulation was to be undertaken with a fifteen regular axes arrangement, the

fifteen co-ordinates of a point in space would be found with the new technique. It has accuracy advantages for simulations of a line and, more importantly, it is quicker to operate.

A.5.4. Randomly orientated axes :-

The axes configurations so far examined are ten and fifteen axes regularly orientated in space. The ten axes configuration was developed in case fifteen axes are too many. But what if fifteen are too few? As mentioned before, it is impossible, using polyhedra, to determine the orientations of more than fifteen regular axes. So, if more than fifteen are required, they have to be orientated randomly.

The orientation of an axis in three dimensional space can be determined by two angles. One is defined as being between the x direction and the projection of the axis on the x-y plane. The other angle is similarly designated to be between the x direction and the projection on the x-z plane.

Therefore if both these angles are chosen at random, from a uniform distribution between 0 and 360 degrees, the axis is randomly orientated.

A.5.5. Comparison of one dimensional axes configurations :-

So, there are now three different methods of transferring the features of the one dimensional simulations into three dimensions. It can be achieved with ten or

fifteen regularly orientated axes, or with any number of randomly orientated axes.

It was decided that a comparison between the relative efficiency of each of these axes configurations was warranted. This was in keeping with the general policy of close scrutiny of the simulation method. The aim was to select the configuration which was the best compromise between accuracy and frugality.

The different axes configurations studied were :-

(a) Ten regularly orientated axes. The simulations were produced by an adapted version of the program SIM3D. The subroutine COORDS2 (which calculates the ten co-ordinates of any point) was substituted for the subroutine COORDS1. Each simulation took 45.3 CP seconds to be produced.

(b) Fifteen regularly orientated axes. The simulations were products of an adaptation of the program SIM3D, with the subroutine COORDS3 in place of COORDS1. This program used 63.8 CP seconds for each simulation.

(c) Fifteen randomly orientated axes. This configuration was chosen to compete against fifteen regular axes. Perhaps the regular orientation of the axes is an unnecessary complication. The simulations were produced from a computer program RANDO (see Appendix 10). Each simulation took 48.0 CP seconds to be produced.

(d) One hundred randomly orientated axes. Obviously the production of a simulation using this axes configuration was costly on computer time. In fact each simulation took 370.9 CP seconds to be produced by the program RANDO. It was felt that one hundred was a sufficiently good representation of infinity without being prohibitively expensive to produce.

So, a comparison was made between the effectiveness of simulations produced with each of these four axes configurations.

The simulations compared were of a line of one thousand points regularly spaced at 1 metre intervals. The models from which the values originated were a semi-variogram of Spherical(50,20,0) and a Normal distribution of N(100,20).

It was considered that a line of one thousand simulated values would be more useful than a two or three dimensional shape. For instance, with a 10*10*10 cube the semi-variogram can only be drawn up to a lag of 15.6. In the principal directions only nine points of the semi-variogram can be found. However, no such limitations exist with a line of one thousand points.

The line of one thousand points was in the y direction. In other words, it was a 1*1000*1 block. It was felt that a simulation in the y direction was more valid than one in the x direction. A fifteen regular axes

simulation in the x direction is likely to be more accurate than one in the y or z directions (see section A.5.3). But when compared with another configuration, any greater accuracy which a x direction simulation may exhibit might not occur with a y or z direction simulation. The conclusion that the fifteen regular axes configuration was superior would be bogus, since the superiority would disappear if the simulated shapes were two or three dimensional.

The comparison was made between the simulations before, and after, they had been conditioned. It has been shown how large an effect the conditioning process has on the simulated values. So, it was essential for the comparison to include conditional simulations.

Twenty five simulations were produced using all the four one dimensional axes configurations under consideration. There were ten formed with fifteen regular axes, and five each with ten regular, fifteen random, and one hundred random axes. The semi-variogram of each of these unconditional simulations is shown in figure 20. The semi-variograms of the simulations formed with fifteen random axes seem to deviate from the model more than the others, particularly at low lag values. Table 2 shows the MS terms with respect to the model before conditioning. They tend to confirm that the unconditional fifteen random axes simulations are less accurate.

As previously, the simulations were conditioned to

FIGURE 20 : SEMI-VARIOGRAMS OF UNCONDITIONAL SIMULATIONS

10 REGULAR AXES

15 REGULAR AXES

15 RANDOM AXES

100 RANDOM AXES

+++ SPHERICAL MODEL

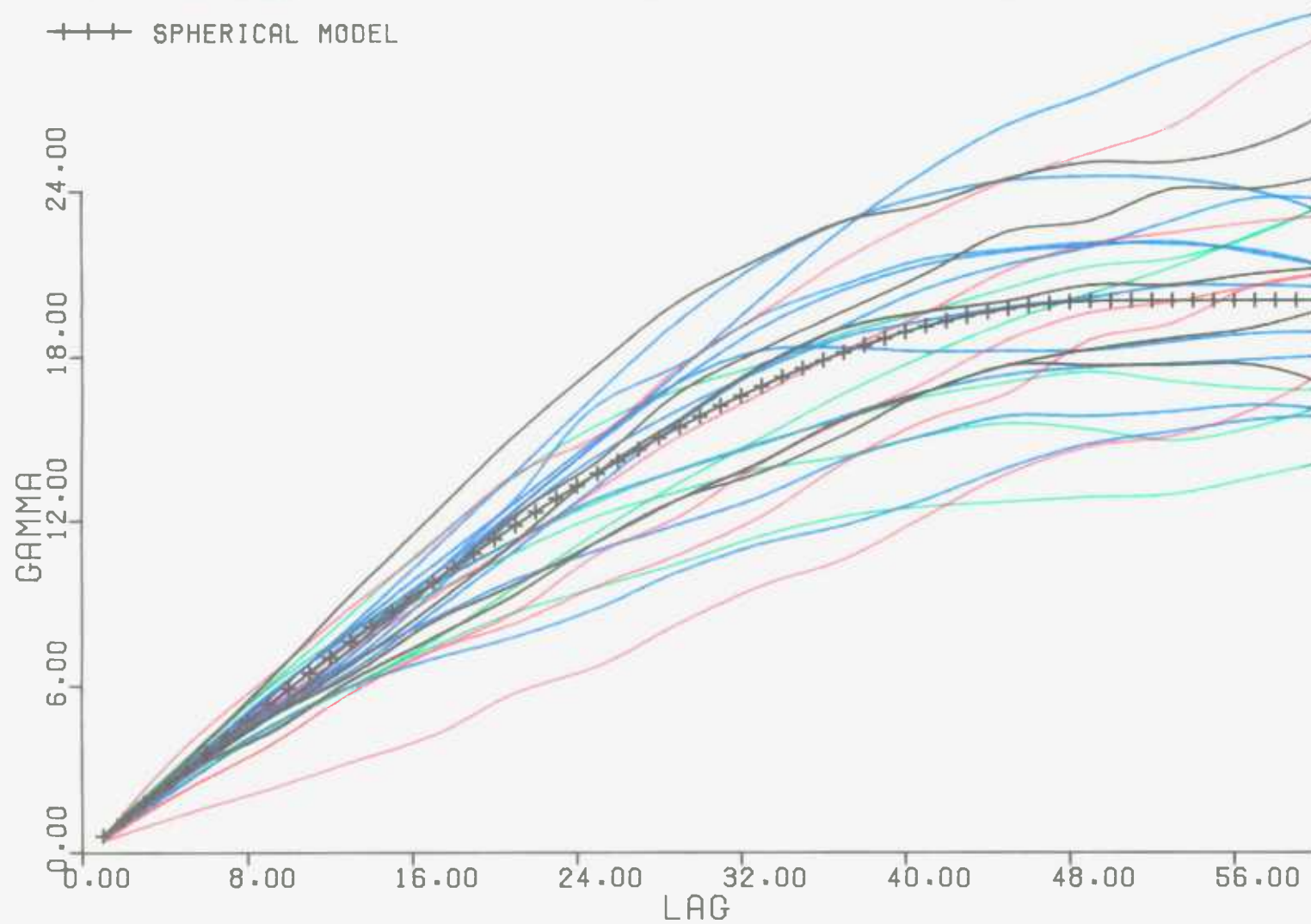


TABLE 2

MS term with respect to	Before conditioning			After conditioning	
	model (figure 20)	data (figure 22)	data (figure 23)	data (figure 22)	data (figure 23)
10 regular axes	11.9	9.1	23.4	0.70	1.08
15 regular axes	12.1	17.3	15.1	0.87	1.17
15 random axes	18.3	22.2	16.7	1.24	0.95
100 random axes	9.3	17.0	25.6	0.78	0.89

data from a similar unconditional simulation. This data simulation was formed using fifteen regular axes, and its values had been altered to fit the models more closely (see figure 21). The computer program CON was used to condition the simulations. Therefore the kriging estimate for every point was made with between two and six data points.

The simulations were conditioned to one hundred data points evenly spaced at 10 metres intervals. The semi-variograms resulting from this conditioning are shown in figure 22. It is clear that the fifteen random axes configuration simulations are less accurate than the others. At low lag values (up to about one fifth of the range), their semi-variograms differ markedly from the desired shape, and from those of the other simulations. The inaccuracies show up in the MS terms of the semi-variograms (see table 2).

This finding was checked by using another data simulation. The values had not been altered to have an average of 100.00 and a variance of 20.00. Indeed, the data values used in the conditioning differ considerably from the models. The semi-variograms of the twenty five simulations, after conditioning to one hundred regularly spaced data points are shown in figure 23. Again the semi-variograms of the fifteen random axes simulations deviate at low lag values. However, on this occasion, the average MS terms do not reflect these inaccuracies (see table 2). They are distracted by the large deviations of

FIGURE 21 : SEMI-VARIOGRAM OF ALL CONDITIONING DATA

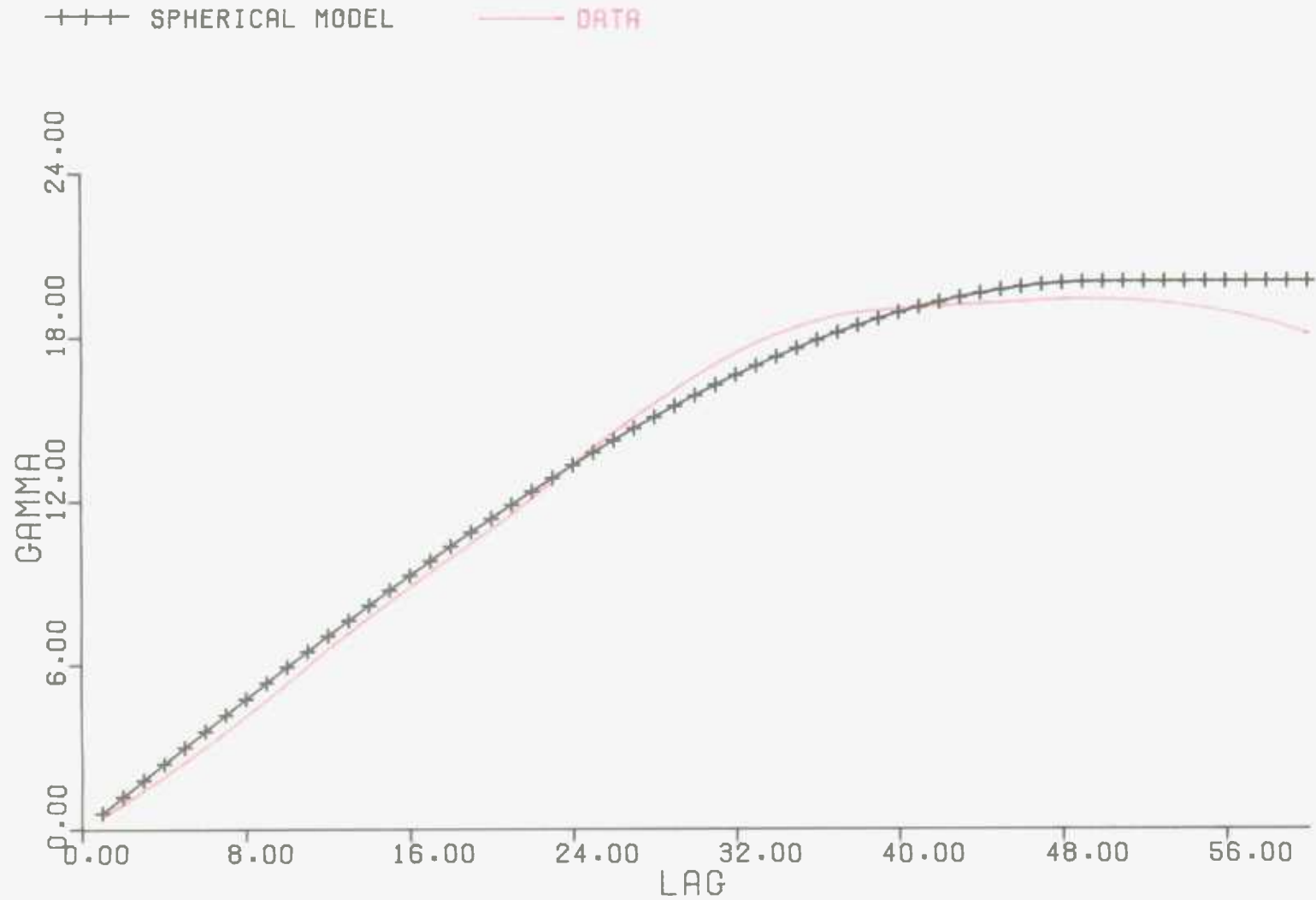


FIGURE 22 : DIFFERENT AXES CONFIGURATIONS

CONDITIONING POINTS EVERY 10 METRES

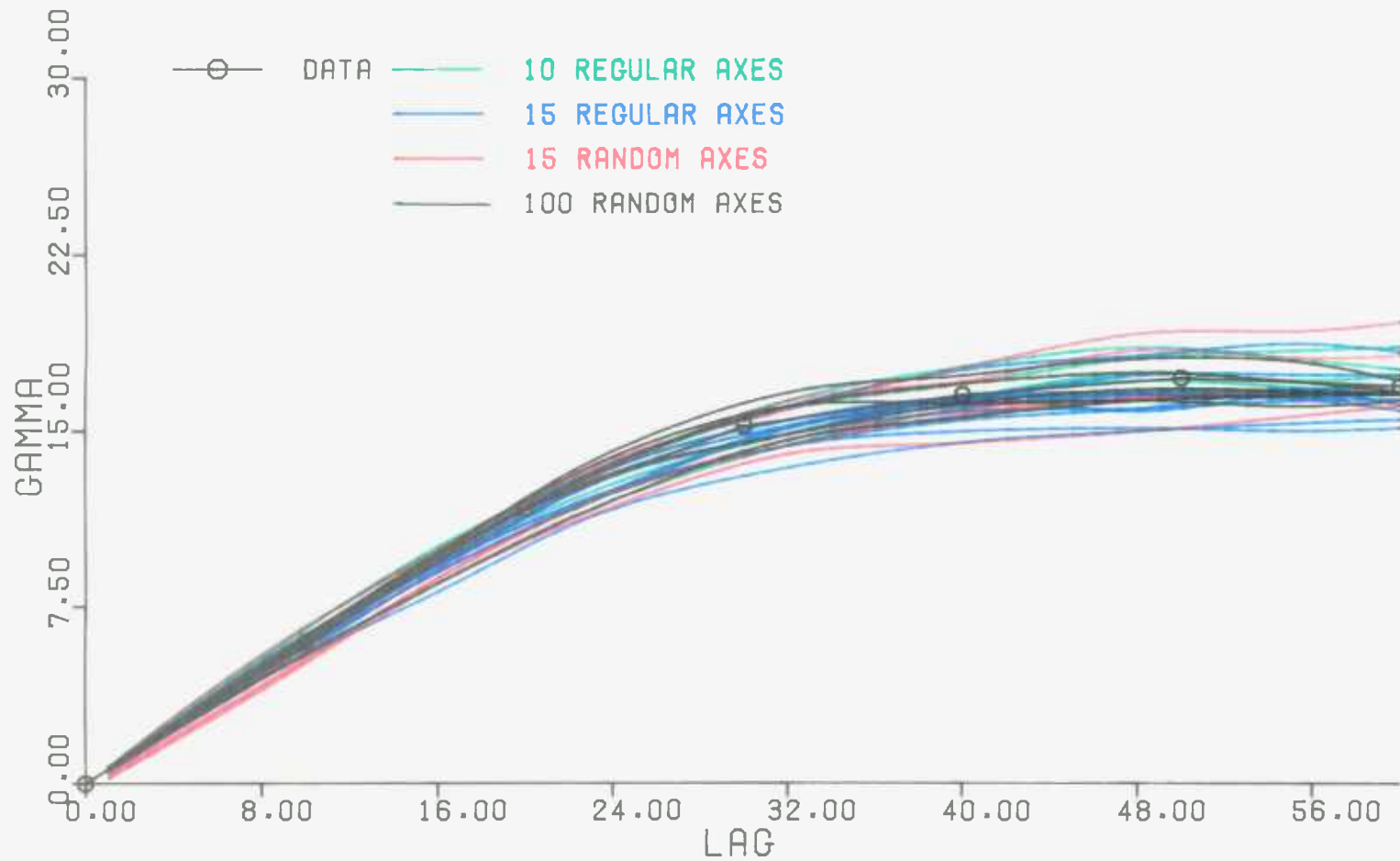
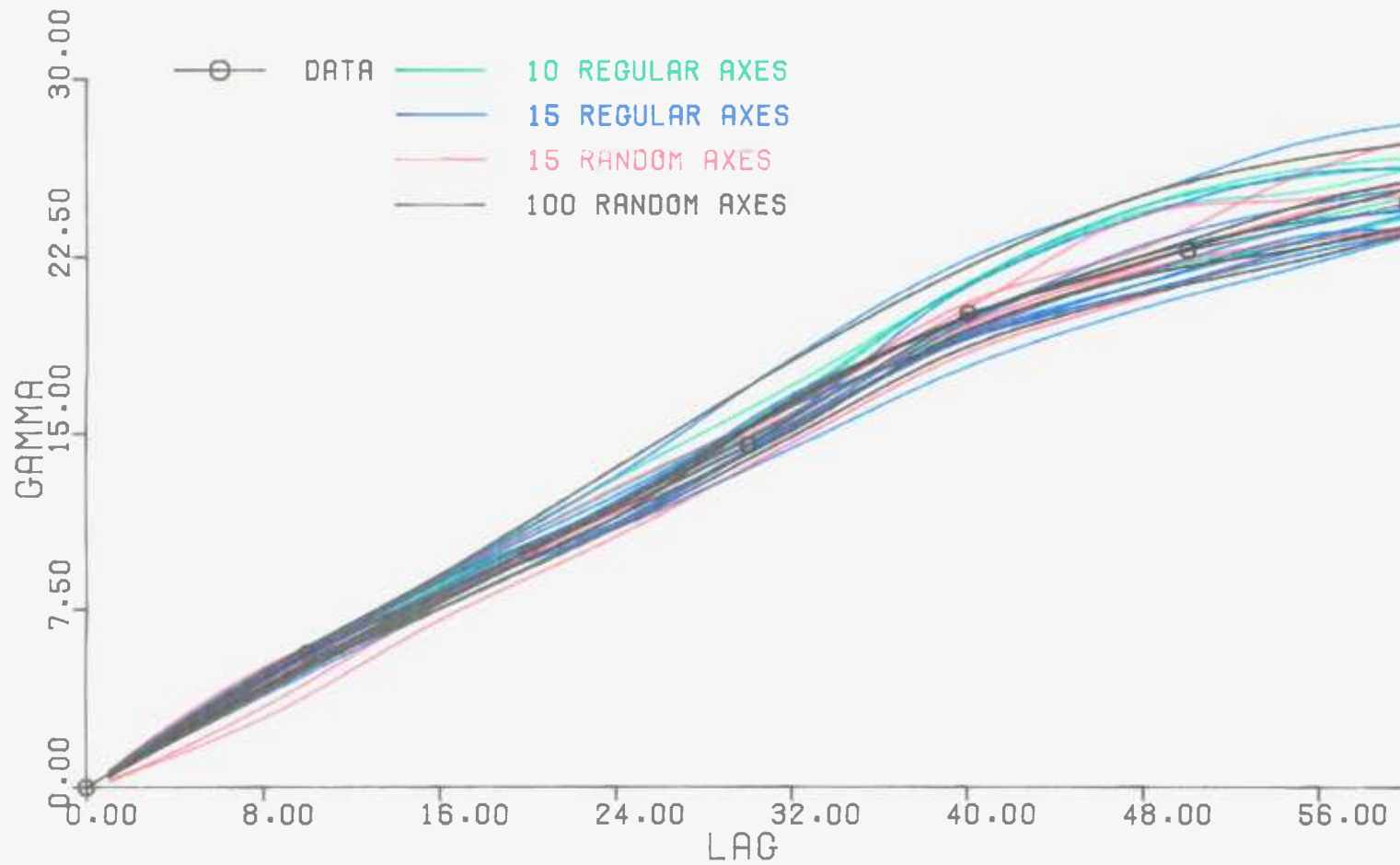


FIGURE 23 : DIFFERENT AXES CONFIGURATIONS

CONDITIONING POINTS EVERY 10 METRES



some of the other simulations at much greater lag values.

The configuration of fifteen random axes was dropped from the comparison since it has no great formation time advantage over fifteen regular axes. Also, the semi-variograms of its simulations, both before and after conditioning, are less accurate. This is true even when the average kriging variance during conditioning is only 10 percent of the sill.

So, the remaining configurations to be compared were ten regular axes, fifteen regular axes, and one hundred random axes. Ten simulations were produced using each of these one dimensional axes arrangements. All thirty simulations were conditioned to one hundred regularly spaced points from the altered data simulation. The results are shown in figure 24. The simulations were also conditioned to the same pattern of conditioning points from the unaltered data simulation. Figure 25 depicts the resulting semi-variograms, and confirms that there are no visible differences between the three methods. This is expected with an average kriging variance of only 10 percent of only the sill.

As well as one hundred, the thirty simulations were also conditioned to fewer points from the altered data simulation. Patterns with points regularly spaced at 20, 30, and 40 metres intervals were used. The semi-variograms of the conditioned values are shown in figures 26, 27, and 28 respectively.

FIGURE 24 : DIFFERENT AXES CONFIGURATIONS

CONDITIONING POINTS EVERY 10 METRES

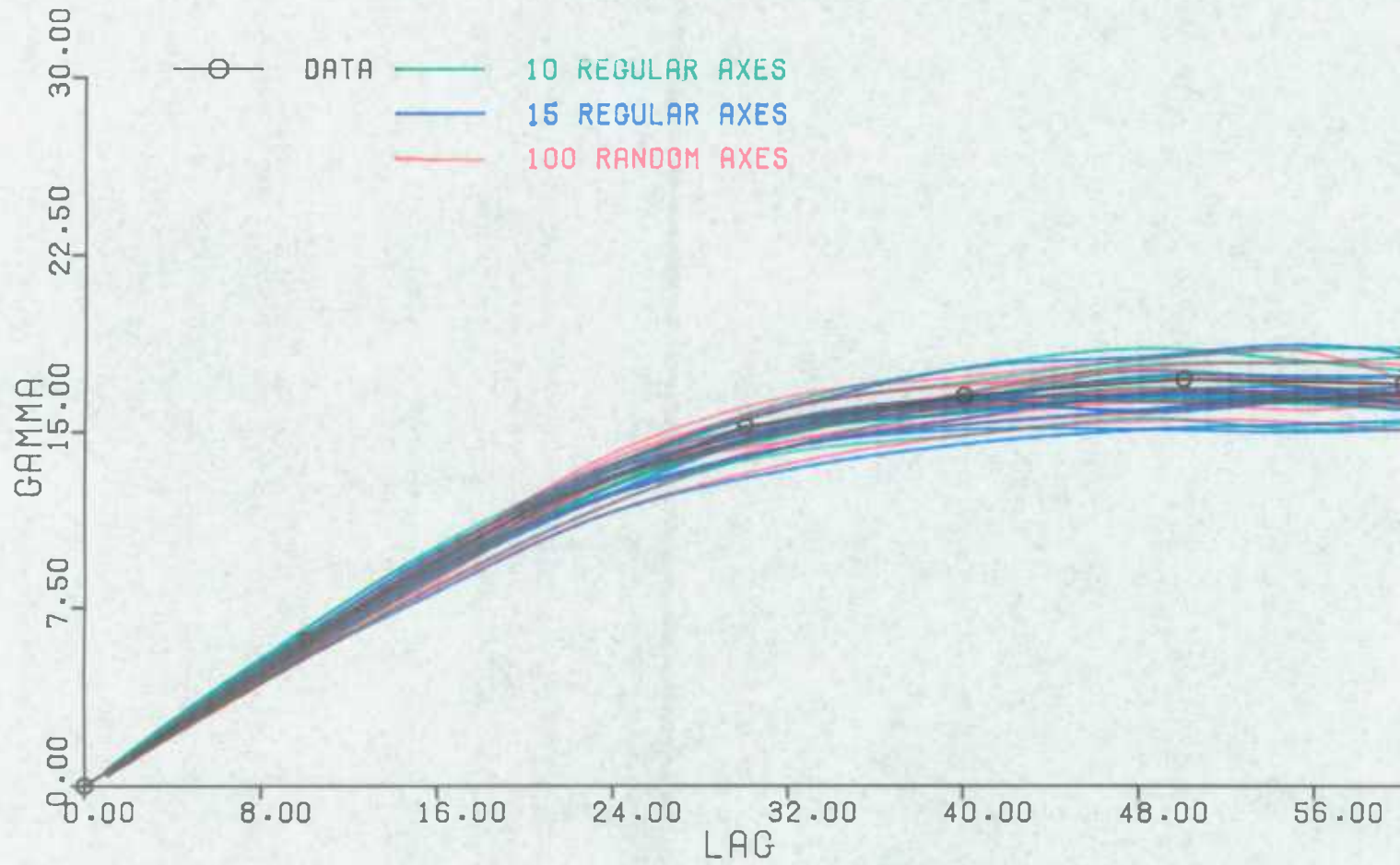


FIGURE 25 : DIFFERENT AXES CONFIGURATIONS

CONDITIONING POINTS EVERY 10 METRES

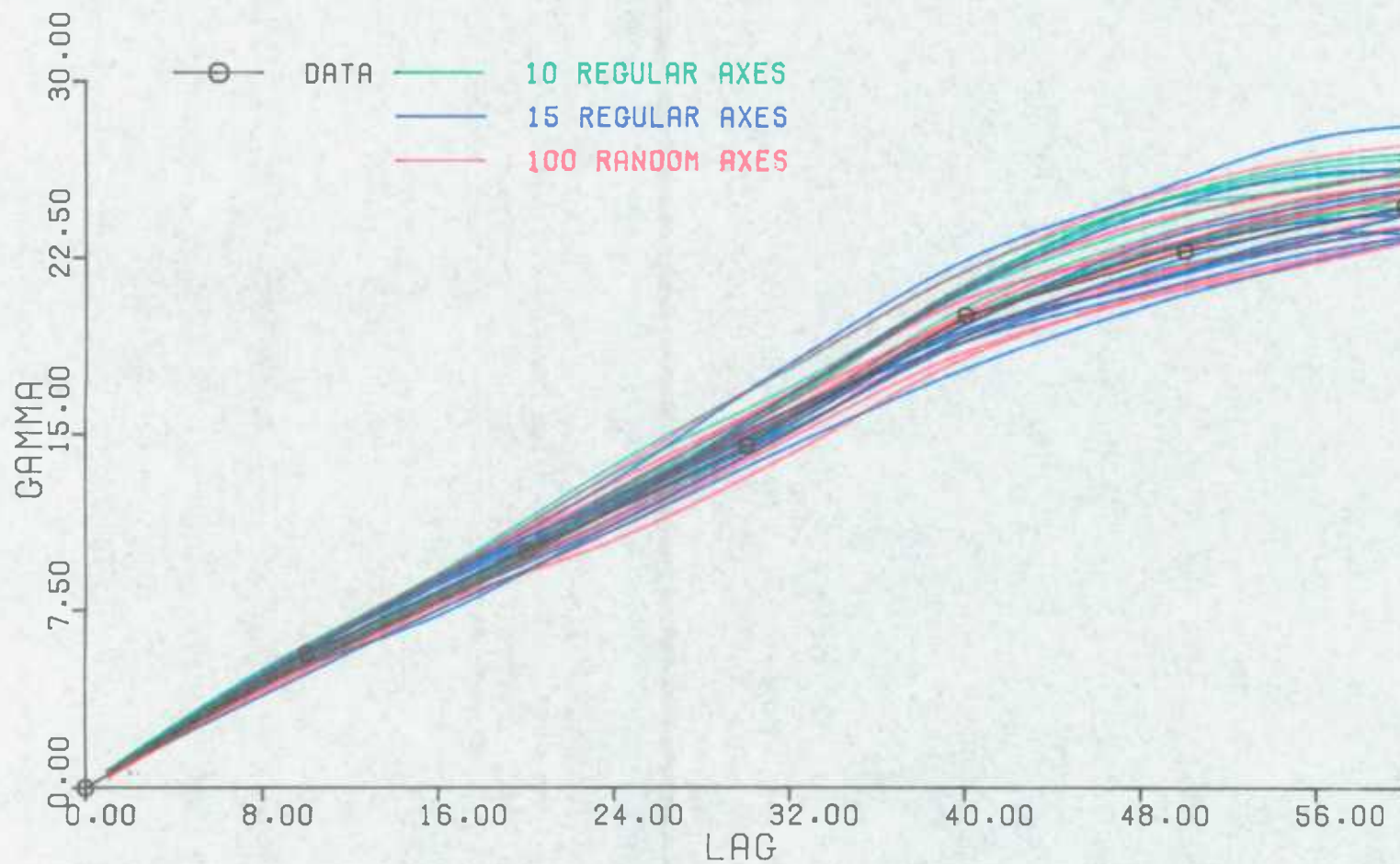


FIGURE 26 : DIFFERENT AXES CONFIGURATIONS

CONDITIONING POINTS EVERY 20 METRES

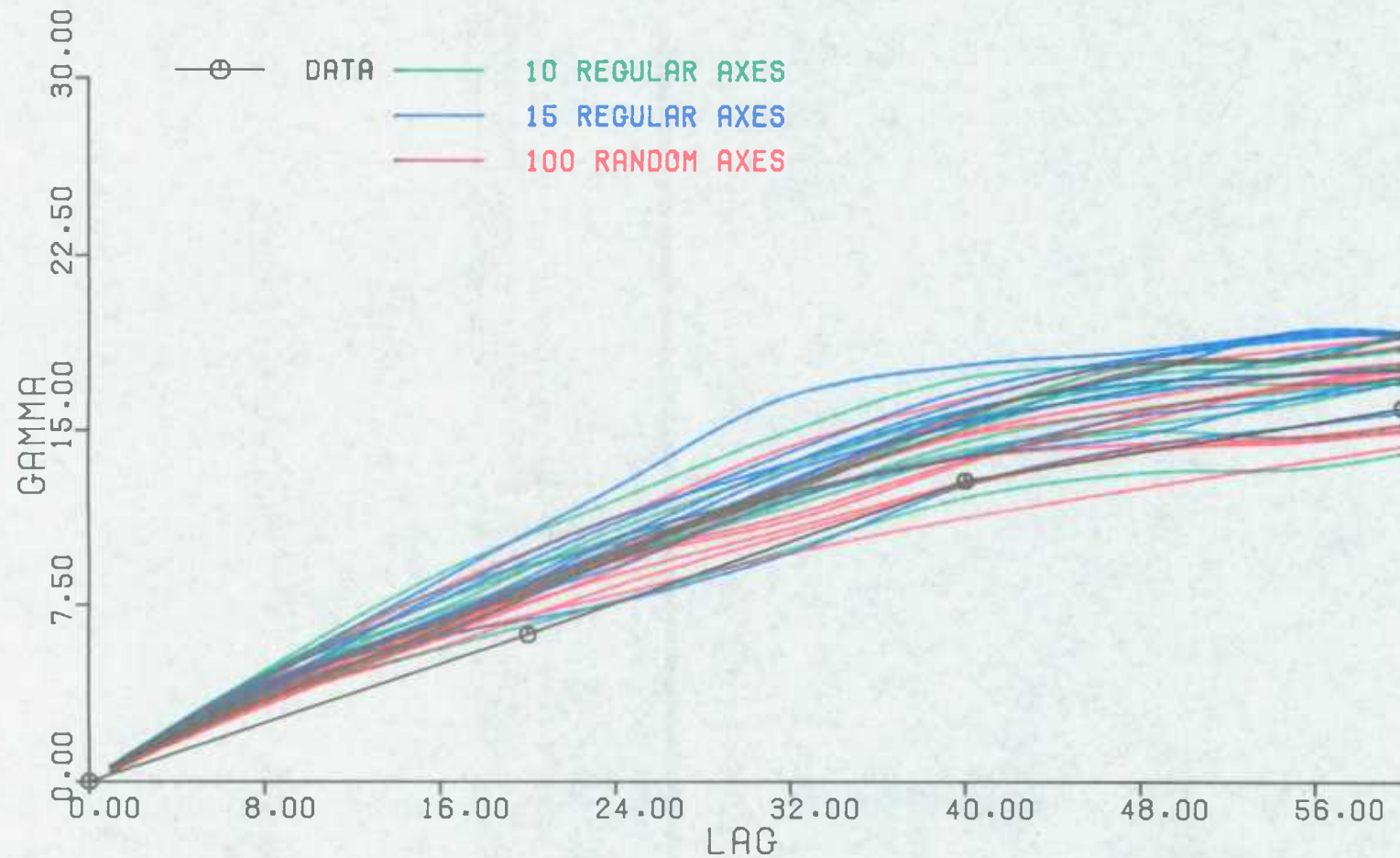


FIGURE 27 : DIFFERENT AXES CONFIGURATIONS

CONDITIONING POINTS EVERY 30 METRES

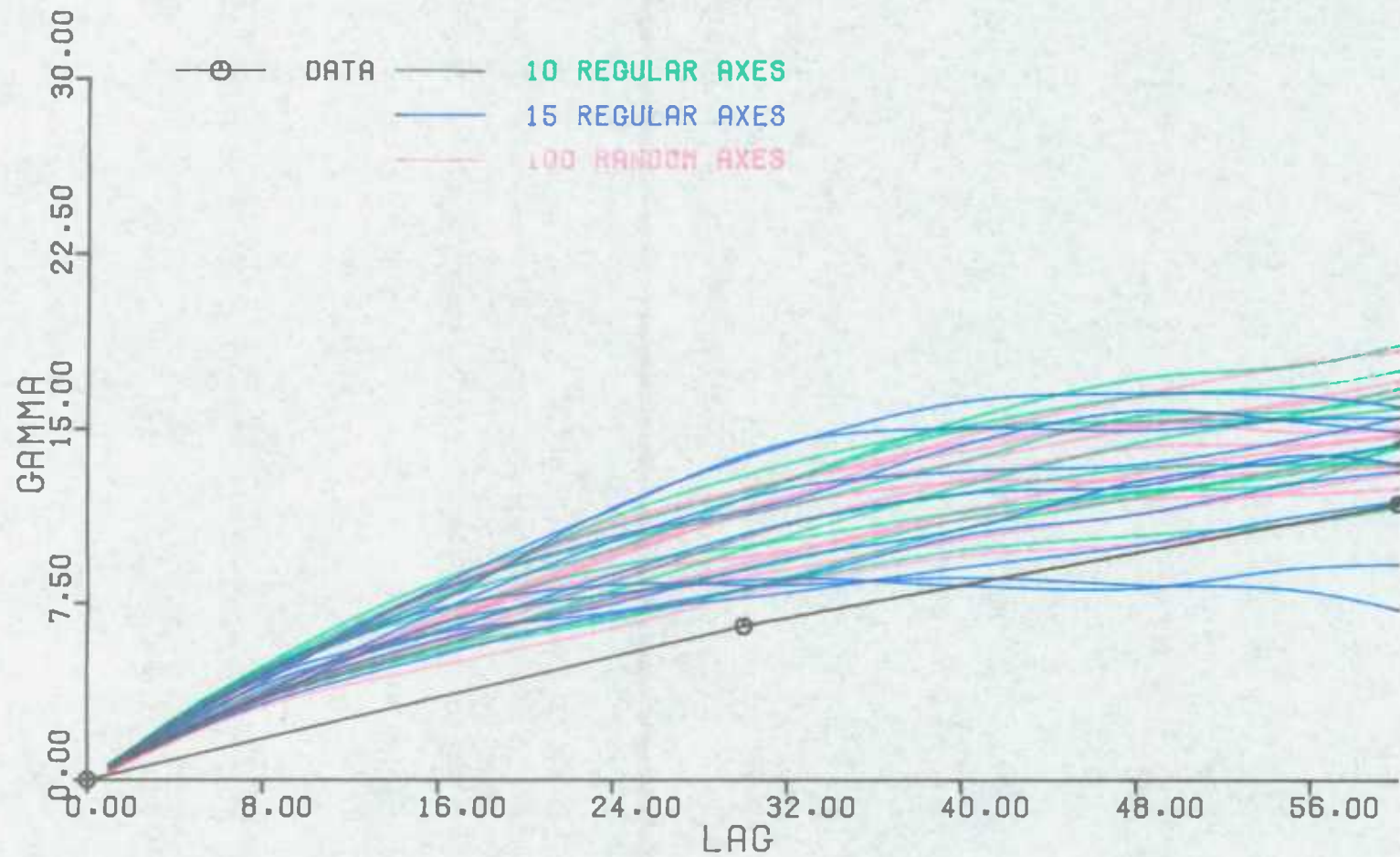
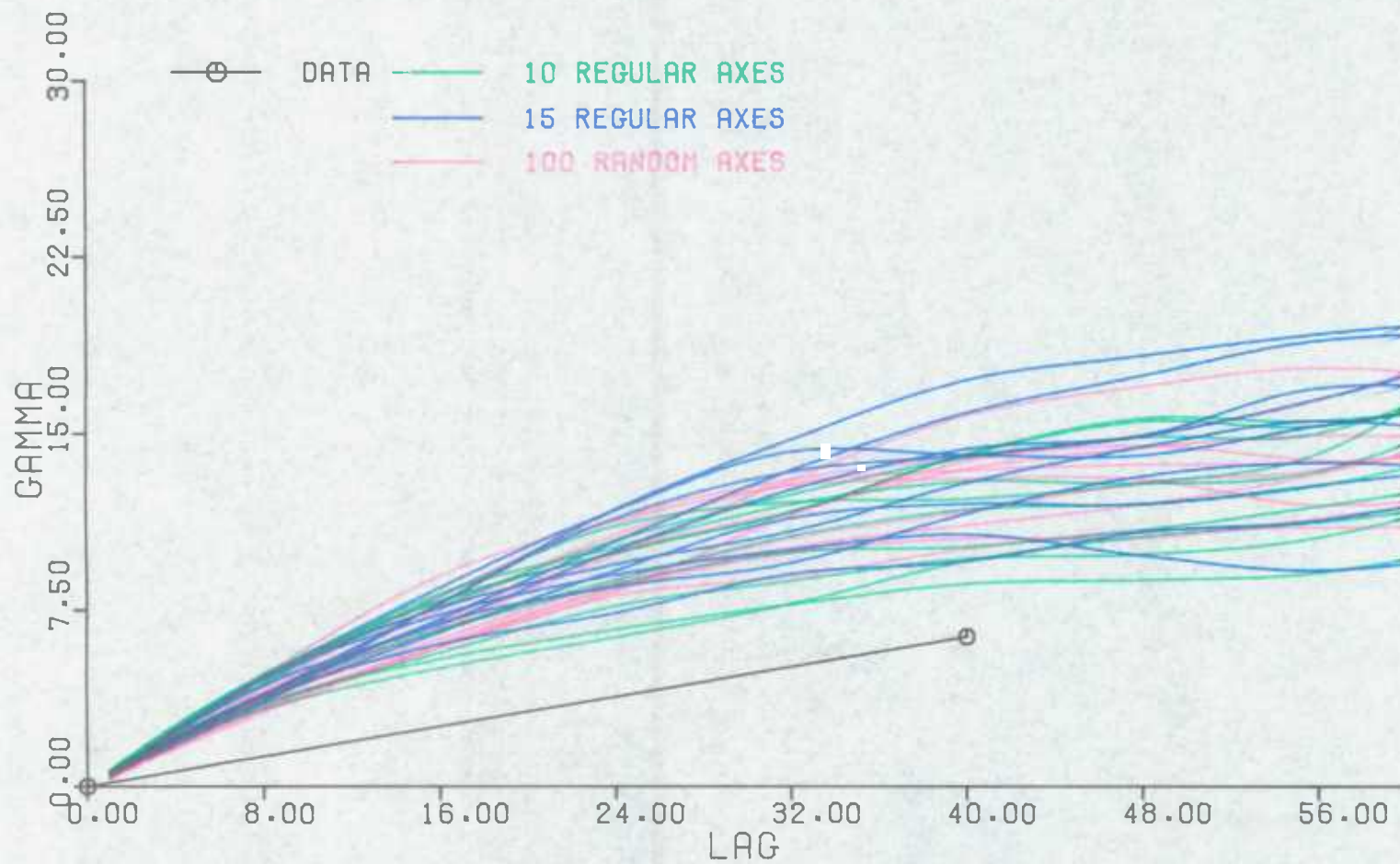


FIGURE 28 : DIFFERENT AXES CONFIGURATIONS

CONDITIONING POINTS EVERY 40 METRES



So, the semi-variograms of the thirty simulations after each of the five different conditioning points patterns can be seen. Visual inspection shows there to be no significant differences between the three sets of simulations. Table 3 shows the three average MS terms before and after conditioning with each pattern. It tends to confirm that no significant differences exist.

A.5.6. Selection of axes configuration :-

It is not certain that simulations formed from one hundred random axes are more accurate than those from the other configurations. However, it is probable that they do have some extra accuracy. The problem is that this extra accuracy is unknown, since it does not make itself distinct with only ten simulations of each type. It may be revealed with a much larger sample size. However, the fact that it is not revealed with ten simulations is itself an indication that it is not large. To be weighed against this unquantified extra accuracy is the greater cost of production. The execution time required to produce a simulation with one hundred random axes was 5.8 times that for one with fifteen regular axes.

It was considered that for all practical purposes the configuration of one hundred random axes was unreasonable. Its extreme cost offset any increased accuracy. However, it may be of use under exceptional circumstances. For instance, if highly accurate unconditional simulations are

TABLE 3

MS term with respect to conditioning data before/after conditioning

	Conditioning data points spacing							
	10 metres (figure 24)		20 metres (figure 26)		30 metres (figure 27)		40 metres (figure 28)	
	before	after	before	after	before	after	before	after
10 regular axes	7.1	0.58	16.6	5.34	49.1	16.8	32.2	95.5
15 regular axes	17.3	0.87	42.0	7.12	95.8	14.4	53.0	175.6
100 random axes	16.5	0.57	39.0	3.55	104.4	15.8	43.6	180.1

required, and cost is of little importance.

So, the two remaining one dimensional axes configurations to be considered are those with ten and fifteen regular axes. As usual, comparison consists of balancing the conflicting interests of accuracy and cost. Obviously fifteen axes simulations are more accurate than those from ten axes. Again, the problem is that this extra accuracy does not manifest itself with ten simulations of each type. To be balanced against it is the 29 percent saving in running time.

There is no correlation between the accuracy of a simulation before and after conditioning, when the average kriging variance is less than about one third of the sill. Under these conditions it was considered that simulations from ten regular axes would suffice. Indeed if cost is at a premium fewer regular axes may be applicable.

For all situations throughout the rest of this study, the average kriging variance achieved during conditioning was greater than one third of the sill. Under these conditions the accuracy of the unconditional simulations is important. So, it was considered that fifteen axes were needed for their extra, albeit unquantified, accuracy.

The arrangement of fifteen, regularly orientated, axes was adopted as the most appropriate configuration of one dimensional axes.

SECTION B : APPLICATIONS IN SOUTH CROFTY MINE

B.1.SOUTH CROFTY MINE

B.1.1.General description :-

South Crofty tin mine is situated in the Camborne and Redruth district of Cornwall. On 31st March 1979 it had 679 employees, and in the twelve months up to that date treated 236,300 tonnes of ore, averaging at 0.68 percent tin.

South Crofty lies within granite overlain by metamorphosed sediments. The area is cut by porphyry dykes which strike at about 70 degrees and are generally North dipping. The mineralisation postdates the dykes, and consists of two sets of lodes intersecting approximately at right angles. A set of 70 degree striking lodes dip at angles up to 40 degrees, in either direction. They have been mined for both tin and copper. The tin being below the granite/sediments contact and the copper above. The mineralisation consists of cassiterite, arsenopyrite, wolframite, and chalcopyrite, with gangue minerals of chlorite, quartz, tourmaline, and hematite. The second set of lodes, striking at about 160 degrees, has no economic significance. They cut and fault the 70 degrees striking lodes.

The lode used throughout this study was No.9 lode. Taylor (1966) provides contour diagrams (of lode width and lode assay), and a Conolly contour diagram for the lode. He

identifies six oreshoots, widely scattered and generally of low grade.

At the time of data collection (November 1976), the workings within No.9 lode extended approximately 200 metres vertically, and 600 metres horizontally. It had five main development level drives in it, spaced at vertical intervals of about 40 metres. Extraction was chiefly carried out using overhand shrinkage stoping methods. There were a total of eighteen stopes within the lode, three of which were being worked.

B.1.2. Available data :-

Sampling results from 2366 points within No.9 lode were obtained. Of these 749 are stope samples, 1149 come from the main development drives, and 468 from raises and inter-level development. The stope samples were taken from each stope approximately after every two working weeks. They were obtained from the worked faces themselves, at intervals of either 5 feet, 10 feet, 3 metres, or 6 metres; depending on when the sampling was carried out. The development samples were taken along the drives at regular intervals of either 5 feet, 10 feet, or 3 metres.

At every sampling point a channel cut was taken across the lode. Similarly, samples of the country rock on both sides of the lode were obtained.

For all the sampling points the following information is available :-

(a) Lode width - Measured perpendicularly across the lode.

(b) Lode assay - Obtained from analysis of the channel cut across the lode.

(c) Lode accumulation - This is equal to lode width times lode assay. Thus :-

$$(c) = (a) * (b)$$

(d) Call width - The anticipated width of stope needed to mine this lode width. This was defined as 1.0 metres (or 3.0 feet), or the lode width plus 0.2 metres (or 0.5 feet); whichever is the greater. Thus it allows for overmining of the lode.

(e) Call accumulation - This is the total accumulation which would be achieved over the call width. It is calculated using the average assay of the adjacent country rock :-

$$(e) = (c) + [(d) - (a)] * \text{average country rock assay}$$

(f) Call assay - The average assay value over the call width, determined from :-

$$(f) = \frac{(e)}{(d)}$$

For all the stope samples three more figures are available.

These are :-

(g) Actual width - The width of the stoping face. In an ideal world this would be equal to the anticipated, or call width.

(h) Actual accumulation - This is the

accumulation over the actual width. Similarly to the call accumulation it is found from :-

$$(h) = (c) + [(g) - (a)] * \text{average country rock assay}$$

(i) Actual assay - The average assay value over the actual width, determined from :-

$$(i) = \frac{(h)}{(g)}$$

Each sampling point is located by the distance along the stope face, or development drive, to a known surveyed position. Its location was recorded on a longitudinal section of the lode.

The longitudinal section of No.9 lode was drawn using distances measured along the lode. Thus it is a projection of the 'unrolled' lode.

From the section, the position of each sample point was fixed. The two co-ordinates, of depth and Easting, were determined to the nearest metre.

B.2.DATA ANALYSIS

B.2.1.Data accuracy :-

The rest of this study uses the data from the samples taken within No.9 lode. Thus their reliability is of great importance.

No detailed examination of the accuracy of South Crofty's sampling technique has ever been carried out. However, it is the author's opinion that the method of sampling employed is not open to a large bias. This view is supported by Taylor (1966), and also by Kuscevic, Thomas, and Penberthy (1972).

Suppose that it is wished to measure the height of an infinite number of people. If the measurements can be made to the nearest 0.5 centimetres, the number of figures ending in .5 should equal that of figures ending in .0. However, if the samplers taking the measurements are not thorough this may not be true. They might take the measurements to the nearest 1.0 centimetres, and visually estimate the last digit. Due to human failings there would probably be many more figures ending in .0 than in .5. However, the distribution of the last digit before the decimal point (measuring single centimetres) would be uniform. This would reflect the true accuracy of the data measurements; that is to the nearest 1.0 centimetres.

Using this approach the accuracy of the available data was examined. Of all the samples, 2038 were measured in

imperial units (feet and pounds per ton), and 328 in metric units (metres and percent). The four groups of data were examined separately :-

(a) Imperial lode widths. These were measured to the nearest 0.25 feet. The histogram of the figures after the decimal point is shown in figure 29. This is clearly not uniform, with greater numbers of measurements ending in .00 and .50. However, if the group size is enlarged the histogram becomes much more uniform :-

.25 and .50 - 933

.75 and .00 - 1111

The imperial lode widths seem to be accurate to 0.5 feet. For the rest of this study, any such measurement was rounded up to the nearest 0.50 feet before use.

(b) Metric lode widths. These measurements were taken to the nearest 0.1 metres. The histogram of the digit after the decimal point is shown in figure 30. This is clearly not a uniform distribution, a fact that is confirmed by the chi-squared 'goodness of fit' statistic. This is 26.43 which, with 9 degrees of freedom, is significant at the 0.5 percent level. Figure 30 also shows the histogram of measurements rounded to the nearest 0.2 metres. The chi-squared statistic (4.27 with 4 degrees of freedom) is not significant

FIGURE 29 : HISTOGRAM OF FIGURES AFTER DECIMAL POINT FOR IMPERIAL LODE
WIDTHS

Average = 0.33 feet

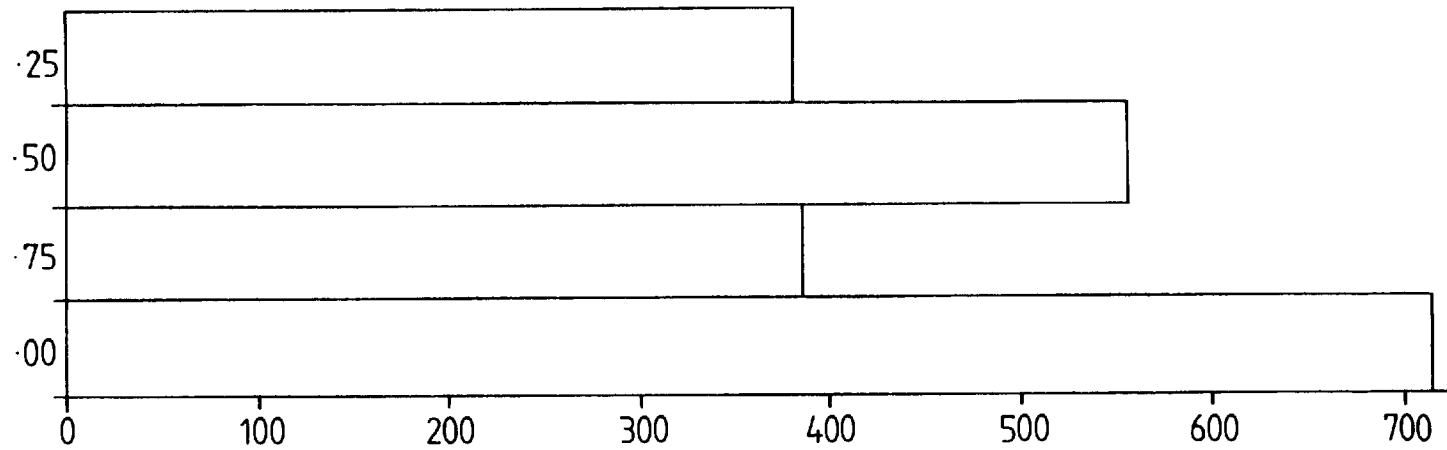
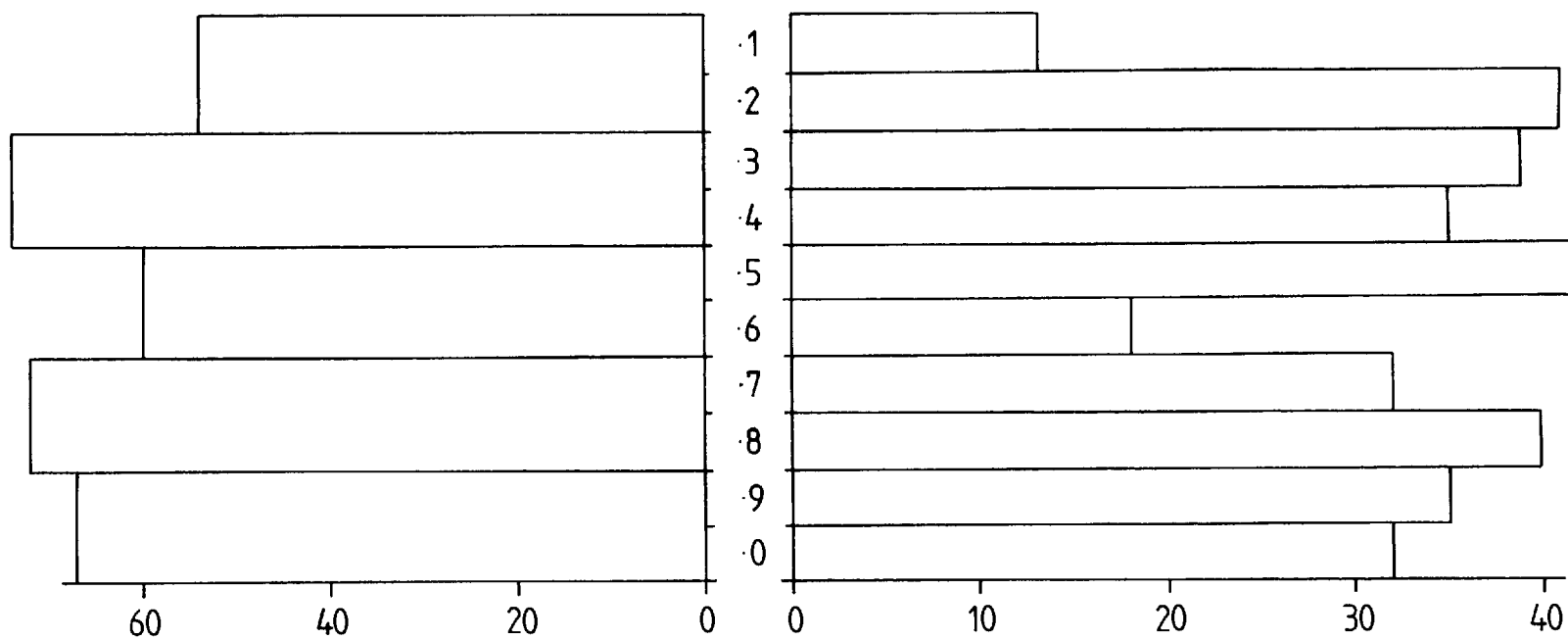


FIGURE 30 : HISTOGRAM OF FIGURES AFTER DECIMAL POINT FOR METRIC LODE WIDTHS

Average = 0.45metres



at the 5 percent level, thus confirming the visual impression of a uniform distribution.

