


This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (<https://dspace.lboro.ac.uk/>) under the following Creative Commons Licence conditions.




**CC creative commons**  
COMMONS DEED


**Attribution-NonCommercial-NoDerivs 2.5**


**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.


 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

**LOUGHBOROUGH  
UNIVERSITY OF TECHNOLOGY  
LIBRARY**

AUTHOR/FILING TITLE	
SCHAGEN, I P	
ACCESSION/COPY NO.	
104553/02	
VOL. NO.	CLASS MARK
<i>date due:-</i> <del>- 2 SEP 1982</del> LOAN 1 MTH + 2 UNLESS RECALLED <i>Date due</i> <del>- 8 SEP 1983</del> LOAN 1 MTH + 2 UNLESS RECALLED	LOAN COPY <i>date due:-</i> <del>27 FEB 1984</del> LOAN 1 MTH + 2 UNLESS RECALLED <del>- 6 JUL 1990</del> <del>- 5 JUL 1991</del> <del>3 JUL 1992</del> <del>- 2 JUL 1993</del>

010. 4553 02



THEORY AND APPLICATIONS OF MULTI-DIMENSIONAL

STATIONARY STOCHASTIC PROCESSES

by

IAN PIETER SCHAGEN, B.A., M.Sc., F.S.S.

A Doctoral Thesis

Submitted in Partial Fulfilment of the Requirements

for the Award of Doctor of Philosophy

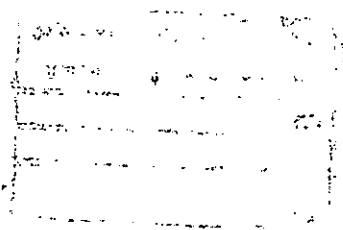
of Loughborough University of Technology

July, 1981.

© Ian Pieter Schagen, 1981.

Loughborough University	
of Technical Library	
Date	July 82
Class	
Acc. No.	104555/02

*To Sandie.*



## ACKNOWLEDGEMENTS

I wish to acknowledge the help and guidance of many people in this work. My thanks go to Professor E.M.L. Beale, who first introduced me to correlated random variables, and to Professor P. Holgate, whose advice at a later stage was invaluable. Also to the referees and editors of the Journal of the Institute of Mathematics and Its Applications and the Journal of the Royal Statistical Society, Series C, for their most helpful comments prior to publication of some of the work in this thesis.

The support and encouragement of the staff of the Computer Studies Department at Loughborough University of Technology is gratefully acknowledged. In particular, I wish to thank Professor D.J. Evans for his guidance and advice. Dr. C.J. Hinde and Dr. B. Negus are also thanked for their comments on this thesis in its draft form. Finally, my thanks and love go to my wife, Sandie, for her constant support and encouragement, and to my sons, Andrew and Paul for not hindering the work.

## DECLARATION

I declare that the following thesis is a record of research work carried out by me, and that the thesis is of my own composition. I also certify that neither this thesis nor the original work contained therein has been submitted to this or any other institution for a degree.

I.P. SCHAGEN.

ABSTRACT

The theory of stationary stochastic processes in several dimensions has been investigated to provide a general model which may be applied to various problems which involve unknown functions of several variables. In particular, when values of the function are known only at a finite set of points, treating the unknown function as a realisation of a stationary stochastic process leads to an interpolating function which reproduces the values exactly at the given points. With suitable choice of auto-correlation for the model, the interpolating function may also be shown to be continuous in all its derivatives everywhere. A few parameters only need to be found for the interpolator, and these may be estimated from the given data.

One problem tackled using such an interpolator is that of automatic contouring of functions of two variables from arbitrarily scattered data points. A "two-stage" model was developed, which incorporates a long-range "trend" component as well as a shorter-range "residual" term. This leads to a contouring algorithm which gives good results with difficult data.

The second area of application is that of optimisation, particularly of objective functions which are expensive to compute. Since the interpolator gives an estimate of the derivatives with little work, it is simple to optimise it using conventional techniques, and to re-evaluate the true function at the apparent optimum point. An iterative algorithm along these lines gives good results with test functions, especially with functions of more than two variables. A program has been developed which incorporates both the optimisation and contouring applications into a single package.



Finally, the theory of excursions of a stationary process above a fixed level has been applied to the problem of modelling the occurrence of oilfields, with special reference to their spatial distribution and tendency to cluster. An intuitively reasonable model with few parameters has been developed and applied to North Sea data, with interesting results.

# CONTENTS

	<u>PAGE</u>
<u>Chapter 1:</u> INTRODUCTION	1
<u>Chapter 2:</u> A BRIEF SURVEY OF THE THEORY	
2.1 Definitions .....	7
2.2 Properties of the Auto-correlation Function ..	10
2.3 Estimation of Values of the Process at Unknown Points .....	18
2.4 Estimation of Parameters of the Stochastic Process Model .....	22
2.5 Excursions of a Stochastic Process Above a Fixed Level .....	29
<u>Chapter 3:</u> OTHER APPROACHES TO THE SAME PROBLEMS	
3.1 The Theory of Regionalised Variables and Kriging .....	42
3.2 Optimisation of Expensive Objective Functions .....	48
3.3 Prediction of the Occurrence of Oilfields ....	52
<u>Chapter 4:</u> INTERPOLATION IN TWO DIMENSIONS AND CONTOURING	
4.1 Contouring Problems and Algorithms .....	61
4.2 Tracking Contour Lines .....	63
4.3 Long-range Trend and the Two-stage Stochastic Model .....	68
4.4 Results with Test Data .....	70
4.5 One-dimensional Applications .....	74

	<u>PAGE</u>
<u>Chapter 5:</u> APPLICATION TO THE OPTIMISATION OF FUNCTIONS OF SEVERAL VARIABLES	
5.1 General Outline .....	104
5.2 Convergence of the Optimisation Algorithm and Related Questions .....	108
5.3 Distribution of Initial Points .....	113
5.4 Anisotropic Correlation .....	116
5.5 Estimation of the Function Integral .....	120
5.6 Results with Test Functions .....	121
 <u>Chapter 6:</u> A MODEL FOR THE OCCURRENCE OF OILFIELDS	
6.1 Description of the Basic Model .....	153
6.2 Expectations of Oilfield Variables .....	154
6.3 Variable Threshold .....	158
6.4 Fitting the Model to Oilfield Data .....	159
6.5 Mean and Variance of Oil Reserves .....	162
 <u>Chapter 7:</u> CONCLUSIONS	171
 REFERENCES	173
 <u>Appendix A:</u> SIMP - A STOCHASTIC INTERPOLATION AND MODELLING PROGRAM	179
<u>Appendix B:</u> RESULTS OF SIMULATION EXPERIMENTS	228

CHAPTER 1

INTRODUCTION

The concept of a stochastic process is a very powerful one, and I hope to show that it has useful applications in several fields where existing ideas and methods are not always entirely satisfactory. One of the most familiar examples of a stochastic process is the random function of one variable. A time series may be considered to be such a random function, sampled at a discrete set of points.

A stochastic process is a probabilistic entity, and has no direct physical reality. We may say that a particular physical function of several variables is a realisation of an underlying stochastic process, but it is not identical to the stochastic process itself, which cannot be directly experienced. In the same way an ordinary random variable cannot be directly experienced, but only its realisations.

The main area of interest to which this concept has been applied is that of interpolation. If a function of several variables is known only at a number of discrete points, and values are required at other points, then some means of estimating the unknown values is required. Any such algorithm gives rise to an "interpolating function", i.e. a function of the variables which estimates the unknown true function given the known values at the data points. By the nature of the problem, such an interpolating function is virtually bound to be in error, and it is only by purest chance that it will exactly match the true function everywhere. Also by the nature of the problem, the number of possible interpolating functions for any given set of data is infinite. The question thus arises of judging which of these possible functions is best in some way.

Two criteria may be used to judge such an interpolating function. First, it should be "exact", in the sense that it should exactly reproduce the true function values at the data points. This criterion is not generally satisfied by fitting a function of given algebraic form (e.g. polynomial) to the data, unless the number of parameters to be fitted is equal to the number of data points. Second, it is usually desirable that the interpolating function be continuous in all its derivatives. This is not the case with interpolating functions of the spline type, or any of the more "ad hoc" methods used.

I aim to show that modelling the unknown function as a realisation of a multi-dimensional stochastic process leads to an interpolating function which satisfies both these criteria. Furthermore, it is simple to compute and requires the estimation of very few parameters. Of course, the simple stochastic model is not always a reasonable representation of a physical function, but it is possible to widen the scope of the basic model to embrace a large range of practical situations.

This type of stochastic interpolating function has been applied in two fields where there seemed to be a need for a better means of interpolation. The first application was in the automatic generation of contour maps, especially from scattered data points. In this case the model is of a stochastic process in only two dimensions, but there is no essential difference between this interpolating function and those in higher dimensions.

To generate contour lines efficiently from scattered data points, it is necessary not only to have a good interpolator, but also a means for keeping track of the contour lines. Conventionally, this is done by

interpolating values to the nodes of a regular mesh, and then using a standard contouring algorithm for gridded data. However, this can lead to a loss of representation of the original data, so a contouring algorithm was developed, using the stochastic interpolating function, which generates contour lines directly, without the use of any kind of grid.

The contouring algorithm has been tested on various sets of physical data. From this it became apparent that in some cases the very simple stochastic model could be extended to cover the situation where there was an underlying long-range trend plus a more quickly varying short-range component. Thus the "two-stage" model was developed, whereby the stochastic process is assumed to be the sum of two such components. Interesting and subjectively reasonable contour maps of physical data have been produced in this way.

As an application for the interpolating function in more than two dimensions, the problem of optimising objective functions of several variables was considered. Much work has been done on this subject, and many excellent algorithms exist for the efficient location of local optima, especially if the function can be differentiated. However, most such techniques can require a fair number of function evaluations to be carried out in order to reach the final optimum value.

The problem was considered from a slightly different angle: suppose we have a function of several variables which is expensive or difficult to compute and the derivative cannot be directly evaluated, but we wish to obtain a good idea of the position and value of an optimum point with a minimum number of function evaluations. How should we proceed? If we pick an arbitrary starting point and apply a conventional technique, it

may take an unacceptable number of function evaluations to reach a result, even if progress is steady.

An alternative approach to this problem is to start with a set of initial points, spread widely throughout the region within which the optimum is known (or hoped) to lie. These initial points should be positioned within the multi-dimensional "region of interest" so as to gain the maximum information about the behaviour of the function. (How this is to be achieved is a problem in itself). A stochastic interpolating function is fitted to this initial set of data, and a standard optimisation technique is used to find an optimum point for the interpolator. This is made simpler by the fact that derivatives of the interpolator are easily computed. At the interpolated optimum point, a new function evaluation is carried out, which is compared with the interpolated value. If they agree, this is taken to be a good approximation to the true optimum value. Otherwise, the interpolator is re-fitted (taking into account the new point) and the process is repeated.

In essence, the philosophy here is to make the maximum possible use of all the data collected at every stage. This will lead to more computing between function evaluations, but it is hoped that this will be offset by a reduction in the total number of function evaluations. This technique has been applied to various test functions, and the results appear to be promising.

It has been found necessary to include in the stochastic model provision for anisotropic correlation. In other words, adjacent function values may be more highly correlated in some dimensions than in others. This is a fairly trivial extension to the theory, the only problem in practice being that of estimating the anisotropy factors.



A program has been written which incorporates all these applications of stochastic processes into a single package. The program, named SIMP (Stochastic Interpolation and Modelling Program), is described in detail in Appendix A. It is designed to handle either a user-defined function of several variables, with data points generated evenly throughout the region of interest, or a set of input data values at arbitrary locations in several dimensions. With the latter type of data input contour maps may be produced in any plane defined by two of the variables, and it is also possible to plot cross-sections of the interpolating function along a line joining any two points. For the user-defined function, it is possible to produce contour maps and sections in the same way, but in addition optimisation may be carried out on the function as described earlier. As a by-product an estimate of the function integral over the region of interest is also given.

Another application of the theory of stochastic processes is in a different area of interest. The particular problem is that of modelling the occurrence of oilfields within some oil-bearing region. This has obvious practical and economic importance, especially if the region is only partially developed. Models used to date have been very "ad hoc", with many parameters defined purely intuitively. To put things on a rather better footing, it is necessary to have a model which is simple and coherent, with a small number of parameters which can be fitted to the given data.

The model suggested is based on a stationary stochastic process in two dimensions. An oilfield is assumed to be a connected region over which the stochastic process exceeds some specified value or limit. From the parameters of the stochastic process and the value of the given limit,

it is possible to calculate expected values for the area of an oilfield, as well as the number of oilfields per unit area and the reserves of an arbitrary oilfield. An extra refinement is to allow the limit value to vary slowly from point to point. This simulates the real situation in which the sizes and numbers of oilfields vary regionally.

This model has been fitted to some data for the British North Sea, and the results obtained are at least subjectively appealing. It is hoped that this type of model can produce a framework on which more reasonable estimates can be based of the reserves of partially explored oil-bearing regions.

As can be seen, the theory of stationary stochastic processes in several dimensions provides access to a set of models which can be applied to various types of problem. Interpolation of functions from finite sets of data points has been the main area of application in this work, but it is by no means the only one.

CHAPTER 2

A BRIEF SURVEY OF THE THEORY

## 2.1 DEFINITIONS

A stochastic process in  $m$  dimensions is a generalisation of the concept of a time series, or random function of one variable. The latter may be defined as follows: for each possible value  $x$  of an indicator variable within some domain, there is defined a random variable  $Z(x)$  with a given probability distribution. This probability distribution may be described by means of the probability distribution function  $F(z, x) = P[Z(x) \leq z]$ . However, knowledge of this distribution function  $F(z, x)$  for all values of  $x$  is not sufficient to define the behaviour of the stochastic process completely - it does not describe the relationships between values of the stochastic process at different points. To specify the behaviour completely, we need to define the finite-dimensional distribution function  $F(z_1, z_2, \dots, z_n, x_1, \dots, x_n)$  for any set of points  $(x_1, \dots, x_n)$ :

$$F(z_1, \dots, z_n, x_1, \dots, x_n) = P[Z(x_1) \leq z_1, \dots, Z(x_n) \leq z_n] . \quad (2.1)$$

The extension of this definition of a time series, or random function of one variable, to that of a random function of  $m$  variables  $Z(\underline{x})$ , is straightforward. The indicator variable  $\underline{x}$  is now a vector of  $m$  elements, and the finite-dimensional distribution function for a set of  $n$  points is

$$F(z_1, \dots, z_n, \underline{x}_1, \dots, \underline{x}_n) = P[Z(\underline{x}_1) \leq z_1, \dots, Z(\underline{x}_n) \leq z_n] . \quad (2.2)$$

An important concept for stochastic processes of this type is that of stationarity. The stochastic process  $Z(\underline{x})$  is said to be stationary "in the wide sense" if, for every set of  $n$  points  $(\underline{x}_1, \dots, \underline{x}_n)$  and arbitrary translation vector  $\underline{1}$ ,

$$F(z_1, \dots, z_n, \underline{x}_1, \dots, \underline{x}_n) = F(z_1, \dots, z_n, \underline{x}_1 + \underline{1}, \dots, \underline{x}_n + \underline{1}) . \quad (2.3)$$

In other words, translation through the  $m$ -dimensional domain does not change the probabilistic structure of the stochastic process.

The assumption of wide-sense stationarity has some important consequences. Firstly, equation (2.3) shows that  $F(z, \underline{x}) = F(z, \underline{x} + \underline{r})$ , and hence that the probability distribution of  $Z(\underline{x})$  is the same for every point  $\underline{x}$ . In particular, this implies that the mean

$$E[Z(\underline{x})] = \text{constant} = \mu \quad , \text{ (say) } . \quad (2.4)$$

Secondly, we may define the covariance of the stochastic process between two points  $\underline{x}_1$  and  $\underline{x}_2$ :

$$\text{Covar}[Z(\underline{x}_1), Z(\underline{x}_2)] = E[(Z(\underline{x}_1) - \mu)(Z(\underline{x}_2) - \mu)] . \quad (2.5)$$

This obviously depends on the distribution function  $F(z_1, z_2, \underline{x}_1, \underline{x}_2)$ , which is independent of absolute position (by wide-sense stationarity) and is a function only of the difference  $\underline{x}_1 - \underline{x}_2$ . Hence the covariance is a function of  $\underline{x}_1 - \underline{x}_2$  only, and we may write:

$$\text{Covar}[Z(\underline{x}_1), Z(\underline{x}_2)] = \gamma(\underline{x}_1 - \underline{x}_2) . \quad (2.6)$$

It is also clear that  $\text{Covar}[Z(\underline{x}_1), Z(\underline{x}_2)] = \text{Covar}[Z(\underline{x}_2), Z(\underline{x}_1)]$  and hence that the covariance function  $\gamma()$  is symmetrical:  $\gamma(\underline{r}) = \gamma(-\underline{r})$ .

The variance of the stochastic process,  $\sigma^2$  say, is clearly

$$\sigma^2 = E[(Z(\underline{x}) - \mu)^2] = \gamma(\underline{0}) . \quad (2.7)$$

We can normalise the covariance function by dividing by the variance to give the auto-correlation function

$$g(\underline{r}) = \gamma(\underline{r}) / \sigma^2 . \quad (2.8)$$

These properties of the stochastic process, derived from wide-sense

stationarity, can be used as an alternative stationarity criterion, that of second-order stationarity.  $Z(\underline{x})$  will be said to be stationary to second order if  $E[Z(\underline{x})]$  is a constant and the covariance between any two points  $\underline{x}_1$  and  $\underline{x}_2$  is a symmetric function of the vector  $\underline{x}_1 - \underline{x}_2$  only. Wide-sense stationarity implies but is not implied by second-order stationarity.

For general definitions of stationarity and auto-covariance and auto-correlation functions, see for example Bartlett (1966, p.174ff), Adler (1981, p.13-15, 22-25), Ripley (1981, p.9ff).

## 2.2 PROPERTIES OF THE AUTO-CORRELATION FUNCTION

Suppose that we have a stochastic process  $Z(\underline{x})$  which is stationary in the wide sense, and thus possesses an auto-correlation function  $g()$ . We may be interested in the continuity of the stochastic process  $Z(\underline{x})$ , so this concept will need to be defined. One definition is to say that  $Z(\underline{x})$  is continuous in the mean square if

$$E [(Z(\underline{x}) - Z(\underline{x+h}))^2] \rightarrow 0 \text{ as } \underline{h} \rightarrow \underline{0} . \quad (2.9)$$

Now

$$\begin{aligned} & E [(Z(\underline{x}) - Z(\underline{x+h}))^2] \\ &= E [(Z(\underline{x}))^2 + (Z(\underline{x+h}))^2 - 2Z(\underline{x})Z(\underline{x+h})] \\ &= 2(\sigma^2 + \mu^2) - 2(\gamma(\underline{h}) + \mu^2) \\ &= 2\sigma^2(1 - g(\underline{h})) . \end{aligned} \quad (2.10)$$

Thus  $Z(\underline{x})$  is mean square continuous if and only if  $g(\underline{h})$  tends to 1 as  $\underline{h}$  tends to 0. The form of the auto-correlation function will also tell us something about the differentiability or otherwise of  $Z(\underline{x})$ .

Let

$$S(\underline{x}) = \text{Lim}_{\underline{h} \rightarrow 0} \left[ \frac{Z(\underline{x+h}) - Z(\underline{x})}{\underline{h}} \right] , \quad (2.11)$$

where  $\underline{h} = |\underline{h}|$  .

Obviously  $S(\underline{x})$  is also a stochastic process, and we shall say that  $Z(\underline{x})$  is differentiable in the direction  $\underline{h}$  if  $S(\underline{x})$  has a finite variance.

$$\begin{aligned} E[S(\underline{x})] &= E \left[ \text{Lim}_{\underline{h} \rightarrow 0} \left( \frac{Z(\underline{x+h}) - Z(\underline{x})}{\underline{h}} \right) \right] \\ &= \text{Lim}_{\underline{h} \rightarrow 0} \left( \frac{E[Z(\underline{x+h})] - E[Z(\underline{x})]}{\underline{h}} \right) = 0 . \end{aligned} \quad (2.12)$$

$$\begin{aligned} \text{Var}[S(\underline{x})] &= E \left[ \text{Lim}_{\underline{h} \rightarrow 0} \left( \frac{Z(\underline{x+h}) - Z(\underline{x})}{\underline{h}} \right)^2 \right] \\ &= \text{Lim}_{\underline{h} \rightarrow 0} \left( \frac{E [(Z(\underline{x+h}))^2 + (Z(\underline{x}))^2 - 2Z(\underline{x+h})Z(\underline{x})]}{\underline{h}^2} \right) \end{aligned}$$

$$\begin{aligned}
&= \lim_{h \rightarrow 0} \left( \frac{2\sigma^2(1 - g(h))}{h^2} \right) \\
&= -\sigma^2 g''(\underline{0}) \quad . \quad (2.13)
\end{aligned}$$

Therefore,  $Z(\underline{x})$  is differentiable if the auto-correlation function is twice differentiable at the origin. By a similar argument, we may derive the covariance of the derivative process  $S(\underline{x})$ .

$$\begin{aligned}
&\text{Covar}[S(\underline{x}), S(\underline{x+r})] \\
&= E[S(\underline{x})S(\underline{x+r})] \\
&= E \left[ \lim_{h \rightarrow 0} \frac{(Z(\underline{x+h}) - Z(\underline{x}))(Z(\underline{x+r+h}) - Z(\underline{x+r}))}{h^2} \right] \\
&= \lim_{h \rightarrow 0} \frac{1}{h^2} [2\gamma(\underline{r}) - \gamma(\underline{r+h}) - \gamma(\underline{r-h})] \\
&= -\sigma^2 g''(\underline{r}) \quad (2.14)
\end{aligned}$$

(See Bartlett, 1966, p.1809, Adler, 1981, p.25-27).

We can therefore use the properties of the auto-correlation function to classify stationary stochastic processes into one of three types:

- a) Non-continuous.  $g(\underline{r})$  does not tend to 1 as  $\underline{r} \rightarrow \underline{0}$ . This implies that the stochastic process contains an element which is totally random, or "white noise". (This is called "nugget effect" in the terminology of regionalised variable theory - see Chapter 3).
- b) Continuous, but not differentiable.  $g(\underline{0})$  is continuous, but is not twice differentiable. This implies that the stochastic process is fairly regular, but by no means smooth.
- c) Continuous and differentiable.  $g(\underline{r})$  is twice differentiable at the origin, which because of symmetry implies that  $g'(\underline{0}) = 0$ .  $z(\underline{x})$  is a smoothly varying function.



Matheron (1971, p.57-58) describes these three classes of stochastic process with relation to their variograms rather than covariance functions.

Generally speaking we shall try, if necessary by suitable scaling, to ensure that the auto-correlation function is an isotropic function of distance only, That is  $g(\underline{r}) = g(r)$  where  $r = |\underline{r}|$ . In Chapter 5 scaling factors will be introduced to model the case of anisotropic correlations. The functional form of the auto-correlation has a controlling influence on the behaviour of the stochastic process, as seen above, but it is important to note that not all arbitrary forms of function are allowable.