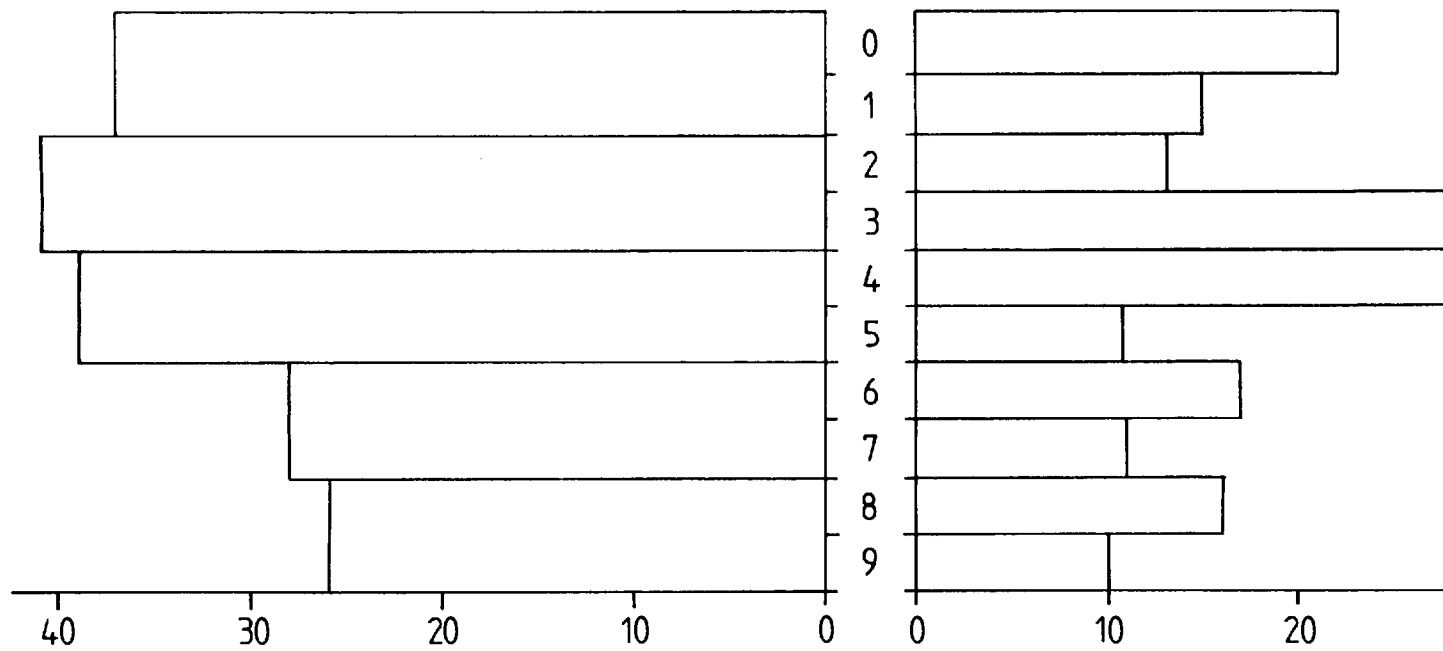
The 5 percent level of significance is used as the decision level as to whether a histogram does or does not fit a certain distribution. The choice of 5 percent is common for geological variables (Mukherjee, 1975).

The metric lode width measurements appear to be accurate to 0.2 metres. Before use in the rest of this study any such measurement was rounded up to the nearest 0.2 metres.

(c) Imperial lode assays. These measurements were taken using standard vanning techniques, and they are given to the nearest 1 pound per ton (lbs/ton). The histogram of the last digit of all the sample values shows a marked skewedness. This is because there are genuinely more samples of 1 lbs/ton than of 9 lbs/ton. To avoid this effect only sample assays between 50 and 100 lbs/ton were considered. The histogram of the final digit of these values is shown in figure 31. Its chi-squared statistic (23.91 with 9 degrees of freedom) is over 5 percent significant, thus confirming the visual impression of a peaked behaviour. Figure 31 also shows the histogram after the digits have been placed in groups of 2 lbs/ton. This seems to be uniform and the chi-squared statistic (5.35 with 4

FIGURE 31 : HISTOGRAM OF LAST DIGIT OF IMPERIAL LODE ASSAYS BETWEEN 50 & 100 lbs / ton

Average = 4.03 lbs / ton



degrees of freedom) is not significant.

The imperial lode assays measurements seem to be accurate to 2 lbs/ton. All such measurements were rounded to the nearest 2 lbs/ton before use in the rest of this study.

Of all the imperial lode assay values, 111 were recorded as a 'trace' value. In other words, the tin content was noticeable but did not reach 1 lb/ton. Such 'trace' values were treated as 0.5 lbs/ton for the purposes of the rest of this study.

(d) Metric lode assays. These measurements were achieved using a portable radioisotope X-ray fluorescence (P.I.F.) analyser (Bowie, Darnley, and Rhodes, 1965). They are recorded to an accuracy of 0.01 percent.

The P.I.F. analyser produces a value which is converted to percentage of tin with the use of a graph. The method employed at South Crofty requires the analyst to visually interpolate from the graph. A line on the graph of about 30 centimetres length has to be estimated, and to obtain an accuracy in the final answer of 0.01, the end of this line has to be accurate to within about 0.2 centimetres. This is asking a lot of any analyst with typical human failings. It was the method of producing the P.I.F. assay values that first alerted the author to the influence of the human factor on the

samples.

As shown in figure 32, the histogram of the last digit of the P.I.F. values demonstrates the influence of the human factor. There is a clear bias towards a second decimal digit of 0, and the chi-squared statistic (62.54 with 9 degrees of freedom) is highly significant. Figure 32 also shows these digits aligned in groups of two. Non uniformity persists to a large extent, with the chi-squared statistic (17.88 with 4 degrees of freedom) still significant. Grouping the last digits of the P.I.F. samples to the nearest 0.05, produces :-

. 8 and . 9 and . 0 and . 1 and . 2 - 182
. 3 and . 4 and . 5 and . 6 and . 7 - 141

This seems acceptable, and so throughout the rest of this study P.I.F. assay values were rounded to the nearest 0.05 percent before use.

B.2.2. Conversion of imperial measurements :-

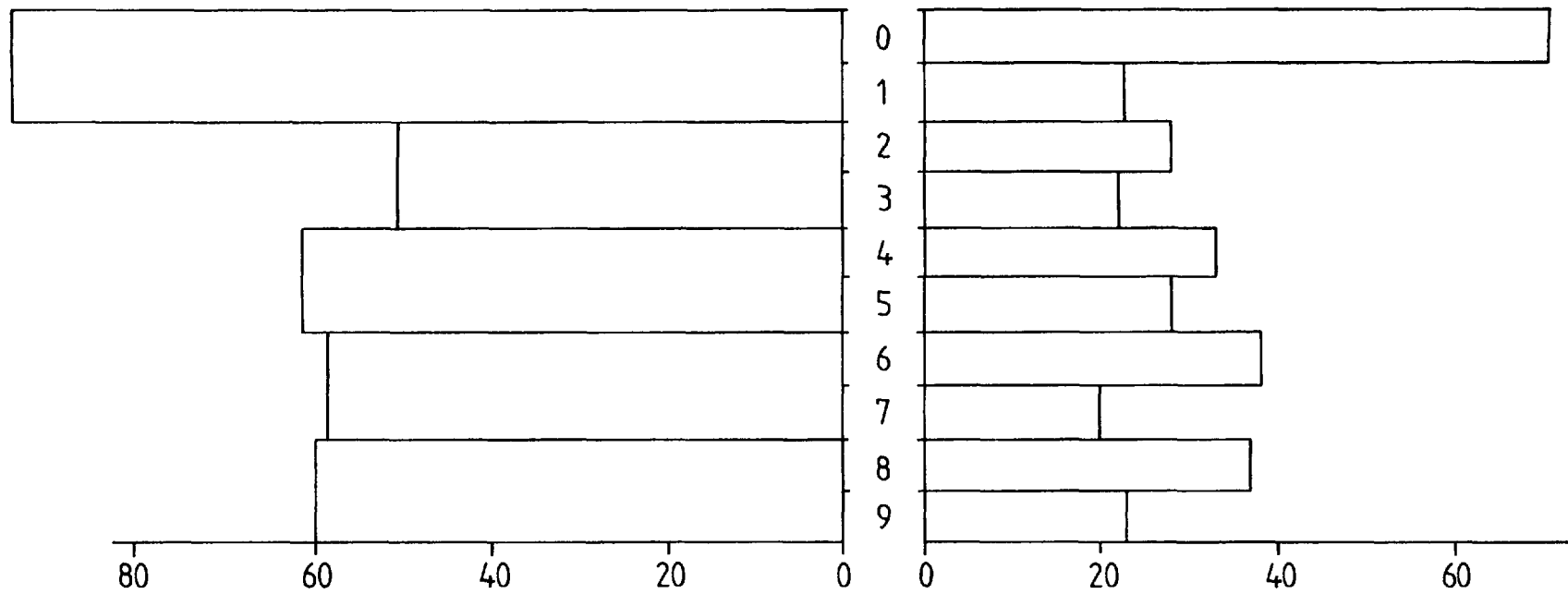
It was decided to proceed with all the sample measurements in metric units. The conversion of the imperial units into metric units required :-

(a) Feet to metres. The conversion here is obviously achieved by multiplication by 0.3048.

(b) Pounds per ton to percent. The imperial

FIGURE 32 : HISTOGRAM OF LAST DIGIT OF METRIC LODE ASSAYS

Average = 5.11%



measurements of lode assay values were produced by the vanning process. This actually records the content of 'black tin' in the sample. 'Black tin' principally comprises cassiterite, but its exact composition can vary between orebodies. Beringer and Beringer (1898) state that for Cornish mines it has an average cassiterite content of 92.5 percent. This means that it is 79 percent tin.

The P.I.F. analyser measures the percentage of pure tin in a sample.

Thus, if the 'black tin' is assumed to be 92.5 percent cassiterite, conversion from a vanning to a P.I.F. value requires multiplication by :-

$$\frac{0.79*100.0}{2240}$$
$$= 0.035$$

An investigation into the vanning/P.I.F. conversion factor at South Crofty was carried out. Normal samples from underground were analysed by both methods. Using the results of 145 such comparisons, the correlation coefficient was found to be 0.97. It was desired to find the best line of the form :-

$$P.I.F. = \text{Constant} * \text{Vanning}$$

The fit of such a line to the data can be represented by the residual sum of squares (RSS). That is, the sum of the 145 values of :-

$$[P.I.F. - (\text{Constant} * \text{Vanning})]^2$$

For various values of the constant, the RSS was found to be as follows :-

Constant = 0.041	RSS = 85.0
" 0.042	" 81.3
" 0.043	" 79.7
" 0.044	" 80.1
" 0.045	" 82.6

Obviously, the line with the smallest RSS term is the closest fit to the actual data. So, a value for the constant of 0.043 was chosen. The standard deviation of this figure is found from :-

$$\sqrt{\frac{\text{RSS}}{(N-1)} * \frac{1}{\text{SUMV2}}}$$

where N = number of data points = 145

SUMV2 = sum of all 'N' values of 'Vanning'²,

This was calculated to be 0.00074. Thus 90 percent confidence limits can be attributed to the 0.043 constant. These are at 0.0418 and 0.0442, and show the conversion factor to be accurately known.

Throughout the rest of this study to obtain the equivalent P.I.F. value of a vanning assay, it was multiplied by 0.043.

B.2.3. Calculation of average assay values :-

Consider a set of samples each with a known width and assay, and therefore also a known accumulation.

The average width of the samples is determined by the

usual arithmetic method.

The average assay value is fairly commonly defined as the average accumulation divided by the average width. Thus the average assay is a function, not only of the individual sample assays, but also of the widths. The assays taken at the larger widths are given a greater influence on the average assay value. An inter-relationship between the average width and average assay of a set of samples is introduced.

The relationship between all the sample lode width and assay values was examined. A scattergram, from the 1036 main development level samples where the lode assay is not a 'trace' value, shows a purely random pattern. The correlation coefficient of -0.04 confirms this, since (with 1034 degrees of freedom) it is not even different from 0.0 at the 25 percent significance level. Similarly, a scattergram from 745 stope samples only displays a random pattern, and has a negligible correlation coefficient (-0.01 with 743 degrees of freedom).

There is no relationship between the lode widths and the lode assay values. They are independent variables. Because of this, it was felt that to introduce an inter-relationship between the average width and average assay of a sample set is unwarranted. The definition of the average assay value stated above was rejected. Instead it was defined as the total of the assay values divided by the number of samples.

B.2.4.Semi-variogram analysis :-

The width measurements obtained at any sampling point were taken perpendicularly across the lode. Similarly, the assay values are from equally sized channel cuts across the lode. Considering the two dimensional longitudinal section of No.9 lode, all the samples were taken in the third dimension. Therefore, the samples can be treated as point samples. This means that the semi-variograms of the data are from punctual samples, and no regularisation needs to be considered.

The mathematics of the semi-variogram function have still to be fully analysed. At the time of writing, there exists no standard method for measuring the reliability of any semi-variogram value. It is merely stated that the accuracy of a semi-variogram value is dependent on the number of pairs used in its calculation.

With only intuitive justification, several methods of finding the reliability, or robustness, of a semi-variogram can be proposed. The three methods outlined below are all basically of the same type. The semi-variogram, from a set of samples, is compared with the semi-variogram calculated from only a certain proportion of the total set.

(a) A proportion of the samples is removed from the total set at a regular space interval.

A line of two hundred regularly spaced points was produced by an unconditional simulation. From this total sample set, every fifth sample was extracted.

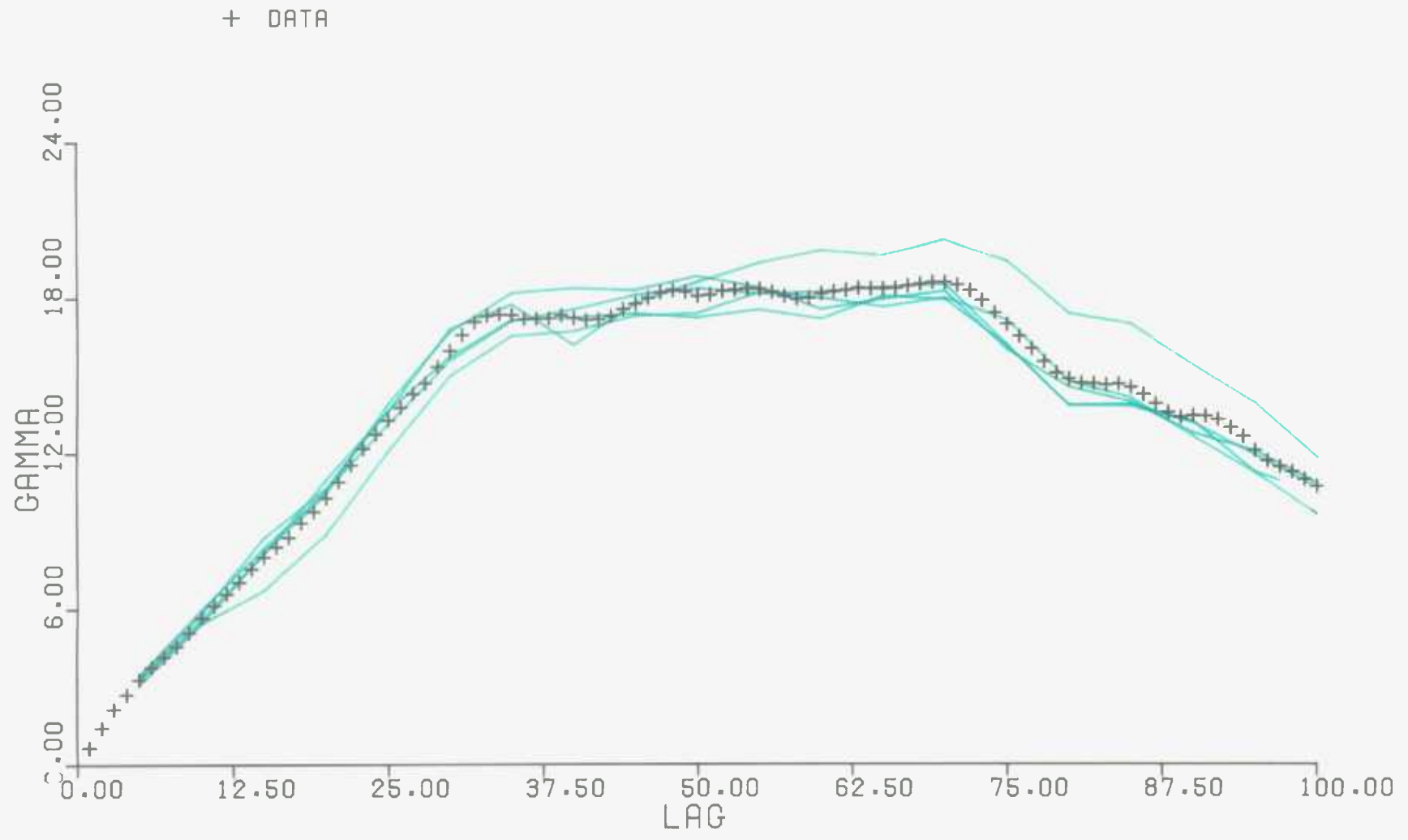
The semi-variogram of these forty points was found. Repeating this, for the other four possible ways of extracting every fifth sample, produced the results shown in figure 33.

The envelope of minimum and maximum semi-variogram values seems to get wider as the average semi-variogram value increases. The semi-variogram from all two hundred samples (the average of the five semi-variograms) lies near the centre of this envelope.

(b) A proportion of the samples is removed from a section of the total set. Consider a line of two hundred regularly spaced samples. From this total set it is possible to take out one hundred consecutive points, and to find their semi-variogram. There are one hundred and one ways of extracting the consecutive points, each producing a slightly different semi-variogram.

Such an operation was carried out, with the line of two hundred points coming from an unconditional simulation. To save computer time, the one hundred consecutive points were removed only eleven times. The first semi-variogram came from the one hundred points starting at position number 1, and the second from the points starting at position number 11. Maintaining this shift the eleventh semi-variogram used the points between position

FIGURE 33 : SEMI-VARIOGRAMS FROM EVERY FIFTH SAMPLE



numbers 101 and 200. The minimum and maximum of the eleven semi-variogram values was found for every lag value. Figure 34 shows the resulting envelope, and the semi-variogram of all two hundred points. Figure 35 shows the result of the same removal operation on a line of two hundred real samples. These were the lode width measurements along part of the 310 level drive.

The width of the minimum/maximum envelope seems to increase as the lag increases. Of course, this is also as the number of contributing sample pairs is decreasing.

(c) A proportion of the samples is taken at random from the total set.

Five samples were extracted at random from a line of two hundred simulated values. The semi-variogram of the remaining samples was found. This is equivalent to extracting and examining one hundred and ninety five samples. This process was repeated many times and the minimum and maximum semi-variogram value for every value of lag was found. The minimum/maximum envelope was plotted. In all there are over two thousand million possible ways of choosing five from two hundred. Fortunately it was found that only fifty such extractions gave a smooth minimum/maximum envelope.

The whole operation was repeated for extractions of

FIGURE 34 : MINIMUM/MAXIMUM ENVELOPE OF SEMI-VARIOGRAMS

FROM 100 CONSECUTIVE SAMPLES - SIMULATION 2637

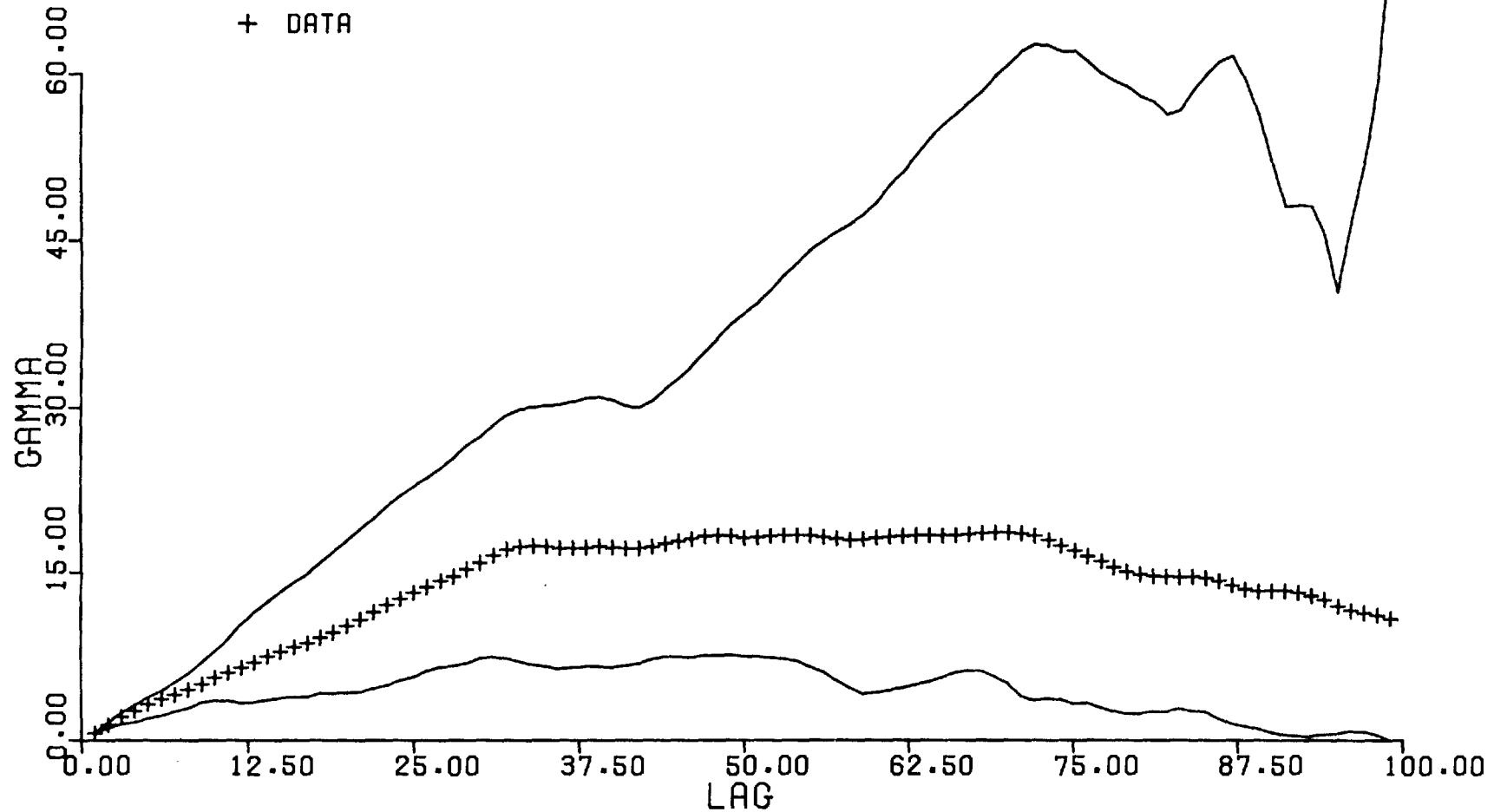
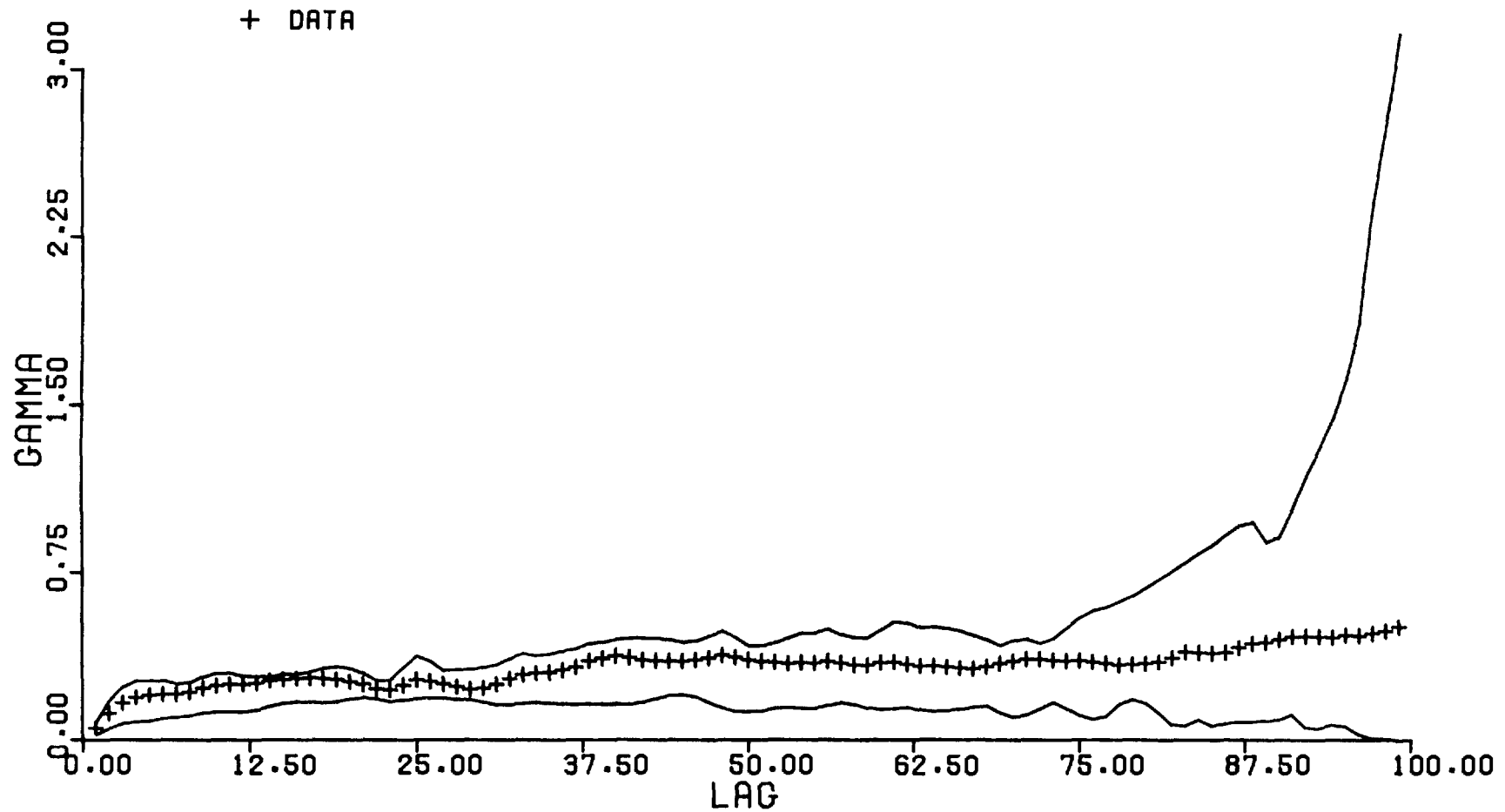


FIGURE 35 : MINIMUM/MAXIMUM ENVELOPE OF SEMI-VARIOGRAMS

FROM 100 CONSECUTIVE SAMPLES - LODE WIDTHS



ten, twenty, and forty samples. Figure 36 shows the results.

Repeating the procedure with the two hundred real samples, of lode widths from 310 level, produced the results shown in figure 37.

The minimum/maximum envelopes shown in figures 36 and 37 indicate the robustness of the semi-variogram values. The wider the envelope, the greater is the susceptibility of the semi-variogram value to changes in its contributing samples.

A semi-variogram value is half the arithmetic mean of a distribution. The distribution is of the term :-

$$[G(x) - G(x+h)]^2 \quad \text{see section A.1.2}$$

Obviously the accuracy of such an arithmetic mean is proportional to the variance, or spread, of this distribution. If the distribution has a small spread the arithmetic mean is highly accurate, or robust. Removal of some members of the $[G(x) - G(x+h)]^2$ distribution would have little effect on it.

The minimum/maximum envelopes shown in figures 36 and 37 show the effect such removals have on each distribution arithmetic mean (twice the semi-variogram value). Clearly they are at their narrowest where the arithmetic mean of the $[G(x) - G(x+h)]^2$ distribution is most robust.

FIGURE 36 : MINIMUM/MAXIMUM ENVELOPE FROM SEMI-VARIOGRAM
AFTER RANDOM REMOVAL OF SAMPLES - SIMULATION 2637

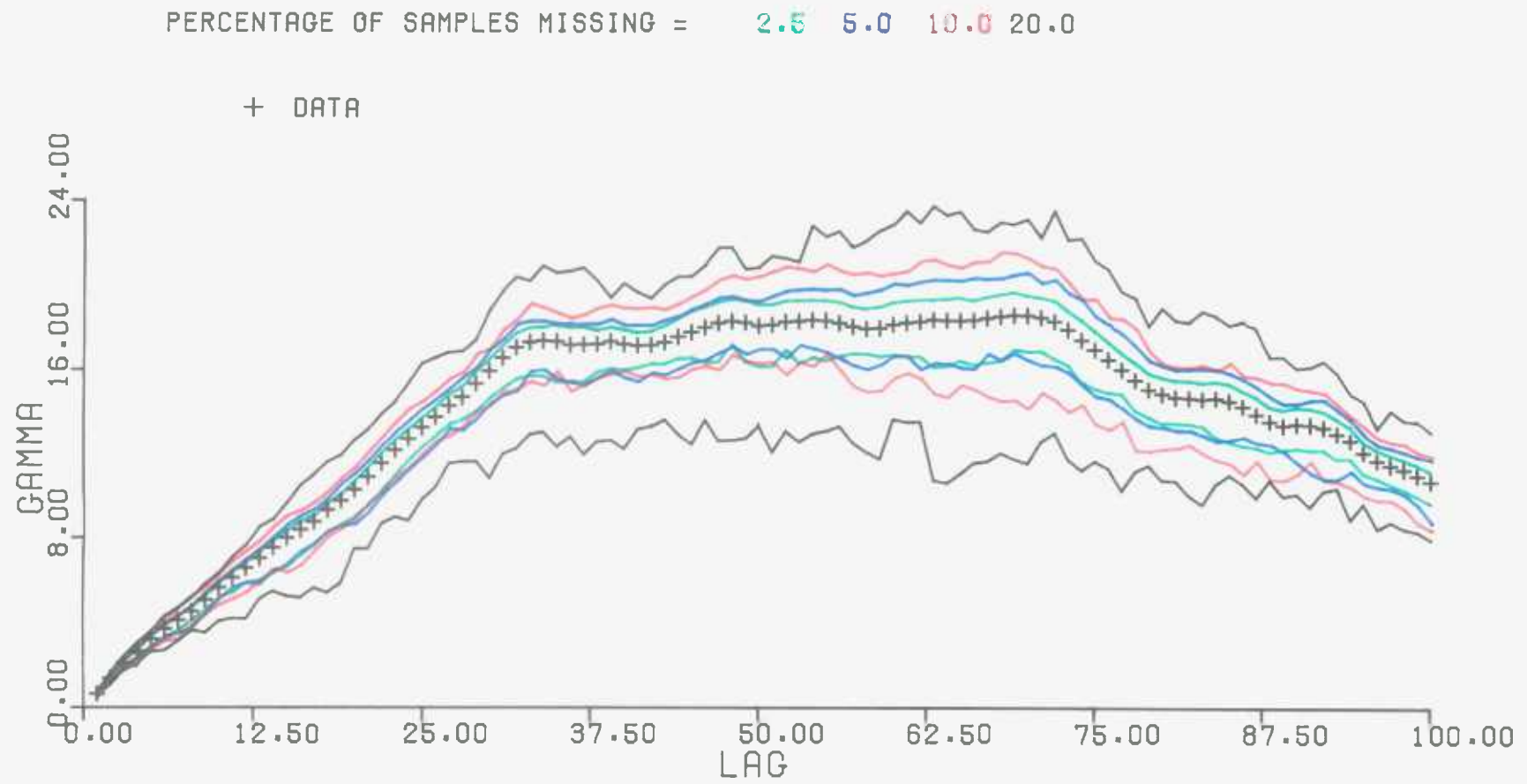
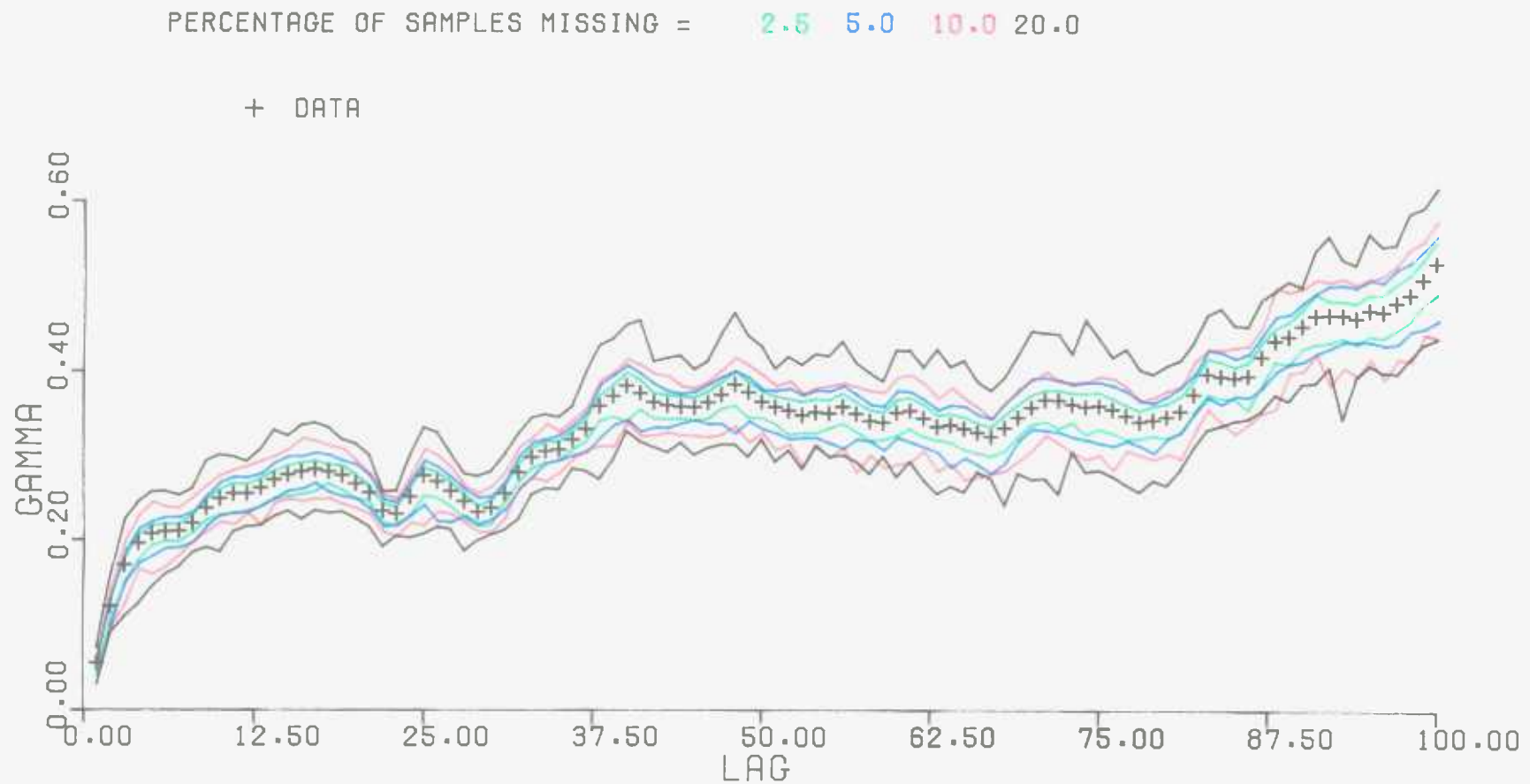


FIGURE 37 : MINIMUM/MAXIMUM ENVELOPE FROM SEMI-VARIOGRAM
AFTER RANDOM REMOVAL OF SAMPLES - LODE WIDTHS

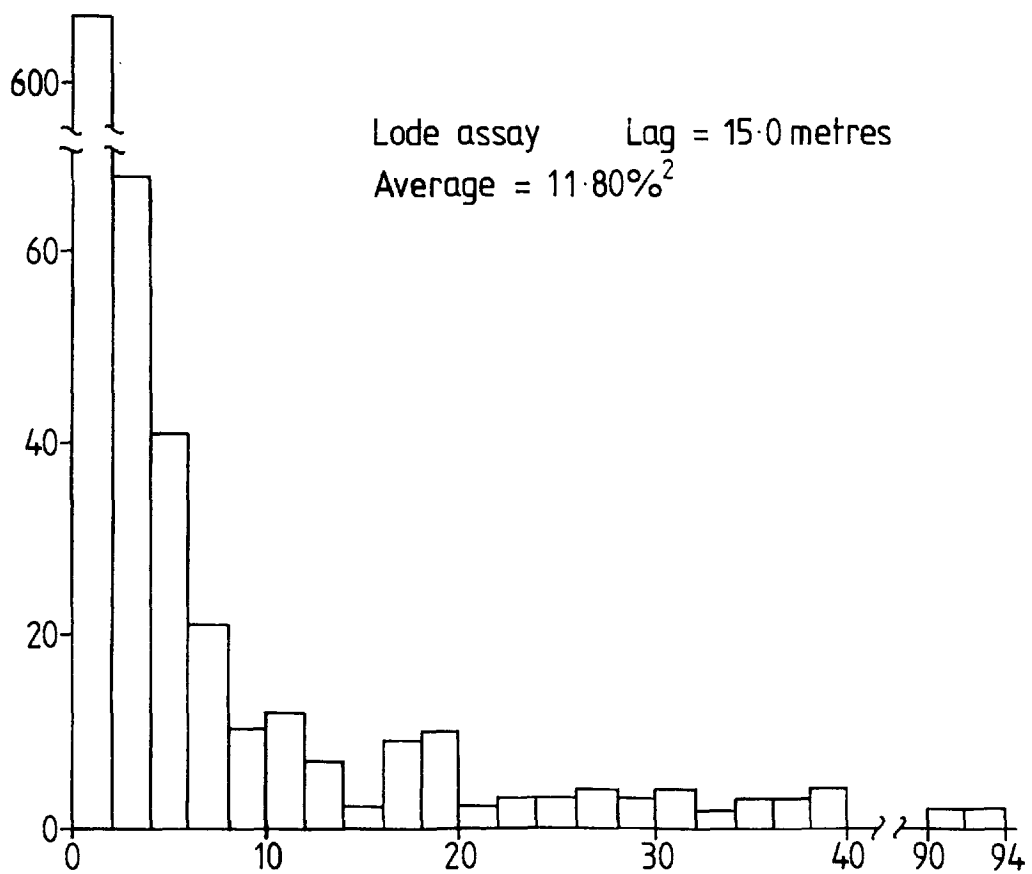
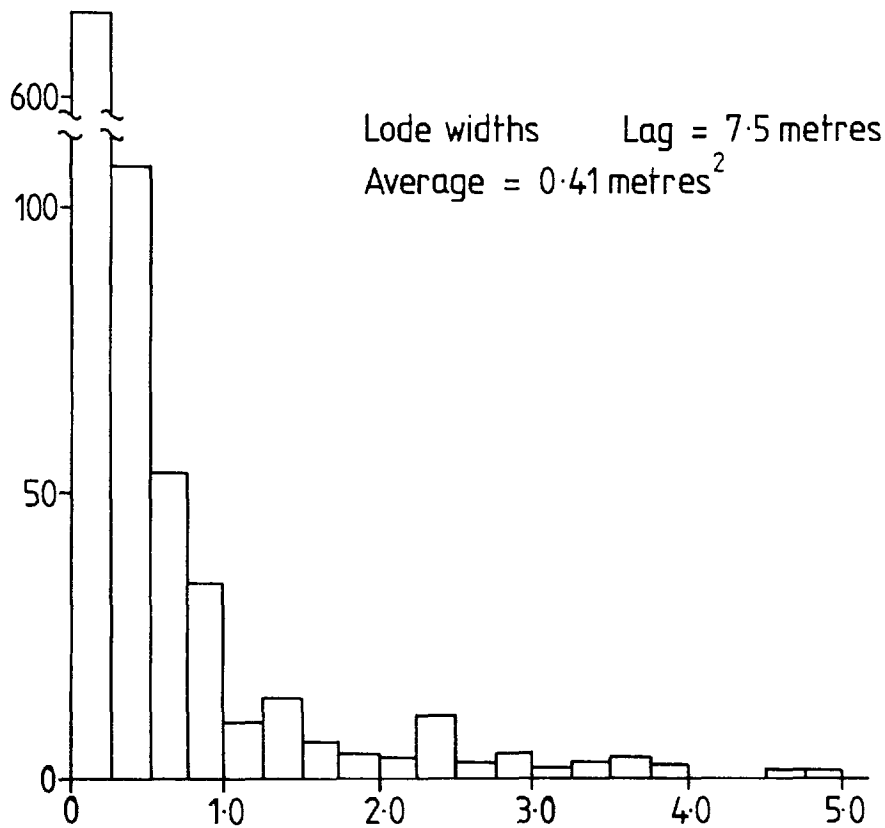


This is where the distribution spread is smallest. The robustness, or resistance to change, of a semi-variogram value is proportional to the $[G(x)-G(x+h)]^2$ distribution spread, which itself is proportional to the minimum/maximum envelope width. The robustness of any semi-variogram could be found by examining the spread of the $[G(x)-G(x+h)]^2$ distribution at every lag value. Clearly it is easier to estimate the robustness by plotting the minimum/maximum envelope.

It was decided that the robustness of a semi-variogram should be estimated by drawing the minimum/maximum envelope. This would be defined by the minimum and maximum values after fifty removals of ten percent of the data.

As stated above, every semi-variogram value is determined as half the arithmetic mean of the $[G(x)-G(x+h)]^2$ distribution. Examples of such a distribution are shown in figure 38. Respectively these are for lode widths at a lag of 7.5 metres, and for lode assays at a lag of 15.0 metres. Both of them have a highly skewed histogram. David, Dagbert, and Belisle (1977) find $[G(x)-G(x+h)]^2$ distributions of a similar shape, and recommend 'cleaning' each one before analysis. This involves trimming the distribution by removing its highest values, which obviously has the effect of greatly reducing the arithmetic mean. The precise reductions achieved are

FIGURE 38 : HISTOGRAM OF $[G(x)-G(x+h)]^2$ DISTRIBUTIONS



shown in Table 4. Also indicated is the effect a similar procedure has on some semi-variogram values from No.9 lode.

It was decided to test the influence of trimming the $[G(x)-G(x+h)]^2$ distribution of all the points of a semi-variogram. Taking a general view of the figures in Table 4, any member of a distribution was removed if it was greater than ten times the mean. If there were many such values then only the top one percent was extracted. In other words, the $[G(x)-G(x+h)]^2$ term of a pair of samples was not included if it was in the top one percent, and was ten times greater than the arithmetic mean of its distribution.

The two semi-variogram plots, before and after trimming of the $[G(x)-G(x+h)]^2$ distribution, are shown in figure 39 for lode widths and lode assays. The effect of trimming the width semi-variogram is small. However, smoothing of the plot does occur with the assay semi-variogram. It is helpful in determination of the range, and analysis of the semi-variogram as a spherical model. Very similar results to those of figure 39 were obtained with the semi-variogram of the natural logarithms (ln) of the lode widths, and of ln lode assays.

It seems that trimming a semi-variogram can make its underlying features more visible. However, to proceed with analysis using the trimmed semi-variogram, as advocated by David, Dagbert, and Belisle (1977), does not have any obvious justification.

TABLE 4

David + Dagbert + Belisle						
2 EW	0.4% of the pairs > 12*mean. Removal reduces mean by 19%					
3 EW	1.4%	"	7	"	"	26%
1 NS	1.6%	"	11	"	"	50%
2 NS	1.0%	"	12	"	"	34%
3 NS	1.0%	"	18	"	"	55%

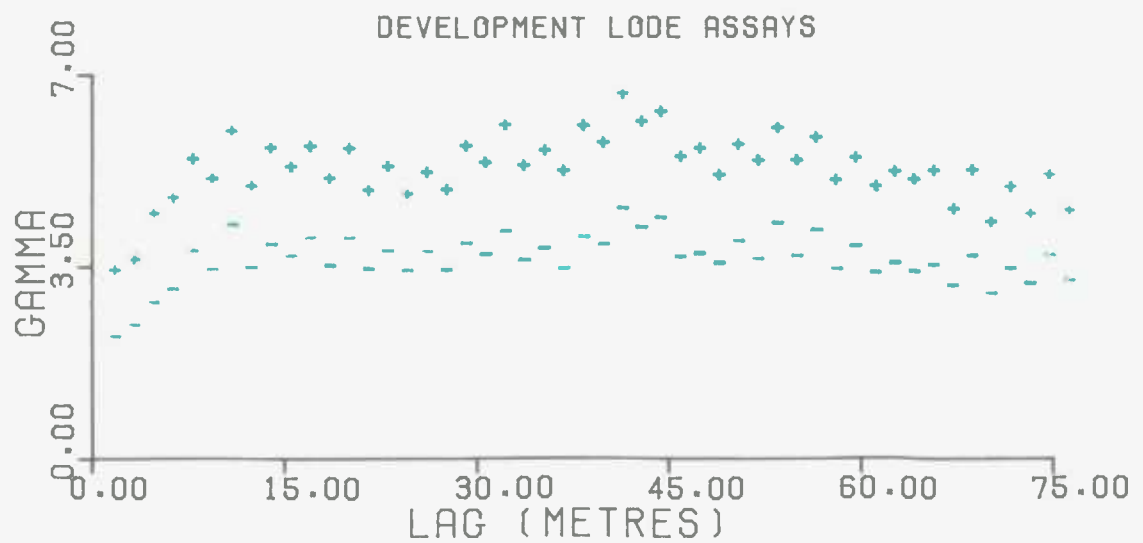
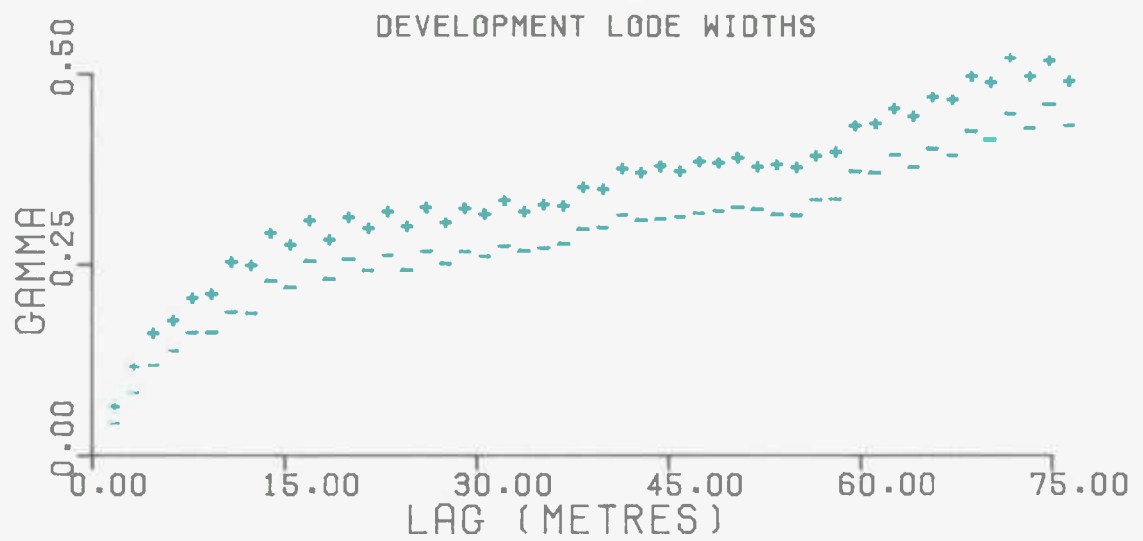
No.9 lode development						
Lode widths lag = 7.5m.	1.0% of the pairs > 12*mean. Removal reduces mean by 12%					
Ln lode widths lag = 7.5m.	0.9%	"	10	"	"	12%
Lode assays lag = 15.0m.	2.5%	"	8	"	"	19%
Ln lode assays lag = 15.0m.	0.3%	"	9	"	"	3%

FIGURE 39 : EFFECT OF TRIMMING

THE SEMI-VARIOGRAM VALUES

+ BEFORE TRIMMING

- AFTER TRIMMING



B.2.5. Preparation of data for simulation :-

It was desired to present a realistic model of No.9 lode. To do this, simulations of lode widths and assays had to be produced.

It has been shown how a simulation can be made of a variable which follows a Normal distribution and a spherical semi-variogram (see sections A.2 and A.3). The values from such a simulation are stationary (Journel, 1975).

Therefore any variable can be simulated, so long as it exhibits a Normal distribution, a spherical semi-variogram, and is stationary. In practice a problem almost invariably arises because the data variable to be simulated does not satisfy these three requirements. To do so, the data has to be altered in a predictable and reasonable manner. Many techniques have been developed to perform this alteration. These include taking natural logarithms, using transfer functions, and removing a trend. Obviously, if such a technique has been used, the simulation is not of the required variable. To achieve this, the simulated values must be passed through the alteration technique in the reverse direction.

B.2.6. Lode widths simulation :-

The lode width measurements from all the samples within No.9 lode were examined. The sampling method used was the same for all the samples. Therefore, there should

not be any bias between the stope and development sample sets. The stope lode width samples should have the same origins (as expressed by the auto-correlation and the distribution) as the development samples.

However, the stopes are not randomly positioned, but are sited in areas of high grade. Since widths and grades are independent (see section B.2.3), this in itself should not mean that the stope and development lode widths exhibit different characteristics.

However, the selection of a block as a stope was not merely based on assay results. It indirectly involved the lode widths.

A block was considered to be worth mining if the call assay value was greater than the economic cut off grade. The call width was defined (see section B.1.2) as the greater of 1.0 metre or the lode width plus 0.2 metres. Consider three samples with the same lode assay value but different widths. In each case the surrounding country rock has no tin content. The call assays are found as follows :-

	Sample A	Sample B	Sample C
Lode width	0.5	1.0	1.5
Lode assay	6.0	6.0	6.0
Call accumulation	3.0	6.0	9.0
Call width	1.0	1.2	1.7
Call assay	3.0	5.0	5.3

So, three samples with identical lode assays can have very different call assays. This may make the difference between

being below, or above, the pay limit. For good mining reasons, this method tends to prefer high grades if they are associated with large widths. Thus the distribution of the stope lode widths has a bias towards higher widths.

Taylor (1966) notes that high lode assay values tend to correspond with high lode widths. He states that the reverse is not necessarily true. The tendency for the stopes to be placed where both the grades and widths are high, accounts for this. The high grade samples are likely to be stope samples, and so their corresponding widths tend to be high also.

It was desired to simulate the lode widths for any area within No.9 lode. To do this the shape of the lode width distribution and semi-variogram had to be determined. As shown above, the stope samples are biased towards regions of high lode width. The same is true for the samples from the raises and inter-level development, since they also are predominantly sited in high grade areas. So, to achieve an unbiased sample set of the lode widths, only the main development level samples should be used. The positions of these drives were determined by mining considerations, and were not influenced by any sampling results.

The first step in the analysis of the behaviour of the lode width was to test for anisotropy. This was carried out by finding the semi-variogram in several directions. Obviously this is practically impossible to do using the

development samples. So all the 749 stope samples were used since they are not arranged in parallel lines. Although it has been stated that the stope samples are a biased sample set, it was felt that this does not preclude them from exhibiting any anisotropies present in the lode. The semi-variograms of the stope lode widths in three directions are shown in figure 40. Defining the vertical direction as 0 degrees, the three semi-variograms are for the 30, 90, and 150 degrees directions. The search angle for each one is thirty degrees. In other words, they include any sample pair with a lag direction between 0 and 59, 60 and 119, and 120 and 179 degrees respectively. Similarly the lag distances had to be given a certain amount of leeway. The semi-variogram value at 10 metres includes all lag values between 9.51 and 10.50 metres. There are no significant differences between the three semi-variogram plots. So, the lode width population of No.9 lode was considered to be isotropic in behaviour.

Figure 41 shows the semi-variogram of all the 1149 development lode width measurements. Also displayed is the semi-variogram of each of the four main levels. The fifth level drive only has 23 samples along it, and so its semi-variogram values were considered to be unreliable on their own. The four semi-variograms have similar shapes. However, these shapes have different vertical scales along the gamma axis. This is known as a proportional effect, and is often the result of a lognormal distribution

FIGURE 40 : SEMI-VARIOGRAMS IN DIFFERENT DIRECTIONS
FOR STOPE LODE WIDTHS

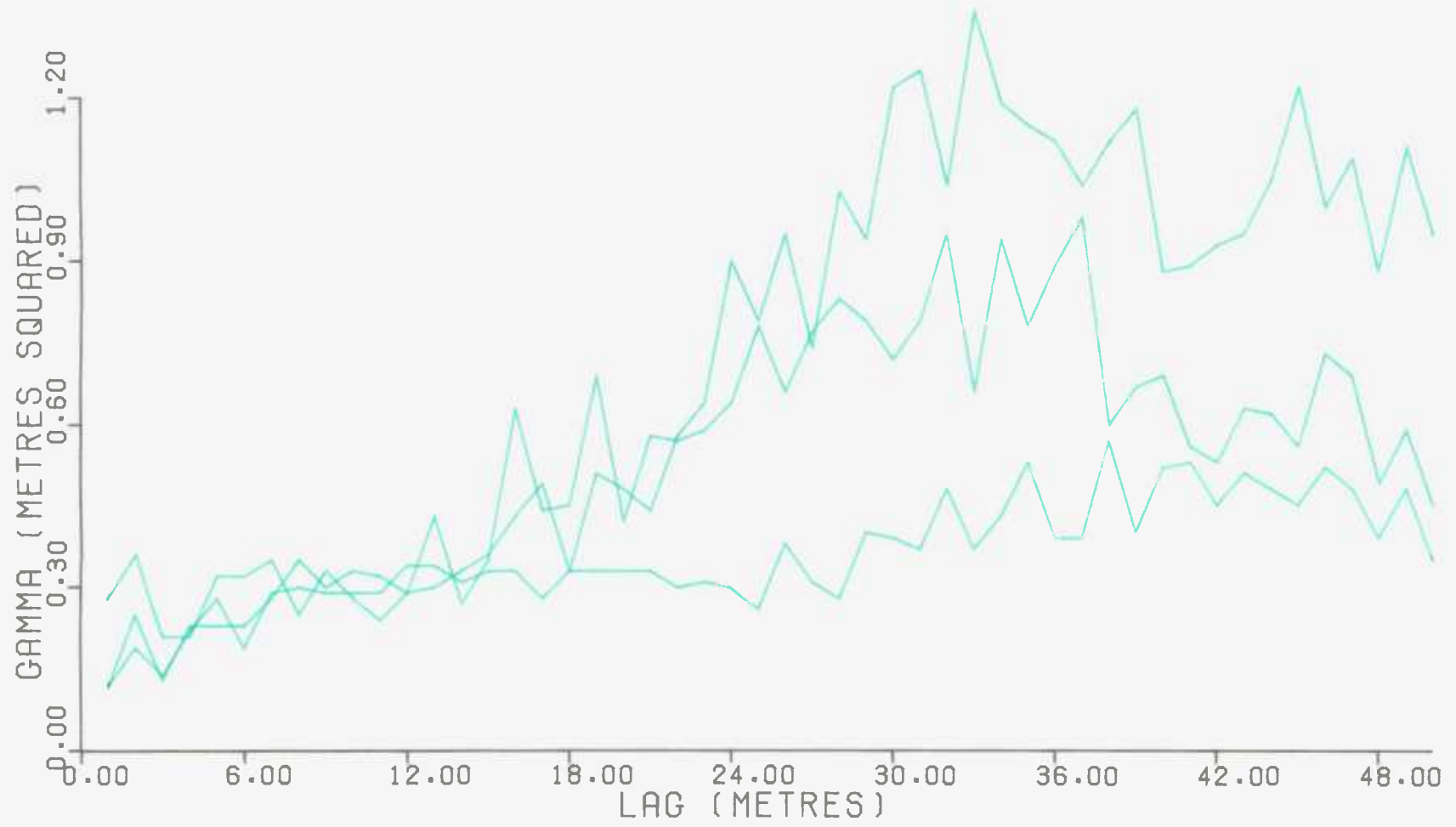
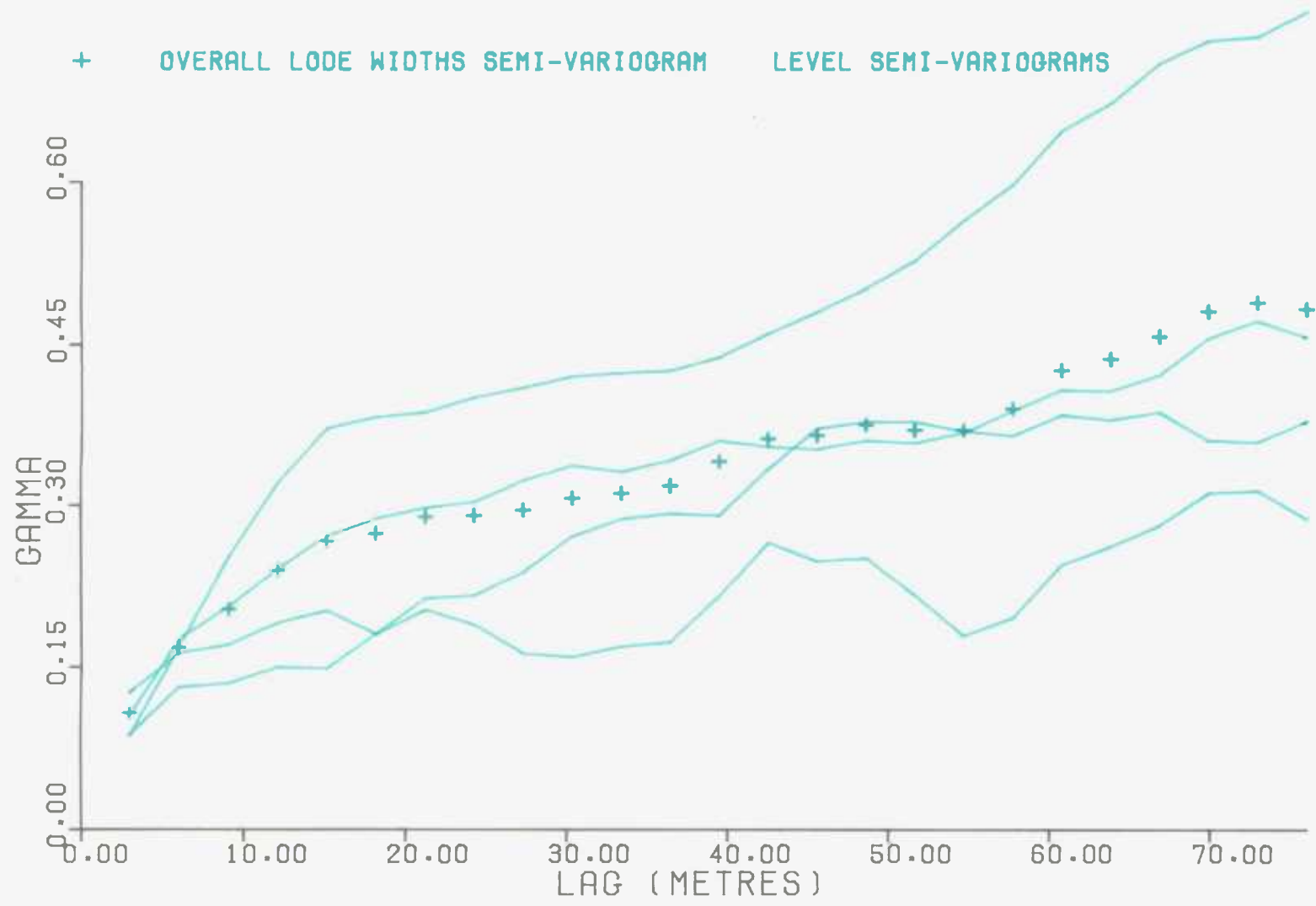


FIGURE 41 : SEMI-VARIOGRAMS FROM DIFFERENT LEVELS



(Clark,1979). This is because with a lognormal distribution the variance is proportional to the mean.