For an arbitrary function  $g()$  with  $g(0)=1$  to be an allowable auto-correlation function it must be positive semi-definite. That is to say, given a set of  $n$  points  $\underline{x}_1, \dots, \underline{x}_n$  and arbitrary multipliers  $\lambda_1, \dots, \lambda_n$ , then

$$\sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j g(\underline{x}_i - \underline{x}_j) \geq 0 . \quad (2.15)$$

The reason for this is simple to see - suppose that it were not so, and it was possible to find a set of  $\underline{x}_i$ 's and  $\lambda_i$ 's such that

$$\sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j g(\underline{x}_i - \underline{x}_j) < 0 .$$

Consider the random variable

$$\begin{aligned} X &= \sum_{i=1}^n \lambda_i (Z(\underline{x}_i) - \mu) . \\ \text{Var}(X) &= \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j E[(Z(\underline{x}_i) - \mu)(Z(\underline{x}_j) - \mu)] \\ &= \sigma^2 \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j g(\underline{x}_i - \underline{x}_j) < 0 . \end{aligned} \quad (2.16)$$

Since it is impossible for the variance of a random variable to be negative, it is clear that the requirement of being positive

semi-definite is necessary for  $g()$ . Another way of interpreting this constraint on the possible forms of  $g()$  is via the Fourier transform.

At this point we need to apply the "ergodic theorem" of stationary stochastic processes (see for example Yaglom 1962). This states that the mathematical expectation of functions of the process  $Z(\underline{x})$  may be replaced by the limit of the average of the function over a large region  $\Omega$ .

If we replace the covariance function  $\gamma(\underline{r})$  by its estimate over a large region  $\Omega$ :

$$\hat{\gamma}(\underline{r}) = \frac{1}{|\Omega|} \int_{\Omega} (Z(\underline{x}) - \mu) (Z(\underline{x} - \underline{r}) - \mu) d\underline{x} ,$$

then as  $\Omega$  tends to  $\infty$ ,  $\hat{\gamma}(\underline{r})$  tends to  $\gamma(\underline{r})$ . If we consider the Fourier transform of  $\hat{\gamma}()$ :

$$\begin{aligned} \hat{G}(\underline{\omega}) &= \int_{\Omega} \left( \frac{1}{|\Omega|} \int_{\Omega} (Z(\underline{x}) - \mu) (Z(\underline{x} - \underline{r}) - \mu) d\underline{x} \right) e^{i\underline{\omega} \cdot \underline{r}} d\underline{r} \\ &= \frac{1}{|\Omega|} \iint (Z(\underline{x}) - \mu) e^{i\underline{\omega} \cdot \underline{x}} \cdot (Z(\underline{x} - \underline{r}) - \mu) e^{i\underline{\omega} \cdot (\underline{r} - \underline{x})} d\underline{x} d\underline{r} \\ &= \frac{1}{|\Omega|} \int (Z(\underline{x}) - \mu) e^{i\underline{\omega} \cdot \underline{x}} d\underline{x} \int (Z(\underline{y}) - \mu) e^{i\underline{\omega} \cdot \underline{y}} d\underline{y} \\ &= \frac{1}{|\Omega|} \phi(\underline{\omega}) \cdot \phi(\underline{\omega}) , \end{aligned} \tag{2.17}$$

where  $\phi()$  is the Fourier transform of  $Z(\underline{x}) - \mu$ .

Therefore

$$\hat{G}(\underline{\omega}) = \frac{1}{|\Omega|} |\phi(\underline{\omega})|^2 \geq 0 . \tag{2.18}$$

Thus the Fourier transform of the covariance (and hence the auto-correlation) function must be greater than or equal to 0 for all values of  $\underline{\omega}$ . Furthermore, by considering the inverse transform we can show that the Fourier transform must have a finite integral.

$$\gamma(\underline{r}) \propto \int_W G(\underline{\omega}) e^{-i\underline{\omega} \cdot \underline{r}} d\underline{\omega} , \tag{2.19}$$

( $W$  is the  $m$ -dimensional region over which  $\underline{\omega}$  is defined).

$$\begin{aligned} \text{Therefore} \quad \gamma(\underline{0}) &= \text{Var}(Z(\underline{x})) \\ &= \int_W G(\underline{\omega}) \, d\underline{\omega} . \end{aligned} \quad (2.20)$$

Thus the integral of the Fourier transform must be finite if  $Z(\underline{x})$  has a finite variance. See Matheron (1971, p.13-14), Bartlett (1966, p.175-176), Ripley (1981, p.10-11).

With these principles in mind, we can consider various possible forms of the auto-correlation function  $g()$ , assumed to be isotropic.

- a)  $g(r) = e^{-\alpha r}$ . Switzer (1965) has shown that a stochastic process can be defined in any number of dimensions with this auto-correlation function. The main disadvantage of the function is that it is not differentiable at the origin, and hence the stochastic process, although continuous, is not differentiable.
- b)  $g(r) = e^{-\alpha r} \cos \beta r$ . This form of auto-correlation function is used by Shvidler (1964) in two dimensions. Unfortunately, it is not positive semi-definite in two dimensions and can therefore give rise to negative variances. Its use is to be avoided.
- c)  $g(r) = e^{-\alpha r^2}$ . This "Gaussian" form of the auto-correlation function has a similar form for its Fourier transform in any number of dimensions. It is positive semi-definite for any number of dimensions. Furthermore it is differentiable at  $r=0$  and is therefore associated with a continuous and differentiable stochastic process. This is the form of auto-correlation function which will generally be used for the rest of this work, usually in the form:

$$g(r) = \exp(-r^2/2\rho^2) ,$$

where  $\rho$  will be known as the "correlation distance".

- d) A stochastic process can be created which is of "moving average" type with a certain form in the following way:

Let  $\epsilon(\underline{x})$  be a totally uncorrelated random white noise process of mean zero, and use a weighting function of arbitrary form,  $q()$  say, such that

$$Z(\underline{x}) = \mu + \int q(\underline{x}-\underline{u}) \epsilon(\underline{u}) d\underline{u} . \quad (2.21)$$

Then  $Z(\underline{x})$  is a well-defined stochastic process and will have a positive semi-definite auto-correlation function, whose form is given by:

$$g(\underline{r}) = \int q(\underline{u}) q(\underline{u}+\underline{r}) d\underline{u} / \int q^2(\underline{u}) d\underline{u} . \quad (2.22)$$

Thus by choice of the form of  $q()$ , we may generate a wide range of forms for  $g()$ . For example in two dimensions, for the case of an isotropic weighting function  $q(x,y)$ , we could write

$$g(\underline{r}) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} q(u-r/2, v) q(u+r/2, v) dudv}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} q^2(u, v) dudv} . \quad (2.23)$$

Two such weighting functions have been used to generate appropriate two-dimensional auto-correlation functions:

1. If  $q(x,y) = \exp(-r^2/2)$  with  $r^2 = x^2 + y^2$ ,

$$\begin{aligned} \text{then } g(\underline{r}) &= \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp[-(u^2 - ru + r^2/4 + v^2 + u^2 + ru + r^2/4 + v^2)/2] dudv}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp[-(u^2 + v^2)] dudv} \\ &= \exp(-r^2/4) . \end{aligned}$$

2. If  $q(x,y) = \exp(-\alpha r^2) \cos(\beta r^2)$ , then we can show that

$$g(\underline{r}) = e^{-t} \left[ \frac{\cos(\gamma t) - \sin(\gamma t)}{2 + \gamma^2} + \frac{(1 + \gamma^2)}{2 + \gamma^2} e^{-\gamma^2 t} \right] , \quad (2.24)$$

where  $t = \frac{\alpha r^2}{2}$  and  $\gamma = \beta/\alpha$  .

Thus the first example of such a "moving average" process gives rise to an auto-correlation function which is of Gaussian form (type (c) above), while the second function, for  $\gamma > 0$ , gives rise to negative values of  $g(r)$  for certain values of  $r$ . This may be useful in modelling practical examples where this type of negative correlation occurs, since function (b) above cannot be used.

Figures 2.1 to 2.3 illustrate some of these forms of auto-correlation function.

This list by no means exhausts all the possibilities for auto-correlation functions which have been suggested in the literature. For example, several authors (Whittle, 1954, p.448; Matérn, 1960, p.56; Ripley, 1981, p.56) have suggested that the "most natural" form of auto-correlation function in two dimensions is given by

$$g(r) = \frac{r}{r_0} K_1\left(\frac{r}{r_0}\right)$$

where  $K_1(\ )$  is the modified Bessel function of the second kind, order 1.

Another popular model is the "spherical" auto-correlation (see David, 1977, p.102; Journel & Huijbregts, 1978, p.163-164; Ripley, 1981, p.56) which in three dimensions is represented as

$$\begin{aligned} g(r) &= 1 - \frac{3r}{2r_0} + \frac{r^3}{2r_0^3}, & r \leq r_0 \\ &= 0, & r > r_0. \end{aligned}$$

However, in the rest of this work I shall assume the simple form of auto-correlation function given as (c) above. It is computationally simple, is continuous at the origin, and requires the estimation of only a single parameter to be fitted to real data. In practice, we shall mainly

be dealing with data which is sparse and widely scattered, so that it will not be possible to differentiate precisely between the various possible forms of auto-correlation function. See Ripley, 1981, p.58-64 for an example where the data does not give clear guidance as to the form of function to be used.

### 2.3 ESTIMATION OF VALUES OF THE PROCESS AT UNKNOWN POINTS

If the stationary stochastic process  $Z(\underline{x})$  has known mean  $\mu$  and auto-correlation function  $g()$ , then we may use the structure of the process to estimate values at unknown points, given a set of known values. Suppose a realisation of the process has measured values  $z_1, \dots, z_n$  at  $n$  points  $\underline{x}_1, \dots, \underline{x}_n$ , and it is required to estimate a value at the point  $\underline{x}_{n+1}$ . We shall consider linear estimators of the form

$$\begin{aligned}\hat{Z}(\underline{x}_{n+1}) &= \mu + \sum_{i=1}^n b_i \cdot (z_i - \mu) \\ &= \mu + \underline{b}'(\underline{z} - \underline{\mu})\end{aligned}\quad (2.25)$$

A least-squares error criterion will be used to find optimal values of  $\underline{b}$ . If  $Z(\underline{x})$  has a Normal probability distribution, then this criterion will lead to a best linear unbiased estimator for  $Z(\underline{x}_{n+1})$ .

$$\begin{aligned}\text{Let } H &= E[(Z(\underline{x}_{n+1}) - \hat{Z}(\underline{x}_{n+1}))^2] \\ &= E[(Z(\underline{x}_{n+1}) - \mu - \underline{b}'(\underline{z} - \underline{\mu}))^2] \\ &= \sigma^2 - 2\underline{b}'\underline{c} + \underline{b}'\underline{S}\underline{b}\end{aligned}\quad (2.26)$$

where  $\underline{c}$  is a vector of covariances

$$\begin{aligned}c_i &= \text{Covar}(Z(\underline{x}_i), Z(\underline{x}_{n+1})) \\ &= \sigma^2 g(\underline{x}_i - \underline{x}_{n+1})\end{aligned}$$

and  $\underline{S}$  is the covariance matrix for the  $n$  known points

$$\begin{aligned}S_{ij} &= \text{Covar}(Z(\underline{x}_i), Z(\underline{x}_j)) \\ &= \sigma^2 g(\underline{x}_i - \underline{x}_j)\end{aligned}$$

To minimise  $H$ , set

$$\underline{b} = \underline{S}^{-1}\underline{c}\quad (2.27)$$

$$\begin{aligned} \text{i.e.} \quad Z(\underline{x}_{-n+1}) &= \mu + \underline{c}' S^{-1} (\underline{z} - \underline{\mu}) \\ &= \mu + \underline{c}' \underline{\gamma} \end{aligned} \quad (2.28)$$

$$\text{where} \quad \underline{\gamma} = S^{-1} (\underline{z} - \underline{\mu}) .$$

The vector  $\underline{\gamma}$  depends only on the values at and correlations between the  $n$  original points. It may be considered to be a vector of uncorrelated values derived from the initial data values corrected for their known correlations. The estimation of an unknown value therefore consists of multiplying this constant vector by the vector of correlations between unknown point and known points, and adding the mean.

The variance in the estimate of  $Z(\underline{x}_{-n+1})$  is equal to the value of  $H$  when  $\underline{b}$  is set equal to  $S^{-1} \underline{c}$ , i.e.

$$\text{Var}[\hat{Z}(\underline{x}_{-n+1})] = \sigma^2 - \underline{c}' S^{-1} \underline{c} . \quad (2.29)$$

We may estimate the value of the stochastic process  $Z(\underline{x})$  at any arbitrary point in this way, and hence we may define an "interpolation function"

$$\begin{aligned} f(\underline{x}) &= E[Z(\underline{x}) | Z(\underline{x}_1) = z_1, \dots, Z(\underline{x}_n) = z_n] \\ &= \mu + \underline{c}'(\underline{x}) \cdot \underline{\gamma} \end{aligned} \quad (2.30)$$

where

$$\underline{c}_i(\underline{x}) = g(\underline{x} - \underline{x}_i) .$$

(see Ripley, 1981, p.44-47; Whittle, 1963, p.46-47).

If  $Z(\underline{x})$  is continuous and differentiable in the mean square, then  $f(\underline{x})$  will also be continuous and differentiable. From the nature of the estimation we know that  $f(\underline{x}_i) = z_i$  for all the known points  $i=1, \dots, n$ .

The interpolating function  $f(\underline{x})$  thus possesses two useful properties:

1. It is continuous and differentiable everywhere.
2. It passes exactly through the given data points.



The first point can be seen easily from the form of  $f()$ , rewritten as:

$$f(\underline{x}) = \mu + \sum_{i=1}^n \gamma_i g(\underline{x} - \underline{x}_i) . \quad (2.31)$$

This is a linear sum of auto-correlation function terms. If the auto-correlation function is continuous in all its derivatives at all points (including  $r=0$ ) then  $f()$  will also be so. An example of an auto-correlation function with this property is

$$g(r) = \exp(-r^2/2\rho^2) , \quad (2.32)$$

which is the form which will be used in most practical applications of this work.

To see the second point from the definition of  $f()$ , suppose that  $\underline{x} = \underline{x}_i$  say. From the definition of the vector  $\underline{\gamma}$ ,

$$S\underline{\gamma} = \underline{z} - \underline{\mu} . \quad (2.33)$$

The  $i^{\text{th}}$  row of this set of equations can be written as

$$\underline{c}'(\underline{x}_i) \cdot \underline{\gamma} = z_i - \mu . \quad (2.34)$$

This implies that

$$f(\underline{x}_i) = z_i ,$$

and hence the function  $f()$  is an exact interpolator.

In addition to the function itself, the derivatives of  $f()$  in any dimension  $k$  can be computed

$$\frac{\partial}{\partial x_k} f(\underline{x}) = \sum_{i=1}^n \gamma_i \frac{\partial}{\partial x_k} g(\underline{x} - \underline{x}_i) . \quad (2.35)$$

Similarly the integral of  $f()$  may be computed:

$$\int_{\mathbf{R}} f(\underline{x}) d\underline{x} = \mu \int_{\mathbf{R}} d\underline{x} + \sum_{i=1}^n \gamma_i \int_{\mathbf{R}} g(\underline{x}-\underline{x}_i) d\underline{x} . \quad (2.36)$$

Both these results will be used later.

The assumption that  $Z(\underline{x})$  is a Normally distributed random variable is one that will be made, implicitly or explicitly, throughout the remainder of this work. As mentioned previously, with this assumption the interpolating function (2.31) gives a best linear unbiased estimator for the value at an unknown point. However, it is possible to study the behaviour of non-Normal random fields - Adler (1981, p.168ff) deals with a  $\chi^2$  field, built as a sum of squares of several independent Normal random fields. Ripley (1981, p.73) advises that the best way of predicting a non-Normal process would be to find a transformation to Normality and predict the transformed process, and shows that a common example of this procedure is when  $Z(\underline{x})$  is assumed to be log-Normally distributed.

Thus the Normal random field is both the simplest to handle and the basis from which we may tackle other forms of stationary stochastic process.

## 2.4 ESTIMATION OF THE PARAMETERS OF THE STOCHASTIC PROCESS MODEL

Before being able to use the stochastic interpolating function  $f()$  defined in the previous section, it is first necessary to estimate values of the parameters of the stochastic process of which the  $n$  data values are assumed to form a realisation. In other words, we need to fit the stochastic process model to the data. One of these model parameters is the "grand mean"  $\mu$ , and we also need at least one other parameter to describe the auto-correlation function. We shall assume for the rest of the current work that the auto-correlation function takes the form

$$g(r) = \exp(-r^2/2\rho^2) , \quad (2.37)$$

where  $\rho$  is the "correlation distance", and corresponds to an extra model parameter to be estimated.

The two parameters  $\mu$  and  $\rho$  control the general form of the interpolating function which is fitted to the data. The grand mean  $\mu$  can be considered to be the value to which the function tends as it moves away from regions of known data. In other words, it is the "best guess" at the function value when no other information is available, or the value to which the stochastic interpolating function will extrapolate. The correlation distance  $\rho$  is the distance over which the correlation between two points is strong. These two parameters interact in an interesting fashion.

As  $\rho$  tends to zero,  $f()$  tends to become equal to  $\mu$  everywhere, except at the measured data points where there are narrow "bumps" in the function which make  $f(\underline{x}_i)$  equal to  $z_i$ . As  $\rho$  tends to infinity, the value of  $\mu$  becomes of less and less importance to the interpolation, which depends heavily upon the data values. Figure 2.4 illustrates these properties of  $\mu$  and  $\rho$ .

These two important parameters may be selected on the basis of some subjective criterion to produce an acceptable interpolator. Alternatively, the parameters which best fit the data may be estimated using maximum likelihood principles, making the assumption that the underlying stochastic process is normally distributed. This can be done in two phases - estimating  $\mu$  for a given  $\rho$  value and then estimating  $\rho$  for a given  $\mu$  value. Best values of both parameters are speedily obtained after a few iterations.

a) Estimation of  $\mu$  given  $\rho$

Let us assume that the  $n$  random variables  $Z(x_1), \dots, Z(x_n)$  are distributed with a multivariate normal distribution with probability density

$$(2\pi)^{-n/2} |S|^{-1/2} \exp[-\frac{1}{2}(\underline{z} - \underline{\mu})' S^{-1} (\underline{z} - \underline{\mu})] \quad (2.38)$$

where  $S$  is the covariance matrix, depending on  $\rho$ ,

$$\underline{z} = (z_1, \dots, z_n)$$

and  $\underline{\mu} = \mu \underline{1}$ .

We may consider this to be the likelihood  $L(\mu)$  given the data  $z_1, \dots, z_n$ , and wish to choose  $\mu$  so as to maximise  $L(\mu)$ . First take logs:

$$\log L(\mu) = -\frac{1}{2}n \log 2\pi - \frac{1}{2} \log |S| - \frac{1}{2} (\underline{z} - \underline{\mu})' S^{-1} (\underline{z} - \underline{\mu}) \quad (2.39)$$

Hence

$$\frac{\partial}{\partial \mu} \log L(\mu) = \underline{1}' S^{-1} \underline{z} + \underline{z}' S^{-1} \underline{1} - 2\mu \underline{1}' S^{-1} \underline{1} = 0, \quad (2.40)$$

i.e.  $\mu = (\underline{1}' S^{-1} \underline{z} + \underline{z}' S^{-1} \underline{1}) / 2 \underline{1}' S^{-1} \underline{1}$ . (2.41)

Therefore the maximum likelihood estimate of  $\mu$  for a certain value of  $\rho$  is given by

$$\hat{\mu} = \frac{\sum_i \left( \sum_j S_{ij}^{-1} \right) z_i}{\sum_i \sum_j S_{ij}^{-1}} \quad (2.42)$$

b) Estimation of  $\rho$  given  $\mu$ 

Let

$$\varepsilon_i = Z(x_i) - \hat{Z}_i, \quad i=1, \dots, n, \quad (2.43)$$

where  $Z(x_i)$  is the value of the stochastic process at the  $i^{\text{th}}$  data point and  $\hat{Z}_i$  is the estimated value at that point, based on the previous  $i-1$  values  $z_1, \dots, z_n$ .  $\varepsilon_i$  is a normally distributed random variable of mean 0 and variance  $\sigma_i^2$ , the residual error variance (see equation (2.29)). If we form the series of values

$$e_i = \frac{\varepsilon_i}{\sigma_i}, \quad i=1, \dots, n, \quad (2.44)$$

this will produce a set of  $n$  independent standard normal random variables, each with mean 0 and variance 1. The log likelihood of the assumed covariance matrix  $S$  used to generate these is proportional to

$$e^{-\frac{1}{2} \sum_{i=1}^n e_i^2} = -\frac{1}{2} \sum_{i=1}^n \varepsilon_i^2 / \sigma_i^2. \quad (2.45)$$

We therefore need to search for a value of  $\rho$  which minimises the sum

$$H = \sum_{i=1}^n (z_i - \hat{Z}_i)^2 / \sigma_i^2. \quad (2.46)$$

The terms in this expression may be most easily generated by the operation of pivoting, as suggested by Beale (1970). Pivoting on the  $q^{\text{th}}$  diagonal element of the matrix  $S$  to form a new matrix  $S^*$  is carried out as follows:

$$\begin{aligned} S_{qq}^* &= -1/S_{qq} \\ S_{qk}^* &= S_{kq}^* = -S_{qk}/S_{qq} \\ S_{jk}^* &= S_{kj}^* = S_{jk} - S_{jq} S_{qk} / S_{qq} \quad (j \neq q, k \neq q). \end{aligned} \quad (2.47)$$

If we perform this operation using the first  $i-1$  diagonal elements of  $S$  in turn, with a new matrix  $S^{**}$  as the result, then we may write the following:

$$\hat{z}_i = \mu - \sum_{j=1}^{i-1} S_{ij}^{**} (z_j - \mu)$$

$$\sigma_i^2 = S_{ii}^{**} . \quad (2.48)$$

These two estimation procedures, used together, give good estimates of the parameters  $\mu$  and  $\rho$ . However, if the number of data points is large then the process of estimation may be slow and time-consuming. This is because of the necessity for computing the  $n$  by  $n$  correlation matrix  $S$  and pivoting on it.

An alternative way of computing the  $\underline{e}$  values for the above estimation procedure is to make use of the Cholesky decomposition. Ripley (1981, p.17) shows that it is always possible to find a lower triangular matrix  $L$  such that  $LL'=S$ . Then  $\underline{e}=L^{-1}\underline{z}$ , and the computation of  $L^{-1}$  is simple. Chambers (1977, p.102-107) describes the computational details of the Cholesky decomposition to produce  $L$  from  $S$ .

For reasonably large  $n$  it may be more practical to substitute estimation procedures which, although not so rigorous as those already described, produce acceptable results with less computation. If  $n$  is large it will probably be reasonable to estimate  $\mu$  by the arithmetic mean or the median of the data values. (The median may well be preferable as being a more robust estimator and less influenced by extreme values). If the data points are fairly evenly spread, a good estimate of  $\mu$  can be obtained with little computation.

One approach to estimating  $\rho$  for large numbers of data points is to consider the points in pairs, and to estimate the value of  $\rho$  by means of each point and its nearest neighbour. Suppose that two such points are a distance  $r$  apart and let  $x = \exp(-r^2/2\rho^2)$ .