The histogram for all the imperial development lode width measurements is shown in figure 42. It confirms that the distribution is of a typical lognormal type, that is with a heavy positive skew.

The behaviour of the ln lode widths figures was examined. Figure 43 displays the four main level semi-variograms, and the overall semi-variogram of ln lode widths. The semi-variograms of the four levels show that the proportional effect has been accounted for by using the logarithms. However, the overall semi-variogram does not exhibit the shape of any semi-variogram model. It has a spherical form at low lag values but does not level out at a sill. Instead it keeps on rising, in a parabolic curve. This is symptomatic of a polynomial trend in the samples (Clark,1979; David,1977). When this situation exists the samples are not stationary. The trend has to be removed and the analysis carried out on the residuals (Journel,1975).

Another approach to the problem of applying geostatistics when a trend exists, is that of universal kriging (Huijbregts and Matheron,1971). In concept this is a fairly straightforward procedure entailing expansion of the usual kriging system of equations. However, it is very heavy on computational time (David,1977). There exists no theoretical or practical explanation of how geostatistical simulation can be carried out when universal kriging is

FIGURE 42 : HISTOGRAM OF IMPERIAL DEVELOPMENT LODGE WIDTHS

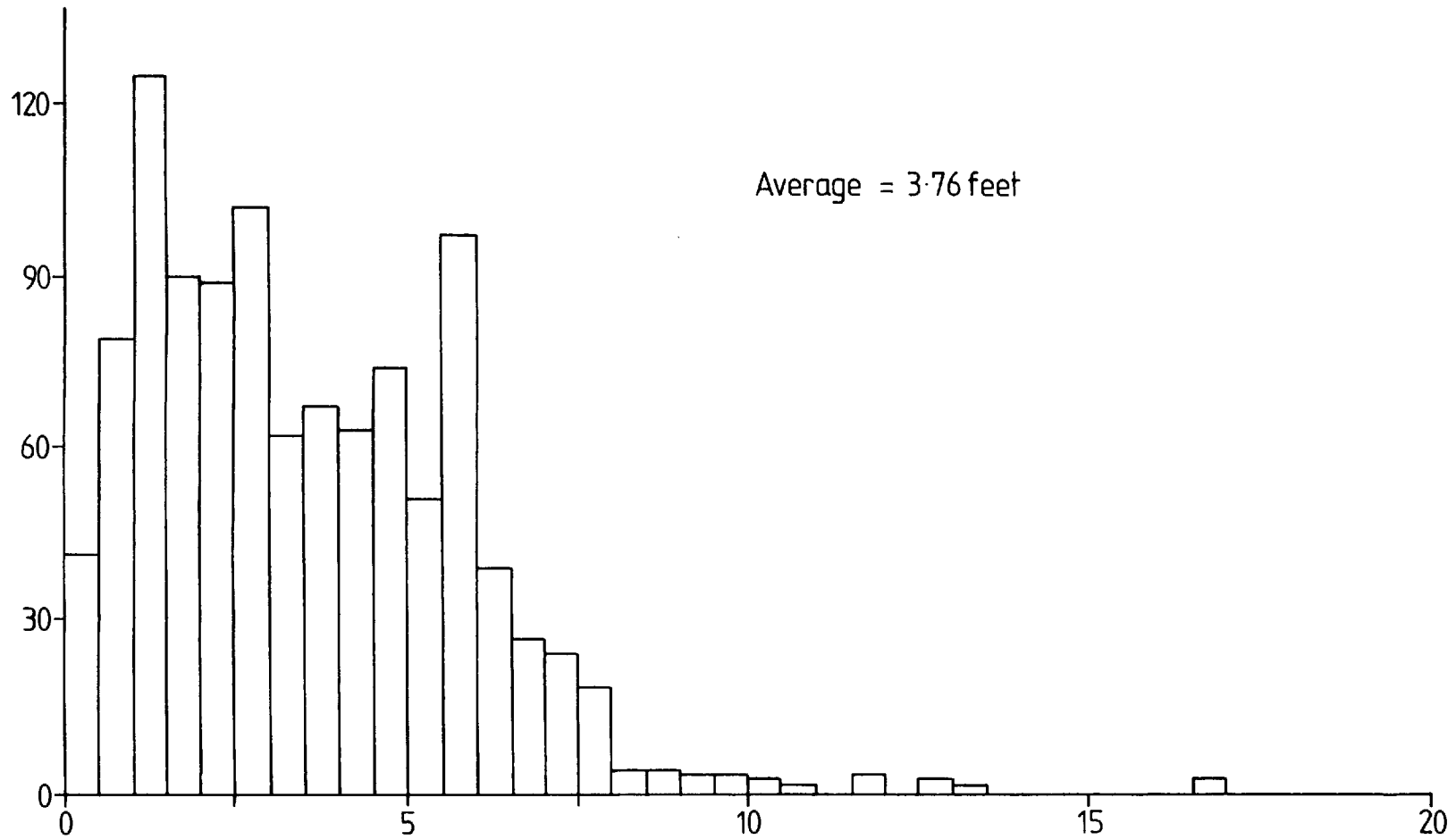
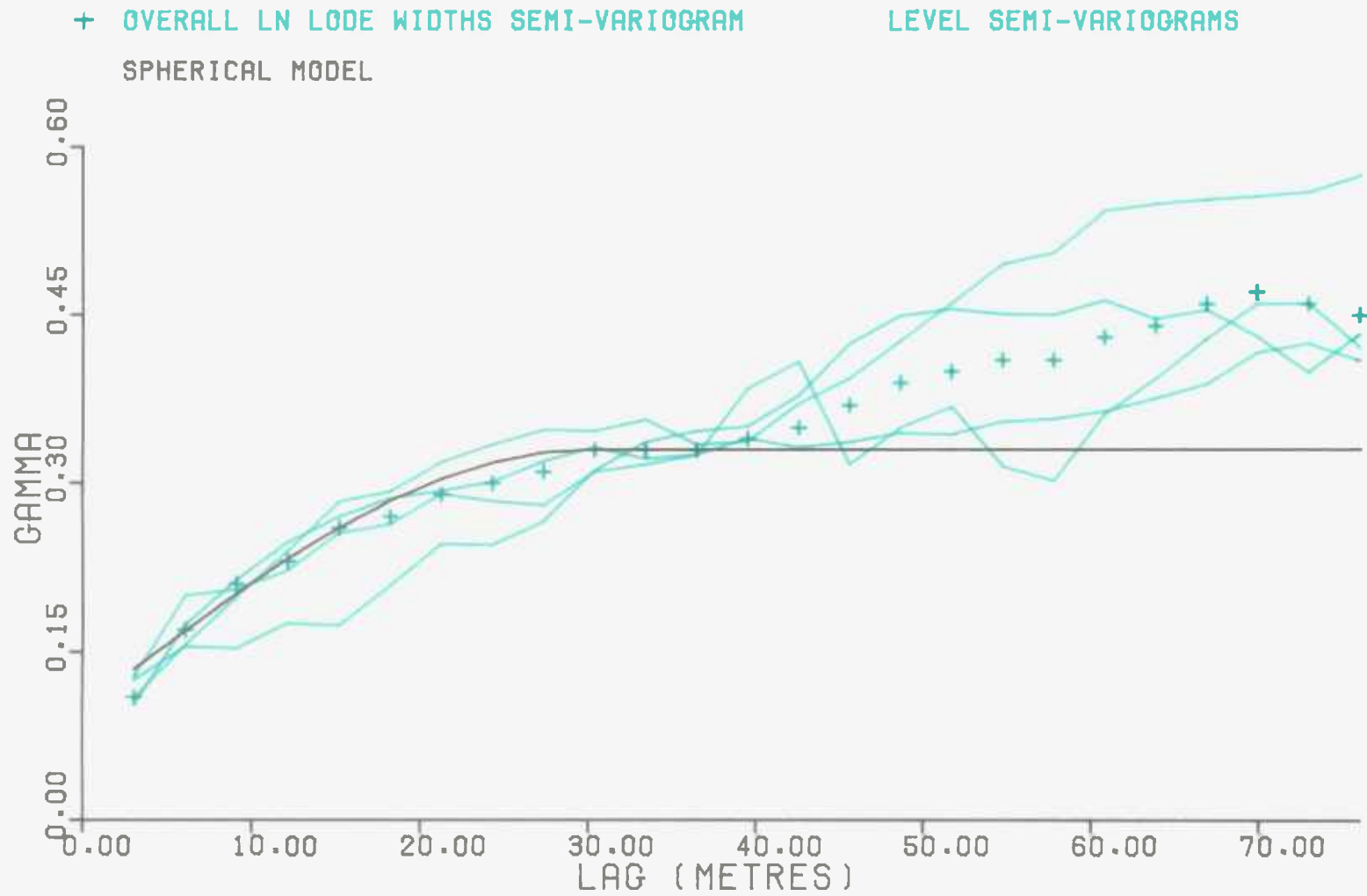


FIGURE 43 : SEMI-VARIOGRAMS FROM DIFFERENT LEVELS



used. After a thorough examination of the theory behind universal kriging, no acceptable simulation method seemed possible. For this reason, and to avoid a sharp increase in cost, universal kriging was not employed.

The trend of the ln lode width values from the development samples was estimated by polynomial equations. A computer program, called SNARK (Clark, 1977), was adapted and used to produce the 'least squares best fit' linear, quadratic, and cubic equations. The results of SNARK (see table 5) show that the cubic polynomial equation is significantly closer to the data than the other two equations. It was felt that to find an equation of a higher order than three would not be worthwhile. Whitten (1973) states that 'with equations of degree greater than three or four, wild extremae appear in the computed surfaces'. Table 5 also records that the cubic trend equation explains 32 percent of the total sum of squares. This term was defined as :-

$$= 100.0 * \frac{(TSS - RSS)}{TSS}$$

where TSS = Total sum of squares

$$= \sum_{i=1}^{i=N} R^2 - (N * \bar{R}^2)$$

TABLE 5

X = Easting Y = Height above datum

N.B. E+02 means $\times 10^2$

SOLUTION OF LINEAR EQUATION IS :

$$Z = 1.4841E+00 + -9.7138E-04*X + -9.3974E-04*Y$$

SOLUTION OF QUADRATIC EQUATION IS :

$$Z = -4.4641E+02 + 9.2733E-02*X + 5.7388E-01*Y \\ + 6.0615E-05*X*X + 5.9785E-05*X*Y + 1.8431E-04*Y*Y$$

SOLUTION OF CUBIC EQUATION IS :

$$Z = -2.2797E+02 + 1.9471E+00*X + 4.8560E-02*Y \\ + -3.3145E-04*X*X + -2.4203E-03*X*Y + 2.2078E-04*Y*Y \\ + 2.0525E-08*X*X*X + -1.0132E-07*Y*Y*Y \\ + 2.0559E-07*X*X*Y + 7.5072E-07*X*Y*Y$$

ANALYSIS OF VARIANCE TABLE

SOURCE	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
LINEAR	3.3295E+01	2	1.6648E+01	3.5826E+01
RESIDUAL	5.3253E+02	1146	4.6468E-01	
QUADRATIC	1.4038E+02	5	2.8076E+01	7.5429E+01
DIFF.	1.0708E+02	3	3.5695E+01	9.5898E+01
RESIDUAL	4.2544E+02	1143	3.7221E-01	
CUBIC	1.8115E+02	9	2.0128E+01	5.9598E+01
DIFF.	4.0772E+01	4	1.0193E+01	3.0181E+01
RESIDUAL	3.8467E+02	1139	3.3773E-01	
TOTAL	5.6582E+02	1149		

PERCENTAGE OF TOTAL SUM OF SQUARES

LINEAR COMPONENT 5.88
 QUADRATIC COMPONENT 24.81
 CUBIC COMPONENT 32.02

RSS = residual sum of squares

$$= \sum_{i=1}^{i=N} (R-T)^2$$

N = Number of data values

R = Real data value

\bar{R} = Average real data value

N = Number of data values

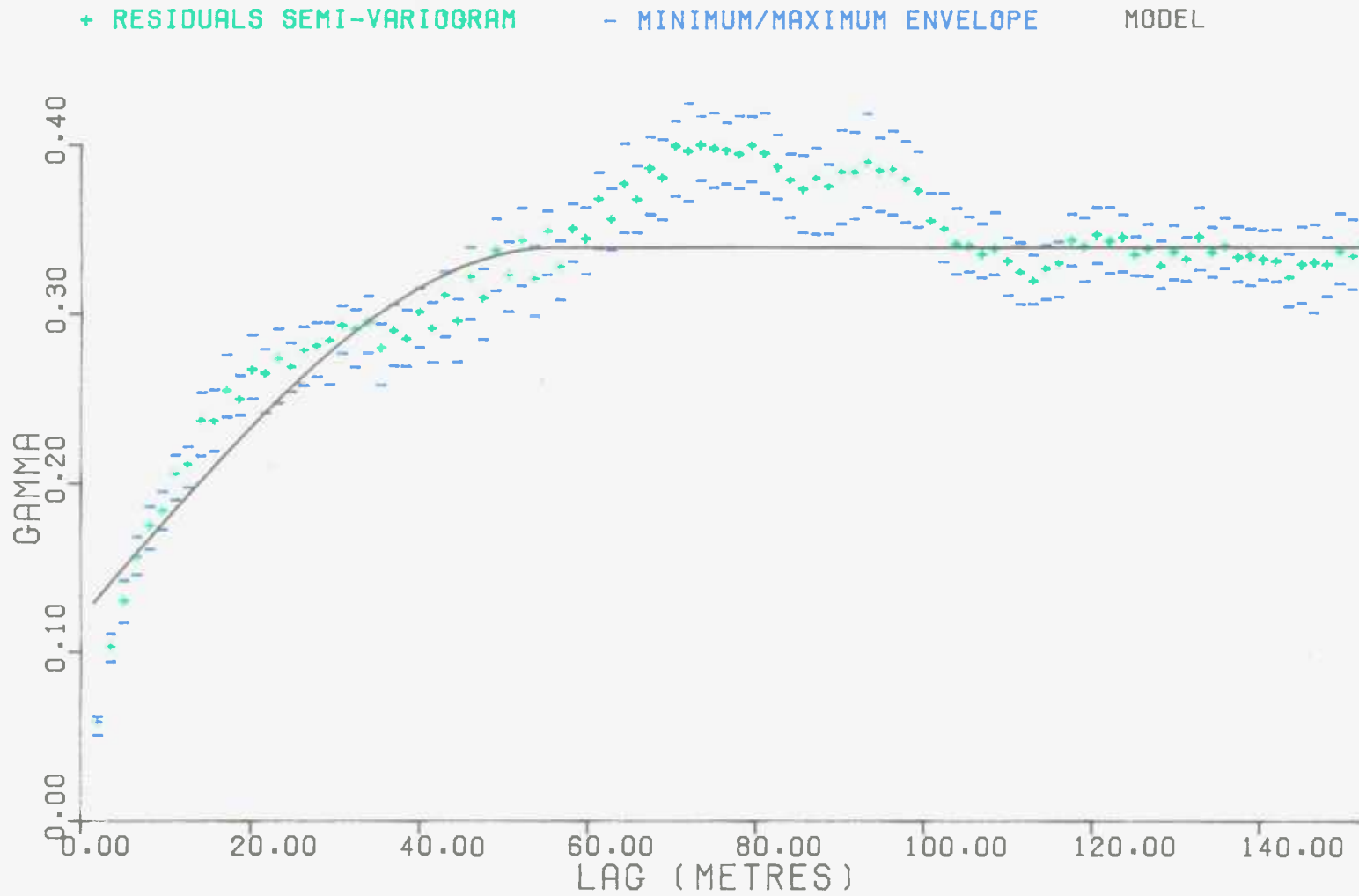
T = Trend predicted value

Harbaugh and Merriam (1968) state that if this is between 15 and 30 percent the goodness of fit of the trend is considered to be low.

The cubic trend equation value was calculated for every development sample position, and the residuals were examined. Figure 44 shows the semi-variogram of all the residuals, together with the fitted spherical model. Trimming the $[G(x)-G(x+h)]^2$ distributions of this semi-variogram has little effect on the shape of it. The minimum/maximum envelope, after fifty random removals of 10 percent of the data, is also displayed. It shows the semi-variogram to be fairly robust. The effect of the trend has been removed, and the semi-variogram oscillates gently around a definite sill. The model parameters were initially estimated visually and the RMS terms (see section A.2.2) up to lag values of 40 and 80 metres calculated. Each of the three parameters was then changed in turn until the model with the minimum RMS terms was found. For the residuals of

FIGURE 44 : SEMI-VARIOGRAM OF RESIDUALS OF DEVELOPMENT

LN LODE WIDTHS CUBIC TREND SURFACE



the ln lode width trend the closest fitting model was Spherical(56 metres,0.34,0.12).

The range of the residuals model semi-variogram is 56 metres, or 184 feet. Clark (1978) finds that the total range for lode widths at the geologically similar Geevor tin mine to be 150 feet. It is interesting to note that the residuals model semi-variogram has a nugget effect of 35 percent of the total sill. The ln lode widths semi-variogram has a negligible nugget effect. It seems that the continuity of the trend is the dominant factor at low lag values. This result has been achieved elsewhere, for instance in ground magnetism readings from the Steiermark region of Austria (Burger and Skala,1978).

It is obvious that a more exact fit could be made with two or more nested spherical models. However, it was felt that this is an over elaboration, of the same sort as using a trend equation of an order four or above. With nested schemes almost any semi-variogram can be fitted closely. Similarly a set of Normal distributions can fit most histograms. But it is the underlying characteristics which are required not the superficial random variations.

For the purposes of geostatistical estimation it has been demonstrated (Krige,1976) that the exact value of each of the semi-variogram parameters is not of paramount importance. David (1977) states that the effects of misinterpreting the sill and range of a semi-variogram are not very significant.

The histogram of the trend residuals is shown in figure 45. Fitting the Normal distribution of $N(0,0.34)$ to it produces a chi-squared statistic of 10.50, with 10 degrees of freedom. This shows the histogram to be significantly different from the model only at the 42 percent level. Since this is greater than 5 percent the distribution of $N(0,0.34)$ was accepted.

So, the prerequisites for simulation of the development lode widths are met. The measurements, after alteration by taking the natural logarithms and removing the trend, are stationary, fit a spherical semi-variogram model, and have a Normal distribution with a variance equal to the sill.

B.2.7. Lode assays simulation :-

The results from all the sample assays of lode value were obtained. The sampling method was constant for all the samples. The assay technique varied between vanning and the P.I.F. analyser, but purely on the basis of when the sample was taken.

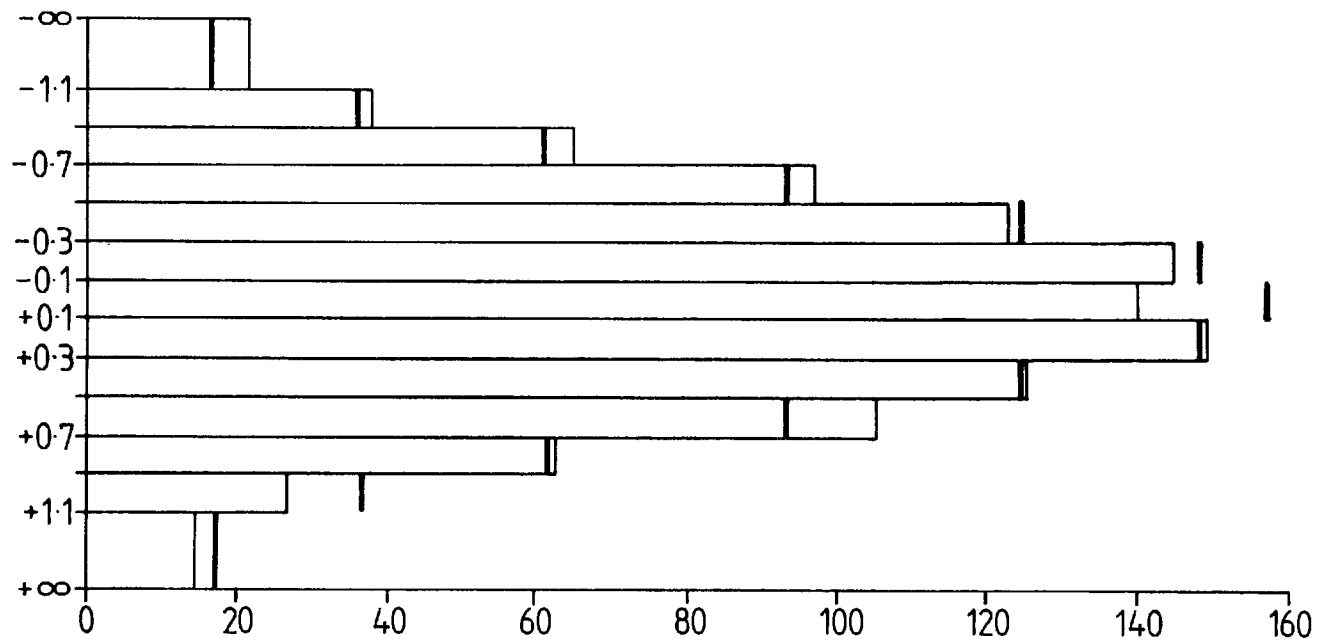
The stopes, the raises, and the inter-level development obviously occur mainly in the high grade areas. Therefore their samples give a biased view of the total lode assay population of No.9 lode. To obtain an unbiased sample set only the main development level sample results can be used.

The exception to this is in the determination of

FIGURE 45 : HISTOGRAM OF RESIDUALS OF LN LODE WIDTHS CUBIC TREND SURFACE PLUS FITTED MODEL

Average = -0.022

Model |



anisotropy. For this purpose the stope samples were considered to be acceptable, and much more useful. Figure 46 shows the lode assay semi-variogram calculated in the same three directions as before (see section B.2.6). There seem to be no great differences. Thus No.9 lode was regarded as being isotropic in terms of its grade.

The semi-variogram was calculated for all the development sample assays of lode value. Figure 47 displays the result along with the semi-variogram from each of the four main development levels. These semi-variograms differ significantly in their sills, thus exhibiting a proportional effect.

The histogram of the development lode assay values, as shown in figure 48, displays a great positive skewedness. So, a lognormal distribution was felt to be the probable cause of the proportional effect.

The semi-variograms of the ln lode assay terms (see figure 49) confirm that the proportional effect has been overcome. The overall semi-variogram gives no indication of a trend in the values. To check this the computer program SNARK was used to find the best fitting cubic equation. This only accounts for 14 percent of the total sum of squares, and therefore was ignored.

Displayed in figure 50 is the overall semi-variogram of ln lode assay values. Also shown is the minimum/maximum envelope after fifty random removals of 10 percent of the samples. Trimming the $[G(x)-G(x+h)]^2$ distributions has

FIGURE 46 : SEMI-VARIOGRAMS IN DIFFERENT DIRECTIONS
FOR STOPE LODE ASSAYS

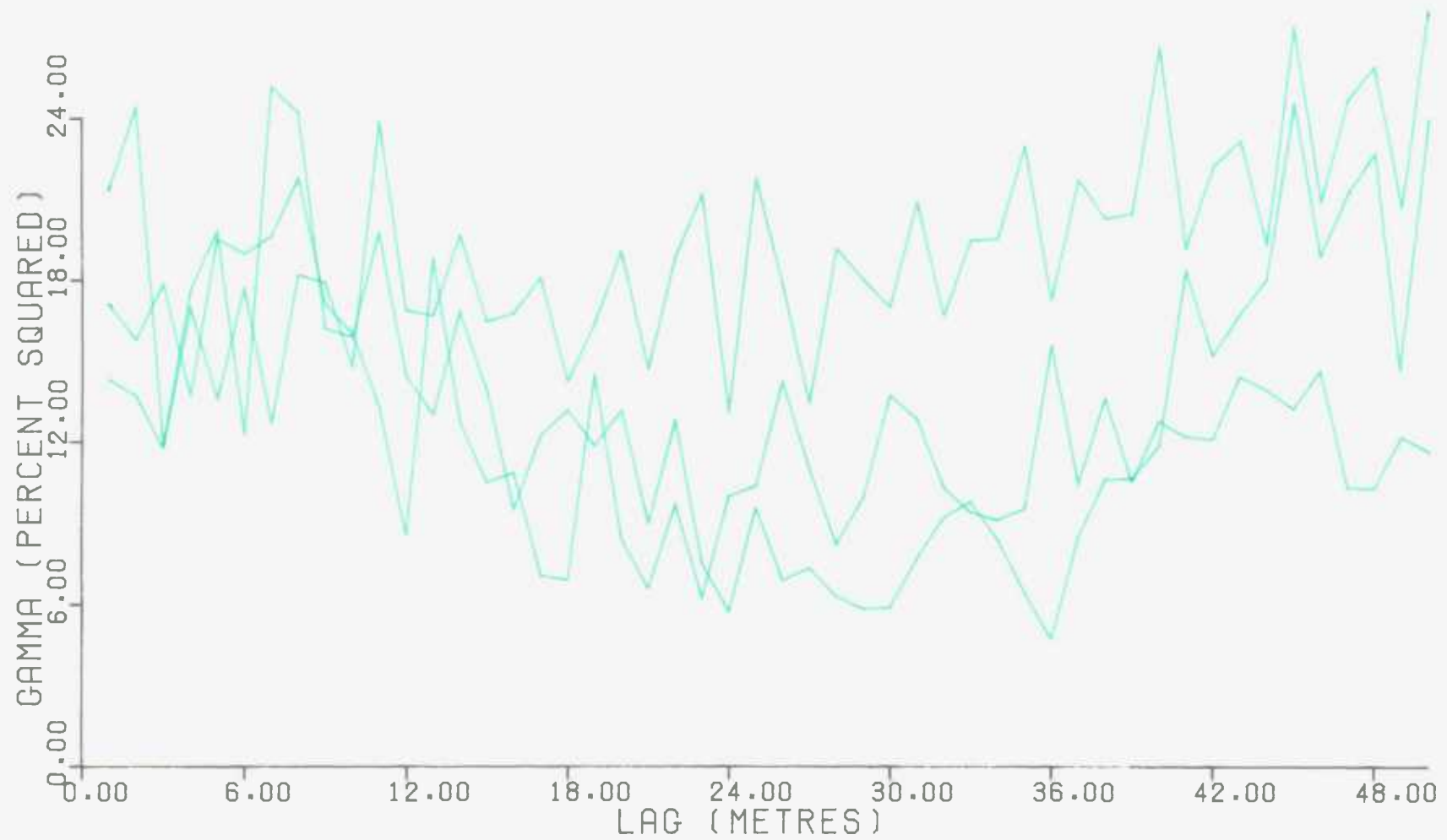


FIGURE 47 : SEMI-VARIOGRAMS FROM DIFFERENT LEVELS

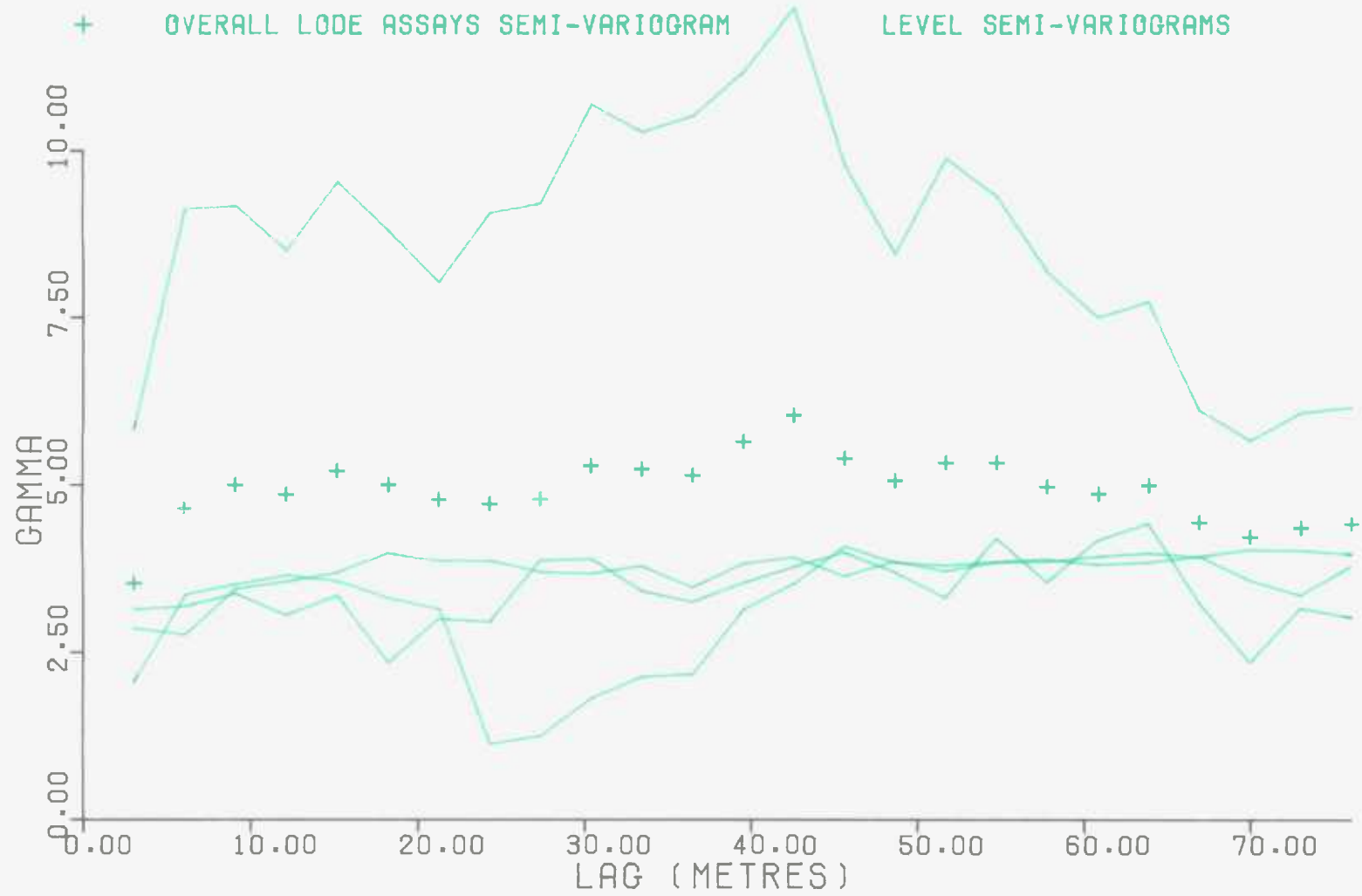


FIGURE 48 : HISTOGRAM OF LODE VALUES PLUS FITTED MODEL

Average = 0.91%

Model |

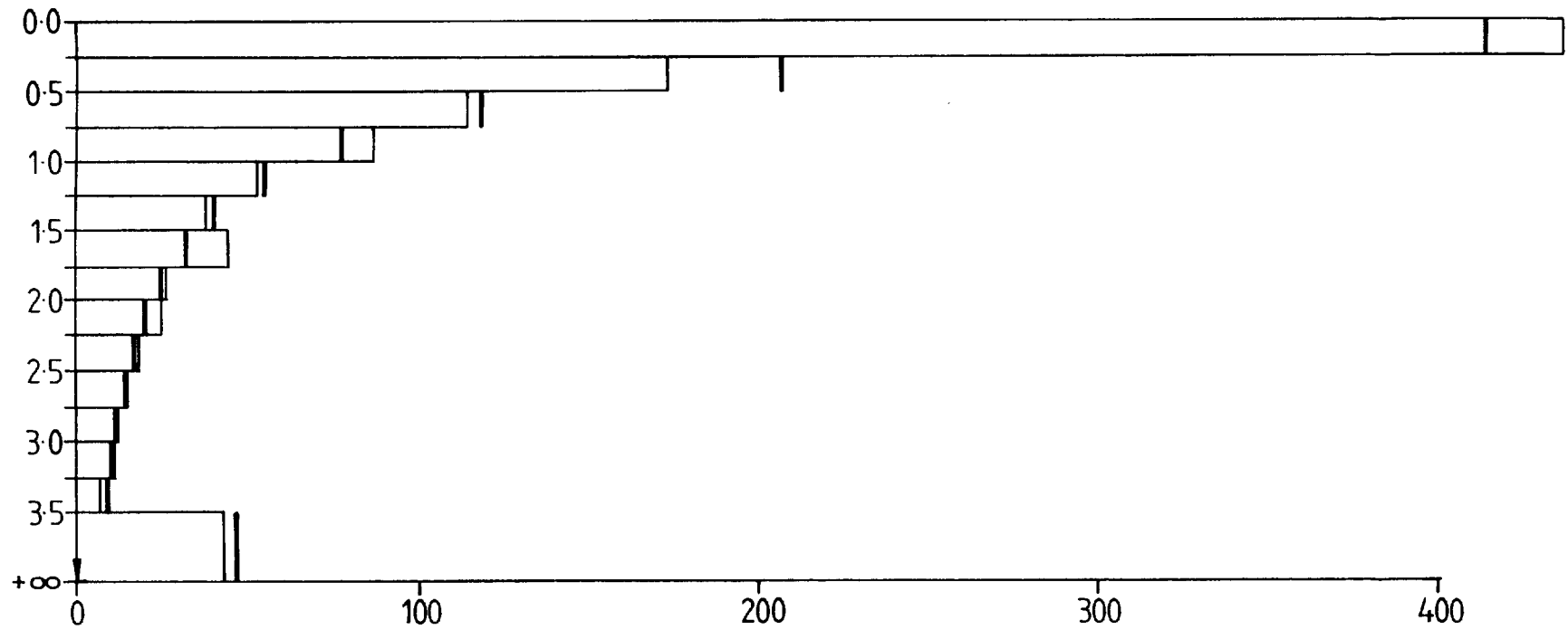


FIGURE 49 : SEMI-VARIOGRAMS FROM DIFFERENT LEVELS

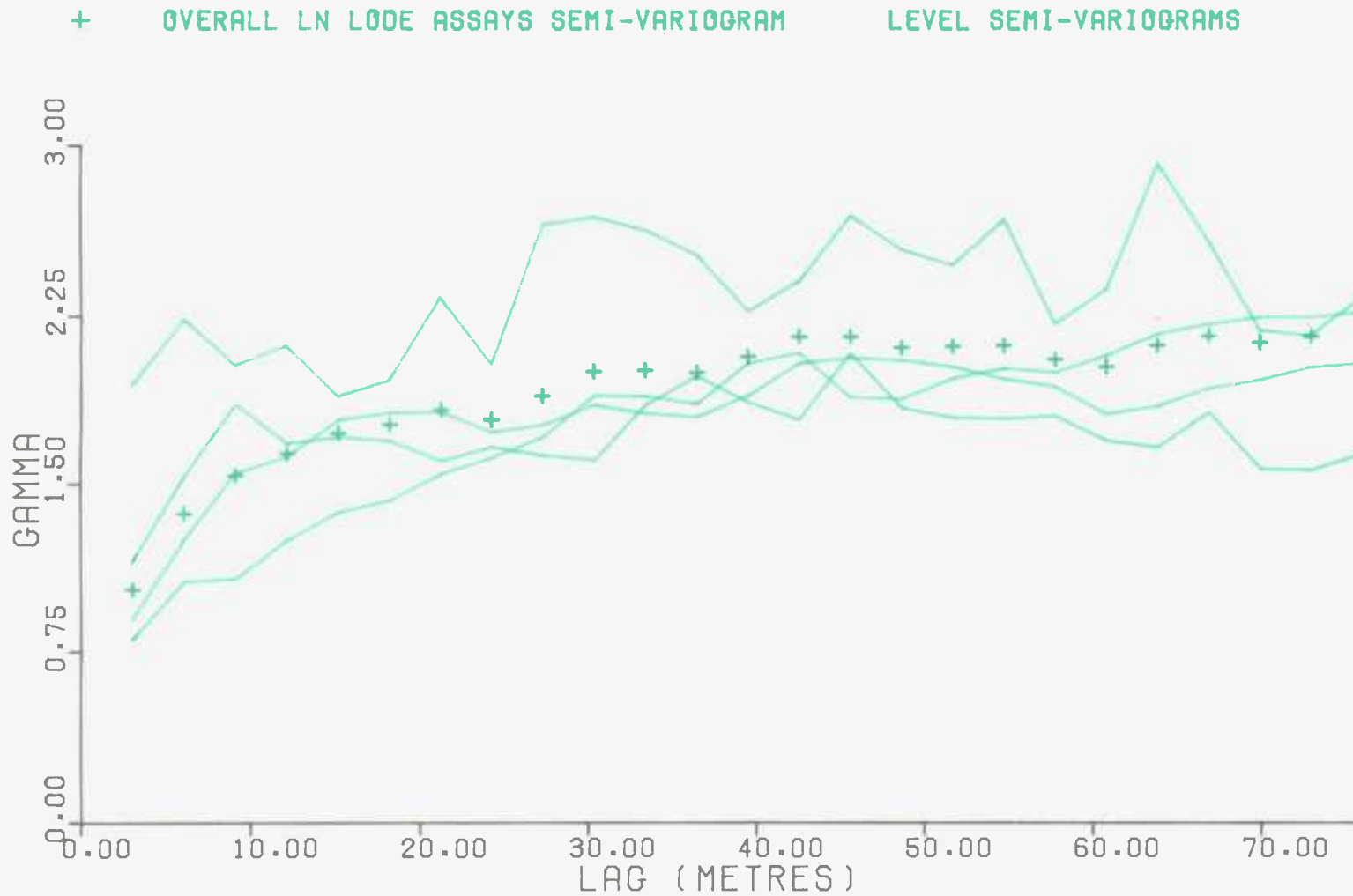
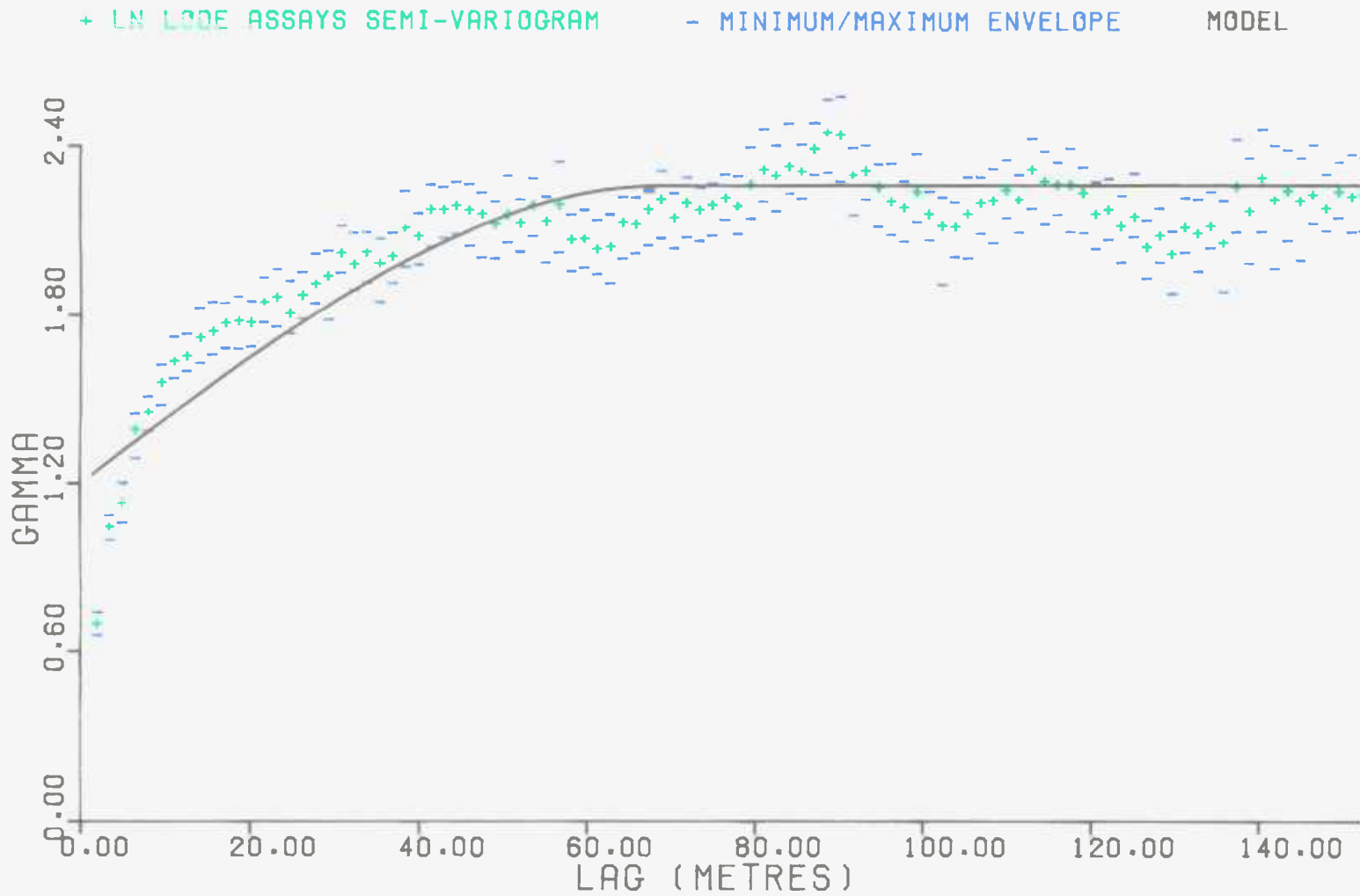


FIGURE 50 : SEMI-VARIOGRAM OF DEVELOPMENT LN LODE ASSAYS



a very similar effect to that shown in figure 39 for the lode assay values. The trimmed semi-variogram helped in the initial estimation of the semi-variogram model parameters. In turn each of these three model parameters was altered to minimise the 40 and 80 metres RMS terms. The final fitted model, as shown in figure 50, is Spherical(69 metres,2.26,1.20).

The range of the ln lode assays semi-variogram is 69 metres, or 226 feet. This is similar to the 175 feet determined by Clark (1978) to be the total range of ln lode assays at the nearby Geevor mine.

The histogram of the lode assay values is shown in figure 48. It also shows a fitted lognormal distribution with a mean of 1.33 and a variance of 3.9. The visual impression of a close fit is supported by the chi-squared statistic of 16.38, which (with 12 degrees of freedom) is only 18 percent significant. So, the histogram of lode assays can be considered to be lognormal. This is equivalent to the histogram of the ln lode assays following a Normal distribution. Converting the parameters of the lognormal model gives a Normal distribution of $N(-0.85,2.26)$ for the ln lode assay figures. So, the variance of the ln lode assays (2.26) is equal to the sill of their semi-variogram model.

The prerequisites for simulation are met. The ln lode assay values are stationary, fit a spherical semi-variogram model, and have a Normal distribution with a variance equal

to the sill.

The histogram of the lode assay values, as shown in figure 48, uses the rounded figures (see section B.2.1). That is the measurements were rounded to the nearest 2 lbs/ton or 0.05 percent, before inclusion. The histogram is fitted by a lognormal distribution with a mean of 1.33 and a variance of 3.9. It is interesting to note that this distribution does not fit the histogram of the raw, unrounded, lode assays. Indeed no lognormal distribution could be found which was not significantly different from the histogram at the 2.5 percent level. This was due to the peaked nature of the histogram.

B.2.8. Actual widths estimation :-

It was desired to find the relationship between the lode width and actual width measurements.

As expected, a scattergram from all 749 stope samples shows the lode and actual widths to be highly correlated with a correlation coefficient of 0.75.

The best method for predicting the actual width from the lode width was required. As stated before (see section B.1.2), a call width was calculated at every sample point. This is an estimate of what the actual width of a stope would be at that position. For each of the stope samples there exists an actual width measurement which can be compared with the estimate. Calculation of the residual sum of squares (RSS; see section B.2.6) produces a measure of

'goodness of fit' for the estimation method.

The actual width estimation technique practised at South Crofty mine is to take the greater of 1.0 metres or the lode width plus 0.2 metres. Testing all 749 of these estimates against the measured value produces a RSS term of 246.4. The total sum of squares (TSS; see section B.2.6) of the stope actual widths was calculated to be 343.8. Therefore this estimation method explains 28 percent of the data TSS.

Other estimation techniques of a similar nature were examined. The minimum RSS value (139.9) is obtained if the actual width estimate is the greater of 1.5 metres or the lode width plus 0.4 metres.

Polynomial equations of the form :-

$$\text{Actual width estimate} = f(\text{LW}, \text{LW}^2, \dots, \text{LW}^N)$$

where LW = Lode width

N = order of the polynomial

were examined for their ability to predict the stope actual width values. Using the least squares criterion, the best fitting line was found for every order of polynomial between zero and nine. With the constraint of the actual width estimate having to be at least equal to the lode width, the RSS of each of these lines was calculated to be :-

Polynomial of order 0		RSS = 181.4
"	1	" 151.5
"	2	" 133.8
"	3	" 133.0
"	4	" 132.3
"	5	" 132.3
"	6	" 132.3
"	7	" 131.4
"	8	" 131.4
"	9	" 130.9

Obviously the polynomial of order nine is the best fit to the data. However, the law of diminishing returns applies and the difference between the RSS term from the equation of order nine and that of order eight is negligible. Division of one RSS term by another produces a 'F' statistic which can be used to test whether one estimation technique is significantly more accurate than another. Doing this produces the following 'F' statistics :-

$$\frac{\text{Order 0 (749 degrees of freedom)}}{\text{Order 1 (748 degrees of freedom)}} \quad F = 1.20$$

5 percent significantly worse

$$\frac{\text{Order 1 (748 degrees of freedom)}}{\text{Order 2 (747 degrees of freedom)}} \quad F = 1.13$$

5 percent significantly worse

$$\frac{\text{Order 2 (747 degrees of freedom)}}{\text{Order 3 (746 degrees of freedom)}} \quad F = 1.00$$

Not 5 percent significantly worse

Compared with the quadratic equation, the extra effort required to use the cubic polynomial does not produce significantly more accurate estimates. Even the polynomial of order nine is not a significantly better estimator (the 'F' statistic is 1.01 with 743 and 740 degrees of freedom). So, it was decided to make the actual width estimator a function of lode width and lode width squared.

It has been stated (see section B.2.6), that the ln lode width measurements exhibit a significant cubic trend surface. Bearing this in mind it was decided to test the actual width measurements for a trend. As before, the computer program SNARK was adapted to produce the 'least squares best fitting' linear, quadratic, and cubic trend surfaces. The cubic trend was discovered to be significantly more accurate than the others, and to explain 23.29 percent of the data TSS.

So, it was decided that the actual width estimate should not only be a function of lode width to order two, but also of Easting and depth to order three. Again using the least squares criterion, the best fit equation of this form was found to be :-

$$\begin{aligned} \text{Actual width estimate} = & \\ & 1.3100068 * 10^4 + 3.5826833 * 10^{-2} * LW \\ & + 1.6650438 * 10^{-1} * LW^2 - 1.5468835 * E \\ & + 9.7820590 * 10^{-5} * E^2 + 1.5368569 * 10^{-8} * E^3 \\ & - 2.5478477 * 10^1 * H + 1.6512423 * 10^{-2} * H^2 \\ & - 3.5656412 * 10^{-6} * H^3 + 2.0147847 * 10^{-3} * E * H \\ & - 6.7110721 * 10^{-8} * E^2 * H - 6.5576605 * 10^{-7} * E * H^2 \end{aligned}$$

where LW = Lode width

E = Easting

H = Height above datum

Testing this equation against the stope sample data shows it to have a RSS of 118.3. This is 5 percent significantly better than the polynomial equation of order nine. It is also 20 percent significantly more accurate than the least squares best fit line of the form :-

$$\text{Actual width estimate} = f(LW, LW^2, E, E^2, H, H^2)$$

This was found to have a RSS term of 125.2.

It was decided that for any position within No.9 lode the best estimate of stope width is obtained using the equation with the twelve terms specified above. This has a RSS term of 118.3, and so explains 66 percent of the data TSS.

B.2.9. Actual assays estimation :-

It was desired to find the relationship between the lode assay and actual assay measurements.

Using the results from all 745 of the possible stope

samples, a scattergram between the lode and actual assays was drawn. As expected, the correlation coefficient of 0.75 was found to be highly significant.

As stated before (see section B.1.2) a call assay value was calculated at every sample point. It was derived from the lode width, lode assay, call width, and average assay of the adjacent country rock. For each of the stope samples there exists a true measured actual assay to compare with this predicted value.

Obviously an actual assay value is a function of the lode assay. So polynomial equations of the form :-

$$\text{Actual assay estimate} = f(\text{LA}, \text{LA}^2, \dots, \text{LA}^N)$$

where LA = Lode assay

N = Order of the polynomial

were examined. The 'least squares best fit' fit equations of order zero to nine were calculated. With the constraint that the actual assay estimate must not be less than the lode accumulation divided by the actual width estimate (calculated as in section B.2.8), the RSS of each of these equations was found to be :-

Polynomial of order 0		RSS = 878.2
"	1	" 705.6
"	2	" 616.0
"	3	" 617.2
"	4	" 617.3
"	5	" 619.7
"	6	" 614.0
"	7	" 582.1
"	8	" 564.5
"	9	" 564.6

Now, using 'F' statistics from these figures, the order eight equation can be shown to be significantly better than that of order two (the 'F' statistic is 1.08 with 743 and 737 degrees of freedom). However, Whitten (1973) states that computation should stop the first time a step up in order does not result in a significant increase in accuracy. Polynomials of higher order should not be used. Obeying this rule the polynomial of order two is found to be the appropriate one. There is a significant increase in accuracy between the equations of order one and two (the 'F' statistic is 1.14 with 744 and 743 degrees of freedom), but not between two and three (the 'F' statistic is 1.00 with 743 and 742 degrees of freedom). Indeed, due to the constraint mentioned above, the order three polynomial is slightly less accurate than that of order two.

So, the actual assay estimator was chosen to be :-

$$= 6.573*10^{-2} + 6.167*10^{-1}*LA - 8.690*10^{-3}*LA^2$$

This estimation method produces a RSS term of 616.0. The TSS of the stope data was calculated to be 3709.9. Therefore, the selected actual assay estimator explains 83 percent of the data TSS.

B.3. STOPE SIMULATION

B.3.1. General approach :-

It was desired to produce simulations of some of the stopes within No.9 lode. The simulations would be of lode widths and lode values. They would be produced using the characteristics of these two parameters discovered in sections B.2.6 and B.2.7. Any relevant data from the main development levels would be used to condition the simulations.

B.3.2. Representation of the stope :-

The 'turning bands' simulation method produces values at finite points within a given area. Therefore, it is impossible to produce an exact simulation of any area. To do that a value is required at every one of the infinite number of points within the area. In practice the simulated region has to be represented by a grid of point values.

The number of points used in such a grid clearly has a great effect on the cost of the simulations. It is necessary to have enough, but only just enough, points to adequately imitate the behaviour of the simulated area. David (1977) states that '16 points or even 9 and sometimes 4 are adequate to obtain a stable block estimate'. Clark and White (1976) report that '64 points are sufficient to characterise the estimation of a block to within 1 percent accuracy'.

It was decided to use a 9*9 rectangular grid to

represent each of the simulated stopes. In practice, this meant that the grid spacing varied between 0.6 and 4.1 metres.

B.3.3. Simulation programs :-

Two computer programs were written to enable any stope within No.9 lode to be simulated. The simulations were conditioned to data from the main development levels immediately above and below the stope.

The first program, CONSIM1 (see Appendix 11), requires the following input of data :-

- (a) The exact size and position of the stope.
- (b) The parameters of the semi-variogram model for lode widths and for lode assays.
- (c) The development data to be used to condition the simulations. In practice, any sample from a main level drive, within 25 metres of an edge of the stope, was included. This meant there were about one hundred conditioning data points.

To condition the simulations a kriging system has to be solved for every one of the 81 grid points (see section A.4.1). It was considered an unwarranted expense to use all the data points in every kriging system. Instead, only the nearest three points in six segments of 60 degrees were included. On average, this meant that each kriging system used about seventeen data points. The average variance of the estimates produced by these systems was about 0.22

(65 percent of the total sill) for the lode widths, and 1.62 (72 percent) for the lode assays. For each of the 81 grid points CONSIM1 finds the eligible data points and calculates the kriging weights. This information is stored on a permanent file for use in the second stope simulation program, CONSIM2 (see Appendix 12).

CONSIM2 produces unconditional simulations of the 9*9 stope grid. Reading the kriging weights calculated by CONSIM1 off the permanent storage file, the simulations are conditioned to the main development level data. The output of CONSIM2 includes, for both lode widths and assays, the average of the 81 simulated values, and their semi-variogram and histogram. Each run of CONSIM2 produces five conditional simulations of the stope. It can only be used after CONSIM1 has been completed. The sole input required (apart from that created on file by CONSIM1) is a four digit number which acts as a trigger for the random number generator. If more than five simulations are needed, CONSIM2 merely has to be re-run with a different four digit trigger number. CONSIM1 does not have to be used again, thus saving considerable time and expense.

B.3.4. Conditional simulation results :-

Two typical worked out stopes within No.9 lode were considered. Five simulations of each stope were produced and the semi-variogram and histogram of each set of simulated values are displayed in figures 51, 52, 53,

FIGURE 51 : CONDITIONAL SIMULATIONS
OF LODE WIDTHS FOR MEREDREW STOPE

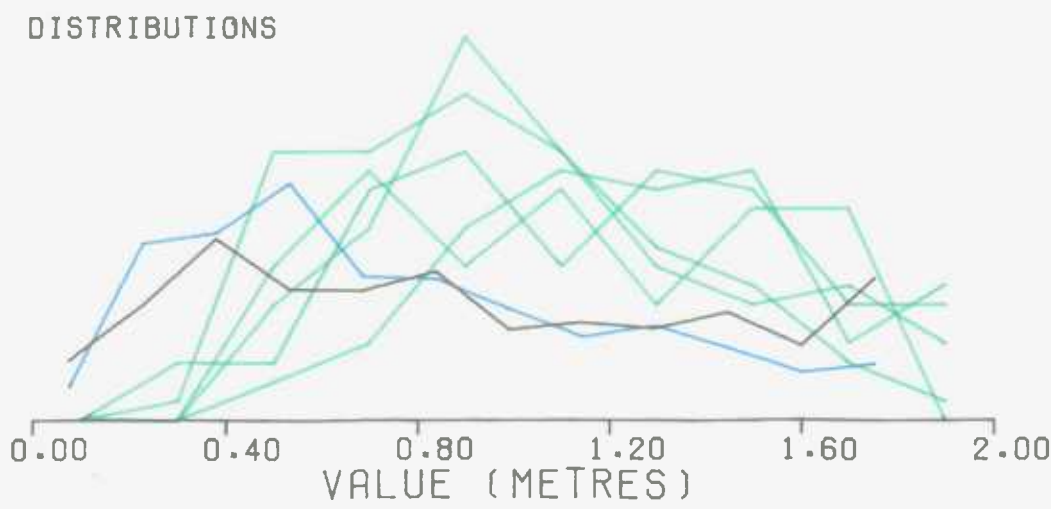
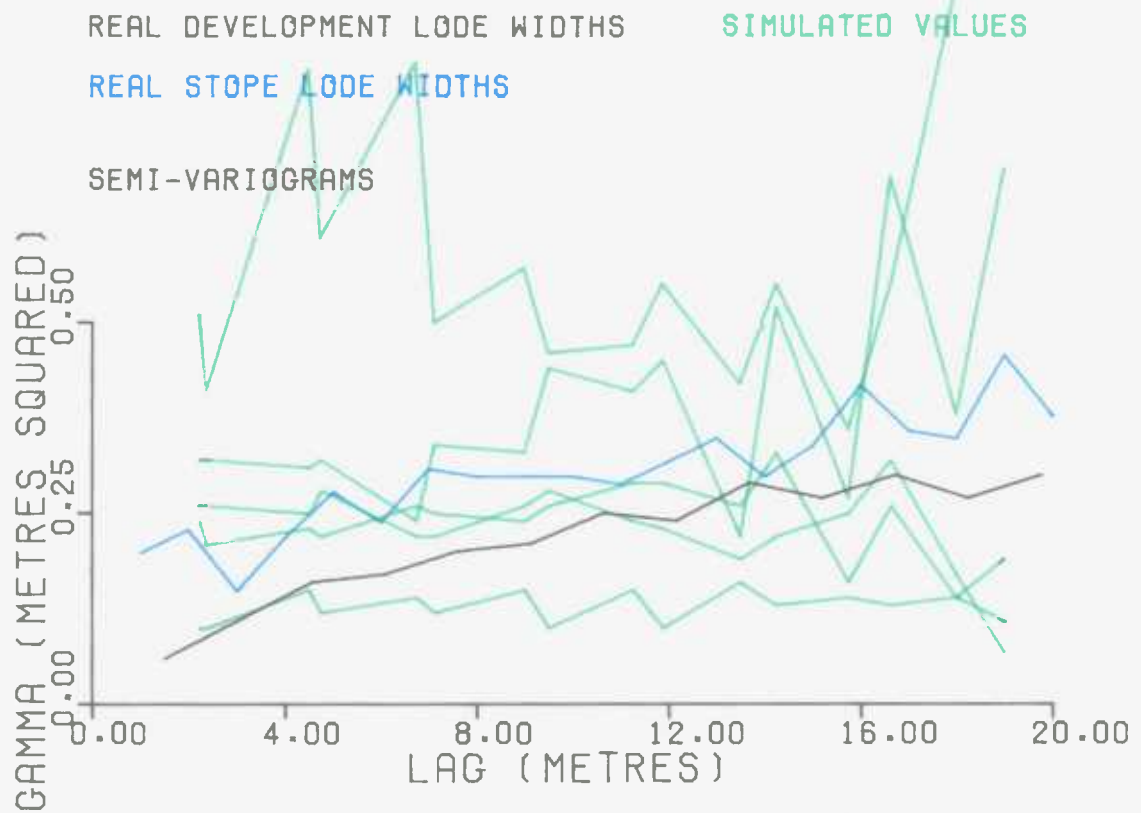
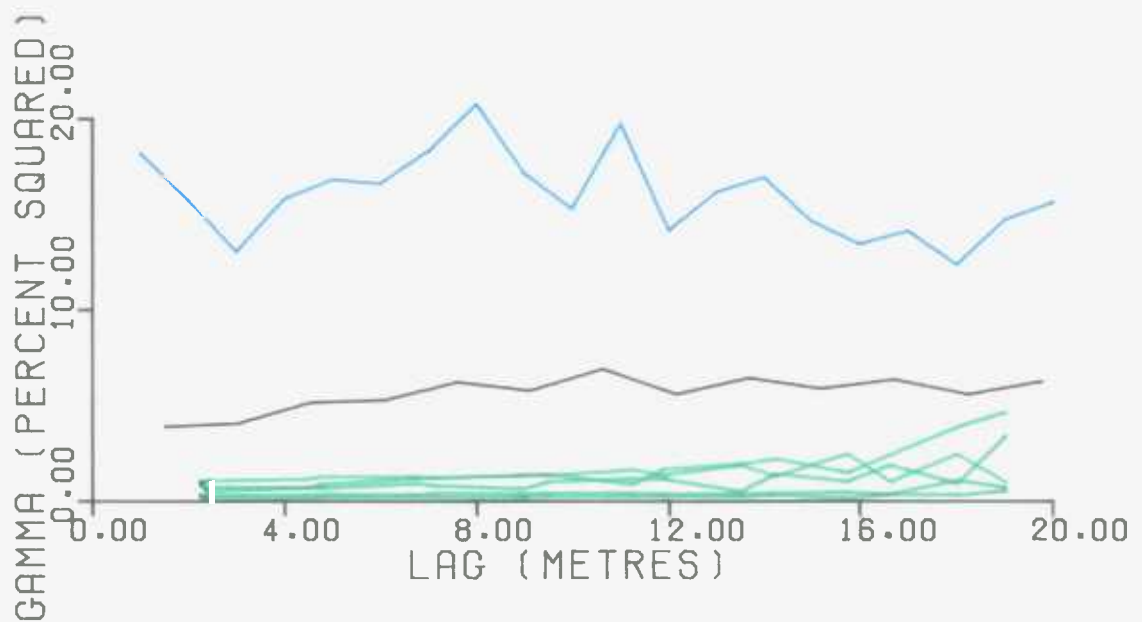


FIGURE 52 : CONDITIONAL SIMULATIONS
OF LODE ASSAYS FOR MEREDREW STOPE

REAL DEVELOPMENT LODE ASSAYS SIMULATED VALUES
REAL STOPE LODE ASSAYS

SEMI-VARIOGRAMS



DISTRIBUTIONS

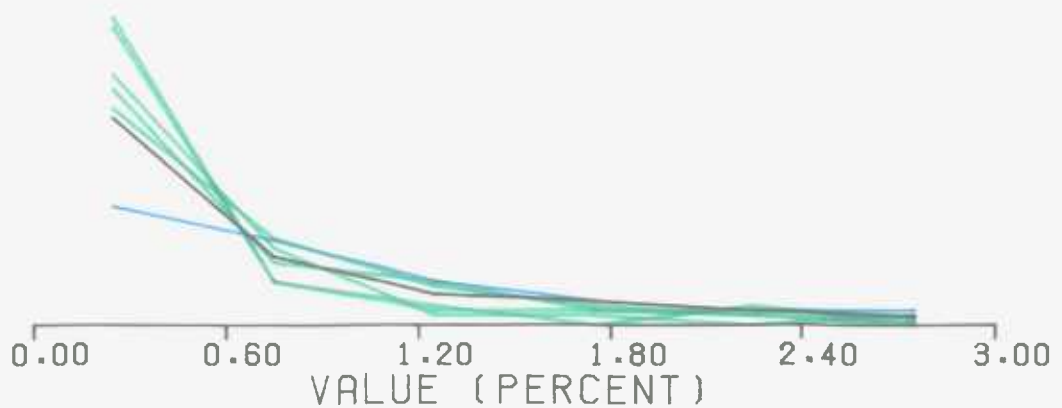
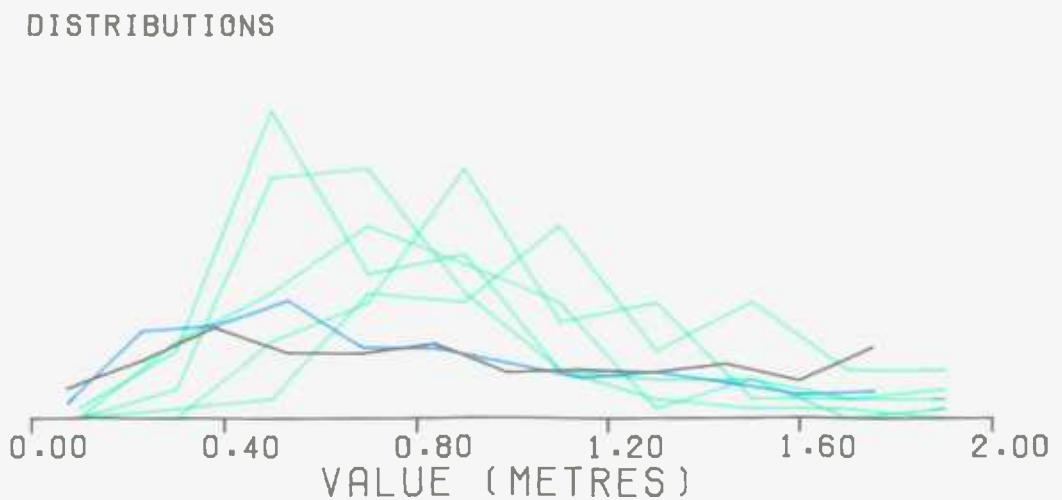
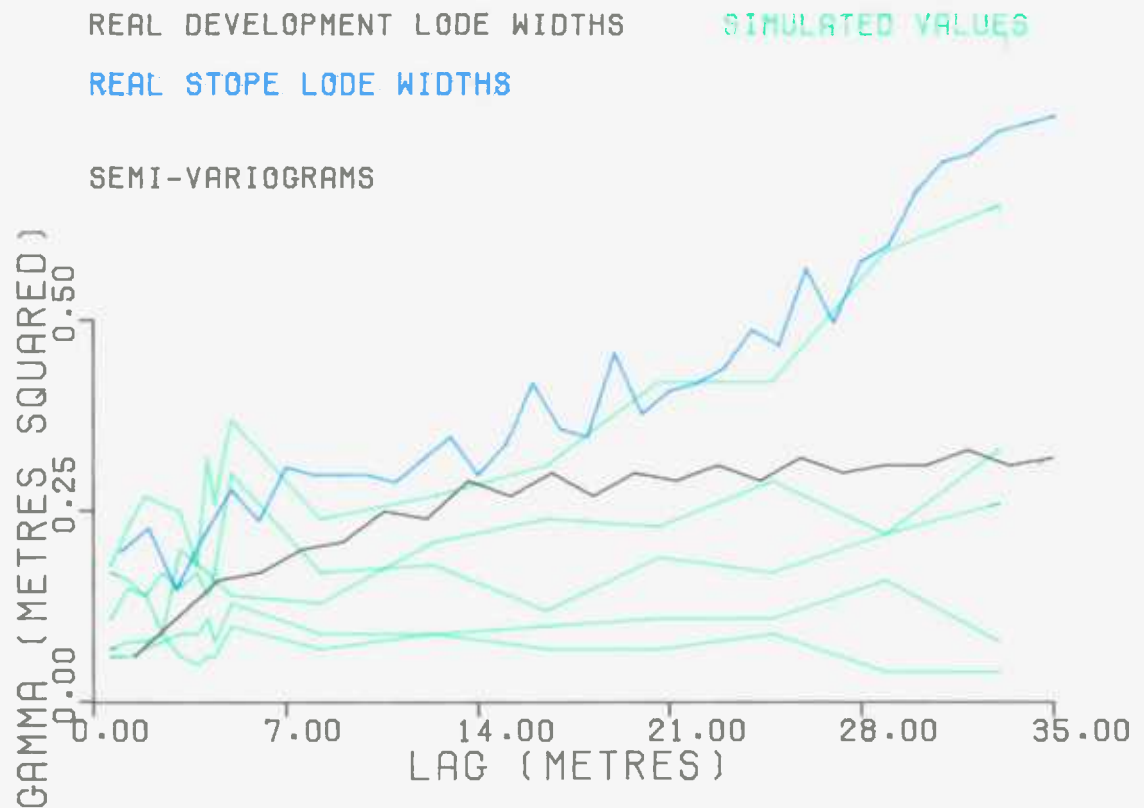


FIGURE 53 : CONDITIONAL SIMULATIONS
OF LODE WIDTHS FOR PISOWOCKI STOPE



and 54. Table 6 summarises the results of the simulations by showing :-

(a) The arithmetic mean of the 81 simulated lode width values.

(b) The percentage of the total sum of squares (TSS; see section B.2.6) of the simulated ln lode width values explained by the cubic ln lode width trend surface. This was calculated from :-

$$= 100.0 * \frac{(TSS - RSS)}{TSS}$$

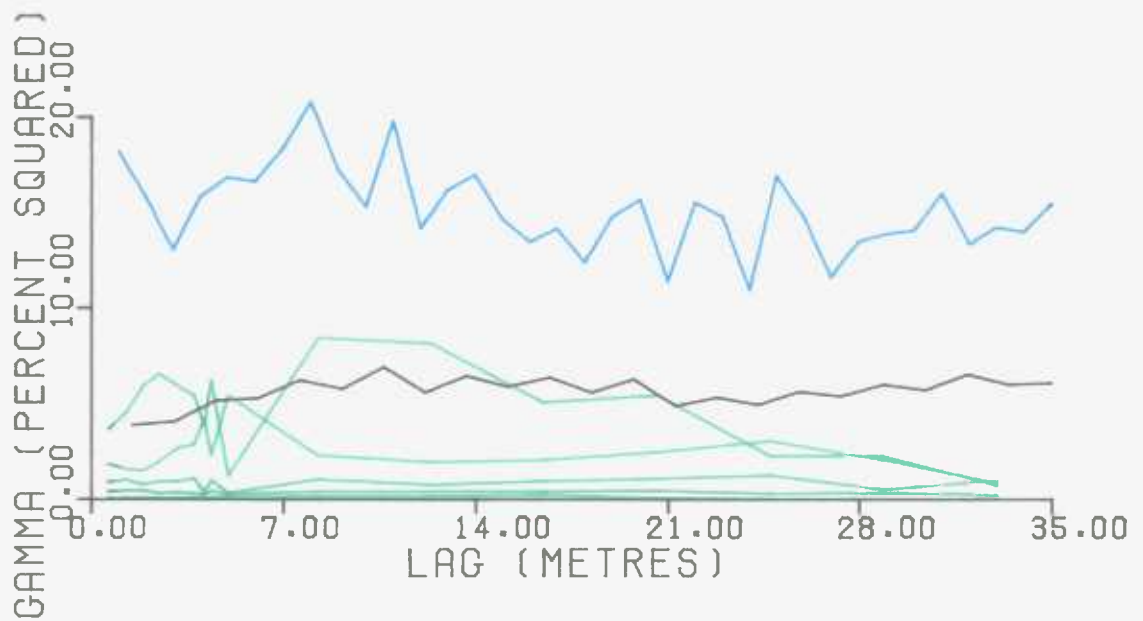
It has been stated before that for all 1149 development samples the cubic trend surface explains 32 percent of the TSS of the ln lode width values. The trend surface is a global estimator based on the development data. The TSS value was defined as the total sum of squares with respect to the data mean. So it is the residual sum of squares of the sample population mean. For almost all of the simulations the percentage of the TSS explained by the cubic trend is negative. This is due to the RSS term being greater than the TSS value. In other words, the RSS of the trend surface is greater than that of the development data mean. Overall, for each set of 81 simulated ln lode widths, the trend is a worse estimator than their arithmetic mean.

(c) The arithmetic mean of the 81 simulated lode assay values.

FIGURE 54 : CONDITIONAL SIMULATIONS
OF LODE ASSAYS FOR PISOWOCKI STOPE

REAL DEVELOPMENT LODE ASSAYS SIMULATED VALUES
REAL STOPE LODE ASSAYS

SEMI-VARIOGRAMS



DISTRIBUTIONS

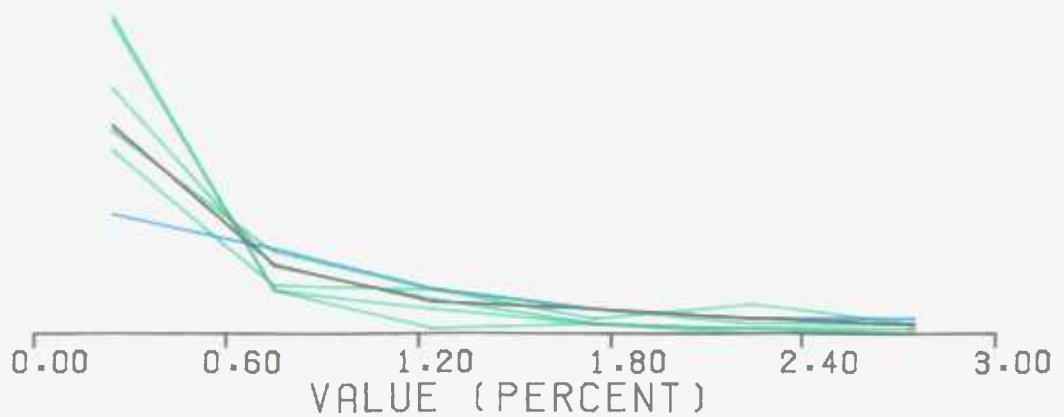


TABLE 6

Stope name	Simulation results			Level data	Stope samples within simulated area			
	Average lode width (a)	% of TSS (b)	Average lode assay (c)	% of TSS (d)	Number (e)	Average lode width 90% limits (f)	% of TSS (g)	Average lode assay 90% limits (h)
Meredew	1.30	9.0	0.76	23.3	29	1.97-2.28	-67.2	0.41-0.55
310	1.59	-3.0	0.37					
	1.20	-9.7	0.79					
	1.20	-6.3	0.72					
level	0.97	-76.1	0.47					
Pisowocki	0.88	-2.8	0.94	25.9	20	1.78-1.95	-674.3	0.83-1.00
	0.78	-6.9	0.38					
360	0.69	-41.2	1.60					
	1.11	-18.6	0.32					
level	1.18	-61.3	0.61					

(d) The percentage of the TSS of the conditioning data in lode width values explained by the cubic trend. The trend surface is a global estimator derived from all the development samples. The percentage of TSS shown under (d) indicates the local 'goodness of fit' with respect to the development samples used to condition the simulations.

(e) The number of stope samples sited within the simulated area.

(f) The 90 percent confidence limits on an estimate of the average lode width of the simulated area.

The lode width measurements, from the stope samples within the simulated area, were converted to their simulation form. That is for each lode width measurement the logarithm was taken and the trend value subtracted. The arithmetic mean of these figures was found. This is an estimate of the average value over the simulated area. Its variance was calculated using the values of standard geostatistical auxiliary functions determined by computer subroutines (Clark, 1976). Assuming a Normal distribution, the 90 percent limits of the estimate were found by subtracting and adding 1.6449 times the estimation standard deviation. These two figures were then converted back to units of metres. This was done by adding the average

trend value over the simulated area (as approximated by a 10*10 grid), and taking the exponential. The figures under (f) in table 6 show the two limits between which the average lode width of the simulated area is 90 percent likely to lie.

(g) The percentage of the TSS of the stope ln lode width values explained by the cubic trend surface. All three of these figures are negative. This indicates that the trend surface, derived from the development samples, has little application within the stopes. For each set of stope ln lode widths the trend is a worse predictor than the average of the set.

(h) The 90 percent confidence limits on an estimate of the average lode assay of the simulated area. The arithmetic mean of the ln lode assay terms from all the stope samples within the area was found. Using auxiliary functions the variance of this estimate was determined. The 90 percent limits were calculated, and transferred back to units of 'percent tin' by taking their exponentials. The resulting limits, within which the average lode assay of the simulated area is 90 percent likely to lie, are shown.

When considering the stope semi-variograms it should be borne in mind that there are only 81 point values in the simulation grid. The grid itself is not square which means

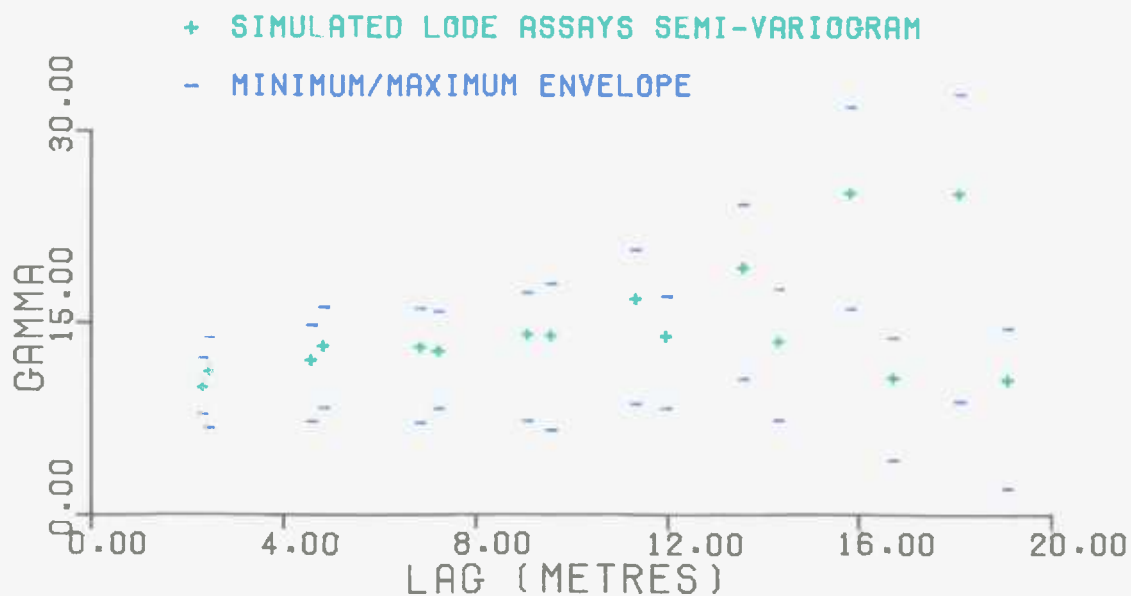
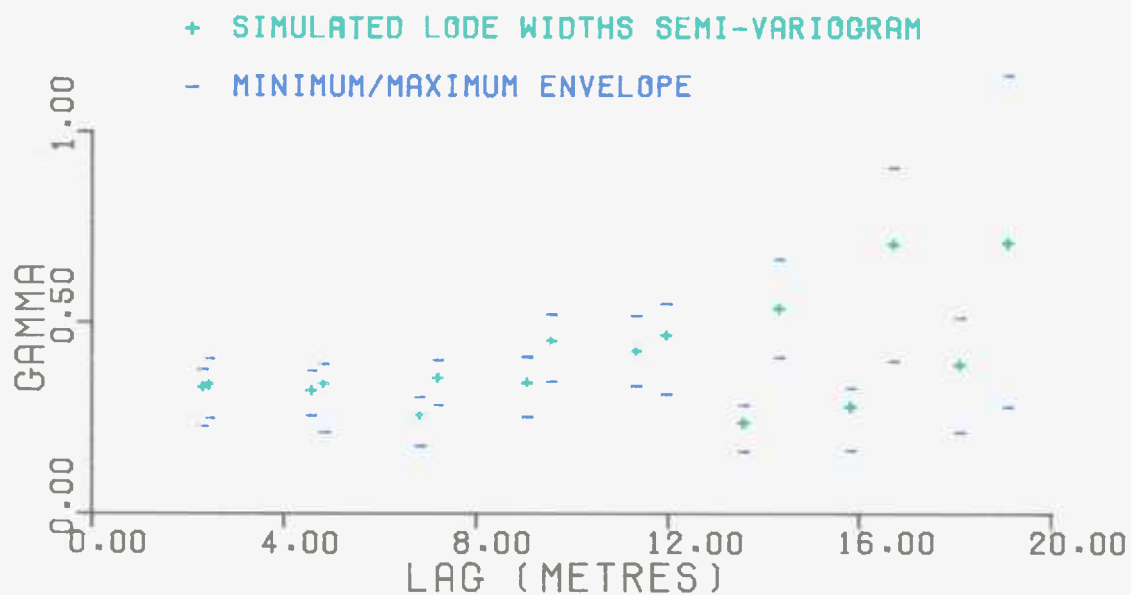
that the spacing of the points differs between the two main directions. In other words, the semi-variogram figures are calculated at different lag intervals. The semi-variogram figure at a certain lag value in one direction does not have a partner at exactly the same lag value in the other direction. Therefore, the number of $[G(x)-G(x+h)]^2$ figures contributing to any one semi-variogram value varies between nine and seventy two. This does not induce great confidence in the reliability of the semi-variogram.

The reliability of the semi-variograms from one of the Meredew stope simulations was tested by fifty random removals of 10 percent of the data (see section B.2.4). Figure 55 shows the resulting minimum/maximum envelope, both for the lode width and lode assay semi-variogram. It indicates the fragile nature of the semi-variogram values.

The results of the lode width simulations seem to be realistic (see figures 51 and 53, and table 6). The semi-variograms are all well within reach of the real development data semi-variogram. The distributions have smaller spreads than that of the development data. This is only to be expected since all the simulated values are close to each other. For any one point all the others lie well within the range of influence. Thus, the variance amongst them is likely to be less than that from the development data where samples are up to 600 metres apart.

The simulations are of development lode widths and assays. For reasons explained before (see sections B.2.6

FIGURE 55 : THE MINIMUM/MAXIMUM ENVELOPES
 FOR MEREDREW STOPE SIMULATION NUMBER 1



and B.2.7) only the development samples were used in the data analysis. So, the semi-variogram and distribution models were obtained from the five horizontal lines of the main development level drives. Any stope simulation involves the vertical interpolation of the development models between two of these horizontal lines.

The errors caused by this phenomenon are particularly evident in the performance of the ln lode width trend surface. It does not fit the stopes at all well, as shown by the percentage of the stope samples TSS explained by the trend ((g) in table 6). The simulations have tended to reflect this characteristic, rather than following the trend as does the development data. This is a result of the conditioning process. Forcing the simulations to agree with both horizontal lines of development data has prejudiced them towards certain values inbetween.

Since the ln lode width trend is such a poor fit to the stopes it was decided to ignore it. The simulated values would be of ln lode width and the conditioning process would be relied upon to allow for any local features.

The semi-variogram model chosen for the ln lode widths was Spherical(30 metres, 0.33, 0.10), as shown in figure 43. Although it does not fit the actual semi-variogram closely above a lag of 35 metres this was felt to be insignificant. Almost all the work involved in stope simulation is at distances of less than 35 metres. The ln lode width figures

do not fit any Normal distribution. As stated before (see section A.4.1) the average of an unconditional simulation is not important, and the true variance is best estimated by the sill of the semi-variogram. It was anticipated that any inaccuracies caused by the poorly fitting auto-correlation and distribution models would be coped with by the conditioning process.

Using this new approach the same two stopes as before were simulated. Figures 56 and 57 display the resulting lode width semi-variograms and histograms. The overall average for each set of simulated values was found, and also the percentage of the TSS explained by the ln lode width trend surface. The results were found to be as follows :-

Meredew 310 level	Simulation 1	Lode width = 1.01	%TSS = 20.0
	" 2	" 1.62	" -147.6
	" 3	" 1.64	" -109.3
	" 4	" 0.58	" -99.1
	" 5	" 1.47	" -10.0
Pisowocki 360 level	Simulation 1	Lode width = 1.11	%TSS = -39.3
	" 2	" 0.57	" -278.3
	" 3	" 1.21	" -91.3
	" 4	" 1.22	" -149.5
	" 5	" 0.72	" -47.2

The results compare well with those achieved using the trend surface approach (see figures 51 and 53, and

FIGURE 56 : CONDITIONAL SIMULATIONS
OF LODE WIDTHS FOR MEREDREW STOPE
IGNORING TREND

REAL DEVELOPMENT LODE WIDTHS SIMULATED VALUES
 REAL STOPE LODE WIDTHS

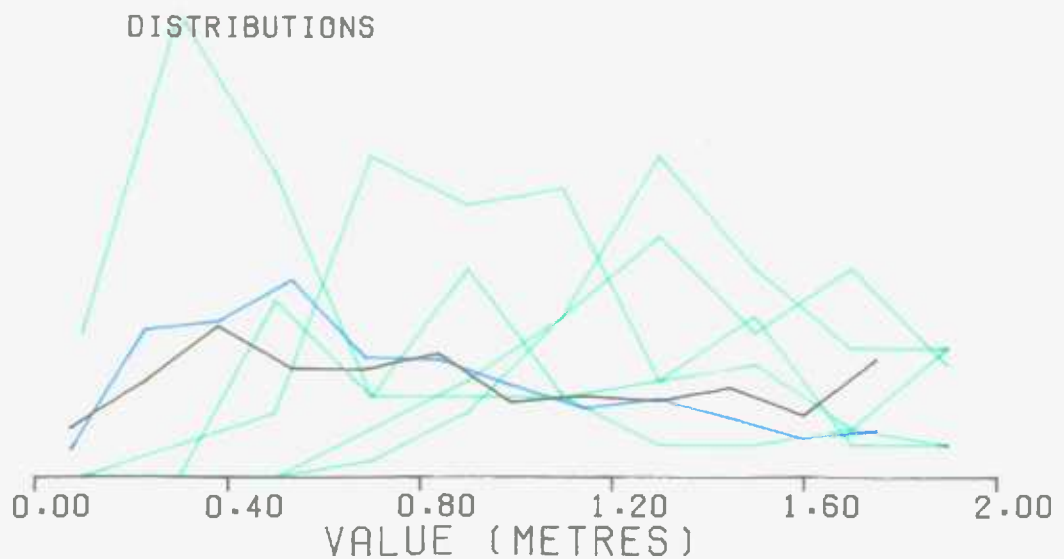
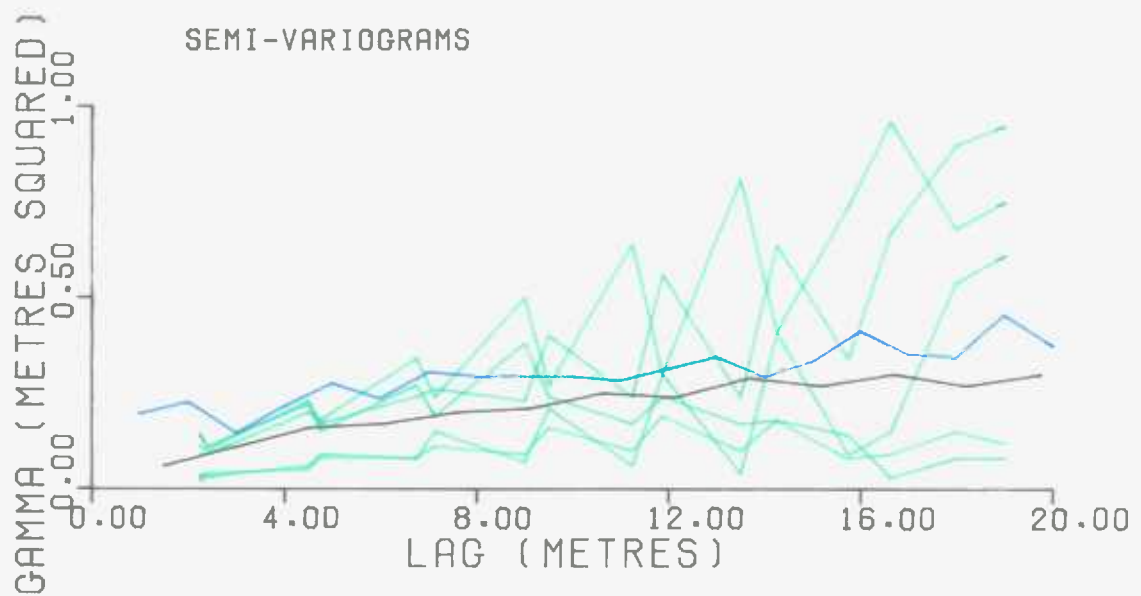


FIGURE 57 : CONDITIONAL SIMULATIONS
OF LODE WIDTHS FOR PISOWOCKI STOPE

IGNORING TREND

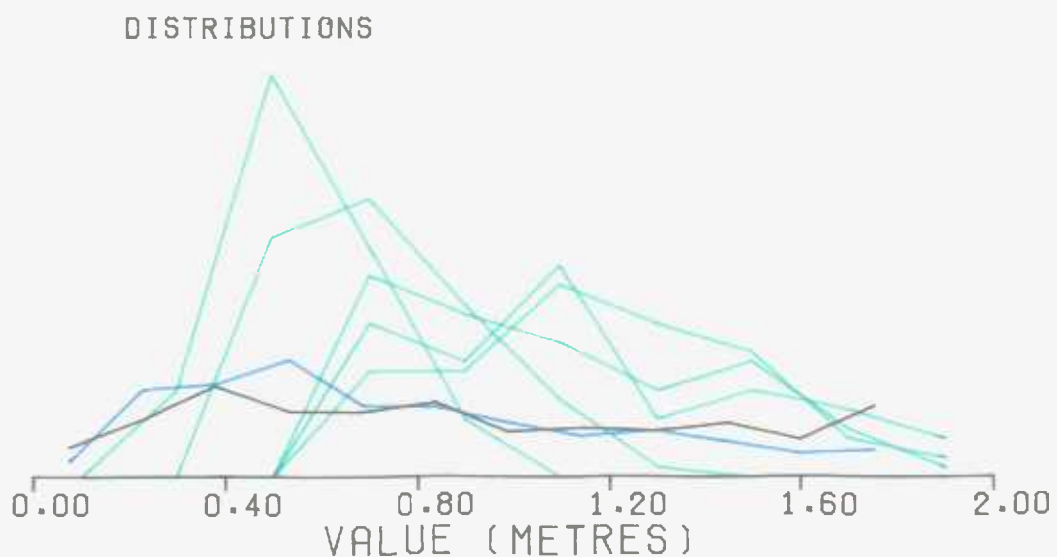
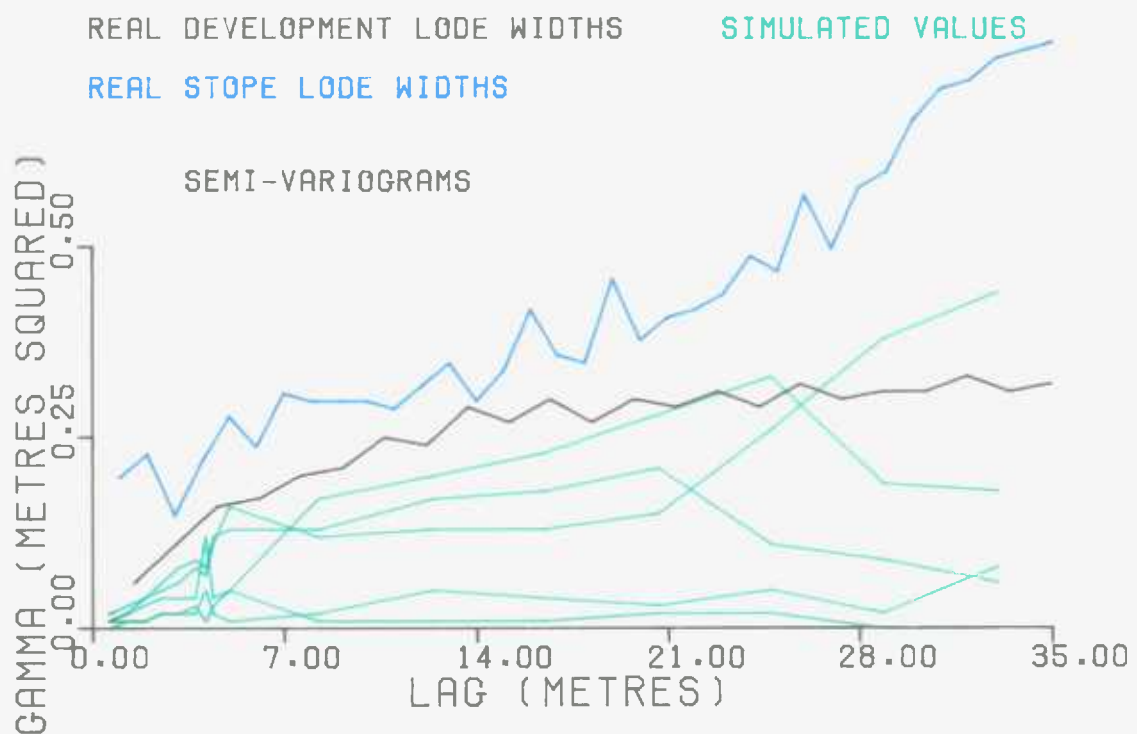


table 6). Throughout the rest of this study the lode width simulations were produced in this new fashion. That is the ln lode widths were treated as following a Normal distribution and a spherical semi-variogram. The conditioning operation was relied upon to cope with the local effects of the trend.

At first sight the results of the lode assay simulations do not appear to be very realistic (see figures 52 and 54, and table 6). In particular, most of the semi-variograms are lower than that from all the development samples. The development semi-variogram itself is much lower than that calculated from the stope samples. This is due to the proportional effect caused by the lognormal distribution of the lode assays (see section B.2.7). Obviously, the samples from the stopes have a much higher average assay value (2.54 percent tin) than those from the development drives (1.35 percent tin). This means that their variance, and therefore also their semi-variogram, is higher.

The proportional effect explains why the simulation semi-variograms are so low. Strangely, neither Meredew nor Pisowocki stope has a high average assay. As shown in table 6, the average of the stope samples taken within them was found to be 0.47 and 0.91 percent tin respectively. Therefore, the simulations tend to have averages lower than that of all the lode development samples. Correspondingly their semi-variograms are lower than average. The one

exception to this is a simulation of Pisowocki stope with high semi-variogram values between 7 and 20 metres. This is simulation number 3 which has an unusually high average of 1.60 percent tin.

The distributions of the simulated lode assays are as expected. They exhibit a lognormal shape, and have a smaller spread than those from the development and stope samples. As with the lode widths simulations this is due to the simulated points being close to each other.

B.3.5. Susceptibility of the simulations to the models :-

The stope simulations are formed using auto-correlation and distribution models for ln lode widths, and ln lode assays. These models are representations of the real values calculated from the No.9 lode development samples. As shown in figures 42, 43, 48, and 50, they are not exact fits to the data. It was anticipated that the discrepancies would become insignificant after the powerful conditioning process had taken place.

To investigate the influence of the models five simulations of Meredew stope were produced. The simulations were created with slightly different models from those used previously.

The semi-variogram model used for the ln lode widths was Spherical(40 metres, 0.33, 0.10). In other words, the range of influence was increased by one third. The results

of the simulations using this model are shown in figure 58. The average lode width of each simulation, and the percentage of the TSS explained by the cubic trend surface, were found to be as follows :-

Meredew	Simulation 1	Lode width = 1.71	%TSS = -25.6
	" 2	" 1.06	" -20.3
310	" 3	" 1.40	" -10.8
	" 4	" 1.65	" -15.6
level	" 5	" 1.00	" -70.6

All these results are remarkably similar to those achieved with a 30 metres range (see figure 51 and table 6). The semi-variograms of the conditional simulations show no evidence of the larger range. The conditioning operation has a stronger influence than the range of the unconditional simulations. This is as expected, since at any lag value the difference in semi-variogram value between a 30 metres and 40 metres range model is small. For instance, at a lag of 10 metres the two models have values of 0.21 and 0.18 respectively.

The stope simulation method is resistant to an error of one third in estimation of the semi-variogram model range of influence. Any larger estimation error seems unlikely.

The results shown in figure 59 were created with a 1n lode assay semi-variogram model of Spherical(69 metres, 2.26, 0.60). In other words, the nugget effect was halved. The average of each set of simulated

FIGURE 58 : CONDITIONAL SIMULATIONS
OF LODE WIDTHS FOR MEREDREW STOPE
IGNORING TREND - WITH DIFFERENT MODELS

REAL DEVELOPMENT LODE WIDTHS SIMULATED VALUES
REAL STOPE LODE WIDTHS

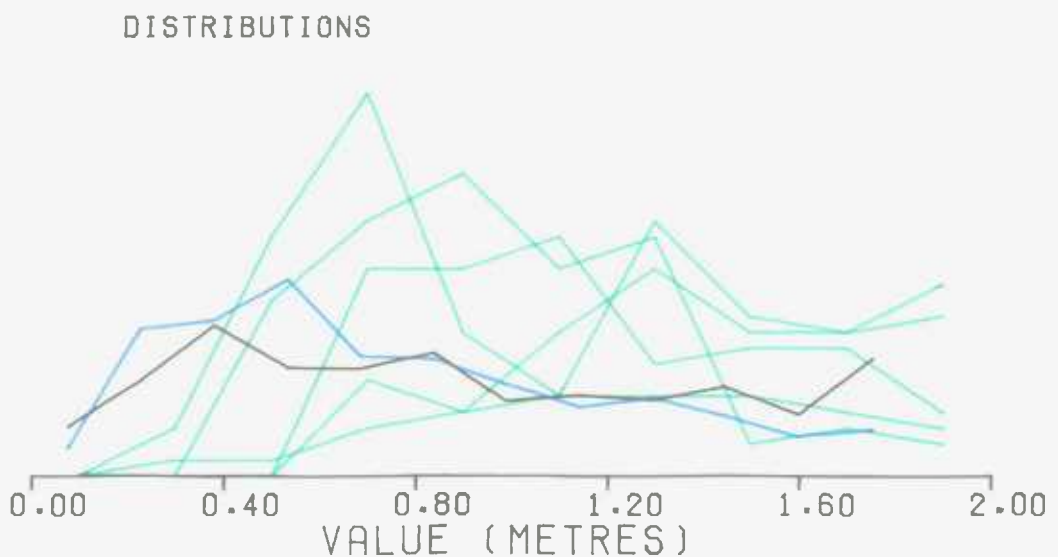
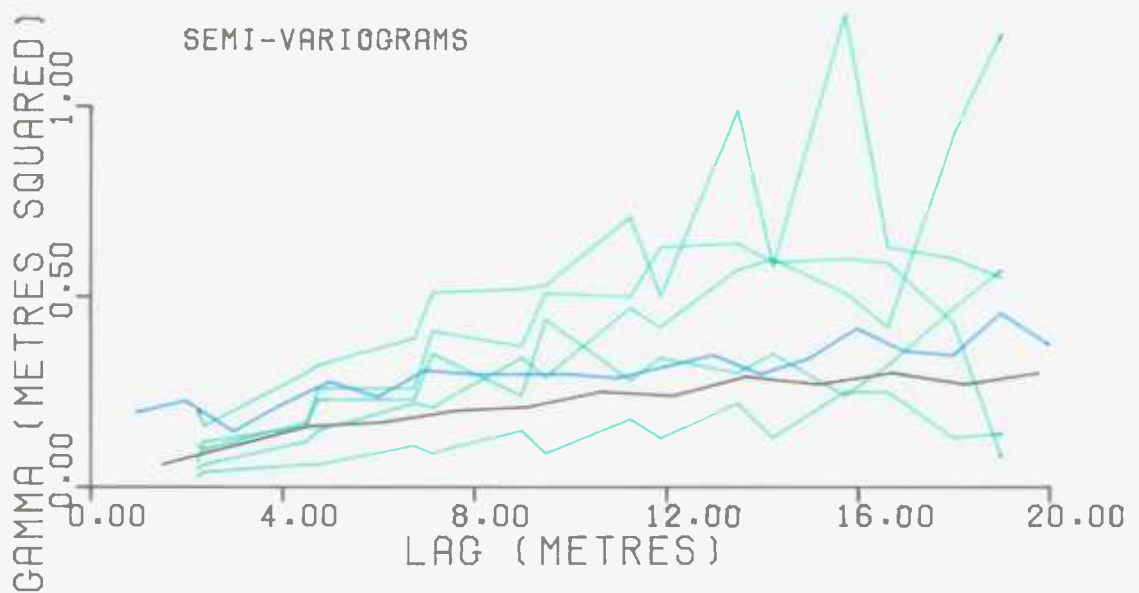
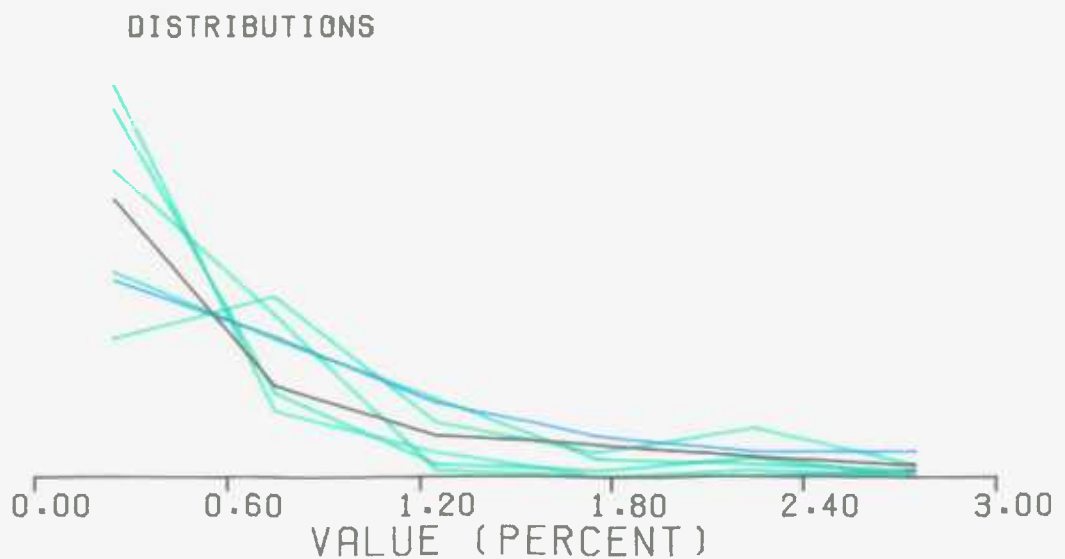
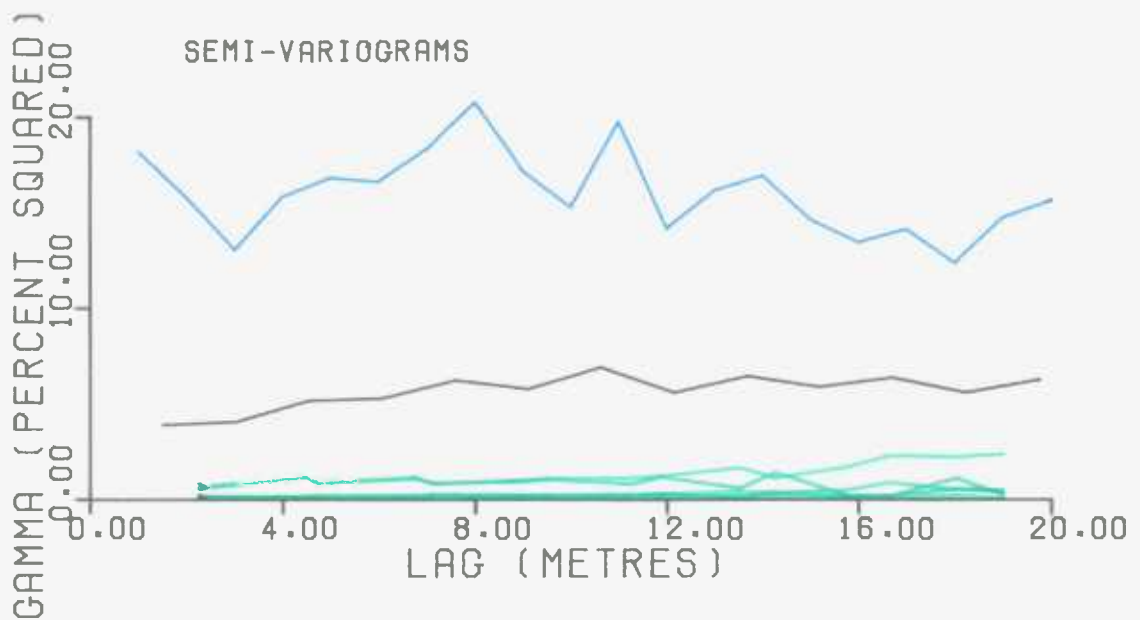


FIGURE 59 : CONDITIONAL SIMULATIONS
OF LODE ASSAYS FOR MEREDREW STOPE
- WITH DIFFERENT MODELS

REAL DEVELOPMENT LODE ASSAYS **SIMULATED VALUES**
REAL STOPE LODE ASSAYS



lode assays was calculated to be :-

Meredew	Simulation 1	Lode assay = 0.48
	" 2	" 1.13
310	" 3	" 0.89
	" 4	" 0.39
level	" 5	" 0.37

There is only one significant difference between these results and those formed with a nugget effect of 1.20 (see figure 52 and table 6). This is in the semi-variograms. Those of figure 59 are lower in value for all lags. So, the smaller nugget effect has been transferred from the unconditional simulations. The conditioning process has not forced both sets of semi-variograms to exhibit similar semi-variograms.

The stope simulation method is influenced by the value of the nugget effect. However, it must be borne in mind that both sets of semi-variograms are low. They are very different from the average for all the lode. This is due to the proportional effect which has an overriding influence on the lode assay semi-variogram. For any given stope its importance far outweighs that of the value of the semi-variogram model nugget effect. Thus, for small scale conditional simulation an error in the estimation of the nugget effect is not very significant. Its only effect may be a small change in the simulation's semi-variogram. However, this change is likely to be small compared to that caused by the conditioning data forcing the local average

on the simulation.

B.3.6. Costs of the simulations :-

For Meredew stope, the time taken for the program CONSIM1 to execute was 40 CP seconds. CONSIM2 required 727 CP seconds to produce five conditional lode widths and lode assays simulations of the stope. This means that 153 CP seconds were used to create each simulation.

The results of each set of five simulations are extremely variable. As stated before, it is dangerous to draw conclusions from only a few simulations. It was considered that five were not enough, and that one hundred was a far more useful number. To create one hundred stope simulations would require CONSIM2 to be run twenty times, since it can only produce five simulations in one execution. Apart from the inconvenience, the cost of such an operation was considered to be prohibitive. To produce one hundred simulations would require $40 + 20 \times 727 = 14580$ CP seconds. Per simulation this is 146 CP seconds, which at current (October 1979) commercial rates costs about 16 pounds Sterling.

It was decided that a less costly simulation technique was necessary, even if it involved a slight decrease in accuracy. Such a technique is examined in section B.4.

B.3.7. Actual width and actual assay simulations :-

Section B.2.8 shows how an actual width estimate can be calculated at a point, given the lode width and the

co-ordinates. The estimating procedure is a fairly good one since it explains 66 percent of the total sum of squares. Similarly, section B.2.9 describes an estimation technique for actual assays using only the lode assay at the point. This is a more accurate predictor, explaining 83 percent of the TSS.

Bearing these two good estimators in mind, it was decided to produce simulations of actual widths and assays. The possible behaviour of these stope parameters is probably of more interest than that of the lode widths and assays.

Two potential methods of simulating actual widths and assays presented themselves :-

(a) Simulate the 9*9 grid of lode values. Condition the simulation to the development data. At each of the grid points find the estimates of the actual values.

(b) Produce a simulation of actual values. For every one of the conditioning data points calculate the actual value estimates. Use these to condition the simulation.

It was considered that the first method was the most appropriate one. Analysis of the lode measurements is likely to be more meaningful than that of the actual, or stope, widths and assays. There are many more samples available with lode measurements. The only recordings of actual values occur in the preferentially placed stopes.

The actual widths and assays are measurements of the stope faces, and as such are subject to great human influence. Since the stope width is a factor of the efficiency of the mining (in terms of overmining of the lode), it is not merely a geological parameter. Changes in behaviour between stopes can be a result of having been worked by different mining crews. For this reason analysis of actual width and assay measurements was considered to be fraught with dangers.

Using method (a) mentioned above, five conditional simulations of actual width and actual assay were produced for Meredew stope. The simulations exhibited semi-variograms and histograms for widths and assays, as shown in figures 60 and 61 respectively. Also displayed are the semi-variogram and histogram from all 749 of the stope samples. The average values of the 9*9 grid of simulated points were found to be :-

Simulation 1	Actual width = 1.88	Actual assay = 0.52
" 2	" = 2.22	" = 0.29
" 3	" = 1.85	" = 0.54
" 4	" = 1.85	" = 0.49
" 5	" = 1.78	" = 0.35

Taking the arithmetic mean of the 29 stope samples within the simulated area produces 'real' values of 3.17 metres and 0.41 percent tin.