The covariance matrix for just two points

$$S = \begin{pmatrix} \sigma^2 & \sigma^2 x \\ \sigma^2 x & \sigma^2 \end{pmatrix} \quad (2.49)$$

The log likelihood function for 'x' derived from the bivariate Normal distribution is

$$\log L = -\log(2\pi) - \frac{1}{2} \log |S| - \frac{1}{2} \underline{z}' S^{-1} \underline{z} \quad (2.50)$$

where

$$\underline{z} = \begin{pmatrix} z_1 - \mu \\ z_2 - \mu \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad \text{say.}$$

To find a maximum likelihood estimator for x set

$$\frac{d}{dx} \log L = \frac{1}{2|S|} \frac{d|S|}{dx} - \frac{1}{2} \frac{d}{dx} (\underline{z}' S^{-1} \underline{z}) = 0$$

$$\text{Therefore} \quad \frac{d|S|}{dx} = -|S| \frac{d}{dx} (\underline{z}' S^{-1} \underline{z}) \quad (2.51)$$

Now

$$\frac{d|S|}{dx} = -2\sigma^4 x$$

$$\text{and} \quad \frac{d}{dx} (\underline{z}' S^{-1} \underline{z}) = -\frac{2\sigma^2 y_1 y_2}{|S|} - \frac{\sigma^2 (y_1^2 + y_2^2 - 2xy_1 y_2)}{|S|^2} \frac{d|S|}{dx} \quad (2.52)$$

which finally leads to

$$x^3 - \frac{y_1 y_2}{\sigma^2} x^2 + \left( \frac{y_1^2 + y_2^2}{\sigma^2} - 1 \right) x - \frac{y_1 y_2}{\sigma^2} = 0 \quad (2.53)$$

If a solution  $x^*$  can be found for this cubic equation between the values 0 and 1, then the maximum likelihood estimator of  $\rho$  derived from

the two given points is

$$\hat{\rho} = r \sqrt{\frac{1}{-2 \log x^*}} \quad (2.54)$$

This procedure may be repeated for  $n$  pairs of neighbouring points and the set of estimates so produced averaged in some way, for example by use of the median of the individual estimated values for different sets of data.

In an attempt to evaluate the effectiveness of both these techniques for estimating  $\rho$ , some simulation experiments have been carried out. Full details are summarised here. Each experiment consisted of generating  $N$  data points, randomly positioned in two dimensions inside a square region of extent  $10 \times 10$ . Values of a Normal process with correlation distance  $\rho$  were generated, and estimates  $\rho$  of this parameter were computed using both methods ("Maximum likelihood" and "pair-point"). 10 such experiments were performed for each value of  $N$  and  $\rho$ . Values of  $N$  were 10, 20 and 40, and  $\rho$  took the values 1.0, 2.0 and 4.0.

Figure 2.5 illustrates the results of these experiments, by plotting both the average value of  $\hat{\rho}$  and the estimated mean square error for each combination of  $N$  and  $\rho$ , for each of the two methods. It can be seen that for  $\rho=1.0$  and 2.0 the maximum likelihood estimator is better than the pair-point for  $N=10$ , but both appear to converge towards the true  $\rho$  value for  $N=20$  and 40. However, both methods severely underestimate when  $\rho=4.0$ . It seems to be quite difficult to estimate  $\rho$  accurately when it is of the order of the dimensions of the region of interest. For smaller values, the pair-point methods seems to be equally as effective as the maximum



likelihood technique for reasonable numbers of data points.

Another possible approach to estimating covariances from sets of data points is described in an appendix to Nelder & Mead (1964). Here in  $n$  dimensions, a simplex of  $n+1$  points is used (together with the "half-way" points between them) to give an estimate of  $B$ , the information matrix, from which the covariance matrix can be obtained. This can be seen to have some similarities to the pair-point method, except that in say two dimensions a number of triangles would be evaluated.

## 2.5 EXCURSIONS OF A STOCHASTIC PROCESS ABOVE A FIXED LEVEL

One aspect of the theory of stochastic processes which has interesting applications is that of excursions above a fixed level. It is probably best to study the one-dimensional case first and then see how it can be generalised. (See Cramer & Leadbetter, 1967, p.190-218).

Let  $Z(t)$  be a stationary stochastic process, with probability density function  $f()$  at any point. Consider the fixed level  $Z(t)=u$ , and define an "excursion" as an interval  $[t_1, t_2]$  for which

$$Z(t_1) = Z(t_2) = u$$

and  $Z(t) > u$  for  $t_1 < t < t_2$  , (2.55)

(see Figure 2.6).

Various properties of these excursions may be investigated. For example, the probability distribution of the length  $L$  (where  $L=t_2-t_1$ ) may be investigated. Alternatively, the probability distribution of  $N$ , the rate of occurrence of excursions per unit interval in  $t$ , may be computed.

Each excursion is bounded by one "upcrossing" and one "downcrossing" of the process  $Z(t)$  relative to the level  $u$ . An upcrossing (such as  $t_1$ ) is a point where  $Z(t)=u$  and  $Z'(t)>0$ . A downcrossing (such as  $t_2$ ) is a point where  $Z(t)=u$  and  $Z'(t)<0$ . Let us assume that  $Z(t)$  has continuous derivatives up to at least second order (so that it is continuous and differentiable from section 2.2), and define the reverse gradient  $\omega$  by

$$\omega(t) = -Z'(t) .$$

Let  $f_\omega()$  be the probability density function for  $\omega$ ,

$$R_\omega(x) = P[\omega > x] = \int_x^\infty f_\omega(x) dy ,$$

$c(z)$  be the probability density function of  $Z(t)$ , conditional on  $Z(t) \geq u$   
 $= f(z)/R(z)$  .

Consider  $P[\text{Downcrossing in } (t, t+\delta t) | Z(t) \geq u]$

$$\begin{aligned} &= \int_0^{\infty} c(u+\delta) P[\omega \geq \delta/\delta t] d\delta \\ &= \int_0^{\infty} c(u+\delta) R_{\omega}(\delta/\delta t) d\delta \end{aligned} \quad (2.56)$$

(see Figure 2.7).

Let  $y = \delta/\delta t$ , whereupon the required probability becomes

$$\delta t \int_0^{\infty} c(u+y\delta t) R_{\omega}(y) dy .$$

As  $\delta t \rightarrow 0$ , this tends to become equal to

$$\delta t c(u) \int_0^{\infty} R_{\omega}(y) dy = s.c(u)\delta t \quad (\text{say}).$$

Therefore  $P[\text{Downcrossing in } (t, t+\delta t) | Z(t) \geq u]$

$$= s.c(u)\delta t = \beta \delta t \quad (\text{say}) . \quad (2.57)$$

where 
$$\beta = \frac{f(u)}{R(u)} \int_0^{\infty} R_{\omega}(y) dy = \frac{f(u)}{R(u)} \int_0^{\infty} y.f_{\omega}(y) dy$$

If  $Z(t)$  is normally distributed with mean zero, then

$$\int_0^{\infty} y.f_{\omega}(y) dy = \frac{\sigma_{\omega}}{\sqrt{2\pi}}$$

where  $\sigma_{\omega}^2 = -\sigma^2 g''(0)$ .

So in this case 
$$\beta = \frac{e^{-u^2/2\sigma^2}}{1-\Phi\left(\frac{u}{\sigma}\right)} \frac{\sigma_{\omega}}{\sqrt{2\pi}} \quad (2.58)$$

Suppose now we consider the random variable  $L$ , defined to be the distance from the last upcrossing to the next downcrossing.

$$P[L \in (\ell, \ell + \delta t) | L \geq \ell] = \beta \delta t .$$

Let  $f_L(\ell)$  be the probability density function of  $L$ , and  $R_L(\ell) = P[L \geq \ell]$ .

Then 
$$f_L(\ell)\delta t = \beta\delta t R_L(\ell) . \quad (2.59)$$

Therefore

$$-\frac{dR_L(\ell)}{d\ell} = \beta R_L(\ell)$$

and

$$\frac{dR_L(\ell)}{R_L(\ell)} = \beta d\ell$$

which implies that  $R_L(\ell) = Ke^{-\beta\ell}$

$$\ell = 0 \Rightarrow R_L(0) = 1 \Rightarrow K = 1 .$$

Therefore

$$R_L(\ell) = e^{-\beta\ell}$$

and

$$f_L(\ell) = \beta e^{-\beta\ell} \quad (2.60)$$

Thus  $L$  has a negative exponential distribution, whose mean is

$$E(L) = \frac{1}{\beta} = \frac{R(u)}{s.f(u)} \quad (2.61)$$

We can show, by a similar argument to the above, without conditionality on  $Z(t) \geq u$ , that

$$P[\text{Downcrossing in } (t, t+\delta t)] = s.f(u) \delta t \quad (2.62)$$

Let  $X(t) = 1$  if there is a downcrossing in  $(t, t+\delta t)$   
 $= 0$  otherwise.

Consider an interval of size  $T$ , and divide it into  $M$  small sub-intervals of size  $\delta t = T/M$ . Assume that  $\delta t$  is sufficiently small that the probability of more than one downcrossing in  $\delta t$  can be neglected.

$$E[\text{No. of downcrossings in } T] = E\left[\sum_{i=1}^M X(t_i)\right]$$

$$= M s f(u) \delta t = s f(u) T \quad . \quad (2.63)$$

Therefore, if  $N$  is the number of downcrossings per unit interval, which is equal to the number of excursions per unit interval,

$$E(N) = s f(u) \quad , \quad (2.64)$$

and if  $Z(t)$  is normally distributed with mean zero, then

$$E(N) = \frac{\sigma_{\omega}}{\sqrt{2\pi}} e^{-u^2/2\sigma^2} \quad . \quad (2.65)$$

It would be nice if these simple results could be extended in a straightforward fashion to higher dimensions - unfortunately this is not the case. Adler (1976) discusses thoroughly the problems involved in dealing with the random variables associated with excursions of a stationary stochastic process above a fixed level in more than one dimension. One result that can be obtained is that the "volumes" of such excursions tend to have a negative exponential distribution, but only asymptotically as  $u$  tends to infinity.

Adler and Hasofer (1976) have generalised the notion of the number of downcrossings of a stochastic process per unit interval to define a "characteristic"  $\chi$  of an  $m$ -dimensional process.  $\chi$  is closely related to the number of connected components of the excursion set of the process above the fixed level  $u$ . The difference arises in the case where components of the excursion set contain "holes" or totally enclosed regions where  $Z(\underline{x}) < u$ . Probabilistic calculations may be carried out on  $\chi$  which are not possible on the more directly useful variable.

For example, if  $Z(\underline{x})$  is a two-dimensional stationary process, Adler and Hasofer show that

$$E(\chi) = (2\pi\sigma^2)^{-3/2} \sigma_{\omega}^2 u e^{-u^2/2\sigma^2} \quad , \quad (2.66)$$

by comparison with equation (2.65) for the one-dimensional case.

In Chapter 6 of this work I shall use the concept of excursion sets in two dimensions to model the occurrence of oilfields, and derive approximately the expected area of an arbitrary such excursion.

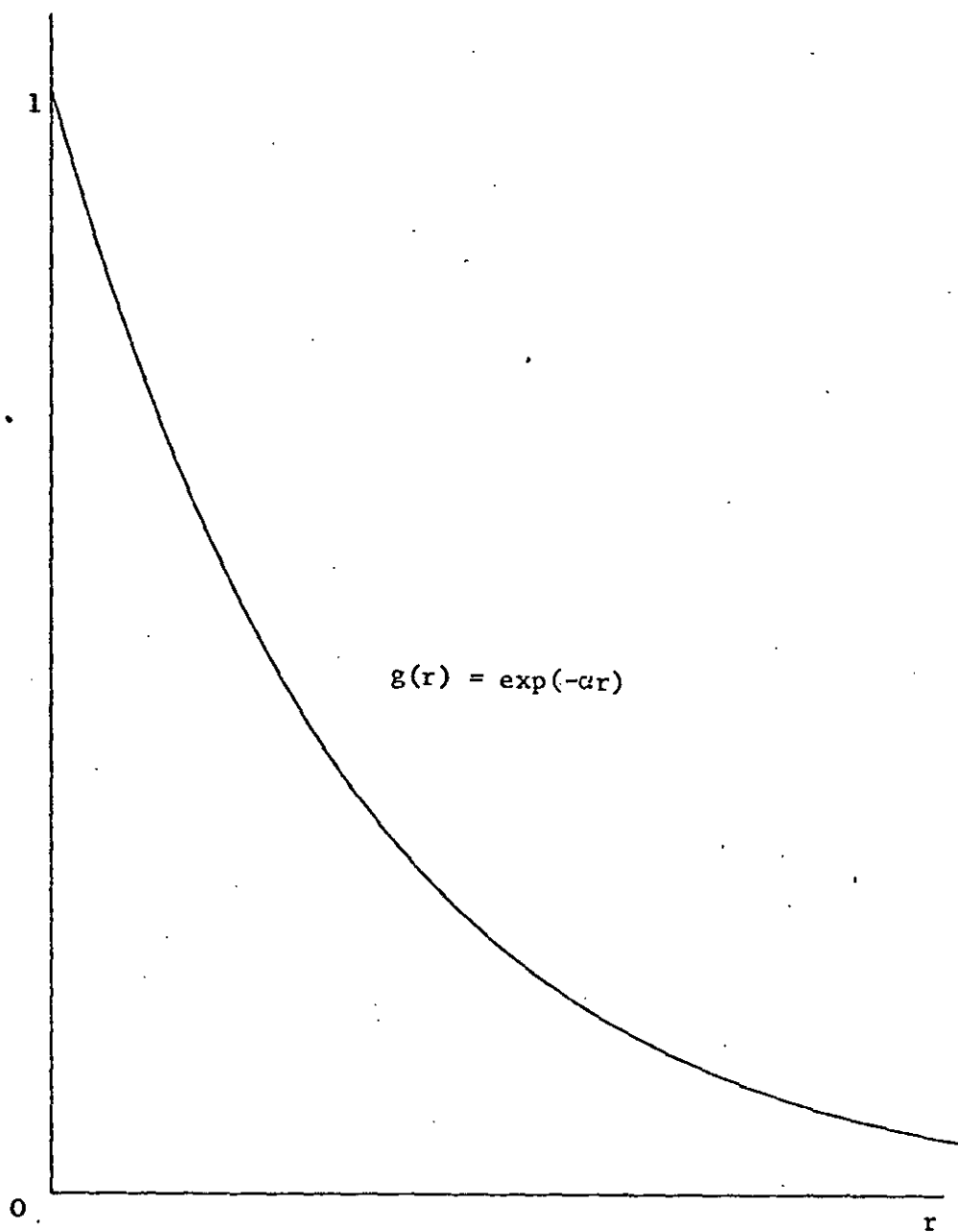


FIGURE 2.1: Auto-correlation function example

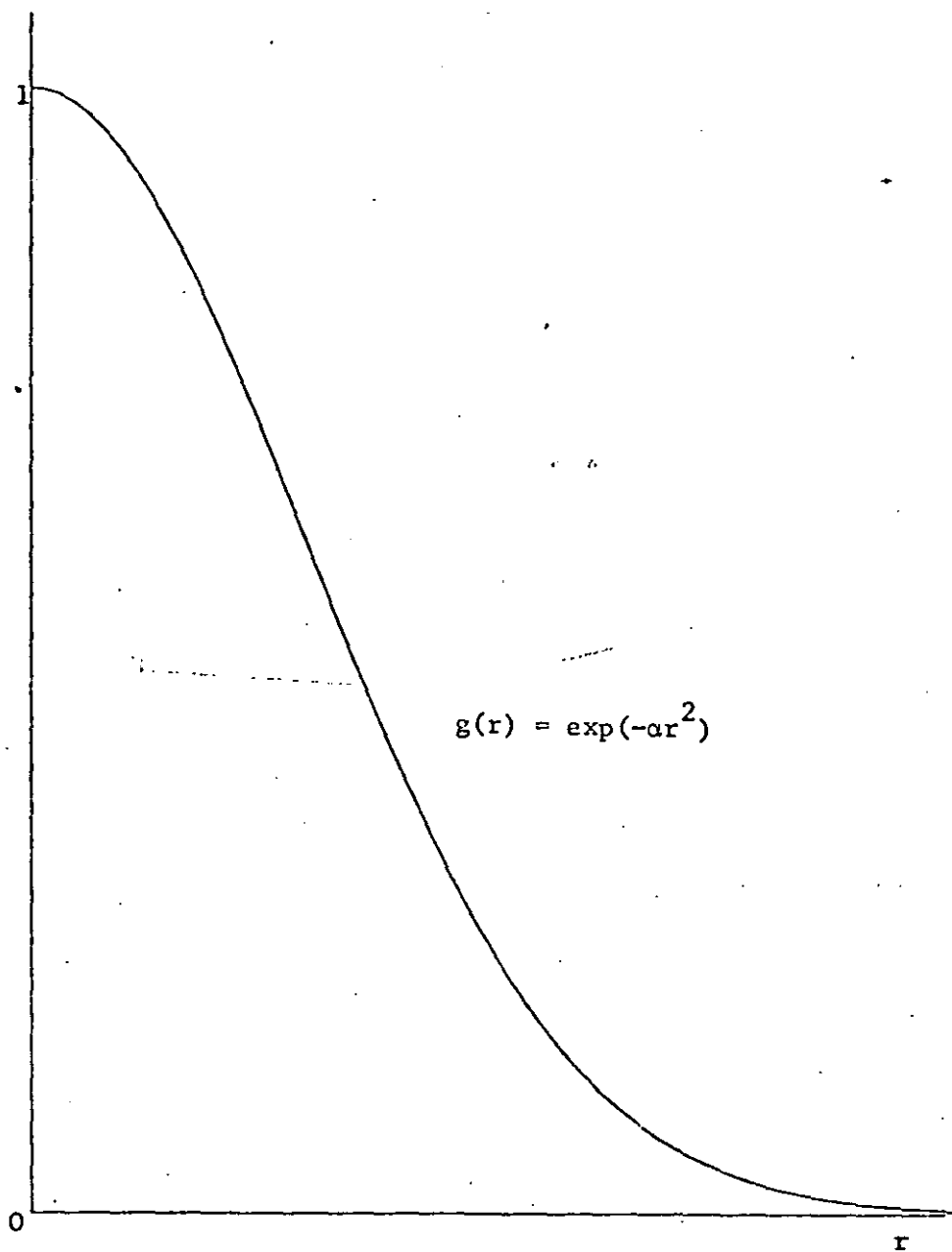


FIGURE 2.2: Auto-correlation function example



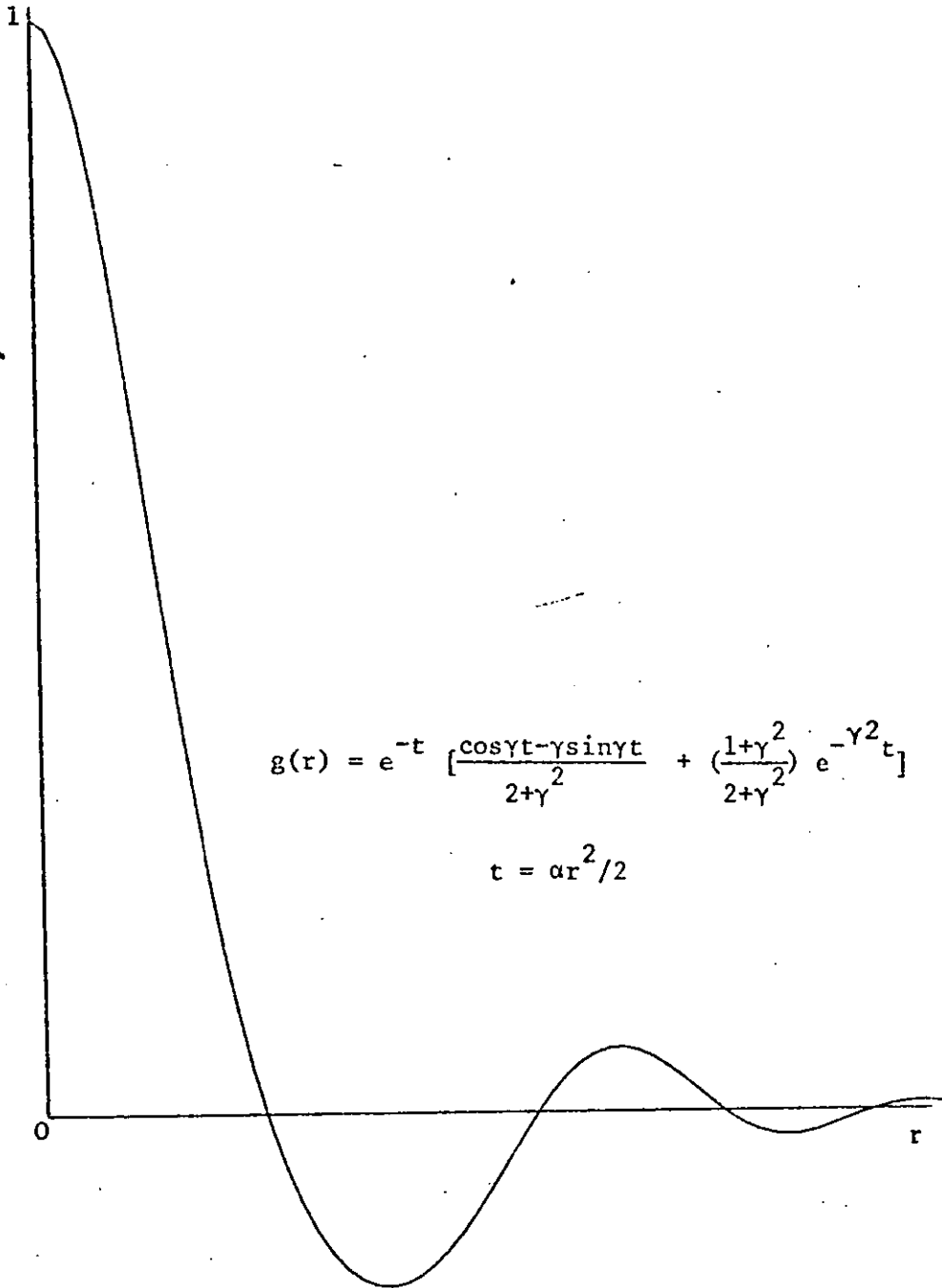


FIGURE 2.3: Auto-correlation function example

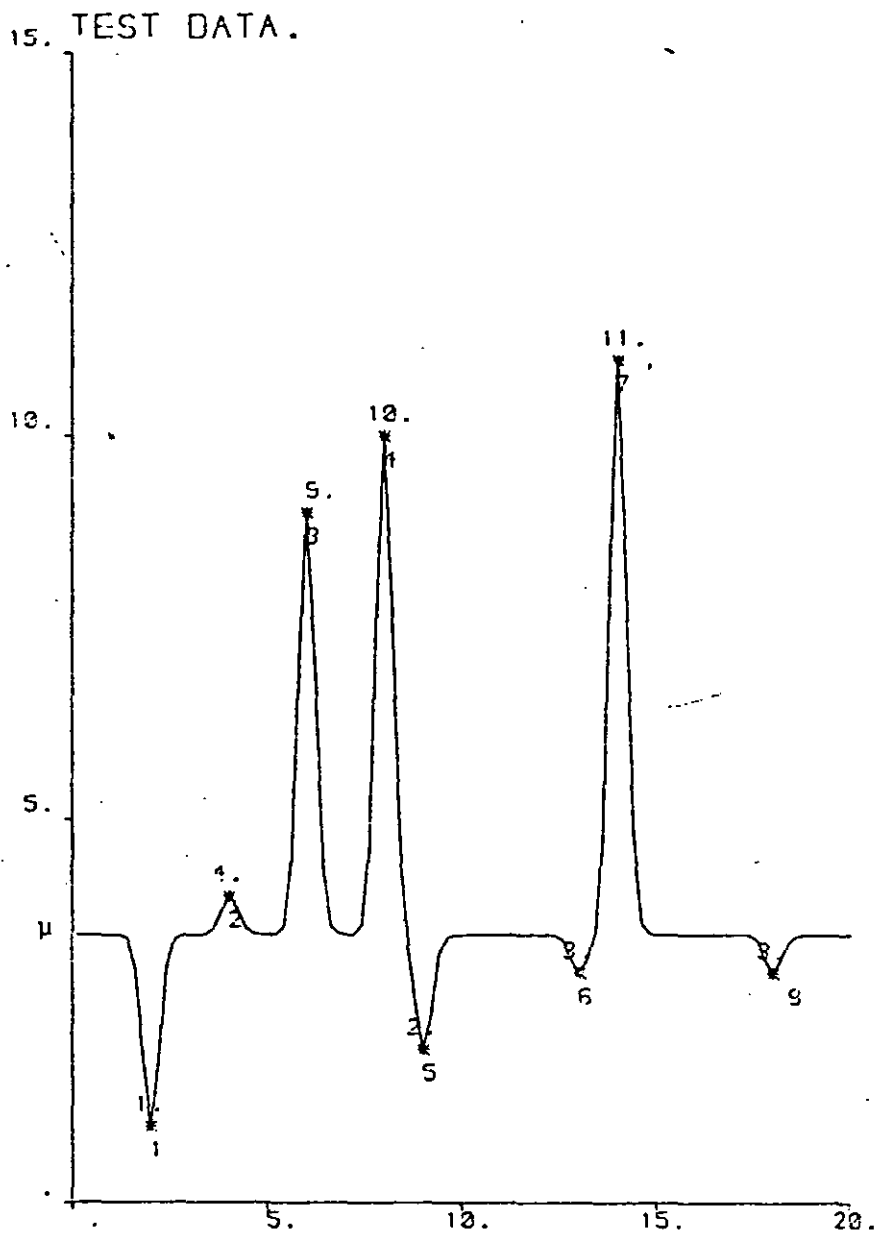


FIGURE 2.4(a): Example of stochastic interpolation  
 - small correlation distance  $\rho$

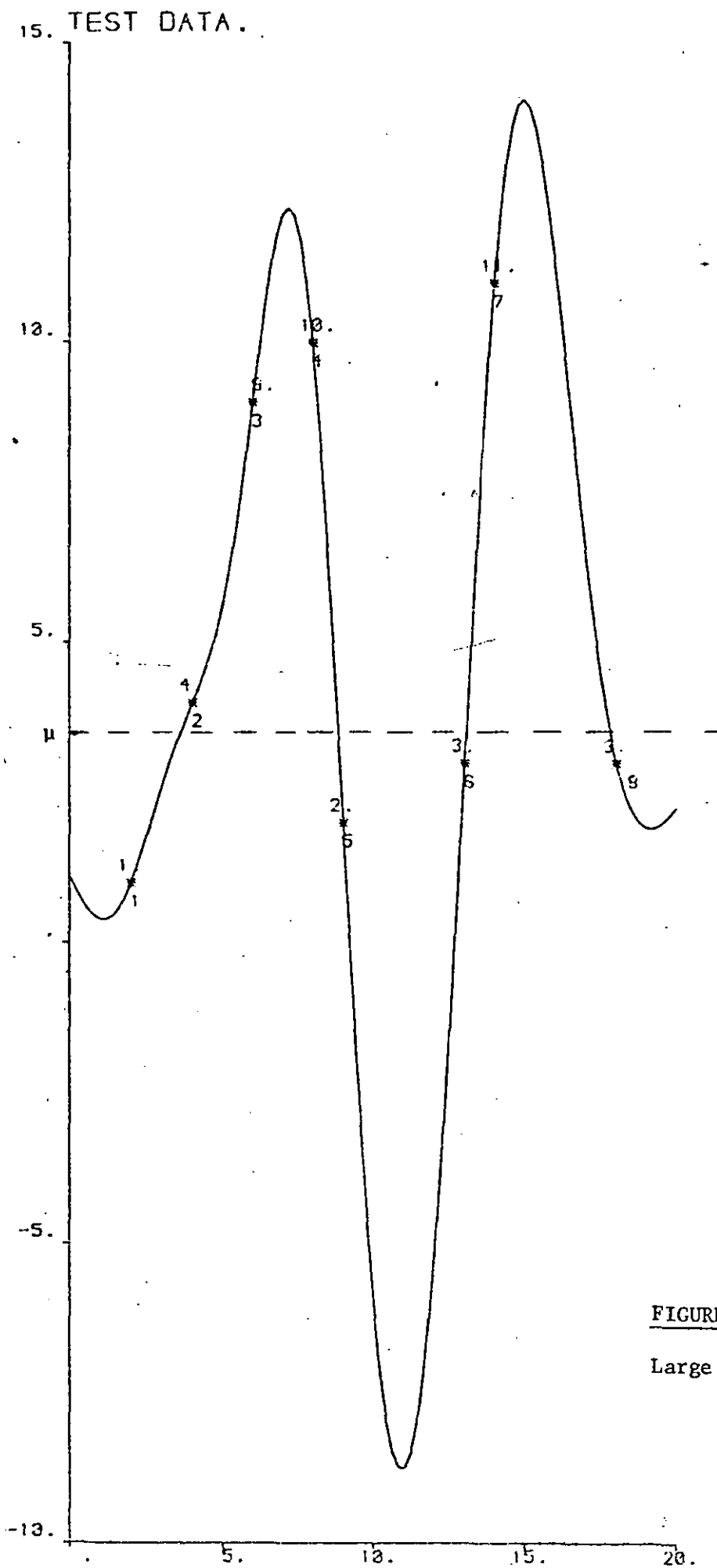


FIGURE 2.4(b):  
Large correlation distance  $\rho$   
 $\rho = ?$

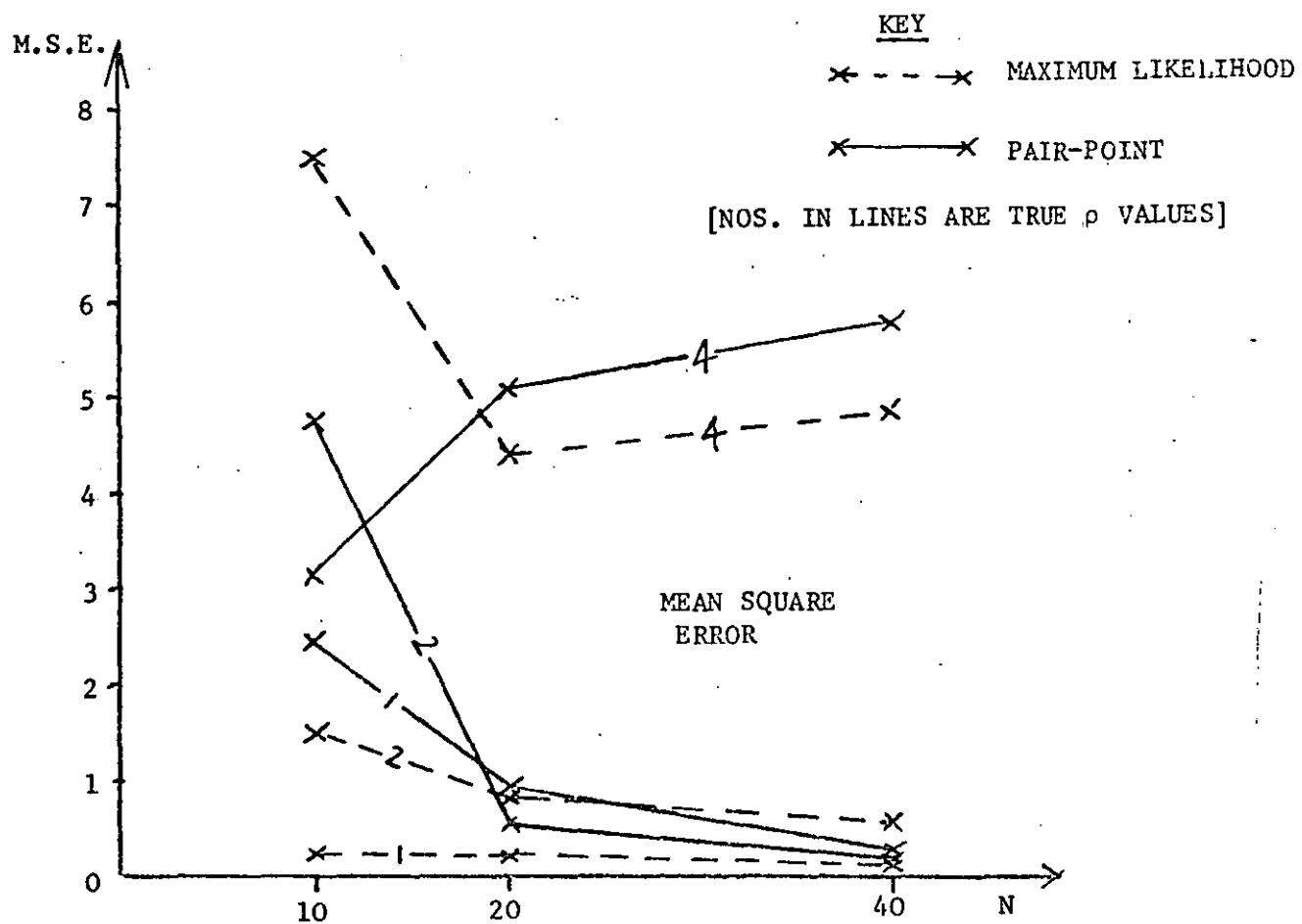
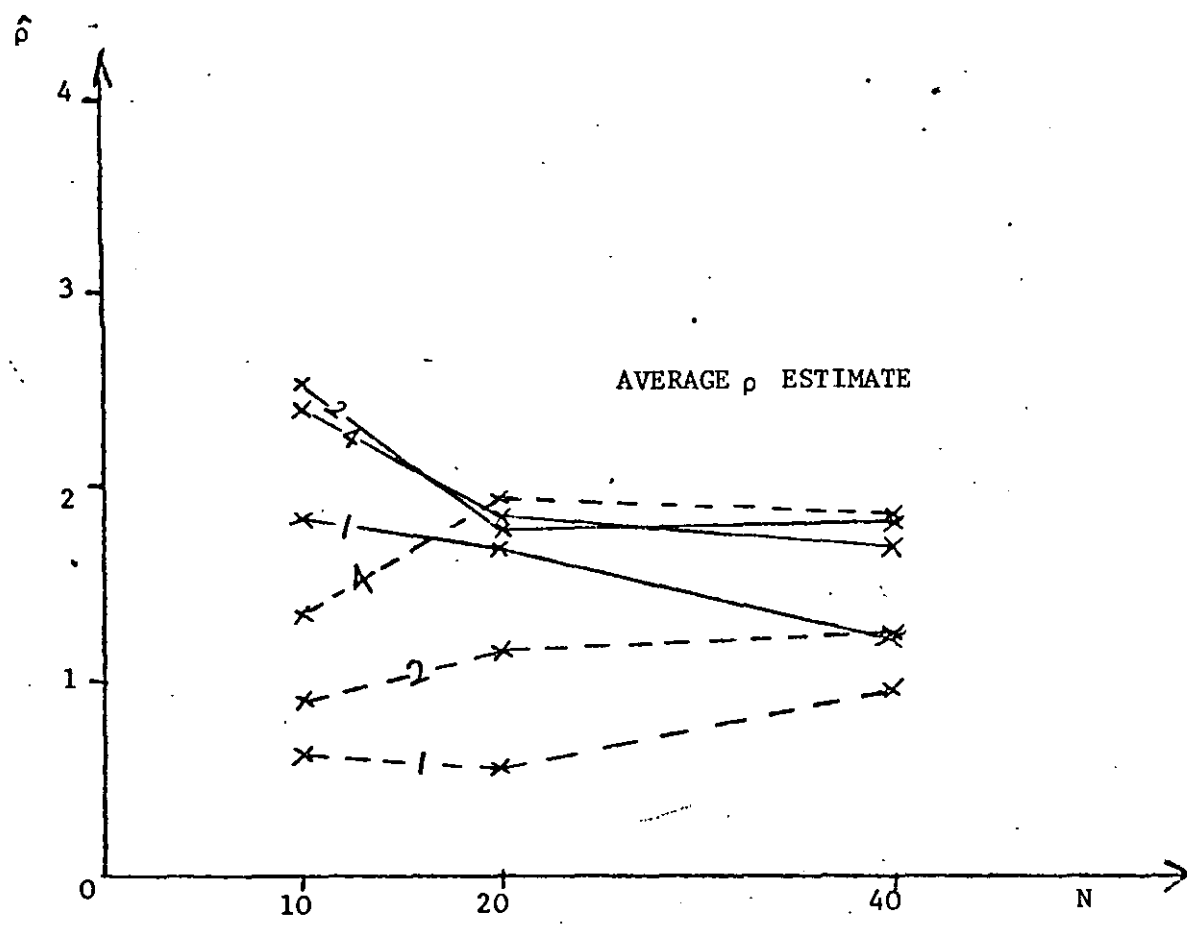


FIGURE 2.5: Results of experiments in  $\rho$  estimation

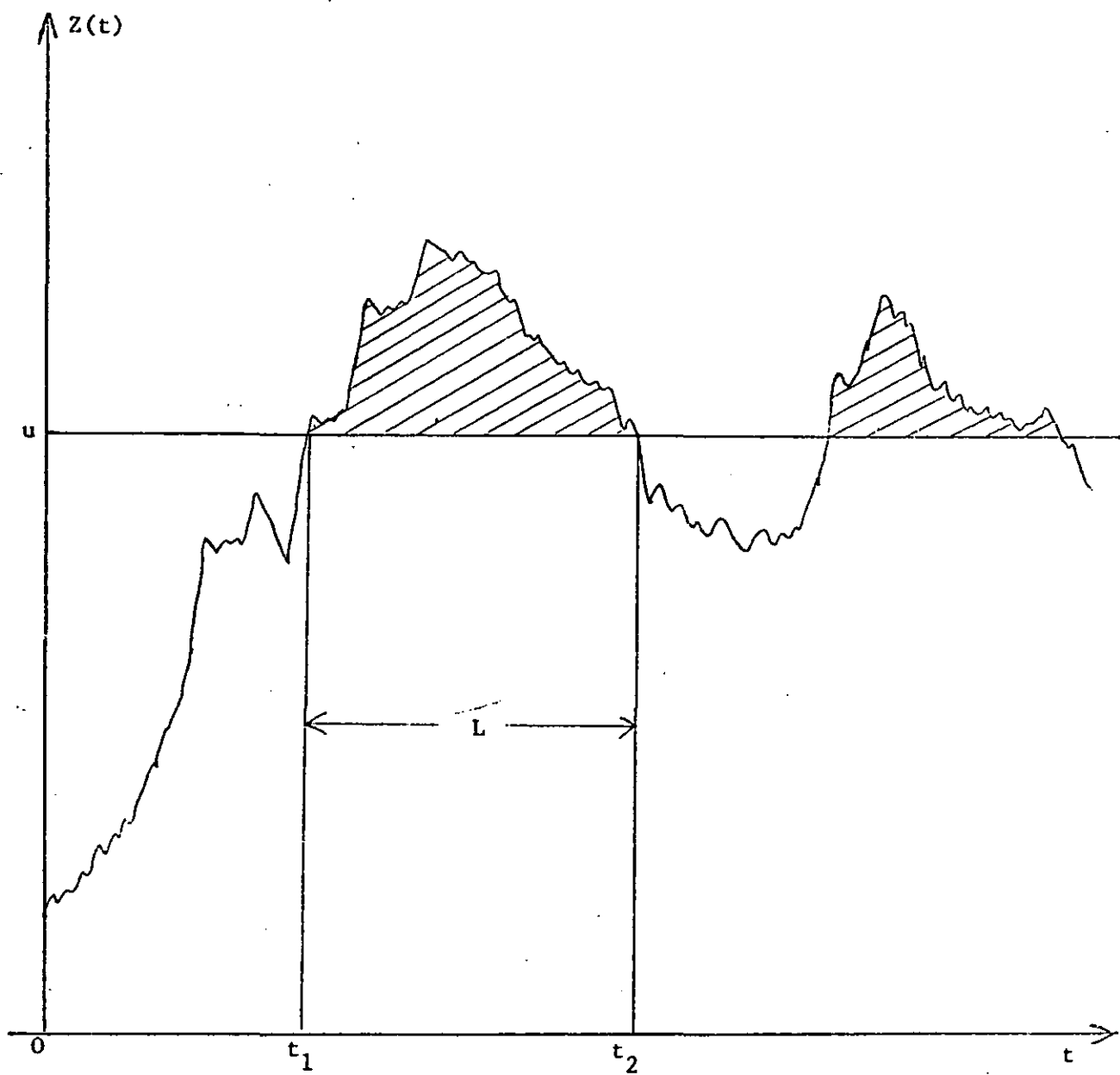


FIGURE 2.6: Excursions of a stochastic process above the level  $u$

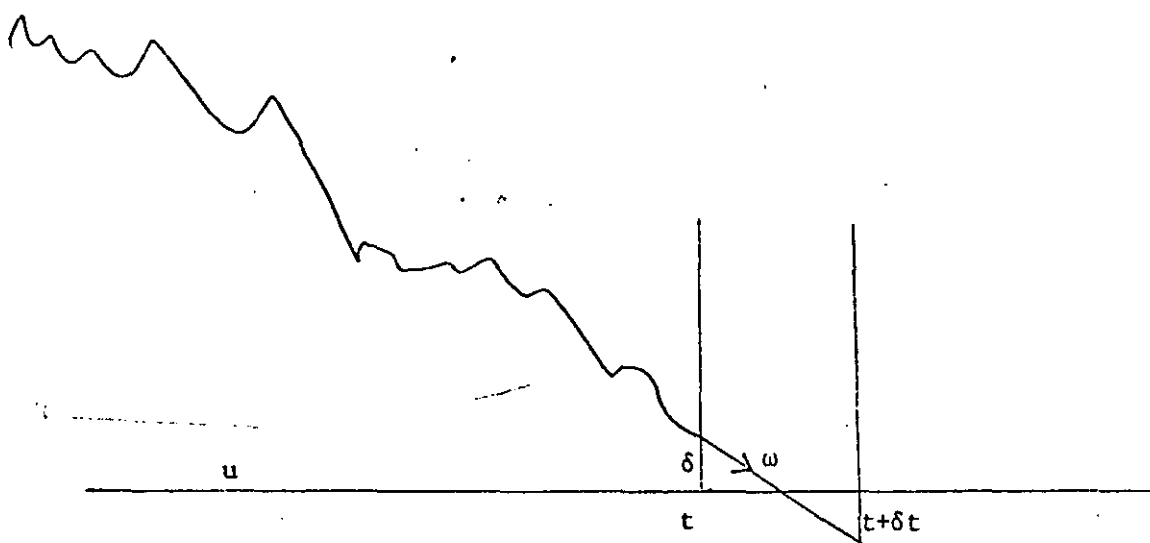


FIGURE 2.7: Downcrossing of the level  $u$  by the stochastic process in a small interval  $\delta t$

CHAPTER 3

OTHER APPROACHES TO THE SAME PROBLEMS

### 3.1 THE THEORY OF REGIONALISED VARIABLES AND KRIGING

The French geostatistician Matheron has developed a theory of "regionalised variables" which he applies to problems in the mining industry. It may be instructive to outline briefly the basic concepts of his theory and compare it with this work. Unfortunately, Matheron has also developed his own terminology to describe his methods, which often makes translation into terms of conventional theory difficult.

Regionalised variables are assumed to be realisations of stochastic processes in one, two or three dimensions. Since the theory is to be applied to physical properties (e.g. of ore-bodies) which are highly variable from point to point and which appear not to possess a constant mean, the stochastic processes considered are not necessarily stationary. If they are stationary, then they need not possess a finite variance or be differentiable or continuous. To allow for the possibility of non-stationary functions with infinite variances, Matheron introduces a weaker condition than stationarity, which he terms the "intrinsic hypothesis". (See Matheron, 1971, p.53; Journel & Huijbregts, 1978, p.33).

This hypothesis is that the increment  $Z(\underline{x}+\underline{r})-Z(\underline{x})$  has a mean and variance which are independent of  $\underline{x}$ .

i.e.

$$\begin{aligned} E[Z(\underline{x}+\underline{r}) - Z(\underline{x})] &= m(\underline{r}) \\ \text{Var}[Z(\underline{x}+\underline{r}) - Z(\underline{x})] &= 2\gamma^*(\underline{r}) \end{aligned} \quad (3.1)$$

The function  $\gamma^*(\underline{r})$  is known as the "semi-variogram" and can be seen to be related to the auto-covariance function  $\gamma(\underline{r})$  if  $Z(\underline{x})$  is stationary and has finite variance.

$$\gamma^*(\underline{r}) = \frac{1}{2}\text{Var}[Z(\underline{x}+\underline{r}) - Z(\underline{x})]$$



$$\begin{aligned}
& \doteq \frac{1}{2} E[(Z(\underline{x}+\underline{r}) - Z(\underline{x}))^2] \quad \text{if } E[Z(\underline{x}+\underline{r})] = E[Z(\underline{x})] \\
& = \frac{1}{2} [2\mu^2 + 2\sigma^2 - 2\gamma(\underline{r}) - 2\mu^2] . \\
& = \sigma^2 - \gamma(\underline{r}) . \tag{3.2}
\end{aligned}$$

Note that if  $\gamma^*(\underline{r})$  tends to a finite limit as  $\underline{r}$  tends to infinity, then the process is stationary, with variance equal to this limiting value. Otherwise, if the semi-variogram tends to infinity, the process is non-stationary (see Figure 3.1).

Thus Matheron uses the "intrinsic function"  $\gamma^*(\cdot)$  in preference to the auto-covariance function  $\gamma(\cdot)$  because of its applicability to a wider class of stochastic processes. However, in most of the applications in this work (contouring, interpolation of unknown functions) the assumption of stationarity is a reasonable one to make, and it is simpler and more natural to use the auto-covariance or auto-correlation function.

In actual practice, the difference between a stationary and a non-stationary process is very difficult to detect - it depends upon the domain over which the process is being studied. An apparently non-stationary trend may in fact be a local manifestation of a long-range variation which is itself stationary (see Figure 3.2).

This leads on to the concept of "universal kriging" in Matheron's theory. "Kriging" relates to the estimation of unknown values or integrals - thus "punctual kriging" refers to the estimation of values of the stochastic process at unknown points. In "universal kriging" the assumption is made that the process has a "trend function"  $m(\cdot)$  such that

$$E[Z(\underline{x})] = m(\underline{x}) , \tag{3.3}$$

as well as the intrinsic function

$$\gamma^*(\underline{r}) = \frac{1}{2} \text{Var}[Z(\underline{x}+\underline{r}) - Z(\underline{x})]$$

(see Figure 3.3).

Some sort of functional form is assumed for  $m()$  (e.g. polynomial) and parameters of this function need to be fitted to the data, and the form of  $\gamma^*()$  must simultaneously be estimated. Given all these parameters of the regionalised variable model fitted to the known data, best linear unbiased estimates of any function of the underlying variable may be obtained. Thus universal kriging leads to an iterative procedure in which an estimate of  $\gamma^*()$  enables a form of  $m()$  to be fitted, and hence residuals can be calculated. The correlation structure of these residuals leads to an updated form of  $\gamma^*()$  and so on.

In general, a fairly large number of parameters need to be estimated, specially if  $m()$  is to be fitted in several dimensions. There may well be no good a priori reason why  $m()$  should have any particular functional form, so attempts to fit polynomials may not be particularly useful, and can lead to dangerous extrapolation tendencies. Whittle (1963, p.84-85) has some cautionary words regarding this approach with regard to time series.