The simulated actual widths do not exhibit the desired

FIGURE 60 : CONDITIONAL SIMULATIONS
OF ACTUAL WIDTHS FOR MEREDREW STOPE

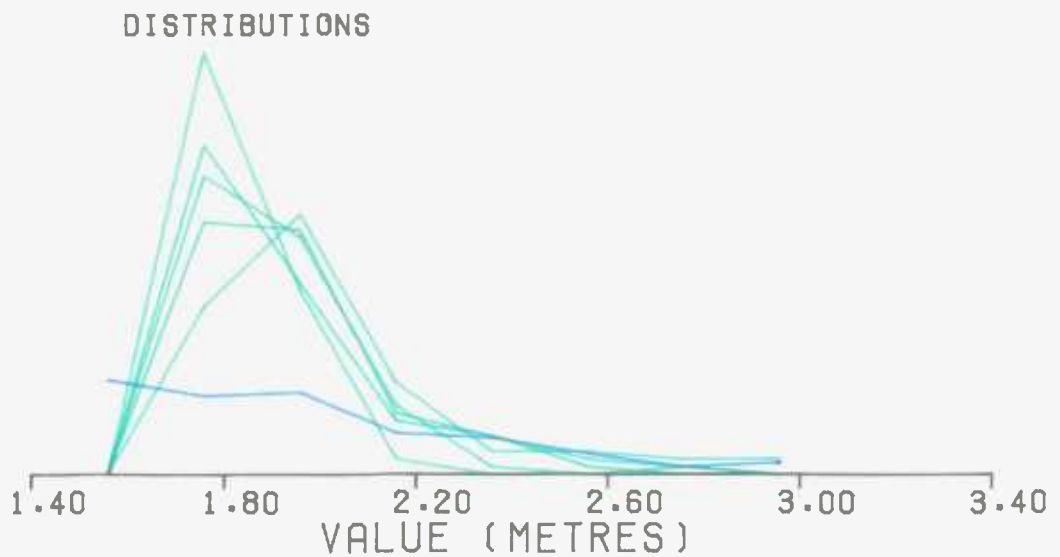
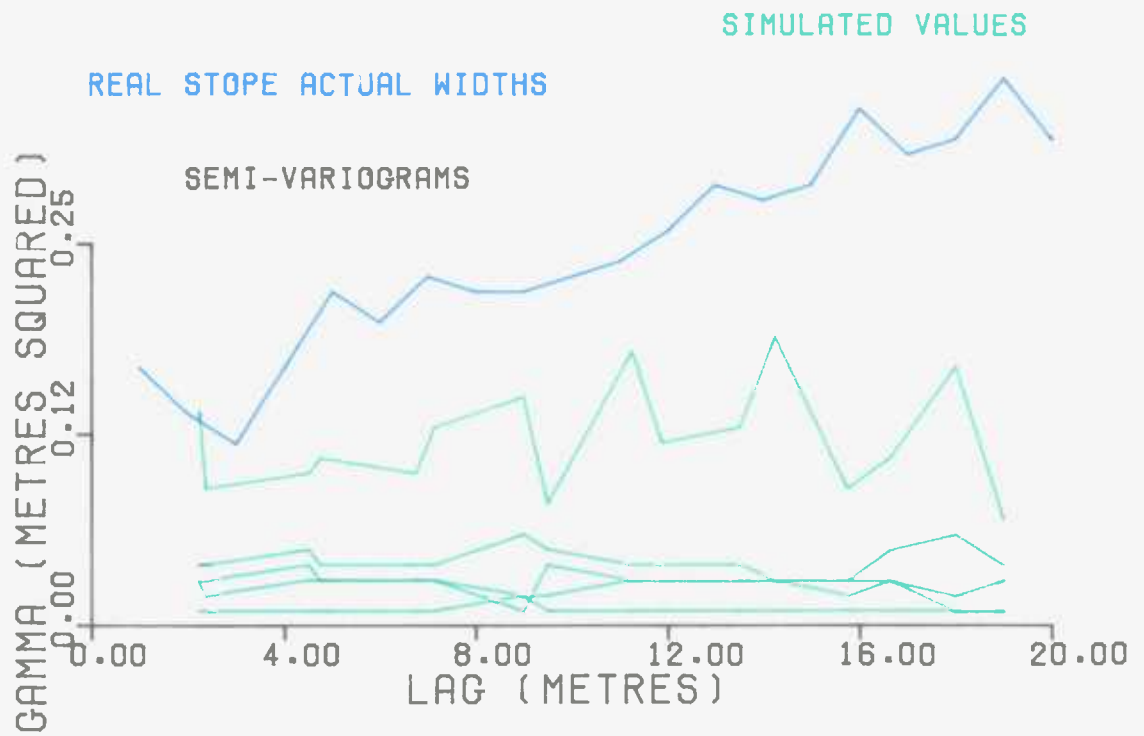
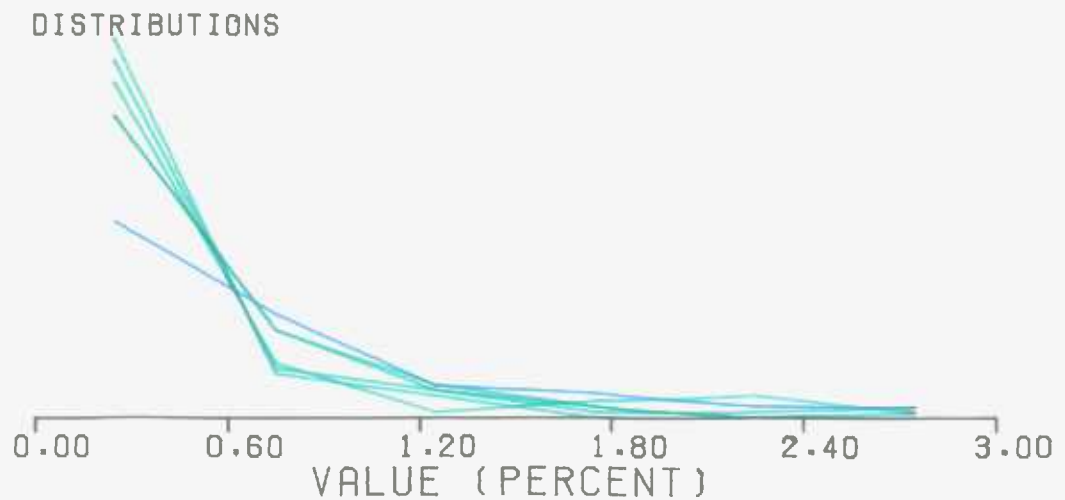
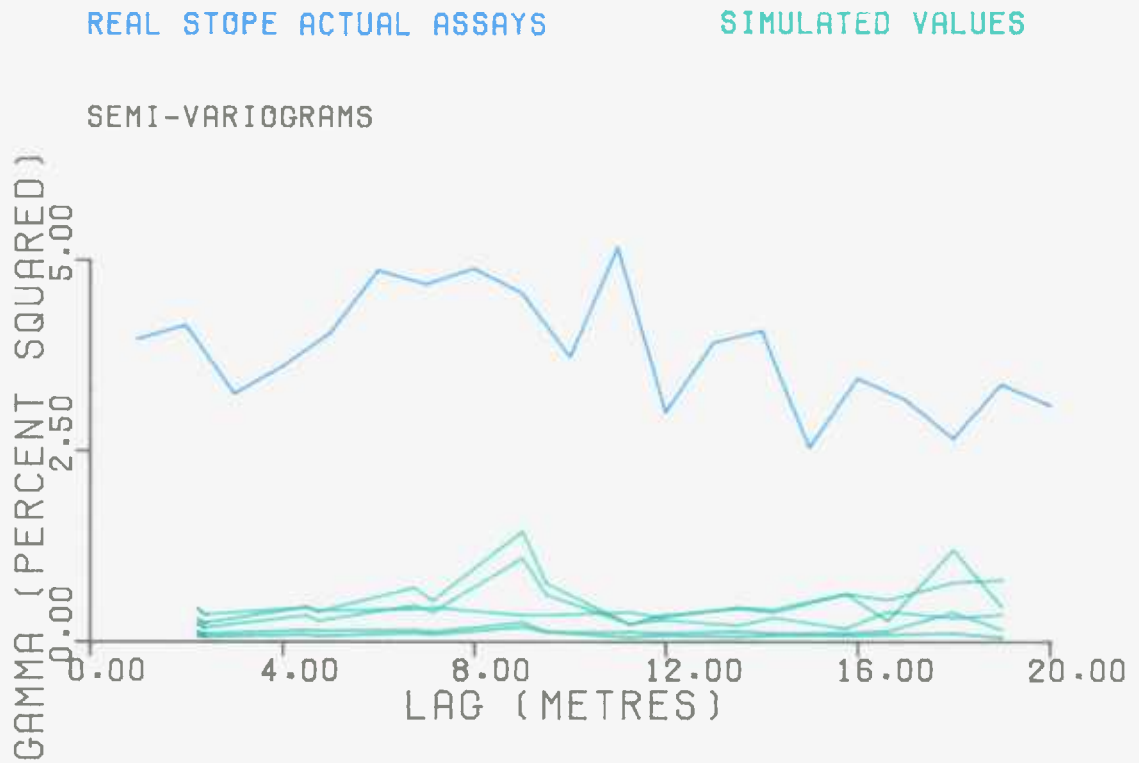


FIGURE 61 : CONDITIONAL SIMULATIONS
OF ACTUAL ASSAYS FOR MEREDREW STOPE



semi-variogram or distribution shapes (see figure 60). In particular, the distributions are of a very different shape with no simulated value below 1.60 metres. This is a result of the actual width estimator. Its operation dominates the whole procedure, as shown by the great similarity amongst the five distributions. It produces values above 1.60 metres and with a low semi-variogram.

One of the faults of the actual width estimator is that, by necessity, it was derived from the stope samples. In other words, it is a function of a stope rather than a development lode width. As explained previously (see section B.2.6), and as can be seen in figure 53, the stope and development lode width samples behave differently. The simulated actual widths are derived from the grid values of lode width. These lode widths are simulated using the development auto-correlation and distribution models. Thus, an unavoidable and unknown error enters the operation.

It was decided that to produce simulations of actual widths was not practicable. The best estimator (explaining 66 percent of the TSS) is not perfect, and was calculated from the stope samples. These facts, together with the poor results, led to the decision to drop the actual width simulations. It was considered that for any given stope, a lode width simulation would suffice. If the actual widths are needed they could be estimated by the mine personnel. The particular circumstances of any stope are unique. As a result, the prediction of overmining and dilution can be

best carried out by an experienced miner, rather than by a polynomial equation.

The results from the actual assay simulations (see figure 61) seem to be acceptable. The estimated actual assay at each point was derived from the simulated lode assay there. Consequently the low semi-variograms of the lode assay simulations (caused by the proportional effect) have been transferred. This is as desired since the average actual assay from all the stope samples was calculated to be 1.42 percent tin. The stope samples within Meredew stope show it to be of low grade (0.41 percent tin). Therefore, the simulations have tended to show low averages, and their semi-variograms are correspondingly lower than the lode average.

The actual assay distributions are all of the correct shape, with the predictable smaller spread. The errors caused by the actual width estimator have been avoided by the accuracy (83 percent of the data TSS), and the simplicity of the actual assay estimator.

Although the simulated actual assays behave in the correct manner, it was decided not to proceed with actual assay simulations. As a result of the inaccuracies of the actual width simulations, they would have had to be continued in isolation. To calculate the average assay of any area does not involve any width figures (see section B.2.3). However, it was felt to be unwise not to include width simulations. The anticipated width can have a great

influence on the decision whether and how to mine an area. A lode assay of 2 percent tin is much more useful if it occurs over a width of 1.0 metres rather than 0.2 metres.

Due to the failure of the actual width simulations, the lode width and lode assay simulations were continued. If actual, or stope, values are needed they could be produced. The actual assays would be simulated, and the actual widths estimated manually.

B.4. BLOCK SIMULATION

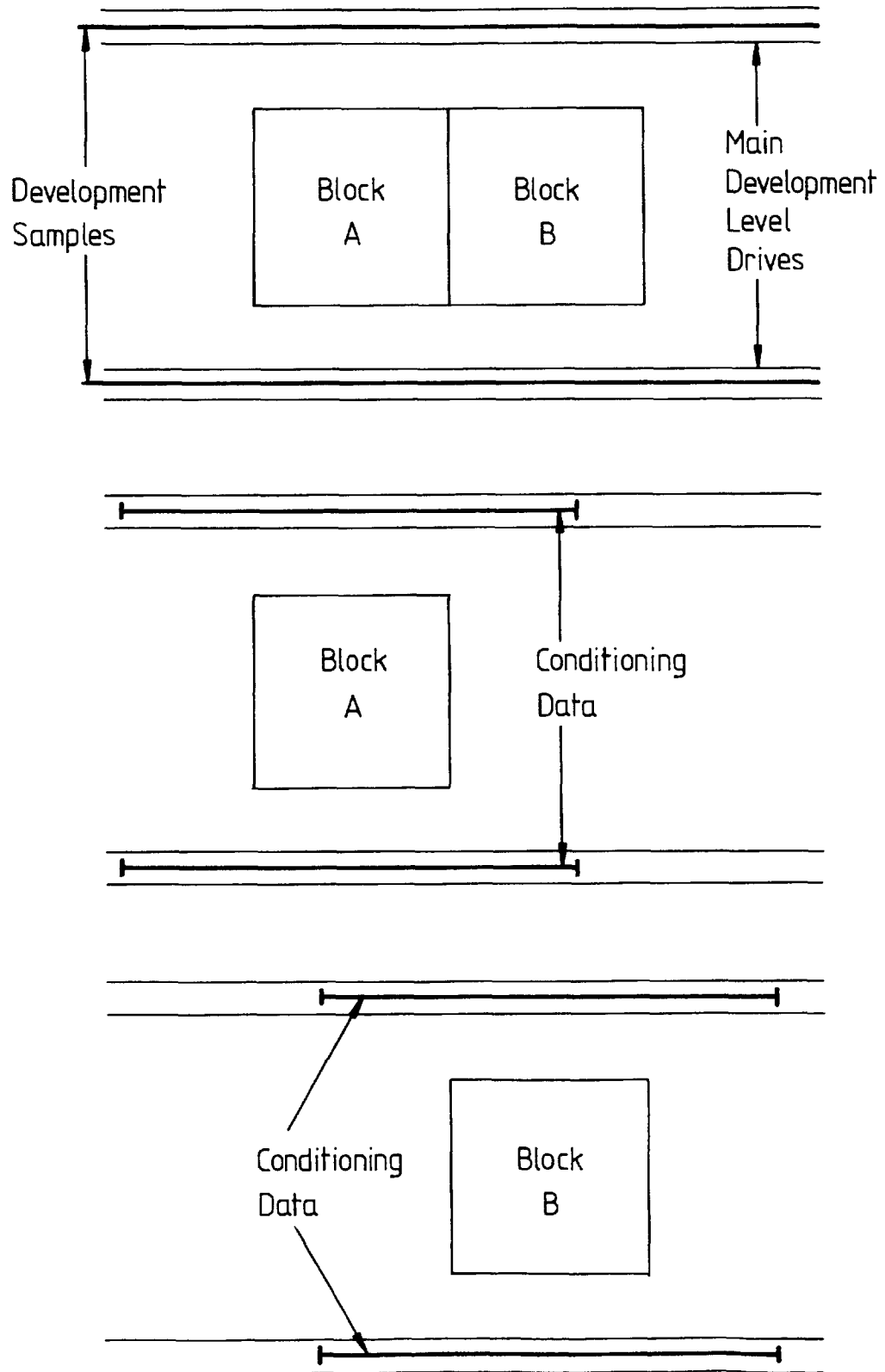
B.4.1. General approach :-

In section B.3.6 the cost of conditionally simulating stope lode widths and assays was shown to be high. In an attempt to reduce the expense a new approach was developed.

Consider the simulation of a rectangular area within No.9 lode, approximating to a stope (see figure 62). The block lies between two main development level drives and is of a known size and shape. A simulation of it can be produced and conditioned to data from the development levels. To do this an unconditional simulation is made of a grid of values representing the block, and also values at the position of every conditioning data sample (see section A.4.1). The conditioning data is in a known position relative to the block. The only difference between this block and one horizontally adjacent to it is in the conditioning data. However, since the development level samples are at a constant spacing, this data is in the same position relative to the block. So, the same unconditional simulation can be used for both blocks. The only difference between them arises from the different values of the conditioning data.

By this means, one unconditional simulation can be used for many block simulations. The expensive procedure of producing the simulated values does not have to be carried out each time.

FIGURE 62 : CONDITIONAL SIMULATION OF TWO
HORIZONTALLY ADJACENT BLOCKS



It was decided to carry out block simulation of this nature within No.9 lode. The precise layout was selected to be as shown in figure 63. The 310 and 335 levels were chosen because they had been more extensively sampled than the others. The number of sample results available from them is 473 and 321 respectively, out of a total of 1149 main development level samples.

B.4.2.Choice of block size :-

As can be seen from figure 63 the block was chosen to be 47.0 metres high and 25.0 metres wide. 47 metres is the average difference in depth between 310 and 335 level drives. In fact they range between 46 and 48 metres apart. The block width of 25 metres was selected as being a typical width for a stope within No.9 lode.

All available development samples up to 25 metres from the block edges were used to condition the simulations.

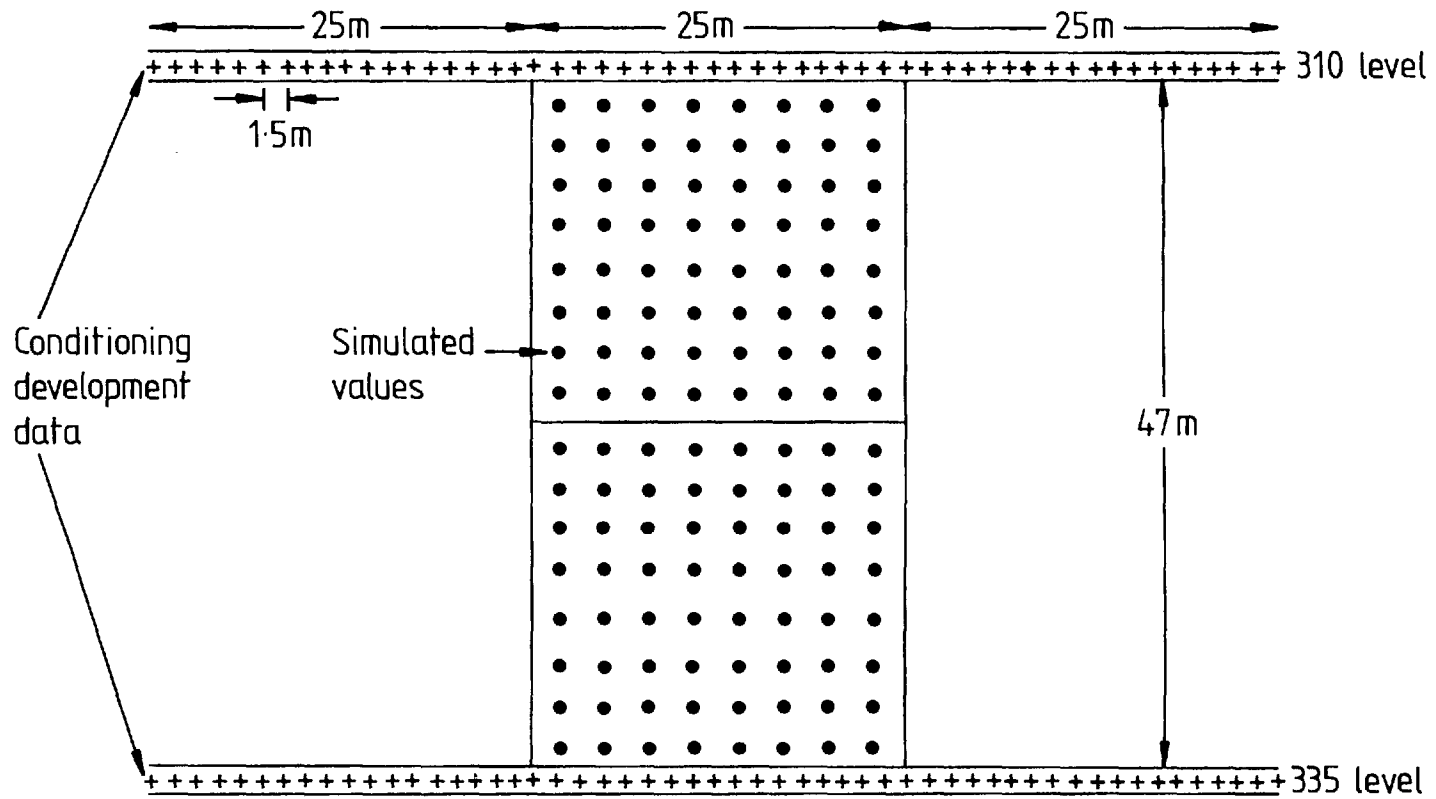
B.4.3.Choice of number of grid points :-

As shown in figure 63, the block was represented by a 16*8 grid of points. This was selected so that the simulated area could be split into two equal blocks, both 23.5 metres high, 25.0 metres wide and represented by a 8*8 grid.

The primary point of interest of each of the two blocks was its average. So, the average of the 64 simulated grid values has to give a sufficiently accurate idea of the real block average.

FIGURE 63 :

BLOCK CONDITIONAL SIMULATION LAYOUT



A better estimate of the true block average could be obtained using kriging. However, it was considered that the extra calculations involved would not produce a markedly more accurate estimate than the average of 64 evenly spaced samples.

The simulated values of ln lode width originated from a known semi-variogram model. Using its parameters, and standard geostatistical auxiliary functions (Clark, 1976), the estimation variance of the arithmetic mean of the 64 simulated values was found. It was calculated to be 0.00007. This means that the average of the 64 logarithmic values is 90 percent likely to lie within 0.0138 of the real block logarithm average. In other words if the average is 'M' the 90 percent limits are at 'M-0.0138' and 'M+0.0138'. Conversion of these figures to units of metres is achieved by taking their exponentials. Comparison with the exponential of 'M' shows the 90 percent limits to be at -1.4 and +1.4 percent. These two figures give some idea of the accuracy of the arithmetic mean of a 8*8 grid under these conditions.

Similarly, the accuracy of the mean of the 64 simulated lode assay values can be determined. The mean is an estimate of the real block lode assay average. The estimation variance of this mean was found to be 0.00026. Proceeding as with the widths, the final 90 percent limits were found to be at -2.6 and +2.7 percent.

Repetition of this process for a 7*7 grid produces 90

percent limits (on the lode assay average) at -3.2 and +3.3 percent. A 9*9 grid has similar limits at -2.2 and +2.2 percent.

Primarily considering the wider average lode assay limits a 8*8 grid was felt to be satisfactory. The real average lode assay of the block is 90 percent likely to lie between 97.4 and 102.7 percent of the mean of the 64 simulated values. For instance, if the simulated values have a mean of 0.5 percent tin these limits become 0.49 and 0.51 percent tin.

B.4.4. Production and storage of unconditional simulations :-

One hundred unconditional lode width simulations of the block layout were produced. Each simulation consisted of a value at each of the 128 grid points, and at the 100 conditioning development data points.

One hundred unconditional simulations of lode assay were also produced.

All the simulation results were stored on computer files. Each of the simulated values was recorded to two decimal digits. The total computer storage facility required for all 45,600 figures was 334,080 characters.

To condition the simulated point values a kriging system for each point has to be formed (see section A.4.1). If all the conditioning data is in the same relative position for every block layout, the kriging system for any

point would not change from layout to layout. The kriging weights would be identical, and therefore also the kriged estimate from the simulated development data ('Sk'). Section A.4.1 states that each conditional simulation value is found from :-

$$S = Sk + Rk$$

where S = unconditional simulation value

Sk = kriged estimate from simulated development data

Rk = kriged estimate from real development data

Now it has just been shown that for any point the value of 'Sk' does not vary with different sites of the block layout. Since the simulation value of 'S' is also constant the obvious procedure is to store values of 'S-Sk' rather than 'S'. For each simulation this would avoid the need to store 100 simulated development data values and also the calculation of 128 values of 'Sk'.

As shown in figure 63 fifty consecutive development data samples from each level drive are required to condition the simulations. These samples are sited at regular intervals along the drive of 1.5 metres (or 5 feet). It was rare for the results from fifty such consecutive sampling points to be available. Either no recordings could be taken because of intersecting dykes (see section B.1.1) or the sampling interval was 3.0 metres (or 10 feet).

If a block layout was sited where some of the development data are missing, new problems arise. The

kriging system for several grid points would include a sample which at that particular block layout position is absent. The missing results would have to be replaced by some means, possibly interpolation.

It was considered that, due to the problems caused by missing development data, the idea of storing 'S-Sk' values had to be dropped. The procedures for resolving these problems would create too many inaccuracies.

B.4.5. Choice of kriging pattern :-

To condition the simulations a kriging system for each of the 128 grid points has to be formed.

Obviously the simplest method would be to use all of the conditioning development samples in every kriging system. Equally obviously this would be exceedingly expensive since each system would consist of one hundred and one simultaneous equations. David (1977) states that the cost of solving simultaneous equations is proportional to the cube of the number of equations. In other words to solve one hundred and one equations costs over one thousand times times that for ten equations.

Using all the development samples in every kriging system produces the most accurate kriging estimates ('Sk' and 'Rk' values). However, a large number of these samples can be ignored with little loss in accuracy.

The kriging pattern used for each of the 128 simulation grid points was developed. It consisted of

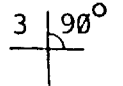
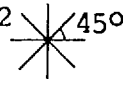
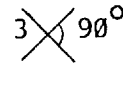
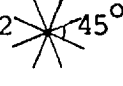
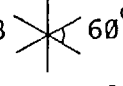
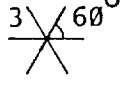
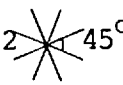
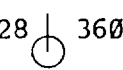
finding all development data within a search circle with a radius equal to the range of the semi-variogram. To avoid the 'shadow effect' (David,1977) the points within this circle were examined and only the nearest three in a 60 degrees segment were retained. Therefore there could not be more than eighteen points in the kriging system. If there were less than four points the search circle radius was increased, and the whole process repeated. This is the method used in the program CONSIM1 (see section B.3.2).

Apart from the 'nearest three in 60 degrees segments' technique several other methods were examined. The results are shown in table 7.

The choice of 'nearest three in 60 degrees segments' (method (f) in table 7) was made because of its extra accuracy over method (e). The maximum kriging variances are both reduced by 3.7 percent, and the averages by 1.0 percent. However, a further expense of 12 CP seconds only causes similar reductions of 0.4 and 0.7 percent. The method (f) was felt to be at the point of balance between cost and accuracy. It also makes full use of the data, with the outermost samples (25 metres horizontally from the corners) being used three times.

It is interesting to note the results obtained if all 100 of the development samples are used in every kriging system (method (h) in table 7). Compared with the adopted method an extra cost of 4274 percent results in decreases in averages and maximums of only 2.6 and 3.1 percent

TABLE 7

Method	Kriging pattern	No. of times outermost points used	Kriging variances				Execution time in CP seconds
			Lode width		Lode assay		
			Average	Maximum	Average	Maximum	
(a)		0	.2189	.2829	1.652	1.881	41
(b)		3	.2163	.2791	1.638	1.858	52
(c)		1	.2160	.2833	1.631	1.891	43
(d)		2	.2153	.2743	1.633	1.840	54
(e)		3	.2148	.2829	1.616	1.831	58
(f)		3	.2123	.2722	1.604	1.812	62
(g)		2	.2109	.2707	1.592	1.805	74
(h)		128	.2057	.2609	1.571	1.775	2712

respectively.

It has been stated (see section A.5.6), that simulations formed with ten regularly orientated one dimensional axes may be of use where the average kriging variance is less than one third of the sill. The averages achieved were at 62 and 71 percent of the model semi-variogram sill. Thus fifteen regularly orientated axes were necessary to obtain accurate final simulations.

B.4.6. Conditional simulation results :-

A computer program, called SIM2BLO (see Appendix 13), was written to condition the one hundred unconditional simulations to the appropriate development data.

The values of the one hundred conditioning samples are read into the program. For the almost inevitable missing samples zero values for lode width and assay are entered. The kriging system for every one of the grid points is solved, and the weights stored on a temporary file. In turn, each set of unconditional simulated values is read from the permanent storage files. For every one of the simulation grid points its kriging weights are found and used to condition that value. Once all these values have been conditioned the next unconditional simulation is read in and the process repeated.

Table 8 summarises some of the output of the program SIM2BLO. The averages from all one hundred simulations are shown. Also displayed, for various pay grades, are the

TABLE 8

Simulated blocks = 50 to 125 East

Average of all 100 simulations	Lower block		Upper block	
	Width	Assay	Width	Assay
	1.71	1.37	1.77	0.97

Lower block/Upper block P = Pay N = Non-pay								
Pay grade	Average assay of blocks				Value of rest of block if 2 non-pay lifts occur			
	P/P	P/N	N/P	N/N	P/	N/	/P	/N
0.1	100	0	0	0	0	0	0	0
0.3	98	2	0	0	1	1	5	2
0.5	93	7	0	0	6	11	32	9
0.7	61	27	4	8	8	23	24	45
0.9	42	37	5	16	12	32	15	58
1.1	22	36	6	36	19	53	13	79
1.3	13	29	6	52	16	66	7	89
1.5	4	29	7	60	14	73	5	91

number of occurrences of both blocks being in the same category (pay or non-pay), and of the blocks being in different categories. The use of such figures is best demonstrated by those at a pay limit of 0.9 percent tin.

With a pay grade of 0.9 percent tin, the most likely situation (having occurred 42 percent of the time) is that both blocks are in the pay category. However, there is an equal chance that the blocks are in different categories. It would be prudent to use a mining technique capable of extracting only one of the blocks. The first step should be to mine the lower block on its own. If it is found to be above the pay limit, the odds are slightly (42 to 37) in favour of the upper block also being pay. Extraction of the upper block should then proceed.

The obvious method suggested by such an order of extraction is overhand stoping. Under different circumstances underhand stoping may be found to give the best opportunity to mine only profitable blocks.

The stoping method used at South Crofty mine involves the extraction of the lode in horizontal strips, or lifts. Each lift advances the stope face by about 3 metres. Samples are taken along the stope face after extraction of every lift. If the averages of two consecutive lifts are both below the pay grade the stope is stopped.

The vertical interval between the points of the simulation grid is 2.94 metres. Thus, each horizontal line of eight simulated values represents an extraction lift.

The average of the simulated lode assays is an estimate of the real lift assay value. As before (see section B.4.3), the 90 percent limits of this estimate can be calculated. They were found to be at 92.7 and 107.8 percent of the estimate.

It was decided to test the validity of the practice of halting a stope if two consecutive non-pay lifts occur. For every simulation produced by the program SIM2BLO the two vertically adjacent blocks were considered separately. The average assay of each of the eight lifts was estimated by the mean of the appropriate horizontal line of eight simulated values. If two adjacent non-pay lifts were found, the mean of the remaining simulated values above them was calculated. It was noted whether this was below or above the pay grade. In other words, whether for that particular block simulation the 'two non-pay lifts' criterion had saved mining an unprofitable area. Table 8 shows the results for all one hundred simulations at various values of pay grade.

The figures show that the 'two non-pay lifts' criterion works for the lower block when the pay grade is 0.5 percent tin. Seventeen simulations were found where there were two consecutive lifts with average assays of less than 0.5 percent tin. In eleven (65 percent) of these seventeen cases the rest of the lower block was determined to be non-pay. So, the existence of a non-pay area had been correctly anticipated.

The 'two non-pay lifts' criterion does not work for the upper block at a 0.5 percent pay grade. For this block if two non-pay lifts occur there is only a 22 percent (nine out of forty one) probability that the rest of the block averages below 0.5 percent tin.

The program SIM2BLO was re-run but using a 'three non-pay lifts' criterion to signal poor areas. The equivalent figures to those of table 8 were found to be as follows :-

Pay	P/P	P/N	N/P	N/N
0.1	0	0	0	0
0.3	0	0	0	2
0.5	1	5	7	7
0.7	0	11	8	28
0.9	4	26	7	67
1.1	5	34	8	72
1.3	4	48	5	82
1.5	5	60	3	88

These indicate that for a pay grade of 0.5 percent tin the 'three non-pay lifts' criterion works for the upper block. There is a 50 percent (seven out of fourteen) chance that the rest of the block is not worth mining.

A 'one non-pay lift' criterion was tested. The results were found to be :-

Pay	P/P	P/N	N/P	N/N
0.1	0	0	3	0
0.3	15	2	29	2
0.5	29	14	70	10
0.7	45	26	52	44
0.9	51	36	31	67
1.1	43	50	20	79
1.3	32	66	10	89
1.5	27	73	8	92

These figures show that that using the 'one non-pay lift' criterion is insufficient for either block at a pay grade of 0.5 percent tin. The probabilities of it correctly forecasting a non-pay area are only 33 and 13 percent, for the lower and upper blocks respectively.

So, if the pay grade is 0.5 percent a 'two non-pay lifts' criterion is required to correctly signal a non-pay portion of the lower block. If this is used for the upper block there is a 78 percent probability that a profitable area is being left. Three consecutive non-pay lifts are needed before the rest of the upper block can be correctly regarded as not worth mining.

B.4.7. Sensitivity of the block simulation method :-

The method used to produce block simulations (performed by the program SIM2BL0) is not as exact as that for stope simulations (carried out by CONSIM1 and CONSIM2). Various assumptions have been made in order that the

conditional simulations are much cheaper to produce. Inevitably these cause inaccuracies in the final results. It was necessary to find out whether the balance between cost and accuracy had been swung too far in favour of cost.

Table 8 shows the results of block simulations for a certain area of No. 9 lode. The averages of the one hundred simulations for this area (comprising both blocks) are 1.74 metres and 1.17 percent tin. This same area was simulated by the programs CONSIM1 and CONSIM2. The averages achieved by these simulations were :-

Simulation 1	Lode width = 1.92	Lode assay = 2.04
" 2	" 1.25	" 0.98
" 3	" 1.27	" 1.19
" 4	" 1.59	" 0.71
" 5	" 2.07	" 1.80

These figures show that the results of the one hundred block simulations lie well within the bounds of possibility outlined by the five stope simulations. Thus, the assumptions made by SIM2BLO have not created any large bias in the results.

Table 9 displays the results of an initial sensitivity analysis on the simulations formed by SIM2BLO. The first set of results shown (situation (a)) are from the same execution of SIM2BLO as those in table 8.

The conditioning data used in the block simulations consists of one hundred sample points, fifty from each level drive. These samples are treated as being exactly

TABLE 9

Simulated blocks = 50 to 125 East

Subscript L = Lower block U = Upper block

Situation	Average block values				Number of occurrences at pay = 0.9% Lower/Upper				Average correlation between 128 simulated grid point values Width vs Assay	Correlation between 100 blocks	
										Width _L vs Width _U	Assay _L vs Assay _U
	Width _L	Assay _L	Width _U	Assay _U	P/P	P/N	N/P	N/N			
(a)	1.71	1.37	1.77	0.97	42	37	5	16	0.06	0.70	0.45
(b)	1.71	1.36	1.76	0.94	39	38	5	18	0.06	0.70	0.45
(c)	1.70	1.36	1.74	0.95	39	38	5	18	0.06	0.70	0.44
(d)	1.70	1.34	1.74	0.91	37	38	6	19	0.06	0.70	0.43
(e)	1.69	1.31	1.72	0.90	37	37	6	20	0.06	0.70	0.43
(f)	1.67	1.27	1.71	0.90	32	39	7	22	0.06	0.70	0.44
(g)	1.71	1.34	1.77	0.95	41	37	6	16	0.06	0.70	0.44
(h)	1.79	1.36	1.82	1.13	33	34	11	22	0.07	0.23	0.26
(i)	1.74	1.45	1.78	1.09	33	37	10	20	0.05	0.23	0.23
(j)	1.71	1.51	1.77	0.97	40	42	4	14	0.06	0.70	0.38

regularly spaced at an interval of $75/49 = 1.53$ metres (or 5.02 feet). Obviously, this is never actually found to be true. However, it is not possible to accurately determine an interval between any two sample points. This is because it was felt that the co-ordinates of the points can not be found, from the longitudinal section of the lode, to any greater accuracy than 1 metre. To assume precise regularity of samples along the drives was considered to be valid. On average it was found that the difference in Easting between two points forty nine sample intervals apart was 75 metres. The Eastings of the outermost samples used in situation (a) were 52 and 124 metres (335 level) and 50 and 123 metres (310 level). These co-ordinates were determined from the longitudinal section and as such are not precise. For instance, the samples regarded as being at 50 metres East may really be at 49 metres. To test the possible effect of this the data from 310 level was altered.

Situation (b) used fifty conditioning data points from 310 level one sample further East than those for situation (a). In other words, the recorded Eastings ranged between 52 and 125 metres. Similarly, situation (c) used samples between 53 and 126 metres East, situation (d) between 54 and 127 metres, and situation (e) between 56 and 129 metres. The results show that any errors made in determination of the sample co-ordinates would not have a large effect. Even if the conditioning data used differs by a distance of four sample intervals (situations (a) and

(e) the results are very similar. The block simulation method seems to be robust as far as the data co-ordinates are concerned. The necessary approximations can be made with confidence.

As stated before (see section A.4.4), the conditioning process is the great weakness of the 'turning bands' simulation method. This is because it attributes 100 percent reliability to all the sample data values. To a large extent this problem was overcome by rounding the data measurements before use (see section B.2.1).

To test the influence of the accuracy of the data, a random error was added to each measurement. Every sample was given a 5 percent chance of being incorrect by at least one measurement unit. In other words, a measurement of 1.40 metres had a 5 percent chance of being changed to less than 1.30 metres or greater than 1.50 metres. This was achieved by adding a random number from a Normal distribution of $N(0.0 \text{ metres}, 0.05^2)$. For metric lode assays a number from $N(0.0 \text{ percent tin}, 0.005^2)$ was added. This created a 5 percent chance of the samples being wrongly assayed by at least 0.01 percent tin. Similarly, for the imperial measurements random numbers from $N(0.0 \text{ feet}, 0.125^2)$ and $N(0.0 \text{ lbs/ton}, 0.5^2)$ were added, for widths and assays respectively.

So, each sample measurement was given a reasonable allowance for a random measuring error. The results of this operation are shown under situation (f) in table 9. The

averages of all the block values have not changed by all that much. This is as expected since the mean of each of the random error distributions was zero. There is a drop (of 24 percent) in the number of occurrences of both blocks being above the pay grade.

The results of situation (f) originate from adding a measurement error to the raw sample values. Another approach is to round all the sample measurements, convert them to metric units, and then to add a random error. Situation (g) depicts the results of such a procedure. A random number from a Normal distribution of $N(0.0 \text{ metres}, 0.1^2)$ was added to every lode width sample. Thus, there was a 5 percent chance of a width being changed by at least 0.1 metres. For the lode assays using a $N(0.0 \text{ percent tin}, 0.025^2)$ distribution gave a similar probability of the value being altered by at least 0.025 percent tin. The results are very similar to those of situation (a). No noticeable change has been caused by these simulated random measurement errors.

Overall the block simulation method seems robust to inaccuracies in the conditioning data. So long as any measurement error is random and unbiased it can be confidently accommodated.

Situations (h) and (i) in table 9 represent the results when conditioning with every other development sample. These results would be obtained if only 3 metres (or 10 feet) sampling had been practised. They show that

the simulation method is fairly resistant to the loss of half its conditioning data. However, the average block assays are changed markedly. In particular, the mean of the one hundred upper block average assays is increased by 16 and 12 percent. Despite this there is a marked decrease in the number of occurrences of both blocks being above the pay grade. This means that the distribution of the upper block average assays has changed shape. Its average has increased as well as the proportion below 0.9 percent tin. Since 0.9 percent tin is less than the average this indicates that the distribution spread has become larger. There are other effects of only using half the available conditioning data. These include the great reductions in the correlations between the averages of the two blocks. As a result of all these changes it was considered inadvisable to use the block simulation method in areas where 3 metres (or 10 feet) development sampling had been employed. The lack of conditioning data could create significant inaccuracies in the results.

It has been mentioned (see section B.2.3), how the average assays were calculated throughout this study. They were defined as the arithmetic mean of the assay values under consideration. This was in preference to the commonly used accumulation based definition, which states that the average assay of a set of values is equal to the average accumulation divided by the average width. Situation (j) in table 9 depicts the results obtained when the second

definition was used. Relative to the results using the first definition (situation (a)), the lower block average assay has increased by 10 percent. Apart from this there are no substantial differences. The use of one definition in preference to the other does not cause any important changes in the final results.

B.4.8. Costs of the simulations :-

The cost of producing the unconditional simulations, and running them through the program SIM2BLO, is obviously dependent on the number of them. One hundred was chosen arbitrarily as a large, but manageable, number. There must be enough simulations to give a good idea of the parameter distributions. To test whether one hundred is enough the figures in table 10 were calculated. They were derived from the results of one hundred simulations obtained under situation (a) of table 9.

The one hundred lower block average lode widths were split into groups of equal size. The minimum and maximum deviations from the overall average, for that particular group size, were found to be as shown. This process was repeated for various group sizes, and for the other three block averages.

The results of table 10 indicate that fifty simulations suffice. The averages are extremely close to those of one hundred simulations. The program SIM2BLO was executed using only fifty simulations. The results were

TABLE 10

Simulated blocks = 50 to 125 East

Number of simulations in groups	% difference from average of 100 simulations			
	Lode width		Lode assay	
	Min	Max	Min	Max
100	0	0	0	0
50	2	2	0	0
33	1	3	3	8
20	0	7	1	16
10	4	17	2	22

found to be as close as possible to those obtained with one hundred simulations.

The cost of producing fifty simulations was found to be 643.7 CP seconds. The total storage required was 167,040 characters. To execute SIM2BLO with fifty simulations cost 228.4 CP seconds. Thus, the total cost for fifty simulations of one block layout was 872.1 CP seconds. At current (October, 1979) commercial rates this is about 192 pence per simulation. It compares extremely favourably with the 16 pounds Sterling per simulation achieved by the stope simulation method (see section B.3.6). When it is considered that the number of simulated values has been increased from eighty one to one hundred and twenty eight the reduction in cost is enormous. The justification of the block simulation approach because of its lower cost has been proved.

One of the major features of the block simulation technique is that the same unconditional simulations can be used for different block layout sites. For instance, ten adjacent block layouts between 310 and 335 levels could be run through SIM2BLO. The total cost to produce fifty conditional simulations for each layout would be $643.7 + 10 * 228.4 = 2927.7$ CP seconds. Per block layout this is a cost of 292.8 CP seconds, or about 32 pounds Sterling. The expense per simulation is now only approximately 64 pence.

B.4.9. Conditioning with grouped data :-

The block simulation technique incorporated in the computer program SIM2BLO requires the results from one hundred sample points. All these measurements are used to condition the simulations. It was decided to test the effect of conditioning the simulations to groups of data. In other words, the conditioning data would consist of a number of sections of the development drives, each comprising several sample points. For instance, the one hundred samples could be split into ten groups of ten samples. Each group would represent a section, of 15.3 metres in length, along a development drive. By this means, the influence of a single exceptionally high value would be reduced.

The program SIM2BLO was adapted to carry out such an operation. Table 11 shows the results. The first set of results (one hundred groups of one sample) are those of situation (a) in table 9. The last grouping arrangement shown (with thirteen groups) involved the conditioning data being subjectively arranged. Each of the groups contained a different number of samples, and represented a length of a drive with a specific quality. This was either low width, or high width, or low assay, or high assay, or any combination of the four.

The results shown in table 11 indicate that conditioning with grouped data has not worked. Even with the conditioning data in fifty groups of two samples the

TABLE 11

Simulated blocks = 50 to 125 East

Subscript L = Lower block U = Upper block

Data groups		Average block values				Number of occurrences at pay = 0.9% Lower/Upper				Average correlation between 128 simulated grid point values Width vs Assay	Correlation between 100 blocks	
											Width _L vs Width _U	Assay _L vs Assay _U
Number	Size	Width _L	Assay _L	Width _U	Assay _U	P/P	P/N	N/P	N/N		Width _L vs Width _U	Assay _L vs Assay _U
100	1	1.71	1.37	1.77	0.90	42	37	5	16	0.06	0.70	0.45
50	2	1.57	2.00	1.72	1.17	61	33	1	5	0.06	0.70	0.37
25	4	1.51	1.93	1.75	1.11	52	42	0	6	0.03	0.71	0.42
20	5	1.50	2.47	1.83	1.40	78	18	1	3	0.02	0.70	0.43
10	10	1.42	1.60	1.80	1.35	71	12	11	6	0.01	0.71	0.59
5	20	1.42	1.16	1.88	1.17	39	9	24	28	0.01	0.72	0.34
4	25	1.46	1.89	1.86	1.32	66	17	5	12	-0.02	0.71	0.60
2	50	1.86	2.28	2.00	1.40	70	16	3	11	-0.01	0.73	0.65
13	-	1.53	1.68	1.95	1.23	62	28	4	6	0.01	0.72	0.50

differences from the 'correct' results are large. Once again the importance of the conditioning process has been emphasised. Any reduction in the amount of conditioning information causes a marked decrease in accuracy.

CONCLUSIONS

The simulation method used in this study was static and probabilistic. It was based on the geostatistical 'turning bands' technique, and essentially consisted of three operations :-

(a) Fifteen one dimensional simulations were formed. These consisted of point values regularly spaced along single axes. It was found that a spacing of one hundredth of the desired range of influence should be used.

(b) The one dimensional simulation axes were regularly orientated in space, and their characteristics transferred into a three dimensional simulation. A new method was developed for deriving the fifteen regular axes directions. It was found to be quicker than the previous method (Journel, 1974), and more accurate for simulating a line.

(c) The three dimensional simulation was conditioned to any available real data. At the data points the simulation value was forced to equal the real value. The conditioning process was found to be the most important operation in the whole simulation technique. Any changes in the number, position, and values of the conditioning data points caused significant differences in the final

simulated values. The correlation between a simulation's accuracy before and after conditioning was found to be insignificant when the average conditioning kriging variance was less than one third of the semi-variogram sill. Under these conditions, an inaccurate unconditional simulation was not necessarily an inaccurate simulation after conditioning.

The standard technique for transferring the characteristics of the one dimensional simulations into three dimensions is to use fifteen regularly orientated axes. This configuration's performance was compared with others, both before and after conditioning. A configuration of fifteen randomly orientated axes was too inaccurate under all circumstances. Using one hundred randomly orientated axes was prohibitively expensive. A ten regular axes configuration was found to be suitable where the conditioning process was strong, that is with an average kriging variance of less than one third of the semi-variogram sill. Under other conditions the fifteen regular axes configuration was the most appropriate.

A major requirement in geostatistics is for a measure of the reliability of a semi-variogram value. The minimum/maximum envelope after fifty random removals of ten percent of the data gave a good impression of the robustness of a semi-variogram.

Each semi-variogram value was calculated as the

arithmetic mean of a distribution. Trimming off the high values of these distributions was helpful in identification of the semi-variogram model parameters.

The simulation method was applied to No.9 lode in South Crofty tin mine. Since the sample measurements used were so important to the whole procedure, great attention was made to their accuracy. It was found that the measurements' accuracy could be determined using the histogram of their final digits. Grouping these digits until a Normal distribution was achieved eliminated the human bias imposed during sampling.

The logarithms of the development lode widths were found to follow a cubic trend surface. The standard geostatistical practice with non-stationary data, such as this, is to use the residuals of the trend. However, the global trend surface was inaccurate over small areas between the development level drives. As a result of this, the trend was ignored and the data was treated as being stationary. Conditional simulations of some stopes within No.9 lode justified this approach. The conditioning process forced the local characteristics of the global trend on the simulated values.

Conditional simulations of lode assays for No.9 lode stopes were produced. Again the conditioning process was found to be dominant. It imposed the local average on the simulations, and correspondingly altered their semi-variograms.

Both lode width and lode assay conditional stope simulations were resistant to change in the semi-variogram model parameters.

A relationship between lode widths and actual, or stope, widths was found. It was used to produce conditional actual width simulations of stopes within the lode. Due to the inaccuracies of the actual width predictor, the simulation results were unrealistic. An actual assay prediction technique was also determined. It was more accurate than the actual width predictor. The conditional stope simulations of actual assays were found to be realistic.

In any simulation approach it is vital to consider enough simulations. It is dangerous to draw conclusions from only a few sets of results. A block simulation technique was developed which produced several simulations at a relatively low cost. The major cost reducer was the fact that the unconditional simulations could be used for more than one block layout site. Fifty simulations of a single site were produced for a total cost of 32 pounds Sterling. The area covered by these simulations was likely to contain over 50 tonnes of tin (equivalent to 350,000 pounds Sterling at October 1979 prices). The simulations were of two vertically adjacent blocks between two development level drives. The technique calculated results concerning the relationship between these two blocks. The results were used to determine the extraction order likely

to be the most profitable. They were also utilised to justify the criterion for signalling the presence of a non-pay block.

The simulations produced by the block simulation technique were conditioned to sample data from the two adjacent development level drives. The results of the technique were found to be resistant to errors in the estimated positions of these samples. They were also largely unaffected by any unbiased errors in the data measurements. The block average lode assays were calculated as the arithmetic mean of sixty four simulated values. The alternative definition, as the average accumulation divided by the average width, was tested. The results from the block simulation technique were similar for both definitions.

Geostatistical simulation of potential stoping areas can be a very useful aid to mine planning. It is possible for its cost to be very small compared with the savings it may produce.

APPENDICES

- 1 - Subroutine SIM
- 2 - Subroutine COORDS1
- 3 - Program SIM3D
- 4 - Program CON
- 5 - Determination of the co-ordinates of any point
on ten regular axes
- 6 - Subroutine COORDS2
- 7 - Determination of the co-ordinates of any point
on fifteen regular axes
- 8 - Determination of sum cos alpha terms
for Journal's fifteen regular axes
- 9 - Subroutine COORDS3
- 10 - Program RANDO
- 11 - Program CONSIM1
- 12 - Program CONSIM2
- 13 - Program SIM2BLO

Appendix 1 : Subroutine SIM

The listing of the subroutine SIM is shown overleaf.

The required input to the subroutine is :-

A - The desired range of influence.

NS - The number of samples to be simulated

The output of the subroutine is :-

Y - An array containing 'NS' simulated values, which follow the one dimensional spherical semi-variogram model with a range of 'A', a sill of 1.0, and a nugget effect of 0.0.

```

00010 SUBROUTINE SIM(Y,A,NS)
00020 DIMENSION Y(2000),T(100)
00030 B=0.01*A
00040 WF=SQRT(12.0*B/(A*(A*A+11.0*B*B)))
00050 DO 50 M=1,100
00060 50 CALL RANDOM(T(M),1.0)
00070 DO 53 I=1,NS
00080 Y(I)=0.0
00090 DIS=-B/2.0
00100 DO 52 K=1,50
00110 DIS=DIS+B
00120 52 Y(I)=Y(I)+DIS*(T(50+K)-T(51-K))
00130 Y(I)=Y(I)*WF
00140 DO 54 M=1,99
00150 54 T(M)=T(M+1)
00160 CALL RANDOM(T(100),1.0)
00170 53 CONTINUE
00180 RETURN
00190 END
00200 SUBROUTINE RANDOM(S,SD)
00210 S=0.0
00220 IF(SD.EQ.0.0) RETURN
00230 DO 10 M=1,12
00240 10 S=S+RANF(0.0)
00250 S=(S-6.0)*SD
00260 RETURN
00270 END

```

Appendix 2 : Subroutine COORDS1

The listing of the subroutine COORDS1 is shown overleaf. The required input to the subroutine is :-

X - An array of 15 stores. The first 3 stores contain the X Y Z co-ordinates of the considered point.

The output of the subroutine is :-

X - The co-ordinate of the point on each of the 15 regular axes.

```
01640 SUBROUTINE COORDS1(X)
01650 DIMENSION X(15),R(3,3)
01660 DATA ((R(I,J),J=1,3),I=1,3)/0.5,-0.80901699,0.30901699,
01670+0.80901699,0.30901699,-0.5,0.30901699,0.5,0.80901699/
01680 N=0
01690 DO 1 K=1,4
01700 DO 2 L=1,3
01710 NL=N+L+3
01720 X(NL)=0.0
01730 DO 2 M=1,3
01740 2 X(NL)=X(NL)+R(M,L)*X(N+M)
01750 N=N+3
01760 1 CONTINUE
01770 RETURN
01780 END
```


Appendix 3 : Program SIM3D

The listing of the program SIM3D is shown overleaf.
The required input to the program is :-

SMAX - An array of 3 stores containing the dimension of the simulated block in the x y z directions.

NSEED - A four digit number used to trigger the random number generator.

A - The semi-variogram model range of influence.

C - The semi-variogram model sill.

ENUG - The semi-variogram model nugget effect.

AVER - The distribution model average.

VMIN - The first group endpoint in the histogram of the simulated values.

VMAX - The last group endpoint in the histogram of the simulated values.

NGRP - The number of groups in the histogram between 'VMIN' and 'VMAX'.

An example of the output of the program is given overleaf.