What seems to be a better philosophy for modelling functions which include some form of "trend" is to treat them as realisations of a stochastic process which is a sum of stationary components. These individual processes will have different ranges over which their correlations extend, and this will lead to a simpler, more uniform model with fewer parameters to be estimated.

Olea (1974) has applied universal kriging to automatic contouring,

and claims that the maps so produced are "optimal" in the sense of producing minimum variance unbiased estimates of the unknown values. However, Akima (1975) has criticised this claim by pointing out that the optimality criterion chosen requires certain fairly strict conditions on the structure of the data, and that other criteria might well be more applicable for other types of data. Generally speaking, it is true that no one method is going to be "optimal" for all possible sets of data.

Akima proceeds to more detailed criticisms of Olea's methods, but also points out a crucial problem with all techniques for fitting a "drift" function together with correlated residuals. This is the problem of the inter-relation between the chosen "drift" function and the form of correlation for the residuals - a large number of possible selections may be made of drift/correlation combinations, all of which will fit the observed data. But, as Akima says: "The question is whether or not such a selection can be made objectively and automatically with a prescribed algorithm". The chances are that it cannot, and that an element of subjective decision and choice will always be present in automatic contouring.

One important difference between geostatistics and the types of problem dealt with in this work concerns the amount and regularity of data. Generally speaking, it is the case that geostatistical data is collected at a large number of points on a regular grid. It is thus reasonable to fit complex models with many parameters and to expect to extract meaningful information from such models. (See Journel & Huijbregts, 1978 - for example their case study 11, pp.272-280).

An important part of geostatistical analysis is the fitting of the variogram to the data - equivalent to our problem of estimating  $\rho$ , (Section 2.4), but normally with enough data to enable the functional form of the variogram to be checked. David (1977, p.119) describes various general strategies for fitting variograms, while Journel & Huijbregts (1978, pp.207ff) give formulae and computer programs for calculating experimental variograms in different situations. Of particular interest is their method for "non-aligned data" (p.211,223) - i.e. randomly scattered data. Their preferred method is to use "angle classes" and "distance classes" to estimate the variogram averages in different directions. From the experimental variograms so derived, fits of more or less complex theoretical variograms may be obtained. On pp.192-195, Journel & Huijbregts discuss the variogram estimation variances, and note that strict goodness-of-fit tests for this problem would almost never invalidate the fit, because of the large "fluctuation variance". On pp.233-235 they investigate the robustness of the geostatistical results with respect to two different model variograms fitted to the same data - the conclusion is that the difference is negligible in the case studied. So that even in geostatistical analysis, with large amounts of data, the form of model variogram fitted may be governed less by the data and more by subjective considerations.

Hawkins & Cressie (1981) have developed a system for the robust estimation of the variogram in the presence of outliers. Taking pairs of points  $Z_t$  and  $Z_{t+h}$  a distance  $h$  apart, they show that  $Y_t = \frac{Z_t - Z_{t+h}}{\sqrt{2}}$  has a probability distribution which is close to Normal if the  $Z_t$  values are Normally distributed. They consider robust estimators of the variogram

proportional to  $\bar{Y}^4$  and  $\tilde{Y}^4$ , where  $\bar{Y}$  and  $\tilde{Y}$  are the mean and median of the  $Y_t$  values respectively, and test them against simulated data with outliers. We shall return to consideration of estimators of this type when we discuss the estimation of anisotropy (Section 5.4).

### 3.2 OPTIMISATION OF EXPENSIVE OBJECTIVE FUNCTIONS

The optimisation problem is that of finding a point  $x$  which maximises (or minimises) an  $m$ -dimensional function  $F()$  within some specified "region of interest"  $R$ . Practical optimisation problems may be broadly grouped into three classes, as follows:

- a) Local optimisation problems. Starting from a given initial point in  $R$ , to move to local maximum (or minimum) of  $F()$ .
- b) Global optimisation problems. To find the local optimum point in  $R$  with the highest (or lowest) value among the class of all local optima.
- c) Expensive optimisation problems. Assuming that each evaluation of  $F()$  at a new point is "expensive" in some way (e.g. in terms of computing time), to find a reasonably good approximation to a global optimum value in an acceptably small number of function evaluations.

Much work has been done on type a) problems, and many excellent algorithms exist (see for example Zoutendijk, 1976 & Fletcher, 1980). Most modern algorithms require knowledge of the first derivatives of the function  $F()$ , either given explicitly or computed numerically.

The global optimisation problem, type b) above, is theoretically impossible to solve. There is no guarantee that any given algorithm will detect a global optimum in a finite number of function evaluations for all possible objective functions. Figure 3.4 illustrates the problems involved. However, this fact has not prevented some work being

done in this field, the assumption being that most functions of interest will be reasonably well-behaved. Dixon et al (1975) have produced a general survey of the problems involved in global optimisation and some of the strategies which have been used.

Most strategies involve some modification of the totally random search technique, which is to generate a large number of random points in  $R$  and choose the largest (or smallest). Alternatively, a number of random points in  $R$  may be generated and from each such point a local optimisation routine initiated. Then choose the largest (or smallest) local optimum so found. These crude methods can be modified in various ways to improve the efficiency with which the global optimum value is found (normally measured in terms of number of function evaluations). (see e.g. Solis & Wets, 1981).

For example, Price (1977) describes a "controlled random search procedure". An  $m$ -dimensional function  $F()$  is optimised by generating an initial set of  $N$  points randomly in the region of interest  $R$ . New points are generated taking into account the  $N$  existing points by selecting a random subset of  $m+1$  points from the full set of  $N$  points. This subset of points forms a simplex in the  $m$ -dimensional space, and a new trial point is generated by reflecting an arbitrary member of the subset in the centroid of the simplex (see Figure 3.5). If the new trial point has a value  $F(\underline{x})$  better than the worst point in the current set of  $N$  points, that worst point is dropped from the current set and the new point is included. In this way it is hoped that the  $N$  points will tend to cluster about global optima as the algorithm continues.

Price quotes some results from tests on different objective functions.

For example, with a 9-dimensional function (described in more detail in Chapter 6), a very good approximation to a global optimum point was found after six runs of the program, each time restarting with a smaller region of interest centred on the end point of the previous run. Each run required of the order of 20,000 to 30,000 function evaluations, so this procedure would not be ideal for an objective function which was expensive to compute.

de Biase and Frontini (1978) describe a method based on random sampling of points within  $R$ . Their first aim is to estimate the function  $\psi()$ , where

$$\psi(\xi) = P[\text{Random point } \underline{x} \in R \text{ has } F(\underline{x}) \leq \xi] \quad , \quad (3.4)$$

or, alternatively,  $\psi(\xi)$  is the normalised Lebesgue measure of the subset of  $R$  for which  $F(\underline{x}) \leq \xi$ .

If the function  $\psi()$  is known, then the minimum value of  $F()$  in  $R$  may be obtained by setting  $\psi(\xi)=0$ . de Biase and Frontini set out first to estimate  $\psi(\xi)$  by random sampling in  $R$ . Sets of  $q$  random points are generated iteratively and pairs of values  $(\xi_1, \hat{\psi}_1)$  are obtained for each such set. This is repeated and spline approximations are used to fit the function  $\psi()$  to these results. This stage of the procedure is terminated when a consistent fit is achieved, and enough points are assumed to have been generated. The predicted minimum value  $\beta^*$  of  $F()$  can be obtained from these results.

The second stage of their procedure is to group the points generated in the first stage into clusters and carry out a search for a local optimum within each cluster. Results for this algorithm for two test functions considered later (see Chapter 6) are tabulated on the next page.



<u>Function</u>	<u>Stage 1</u> <u>Function</u> <u>Evaluations</u>	<u>Total</u> <u>Function</u> <u>Evaluations</u>	$\beta^*$ <u>(Predicted</u> <u>Minimum)</u>	$F^*$ <u>(Final</u> <u>Minimum)</u>
Branin's RCOS	142	208	2.360	1.250
Goldstein & Price	72	144	3.5513	2.9997

Thus the first stage (initial random sampling) and the second (local searches) take a similar number of function evaluations.

It seems obvious that methods based on random sampling are not going to be of maximum efficiency. Points will not be evenly spread throughout the region of interest, but will tend to clump together, leaving uneven spaces between (see Figure 3.6). Two points which are very close are not contributing fully to a knowledge of the function behaviour, assuming the function is spatially correlated to some degree, since the value at one point could have been inferred, to a greater or lesser extent, from the other point. At the same time information is being lost in the empty spaces.

For an expensive objective function it seems clear that random sampling is not efficient enough, and that points must be spread as evenly as possible throughout the region of interest so as to maximise the information gained from a small number of function evaluations.

### 3.3 PREDICTION OF THE OCCURRENCE OF OILFIELDS

An "oil province" may be defined as a geological area within which oilfields have been or may be discovered. Such an oil province is explored by conducting geophysical and geological surveys which locate subsurface structures with potential for accumulating oil, and by drilling "wildcat" wells to determine whether or not an oilfield is present at each such location. It is obviously of considerable interest to be able to predict for a given oil province the number of oilfields which are actually present and the reserves of oil which they contain. If the oil province has been thoroughly explored, this is not difficult since the majority of the fields will have been discovered. If however the exploration has just begun, it requires much more insight to be able to make useful predictions.

Some work has already been done in putting these predictions on a sounder footing than sheer guesswork. In particular the Russians have studied the subject, see for example Juca and Nitkiewicz (1975). The best-known attempt in the West is probably that by Odell and Rosing (1974) to predict the development of the North Sea oil province.

Odell and Rosing set out to predict not only the total recoverable reserves of the North Sea but also its future rate of development and production. However, the model they used was unfortunately full of ad hoc assumptions which rendered the results of little objective value. The numbers and sizes of potential oil-bearing structures were assumed, as were the success probabilities for wells drilled into the various structures. An oilfield having been discovered, the reserves initially estimated for it were assumed to appreciate consistently with time (in practice in the North Sea, unlike some other regions of the world,

initial estimates of reserves can be either too high or too low, and are not consistently low). Most of the model parameters were treated as random variables, and a Monte Carlo program was run to produce a range of results. 100 iterations of the program generated a range of total recoverable reserves for the whole North Sea of  $79 \times 10^9$  STB (stock tank barrels) to  $138 \times 10^9$  STB.

The main critique of this model is the large number of assumptions built into it, without any possibility of fitting the model parameters to the existing data in any meaningful way. Furthermore, the model takes no account of the very obvious spatial correlation between the locations of oilfields. It is normal for oilfields to cluster together in certain regions of an oil province, and for other regions to remain relatively barren. This type of behaviour should be taken into account.

A better attempt at developing a consistent methodology for forecasting oil reserves is provided by Meisner and Demirmen (1981) with their "creaming method". This consists of a model of oilfield discovery which allows the larger fields to be discovered, or "creamed off" earlier, leaving smaller and smaller fields to be found later in the exploration process. They assume that the probability of an oilfield being found with reserves between  $v$  and  $v+dv$  will be proportional to  $v^\lambda$ . They then postulate that the parameter  $\lambda$  is not a constant, but a (decreasing) linear function of the number  $n$  of exploration wells already drilled, so that

$$\lambda = \gamma_1 + \gamma_2 n, \quad (3.5)$$

with  $\gamma_2 \leq 0$ .

The mechanics of fitting this type of model to data for an oil

province are fairly complex, but the results appear quite good for their test data, although it would seem that a reasonably long exploration history is required to fit the model. Also Meisner and Demirmen's creaming model, like Odell and Rosing's model, fails to take any account of oilfield clustering or spatial correlation.

Thus I believe that any useful model for this problem of oilfield occurrence prediction should satisfy the following criteria, as far as is possible:

1. It should be simple, with only a few parameters which can be estimated from existing data at an early stage of exploration.
2. It should take into account the spatial correlation between oilfields.
3. It should predict, in addition to the total reserves of an oil province, the approximate distribution of oilfields within the oil province.

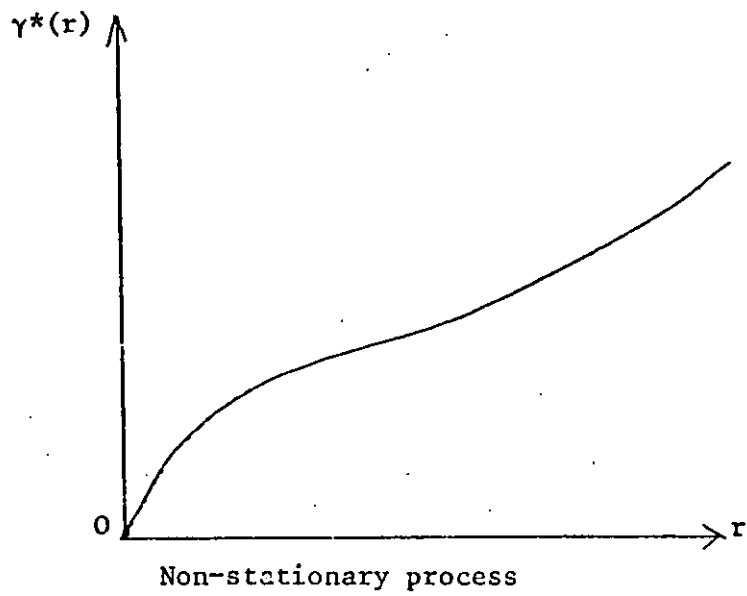
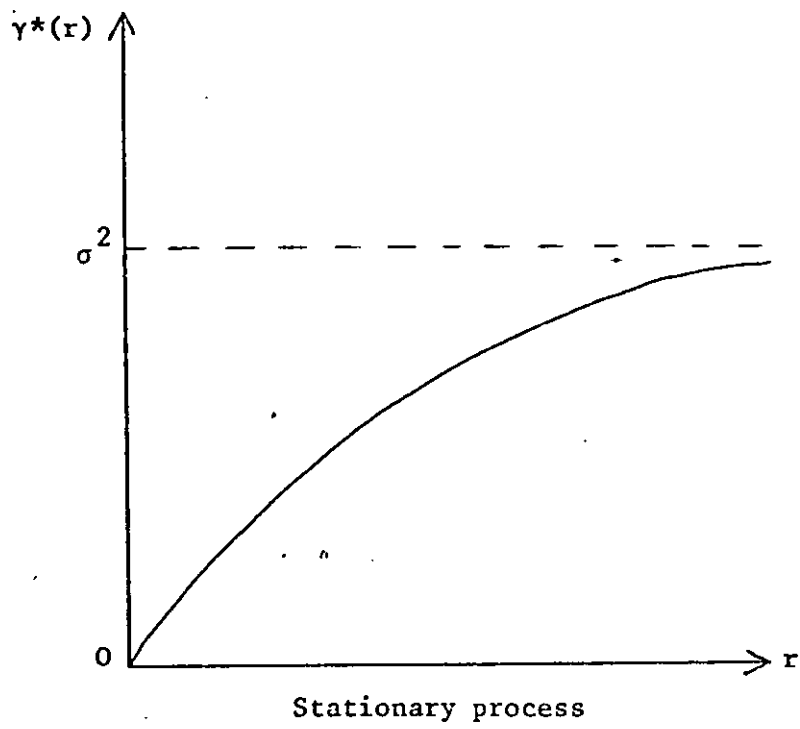


FIGURE 3.1: Examples of semi-variograms

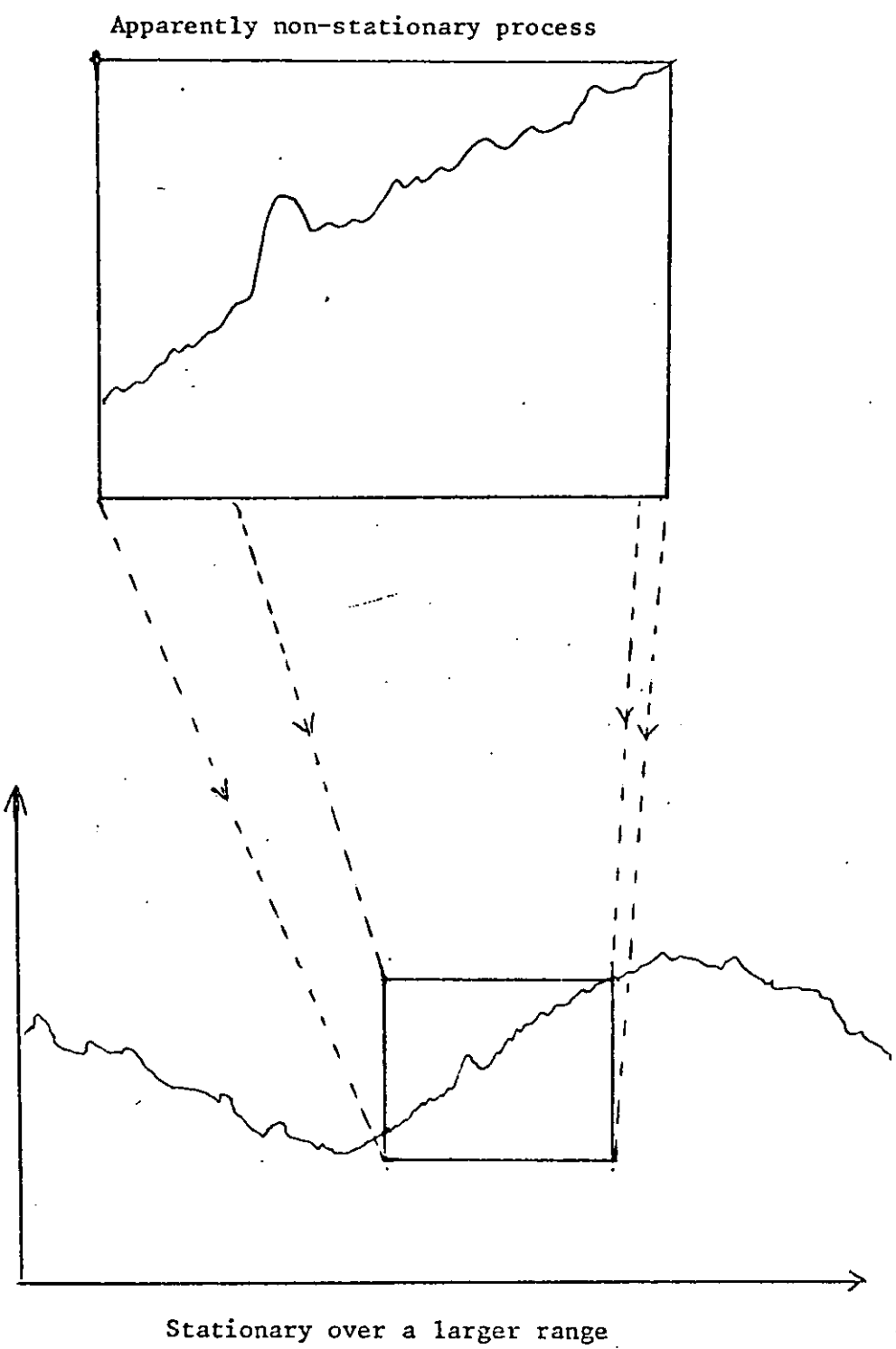


FIGURE 3.2: Illustration of the difficulties involved in deciding a process is non-stationary

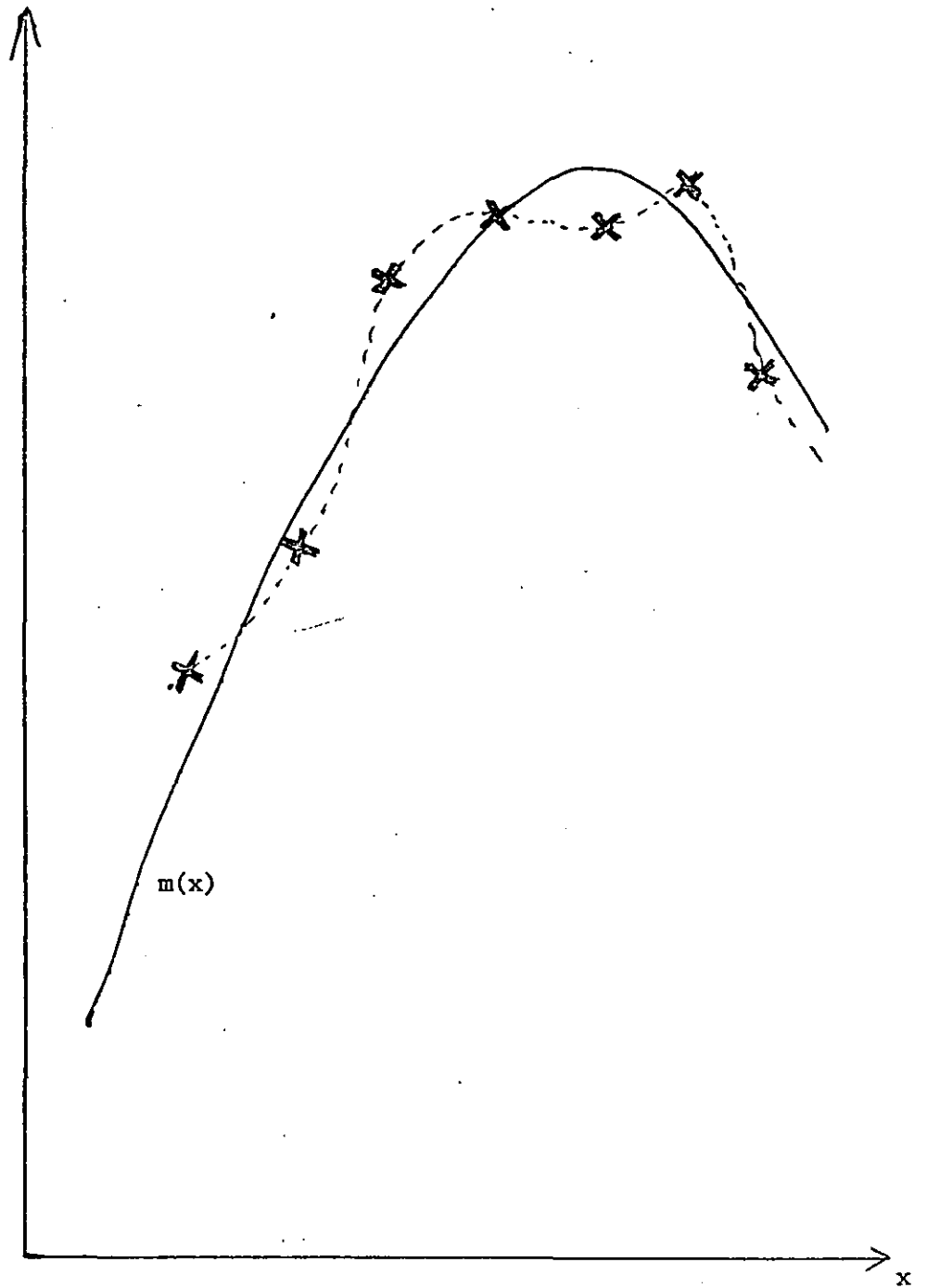


FIGURE 3.3: Illustration of "Universal kriging"

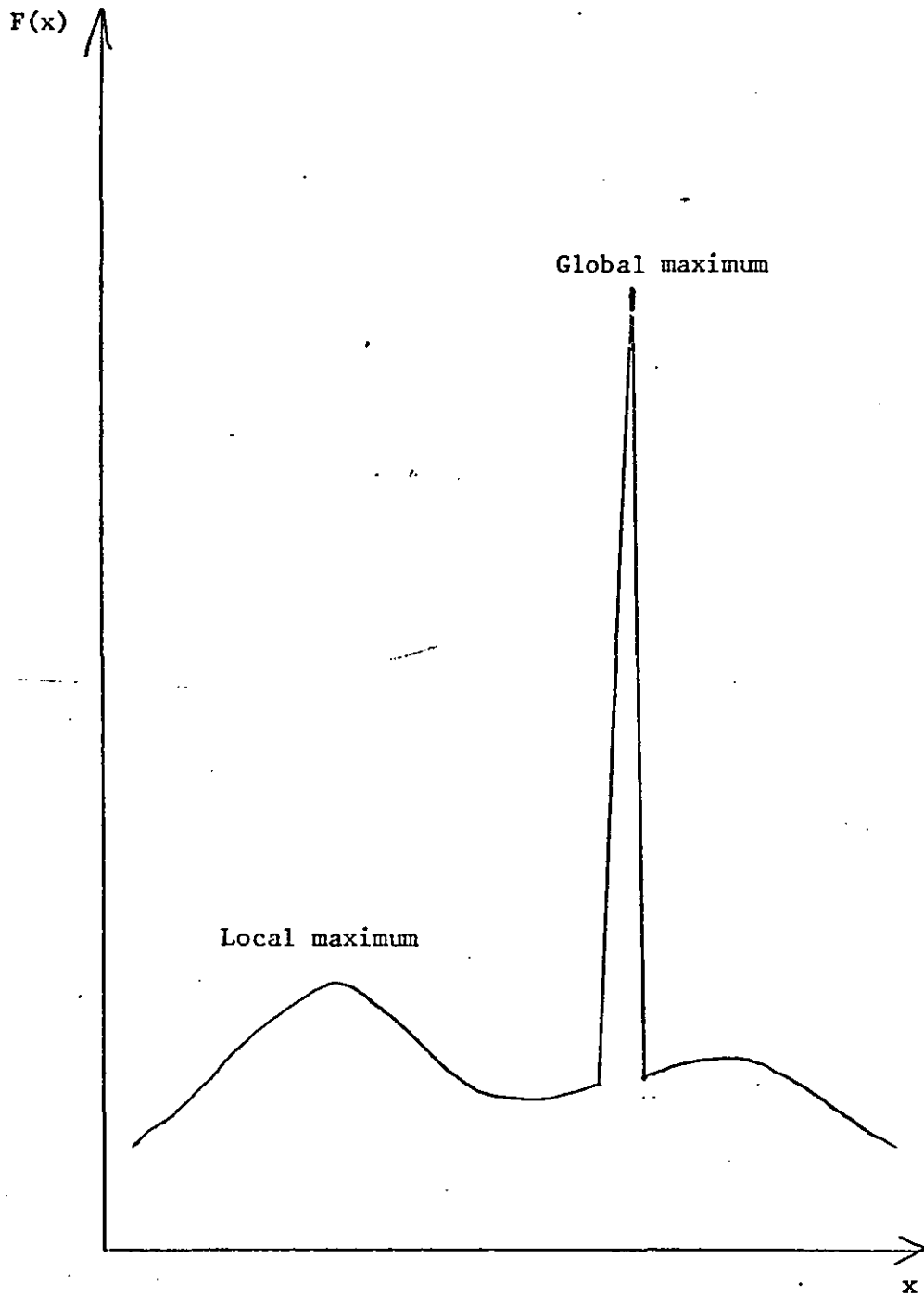


FIGURE 3.4: Pathological case for global optimisation



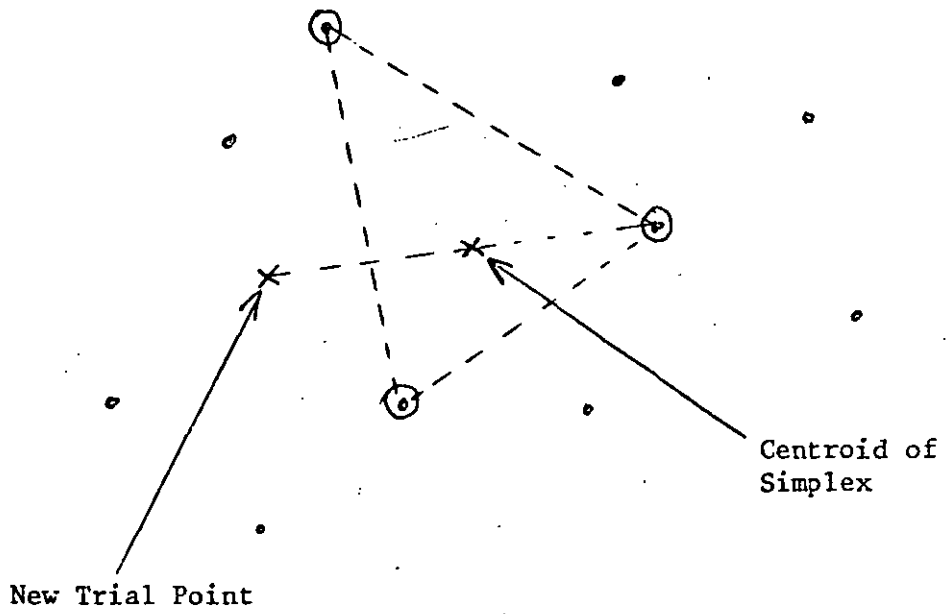


FIGURE 3.5: Illustration of choice of new point in controlled random search procedure

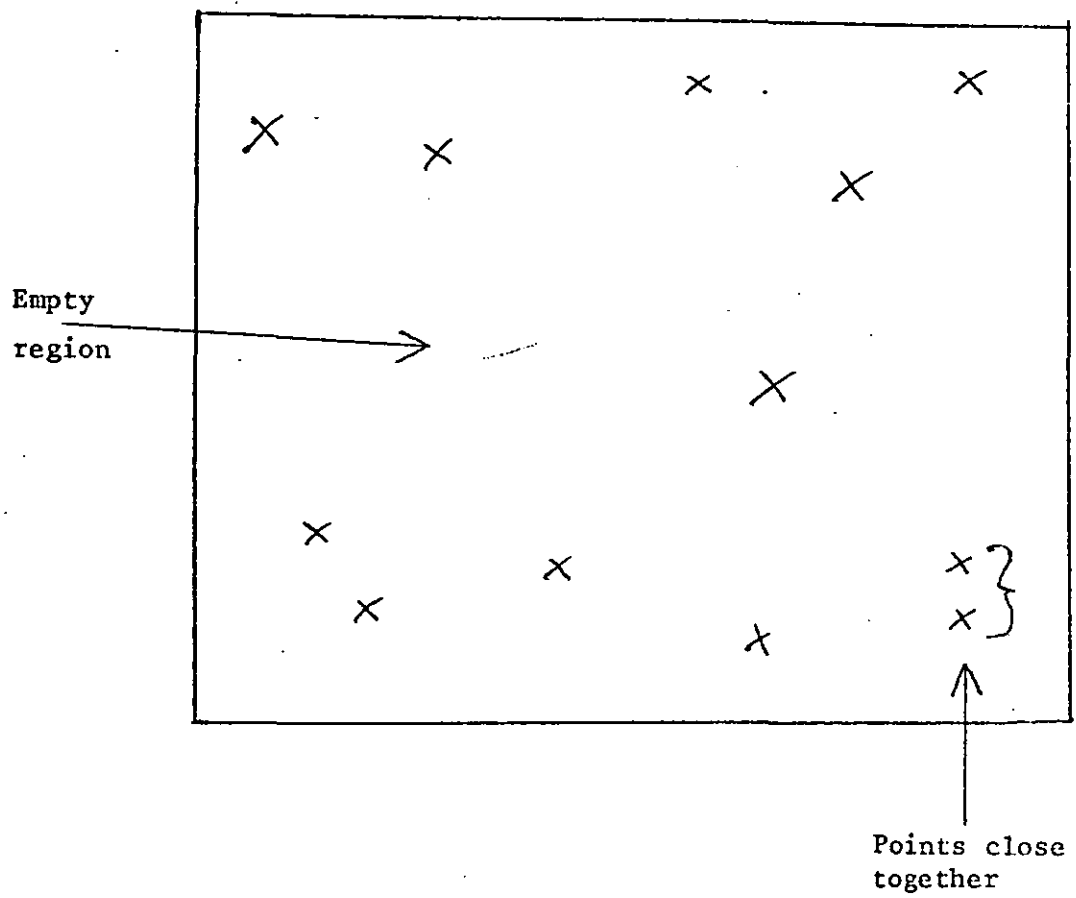


FIGURE 3.6: Disadvantages of random positioning of data points

CHAPTER 4

INTERPOLATION IN TWO DIMENSIONS AND CONTOURING

#### 4.1 CONTOURING PROBLEMS AND ALGORITHMS

The problem of drawing contour lines to represent the behaviour of a two-dimensional function over a plane region may be classified into three sub-problems, depending on the way in which the function is presented.

- a) Algebraic - the algebraic form of the function is given, and it may be computed at any arbitrary point.
- b) Gridded - function values are only given at the nodes of a grid (usually rectangular) which spans the area of interest.
- c) Scattered - function values are only given at a finite set of points, distributed in an arbitrary fashion over the area of interest.

Case b) is thus a special case of c), but in practice most contouring algorithms are based on the assumption of gridded data (see for example Sutcliffe, 1976). Case a) can be easily converted to gridded form by evaluating the algebraic function at the grid nodes, although this naturally leads to loss of definition in the spaces between nodes. However, with a sufficiently fine grid acceptable results can be obtained.

Conversion of case c) to gridded form poses more problems. A means is required to interpolate from the scattered points to the nodes of the grid. Various methods have been used for this (see for example McLain, 1976 and Sabin, 1978), but problems can be encountered in this, especially when the data points are not evenly scattered. In practice it has been found that sets of data points which leave large "blank" areas

can lead to totally meaningless contour lines being produced in these areas.

Even if the interpolation process is reasonable, the fact that the contour lines are based on the grid rather than on the actual data points may lead to inconsistencies between the contour lines and the original data. Figure 4.1 shows an illustration of this kind of error. One solution to this problem might be to use an irregular grid tailored to the data points. If  $N$  points were given, a grid of at most  $N \times N$  nodes would be needed, as in Figure 4.2, to guarantee that each data point coincided with a grid node.

For these reasons it was felt to be better not to base the contouring algorithm on the assumption of gridded data, but to contour directly from an algebraic function. If scattered data is given, then the use of an interpolating function as defined in equation 2.30 leads directly to the production of contour lines without using any intermediate grid system. The stochastic interpolating function has the added advantage of an easily computable derivative, which will be shown to be useful in defining the contour lines.

An algorithm of this type, designed to handle the most difficult case of scattered data, includes the other cases within its scope. Data presented in gridded form is merely a special case of arbitrarily scattered data.

#### 4.2 TRACKING CONTOUR LINES

The aim is to draw a set of contour lines  $f(x,y)=c_i$ ,  $i=1,\dots,n$  for the interpolating function  $f()$  based on the  $N$  data points  $\{(x_1,y_1)\dots(x_N,y_N)\}$  with values  $\{z_1,\dots,z_N\}$ , without using any kind of superimposed grid system. A method of tracking contours needs to fulfil the following requirements:

1. It must ensure that all the contour segments appropriate to the given set of data are drawn.
2. It must define a starting point for the drawing of each such contour segment.
3. It must decide when to terminate a contour segment, either because the starting point has been reached again, or because the area of interest has been left in both directions.

We shall assume that a rectangular border is defined for the area of interest, within which the contour lines are to be drawn. Values of the interpolating function  $f()$  are computed at the vertices of the border rectangle, and these are treated essentially as extra data points. Thus the data set consists of the "real" data points plus the "dummy" border points.

The system for keeping track of the contours works by means of a set of "reference points". Such a set is defined for each contour level  $c_i$ , and consists of a number of points where the interpolating function value exactly equals the contour level. The set is chosen so that at least one reference point lies on each "definable" contour segment within the area of interest.

A definable contour segment is one which divides the area of interest into two parts, each containing at least one data point or border point. It is possible for undefinable contour segments to exist, which enclose no data points and cannot be detected by this algorithm. Figure 4.3 shows such a segment. One way of detecting such segments would be by the introduction of internal dummy points with values given by the interpolating function.

Reference points are defined by drawing a set of straight lines joining data points. Each such line joins a data point with value greater than the contour level to a point with value less than the contour level. Border points are all connected to internal data points. A search is carried out along each line until a point is found with estimated function value equal to the contour level, and this becomes the reference point. Figure 4.3 illustrates this process.

The technique for carrying out the search for a given value along a straight line is basically a Newton's method solution of the equation  $f(x,y) - c_i = 0$  along the line. If the current point is at position  $l$  along the line from the starting point, then a change of linear position is given by

$$\Delta l = (c_i - f(x,y)) / \frac{\partial f}{\partial l} \quad (4.1)$$

If the line is at an angle  $\theta$  to the x-direction then

$$\frac{\partial f}{\partial l} = \frac{\partial f}{\partial x} \cos \theta + \frac{\partial f}{\partial y} \sin \theta, \quad (4.2)$$

where the derivatives of  $f()$  are given by equation 2.35.

This process is repeated until  $|f(x,y) - c_i| < \epsilon$ , a prescribed tolerance.

The main problem that can arise is illustrated in Figure 4.4. The gradient at the starting point of the search may be opposite to the secant gradient joining the values at the two points A and B - this means that the search for the reference point would be initially conducted in the wrong direction, and the required point might never be found. This problem is overcome by successive sub-division of the interval AB until a sub-interval containing the reference point is found with the starting gradient in the correct direction.

Using the set of reference points generated in this way, the algorithm for drawing all the definable contour segments appropriate to a particular contour level works as follows:

1. The first reference point on the list is taken as the starting point of a new contour segment.
2. From the present point on the contour, a new point is computed (the algorithm for this is described later). This is repeated until

Either

3. If the contour segment has reached its starting point again, then the tracking of this segment is ended and it is drawn.

Or

4. If the new point is on or outside the boundary of the area of interest, this arm of the segment is ended. If the other arm has also been ended then the contour segment is drawn.

Otherwise, the tracking is begun again from the start point.

5. As a contour segment is being drawn, all the reference points which lie on that segment are deleted from the list.



6. If any reference points still exist on the list, a new segment is started from step 1.

The algorithm for generating a new point on the contour from the previous point operates in two stages:

1. A tangent is drawn to the contour at the current point and a point a distance  $\Delta r$  along it is chosen.  $\Delta r$  is the "step length" for generating the new point.
2. From this point a perpendicular is drawn to the tangent, and a search is carried out along this line until a function value  $f(x,y)$  is found which is within a specified tolerance of the desired contour level. If no such point is found (due to the contour forming a sharp bend in the neighbourhood), then the value of  $\Delta r$  is halved and the process repeated.

It is necessary to find the angle  $\theta$  which the tangent makes with the x-direction, and this can be done quite simply.

If  $\partial f/\partial r$  is the derivative of  $f()$  along the contour tangent, then

$$\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x} \cos \theta + \frac{\partial f}{\partial y} \sin \theta = 0, \quad (4.3)$$

therefore 
$$\tan \theta = - \frac{\partial f/\partial x}{\partial f/\partial y}, \quad (4.4)$$

where  $\frac{\partial f}{\partial x}$  and  $\frac{\partial f}{\partial y}$  may be computed by the interpolating function.

Figure 4.5 illustrates this procedure.

It is obviously worthwhile to pay some attention to the selection of  $\Delta r$  so as to use the minimum number of points to define a reasonably smooth contour. In areas where the contour is almost straight,  $\Delta r$  can be large. Where the contour is sharply curved,  $\Delta r$

should be much smaller. It is possible to estimate a desirable value for  $\Delta r$ , based on the second derivatives of the interpolating function.

$$\text{Let } s = -\tan \theta = \frac{\partial f / \partial f}{\partial x / \partial y}$$

$$\text{therefore } \frac{\partial s}{\partial x} = \frac{\partial^2 f / \partial f}{\partial x^2 \partial y} - \frac{\partial f}{\partial x} \cdot \frac{\partial^2 f}{\partial x \partial y} / \left(\frac{\partial f}{\partial y}\right)^2$$

$$\text{and } \frac{\partial s}{\partial y} = \frac{\partial^2 f / \partial f}{\partial x \partial y \partial y} - \frac{\partial f}{\partial x} \cdot \frac{\partial^2 f}{\partial y^2} / \left(\frac{\partial f}{\partial y}\right)^2, \quad (4.5)$$

$$\text{Now } \cos \theta = \frac{\partial f / \partial y}{\sqrt{\left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial x}\right)^2}} = \gamma \frac{\partial f}{\partial y}$$

$$\text{and } \sin \theta = -\gamma \frac{\partial f}{\partial x}$$

$$\text{where } \gamma = \left[ \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \right]^{-1/2}. \quad (4.6)$$

Along the contour,

$$\begin{aligned} \frac{\partial s}{\partial r} &= \frac{\partial s}{\partial x} \cos \theta + \frac{\partial s}{\partial y} \sin \theta \\ &= \gamma \frac{\partial^2 f}{\partial x^2} - 2\gamma \left(\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y}\right) \frac{\partial^2 f}{\partial x \partial y} + \gamma \left(\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y}\right)^2 \frac{\partial^2 f}{\partial y^2} \end{aligned} \quad (4.7)$$

$$\text{Since } s = -\tan \theta$$

$$\frac{\partial s}{\partial r} = -\sec^2 \theta \frac{\partial \theta}{\partial r}$$

$$\text{Therefore } \frac{\partial \theta}{\partial r} = -\gamma^2 \left(\frac{\partial f}{\partial y}\right)^2 \frac{\partial s}{\partial r}$$

$$\begin{aligned} &= -\gamma^3 \left(\frac{\partial f}{\partial y}\right)^2 \frac{\partial^2 f}{\partial x^2} + 2\gamma^3 \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial y} \cdot \frac{\partial^2 f}{\partial x \partial y} \\ &\quad - \gamma^3 \left(\frac{\partial f}{\partial x}\right)^2 \frac{\partial^2 f}{\partial y^2}. \end{aligned} \quad (4.8)$$

If we specify a required change in direction  $\Delta \theta$ , then we can relate the step length  $\Delta r$  to this by

$$\Delta r = \Delta \theta / \left| \frac{\partial \theta}{\partial r} \right|. \quad (4.9)$$

#### 4.3 LONG-RANGE TREND AND THE TWO STAGE MODEL

It is often felt that the function being contoured consists of more than one component. In particular it may be assumed to consist of a long-range "trend" with a more short-range "residual" component superimposed on top. Often (as in universal kriging) this trend is modelled by some algebraic function, such as a polynomial, but this has nothing to recommend it unless there is a good reason to suppose that the trend takes such a form. Within the context of modelling functions as realisations of stationary stochastic processes, it is felt to be more natural to allow the trend to be another stationary stochastic process, so that the "two stage" model is

$$Z(\underline{x}) = Z_L(\underline{x}) + Z_S(\underline{x}) , \quad (4.10)$$

where  $Z_L(\underline{x})$  is a stationary, normally distributed, random process of mean  $\mu$  and correlation distance  $\rho_L$ , and  $Z_S(\underline{x})$  is a similar process with mean 0 and correlation distance  $\rho_S$ , and

$$\rho_L \gg \rho_S .$$

Three parameters ( $\mu$ ,  $\rho_L$  and  $\rho_S$ ) are needed to fit this model to the data, and this can be carried out in various ways. One technique which has been used to fit the model to scattered data for contouring is to cluster the  $N$  data points into  $n$  ( $\ll N$ ) clusters. The average value  $z_i^A$  and centroid  $(x_i^A, y_i^A)$  of each cluster  $i$  is calculated. Values of  $\mu$  and  $\rho_L$  are fitted to these cluster average points, and this gives a model for the long-range trend  $Z_L(\underline{x})$ . For each of the  $N$  original points a residual error value is found.

$$z_j^E = z_j - \hat{z}_j , \quad (4.11)$$

where

$z_j$  is the actual value at point  $j$   
 and  $\hat{z}_j$  is the trend value at point  $j$ , estimated from the  $n$   
 cluster average points.

The value of  $\rho_s$  is fitted to these  $N$  residual values, giving a complete model. The interpolating function at any point  $(x,y)$

$$f(x,y) = \text{Estimated trend from } n \text{ average points using } \mu \\
\text{and } \rho_L \\
+ \text{Estimated residual from } N \text{ data points using } \mu_s.$$

The estimation procedure outlined here is bound to suffer from the problems described in Section 3.1 when fitting "trend" and correlated "residual" functions to data - those associated with the circular nature of whatever technique is used. The number of clusters ( $n$ ) used will obviously influence and constrain the long-range correlation distance fitted. Errors in the estimation of  $\rho_L$  will inevitably affect the estimation of  $\rho_s$ . Thus in adding to the complexity of the model, we are increasing the difficulty of obtaining accurate estimates of all the parameters.

As a check on the feasibility of estimating the parameters of such a "two-stage" model, some simulation experiments have been carried out (See Appendix B). Although both  $\rho_L$  and  $\rho_s$  are under-estimated, it is significant that in general the ratio  $\rho_L/\rho_s$  is approximately correct (especially for  $N=40$  rather than  $N=20$ ). Thus, although there may be errors in the parameter estimation, the structure of the two-stage model is being correctly reflected.

#### 4.4 RESULTS WITH TEST DATA

A program SIMP (Stochastic Interpolation and Modelling Program) has been written in ALGOL-68R to implement the contouring concepts in this section, as well as containing other features. It has been tested on various sets of data, mainly related to genuine examples of information available only at scattered points. A brief description of the program and its operation is given in Appendix A.

As a comparison with a "conventional" contouring program, the CALCOMP package GPCP (General Purpose Contouring Program) has been used to produce a contour map of permeability based on measurements at 72 oil wells in a Russian oilfield - data obtained from Schvidler (1964). This data is listed in Table 4.1. These points are not scattered evenly across the area of interest, but tend to leave large empty regions. The results of GPCP are shown in Figure 4.6. From this it is obvious that a number of features produced by GPCP are purely imaginary - in particular the large "cliff" in the southern part of the map where there are no data points. Also, in several places the contour lines are not entirely consistent with the data points, because of the gridding introduced by the program.

By comparison, Figure 4.7 shows the same data contoured by SIMP (no trend assumed). The value of grand mean ( $\mu=299.5$ ) is the median of the data, and the correlation distance ( $\rho=0.5076$ ) was estimated by the "pair-point" method. The interpolating function is "flat", with little or no structure, in areas where there is no data, and only shows significant variability close to the data points. Subjectively, this would seem to give a better representation of the (necessarily incomplete) data available than the GPCP map.

An additional feature which was a simple matter to include in the program was that of being able to draw cross-sections of the interpolating function along any specified line. Two such sections are shown in Figures 4.8 and 4.9 for the Shkapovskii data - they illustrate graphically the high degree of variability in this data.

As an example of the two stage model including a long-range trend, height contours from the Ordnance Survey map for the Charnwood Forest area to the south-west of Loughborough were used. These original contours are shown in Figure 4.10. 40 points were scattered at random on this map, and heights in metres above sea level at these points were input to the program. The value of grand mean ( $\mu=60$ ) was input as a subjective estimate, based on the knowledge that the ground continues to slope downhill to the north-east. The parameters of the "two-stage" model ( $\rho_L=2.39$  and  $\rho_S=0.5$ ) were estimated using the procedure described in the previous section. The data is given in Table 4.2, and the contours produced are shown in Figures 4.11 and 4.12. Figure 4.11 is a contour map of the long-range trend only, and Figure 4.12 is a full map including the short-range residuals. The final result is not dissimilar to the actual topography of the area, given the limited amount of data used. Figure 4.13 is a cross-section from south-west to north-east of the area.