The output also includes :-

V - An array of 20*20*20 stores. It contains the unconditional simulation values. The dimensions of this array can be changed if a different shape is to be simulated (for instance 1*1000*1).

```

00100 PROGRAM SIM3D (INPUT=131B,OUTPUT=131B,
00110+TAPE6=INPUT,TAPE7=OUTPUT,TAPE4,TAPE3)
00120C
00130C THIS PROGRAM SIMULATES VALUES WHICH FOLLOW SPH(A,C,ENUG)
00140C AND N(AVER,C) .
00150C
00160C THE NUGGET EFFECT COMES FROM A RANDOM N(0,ENUG) .
00170C THE REST OF THE VARIATION (=CO=C-ENUG) COMES FROM N(AVER,CO)
00180C AND FOLLOWS SPH(A,CO,0) .
00190C ADDING THESE 2 TOGETHER PRODUCES VALUES WHICH COME FROM
00200C N(AVER,C) AND FOLLOWS SPH(A,C,ENUG) .
00210C
00220C TAPE3=OUTPUT FOR SIMULATED VALUES
00230C TAPE4=OUTPUT FOR SEMI-VARIOGRAMS, HISTOGRAMS ETC.
00240C TAPE5=INPUT FROM TELEX
00250C TAPE7=OUTPUT ON TELEX
00260C
00270C THIS PROGRAM IS TO BE RUN ON TELEX.
00280C
00290 COMMON XINT(3),NEXT(3),V(20,20,20),Y(2000),X(15),SMAX(3)
00300 READ(6,1000)NSEED
00310 1000 FORMAT(I4)
00320 XSEED=FLOAT(2*#NSEED+1)
00330 CALL RANSET(XSEED)
00340 DO 71 I=1,100
00350 71 T=RANF(0.0)
00360 WRITE(4,2000)
00370 WRITE(7,2000)
00380 2000 FORMAT(3X,MAX IN X,Y,Z DIRECTIONS 3F6.1*)
00390 READ(6,1001)(SMAX(I),I=1,3)
00400 WRITE(4,1001)(SMAX(I),I=1,3)
00410 1001 FORMAT(3F6.1)
00420 DO 80 I=1,3
00430 NEXT(I)=IROUND(SMAX(I))
00440 80 SMAX(I)=(SMAX(I)-1.0)/2.0
00450 WRITE(4,2001)
00460 WRITE(7,2001)
00470 2001 FORMAT(3X,RANGE,SILL,NUGGET EFFECT + AVERAGE 4F6.2*)
00480 READ(6,1002)A,C,ENUG,AVER
00490 1002 FORMAT(4F6.2)
00500 WRITE(4,1003)A,C,ENUG,AVER,NSEED
00510 1003 FORMAT(4(2X,F7.2) //5X,I4)
00520 CO=C-ENUG
00530 B=A/100.0
00540 DO 30 J=1,NEXT(1)
00550 DO 30 K=1,NEXT(2)
00560 DO 30 L=1,NEXT(3)
00570 30 V(J,K,L)=0.0
00580 DO 15 I=1,15
00590 WRITE(7,2003)I
00600 2003 FORMAT(3X,I AM NOW CONSIDERING AXIS NO. *,I2)
00610 DO 31 J=1,3
00620 X(1)=0.0
00630 X(2)=0.0
00640 X(3)=0.0
00650 X(J)=X(J)+1.0
00660 CALL COORDS1(X)
00670 XINT(J)=X(I)
00680 31 CONTINUE
00690 DMAX=SMAX(1)*ABS(XINT(1))+SMAX(2)*ABS(XINT(2))+SMAX(3)*ABS(XINT(3))
00700 NY=IROUND(2.0*DMAX/B)+2
00710 NORG=IROUND(-DMAX/B)
00720 CALL SIM(Y,A,NY,B)
00730 WRITE(4,2004)I,NORG,NY,DMAX

```

```

00740 2004 FORMAT('K=1,NGRP,NY,DMAX',I2,2I5,2X,F6.1)
00750 DO 19 J=1,NEXT(1)
00760 XJ=FLOAT(J-1)/SMAX(1)
00770 DO 19 K=1,NEXT(2)
00780 XK=FLOAT(K-1)/SMAX(2)
00790 DO 19 L=1,NEXT(3)
00800 XL=FLOAT(L-1)/SMAX(3)
00810 XC=XJ*XINT(1)+XK*XINT(2)+XL*XINT(3)
00820 NC=IROUND(XC/B)-NGRP+1
00830 V(X,J,K,L)=V(X,K,L)+Y(NC)
00840 15 CONTINUE
00850 SUMV=0.0
00860 SUMV2=0.0
00870 STD=SQRT(CO/15.0)
00880 SDENUG=SQRT(ENUG)
00890 XN=FLOAT(NEXT(1)*NEXT(2)*NEXT(3))
00900 DO 21 J=1,NEXT(1)
00910 DO 21 K=1,NEXT(2)
00920 DO 21 L=1,NEXT(3)
00930 CALL RANDOM(S,SDENUG)
00940 VX=V(J,K,L)*STD+S+AVER
00950 SUMV=SUMV+VX
00960 SUMV2=SUMV2+VX*VX
00970 21 V(J,K,L)=VX
00980 WRITE(3,2005)((V(J,K,L),L=1,NEXT(3)),K=1,NEXT(2)),J=1,NEXT(1))
00990 2005 FORMAT(10(1X,F6.3))
01000 CALL GAM(V,A,C,NEXT,ENUG)
01010 AV=SUMV/XN
01020 VARE=(SUMV2-SUMV*AV)/(XN-1.0)
01030 WRITE(4,2006)AV,VAR,XN
01040 WRITE(7,2006)AV,VAR,XN
01050 2006 FORMAT('AV,VAR,XN FOR V',F8.4,1X,F8.4,3X,F5.0)
01060 CALL DIST(V,NEXT)
01070 ENDFILE 3
01080 ENDFILE 4
01090 STOP
01100 END
01110C
01120 SUBROUTINE DIST(V,NEXT)
01130 DIMENSION V(20,20,20),NEXT(3)
01140 DIMENSION IHIST(50)
01150 DATA IAST/1A**/
01160 WRITE(7,2000)
01170 2000 FORMAT('WHAT ARE VMIN,VMAX,NGRP',2F6.2,I2)
01180 READ(6,1000)VMIN,VMAX,NGRP
01190 1000 FORMAT(2F6.2,I2)
01200 DO 20 I=1,50
01210 20 IHIST(I)=0
01220 VINT=(VMAX-VMIN)/FLOAT(NGRP)
01230 DO 10 J=1,NEXT(1)
01240 DO 10 K=1,NEXT(2)
01250 DO 10 L=1,NEXT(3)
01260 IF(V(J,K,L).LE.VMIN)GOTO 11
01270 IF(V(J,K,L).GT.VMAX)GOTO 12
01280 I4=IFIX((V(J,K,L)-VMIN)/VINT+0.99999)+1
01290 IHIST(I4)=IHIST(I4)+1
01300 GO TO 10
01310 11 IHIST(1)=IHIST(1)+1
01320 GO TO 10
01330 12 IHIST(NGRP+2)=IHIST(NGRP+2)+1
01340 10 CONTINUE
01350 MAX=0
01360 DO 15 I=1,NGRP+2
01370 15 IF(IHIST(I).GT.MAX) MAX=IHIST(I)

```

```

01380 D0 14 I=1,NGRP+2
01390 VUPPER=VMIN+FLOAT(I-1)*VINT
01400 IH=IHIST(I)*110/MAX+1
01410 WRITE(4,2002)VUPPER,IHIST(I),(IAST,J=1,IH)
01420 2002 FORMAT(*  =*,F6.2,I4,120A1)
01430 14 CONTINUE
01440 RETURN
01450 END
01460C
01470 SUBROUTINE SIM(Y,A,NS,B)
01480 DIMENSION Y(2000),T(100)
01490 WF=SQRT(12.0*B/(A*(A*A+11.0*B*B)))
01500 D0 50 M=1,100
01510 50 CALL RANDOM(T(M),1.0)
01520 D0 53 I=1,NS
01530 Y(I)=0.0
01540 DIS=-B/2.0
01550 D0 52 K=1,50
01560 DIS=DIS+B
01570 52 Y(I)=Y(I)+DIS*(T(50+K)-T(51-K))
01580 Y(I)=Y(I)*WF
01590 D0 54 M=1,99
01600 54 T(M)=T(M+1)
01610 CALL RANDOM(T(100),1.0)
01620 53 CONTINUE
01630 RETURN
01640 END
01650C
01660 SUBROUTINE COORDS1(X)
01670 DIMENSION X(15),R(3,3)
01680 DATA ((R(I,J),J=1,3),I=1,3)/0.5,-0.80901699,0.30901699,
01690+0.80901699,0.30901699,-0.5,0.30901699,0.5,0.80901699/
01700 N=0
01710 D0 1 K=1,4
01720 D0 2 L=1,3
01730 NL=N+L+3
01740 X(NL)=0.0
01750 D0 2 M=1,3
01760 2 X(NL)=X(NL)+R(M,L)*X(N+M)
01770 N=N+3
01780 1 CONTINUE
01790 RETURN
01800 END
01810C
01820 SUBROUTINE GAM(V,A,C,NEXT,ENUG)
01830 DIMENSION V(20,20,20),IGRA(100),NEXT(3)
01840 DATA IAST,IBLNK,IPLUS/1H*,1H ,1H+/
01850 D0 15 M=1,3
01860 RMS=0.0
01870 WRITE(7,2001)M
01880 2001 FORMAT(* M= *,I1)
01890 IF(NEXT(M).EQ.1) GO TO 15
01900 LAGLIM=NEXT(M)-1
01910 IF(LAGLIM.GT.200) LAGLIM=200
01920 D0 10 LAG=1,LAGLIM
01930 LIM=NEXT(M)-LAG
01940 GAMMA=0.0
01950 GO TO (20,21,22)M
01960 20 CONTINUE
01970 D0 11 I=1,LIM
01980 D0 11 J=1,NEXT(2)
01990 D0 11 K=1,NEXT(3)
02000 DIFF=V(I,J,K)-V(I+LAG,J,K)
02010 11 GAMMA=GAMMA+DIFF*DIFF

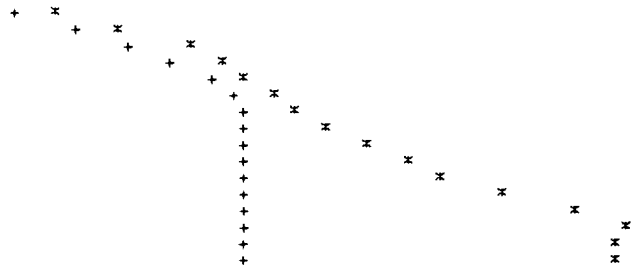
```

```

02020 COUN=FLOAT(LIM**NEXT(2)**NEXT(3))
02030 GO TO 25
02040 21 CONTINUE
02050 DO 31 I=1,LIM
02060 DO 31 J=1,NEXT(3)
02070 DO 31 K=1,NEXT(1)
02080 DIFF=V(K,I,J)-V(K,I+LAG,J)
02090 31 GAMMA=GAMMA+DIFF**DIFF
02100 COUN=FLOAT(LIM**NEXT(3)**NEXT(1))
02110 GO TO 25
02120 22 CONTINUE
02130 DO 41 I=1,LIM
02140 DO 41 J=1,NEXT(1)
02150 DO 41 K=1,NEXT(2)
02160 DIFF=V(J,K,I)-V(J,K,I+LAG)
02170 41 GAMMA=GAMMA+DIFF**DIFF
02180 COUN=FLOAT(LIM**NEXT(1)**NEXT(2))
02190 25 CONTINUE
02200 GAMMA=GAMMA**0.5/COUN
02210 SOFAR=FLOAT(LAG)
02220 HOVERA=SOFAR/A
02230 GMODEL=1.5**HOVERA-(0.5**HOVERA**HOVERA**HOVERA)
02240 GMODEL=GMODEL*(1-ENUG)+ENUG
02250 IF (HOVERA.GE.1.0) GMODEL=C
02260 RMS=RMS+((GAMMA-GMODEL)/GMODEL)**2
02270 IMODEL=FIX(GMODEL**50.0/C)+1
02280 IGAMMA=FIX(GAMMA**50.0/C)+1
02290 DO 12 I=1,100
02300 12 IGRA(I)=IBLNK
02310 IF (IGAMMA.GT.100) IGRA(100)=IPLUS
02320 IF (IGAMMA.LE.100) IGRA(IGAMMA)=IAST
02330 IGRA(IMODEL)=IPLUS
02340 WRITE(4,2000) SOFAR,GAMMA,GMODEL,(IGRA(I),I=1,100)
02350 2000 FORMAT(1X,F4.0,1X,F7.2,1X,F7.2,1X,*,*,100A1)
02360 10 CONTINUE
02370 RMS=100.0**SQRT(RMS/FLOAT(LAGLIM))
02380 WRITE(4,2003) RMS
02390 WRITE(7,2003) RMS
02400 2003 FORMAT(* RMS = *,F7.3,* PERCENT*)
02410 15 CONTINUE
02420 RETURN
02430 END
02440C
02450 SUBROUTINE RANDOM(S,SD)
02460 S=0.0
02470 IF (SD.EQ.0.0) RETURN
02480 DO 10 M=1,12
02490 10 S=S+RANF(0.0)
02500 S=(S-6.0)**SD
02510 RETURN
02520 END
02530C
02540 FUNCTION IROUND(X)
02550 Y=ABS(X)
02560 IROUND=FIX(Y+0.4999999)
02570 IF (X.LT.0.0) IROUND=-IROUND
02580 RETURN
02590 END

```


4. .64 .57+
 5. .78 .69+
 6. .90 .79+
 7. .97 .88+
 8. 1.01 .94+
 9. 1.07 .99+
 10. 1.11 1.00+
 11. 1.17 1.00+
 12. 1.25 1.00+
 13. 1.33 1.00+
 14. 1.40 1.00+
 15. 1.51 1.00+
 16. 1.64 1.00+
 17. 1.75 1.00+
 18. 1.73 1.00+
 19. 1.72 1.00+



RMS = 37.966 PERCENT
 AV,VAR,XN FOR V 9.9779 1.0243 8000.

```

- 8.00 193*****
- 8.10 55*****
- 8.20 68*****
- 8.30 84*****
- 8.40 80*****
- 8.50 126*****
- 8.60 139*****
- 8.70 131*****
- 8.80 145*****
- 8.90 187*****
- 9.00 177*****
- 9.10 181*****
- 9.20 217*****
- 9.30 260*****
- 9.40 272*****
- 9.50 290*****
- 9.60 308*****
- 9.70 289*****
- 9.80 286*****
- 9.90 266*****
- 10.00 295*****
- 10.10 319*****
- 10.20 296*****
- 10.30 272*****
- 10.40 287*****
- 10.50 277*****
- 10.60 266*****
- 10.70 260*****
- 10.80 221*****
- 10.90 237*****
- 11.00 205*****
- 11.10 169*****
- 11.20 182*****
- 11.30 153*****
- 11.40 129*****
- 11.50 118*****
- 11.60 93*****
- 11.70 86*****
- 11.80 68*****
- 11.90 63*****
- 12.00 48*****
- 12.10 182*****
  
```

Appendix 4 : Program CON

The listing of the program CON is shown overleaf. The required input to the program is :-

NSEED - A four digit number used to trigger the random number generator.

R - The semi-variogram model range of influence.

C - The semi-variogram model sill.

E - The semi-variogram model nugget effect.

AVER - The distribution model average.

GB - An array of 1000 stores. It contains all the data values from which the 'NDATA' conditioning data points are taken.

G - An array of 1000 stores. It contains the unconditional simulation values.

The output of the program includes :-

The histogram of each simulation before and after conditioning.

The semi-variogram of each conditioned simulation.

The average, variance, sill, and MS term of each simulation before and after conditioning.


```

00010 PROGRAM CON(OUTPUT,TAPE7=OUTPUT,TAPE2,TAPE4,TAPE62)
00020C
00030C
00040C THIS PROGRAM CONDITIONS SIMULATIONS TO 100 DATA POINTS WITHIN
00050C A LINE OF 1000 VALUES. THE DATA POINTS ARE LOCATED AS DEFINED
00060C IN THE ARRAY L0C0 AND HAVE VALUES AS READ FROM TAPE2.
00070C THE SIMULATIONS CONDITIONED TO THESE DATA POINTS HAVE BEEN
00080C PREVIOUSLY PRODUCED AND SAVED ON FILES.
00090C IN ITS PRESENT FORM THE PROGRAM USES 5 SIMULATIONS
00100C THE OUTPUT IS ON THE LINE PRINTER AND THE KINGMATIC.
00110C
00120C TAPE2=INPUT TAPE FOR ACTUAL AND SIMULATED DATA
00130C TAPE4=TEMPORARY STORE TAPE FOR KRIGING WEIGHTS
00140C TAPE7=OUTPUT TAPE FOR HISTOGRAMS AND RESULTS ETC.
00150C TAPE62=NOTIFICATION OF USE OF KINGMATIC DRAFTING MACHINE AND
00160C ITS SUBROUTINES OF START,AXIS,PLOT,NUMBER,LINE,ENPLOT.
00170C
00180 COMMON /DICK/L0C(100)
00190 COMMON /HARRY/GSEMI(60)
00200 COMMON/FRED/GACT(1000)
00210 DIMENSION GB(1000),G(1000),WGT(6)
00220C
00230 NDATA=100
00240 DO 121 I=1,NDATA
00250 121 L0C(I)=I*10
00260 L0CLOW=L0C(3)
00270 L0CHIGH=L0C(NDATA-2)
00280 CALL GET(5HTAPE2,5HR2123,7HUMCKA07)
00290 READ(2,1009)NSEED,R,C,E,AVER
00300 1009 FORMAT(3X,I4,44X,4F6.2)
00310 CALL WEIGHT(R,C,E,NDATA)
00320 CALL START(2)
00330 CALL AXIS(0.0,0.0,5HGAMMA,5,10.0,90.0,0.0,3.0)
00340 CALL AXIS(0.0,0.0,3HLAG,-3,15.0,0.0,0.0,4.0)
00350 CALL PLOT(1.0,12.0,3)
00360 CALL PLOT(2.0,12.0,2)
00370 CALL PLOT(1.0,12.0,2)
00380C
00390 READ(2,1000)(GB(I),I=1,NDATA)
00400 1000 FORMAT(10(2X,F6.2))
00410C
00420 WRITE(7,2003)R,C,E,AVER,NDATA
00430 2003 FORMAT(1H1//* R,C,E,AVER = *,4F6.2,* NDATA = *,I3)
00440 WRITE(7,2001)(L0C(I),I=1,NDATA)
00450 2001 FORMAT(/20(1X,I3))
00460 WRITE(7,2002)NSEED
00470 2002 FORMAT(5(/,49X,9HACTUAL (R,I4,1H)/49X,14(1H-))
00480 LH=NDATA-2
00490 CALL DIST(GB,GB,3,LH)
00500 DO 16 I=1,1000
00510 16 GACT(I)=-9.99
00520 DO 17 I=1,NDATA
00530 17 GACT(L0C(I))=GB(I)
00540 LL=L0C(3)
00550 LH=L0C(NDATA-2)
00560 CALL SEMIB(LL,LH)
00570C
00580 L50=4
00590 DO 99 KLM=L0CLOW+1,L0CHIGH-1
00600 IF(KLM.EQ.L0C(L50)) GO TO 98
00610 READ(4,1003)(WGT(I),I=1,6)
00620 1003 FORMAT(2X,6F6.4)
00630 T=0.0
00640 DO 97 J=1,6

```

```

00650 IG=L0C(LS0-4+J)
00660 97 T=T+GACT(IG)*WGT(J)
00670 GACT(KLM)=T
00680 G0 T0 99
00690 98 LS0=LS0+1
00700 99 CONTINUE
00710C
00720 2004 FORMAT(1H1)
00730C
00740 D0 13 LM=1,5
00750 REWIND 4
00760 IF(LM.EQ.1) CALL GET(5HTAPE2,5HR2406,7HUMCKA07)
00770 IF(LM.EQ.2) CALL GET(5HTAPE2,5HR2430,7HUMCKA07)
00780 IF(LM.EQ.3) CALL GET(5HTAPE2,5HR9949,7HUMCKA07)
00790 IF(LM.EQ.4) CALL GET(5HTAPE2,5HR6946,7HUMCKA07)
00800 IF(LM.EQ.5) CALL GET(5HTAPE2,5HR7281,7HUMCKA07)
00810C
00820 READ(2,1001)NSEED
00830 1001 FORMAT(3X,I4)
00840 READ(2,1002)(G(I),I=1,1000)
00850 1002 FORMAT(10(2X,F6.2))
00860 D0 14 I=1,1000
00870 14 GB(I)=G(I)
00880 LS0=4
00890 D0 10 KLM=L0CLOW+1,L0CHIGH-1
00900 IF(KLM.EQ.L0C(LS0)) G0 T0 100
00910 V=0.0
00920 READ(4,1003)(WGT(I),I=1,6)
00930 D0 11 J=1,6
00940 IG=L0C(LS0-4+J)
00950 11 V=V+G(IG)*WGT(J)
00960 G(KLM)=G(KLM)+GACT(KLM)-V
00970 G0 T0 10
00980 100 LS0=LS0+1
00990 10 CONTINUE
01000 D0 12 I=3,NDATA-2
01010 12 G(L0C(I))=GACT(L0C(I))
01020 WRITE(7,2000)NSEED
01030 2000 FORMAT(7(/,50X,*NSEED = *,I4/50X,12(1H-)//)
01040 CALL DIST(GB,G,L0CLOW,L0CHIGH)
01050 CALL SEMI(GB,G,L0CLOW,L0CHIGH,XS)
01060 XLM=FLOAT(LM)
01070 XPAGE=15.14
01080 CALL NUMBER(XPAGE,XS,0.07,XLM,0.0,-1)
01090 13 CONTINUE
01100C
01110 WRITE(7,2004)
01120 CALL ENPLOT
01130C
01140 STOP
01150 END
01160C
01170C+++++
01180 SUBROUTINE SEMIB(L0CLOW,L0CHIGH)
01190 COMMON/HARRY/GSEMI(60)
01200 COMMON/FRED/GACT(1000)
01210 DIMENSION S(63),H(63)
01220 XRMS=0.0
01230 RMS=0.0
01240 D0 10 LAG=1,60
01250 LIM=L0CHIGH-LAG
01260 GAM=0.0
01270 X=0.0
01280 D0 11 I=L0CLOW,LIM

```

```

01290 IF (GACT(I).EQ.-9.99.OR.GACT(I+LAG).EQ.-9.99) GO TO 11
01300 X=X+1.0
01310 DIFF=GACT(I)-GACT(I+LAG)
01320 GAM=GAM+DIFF*DIFF
01330 11 CONTINUE
01340 S(LAG)=-9.99
01350 IF (X.EQ.0.0) GO TO 10
01360 S(LAG)=0.5*GAM/X
01370 HOVERA=FLOAT(LAG)/50.0
01380 XMODEL=20.0*(1.5*HOVERA-0.5*HOVERA*HOVERA*HOVERA)
01390 DIFF=S(LAG)-XMODEL
01400 RMS=RMS+DIFF*DIFF
01410 XRMS=XRMS+1.0
01420 10 CONTINUE
01430 DO 13 I=1,60
01440 13 GSEMI(I)=S(I)
01450 RMS=SQRT(RMS/XRMS)
01460 WRITE(7,2000)RMS,RMS
01470 2000 FORMAT(44X,1H0,5X,14HRMS WPT ACTUAL,9X,1H0/
01480+40X,F5.2,5X,14HRMS WRT MODEL ,5X,F5.2)
01490 SILL=0.0
01500 X=0.0
01510 DO 12 I=50,60
01520 IF (S(I).EQ.-9.99) GO TO 12
01530 SILL=SILL+S(I)
01540 X=X+1.0
01550 12 CONTINUE
01560 SILL=SILL/X
01570 WRITE(7,2001)SILL,SILL
01580 2001 FORMAT(40X,F5.2,5X,14HEST. OF SILL ,5X,F5.2///)
01590 IX=0
01600 DO 20 I=1,60
01610 IF (S(I).EQ.-9.99) GO TO 20
01620 IX=IX+1
01630 S(IX)=S(I)
01640 H(IX)=FLOAT(I)
01650 20 CONTINUE
01660 DO 21 I=1,IX
01670 S(IX+2-I)=S(IX+1-I)
01680 21 H(IX+2-I)=H(IX+1-I)
01690 S(1)=H(1)=0.0
01700 IX=IX+1
01710 S(IX+1)=0.0
01720 S(IX+2)=3.0
01730 H(IX+1)=0.0
01740 H(IX+2)=4.0
01750 CALL LINE(H,S,IX,1,-1,1)
01760 YSCALE=1.0/3.0
01770 CALL ARKIST(H,S,1,IX,10,0.25,YSCALE,0.0,0.0,2,1)
01780 S(IX+1)=-10.0
01790 CALL LINE(H,S,IX,1,1,1)
01800C
01810 RETURN
01820 END
01830C
01840C-----
01850 SUBROUTINE SEMI(U,V,LOCL0W,LOCHIGH,XS)
01860 COMMON /HARRY/GSEMI(60)
01870 DIMENSION U(1000),V(1000),S(62),H(62),SB(60)
01880 DO 10 LAG=1,60
01890 LIM=LOCHIGH-LAG
01900 GAMTB=0.0
01910 GAMT=0.0
01920 DO 11 I=LOCL0W,LIM

```

```

01930 DIFF=U(I)-U(I+LAG)
01940 GAMMB=GAMMB+DIFF*DIFF
01950 DIFF=V(I)-V(I+LAG)
01960 GAMM=GAMM+DIFF*DIFF
01970 11 CONTINUE
01980 XN=FLOAT(LIM-L0CLOW+1)
01990 SB(LAG)=0.5*GAMMB/XN
02000 S(LAG)=0.5*GAMM/XN
02010 H(LAG)=LAG
02020 10 CONTINUE
02030C
02040 S(61)=0.0
02050 S(62)=3.0
02060 H(61)=0.0
02070 H(62)=4.0
02080 CALL LINE(H,S,60,1,0,3)
02090 XS=S(60)/3.0
02100 RMSAB=0.0
02110 RMSAA=0.0
02120 RMSMB=0.0
02130 RMSMA=0.0
02140 XP=0.0
02150 DO 13 LAG=1,60
02160 IF(GSEMI(LAG).EQ.-9.99) GO TO 13
02170 XP=XP+1.0
02180 DIFF=SB(LAG)-GSEMI(LAG)
02190 RMSAB=RMSAB+DIFF*DIFF
02200 DIFF=S(LAG)-GSEMI(LAG)
02210 RMSAA=RMSAA+DIFF*DIFF
02220 HOVERA=FLOAT(LAG)/50.0
02230 XMODEL=20.0*(1.5*HOVERA-0.5*HOVERA*HOVERA*HOVERA)
02240 DIFF=SB(LAG)-XMODEL
02250 RMSMB=RMSMB+DIFF*DIFF
02260 DIFF=S(LAG)-XMODEL
02270 RMSMA=RMSMA+DIFF*DIFF
02280 13 CONTINUE
02290 RMSAB=RMSAB/XP
02300 RMSAA=RMSAA/XP
02310 RMSMB=RMSMB/XP
02320 RMSMA=RMSMA/XP
02330 WRITE(7,2001) RMSAB,RMSAA,RMSMB,RMSMA
02340 2001 FORMAT(39X,F6.2,5X,14HRMS WRT ACTUAL,4X,F6.2/
02350+39X,F6.2,5X,14HRMS WRT MODEL,4X,F6.2)
02360 SILLB=0.0
02370 SILLA=0.0
02380 DO 14 LAG=50,60
02390 SILLB=SILLB+SB(LAG)/11.0
02400 14 SILLA=SILLA+S(LAG)/11.0
02410 WRITE(7,2002) SILLB,SILLA
02420 2002 FORMAT(40X,F5.2,5X,14HEST. OF SILL,5X,F5.2///)
02430C
02440 RETURN
02450 END
02460C
02470C-----
02480 SUBROUTINE DIST(U,V,LL,LH)
02490 DIMENSION U(1000),V(1000),NF(22),NFB(22),IPL0T(50)
02500 DATA IBLNK,IAST/1H,1H*/
02510C
02520 L0CLOW=LL
02530 L0CHIGH=LH
02540 DO 30 I=1,22
02550 NFB(I)=0
02560 30 NF(I)=0

```

```

02570 SUMU=0.0
02580 SUMU2=0.0
02590 D0 20 I=L0CLOW,L0CHIGH
02600 UX=U(I)
02610 SUMU=SUMU+UX
02620 SUMU2=SUMU2+UX*UX
02630 IF(UX.LE.90.0) G0 T0 21
02640 IF(UX.GT.110.0) G0 T0 22
02650 IF=IFIX(UX-90.0+0.99999)+1
02660 NFB(IF)=NFB(IF)+1
02670 G0 T0 20
02680 21 NFB(1)=NFB(1)+1
02690 G0 T0 20
02700 22 NFB(22)=NFB(22)+1
02710 20 CONTINUE
02720 SUMV=0.0
02730 SUMV2=0.0
02740C
02750 D0 10 I=L0CLOW,L0CHIGH
02760 VX=V(I)
02770 SUMV=SUMV+VX
02780 SUMV2=SUMV2+VX*VX
02790 IF(VX.LE.90.0) G0 T0 11
02800 IF(VX.GT.110.0) G0 T0 12
02810 IF=IFIX(VX-90.0+0.99999)+1
02820 NF(IF)=NF(IF)+1
02830 G0 T0 10
02840 11 NF(1)=NF(1)+1
02850 G0 T0 10
02860 12 NF(22)=NF(22)+1
02870 10 CONTINUE
02880C
02890 D0 13 I=1,22
02900 GRAD=89.0+FLOAT(I)
02910 NA=49-NFB(I)/4
02920 D0 15 J=1,NA
02930 15 IPL0T(J)=IBLNK
02940 D0 16 J=NA+1,50
02950 16 IPL0T(J)=IAST
02960 NV=NF(I)/4+1
02970 WRITE(7,2000)(IPL0T(J),J=1,50),NFB(I),GRAD,NF(I),(IAST,J=1,NV)
02980 2000 FORMAT(1X,50A1,I3,1X,F5.1,1X,I3,50A1)
02990 13 CONTINUE
03000 XN=FLOAT(L0CHIGH-L0CLOW+1)
03010 AV=SUMV/XN
03020 VAR=(SUMV2-AV*SUMV)/(XN-1.0)
03030 AVB=SUMU/XN
03040 VARB=(SUMU2-AVB*SUMU)/(XN-1.0)
03050 WRITE(7,2001)AVB,L0CLOW,L0CHIGH,AV,VARB,VAR
03060 2001 FORMAT(/39X,F6.2,5X,*AVERAGE*,I3,*,*,I3,4X,F6.2/
03070+40X,F5.2,5X,14HVARIANCE ' ',5X,F5.2)
03080C
03090 RETURN
03100 END
03110C
03120C-----
03130 SUBROUTINE WEIGHT(R,C,E,NDATA)
03140 COMMON /DICK/L0C(100)
03150 COMMON /BILL/L0CN(6),B(5)
03160 DIMENSION A(5,5),T(7,7),S(7)
03170 LSO=4
03180C
03190 D0 99 KLM=L0C(3)+1,L0C(NDATA-2)-1
03200 IF(KLM.EQ.L0C(LSO)) G0 T0 100

```

```

03210 IF (KLM.GT.(LOC(LS0-1)+1)) GO TO 111
03220C
03230 D0 103 I=1,3
03240 D0 103 J=I+1,4
03250 DIFF=FLOAT(LOC(LS0-3+J)-LOC(LS0-3+I))
03260 A(I,J)=GSPH(DIFF,R,C,E)
03270 103 A(J,I)=A(I,J)
03280 D0 104 I=1,5
03290 A(5,I)=1.0
03300 A(I,5)=1.0
03310 104 A(I,I)=0.0
03320C
03330 111 CONTINUE
03340 D0 112 I=1,5
03350 D0 112 J=1,5
03360 112 T(I,J)=A(I,J)
03370 D0 105 I=1,4
03380 DIFF=ABS(FLOAT(LOC(LS0-3+I)-KLM))
03390 105 B(I)=GSPH(DIFF,R,C,E)
03400 B(5)=1.0
03410 D0 109 I=1,5
03420 109 S(I)=B(I)
03430 CALL KARON(T,S,5,KS)
03440 D0 115 I=1,5
03450 115 B(I)=S(I)
03460 D0 116 I=1,6
03470 116 LOCN(I)=LOC(LS0-4+I)
03480 IF (B(1).GE.-0.003.AND.B(1).LE.0.6.
03490+AND.B(4).GE.-0.003.AND.B(4).LE.0.6) GO TO 101
03500C
03510 CALL ACCUR(R,C,E,KLM)
03520 GO TO 99
03530C
03540 101 WRITE(4,2001) (B(I),I=1,4)
03550 2001 FORMAT(2X,6H      0,4F6.4,6H      0)
03560 GO TO 99
03570C
03580 100 LS0=LS0+1
03590 99 CONTINUE
03600C
03610 REWIND 4
03620 RETURN
03630 END
03640C
03650C-----
03660 FUNCTION GSPH(H,A,C,E)
03670 GSPH=C+E
03680 IF (H.GE.A) RETURN
03690 X=H/A
03700 GSPH=C*(1.5*X-0.5*X*X*X)+E
03710 RETURN
03720 END
03730C
03740C-----
03750 SUBROUTINE ACCUR(R,C,E,KLM)
03760 COMMON /BILL/LOCN(6),B(5)
03770 DIMENSION T(7,7),D(7),F(6)
03780 WGTf=B(1)
03790 WGTl=B(4)
03800 D0 104 J=1,4
03810 104 F(J+1)=B(J)
03820 F(1)=0.0
03830 F(6)=0.0
03840 NR=24

```

```

03850C
03860 D0 102 M=1,2
03870 MNR=NR
03880 IF (WGTF.LT.-0.003) NR=NR+9
03890 IF (WGTL.LT.-0.003) NR=NR-1
03900 IF (WGTF.GT.0.6) NR=NR-9
03910 IF (WGTL.GT.0.6) NR=NR+1
03920 IF (MNR.EQ.NR) GO TO 101
03930 NS=NR-(NR/10)*10
03940 NT=NR/10
03950 IF ((NT+NS).GT.7.0R.(NT+NS).LT.4) GO TO 101
03960C
03970 D0 110 I=1,NS-1
03980 D0 110 J=I+1,NS
03990 I1=NT+I-1
04000 J1=NT+J-1
04010 DIFF=FLOAT (LOCN (J1) -LOCN (I1))
04020 T (I, J) =GSPH (DIFF, R, C, E)
04030 110 T (J, I) =T (I, J)
04040 D0 111 I=1,NS+1
04050 T (I, NS+1) =1.0
04060 T (NS+1, I) =1.0
04070 111 T (I, I) =0.0
04080 D0 112 I=1,NS
04090 DIFF=ABS (FLOAT (LOCN (NT-1+I) -KLM))
04100 112 D (I) =GSPH (DIFF, R, C, E)
04110 D (NS+1) =1.0
04120 D0 113 I=1,6
04130 113 F (I) =D (I)
04140 CALL KARON (T, D, NS+1, KS)
04150 D0 116 I=1,6
04160 116 F (I) =0.0
04170 D0 117 I=1,NS
04180 117 F (NT+I-1) =D (I)
04190 WGTF=D (1)
04200 WGTL=D (NS)
04210 102 CONTINUE
04220C
04230 101 WRITE (4,2001) (F (I), I=1,6)
04240 2001 FORMAT (2X,6F6.4)
04250 RETURN
04260 END
04270C
04280C-----
04290 SUBROUTINE KARON (A,D,NPAR,KS)
04300 DIMENSION A (7,7), D (7)
04310 KS=0
04320 TOL=0.000001
04330 N=NPAR
04340 D0 65 J=1,N
04350 JY=J+1
04360 BIGA=0.
04370 D0 30 I=J,N
04380 IF (ABS (BIGA) -ABS (A (I, J))) 35,30,30
04390 35 BIGA=A (I, J)
04400 IMAX=I
04410 30 CONTINUE
04420 IF (ABS (BIGA) -TOL) 10,10,40
04430 40 D0 50 K=J,N
04440 SAVE=A (IMAX, K) /BIGA
04450 A (IMAX, K) =A (J, K)
04460 A (J, K) =SAVE
04470 50 CONTINUE
04480 SAVE=D (IMAX) /BIGA

```

```
04490 D(IMAX)=D(J)
04500 D(J)=SAVE
04510 IF (J=N) 55,70,55
04520 55 DO 65 IX=JY,N
04530 DO 60 JX=JY,N
04540 60 A(IX,JX)=A(IX,JX)-A(IX,J)*A(J,JX)
04550 65 D(IX)=D(IX)-D(J)*A(IX,J)
04560 70 NY=N-1
04570 DO 80 J=1,NY
04580 IB=N-J
04590 DO 80 K=1,J
04600 IC=N-K+1
04610 80 D(IB)=D(IB)-A(IB,IC)*D(IC)
04620 RETURN
04630 10 KS=1
04640 WRITE(7,1000)NPAR
04650 1000 FORMAT(5X,*THERE IS NO SOLUTION WITH NPAR =*,I3,* SO THERE*)
04660 RETURN
04670 END
04680C
04690C-----
04700 FUNCTION IROUND(X)
04710 Y=ABS(X)
04720 IROUND=IFIX(Y+0.5)
04730 IF (X.LT.0.0) IROUND=-IROUND
04740 RETURN
04750 END
```


Appendix 5 : Determination of the co-ordinates of any point
on ten regular axes

Figure A1 shows two projections of a dodecahedron. The ten pairs of opposite vertices are numbered. The standard orthogonal x, y, z directions are as shown.

Given the co-ordinates on the x, y, z axes (X, Y, Z), the task was to find the co-ordinate on each of the ten axes (C1, C2, C3,....C10). The method adopted was :-

Input of X Y Z

Do for I=1 to 5

Rotate X Y Z around y direction

by angle(A+B) to give Ci

Rotate X Y Z around z direction

by 72° to give new X Y Z

Do for I=6 to 10

Rotate X Y Z around y direction

by angle(A) to give Ci

Rotate X Y Z around z direction

by 72° to give new X Y Z

When rotating the x, y, z directions around one of them (say the y direction as in figure A2) the new co-ordinates are derived from :-

FIGURE A 1 : DODECAHEDRON WITH POSITION OF 10
REGULAR AXES

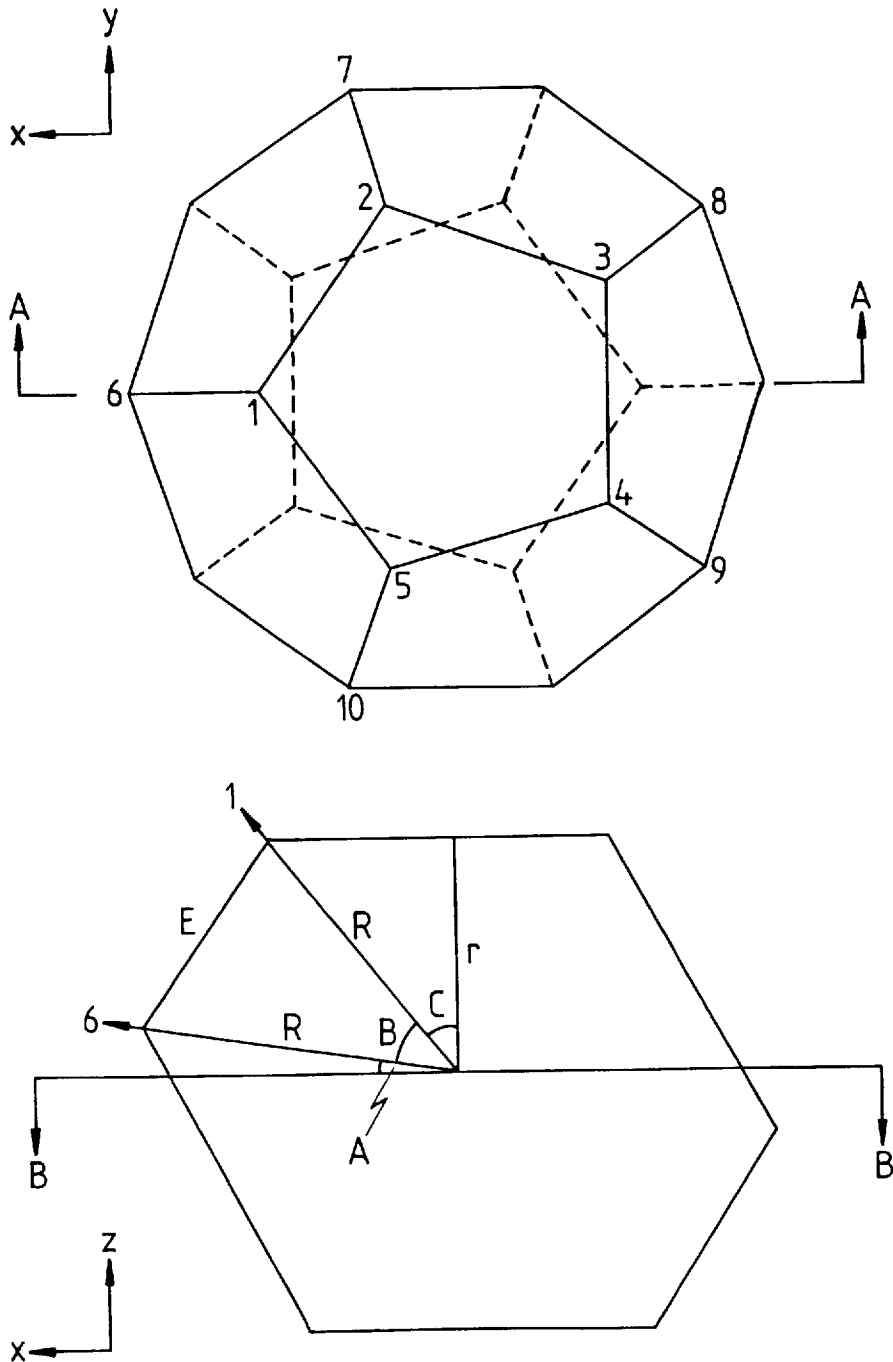
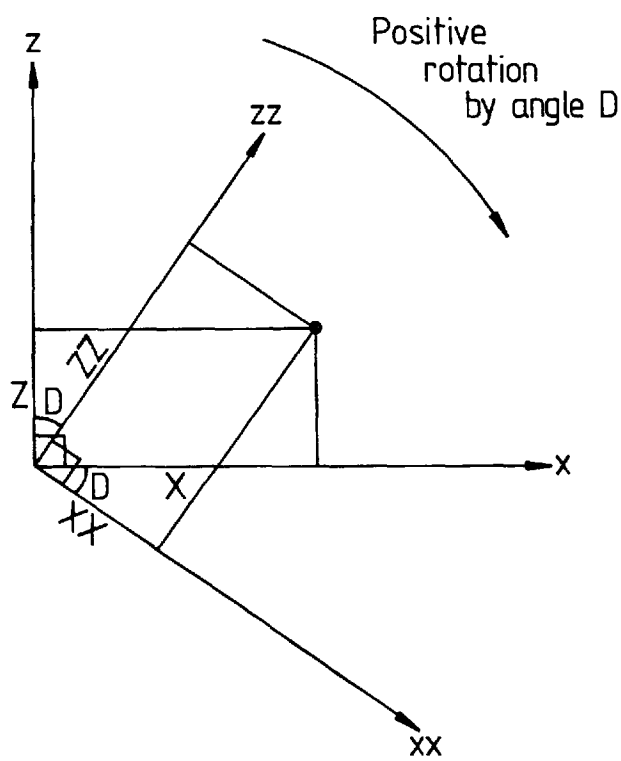


FIGURE A2: TRANSFORMATION OF
CO-ORDINATES



$$XX=X\cos(D)-Z\sin(D)$$

$$YY=Y$$

$$ZZ=Z\sin(D)+Z\cos(D)$$

Now from standard geometry :-

$$\text{Inscribed circle radius} = r = 1.114E$$

$$\text{Circumscribed circle radius} = R = 1.401E$$

where E = length of an edge

$$\text{therefore angle B} = 41.8^\circ$$

$$\text{angle A} = 10.8^\circ$$

So the technique becomes :-

Input of X Y Z

Do for I=1 to 5

XX=X YY=Y

Ci=XXcos(-52.6)-Zsin(-52.6)

X=XXcos(72)-YYsin(72)

Y=XXsin(72)+YYcos(72)

N.B. X and Y having been rotated by 72° 5 times
are equal to the original X and Y.

Do for I=6 to 10

XX=X YY=Y

Ci=XXcos(-10.8)-Zsin(-10.8)

X=XXcos(72)-YYsin(72)

Y=XXsin(72)+YYcos(72)

Output of X Y Z C1 C2 C3.....C10

Using this technique the point where each axis leaves a 10*10*10 cube was found to be as follows :-

Axis number	Co-ordinates where axis leaves cube			Angle to standard directions		
	x	y	z	x	y	z
1	3.8	0.0	5.0	78	90	37
2	1.2	-3.6	5.0	79	55	37
3	-3.1	-2.2	5.0	60	69	37
4	-3.1	2.2	5.0	60	69	37
5	1.2	3.6	5.0	79	55	37
6	5.0	0.0	1.0	11	90	79
7	-1.6	5.0	-1.0	73	21	79
8	5.0	3.6	-1.2	37	55	79
9	5.0	-3.6	-1.2	37	55	79
10	1.6	5.0	1.0	73	21	79

$$\sum \cos = 4.75 \quad 4.86 \quad 4.95$$

From this information a check for the regularity of the spacing of the axes was carried out. The angle between all possible pairs of axes was calculated to be :-

Axis number	1	2	3	4	5	6	7	8	9	10
1	0	41	70	70	41	41	70	70	70	70
2	41	0	41	70	70	70	41	70	70	70
3	70	41	0	41	70	70	70	41	70	70
4	70	70	41	0	41	70	70	70	41	70
5	41	70	70	41	0	70	70	70	70	41
6	41	70	70	70	70	0	70	41	41	70
7	70	41	70	70	70	70	0	70	41	41
8	70	70	41	70	70	41	70	0	70	41
9	70	70	70	41	70	41	41	70	0	70
10	70	70	70	70	41	70	41	41	70	0

The 9 angles between any one axis and its colleagues are the same for all 10 axes. This is evidence that the axes are regularly orientated in space.

Appendix 6 : Subroutine COORDS2

The listing of the subroutine COORDS2 is shown overleaf. The required input to the subroutine is :-

X Y Z - The 3 co-ordinates of the point on the standard x y z axes.

The output of the subroutine is :-

C - An array of 10 stores containing the co-ordinate of the point on each of the 10 regular axes.

```
00100 SUBROUTINE COORDS2(C,X,Y,Z)
00110 DIMENSION C(10),B(2,2)
00120 DATA ((B(I,J),J=1,2),I=1,2),COS72,SIN72/
00130+.6149781,-.7885442,.9822470,-.1875924,.30901699,.95105652/
00140 N=0
00150 DO 11 M=1,2
00160 COSANG=B(M,1)
00170 ZSINANG=Z*B(M,2)
00180 DO 11 I=1,5
00190 N=N+1
00200 C(N)=X*COSANG-ZSINANG
00210 TX=X
00220 X=X*COS72-Y*SIN72
00230 11 Y=TX*SIN72+Y*COS72
00240 RETURN
00250 END
```

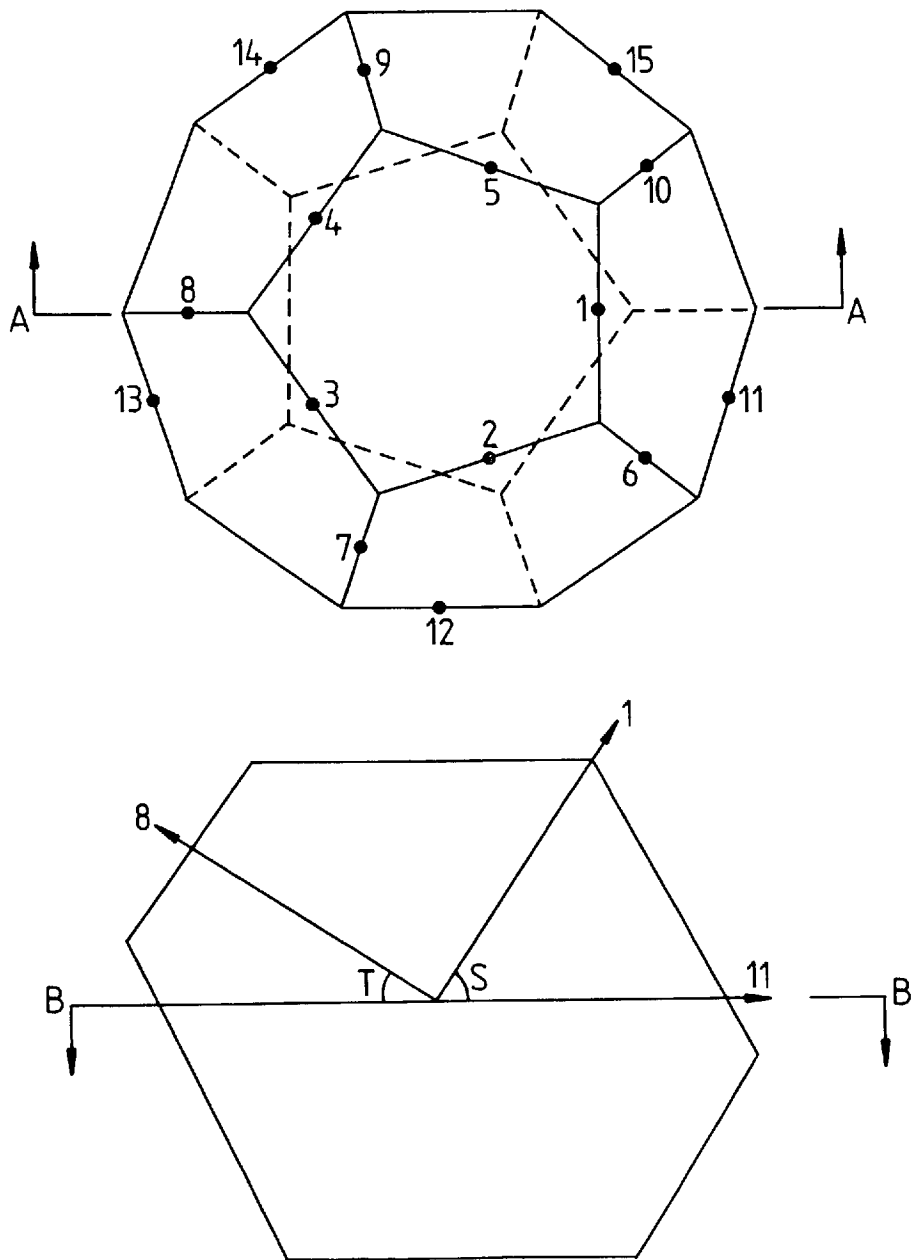

Appendix 7 : Determination of the co-ordinates of any point
on fifteen regular axes

Figure A3 shows two projections of a dodecahedron. The fifteen pairs of opposite edge mid-points are numbered. The standard orthogonal x, y, z directions are as shown.

Given the co-ordinates on the x, y, z axes (X, Y, Z), the task was to find the co-ordinate on each of the fifteen axes (C1, C2, C3,....C15). The method adopted was :-

```
Input of X Y Z
Do for I=1 to 5
Rotate X Y Z around y direction
    by angle(S) to give Ci
Rotate X Y Z around z direction
    by 72° to give new X Y Z
Rotate X Y Z around z direction
    by 36° to give new X Y Z
Do for I=6 to 10
Rotate X Y Z around y direction
    by angle(T) to give Ci
Rotate X Y Z around z direction
    by 72° to give new X Y Z
Rotate X Y Z around z direction
    by -18° to give new X Y Z
Do for I=11 to 15
Rotate X Y Z around z direction
    by 72° to give new X Y Z
```

FIGURE A3. DODECAHEDRON WITH POSITION OF 15
REGULAR AXES



Now using the geometry calculated in Appendix 5 the angles can be found to be :-

$$\text{angle S} = 26.6^\circ$$

$$\text{angle T} = 31.7^\circ$$

So the technique becomes :-

Input of X Y Z

Do for I=1 to 5

$$C_i = X \cos(-58.3) - Z \sin(-58.3)$$

$$XX = X \cos(72) - Y \sin(72)$$

$$YY = X \sin(72) + Y \cos(72)$$

$$X = XX \quad Y = YY$$

$$X = XX \cos(36) - YY \sin(36)$$

$$Y = XX \sin(36) + YY \cos(36)$$

Do for I=6 to 10

$$C_i = X \cos(-31.7) - Z \sin(-31.7)$$

$$XX = X \cos(72) - Y \sin(72)$$

$$YY = X \sin(72) + Y \cos(72)$$

$$X = XX \quad Y = YY$$

$$X = XX \cos(-18) - YY \sin(-18)$$

$$Y = XX \sin(-18) + YY \cos(-18)$$

Do for I=11 to 15

$$C_i = X$$

$$XX = X \cos(72) - Y \sin(72)$$

$$YY = X \sin(72) + Y \cos(72)$$

$$X = XX \quad Y = YY$$

Output of X Y Z C1 C2 C3.....C15

Using this technique the point where each axis leaves a 10*10*10 cube was found to be as follows :-

Axis number	Co-ordinates where axis leaves cube			Angle to standard directions		
	x	y	z	x	y	z
1	3.1	0.0	5.0	58	90	32
2	1.0	-2.9	5.0	80	60	32
3	-2.5	-1.8	5.0	65	72	32
4	-2.5	1.8	5.0	65	72	32
5	1.0	2.9	5.0	80	60	32
6	5.0	-3.6	3.8	46	60	58
7	1.6	5.0	-3.2	36	36	58
8	5.0	0.0	-3.1	65	90	58
9	-1.6	5.0	3.2	36	36	58
10	5.0	3.6	3.8	46	60	58
11	5.0	-1.6	0.0	18	72	90
12	0.0	5.0	0.0	90	0	90
13	5.0	1.6	0.0	18	72	90
14	-3.6	5.0	0.0	36	36	90
15	3.6	5.0	0.0	36	36	90

$$\sum \cos = 8.67 \quad 7.47 \quad 6.89$$

From this information a check for the regularity of the spacing of the axes was carried out. The angle between all possible pairs of axes was calculated to be :-

Axis number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	36	60	60	36	36	72	90	72	36	60	90	60	72	72
2	36	0	36	60	60	36	36	72	90	72	72	60	90	60	72
3	60	36	0	36	60	72	36	36	72	90	72	72	60	90	60
4	60	60	36	0	36	90	72	36	36	72	60	72	72	60	90
5	36	60	60	36	0	72	90	72	36	36	90	60	72	72	60
6	36	36	72	90	72	0	60	72	72	60	36	60	60	36	90
7	72	36	36	72	90	60	0	60	72	72	90	36	60	60	36
8	90	72	36	36	72	72	60	0	60	72	36	90	36	60	60
9	72	90	72	36	36	72	72	60	0	60	60	36	90	36	60
10	36	72	90	72	36	60	72	72	60	0	60	60	36	90	36
11	60	72	72	60	90	36	90	36	60	60	0	72	36	36	72
12	90	60	72	72	60	60	36	90	36	60	72	0	72	36	36
13	60	90	60	72	72	60	60	36	90	36	36	72	0	72	36
14	72	60	90	60	72	36	60	60	36	90	36	36	72	0	72
15	72	72	60	90	60	90	36	60	60	36	72	36	36	72	0

The 14 angles between any one axis and its colleagues are the same for all 15 axes. This is evidence that the axes are regularly orientated in space.

Appendix 8 : Determination of sum cos alpha terms for Journel's
fifteen regular axes

Using subroutine COORDS1 (see Appendix 2) the point where each axis leaves a 10*10*10 cube was found to be as follows :-

Axis number	Co-ordinates where axis leaves cube			Angle to standard directions		
	x	y	z	x	y	z
1	5.0	0.0	0.0	0	90	90
2	0.0	5.0	0.0	90	0	90
3	0.0	0.0	5.0	0	0	90
4	3.1	5.0	1.9	60	36	72
5	-5.0	1.9	3.1	36	72	60
6	1.9	-3.1	5.0	72	60	36
7	-1.9	3.1	5.0	72	60	36
8	-3.1	5.0	1.9	60	36	72
9	5.0	-1.9	3.1	36	72	60
10	-1.9	-3.1	5.0	72	60	36
11	3.1	-5.0	-1.9	60	36	72
12	5.0	1.9	3.1	36	72	60
13	3.1	-5.0	1.9	60	36	72
14	5.0	1.9	-3.1	36	72	60
15	1.9	3.1	5.0	72	60	36

$$\sum \cos = 7.47 \quad 7.47 \quad 7.47$$

Appendix 9 : Subroutine COORDS3

The listing of the subroutine COORDS3 is shown overleaf. The required input to the subroutine is :-

X Y Z - The 3 co-ordinates of the point on the standard x y z axes.

The output of the subroutine is :-

C - An array of 15 stores containing the co-ordinate of the point on each of the 15 regular axes.

```
00100 SUBROUTINE COORDS3(C,X,Y,Z)
00110 DIMENSION C(15),B(3,2)
00120 DATA ((B(I,J),J=1,2),I=1,3),COS36,SIN36,COS72,SIN72/
00130+.52573111,-.85065081,.85065081,-.52573111,1.0,0.0,
00140+.80901699,.58778525,.30901699,.95105652/
00150 N=0
00160 DO 10 M=1,3
00170 COSANG=B(M,1)
00180 ZSINANG=Z*B(M,2)
00190 DO 11 I=1,5
00200 N=N+1
00210 C(N)=X*COSANG-ZSINANG
00220 TX=X
00230 X=X*COS72-Y*SIN72
00240 11 Y=TX*SIN72+Y*COS72
00250 TX=X
00260 IF (M.EQ.1) GO TO 12
00270 X=X*SIN72+Y*COS72
00280 Y=Y*SIN72-TX*COS72
00290 GO TO 10
00300 12 X=X*COS36-Y*SIN36
00310 Y=TX*SIN36+Y*COS36
00320 10 CONTINUE
00330 RETURN
00340 END
```


Appendix 10 : Program RANDO

The listing of the program RANDO is shown overleaf.

The required input to the program is :-

NSEED - A four digit number used to trigger the random number generator.

A - The semi-variogram model range of influence.

C - The semi-variogram model sill.

ENUG - The semi-variogram model nugget effect.

AVER - The distribution model average.

NAXES - The number of randomly orientated axes to be used.

The output of the program is :-

V - An array of 1000 stores containing the unconditional simulation values.

The average and variance of the simulated values.