As a further example of the two stage model, data provided by a colleague was used. This relates to erosion of a microscopic irridium projection. 115 data points were provided (see Table 4.3) and a value of grand mean ( $\mu=32$ ) was input, because it was known that the object was a single projection on an otherwise flat surface. A "two-stage" model was fitted, as before, with the long-range correlation ( $\rho_L=42.53$ ) being

given by the estimation procedure, but the short-range ( $\rho_s=10.0$ ) being equal to a pre-set minimum. A trend was fitted which is shown in Figure 4.14. In many ways this map may be more informative than the full map including the residuals and fitting all the data exactly, which is shown in Figure 4.15. Cross-sections are shown in Figures 4.16 and 4.17.

Finally, a set of data (see Table 4.35) was invented which was designed as a "difficult case" for contouring programs. Figure 4.18 illustrates this data set - it is intended to represent a circular "hole" with very steep sides. Three different programs have been tested on this data - GPCP, SIMP and the GINO-F library routine GINOSURF.

GPCP was run using two different grids - a 20x20 grid and a 100x100 grid. Results for the former are shown in Figure 4.19, from which it is apparent that the contour lines do not fit the data values very closely. The outer set of points have values of 100, but from the map they appear to have values between 70 and 100. Similarly, the inner set have nominal values of 10, but from the map apparent values between 35 and 40. The results with the 100x100 grid are more satisfactory in terms of fitting the actual data, although the overall map has a rather strange "rosette" shape and lacks the expected symmetry (see Figure 4.20).

The results from SIMP are shown in Figure 4.21. The fit of the contours to the data is good, and a symmetrical, circular shape is achieved. Values of  $\mu$  and  $\rho$  equal to 100 and 1.0 respectively were used to achieve this plot.

The results from the GINOSURF package are shown in Figure 4.22. These are quite good, although the "gridded" nature of the contouring algorithm used is probably apparent. There is still a certain amount of discrepancy between data points and contour lines.

It would seem from this set of tests that the kind of contouring algorithm used in SIMP shows up well in comparison to the "gridded" algorithms used in other programs. It produces contours which are smooth, symmetrical and fit the data.

In producing contour maps from data using SIMP, the user has the option of allowing the program to estimate all the parameters to be used, or of supplying some of them himself. As can be seen from some of the above examples, it is often the case that specifying one of the parameters reflects a subjective knowledge about the surface to be contoured which is not explicit in the data points. For example, if it is known that the surface to be contoured is a "mound" or "hollow" in an otherwise flat "plain", then it is wise to set  $\mu$  equal to the surrounding, flat, value. On the other hand, it appears that the estimation procedures sometimes under-estimate the value of the correlation distance (see Section 2.4), so that it is valuable to be able to over-rule their judgement and supply an increased value.



#### 4.5 ONE-DIMENSIONAL APPLICATIONS

Before progressing to applications of stochastic interpolation in more than two dimensions, it is worth briefly considering the one-dimensional case. Some work has been done (see Brodlie, 1978) on the problem of fitting a curve of the form  $y=f(x)$  to a set of data points  $\{(x_1, y_1), \dots, (x_N, y_N)\}$ . A variety of methods are available, mostly based on some form of spline interpolation.

The cross-section option in the program SIMP will obviously produce plots of one-dimensional stochastic interpolating functions. Two sets of test data were used to illustrate the results of the program applied to one-dimensional data. The details of the test data are given in Table 4.4.

Figure 4.23 shows the results from SIMP applied to the first set of data. The curve fits the data reasonably well, although it could be criticised for being somewhat oscillatory. The mean ( $\mu=3.5$ ) was estimated by the program, but the correlation distance ( $\rho=1.0$ ) was fixed. Fig. 2.4 shows other examples of the same data with different values of  $\rho$ .

The second set of test data is "better-behaved" than the first, and this is shown in Figure 4.24 which gives the curve produced by SIMP. The mean ( $\mu=0$ ) was fixed so that the curve approached zero to the right, and the correlation distance ( $\rho=1.17$ ) was fitted by the program.

Thus stochastic interpolation is easily capable of application to functions of a single variable - whether the results are acceptable depends on subjective preconceptions about the likely shape of the underlying functions.

TABLE 4.1

Shkapovskii Oil Deposit Data

(x,y - Co-ordinates in arbitrary units

z - Permeabilities in millidarcies)

i	x <sub>i</sub>	y <sub>i</sub>	z <sub>i</sub>	i	x <sub>i</sub>	y <sub>i</sub>	z <sub>i</sub>	i	x <sub>i</sub>	y <sub>i</sub>	z <sub>i</sub>
1	5.9	5.35	304	25	2.6	4.75	255	49	9.2	6.95	406
2	7.5	0.85	360	26	2.15	4.5	608	50	10.5	5.95	64
3	7.55	4.45	418	27	2.25	5.9	346	51	10.75	5.6	360
4	4.0	6.45	415	28	2.3	6.35	575	52	10.6	3.0	361
5	8.2	7.1	400	29	2.75	7.3	197	53	6.75	4.55	343
6	2.0	5.4	269	30	2.0	7.3	224	54	5.7	4.75	276
7	8.3	5.6	198	31	3.1	7.45	174	55	3.35	5.55	196
8	10.05	5.9	70	32	2.7	7.85	364	56	3.9	8.65	254
9	8.75	4.05	668	33	4.25	8.25	271	57	4.65	8.7	263
10	1.7	2.75	480	34	5.1	8.35	48	58	7.15	4.15	321
11	0.95	3.65	66	35	7.4	7.35	295	59	9.3	1.9	385
12	10.55	4.45	273	36	6.65	8.2	65	60	8.9	1.95	642
13	1.05	3.3	88	37	7.75	7.1	450	61	8.55	2.1	241
14	10.65	2.6	175	38	7.05	8.0	238	62	8.1	2.25	315
15	10.65	2.2	220	39	8.45	6.8	430	63	2.85	5.5	346
16	9.6	0.8	232	40	7.4	7.85	183	64	8.15	6.4	376
17	9.6	1.3	255	41	8.8	6.65	248	65	8.8	6.15	314
18	8.85	1.5	396	42	7.75	7.65	620	66	9.8	5.3	70
19	8.4	1.6	341	43	9.1	6.5	153	67	10.15	3.5	310
20	8.0	1.75	200	44	9.35	6.15	116	68	2.85	3.2	458
21	7.7	1.4	372	45	8.5	7.25	107	69	4.45	7.45	335
22	2.85	3.55	580	46	9.7	5.95	106	70	4.0	7.95	289
23	2.55	3.85	542	47	8.85	7.1	207	71	8.7	2.25	313
24	2.3	4.15	346	48	10.35	5.55	59	72	9.65	2.05	510

Model fitted:Grand mean  $\mu = 299.5$ Correlation distance  $\rho = 0.5076$

TABLE 4.2

Charnwood Data

(Heights in metres above sea level)

$i$	$x_i$	$y_i$	$z_i$	$i$	$x_i$	$y_i$	$z_i$
1	1.0	2.9	218	21	5.3	8.5	76
2	9.0	7.0	65	22	3.4	1.3	178
3	0.2	6.5	150	23	7.7	3.6	85
4	6.1	8.4	78	24	0.6	6.9	153
5	6.0	3.8	122	25	4.8	5.0	130
6	1.7	2.0	218	26	5.8	8.3	79
7	6.9	9.9	65	27	8.7	3.8	74
8	8.1	9.4	61	28	5.9	4.9	108
9	5.9	3.6	128	29	3.6	4.7	169
10	6.9	5.7	79	30	3.3	3.1	187
11	4.7	1.1	144	31	9.6	2.4	76
12	1.4	8.0	115	32	0.4	3.6	194
13	9.0	2.3	76	33	4.2	2.4	153
14	2.9	1.7	200	34	6.3	7.3	75
15	4.3	3.3	172	35	8.7	3.6	75
16	1.6	5.2	221	36	7.4	3.8	90
17	3.0	7.5	140	37	4.8	9.3	74
18	4.8	8.1	83	38	4.2	5.2	144
19	2.8	6.9	146	39	6.2	3.0	122
20	8.6	6.5	70	40	7.9	9.2	63

Model fitted:Grand mean  $\mu = 60$  metresLong-range correlation distance  $\rho_L = 2.39$ Short-range correlation distance  $\rho_s = 0.5$

TABLE 4.3

Symmetric Iridium Tip Data

i	x <sub>i</sub>	y <sub>i</sub>	z <sub>i</sub>	i	x <sub>i</sub>	y <sub>i</sub>	z <sub>i</sub>	i	x <sub>i</sub>	y <sub>i</sub>	z <sub>i</sub>
1	136.5	69.5	26.0	40	75.0	65.5	1.0	78	101.5	61.5	5.5
2	124.0	69.5	17.5	41	79.5	60.0	2.5	79	105.0	57.0	8.0
3	118.5	69.0	13.5	42	98.0	42.0	10.5	80	113.5	55.5	12.0
4	104.5	70.0	7.5	43	103.0	37.0	17.5	81	117.0	55.0	16.0
5	98.0	69.5	4.5	44	48.0	12.0	26.0	82	132.5	48.0	29.5
6	93.5	70.0	3.5	45	54.0	25.0	14.0	83	130.5	91.0	27.5
7	77.0	70.5	0.5	46	57.5	30.5	10.0	84	70.0	70.0	0.0
8	60.5	69.5	1.0	47	59.5	37.5	7.0	85	89.0	52.0	4.0
9	49.5	70.5	3.0	48	62.5	40.5	5.0	86	89.5	53.5	4.0
10	42.0	70.5	4.5	49	78.0	99.0	5.0	87	52.0	90.0	4.0
11	34.5	70.0	7.5	50	81.5	102.5	7.0	88	51.5	52.5	4.0
12	25.5	72.0	13.5	51	84.0	110.5	10.5	89	53.5	50.5	4.0
13	17.5	70.5	18.0	52	85.0	115.5	14.5	90	89.0	89.0	4.0
14	2.0	67.0	27.0	53	90.5	129.5	27.0	91	88.5	90.5	4.0
15	66.0	13.0	23.0	54	53.5	129.0	26.5	92	35.5	124.5	29.0
16	63.5	21.0	14.5	55	55.5	115.0	14.5	93	13.0	104.5	29.5
17	64.0	26.5	11.5	56	56.0	109.0	10.0	94	10.5	32.0	32.5
18	69.0	39.5	6.5	57	58.0	102.0	7.0	95	30.5	15.5	30.0
19	70.5	45.5	4.0	58	63.0	99.0	5.0	96	105.0	16.5	28.5
20	70.0	58.5	1.5	59	75.5	40.5	5.0	97	127.5	36.5	30.0
21	70.0	82.5	1.5	60	80.5	37.5	7.0	98	127.5	106.5	31.0
22	70.0	95.0	4.0	61	84.0	31.0	10.0	99	105.5	126.0	29.5
23	70.5	103.0	6.5	62	83.5	2.55	14.0	100	38.0	114.5	19.5
24	69.5	110.5	12.0	63	88.5	13.0	25.5	101	47.0	113.0	15.0
25	71.5	116.5	16.0	64	10.0	48.5	27.5	102	22.0	102.0	21.0
26	72.0	129.0	25.0	65	24.5	53.5	15.0	103	25.0	94.5	16.5
27	104.5	104.5	16.0	66	27.0	56.0	12.0	104	21.0	35.5	22.5
28	99.5	99.0	10.0	67	35.5	57.5	8.0	105	23.0	43.5	18.5
29	76.5	77.0	2.5	68	39.0	61.5	5.5	106	35.0	25.0	20.0
30	73.5	73.5	1.0	69	101.5	79.0	5.5	107	45.0	26.0	16.0
31	66.5	66.0	1.5	70	104.5	83.5	8.0	108	101.5	27.5	18.0
32	62.5	62.0	2.5	71	113.5	84.5	12.0	109	93.0	29.0	14.0
33	39.5	39.0	11.0	72	116.0	84.5	15.5	110	116.0	39.5	19.5
34	35.0	35.0	17.0	73	8.5	91.5	29.0	111	115.5	47.0	16.5
35	36.5	102.5	17.5	74	24.0	85.0	16.0	112	116.5	103.5	20.0
36	43.5	96.5	10.0	75	27.0	84.5	11.5	113	115.0	94.5	16.5
37	52.0	88.5	5.0	76	36.5	83.0	7.5	114	103.0	116.5	20.0
38	61.0	79.5	2.5	77	40.0	79.5	5.5	115	94.0	113.0	15.0
39	66.5	74.0	1.0								

Model fitted:Grand mean  $\mu = 32.0$ Long-range correlation distance  $\rho_L = 42.53$ Short-range correlation distance  $\rho_S = 10.0$

TABLE 4.35"Hole" Test Data for Contouring

$i$	$x_i$	$y_i$	$z_i$
1	2.9	5.0	100.0
2	3.1	5.0	10.0
3	3.5	3.5	100.0
4	3.7	3.7	10.0
5	5.0	2.9	100.0
6	5.0	3.1	10.0
7	6.5	3.5	100.0
8	6.3	3.7	10.0
9	6.9	5.0	10.0
10	7.1	5.0	100.0
11	6.3	6.3	10.0
12	6.5	6.5	100.0
13	5.0	6.9	10.0
14	5.0	7.1	100.0
15	3.7	6.3	10.0
16	3.5	6.5	100.0

TABLE 4.4

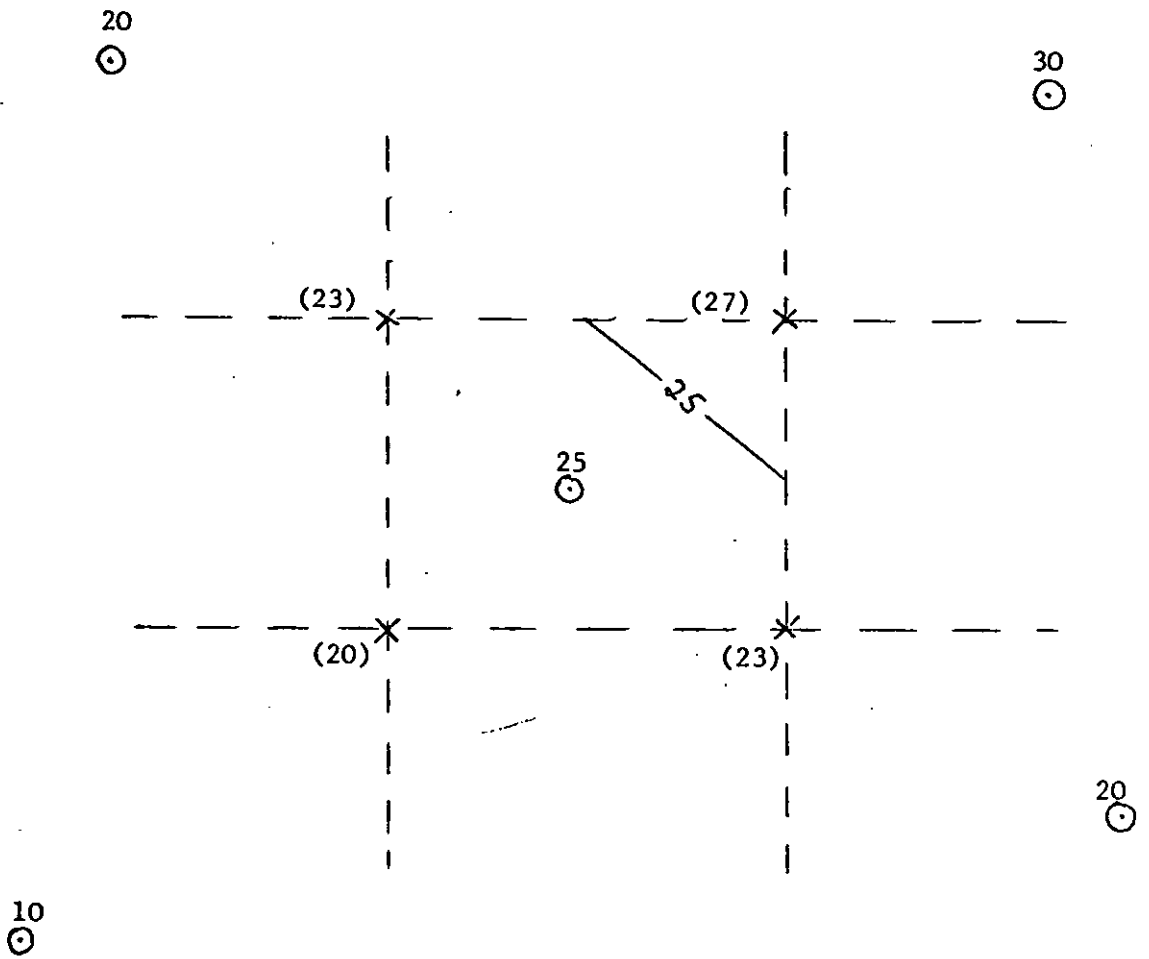
One-dimensional Test DataTest Data Set 1

$i$	$x_i$	$z_i$
1	2.0	1.0
2	4.0	4.0
3	6.0	9.0
4	8.0	10.0
5	9.0	2.0
6	13.0	3.0
7	14.0	11.0
8	18.0	3.0

Model fitted:Grand mean  $\mu = 3.5$  (fitted)Correlation distance  $\rho = 1.0$  (fixed)Test Data Set 2

$i$	$x_i$	$z_i$
1	0.0	15.0
2	1.0	13.0
3	2.0	10.0
4	3.0	6.0
5	4.0	4.0
6	5.0	3.0
7	6.0	3.0
8	8.0	2.0
9	10.0	3.0
10	12.0	1.0

Model fitted:Grand mean  $\mu = 0.0$  (fixed)Correlation distance  $\rho = 1.17$  (fitted)



KEY:

(23) × - Grid node with interpolated value

20 ○ - Given data point with value

—25— - Part of contour at level 25

FIGURE 4.1: Illustration of contouring errors using a gridded algorithm for scattered data.

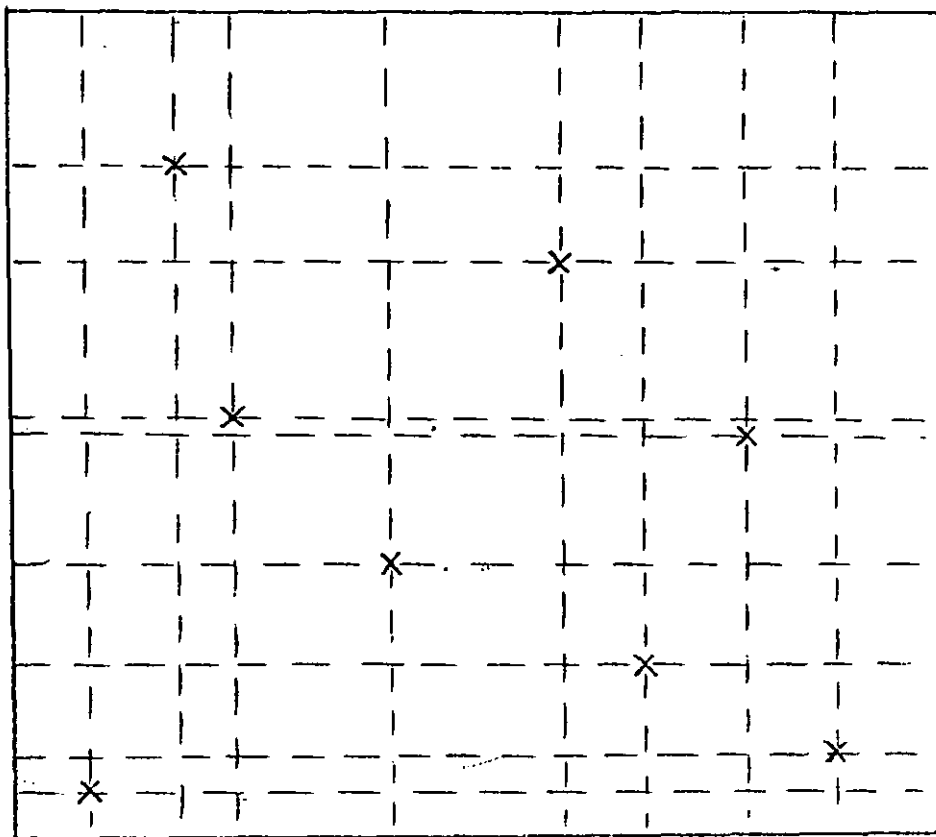
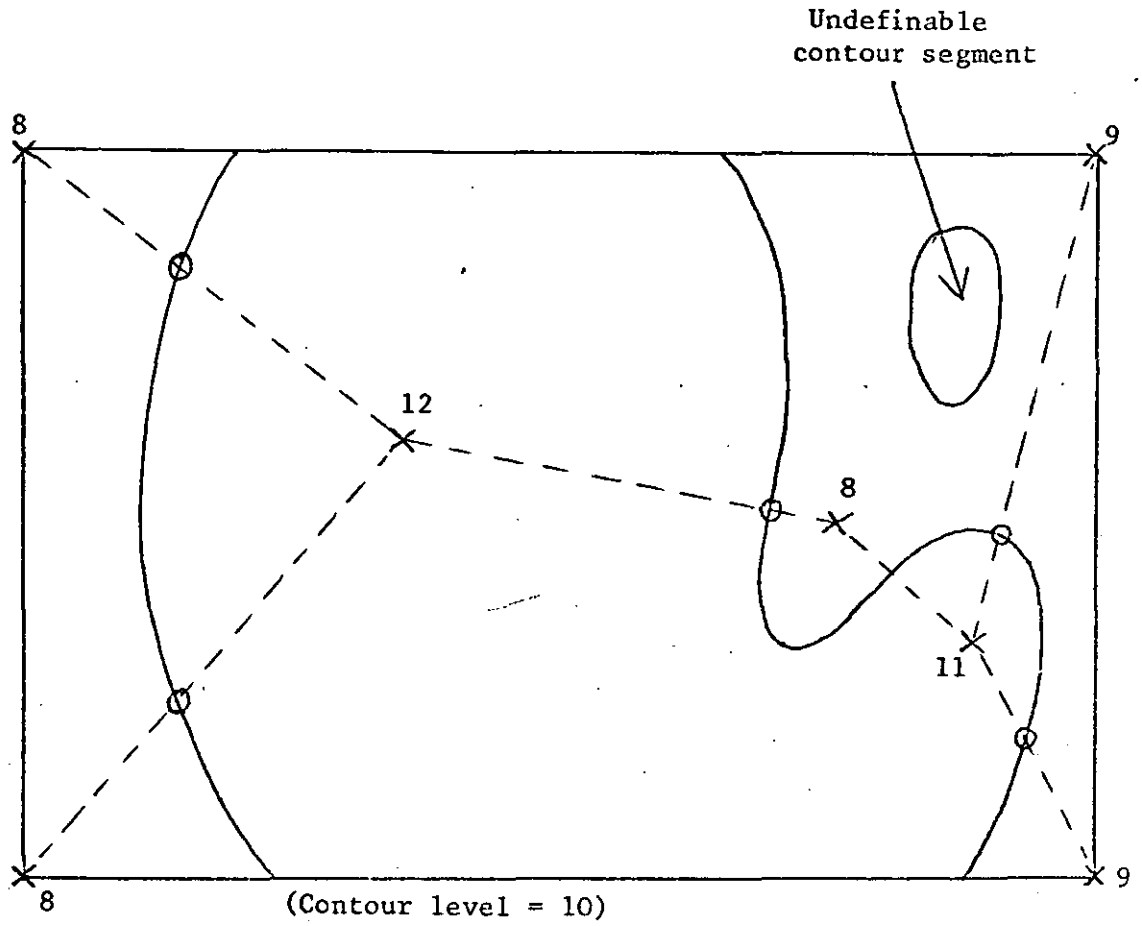