The semi-variogram of the simulated values.

```

00100 PROGRAM RAND0 (INPUT,OUTPUT,TAPE6=INPUT,TAPE7=OUTPUT,TAPE3,TAPE4)
00110C
00120C THIS PROGRAM SIMULATES A LINE OF 1000 POINTS WHICH FOLLOW
00130C SPHERICAL (A,C,ENUG) AND N(AVER,C).
00140C THE SIMULATION IS PRODUCED USING 'NAXES' RANDOMLY SPACED AXES
00150C
00160C TAPE3=OUTPUT FOR SIMULATED VALUES
00170C TAPE4=OUTPUT FOR SEMI-VARIOGRAM
00180C TAPE6=INPUT FROM TELEX
00190C TAPE7=OUTPUT ON TELEX (SIMPLIFIED VERSION OF TAPE4)
00200C
00210 DIMENSION V(1000),Y(2000)
00220 DATA PI/3.14159265/
00230 READ(6,1000)NSEED
00240 1000 FORMAT(I4)
00250 XSEED=FLOAT(2*NSEED+1)
00260 CALL RANSET(XSEED)
00270 DO 71 I=1,1000
00280 V(I)=0.0
00290 71 S=RANF(0.0)
00300 READ(6,1001)A,C,ENUG,AVER,NAXES
00310 1001 FORMAT(4F6.2,I3)
00320 WRITE(4,3345)NAXES
00330 3345 FORMAT(* NUMBER OF RANDOM AXES IS *,I3)
00340 DO 99 I=1,NAXES
00350 XYANG=2.0*PI*RANF(0.0)
00360 XZANG=2.0*PI*RANF(0.0)
00370 WRITE(4,2000)XYANG,XZANG
00380 2000 FORMAT(2(2X,F6.4))
00390 AY=TAN(XYANG)
00400 BZ=TAN(XZANG)
00410 B=0.01*A
00420 XINT=1.0/SQRT(1.0+AY*AY+BZ*BZ)
00430 XLIM=499.5*XINT
00440 NY=IROUND((2.0*XLIM)/B)+2
00450 NORG=IROUND(-XLIM/B)
00460 CALL SIM(Y,A,NY)
00470 DO 10 J=1,1000
00480 XJ=FLOAT(J-1)-499.5
00490 XC=XJ*XINT
00500 NC=IROUND(XC/B)-NORG+1
00510 10 V(J)=V(J)+Y(NC)
00520 99 CONTINUE
00530 CO=C-ENUG
00540 STD=SQRT(CO/FLOAT(NAXES))
00550 SDENUG=SQRT(ENUG)
00560 SUMV=0.0
00570 SUMV2=0.0
00580 DO 72 I=1,1000
00590 CALL RANDOM(S,SDENUG)
00600 V(I)=V(I)*STD+S+AVER
00610 SUMV=SUMV+V(I)
00620 72 SUMV2=SUMV2+V(I)*V(I)
00630 AV=SUMV/1000.0
00640 VAR=(SUMV2-SUMV*AV)/999.0
00650 WRITE(7,1003)AV,VAR
00660 WRITE(4,1003)AV,VAR
00670 1003 FORMAT(* AV,VAR *,2(2X,F8.4))
00680 WRITE(3,1002)(V(J),J=1,1000)
00690 1002 FORMAT(10(2X,F6.2))
00700 CALL PLOT(V,A)
00710 ENDFILE 3
00720 ENDFILE 4
00730 STOP

```

```

00740 END
00750C
00760 FUNCTION IRROUND(X)
00770 Y=ABS(X)
00780 IRROUND=IFIX(Y+0.5)
00790 IF(X.LT.0.0) IRROUND=-IRROUND
00800 RETURN
00810 END
00820C
00830 SUBROUTINE SIM(Y,A,NS)
00840 DIMENSION Y(2000),T(100)
00850 B=0.01*A
00860 WRITE(7,2012)NS
00870 2012 FORMAT(* NY=*,I5)
00880 WF=SQRT(12.0*B/A/(A*A+11.0*B*B))
00890 DO 50 M=1,100
00900 50 CALL RANDOM(T(M),1.0)
00910 DO 53 I=1,NS
00920 Y(I)=0.0
00930 DIS=0.0-(A+B)/2.0
00940 DO 52 K=1,50
00950 DIS=DIS+B
00960 52 Y(I)=Y(I)+DIS*(T(K)-T(101-K))
00970 Y(I)=Y(I)*WF
00980 DO 54 M=1,99
00990 54 T(M)=T(M+1)
01000 CALL RANDOM(T(100),1.0)
01010 53 CONTINUE
01020 RETURN
01030 END
01040C
01050 SUBROUTINE RANDOM(S,SD)
01060 S=0.0
01070 DO 10 M=1,12
01080 10 S=S+RANF(0.0)
01090 S=(S-6.0)*SD
01100 RETURN
01110 END
01120C
01130 SUBROUTINE PLOT(V,A)
01140 DIMENSION V(1000),IGRA(100)
01150 DATA IAST,IPLUS,IBLNK/1H*,1H+,1H /
01160 RMS=0.0
01170 DO 20 LAG=1,100
01180 LIM=1000-LAG
01190 GAM=0.0
01200 DO 21 I=1,LIM
01210 DIFF=V(I)-V(I+LAG)
01220 21 GAM=GAM+DIFF*DIFF
01230 SEMI=0.5*GAM/FLOAT(LIM)
01240 DO 22 J=1,100
01250 22 IGRA(J)=IBLNK
01260 HOVERA=FLOAT(LAG)/A
01270 XMODEL=20.0*(1.5*HOVERA-0.5*HOVERA*HOVERA*HOVERA)
01280 IF(HOVERA.GT.1.0) XMODEL=20.0
01290 RMS=RMS+((SEMI-XMODEL)/XMODEL)**2
01300 IS=IFIX(2.0*SEMI)+2
01310 IM=IFIX(2.0*XMODEL)+2
01320 IGRA(1)=IPLUS
01330 IGRA(IM)=IPLUS
01340 IGRA(IS)=IAST
01350 WRITE(4,2111)LAG,(IGRA(K),K=1,100)
01360 2111 FORMAT(2X,I3,100A1)
01370 20 CONTINUE

```

```
01380 RMS=100.0*SQRT(RMS/100.0)
01390 WRITE(4,2004)RMS
01400 WRITE(7,2004)RMS
01410 2004 FORMAT(* RMS = *,F7.2,* PERCENT*)
01420 RETURN
01430 END
```

Appendix 11 : Program CONSIM1

The listing of the program CONSIM1 is shown overleaf.

The required input to the program is :-

TITLE - A title phrase of up to 80 characters.

F - An array of 8 stores. The first 4 stores contain the range, sill, nugget effect, and average of the models of the first system (lode widths). The second set of 4 stores hold the same information for the second system (lode assays).

NLN - The number of levels from which conditioning data is to be inputted.

T B P S - The top, bottom, left, and right limits of the simulated area.

XT XB XP XS - The top, bottom, left, and right limits of the rectangle which contains all the conditioning data.

NLEVEL - The number of conditioning data points on this level drive.

XLEVEL - The height above datum of this level drive.

EAST - The Easting of the most Westerly sample on this level drive.

EAST2 - The Easting of the most Easterly sample on this level drive.

IFN - A number which defines whether a sample's measurements are in imperial (IFN=1) or metric (IFN=2) units.

WL - The sample lode width measurement.

VL - The sample lode assay measurement.

FILE - The file name for storage of the sample data, the lode width kriging weights, and the lode assay kriging weights.

The output of the program includes :-

The minimum, average, and maximum number of data points used in the conditioning kriging systems.

The minimum, average, and maximum kriging variance achieved by the kriging systems during conditioning.

```

00100 PROGRAM CONSIM1 (INPUT=131B,OUTPUT=131B,TAPE6=INPUT,TAPE7=OUTPUT,
00110+TAPE4)
00120C
00130C TAPE4=OUTPUT FOR GENERAL DATA AND KRIGING WEIGHTS ONTO FILES
00140C DLSTOPE 'VLSTOPE'VLSTOPE FOR USE IN 'CONSIM2'
00150C TAPE6=INPUT SAMPLES DATA AND BLOCK SIZE ETC.
00160C
00170C

```

```

00180C          DEV. DATA
00190C  -----XT
00200C  |
00210C  |          ++++++++T
00220C  |          +         +
00230C  |          +         +
00240C  |          ++++++++B
00250C  |          P         S
00260C  -----XB
00270C  XP          DEV. DATA          XS
00280C

```

```

00290C GIVEN SUCH A BLOCK THIS PROGRAM PREPARES FOR SIMULATION OF IT
00300C BY GIVING THE KRIGING WEIGHTS AND SAMPLE NUMBERS TO WHICH THEY
00310C APPLY, FOR 9*9 POINTS IN THE BLOCK.
00320C TO BE RUN ON INSTANT TURNAROUND.
00330C
00340C

```

```

00350 COMMON /FRED/X(110),Y(110),IFUSED(110)
00360 DIMENSION DATA(2,110),TITLE(8)
00370 DIMENSION F(2,4)
00380C
00390C          X=UPWARDS(IN)   Y=ALONG(IE)
00400C          NEXT(1)        NEXT(2)
00410C

```

```

00420 READ(6,1000)(TITLE(K),K=1,8)
00430 WRITE(7,1000)(TITLE(K),K=1,8)
00440 1000 FORMAT(1X,8A10)
00450C
00460C THE SEMIVARIOGRAM AND DISTRIBUTION DATA FOR ALL THE SYSTEMS
00470C IS READ IN .

```

```

00480 READ(6,1006)((F(NV,K),K=1,4),NV=1,2)
00490 WRITE(7,1006)((F(NV,K),K=1,4),NV=1,2)
00500 1006 FORMAT(1X,4F6.3)
00510 N=0

```

```

00520 READ(6,1005)NLN,T,B,P,S,XT,XB,XP,XS
00530 1005 FORMAT(1X,I1,8F6.0)
00540C NLN=NUMBER OF LEVELS
00550C T,B,P,S=LIMITS OF BLOCK TO BE SIMULATED
00560C XT,XB,XP,XS=LIMITS OF BLOCK WHICH INCLUDES ALL DATA
00570C

```

```

00580 DO 11 I=1,NLN
00590 READ(6,1007)NLEVEL,XLEVEL,EAST,EAST2
00600 1007 FORMAT(1X,I3,3F6.0)
00610C NLEVEL=NUMBER OF SAMPLES ON THIS LEVEL TO BE READ IN
00620C XLEVEL=LEVEL (METRES) OF THE SAMPLES ON THIS LEVEL
00630C EAST=EASTING (METRES) OF MOST WESTERLY SAMPLE ON THIS LEVEL
00640C EAST2=EASTING (METRES) OF MOST EASTERLY SAMPLE ON THIS LEVEL
00650C          SAMPLES WILL BE GIVEN EASTINGS INCREASING BY
00660C          (EAST2-EAST)/(NLEVEL-1)
00670C

```

```

00680 DIFF=(EAST2-EAST)/FLOAT(NLEVEL-1)
00690C
00700 DO 11 J=1,NLEVEL
00710C
00720 READ(6,1001)IFN,WL,VL,VWL
00730 1001 FORMAT(9X,I1,F5.2,F7.2,F7.2)

```

```

00740 IF (VL.EQ.0.0) GO TO 11
00750 N=N+1
00760 X(N)=XLEVEL
00770 Y(N)=EAST+FLOAT(J-1)*DIFF
00780 IF (IFN.EQ.2) GO TO 21
00790 WL=0.3048*(FLOAT(IFIX(2.0*WL+0.5))/2.0)
00800 IF (VL.NE.99.11) VL=0.043*(FLOAT(IFIX((VL+2.0)/2.0))*2.0-1.0)
00810 IF (VL.EQ.99.11) VL=0.043*0.5
00820 IF (VWL.EQ.99.11) VWL=0.5
00830 VWL=0.043*0.3048*VWL
00840 GO TO 27
00850 21 WL=FLOAT(IFIX(5.0*WL+0.5))/5.0
00860 IF (VL.NE.99.11) VL=FLOAT(IFIX(20.0*VL+0.5))/20.0
00870 IF (VL.EQ.99.11) VL=0.025
00880 IF (VWL.EQ.99.11) VWL=0.005
00890 27 DATA(1,N)=WL
00900 DATA(2,N)=VL
00910 WRITE(7,2014)N,Y(N),X(N),IFN,WL,VL,VWL
00920 2014 FORMAT(1X,I3,2F6.1,I1,3F7.2)
00930 11 CONTINUE
00940C
00950 WRITE(7,1000)(TITLE(K),K=1,8)
00960C
00970 WRITE(4,1000)(TITLE(K),K=1,8)
00980 WRITE(4,1006)((F(NV,K),K=1,4),NV=1,2)
00990 SINTX=(T-B)/8.0
01000 SINTY=(S-P)/8.0
01010 WRITE(4,2009)T,B,P,S,XT,XB,XP,XS
01020 2009 FORMAT(1X,8F6.1)
01030 WRITE(4,2003)N
01040 2003 FORMAT(1X,I3)
01050 WRITE(4,2000)((X(J),Y(J),DATA(1,J),DATA(2,J)),J=1,N)
01060 2000 FORMAT(1X,2F6.1,2F6.2)
01070 READ(6,1002)FILE
01080 ENDFILE 4
01090 CALL REPLACE(5HTAPE4,FILE)
01100C
01110 D0 26 NV=1,2
01120 RANGE=F(NV,1)
01130 C=F(NV,2)
01140 ENUG=F(NV,3)
01150C
01160 XX=SIGKMAX=SUMSIGK=0.0
01170 SIGKMIN=100000.0
01180 NSMAX=NSSUM=0
01190 NSMIN=100000
01200 D0 18 M=1,N
01210 18 IFUSED(M)=0
01220 D0 99 I=1,9
01230 XI=B+FLOAT(I-1)*SINTX
01240 D0 99 J=1,9
01250 YJ=P+FLOAT(J-1)*SINTY
01260 CALL SEARCH(RANGE,C,ENUG,XI,YJ,N,NS,SIGK)
01270C
01280 SUMSIGK=SUMSIGK+SIGK
01290 IF (SIGK.GT.SIGKMAX) SIGKMAX=SIGK
01300 IF (SIGK.LT.SIGKMIN) SIGKMIN=SIGK
01310 XX=XX+1.0
01320 NSSUM=NSSUM+NS
01330 IF (NS.GT.NSMAX) NSMAX=NS
01340 IF (NS.LT.NSMIN) NSMIN=NS
01350 99 CONTINUE
01360C
01370 SUMSIGK=SUMSIGK/XX

```



```

01380 AVNS=FLOAT(NSSLIM)/XX
01390 WRITE(7,2005)SIGKMIN,SUMSIGK,SIGKMAX,NSMIN,AVNS,NSMAX
01400 2005 FORMAT(//10X,*KRIGING WITH NEAREST 3 IN 60 DEGREE BANDS*/
01410+10X,*NS.GE.4 NS.LE.18*//10X,*MINIMUM AVERAGE MAXIMUM VALUES OF *,
01420+/5X,*SIGK2*,3F7.4/7X,*NS *,I7,F7.4,I7//)
01430 WRITE(7,2006)(IFUSED(M),M=1,N)
01440 2006 FORMAT(5X,*NUMBER OF TIMES DATA POINTS USED FOR KRIGING*/
01450+10(5X,20I4//))
01460 READ(6,1002)FILE
01470 1002 FORMAT(A7)
01480 ENDFILE 4
01490 CALL REPLACE(5HTAPE4,FILE)
01500C
01510 26 CONTINUE
01520 STOP
01530 END
01540C-----+-----
01550 SUBROUTINE SEARCH(RANG,C,ENUG,XI,YJ,N,NSS,SIGK)
01560 COMMON /FRED/X(110),Y(110),IFUSED(110)
01570 COMMON /TOM/A(19,19),D(19)
01580 DIMENSION LB(110),NEAR(18),DISTM(18),DD(18)
01590 THIRTY=ASIN(0.5)
01600 HALF=THIRTY/60.0
01610 SIXTY=THIRTY*2.0
01620 NINETY=THIRTY*3.0
01630 ONE80=THIRTY*6.0
01640 THREE60=THIRTY*12.0
01650C
01660 RANGE=RANG
01670 RADIUS=RANGE
01680 40 NS=0
01690 DO 12 M=1,N
01700 MM=M
01710 XDIFF=ABS(X(M)-XI)
01720 YDIFF=ABS(Y(M)-YJ)
01730 IF(XDIFF.LT.0.05.AND.YDIFF.LT.0.05) GO TO 50
01740 IF(YDIFF.GT.RADIUS) GO TO 12
01750 IF(XDIFF.GT.RADIUS) GO TO 12
01760 DIST=SQRT(XDIFF*XDIFF+YDIFF*YDIFF)
01770 IF(DIST.GT.RADIUS) GO TO 12
01780 NS=NS+1
01790 LB(NS)=M
01800 12 CONTINUE
01810 IF(NS.GT.4) GO TO 41
01820C IF THERE ARE 4 OR LESS POINTS WITHIN THE SEARCH CIRCLE ,
01830C THE RADIUS OF THE SEARCH CIRCLE IS INCREASED BY HALF THE RANGE
01840 42 RADIUS=RADIUS+0.5*RANGE
01850 GO TO 40
01860C
01870 41 DO 20 M=1,18
01880 NEAR(M)=-99
01890 20 DISTM(M)=100000.0
01900C THE DATA POINTS WITHIN THE SEARCH CIRCLE ARE SORTED. THE NEAREST
01910C 3 POINTS IN EACH SWEEP OF 60 DEGREES CENTRED ON BEARINGS 0,300,60
01920C ARE TAKEN. THEREFORE THERE CAN BE NO MORE THAN 18 POINTS IN
01930C THE KRIGING SYSTEM.
01940 DO 24 K=1,NS
01950 XDIFF=(X(LB(K))-XI)
01960 YDIFF=(Y(LB(K))-YJ)
01970 ANGLE=ATAN(YDIFF/XDIFF)
01980 IF(XDIFF.EQ.0.0) ANGLE=ONE80-NINETY*YDIFF/ABS(YDIFF)
01990 IF(XDIFF.LT.0.0) ANGLE=ANGLE+ONE80
02000 IF(ANGLE.LT.0.0) ANGLE=ANGLE+THREE60
02010 ANGLE=ANGLE+THIRTY

```

```

02020 IF (ANGLE.GT.THREE60) ANGLE=ANGLE-THREE60
02030 D0 24 M=1,6
02040 IF (ANGLE.LT.(SIXTY*FLOAT(M-1)-HALF)) GO TO 24
02050 IF (ANGLE.GT.(SIXTY*FLOAT(M)+HALF)) GO TO 24
02060 DIST=SQRT(XDIFF*XDIFF+YDIFF*YDIFF)
02070 IF (DIST.GE.DISTMN(3*M)) GO TO 24
02080 IF (DIST.GE.DISTMN(3*M-1)) GO TO 23
02090 IF (DIST.GE.DISTMN(3*M-2)) GO TO 22
02100 DISTMN(3*M)=DISTMN(3*M-1)
02110 NEAR(3*M)=NEAR(3*M-1)
02120 DISTMN(3*M-1)=DISTMN(3*M-2)
02130 NEAR(3*M-1)=NEAR(3*M-2)
02140 DISTMN(3*M-2)=DIST
02150 NEAR(3*M-2)=LB(K)
02160 GO TO 24
02170 22 DISTMN(3*M)=DISTMN(3*M-1)
02180 NEAR(3*M)=NEAR(3*M-1)
02190 DISTMN(3*M-1)=DIST
02200 NEAR(3*M-1)=LB(K)
02210 GO TO 24
02220 23 DISTMN(3*M)=DIST
02230 NEAR(3*M)=LB(K)
02240 24 CONTINUE
02250C
02260 NS=1
02270 D0 29 M=1,18
02280 D0 25 K=1,18
02290C IF A POINT IS INCLUDED TWICE (I.E. IT IS IN THE 1 DEGREE
02300C OVERLAP BETWEEN 2 SWEEPS) THEN ONE COPY IS REMOVED
02310 25 IF (NEAR(K).EQ.NEAR(M).AND.K.NE.M) NEAR(M)=-99
02320 IF (NEAR(M).EQ.-99) GO TO 29
02330 NEAR(NS)=NEAR(M)
02340 NS=NS+1
02350 29 CONTINUE
02360C
02370C IF THERE ARE LESS THAN 4 POINTS(=5 EQNS.) IN THE KRIGING SYSTEM
02380C THE SEARCH CIRCLE RADIUS IS INCREASED AND THE SYSTEM RE-CALCULATED
02390 IF (NS.LT.5) GO TO 42
02400C
02410 D0 13 K=1,NS-2
02420 D0 13 L=K+1,NS-1
02430 XDIFF=(X(NEAR(K))-X(NEAR(L)))
02440 YDIFF=(Y(NEAR(K))-Y(NEAR(L)))
02450 DIST=SQRT(XDIFF*XDIFF+YDIFF*YDIFF)
02460 13 A(K,L)=GSPH(DIST,RANGE,C,ENUG)
02470 D0 17 K=2,NS-1
02480 D0 17 L=1,K-1
02490 17 A(K,L)=A(L,K)
02500 D0 14 K=1,NS
02510 A(K,NS)=A(NS,K)=1.0
02520 14 A(K,K)=0.0
02530 D0 15 K=1,NS-1
02540 XDIFF=(X(NEAR(K))-XI)
02550 YDIFF=(Y(NEAR(K))-YJ)
02560 DIST=SQRT(XDIFF*XDIFF+YDIFF*YDIFF)
02570 D(K)=GSPH(DIST,RANGE,C,ENUG)
02580 15 DD(K)=D(K)
02590 D(NS)=1.0
02600 CALL KARON(NS,KS)
02610 NSS=NS-1
02620 SIGK=D(NS)
02630 D0 32 K=1,NSS
02640 32 SIGK=SIGK+D(K)*DD(K)
02650 D0 19 M=1,NSS

```

```

02660 19 IFUSED (NEAR (M)) = IFUSED (NEAR (M)) + 1
02670 WRITE (4, 4000) NSS, ((NEAR (M), D (M)), M=1, NSS)
02680 4000 FORMAT (9X, I3, 2 (/1X, 9 (I3, F6.4)))
02690 RETURN
02700C
02710C THE POINT BEING CONSIDERED IS A DATA POINT
02720 50 IFUSED (MM) = IFUSED (MM) + 1
02730 WRITE (4, 2001) MM
02740 2001 FORMAT (9X, 3H 1/1X, I3, 6H1.0000)
02750 NSS=1
02760C
02770 SIGK=0.0
02780 RETURN
02790 END
02800C-----
02810C
02820 FUNCTION IRROUND (X)
02830 IRROUND = IFIX (ABS (X) + 0.49999)
02840 IF (X.LT.0.0) IRROUND = -IRROUND
02850 RETURN
02860 END
02870C-----
02880C
02890C
02900 FUNCTION GSPH (DIST, A, C, ENUG)
02910 IF (DIST.GE.A) GO TO 10
02920 X = DIST/A
02930 GSPH = (C - ENUG) * (1.5 * X - 0.5 * X * X * X) + ENUG
02940 RETURN
02950 10 GSPH = C
02960 RETURN
02970 END
02980C-----
02990C
03000C
03010 SUBROUTINE KARON (NPAR, KS)
03020 COMMON /TOM/ A (19, 19), D (19)
03030 KS = 0
03040 TOL = 0.000001
03050 N = NPAR
03060 DO 65 J = 1, N
03070 JY = J + 1
03080 BIGA = 0.
03090 DO 30 I = J, N
03100 IF (ABS (BIGA) - ABS (A (I, J))) > 35, 30, 30
03110 35 BIGA = A (I, J)
03120 IMAX = I
03130 30 CONTINUE
03140 IF (ABS (BIGA) - TOL) 10, 10, 40
03150 40 DO 50 K = J, N
03160 SAVE = A (IMAX, K) / BIGA
03170 A (IMAX, K) = A (J, K)
03180 A (J, K) = SAVE
03190 50 CONTINUE
03200 SAVE = D (IMAX) / BIGA
03210 D (IMAX) = D (J)
03220 D (J) = SAVE
03230 IF (J - N) 55, 70, 55
03240 55 DO 65 IX = JY, N
03250 DO 60 JX = JY, N
03260 60 A (IX, JX) = A (IX, J) * A (J, JX)
03270 65 D (IX) = D (IX) - D (J) * A (IX, J)
03280 70 NY = N - 1
03290 DO 80 J = 1, NY

```

```
03300 IB=N-J
03310 D0 80 K=1,J
03320 IC=N-K+1
03330 80 D<IB>=D<IB>-A<IB,IC>*D<IC>
03340 RETURN
03350 10 KS=1
03360 WRITE<7,1000>NPAR
03370 1000 FORMAT<5X,*THERE IS NO SOLUTION WITH NPAR =*,I3,* SO THERE*>
03380 RETURN
03390 END
```

Appendix 12 : Program CONSIM2

The listing of the program CONSIM2 is shown overleaf.

The required input to the program is :-

NSEED.- A four digit number used to trigger the
random number generator.

An example of the output of the program is shown overleaf.

```

00100 PROGRAM CONSIM2 (INPUT=131B,OUTPUT=131B,TAPE6=INPUT,TAPE7=OUTPUT,
00110+TAPE5,TAPE2,TAPE4)
00120C
00130C USED AFTER 'CONSIM1' THIS PROGRAM PRODUCES SIMULATIONS OF THE
00140C BLOCK WITH 9*9 POINTS.
00150C TO BE RUN ON INSTANT TURNAROUND.
00160C
00170C TAPE7=OUTPUT FOR RESULTS (HISTOGRAMS VARIOGRAMS)
00180C TAPE2=INPUT FOR GENERAL DATA AND KRIGING WEIGHTS FROM FILES
00190C DLSTOPE 'WLSTOPE 'VLSTOPE PRODUCED BY PROGRAM 'CONSIM1'
00200C TAPE5=TEMPORARY STORE FOR 1-D AXIS SIMULATIONS FORMED IN
00210C 'SIM2D15' AND USED IN 'CONSIM2' TO PRODUCE SDATA (NV,N)
00220C
00230 DIMENSION V(9,9),NEXT(2)
00240 DIMENSION FILE(2),W(9,9),F(2,4),AXIS(1000)
00250 DIMENSION DATA(2,110),X(110),Y(110),SDATA(110),NEAR(18),D(18),TITLE(8)
00260 DATA FILE(1),FILE(2)/7HWLSTOPE,7HVLSTOPE/
00270C
00280C X=UPWARDS(IN) Y=ALONG(IE)
00290C NEXX NEXY
00300C
00310 CALL GETPF(5HTAPE2,7HDLSTOPE,7HUMCKA07)
00320 READ(2,1000) (TITLE(K),K=1,8)
00330 1000 FORMAT(1X,8A10)
00340C THE SEMIVARIOGRAM AND DISTRIBUTION DATA FOR ALL THE SYSTEMS
00350C IS READ IN .
00360 READ(2,1006) ((F(NV,K),K=1,4),NV=1,2)
00370 WRITE(7,1006) ((F(NV,K),K=1,4),NV=1,2)
00380 1006 FORMAT(1X,4F6.3)
00390 READ(2,1009) T1,B1,P1,S1,T2,B2,P2,S2
00400 1009 FORMAT(1X,8F6.1)
00410 SINTX=(T1-B1)/8.0
00420 SINTY=(S1-P1)/8.0
00430 NEXT(1)=9
00440 NEXT(2)=9
00450 SMAXX=(T2-B2)/2.0
00460 SMAXY=(S2-P2)/2.0
00470 WRITE(7,2006) (TITLE(K),K=1,8)
00480 2006 FORMAT(10(/),5X,8A10)
00490 READ(2,1007) N
00500 1007 FORMAT(1X,I3)
00510 TSS=RSS=SS=0.0
00520 DO 68 I=1,N
00530 READ(2,1008) X(I),Y(I),DATA(1,I),DATA(2,I)
00540 1008 FORMAT(1X,2F6.1,2F6.2)
00550 TSS=TSS+ALOG(DATA(1,I))**2
00560 SS=SS+ALOG(DATA(1,I))
00570 DATA(1,I)=ALOG(DATA(1,I))
00580 RSS=RSS+DATA(1,I)**2
00590 DO 68 DATA(2,I)=ALOG(DATA(2,I))
00600 SS=100.0*(1.0-RSS/(TSS-SS*SS/FLOAT(N)))
00610 WRITE(7,4321) SS
00620 4321 FORMAT(5X,*PERCENT TSS FOR DATA *,F8.3)
00630C
00640 READ(6,1010) NSEED
00650 1010 FORMAT(I4)
00660 DO 101 NSIM=1,5
00670C
00680 TSS=RSS=SS=0.0
00690 DO 60 NV=1,2
00700 RANGE=F(NV,1)
00710 C=F(NV,2)
00720 ENUG=F(NV,3)
00730 AVER=F(NV,4)

```

```

00740 NSEED=NSEED+1
00750C
00760 B8=B1-B2
00770 PP=P1-P2
00780 CALL SIMQD15(V,NEXT,SINTX,SINTY,BB,PP,SMAXX,SMAXY,
00790+RANGE,C,ENUG,AVER,NSEED)
00800C
00810 WRITE(7,2003) (TITLE(K),K=1,8),NSIM,NV,NSEED
00820 2003 FORMAT(1H1/10X,8A10//30X,*SIMULATION NO. *,I2,* SYSTEM NO. *,I1,
00830+* NSEED = *,I4/30X,44(1H-)//)
00840 WRITE(7,2007)
00850 2007 FORMAT(30X,*BEFORE CONDITIONING (I.E. SHOULD FIT MODELS)* )
00860 CALL VDIST(V,NEXT,-2.1,1.9,20)
00870 CALL GAM(V,NEXT,SINTX,SINTY,RANGE,C,ENUG)
00880C
00890 B=RANGE/100.0
00900 STD=SQRT((C-ENUG)/15.0)
00910 SDENUG=SQRT(ENUG)
00920 D0 51 M=1,N
00930 REWIND 5
00940 XI=X(M)-B2-SMAXX
00950 XJ=Y(M)-P2-SMAXY
00960 G=0.0
00970 D0 69 I=1,15
00980 READ(5,1002) NORG,XINTX,XINTY,NY
00990 1002 FORMAT(1X,I5,2E12.6,I5)
01000 READ(5,1001) (AXIS(J),J=1,NY)
01010 1001 FORMAT(1X,10E12.6)
01020 XC=XI*XINTX+XJ*XINTY
01030 NC=IROUND(XC/B)-NORG+1
01040 69 G=G+AXIS(NC)
01050 CALL RANDOM(R,SDENUG)
01060 51 SDATA(M)=G*STD+R+AVER
01070 CALL GETPF(5HTAPE2,FILE(NV),7HUMCKAD7)
01080 D0 61 I=1,9
01090 D0 61 J=1,9
01100 READ(2,1003) NS
01110 1003 FORMAT(9X,I3)
01120 READ(2,1004) ((NEAR(K),D(K)),K=1,NS)
01130 1004 FORMAT(1X,9(I3,F6.4)/1X,9(I3,F6.4))
01140 AK=SK=0.0
01150 D0 52 M=1,NS
01160 AK=AK+DATA(NV,NEAR(M))*D(M)
01170 52 SK=SK+SDATA(NEAR(M))*D(M)
01180 V(I,J)=V(I,J)+AK-SK
01190 61 CONTINUE
01200C
01210 WRITE(7,2008)
01220 2008 FORMAT(1H1,7(/,30X,*AFTER CONDITIONING BEFORE TAKING EXP*)
01230 CALL VDIST(V,NEXT,-2.1,1.9,20)
01240 CALL GAM(V,NEXT,SINTX,SINTY,RANGE,C,ENUG)
01250C
01260 IF(NV.EQ.2) GO TO 65
01270 D0 67 I=1,9
01280 XI=B1+FLOAT(I-1)*SINTX
01290 D0 67 J=1,9
01300 YJ=P1+FLOAT(J-1)*SINTY
01310 RSS=RSS+V(I,J)**2
01320 WLL=V(I,J)+TREND(YJ,XI)
01330 TSS=TSS+WLL**2
01340 SS=SS+WLL
01350 WLL=EXP(V(I,J))
01360 W(I,J)=WLL
01370C W(I,J)=.13100068E+05-.15468835E+01*YJ-.25478477E+02*XI+

```

```

01380C+.35826833E-01*WLL+.97820590E-04*YJ*YJ+.16512423E-01*XI*XI+
01390C+.16650438E+00*WLL*WLL+.20147847E-02*YJ*XI+
01400C+.15368569E-07*YJ*YJ*YJ-.35656412E-05*XI*XI*XI-
01410C+.67110721E-07*YJ*YJ*XI-.65576605E-06*YJ*XI*XI
01420C IF (W(I,J) .LT. WLL) W(I,J) =WLL
01430 67 CONTINUE
01440 GO TO 60
01450 65 DO 66 I=1,9
01460 66 DO 66 J=1,9
01470 VLL=EXP(V(I,J))
01480 V(I,J)=VLL
01490C V(I,J)=.65726173E-01+.61670581*WLL-.86895918E-02*WLL*WLL
01500C IF (V(I,J) .LT. (WLL*VLL/W(I,J))) V(I,J)=WLL*VLL/W(I,J)
01510 66 IF (V(I,J) .GT. 100.0) V(I,J)=100.0
01520C
01530 60 CONTINUE
01540C
01550 WRITE (7,2009)
01560 2009 FORMAT(1H1,7(/),30X,*AFTER TAKING EXPONENTIALS*)
01570 CALL VDIST(W,NEXT,0.0,4.0,20)
01580 CALL GAM(W,NEXT,SINTX,SINTY,F(1,1),F(1,2),F(1,3))
01590 CALL VDIST(V,NEXT,0.0,10.0,20)
01600 CALL GAM(V,NEXT,SINTX,SINTY,F(2,1),F(2,2),F(2,3))
01610C
01620 WRITE (4,3000) ((W(I,J),J=1,9),I=1,9),((V(K,L),L=1,9),K=1,9)
01630 3000 FORMAT(1X,9F7.2)
01640 SUMW=SUMWL=SUMVL=0.0
01650 DO 70 I=1,9
01660 DO 70 J=1,9
01670 SUMWL=SUMWL+W(I,J)
01680 SUMVL=SUMVL+V(I,J)
01690 70 SUMN=SUMN+1.0
01700 SUMWL=SUMWL/SUMN
01710 SUMVL=SUMVL/SUMN
01720 WRITE (7,2004) SUMN,NSIM,SUMWL,SUMVL
01730 2004 FORMAT(/,10X,*FOR ALL*,F6.0,* SIMULATED SAMPLES*,
01740+* OF SIMULATION NO. *,I2/12X,2HLW,4X,2HLA/10X,2F6.2)
01750 SS=100.0*(1.0-RSS/(TSS-SS*SS/SUMN))
01760 WRITE (7,2000) SS
01770 2000 FORMAT(10(/),10X,*LODE WIDTHS PERCENT TOTAL SUM OF SQUARES*,F7.2)
01780C
01790 101 CONTINUE
01800C
01810 ENDFILE 4
01820 CALL REPLACE(5HTAPE4,4HSIMS)
01830 WRITE (7,1234)
01840 1234 FORMAT(1H1)
01850 STOP
01860 END
01870C ++++++
01880C
01890C
01900 FUNCTION IROUND(X)
01910 IROUND=IF IX (ABS(X)+0.49999)
01920 IF (X.LT.0.0) IROUND=-IROUND
01930 RETURN
01940 END
01950C-----
01960 SUBROUTINE VDIST(V,NEXT,VMIN,VMAX,NG)
01970 DIMENSION V(9,9),NEXT(2)
01980 DIMENSION IHIST(50)
01990 DATA IAST/1H*/
02000 NGRP=NG
02010 DO 20 I=1,50

```



```

02020 20 IHIST(I)=0
02030 AV=0.0
02040 VINT=(VMAX-VMIN)/FLOAT(NGRP)
02050 D0 10 J=1,NEXT(1)
02060 D0 10 K=1,NEXT(2)
02070 AV=AV+V(J,K)
02080 IF(V(J,K).LE.VMIN)GOTO 11
02090 IF(V(J,K).GT.VMAX)GOTO 12
02100 IH=FIX((V(J,K)-VMIN)/VINT+0.99999)+1
02110 IHIST(IH)=IHIST(IH)+1
02120 G0 T0 10
02130 11 IHIST(1)=IHIST(1)+1
02140 G0 T0 10
02150 12 IHIST(NGRP+2)=IHIST(NGRP+2)+1
02160 10 CONTINUE
02170 MAX=0
02180 D0 15 I=1,NGRP+2
02190 15 IF(IHIST(I).GT.MAX) MAX=IHIST(I)
02200 D0 14 I=1,NGRP+2
02210 VUPPER=VMIN+FLOAT(I-1)*VINT
02220 IH=IHIST(I)*110/MAX+1
02230 WRITE(7,2002)VUPPER,IHIST(I),(I,AST,J=1,IH)
02240 2002 FORMAT(* *,F6.2,I4,120A1)
02250 14 CONTINUE
02260 AV=AV/FLOAT(NEXT(1)*NEXT(2))
02270 WRITE(7,2000)AV
02280 2000 FORMAT(12X,*ARITHMETIC AVERAGE IS *,F6.2)
02290 RETURN
02300 END
02310C-----
02320C
02330C
02340 SUBROUTINE GAM(V,NEXT,SX,SY,RANGE,C,ENUG)
02350 DIMENSION V(9,9),NEXT(2),SINT(2)
02360 DIMENSION IGRA(100),GAMM(75),DIR(2)
02370 DATA IAST,IBLNK,IPLUS/1H*,1H ,1H+/
02380 DATA DIR(1),DIR(2)/10H VERTICAL ,10HHORIZONTAL/
02390 SINT(1)=SX
02400 SINT(2)=SY
02410 D0 15 M=1,2
02420 RMS=0.0
02430 IF(NEXT(M).EQ.1) G0 T0 15
02440 LAGLIM=NEXT(M)-1
02450 IF(LAGLIM.GT.75) LAGLIM=75
02460 D0 25 LAG=1,LAGLIM
02470 LIM=NEXT(M)-LAG
02480 GAMMA=0.0
02490 IF(M.EQ.2) G0 T0 21
02500C
02510 D0 11 I=1,LIM
02520 D0 11 J=1,NEXT(2)
02530 DIFF=V(I,J)-V(I+LAG,J)
02540 11 GAMMA=GAMMA+DIFF*DIFF
02550 COUN=FLOAT(LIM*NEXT(2))
02560 G0 T0 25
02570C
02580 21 D0 41 I=1,NEXT(1)
02590 D0 41 J=1,LIM
02600 DIFF=V(I,J)-V(I,J+LAG)
02610 41 GAMMA=GAMMA+DIFF*DIFF
02620 COUN=FLOAT(LIM*NEXT(1))
02630C
02640 25 GAMM(LAG)=GAMMA*.5/COUN
02650 BIG=C

```

```

02660 D0 14 LAG=1,LAGLIM
02670 14 IF (GAMM(LAG).GT.BIG) BIG=GAMM(LAG)
02680 A=RANGE
02690 WRITE (7,2001) DIR (M)
02700 2001 FORMAT (///10X,*SEMI-VARIOGRAM IN *,A10,* DIRECTION*)
02710 D0 10 LAG=1,LAGLIM
02720 GAMMA=GAMM(LAG)
02730 SOFAR=FLOAT(LAG)*SINT(M)
02740 HOVERA=SOFAR/A
02750 GMODEL=(1.5*HOVERA-(0.5*HOVERA*HOVERA*HOVERA))* (C-ENUG)+ENUG
02760 IF (HOVERA.GE.1.0) GMODEL=C
02770 RMS=RMS+( (GAMMA-GMODEL)/GMODEL)**2
02780 IMODEL=IF IX(GMODEL*98.0/BIG)+1
02790 IGAMMA=IF IX(GAMMA*98.0/BIG)+1
02800 D0 12 I=1,100
02810 12 IGRA(I)=IBLNK
02820 IGRA(IGAMMA)=IAST
02830 IGRA(IMODEL)=IPLUS
02840 IC=(NEXT(M)-LAG)*NEXT(3-M)
02850 WRITE (7,2000) LAG,IC,GAMMA,GMODEL,(IGRA(I),I=1,100)
02860 2000 FORMAT (1X,I3,1X,I5,1X,F7.2,1X,F7.2,*+*,100A1)
02870 10 CONTINUE
02880 RMS=100.0*SQRT(RMS/FLOAT(LAGLIM))
02890 WRITE (7,2003) RMS
02900 2003 FORMAT (* RMS = *,F6.1,* PERCENT*////)
02910 15 CONTINUE
02920 RETURN
02930 END
02940C -----
02950C
02960C
02970 SUBROUTINE SIM2D15 (V,NEXT,SINTX,SINTY,BB,PP,SMAXX,SMAXY,
02980+A,C,ENUG,AVER,NSEED)
02990C
03000C THIS SUBROUTINE SIMULATES VALUES WHICH FOLLOW SPH(A,C,ENUG)
03010C AND ARE N(AVER,C) .
03020C
03030C THE NUGGET EFFECT COMES FROM A RANDOM N(0,ENUG) .
03040C THE REST OF THE VARIATION (=CO=C-ENUG) COMES FROM N(AVER,CO)
03050C AND FOLLOWS SPH(A,CO,0) .
03060C ADDING THESE 2 TOGETHER PRODUCES VALUES WHICH COME FROM
03070C N(AVER,C) AND FOLLOWS SPH(A,C,ENUG) .
03080C
03090 DIMENSION V(9,9),NEXT(2)
03100 COMMON /DICK/Y(1000)
03110 DIMENSION X(15)
03120 XSEED=FLOAT(2*NSEED+1)
03130 CALL RANSET(XSEED)
03140 D0 71 I=1,100
03150 71 B=RANF(0.0)
03160C
03170 D0 30 J=1,NEXT(1)
03180 D0 30 K=1,NEXT(2)
03190 30 V(J,K)=0.0
03200C
03210 REWIND 5
03220 B=A/100.0
03230 D0 15 I=1,15
03240 X(1)=1.0
03250 X(2)=X(3)=0.0
03260 CALL COORDS(X)
03270 XINTX=X(I)
03280 X(1)=X(3)=0.0
03290 X(2)=1.0

```

```

03300 CALL COORDS (X)
03310 XINTY=X (I)
03320 DMAX=SMAXX*ABS (XINTX) +SMAXY*ABS (XINTY)
03330 NY=IROUND (2.0*DMAX/B) +2
03340 NORG=IROUND (-DMAX/B)
03350 CALL SIM (A, NY, B)
03360 DO 19 J=1, NEXT (1)
03370 XJ=BB+FLOAT (J-1) *SINTX-SMAXX
03380 DO 19 K=1, NEXT (2)
03390 XK=PP+FLOAT (K-1) *SINTY-SMAXY
03400 XC=XJ*XINTX+XK*XINTY
03410 NC=IROUND (XC/B) -NORG+1
03420 19 V (J, K) =V (J, K) +Y (NC)
03430 WRITE (5, 2002) NORG, XINTX, XINTY, NY
03440 2002 FORMAT (1X, I5, 2E12.6, I5)
03450 WRITE (5, 2001) (Y (M), M=1, NY)
03460 2001 FORMAT (1X, 10E12.6)
03470 15 CONTINUE
03480 ENDFILE 5
03490C
03500C      V HERE SHOULD BE NORMAL (0, 15)
03510C
03520 STD=SQRT ((C-ENUG) /15.0)
03530 SDENUG=SQRT (ENUG)
03540 DO 21 J=1, NEXT (1)
03550 DO 21 K=1, NEXT (2)
03560 CALL RANDOM (S, SDENUG)
03570 VX=V (J, K) *STD+S+AVER
03580C
03590C      V HERE SHOULD BE NORMAL (AVER, C)
03600C
03610 21 V (J, K) =VX
03620 RETURN
03630 END
03640C-----
03650C
03660C
03670 SUBROUTINE SIM (A, NS, B)
03680 COMMON /DICK/Y (1000)
03690 DIMENSION T (100)
03700 WF=SQRT (12.0*B / (A*(A*A+11.0*B*B)))
03710 DO 50 M=1, 100
03720 50 CALL RANDOM (T (M), 1.0)
03730 DO 53 I=1, NS
03740 YY=0.0
03750 DIS=-B/2.0
03760 DO 52 K=1, 50
03770 DIS=DIS+B
03780 52 YY=YY+DIS*(T (50+K) -T (51-K))
03790 Y (I) =YY*WF
03800 DO 54 M=1, 99
03810 54 T (M) =T (M+1)
03820 CALL RANDOM (T (100), 1.0)
03830 53 CONTINUE
03840 RETURN
03850 END
03860C-----
03870C
03880 SUBROUTINE COORDS (C)
03890 DIMENSION C (15), B (3, 2)
03900 DATA ((B (I, J), J=1, 2), I=1, 3), C0536, SIN36, C0572, SIN72/
03910+.52573111, -.85065081, .85065081, -.52573111, 1.0, 0.0,
03920+.80901699, .58778525, .30901699, .95105652/
03930 X=C (1)

```

```

03940 Y=C (2)
03950 Z=C (3)
03960 N=0
03970 DO 10 M=1,3
03980 COSANG=B (M,1)
03990 ZSINANG=Z*B (M,2)
04000 DO 11 I=1,5
04010 N=N+1
04020 C (N)=X*COSANG-ZSINANG
04030 TX=X
04040 X=X*COS72-Y*SIN72
04050 11 Y=TX*SIN72+Y*COS72
04060 TX=X
04070 IF (M.EQ.1) GO TO 12
04080 X=X*SIN72+Y*COS72
04090 Y=Y*SIN72-TX*COS72
04100 GO TO 10
04110 12 X=X*COS36-Y*SIN36
04120 Y=TX*SIN36+Y*COS36
04130 10 CONTINUE
04140 RETURN
04150 END
04160C-----
04170C
04180C
04190 SUBROUTINE RANDOM(S,SD)
04200 S=0.0
04210 IF (SD.EQ.0.0) RETURN
04220 DO 10 M=1,12
04230 10 S=S+RANF (0.0)
04240 S=(S-6.0)*SD
04250 RETURN
04260 END
04270C-----
04280C
04290 FUNCTION TREND(X,Y)
04300C CALCULATES CUBIC TREND FOR LN DEVT. LOBE WIDTHS (32.02TSS) FOR
04310C POINT AT IE=X IN=Y
04320 TREND=-2.2797E+02+1.9471*X+4.8560E-02*Y-3.3145E-04*X*X-
04330+2.4203E-03*X*Y+2.2078E-04*Y*Y+2.0525E-08*X*X*X-1.0132E-07*Y*Y*Y+
04340+2.0559E-07*X*X*Y+7.5072E-07*X*Y*Y
04350 RETURN
04360 END

```

SIMULATION NO. 1 SYSTEM NO. 1 NSEED = 1234

 BEFORE CONDITIONING (I.E. SHOULD FIT MODELS)

```

- -2.10 0*
- -1.90 0*
- -1.70 0*
- -1.50 0*
- -1.30 0*
- -1.10 0*
-  - .90 0*
-  - .70 2*****
-  - .50 8*****
-  - .30 9*****
-  - .10 10*****
-   .10 8*****
-   .30 14*****
-   .50 11*****
-   .70 14*****
-   .90 1*****
-  1.10 2*****
-  1.30 1*****
-  1.50 1*****
-  1.70 0*
-  1.90 0*
-  2.10 0*
  ARITHMETIC AVERAGE IS .11
  
```

SEMI-VARIOGRAM IN VERTICAL DIRECTION

1	72	.16	.14+
2	63	.17	.15+
3	54	.21	.17+
4	45	.26	.18+
5	36	.27	.20+
6	27	.32	.21+
7	18	.40	.22+
8	9	.31	.24+

RMS = 41.6 PERCENT



SEMI-VARIOGRAM IN HORIZONTAL DIRECTION

1	72	.18	.13+
2	63	.17	.15+
3	54	.13	.16+
4	45	.18	.17+
5	36	.26	.19+
6	27	.15	.20+
7	18	.15	.21+
8	9	.26	.22+

RMS = 24.5 PERCENT



AFTER CONDITIONING BEFORE TAKING EXP

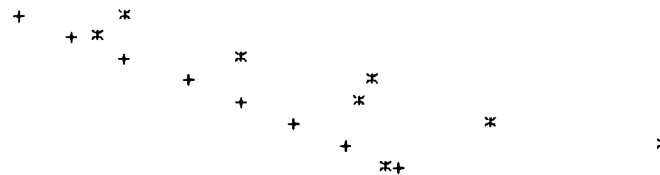
```

- -2.10 0*
- -1.90 0*
- -1.70 0*
- -1.50 0*
- -1.30 0*
- -1.10 0*
- -.90 3*****
- -.70 6*****
- -.50 7*****
- -.30 10*****
- -.10 10*****
- .10 15*****
- .30 15*****
- .50 6*****
- .70 6*****
- .90 3*****
- 1.10 0*
- 1.30 0*
- 1.50 0*
- 1.70 0*
- 1.90 0*
- 2.10 0*
    ARITHMETIC AVERAGE IS  -.08
  
```

SEMI-VARIOGRAM IN VERTICAL DIRECTION

1	72	.16	.14+
2	63	.16	.15+
3	54	.20	.17+
4	45	.23	.18+
5	36	.23	.20+
6	27	.26	.21+
7	18	.31	.22+
8	9	.23	.24+

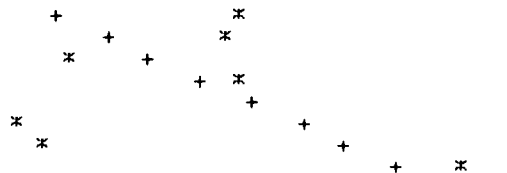
RMS = 22.0 PERCENT



SEMI-VARIOGRAM IN HORIZONTAL DIRECTION

1	72	.18	.13+
2	63	.18	.15+
3	54	.14	.16+
4	45	.18	.17+
5	36	.25	.19+
6	27	.12	.20+
7	18	.13	.21+
8	9	.24	.22+

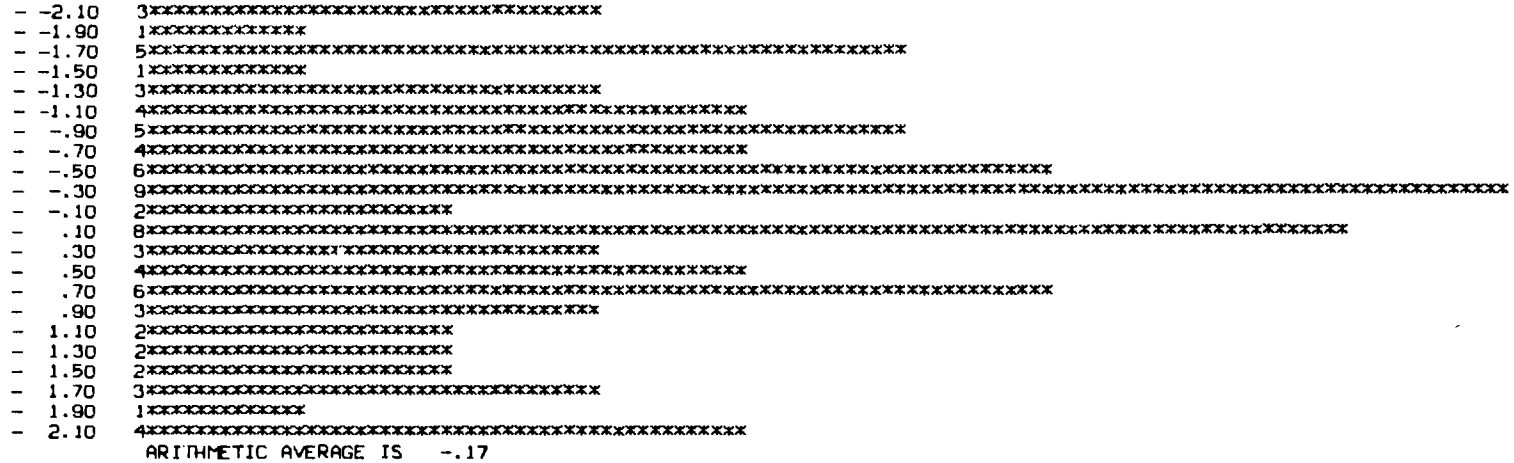
RMS = 28.3 PERCENT



MEREDUX 310 LEVEL

SIMULATION NO. 1 SYSTEM NO. 2 NSEED = 1235

BEFORE CONDITIONING (I.E. SHOULD FIT MODELS)



SEMI-VARIOGRAM IN VERTICAL DIRECTION

1	72	1.12	1.26+
2	63	1.43	1.32+
3	54	.99	1.38+
4	45	1.29	1.44+
5	36	1.40	1.50+
6	27	.89	1.56+
7	18	1.61	1.61+
8	9	2.34	1.67+

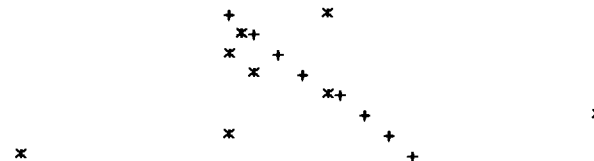
RMS = 23.9 PERCENT



SEMI-VARIOGRAM IN HORIZONTAL DIRECTION

1	72	1.45	1.25+
2	63	1.27	1.30+
3	54	1.25	1.36+
4	45	1.29	1.41+
5	36	1.44	1.46+
6	27	1.95	1.51+
7	18	1.25	1.56+
8	9	.87	1.61+

RMS = 21.7 PERCENT



AFTER CONDITIONING BEFORE TAKING EXP

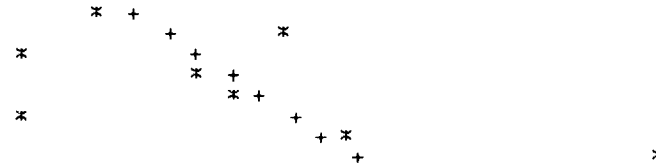
```

- -2.10 16*****
- -1.90 5*****
- -1.70 2*****
- -1.50 6*****
- -1.30 4*****
- -1.10 5*****
- -.90 6*****
- -.70 7*****
- -.50 7*****
- -.30 5*****
- -.10 3*****
- .10 4*****
- .30 0*
- .50 ~*****
- .70 0*
- .90 0*
- 1.10 2*****
- 1.30 4*****
- 1.50 2*****
- 1.70 0*
- 1.90 1*****
- 2.10 0*
    ARITHMETIC AVERAGE IS -1.03
    
```

SEMI-VARIOGRAM IN VERTICAL DIRECTION

1	72	1.18	1.26+
2	63	1.53	1.32+
3	54	1.04	1.38+
4	45	1.36	1.44+
5	36	1.44	1.50+
6	27	1.04	1.56+
7	18	1.66	1.61+
8	9	2.23	1.67+

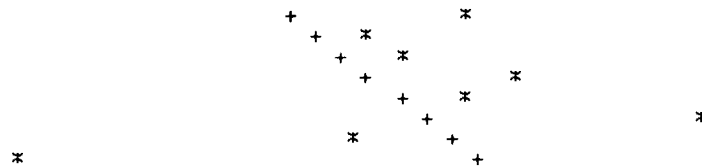
RMS = 19.9 PERCENT



SEMI-VARIOGRAM IN HORIZONTAL DIRECTION

1	72	1.58	1.25+
2	63	1.40	1.30+
3	54	1.46	1.36+
4	45	1.68	1.41+
5	36	1.57	1.46+
6	27	2.03	1.51+
7	18	1.37	1.56+
8	9	.75	1.61+

RMS = 26.0 PERCENT




```

                                AFTER TAKING EXPONENTIALS
-   0   0*
-  .20  0*
-  .40  2*****
-  .60  14*****
-  .80  14*****
- 1.00  14*****
- 1.20  13*****
- 1.40  10*****
- 1.60  5*****
- 1.80  2*****
- 2.00  4*****
- 2.20  1*****
- 2.40  1*****
- 2.60  1*****
- 2.80  0*
- 3.00  0*
- 3.20  0*
- 3.40  0*
- 3.60  0*
- 3.80  0*
- 4.00  0*
- 4.20  0*
      ARITHMETIC AVERAGE IS   1.02

```

```

SEMI-VARIOGRAM IN VERTICAL DIRECTION
1  72  .19  .14+
2  63  .19  .15+
3  54  .19  .17+
4  45  .24  .18+
5  36  .23  .20+
6  27  .26  .21+
7  18  .31  .22+
8   9  .29  .24+
RMS = 27.7 PERCENT

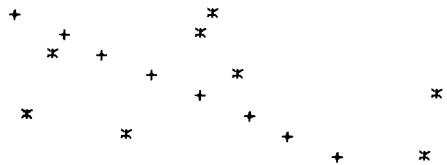
```



```

SEMI-VARIOGRAM IN HORIZONTAL DIRECTION
1  72  .19  .13+
2  63  .18  .15+
3  54  .15  .16+
4  45  .19  .17+
5  36  .25  .19+
6  27  .14  .20+
7  18  .17  .21+
8   9  .25  .22+
RMS = 26.2 PERCENT

```



```

- 0 0*
- .50 52*****
- 1.00 16*****
- 1.50 2*****
- 2.00 2*****
- 2.50 0*
- 3.00 2*****
- 3.50 2*****
- 4.00 3*****
- 4.50 1***
- 5.00 0*
- 5.50 0*
- 6.00 0*
- 6.50 1***
- 7.00 0*
- 7.50 0*
- 8.00 0*
- 8.50 0*
- 9.00 0*
- 9.50 0*
- 10.00 0*
- 10.50 0*

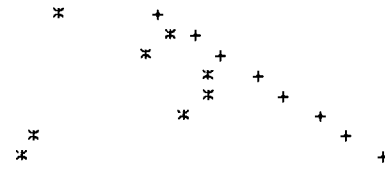
```

ARITHMETIC AVERAGE IS .76

SEMI-VARIOGRAM IN VERTICAL DIRECTION

1	72	1.07	1.26+
2	63	1.28	1.32+
3	54	1.23	1.38+
4	45	1.35	1.44+
5	36	1.35	1.50+
6	27	1.30	1.56+
7	18	1.02	1.61+
8	9	1.01	1.67+

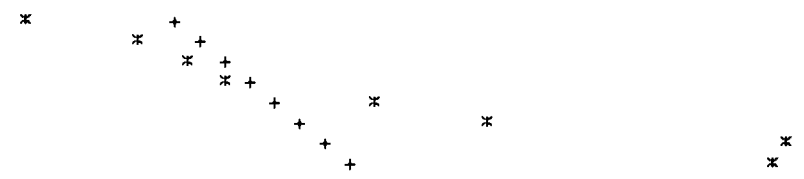
RMS = 21.5 PERCENT



SEMI-VARIOGRAM IN HORIZONTAL DIRECTION

1	72	.95	1.25+
2	63	1.17	1.30+
3	54	1.26	1.36+
4	45	1.36	1.41+
5	36	1.64	1.46+
6	27	1.88	1.51+
7	18	2.47	1.56+
8	9	2.46	1.61+

RMS = 31.1 PERCENT



FOR ALL 81. SIMULATED SAMPLES OF SIMULATION NO. 1
LW LA
1.02 .76

Appendix 13 : Program SIM2BLO

The listing of the program SIM2BLO is shown overleaf.

The required input to the program is :-

TITLE - A title phrase of up to 80 characters.

IY1 IX1 IY50 IX50 - The Easting and height above datum of the most Westerly and most Easterly samples on the lower level (335) drive.

IY51 IX51 IY100 IX100 - The Easting and height above datum of the most Westerly and most Easterly samples on the upper level (310) drive.

F - An array of 6 stores. The first 3 stores contain the range, sill, and nugget effect of the semi-variogram model of the first system (lode widths). The second set of 3 stores contains the same information for the second system (lode assays).

IFN - A number which defines whether a sample's measurements are in imperial (IFN=1) or metric (IFN=2) units.

D1 - The sample lode width measurement.

D2 - The sample lode assay measurement.

N2 - The number of simulations to be considered.

An example of the output of the program is shown overleaf.

```

00100 PROGRAM SIM2BL0 (INPUT=131B,TAPE3=INPUT,TAPE7,
00110+TAPE1=131B,TAPE2=131B,TAPE4=131B)
00120C
00130C TAPE1=INPUT FOR WL SIMULATED VALUES
00140C TAPE2=INPUT FOR VL SIMULATED VALUES
00150C TAPE3=INPUT FOR GENERAL DATA AND DEV. DATA
00160C TAPE4=TEMPORARY STORE FOR KRIGING WEIGHTS
00170C TAPE7=OUTPUT FOR SIMULATION AVERAGE ETC.
00180C
00190C
00200C          DEV. DATA
00210C          -----47M.  310FM. LEVEL
00220C          +          +
00230C          +          +
00240C          +          +
00250C          ++++++++23.5M.
00260C          +          +
00270C          +          +
00280C          -----0M.  335FM. LEVEL
00290C          DEV. DATA
00300C          0          25          50          75M.
00310C
00320C THE BLOCK IS REPRESENTED BY A 16*8 GRID (I.E.3M. GRID WITH
00330C POINTS AT 26.56,48.44,3.13 AND 1.47,45.53,2.94)
00340C THE 100 SIMULATED DEVELOPMENT DATA POINTS ON THE 2 LEVELS ARE SITED
00350C EQUALLY EVERY 75/49=1.53M.=5.02FT. (I.E. POINTS ARE AT
00360C 0.0,75.0,1.53) .
00370C 100 ACTUAL DEVELOPMENT DATA SAMPLES ARE READ IN AND ASSIGNED THESE
00380C SAME LOCATIONS (I.E. THERE MAY BE A SMALL POSITIONAL ERROR) .
00390C THE CONDITIONISATION PROCESS MUST USE THE SAME SAMPLE PATTERN OF
00400C SIMULATED AND ACTUAL DEVELOPMENT DATA FOR FINDING 'SK' AND 'AK'
00410C
00420 COMMON X(100),Y(100),WL(2,100),VL(2,100)
00430 COMMON DATA(2,100),SDATA(2,100)
00440 COMMON TITLE(8),F(2,3),NEAR(18),D(18),W(8),V(8)
00450 COMMON LIFT(20,4),SWL(8),SVL(8)
00460 READ(3,1000)(TITLE(K),K=1,8)
00470 1000 FORMAT(1X,8A10)
00480 READ(3,1006)IY1,IX1,IY50,IX50,IY51,IX51,IY100,IX100
00490 1006 FORMAT(1X,8I4)
00500 READ(3,1001)((F(NV,K),K=1,3),NV=1,2)
00510 1001 FORMAT(1X,3F6.3)
00520 WRITE(7,1000)(TITLE(K),K=1,8)
00530 WRITE(7,1006)IY1,IX1,IY50,IX50,IY51,IX51,IY100,IX100
00540 WRITE(7,1001)((F(NV,K),K=1,3),NV=1,2)
00550 N=0
00560 XLEVEL=0.0
00570 DIFF=75.0/49.0
00580C
00590 DO 10 I=1,2
00600 EAST=-DIFF
00610 DO 11 J=1,50
00620 READ(3,1002)IFN,D1,D2
00630 1002 FORMAT(9X,I1,F5.2,F7.2)
00640 IF(D1.EQ.0.0) GO TO 11
00650 IF(IFN.EQ.2) GO TO 12
00660 D1=0.3048*(FLOAT(IFIX(2.0*D1+0.5)))/2.0)
00670 IF(D2.NE.99.11) D2=0.043*(FLOAT(IFIX((D2+2.0)/2.0))*2.0-1.0)
00680 IF(D2.EQ.99.11) D2=0.043*0.5
00690 GO TO 13
00700 12 D1=FLOAT(IFIX(5.0*D1+0.5))/5.0
00710 IF(D2.NE.99.11) D2=FLOAT(IFIX(20.0*D2+0.5))/20.0
00720 IF(D2.EQ.99.11) D2=0.025
00730 13 N=N+1

```

```

00740 SDATA(1,N)=D1
00750 SDATA(2,N)=D2
00760 DATA(1,N)=ALOG(D1)
00770 DATA(2,N)=ALOG(D2)
00780 X(N)=XLEVEL
00790 Y(N)=EAST+FLOAT(J)*DIFF
00800 11 WRITE(7,2002)N,D1,D2
00810 2002 FORMAT(2X,I4,F5.2,F7.2)
00820 10 XLEVEL=47.0
00830C
00840 SINTX=47.0/16.0
00850 SINTY=25.0/8.0
00860 S1SIGK2=S2SIGK2=0.0
00870C
00880 DO 15 I=1,16
00890 XI=FLOAT(I-1)*SINTX+0.5*SINTX
00900 DO 15 J=1,8
00910 YJ=25.0+FLOAT(J-1)*SINTY+0.5*SINTY
00920 CALL SEARCH(X,Y,N,F(1,1),F(1,2),F(1,3),XI,YJ,SIGK2)
00930 S1SIGK2=S1SIGK2+SIGK2/128.0
00940 CALL SEARCH(X,Y,N,F(2,1),F(2,2),F(2,3),XI,YJ,SIGK2)
00950 S2SIGK2=S2SIGK2+SIGK2/128.0
00960 15 CONTINUE
00970C
00980 ENDFILE 4
00990 WRITE(7,3004)
01000 3004 FORMAT(1H1/////10X,* DATA WL VS. DATA VL*/11X,19(1H-)//
01010+32X,*DATA WL*)
01020 DO 88 I=1,N
01030 X(I)=SDATA(1,I)
01040 88 Y(I)=SDATA(2,I)
01050 CALL PLOTY(X,Y,N)
01060 READ(3,1008)N2
01070 1008 FORMAT(1X,I3)
01080 WRITE(7,2006)S1SIGK2,S2SIGK2
01090 2006 FORMAT(///5X,*AVERAGE SIGK2 OVER BOTH BLOCKS FOR*/
01100+10X,* WL VL */10X,2F6.4)
01110 CALL GETPF(5HTAPE1,7HWSIM100,7HUMCKA07)
01120 CALL GETPF(5HTAPE2,7HVSIM100,7HUMCKA07)
01130 WRITE(7,2000)
01140 2000 FORMAT(1H1////,18X,*LOWER BLOCK UPPER BLOCK*,
01150+/5X,*SIMULATION*,5X,*WL VL*,9X,*WL VL*/5X,40(1H-))
01160 DO 50 I=1,20
01170 DO 50 J=1,4
01180 50 LIFT(I,J)=0
01190 SR=ST=0.0
01200 RECN2=1.0/FLOAT(N2)
01210C
01220 DO 101 NSIM=1,N2
01230 SUMXY=SUMX=SUMY=SUMX2=SUMY2=0.0
01240 READ(1,1003)(SDATA(1,M),M=1,100)
01250 READ(2,1003)(SDATA(2,M),M=1,100)
01260 1003 FORMAT(1X,10F6.0)
01270 DO 20 M=1,100
01280 SDATA(1,M)=SDATA(1,M)/1000.0
01290 SDATA(2,M)=SDATA(2,M)/1000.0
01300 20 CONTINUE
01310 REWIND 4
01320 NB=1
01330 NT=8
01340C
01350 DO 19 NBLOCK=1,2
01360 SUMWL=SUMVL=0.0
01370C

```

```

01380 D0 16 I=NB,NT
01390 INB=I-NB+1
01400 SWL (INB)=SVL (INB)=0.0
01410 READ (1,1003) (W(K),K=1,8)
01420 READ (2,1003) (V(K),K=1,8)
01430 D0 14 J=1,8
01440 READ (4,1004) NS
01450 READ (4,1005) ((NEAR(K),D(K)),K=1,NS)
01460 AK=SK=0.0
01470 D0 17 K=1,NS
01480 SK=SK+D(K)*SDATA(1,NEAR(K))
01490 17 AK=AK+D(K)*DATA(1,NEAR(K))
01500 WLL=(W(J)/1000.0-SK+AK)
01510 WLL=EXP(WLL)
01520 W(J)=WLL
01530C W(J)=.13100068E+05-.15468835E+01*YJ-.25478477E+02*XI+
01540C+.35826833E-01*WLL+.97820590E-04*YJ*YJ+.16512423E-01*XI*XI+
01550C+.16650438E+00*WLL*WLL+.20147847E-02*YJ*XI+
01560C+.15368569E-07*YJ*YJ*YJ-.35656412E-05*XI*XI*XI-
01570C+.67110721E-07*YJ*YJ*XI-.65576605E-06*YJ*XI*XI
01580C IF (W(J).LT.WLL) W(J)=WLL
01590 SWL (INB)=SWL (INB)+W(J)
01600 SUMX=SUMX+W(J)
01610 SUMX2=SUMX2+W(J)*W(J)
01620 READ (4,1004) NS
01630 1004 FORMAT(9X,I2)
01640 READ (4,1005) ((NEAR(K),D(K)),K=1,NS)
01650 1005 FORMAT(I2,F6.4,8(I3,F6.4)/I2,F6.4,8(I3,F6.4))
01660 AK=SK=0.0
01670 D0 18 K=1,NS
01680 SK=SK+D(K)*SDATA(2,NEAR(K))
01690 18 AK=AK+D(K)*DATA(2,NEAR(K))
01700 VLL=EXP(V(J)/1000.0-SK+AK)
01710 IF (VLL.GT.100.0) VLL=100.0
01720 V(J)=VLL
01730C V(J)=.65726173E-01+.61670581*VLL-.86895918E-02*VLL*VLL
01740C IF (V(J).LT.(WLL*VLL/W(J))) V(J)=WLL*VLL/W(J)
01750 SVL (INB)=SVL (INB)+V(J)
01760 SUMXY=SUMXY+W(J)*V(J)
01770 SUMY=SUMY+V(J)
01780 14 SUMY2=SUMY2+V(J)*V(J)
01790 SUMWL=SUMWL+SWL (INB)
01800 SUMML=SUMML+SVL (INB)
01810 16 CONTINUE
01820C
01830 VL (NBL0CK,NSIM)=SUMML/64.0
01840 WL (NBL0CK,NSIM)=SUMWL/64.0
01850 D0 21 NPAY=1,20
01860 PAY=FLOAT(NPAY)/10.0
01870 I=-1
01880 22 I=I+1
01890 23 I=I+1
01900 IF (I.GT.6) GO TO 21
01910 IF ((SVL (I)/8.0).GE.PAY) GO TO 23
01920 IF ((SVL (I+1)/8.0).GE.PAY) GO TO 22
01930 SN=S3=0.0
01940 D0 24 J=I,8
01950 SN=SN+8.0
01960 24 S3=S3+SVL (J)
01970 IF ((S3/SN).GE.PAY) GO TO 25
01980 LIFT (NPAY,2*NBL0CK)=LIFT (NPAY,2*NBL0CK)+1
01990 GO TO 21
02000 25 LIFT (NPAY,2*NBL0CK-1)=LIFT (NPAY,2*NBL0CK-1)+1
02010 21 CONTINUE

```

```

02020 NB=9
02030 NT=16
02040 19 CONTINUE
02050C
02060 WRITE (7,2001) NSIM,WL (1,NSIM),VL (1,NSIM),WL (2,NSIM),VL (2,NSIM)
02070 2001 FORMAT (9X,I3,6X,F5.2,1X,F5.2,5X,F5.2,1X,F5.2)
02080 T=SUMXY-(SUMX*SUMY/128.0)
02090 TT=T/SQRT((SUMX2-SUMX*SUMX/128.0)*(SUMY2-SUMY*SUMY/128.0))
02100 ST=ST+REC2*ABS(TT)*SQRT(126.0)/SQRT(1.0-TT*TT)
02110 SR=SR+REC2*TT
02120 IF(NSIM.NE.65) GO TO 101
02130 CALL GETPF(5HTAPE1,7HWLBLOCK,7HUMCKA07)
02140 CALL GETPF(5HTAPE2,7HVLBLOCK,7HUMCKA07)
02150 101 CONTINUE
02160C
02170 SUMWL=SUMVL=SUMWL2=SUMVL2=0.0
02180 DO 70 NSIM=1,N2
02190 SUMWL=SUMWL+WL (1,NSIM)
02200 SUMVL=SUMVL+VL (1,NSIM)
02210 SUMWL2=SUMWL2+WL (2,NSIM)
02220 SUMVL2=SUMVL2+VL (2,NSIM)
02230 70 CONTINUE
02240 SUMVL=SUMVL*REC2
02250 SUMWL=SUMWL*REC2
02260 SUMVL2=SUMVL2*REC2
02270 SUMWL2=SUMWL2*REC2
02280 WRITE (7,2003) SUMWL,SUMVL,SUMWL2,SUMVL2
02290 2003 FORMAT (9X,36(1H-)/9X,9HAVERAGE ,F5.2,1X,F5.2,5X,F5.2,1X,F5.2////
02300+16X,19(1H-)/16X,1H!,3X,*LOWER/UPPER*,3X,*! 2 NON-PAY LIFTS WHEN*,
02310+* REST OF BLOCK IS*/10X,25(1H-)/
02320+10X,*! PAY ! P/P P/N N/P N/N ! P/ N/ /P /N*/10X,25(1H-)
02330 DO 72 NPAY=1,20
02340 PAY=FLOAT(NPAY)/10.0
02350 NYESYES=NYESNO=NNYES=NNNO=0
02360 DO 71 NSIM=1,N2
02370 IF(VL (1,NSIM).GT.PAY.AND.VL (2,NSIM).GT.PAY) NYESYES=NYESYES+1
02380 IF(VL (1,NSIM).GT.PAY.AND.VL (2,NSIM).LE.PAY) NYESNO=NYESNO+1
02390 IF(VL (1,NSIM).LE.PAY.AND.VL (2,NSIM).GT.PAY) NNYES=NNYES+1
02400 71 IF(VL (1,NSIM).LE.PAY.AND.VL (2,NSIM).LE.PAY) NNNO=NNNO+1
02410 72 WRITE (7,2004) PAY,NYESYES,NYESNO,NNYES,NNNO,(LIFT(NPAY,J),J=1,4)
02420 2004 FORMAT (10X,2H!,F3.1,2X,4(1X,I3),2H!,4(1X,I3))
02430 WRITE (7,2005) N2,SR,ST
02440 2005 FORMAT (////10X,*AVERAGE CORR. COEFF. BETWEEN WL AND VL*,
02450+* OVER *,I3,* SIMULATIONS = *,F6.4/10X,*AVERAGE T-TEST VALUE = *,F5.2)
02460C
02470 WRITE (7,3000)
02480 3000 FORMAT (1H1////10X,*WL LOWER VS. VL LOWER*/10X,21(1H-)//
02490+31X,8HWL LOWER)
02500 DO 30 I=1,N2
02510 X(I)=WL (1,I)
02520 30 Y(I)=VL (1,I)
02530 CALL PLOTY(X,Y,N2)
02540 WRITE (7,3001)
02550 3001 FORMAT (1H1////10X,*WL UPPER VS. VL UPPER*/10X,21(1H-)//
02560+31X,8HWL UPPER)
02570 DO 31 I=1,N2
02580 X(I)=WL (2,I)
02590 31 Y(I)=VL (2,I)
02600 CALL PLOTY(X,Y,N2)
02610 WRITE (7,3002)
02620 3002 FORMAT (1H1////10X,*WL LOWER VS. WL UPPER*/10X,21(1H-)//
02630+31X,8HWL LOWER)
02640 DO 32 I=1,N2
02650 X(I)=WL (1,I)

```

```

02660 32 Y(I)=VL(2,I)
02670 CALL PLOTY(X,Y,N2)
02680 WRITE(7,3003)
02690 3003 FORMAT(1H1/////10X,*VL LOWER VS. VL UPPER*/10X,21(1H-)//
02700+31X,8HVL LOWER)
02710 DO 33 I=1,N2
02720 X(I)=VL(1,I)
02730 33 Y(I)=VL(2,I)
02740 CALL PLOTY(X,Y,N2)
02750C
02760 ENDFILE 7
02770 CALL REPLACE(5HTAPE7,7HSIM2OUT)
02780 STOP
02790 END
02800C+++++
02810C
02820 SUBROUTINE SEARCH(X,Y,N,RANG,C,ENUG,XI,YJ,SIGK2)
02830 DIMENSION X(100),Y(100)
02840 DIMENSION A(19,19),D(19),DD(19)
02850 DIMENSION LB(100),NEAR(18),DISTM(18)
02860 THIRTY=ASIN(0.5)
02870 HALF=THIRTY/60.0
02880 SIXTY=THIRTY*2.0
02890 XNINETY=THIRTY*3.0
02900 ONE80=THIRTY*6.0
02910 THREE60=THIRTY*12.0
02920C
02930 RANGE=RANG
02940 RADIUS=RANGE
02950 40 NS=0
02960 DO 12 M=1,N
02970 MM=M
02980 XDIFF=ABS(X(M)-XI)
02990 YDIFF=ABS(Y(M)-YJ)
03000 IF(XDIFF.LT.0.05.AND.YDIFF.LT.0.05) GO TO 50
03010 IF(YDIFF.GT.RADIUS) GO TO 12
03020 IF(XDIFF.GT.RADIUS) GO TO 12
03030 DIST=SQRT(XDIFF*XDIFF+YDIFF*YDIFF)
03040 IF(DIST.GT.RADIUS) GO TO 12
03050 NS=NS+1
03060 LB(NS)=M
03070 12 CONTINUE
03080 IF(NS.GT.4) GO TO 41
03090C IF THERE ARE 4 OR LESS POINTS WITHIN THE SEARCH CIRCLE ,
03100C THE RADIUS OF THE SEARCH CICLE IS INCREASED BY HALF THE RANGE
03110 42 RADIUS=RADIUS+0.5*RANGE
03120 GO TO 40
03130C
03140 41 DO 20 M=1,18
03150 NEAR(M)=-99
03160 20 DISTM(M)=100000.0
03170C THE DATA POINTS WITHIN THE SEARCH CIRCLE ARE SORTED. THE NEAREST
03180C 3 POINTS IN EACH SWEEP OF 61 DEGREES CENTRED ON BEARINGS 0,300,60
03190C ARE TAKEN. THEREFORE THERE CAN BE NO MORE THAN 18 POINTS IN
03200C THE KRIGING SYSTEM.
03210 DO 24 K=1,NS
03220 XDIFF=(X(LB(K))-XI)
03230 YDIFF=(Y(LB(K))-YJ)
03240 ANGLE=ATAN(YDIFF/XDIFF)
03250 IF(XDIFF.EQ.0.0) ANGLE=ONE80-XNINETY*YDIFF/ABS(YDIFF)
03260 IF(XDIFF.LT.0.0) ANGLE=ANGLE+ONE80
03270 IF(ANGLE.LT.0.0) ANGLE=ANGLE+THREE60
03280 ANGLE=ANGLE+THIRTY
03290 IF(ANGLE.GT.THREE60) ANGLE=ANGLE-THREE60

```



```

03300 D0 24 M=1,6
03310 IF (ANGLE.LT. (SIXTY*FLOAT(M-1)+HALF)) GO TO 24
03320 IF (ANGLE.GT. (SIXTY*FLOAT(M)+HALF)) GO TO 24
03330 DIST=SQRT(XDIFF*XDIFF+YDIFF*YDIFF)
03340 IF (DIST.GE.DISTMN(3*M)) GO TO 24
03350 IF (DIST.GE.DISTMN(3*M-1)) GO TO 23
03360 IF (DIST.GE.DISTMN(3*M-2)) GO TO 22
03370 DISTMN(3*M)=DISTMN(3*M-1)
03380 NEAR(3*M)=NEAR(3*M-1)
03390 DISTMN(3*M-1)=DISTMN(3*M-2)
03400 NEAR(3*M-1)=NEAR(3*M-2)
03410 DISTMN(3*M-2)=DIST
03420 NEAR(3*M-2)=LB(K)
03430 GO TO 24
03440 22 DISTMN(3*M)=DISTMN(3*M-1)
03450 NEAR(3*M)=NEAR(3*M-1)
03460 DISTMN(3*M-1)=DIST
03470 NEAR(3*M-1)=LB(K)
03480 GO TO 24
03490 23 DISTMN(3*M)=DIST
03500 NEAR(3*M)=LB(K)
03510 24 CONTINUE
03520C
03530 NS=1
03540 D0 29 M=1,18
03550 D0 25 K=1,18
03560C IF A POINT IS INCLUDED TWICE (I.E. IT IS IN THE 1 DEGREE
03570C OVERLAP BETWEEN 2 SWEEPS) THEN ONE COPY IS REMOVED
03580 25 IF (NEAR(K).EQ.NEAR(M).AND.K.NE.M) NEAR(M)=-99
03590 IF (NEAR(M).EQ.-99) GO TO 29
03600 NEAR(NS)=NEAR(M)
03610 NS=NS+1
03620 29 CONTINUE
03630C
03640C IF THERE ARE LESS THAN 4 POINTS(=5 EQNS.) IN THE KRIGING SYSTEM
03650C THE SEARCH CIRCLE RADIUS IS INCREASED AND THE SYSTEM RE-CALCULATED
03660 IF (NS.LT.5) GO TO 42
03670C
03680 D0 13 K=1,NS-2
03690 D0 13 L=K+1,NS-1
03700 XDIFF=(X(NEAR(K))-X(NEAR(L)))
03710 YDIFF=(Y(NEAR(K))-Y(NEAR(L)))
03720 DIST=SQRT(XDIFF*XDIFF+YDIFF*YDIFF)
03730 13 A(K,L)=GSPH(DIST,RANGE,C,ENUG)
03740 D0 17 K=2,NS-1
03750 D0 17 L=1,K-1
03760 17 A(K,L)=A(L,K)
03770 D0 14 K=1,NS
03780 A(K,NS)=A(NS,K)=1.0
03790 14 A(K,K)=0.0
03800 D0 15 K=1,NS-1
03810 XDIFF=(X(NEAR(K))-X)
03820 YDIFF=(Y(NEAR(K))-Y)
03830 DIST=SQRT(XDIFF*XDIFF+YDIFF*YDIFF)
03840 15 D(K)=DD(K)=GSPH(DIST,RANGE,C,ENUG)
03850 D(NS)=DD(NS)=1.0
03860 CALL KARON(A,D,NS)
03870 SIGK2=D(NS)
03880 NS=NS-1
03890 D0 10 M=1,NS
03900 10 SIGK2=SIGK2+D(M)*DD(M)
03910 WRITE(4,4000)NS,(NEAR(M),D(M)),M=1,NS)
03920C NEAR(1) OR NEAR(10) SHOULD NOT BE 100 SO SPACE CAN BE SAVED BY USING I2
03930 4000 FORMAT(9X,I2,2(/I2,F6.4,8(I3,F6.4)))

```

```

03940 RETURN
03950C
03960C THE POINT BEING CONSIDERED IS A DATA POINT
03970 50 WRITE(4,2001)MM
03980 2001 FORMAT(9X,2H 1/1X,I2,6H1.0000)
03990 SIGK2=0.0
04000C
04010 RETURN
04020 END
04030C-----
04040C
04050C
04060 FUNCTION IROUND(X)
04070 IROUND=IFIX(ABS(X)+0.5)
04080 IF(X.LT.0.0) IROUND=-IROUND
04090 RETURN
04100 END
04110C-----
04120C
04130C
04140 FUNCTION GSPH(DIST,A,C,ENUG)
04150 IF(DIST.GE.A) GO TO 10
04160 X=DIST/A
04170 GSPH=(C-ENUG)*(1.5*X-0.5*X*X*X)+ENUG
04180 RETURN
04190 10 GSPH=C
04200 RETURN
04210 END
04220C-----
04230C
04240C
04250 SUBROUTINE KARON(A,D,NPAR)
04260 DIMENSION A(19,19),D(19)
04270 TOL=0.000001
04280 N=NPAR
04290 DO 65 J=1,N
04300 JY=J+1
04310 BIGA=0.
04320 DO 30 I=J,N
04330 IF (ABS(BIGA)-ABS(A(I,J))) 35,30,30
04340 35 BIGA=A(I,J)
04350 IMAX=I
04360 30 CONTINUE
04370 IF (ABS(BIGA)-TOL) 10,10,40
04380 40 DO 50 K=J,N
04390 SAVE=A(IMAX,K)/BIGA
04400 A(IMAX,K)=A(J,K)
04410 A(J,K)=SAVE
04420 50 CONTINUE
04430 SAVE=D(IMAX)/BIGA
04440 D(IMAX)=D(J)
04450 D(J)=SAVE
04460 IF (J-N) 55,70,55
04470 55 DO 65 IX=JY,N
04480 DO 60 JX=JY,N
04490 60 A(IX,JX)=A(IX,JX)-A(IX,J)*A(J,JX)
04500 65 D(IX)=D(IX)-D(J)*A(IX,J)
04510 70 NY=N-1
04520 DO 80 J=1,NY
04530 IB=N-J
04540 DO 80 K=1,J
04550 IC=N-K+1
04560 80 D(IB)=D(IB)-A(IB,IC)*D(IC)
04570 RETURN

```

```

04580 10 WRITE (7,1000) NPAR
04590 1000 FORMAT (5X, *THERE IS NO SOLUTION WITH NPAR =*, I3, * SO THERE*)
04600 STOP
04610 END
04620C-----
04630 SUBROUTINE PLOTY (X, Y, N4)
04640C PLOTS X (N2) ACROSS AGAINST Y (N2) DOWN
04650 DIMENSION X (100), Y (100)
04660 DIMENSION IP (53), IG (53)
04670 SUMX=SUMX2=SUMXY=SUMY2=SUMY=0.0
04680 N2=N4
04690 DO 20 I=1, N2
04700 NCH=0
04710 DO 21 J=1, N2-1
04720 IF (Y (J) .LE. Y (J+1)) GO TO 21
04730 TT=Y (J)
04740 Y (J)=Y (J+1)
04750 Y (J+1)=TT
04760 TT=X (J)
04770 X (J)=X (J+1)
04780 X (J+1)=TT
04790 NCH=1
04800 21 CONTINUE
04810 IF (NCH.EQ.0) GO TO 22
04820 20 CONTINUE
04830 22 WLMAX=-100000.0
04840 DO 23 I=1, N2
04850 SUMX=SUMX+X (I)
04860 SUMX2=SUMX2+X (I) *X (I)
04870 SUMXY=SUMXY+X (I) *Y (I)
04880 SUMY2=SUMY2+Y (I) *Y (I)
04890 SUMY=SUMY+Y (I)
04900 23 IF (X (I) .GT. WLMAX) WLMAX=X (I)
04910 W1=WLMAX/4.0
04920 W2=WLMAX/2.0
04930 W3=WLMAX*3.0/4.0
04940 WRITE (7, 2002) W1, W2, W3, WLMAX
04950 2002 FORMAT (9X, 3H0 , 4 (8X, F5.2) /8X, 2H+!, 4 (12 (1H+), 1H!))
04960 JS0FAR=1
04970 VL INT=Y (N2) /40.0
04980 DO 24 I=1, 41
04990 ABOVE=FLOAT (I) *VL INT
05000 DO 25 J=1, 53
05010 25 IG (J)=1
05020 IF (JS0FAR.GT.N2) GO TO 27
05030 DO 26 J=1, N2-JS0FAR+1
05040 IF (Y (JS0FAR) .GT. ABOVE) GO TO 27
05050 XK=52.0*(X (JS0FAR)) / (WLMAX)
05060 K=IROUND (XK) +1
05070 IG (K)=IG (K) +1
05080 JS0FAR=JS0FAR+1
05090 26 CONTINUE
05100 27 DO 28 J=1, 53
05110 IF (IG (J) .GT. 11) IG (J)=11
05120 GO TO (30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40) IG (J)
05130 30 IP (J)=1H
05140 GO TO 28
05150 31 IP (J)=1H1
05160 GO TO 28
05170 32 IP (J)=1H2
05180 GO TO 28
05190 33 IP (J)=1H3
05200 GO TO 28
05210 34 IP (J)=1H4

```

```

05220 G0 T0 28
05230 35 IP (J) =1H5
05240 G0 T0 28
05250 36 IP (J) =1H6
05260 G0 T0 28
05270 37 IP (J) =1H7
05280 G0 T0 28
05290 38 IP (J) =1H8
05300 G0 T0 28
05310 39 IP (J) =1H9
05320 G0 T0 28
05330 40 IP (J) =1H*
05340 28 CONTINUE
05350 WRITE (7,2003) ABOVE, (IP (J) ,J=1,53)
05360 2003 FORMAT (2X,F6.2,1H+,53A1)
05370 24 CONTINUE
05380 T=SUMXY-(SUMX*SUMY)/FLOAT(N2)
05390 R=T/SQRT((SUMX2-SUMX*SUMX/FLOAT(N2))*(SUMY2-SUMY*SUMY/FLOAT(N2)))
05400 T=R*SQRT(FLOAT(N2-2))/SQRT(1.0-R*R)
05410 WRITE (7,2000) R,N2,T
05420 2000 FORMAT(/////5X,*CORRELATION COEFFICIENT BETWEEN THESE 2 *,
05430+*SETS OF VALUES = *,F6.4/19X,*THIS GIVES A T-TEST VALUE ( *,
05440+I3,* SAMPLES) = *,F5.2)
05450
05460 RETURN
05470 END

```


SIMULATION	LOWER WL	BLOCK VL	UPPER WL	BLOCK VL
1	1.75	1.12	2.17	1.03
2	2.28	1.08	2.08	.96
3	1.95	1.06	1.59	1.69
4	2.12	.70	2.19	.69
5	1.34	1.35	1.41	.71
6	2.04	1.55	1.96	.89
7	2.29	.64	2.63	.46
8	1.28	.80	1.64	.52
9	1.69	1.63	1.77	.65
10	1.68	3.22	1.86	1.27
11	1.37	1.71	1.20	1.17
12	1.54	2.05	1.26	2.11
13	1.57	1.49	1.70	1.32
14	1.67	4.58	1.40	2.45
15	1.54	.61	1.20	.58
16	1.76	2.03	1.22	.53
17	1.41	.73	1.48	1.11
18	1.86	.92	1.88	.81
19	1.49	.96	1.41	.50
20	1.49	1.92	1.58	.57
21	1.62	.90	1.44	.62
22	1.56	1.07	1.39	1.12
23	1.77	1.39	1.43	.81
24	1.22	1.26	1.77	.69
25	3.14	4.27	2.39	2.06
26	1.49	1.52	1.96	.83
27	2.23	1.45	2.59	1.02
28	1.94	1.90	2.30	1.48
29	1.07	.68	.99	.57
30	2.29	1.73	1.59	.76
31	1.87	.71	2.18	.85
32	1.51	.91	1.88	.85
33	1.10	1.33	1.74	.64
34	1.64	4.10	2.25	.64
35	1.42	.60	1.23	.28
36	2.19	3.28	1.82	1.45
37	1.37	1.83	1.38	.45
38	1.57	1.07	1.35	.73
39	1.42	.77	1.56	1.23
40	1.93	1.44	1.78	1.61
41	2.01	1.11	2.77	2.43
42	1.66	1.03	1.79	.69
43	1.40	1.45	1.35	1.14
44	1.44	1.29	1.86	.63
45	1.36	1.26	1.44	1.14
46	1.92	1.44	1.79	1.14
47	1.34	1.44	1.70	.49
48	1.17	1.49	1.50	1.33
49	1.54	1.35	2.15	.54
50	1.65	1.75	2.06	1.31
51	1.19	.80	1.42	.72
52	1.89	1.16	2.11	.49
53	1.62	1.18	1.29	.70
54	2.09	.94	2.10	.65
55	1.98	1.46	1.92	1.48
56	1.99	.94	1.58	1.09
57	2.44	1.51	2.61	.92

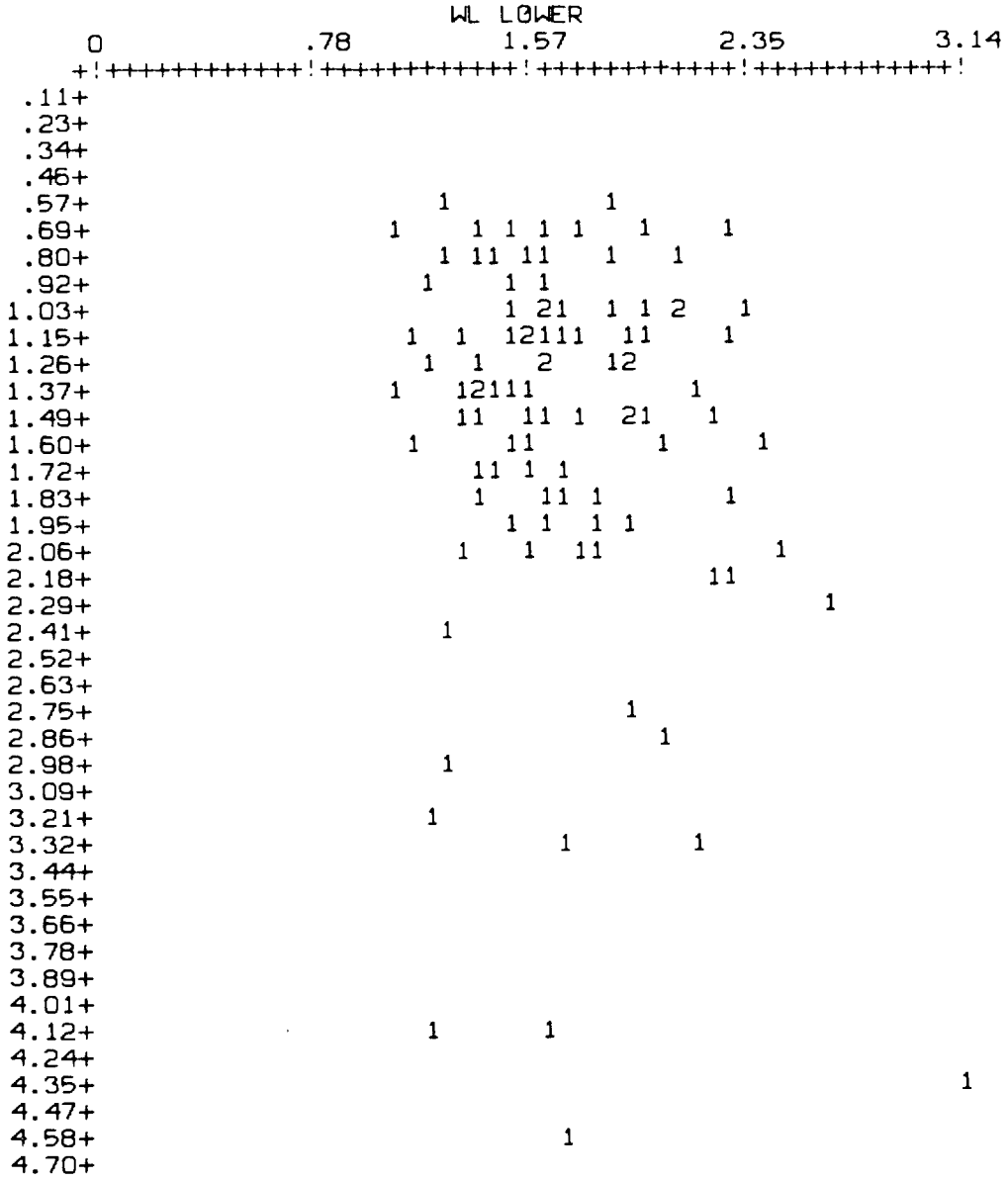
58	2.21	2.11	2.11	1.81
59	1.17	1.08	1.42	.88
60	1.64	1.46	1.79	.89
61	1.66	.98	1.97	.73
62	1.23	3.19	1.33	1.05
63	1.35	1.09	1.47	.65
64	1.28	2.87	1.75	.68
65	1.27	.53	.96	.48
66	1.83	1.86	1.88	1.40
67	1.44	1.71	1.69	1.24
68	1.28	2.38	1.78	1.06
69	1.63	1.20	1.81	.70
70	1.23	4.07	.96	.89
71	1.78	2.01	1.96	.59
72	1.62	.70	1.68	.97
73	1.54	1.67	1.03	1.00
74	2.06	2.81	2.33	.77
75	1.61	1.00	1.51	.46
76	1.51	1.32	1.25	.45
77	1.92	2.74	2.45	2.04
78	2.50	1.99	2.45	1.22
79	1.66	.62	1.51	1.17
80	1.50	1.12	1.38	.84
81	1.98	.64	1.69	.67
82	1.63	1.05	1.40	.79
83	2.11	1.02	1.58	.64
84	1.84	1.79	2.06	1.03
85	1.60	.98	1.88	.65
86	2.29	2.17	2.17	1.20
87	1.32	2.05	1.29	.68
88	1.95	1.22	2.26	.71
89	2.15	1.31	1.82	1.82
90	1.89	.57	1.86	.22
91	1.36	1.21	1.90	1.06
92	1.73	.63	2.08	.82
93	1.37	1.28	1.11	1.17
94	1.72	1.74	2.09	1.35
95	1.64	1.94	1.87	.78
96	2.67	2.27	3.07	1.11
97	1.57	.78	1.64	.64
98	1.58	1.56	1.15	1.26
99	1.93	1.17	2.54	.57
100	2.37	.98	2.85	2.57
AVERAGE	1.71	1.51	1.77	.97

! </th <th colspan="4">LOWER/UPPER</th> <th colspan="4">! 2 NON-PAY LIFTS WHEN REST OF BLOCK IS</th>	LOWER/UPPER				! 2 NON-PAY LIFTS WHEN REST OF BLOCK IS				
	! PAY !	P/P	P/N	N/P	N/N	! P/	N/	/P	/N
! .1	100	0	0	0	!	0	0	0	0
! .2	100	0	0	0	!	0	0	1	0
! .3	98	2	0	0	!	4	1	5	2
! .4	98	2	0	0	!	7	3	24	2
! .5	90	10	0	0	!	6	11	27	12
! .6	82	15	0	3	!	8	13	31	24
! .7	63	27	2	8	!	10	18	30	39
! .8	49	35	6	10	!	9	25	22	57
! .9	40	42	4	14	!	13	31	17	68
! 1.0	35	39	5	21	!	14	41	18	70
! 1.1	27	37	6	30	!	20	46	15	77

! 1.2	19	39	4	38	! 16	55	14	82
! 1.3	15	36	3	46	! 12	61	10	87
! 1.4	10	35	4	51	! 12	65	8	89
! 1.5	5	31	5	59	! 9	71	6	92
! 1.6	5	27	5	63	! 9	76	6	92
! 1.7	5	25	3	67	! 9	79	5	93
! 1.8	5	19	3	73	! 7	83	6	93
! 1.9	4	17	2	77	! 6	85	4	95
! 2.0	4	14	2	80	! 7	86	4	95

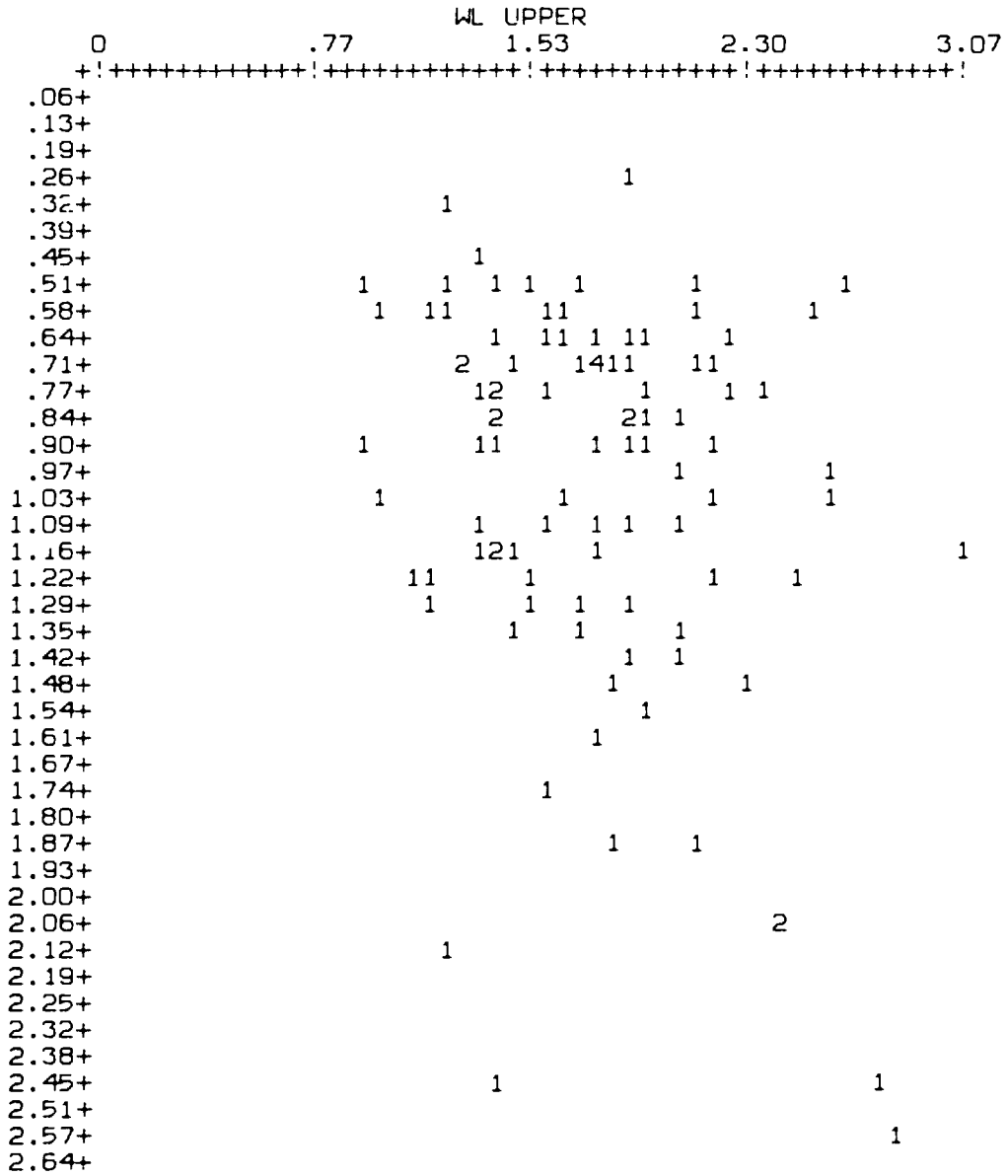
AVERAGE CORR. COEFF. BETWEEN WL AND VL OVER 100 SIMULATIONS = .0587
 AVERAGE T-TEST VALUE = 1.03

WL LOWER VS. VL LOWER



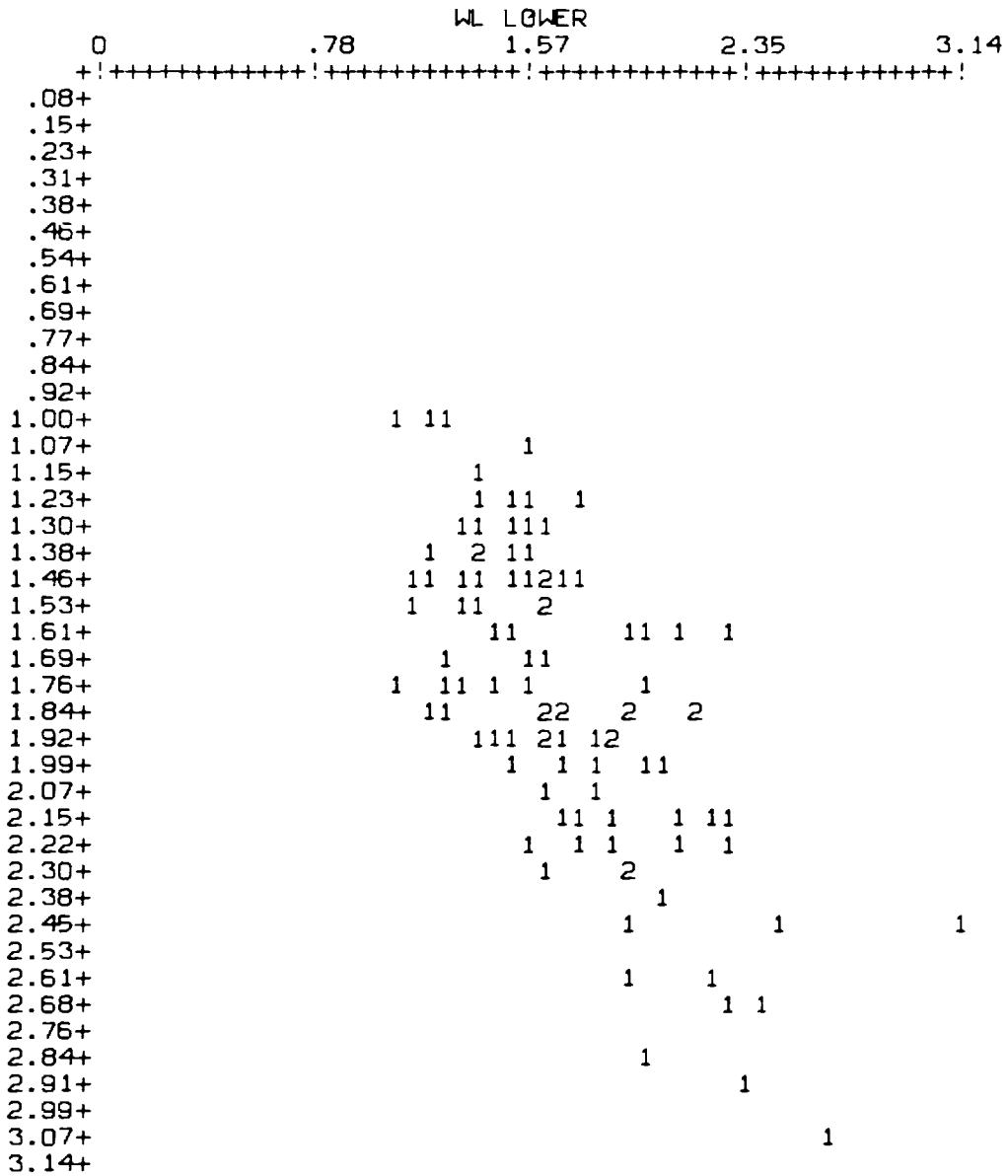
CORRELATION COEFFICIENT BETWEEN THESE 2 SETS OF VALUES = .1632
 THIS GIVES A T-TEST VALUE (100 SAMPLES) = 1.64

WL UPPER VS. VL UPPER



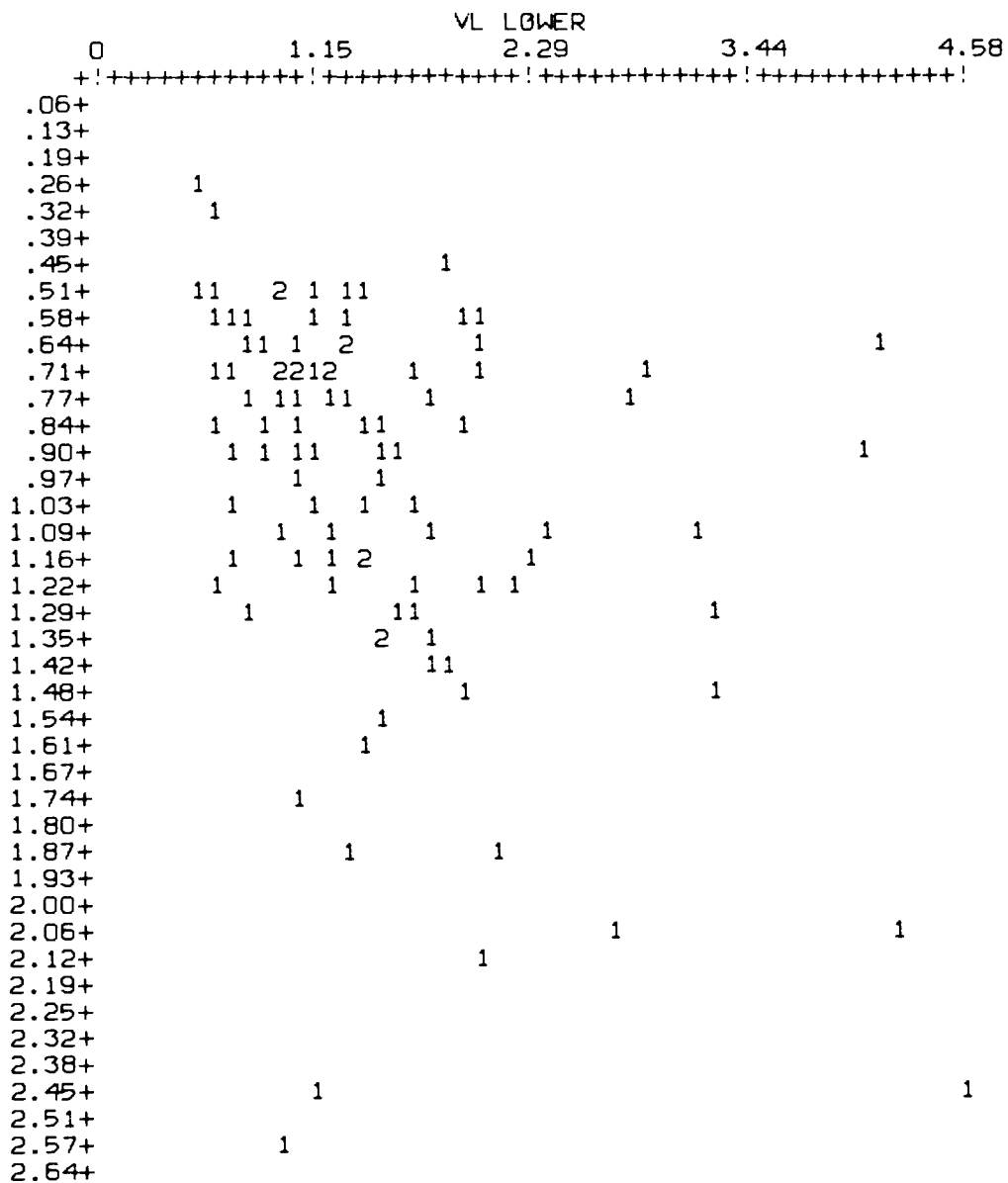
CORRELATION COEFFICIENT BETWEEN THESE 2 SETS OF VALUES = .2662
 THIS GIVES A T-TEST VALUE (100 SAMPLES) = 2.73

WL LOWER VS. WL UPPER



CORRELATION COEFFICIENT BETWEEN THESE 2 SETS OF VALUES = .6965
 THIS GIVES A T-TEST VALUE (100 SAMPLES) = 9.61

VL LOWER VS. VL UPPER



CORRELATION COEFFICIENT BETWEEN THESE 2 SETS OF VALUES = .3783
 THIS GIVES A T-TEST VALUE (100 SAMPLES) = 4.05

REFERENCES

Beringer, C. and Beringer, J.J. (1898) A text book of assaying. Griffin (London).

Blais, R.A. and Carlier, P.A. (1968) Application of geostatistics in ore valuation. Conference on ore reserve estimation and grade control, L'Estereel, 1967. Can.Inst.Min.Metall. Special Vol.9.

Bowie, S.H.U. and Darnley, A.G. and Rhodes, J.R. (1965). Portable radioisotope x-ray fluorescence analyser. Trans.Inst.Min.Metall. Vol.74 April 1965.

Brooker, P.I. (1979) Kriging. Eng.Min.Jour. Vol.180 No.9 Sept. 1979.

Burger, H. and Skala, W. (1978) Die Untersuchung ortsabhangiger Variablen: Modell, Methoden und Probleme. Geologische Rundschau Band 67 Heft 3 1978.

Clark, I. (1976) Some auxiliary functions for the spherical model of geostatistics. Computers and Geosciences Vol.1 1976.

Clark, I. (1977) SNARK - four dimensional trend surface computer program. Computers and Geosciences Vol.3 1977.

Clark, I. (1978) Geostatistical reserve estimation in Cornish tin mining. Ph.D. thesis, Univ. of London.

Clark, I. (1979) Practical geostatistics. Applied Science (London).

Clark, I. and White, B. (1976) Geostatistical modelling of an orebody as an aid to mine planning. 14th APCOM Symp., Penn. State Univ., 1976.

David, M. (1977) Geostatistical ore reserve estimation. Elsevier (New York).

David, M. and Blais, R.A. (1972) Geostatistical ore reserve estimation. 10th APCOM Symp., Johannesburg, 1972.

David, M. and Dagbert, M. and Belisle, J-M (1977) The practice of porphyry copper deposit estimation for grade and ore-waste tonnages demonstrated by several case studies. 15th APCOM Symp., Brisbane, 1977.

Delfiner, P. and Delhomme J.P. (1973) Optimum interpolation by kriging. In: J.C.Davis et al (Ed.). NATO Advanced Study Institute on Display and analysis of spatial data, Nottingham, 1973. Wiley (London).

Harbaugh, J.W. and Bonham-Carter, G. (1970) Computer simulation in geology. Wiley (London).

Harbaugh, J.W. and Merriam, D.F. (1968) Computer applications in stratigraphic analysis. Wiley (London).

Huijbregts, C.J. (1973) Regionalized variables and quantitative analysis of spatial data. In: J.C.Davis et al (Ed.). NATO Advanced Study Institute on Display and analysis of spatial data, Nottingham, 1973. Wiley (London).

Huijbregts, C.J. and Matheron, G. (1971) Universal kriging. Can.Inst.Min.Metall. Special Vol.12.

Jenkins, G.M. and Watts, D.G. (1968) Spectral analysis and its applications. Holden-Day (London).

Journel, A.G. (1974) Geostatistics for conditional simulation of orebodies. Econ.Geol. Vol.69 No.5 Aug. 1974.

Journel, A.G. (1975) Ore grade distributions and conditional simulations - two geostatistical approaches. In: M.Guarascio et al (Ed.). Advanced geostatistics in the mineral industry. Reidel (Dordrecht).

Journel, A.G. and Huijbregts, C.J. (1978) Mining geostatistics. Academic Press (London).

Krige, D.G. (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. Jour.Chem.Metall.Min.Soc.Sth.Afr. Vol.52 No.6 Dec. 1951.

Krige, D.G. (1976) Some basic considerations in the application of geostatistics to the valuation of ore in the South African gold mines. Jour.Sth.Afr.Inst.Min.Metall. Vol.76 No.9 April 1976.

Kuscevic, B. and Thomas, T.L. and Penberthy, J. (1972) Sampling distributions in South Crofty tin mine, Cornwall. Trans.Inst.Min.Metall. Vol.81 Bull.788 July 1972.

Matheron, G. (1973) The intrinsic random functions and their applications. Adv.Appl.Prob. Vol.5 Dec. 1973.

Mukherjee, N.K. (1975) Application of statistical analysis in computing ore reserves and grades utilising sample data from Rakha Mines copper deposits, Singhbhum district, Bihar. Rec.Geol.Surv.India Vol.106 No.2 1975.

Newton, M.J. (1972) Mathematical models of orebodies. 10th APCOM Symp., Johannesburg, 1982.

Oertel, G. and Walton, E.K. (1967) Lessons from a feasibility study for computer models of coal-bearing deltas. Sedimentology Vol.9 1967.

Taylor, R.G. (1966) Distribution and deposition of cassiterite at South Crofty mine, Cornwall. Trans.Inst.Min.Metall. Vol.75 Bull.711 Feb. 1966.

Whitten, E.H.T. (1973) The practical use of trend-surface analyses in the geological sciences. In: J.C.Davis et al (Ed.). NATO Advanced Study Institute on Display and analysis of spatial data, Nottingham, 1973. Wiley (London).