FIGURE 4.2: Use of irregular rectangular grid to fit scattered data points exactly





- x - Data point
- ⊙ - Reference point
- ~ - Contour segment
- - - - - Line of search for reference point

FIGURE 4.3: Use of reference points for tracking contours

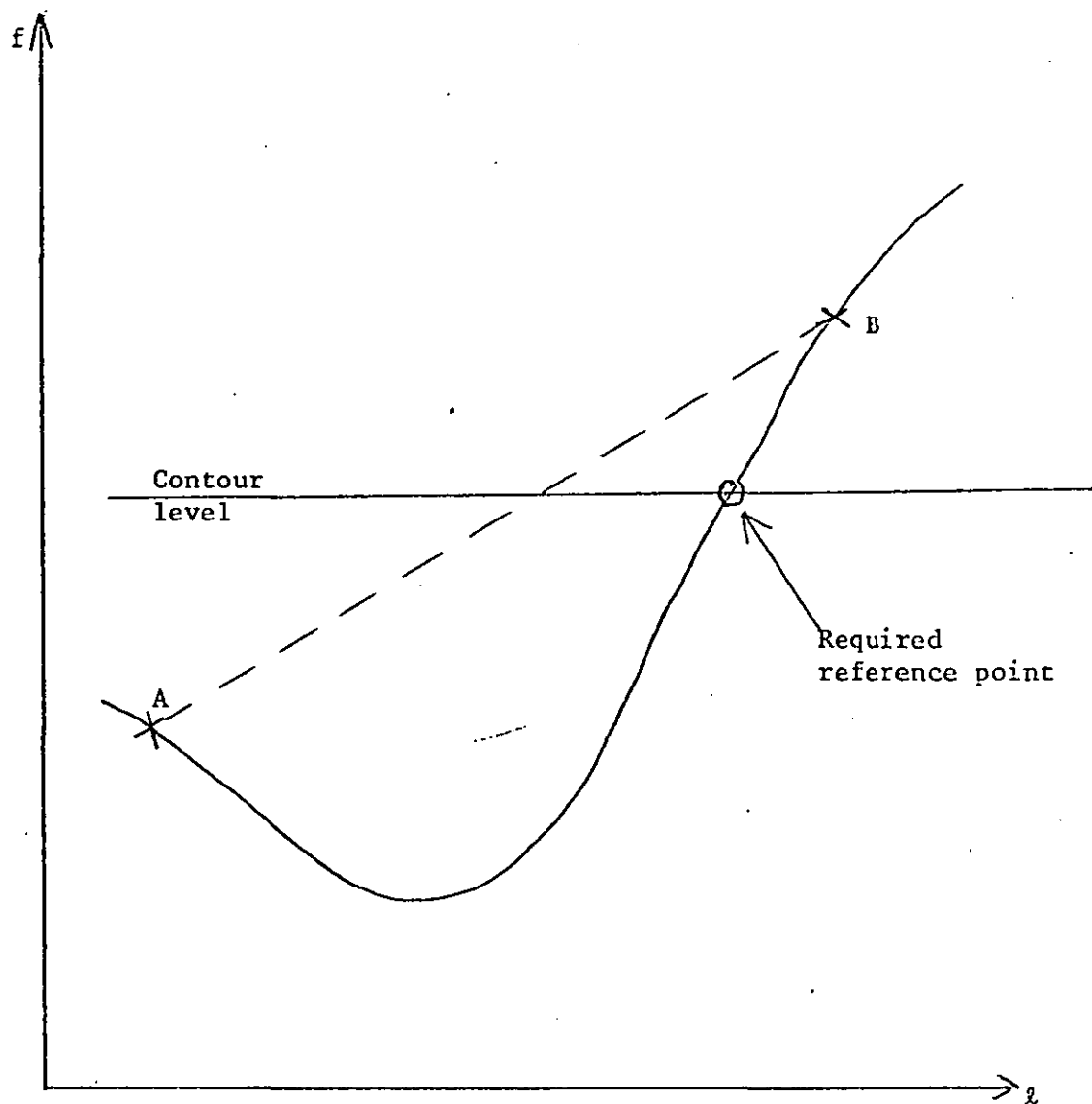


FIGURE 4.4: Searching for a reference point between two data points

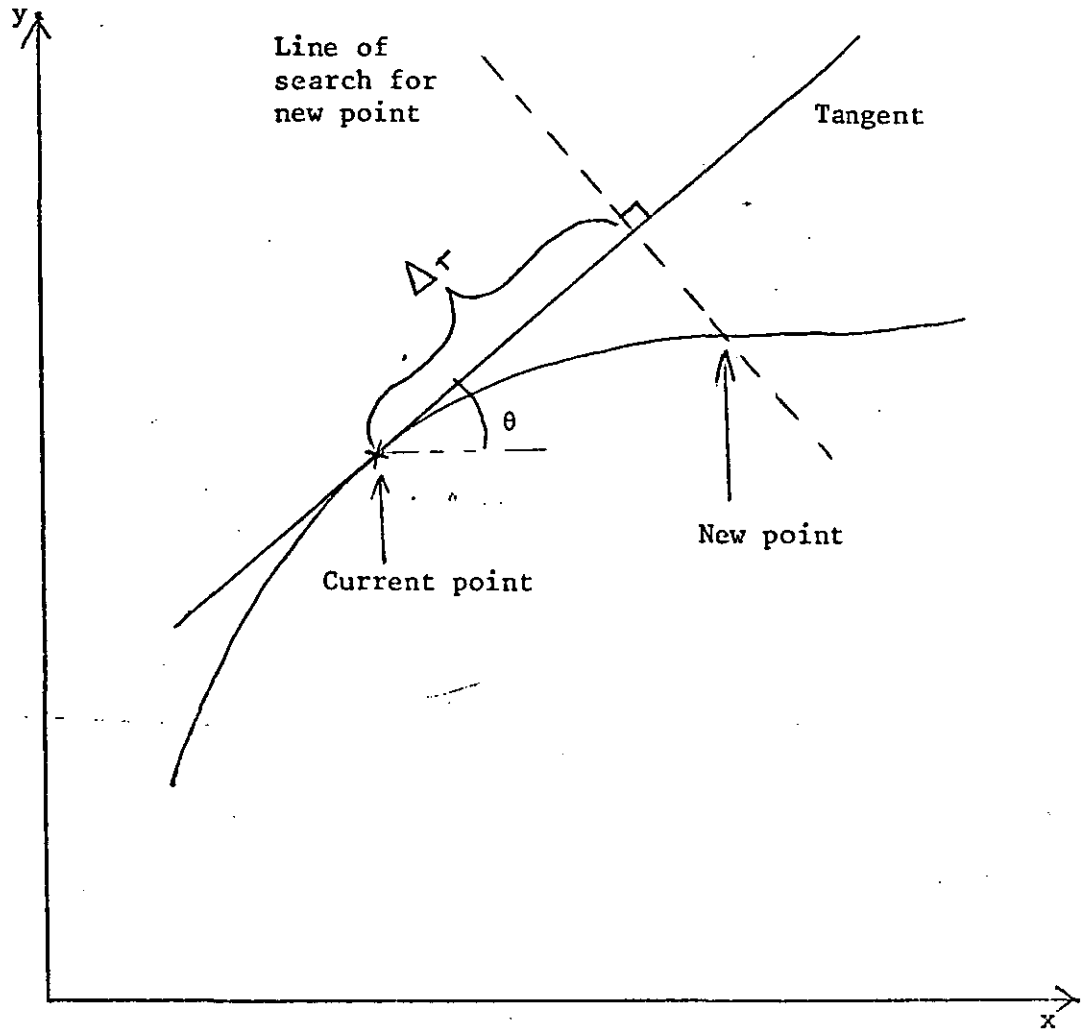


FIGURE 4.5: Finding a new point on the Contour

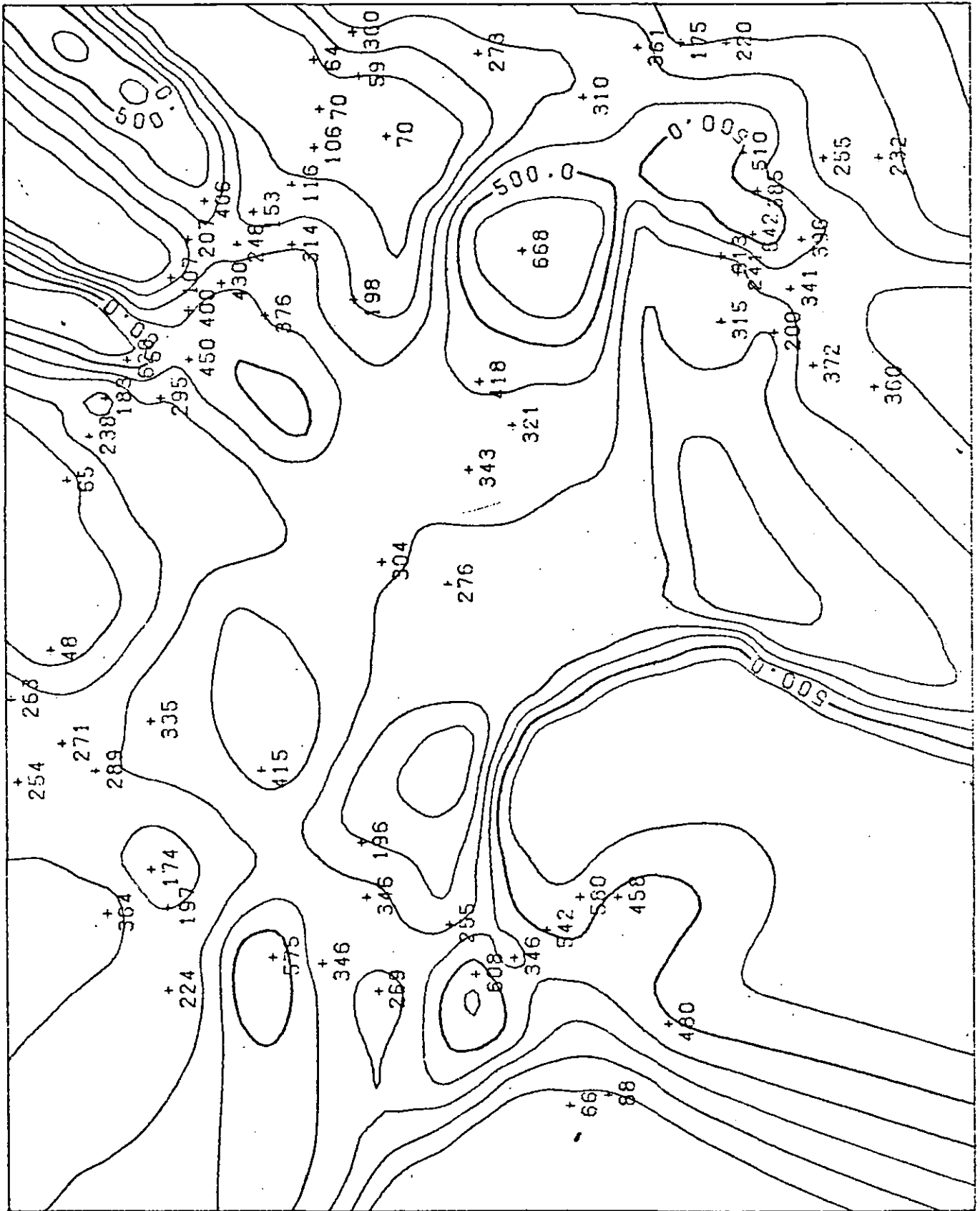


FIGURE 4.6: Shkapovskii oil deposit permeabilities contoured by GPCP

SHKAPOVSKII OIL DEPOSIT.

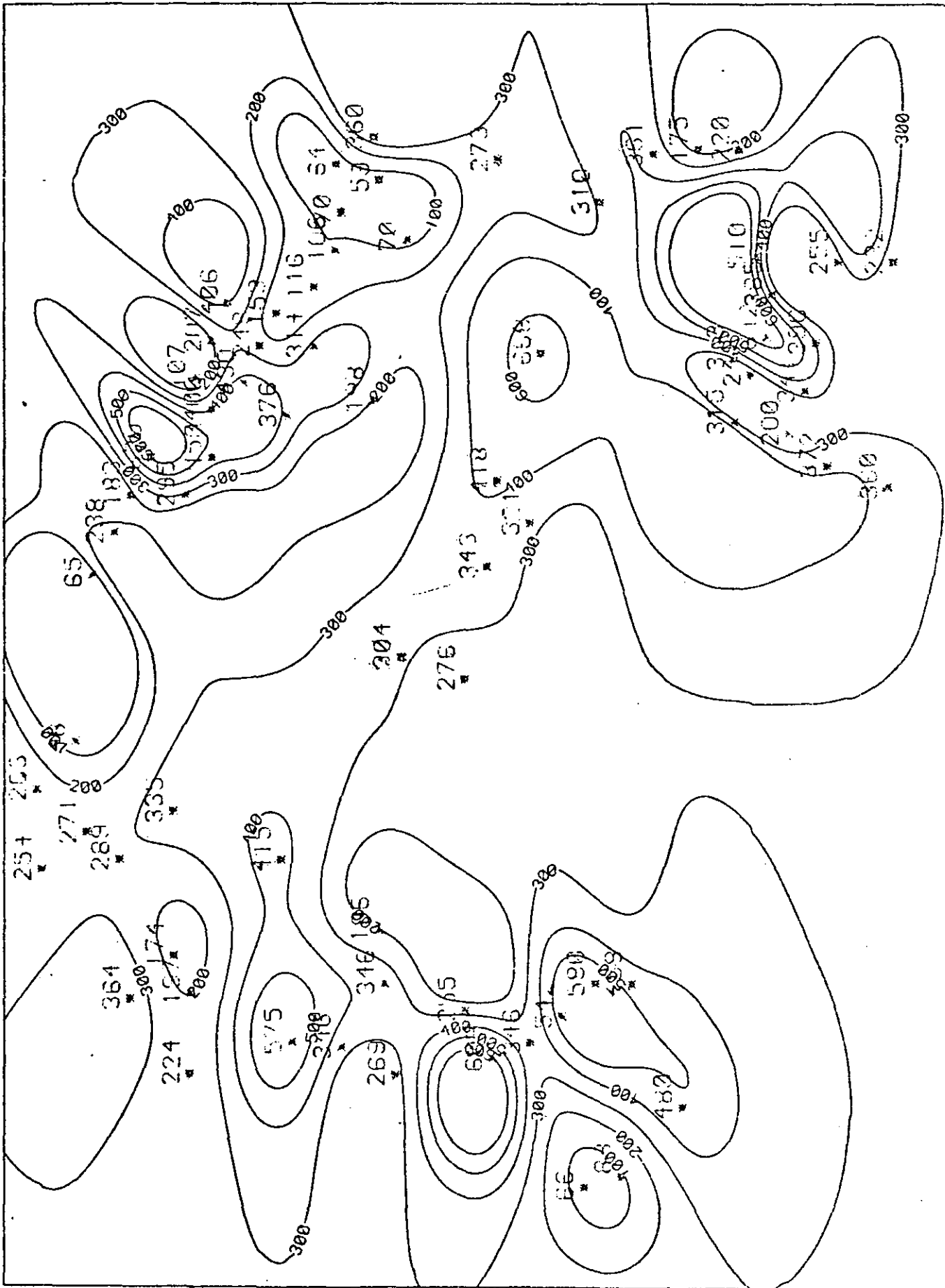


FIGURE 4.7: Shkapovskii oil deposit permeabilities contoured by SIMP

SHKAPOVSKII (BOTTOM LEFT TO TOP RIGHT).

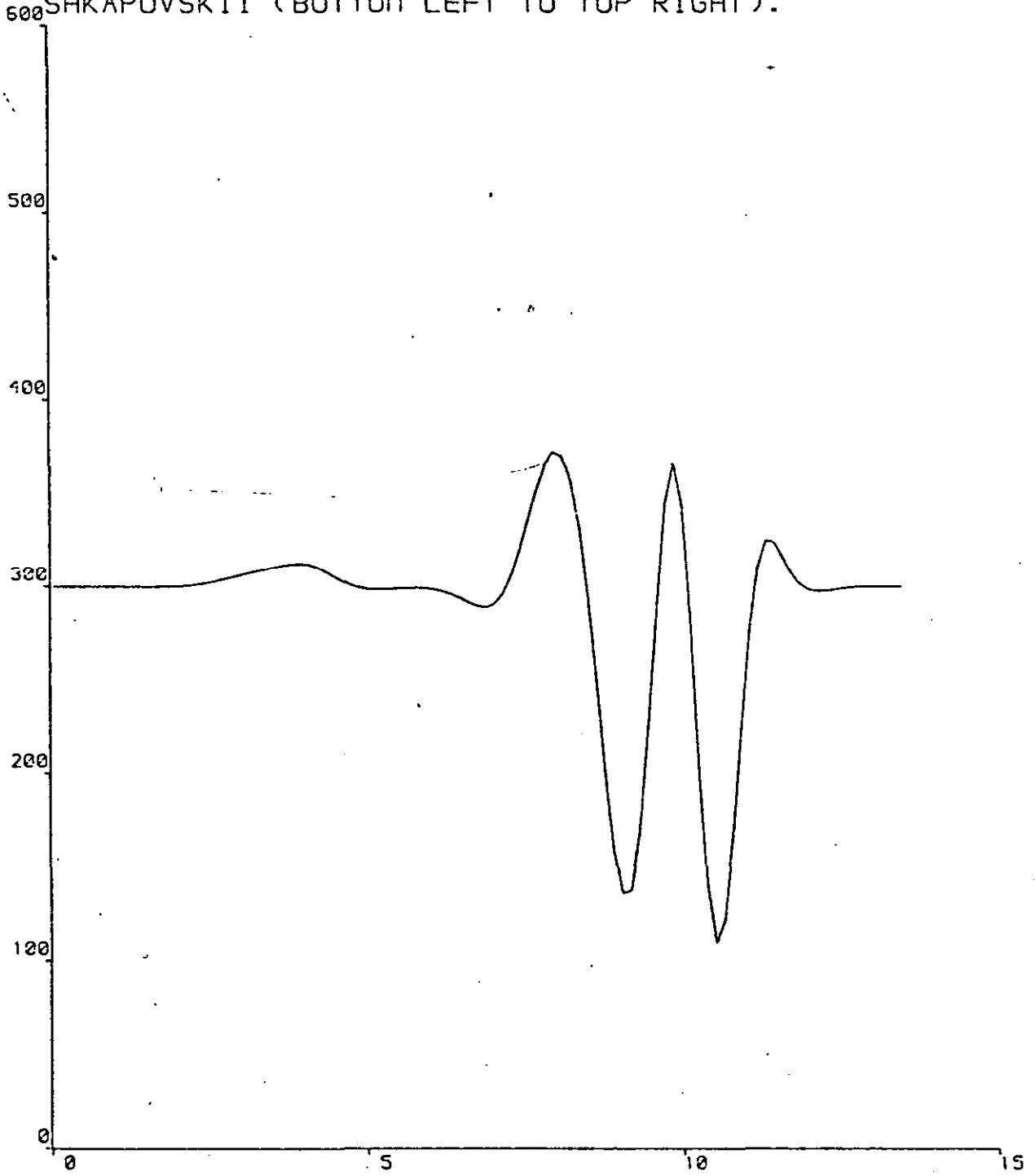


FIGURE 4.8: Shkapovskii data - Cross-section SW to NE

SHKAPOVSKII (TOP LEFT TO BOTTOM RIGHT).

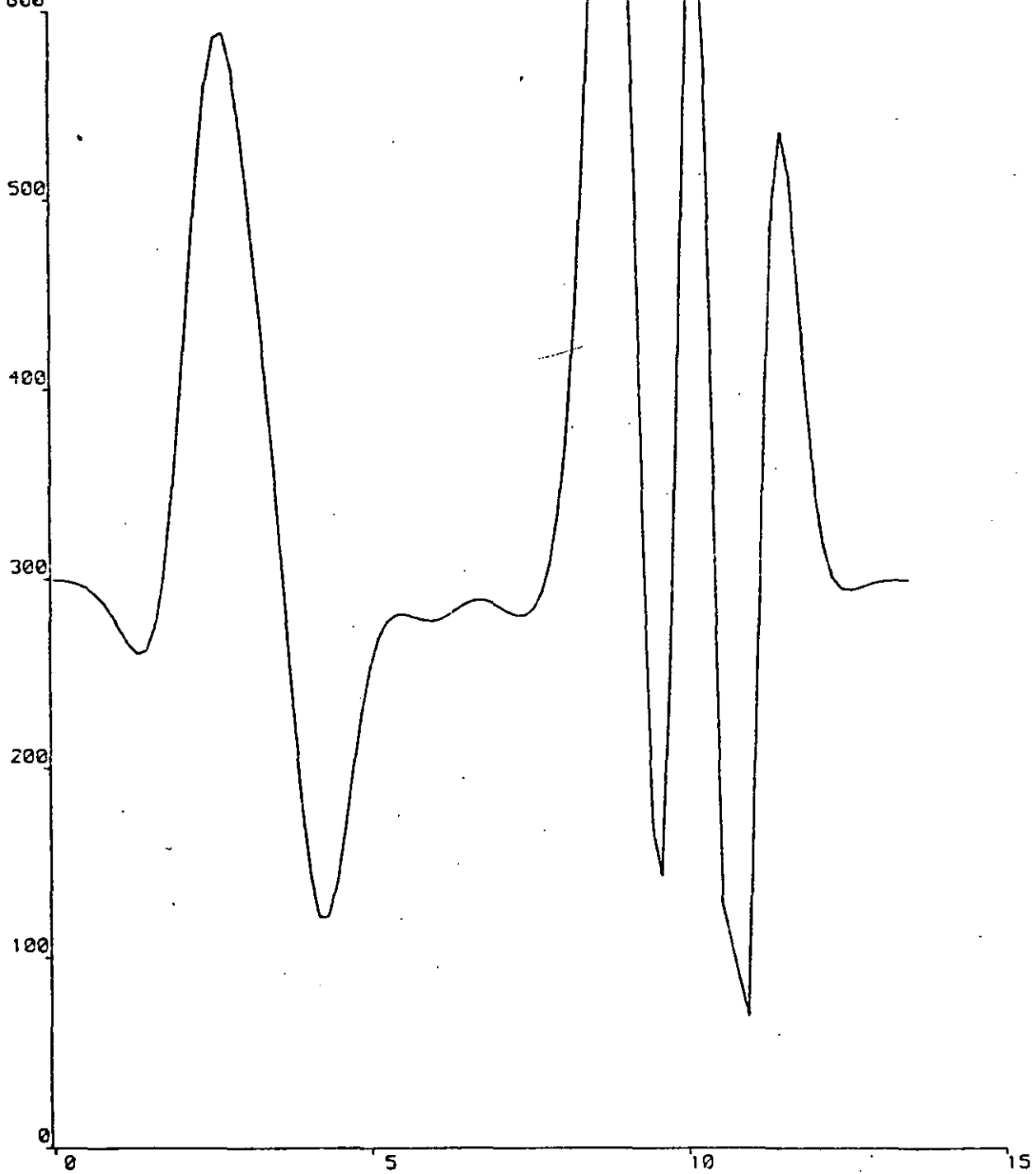
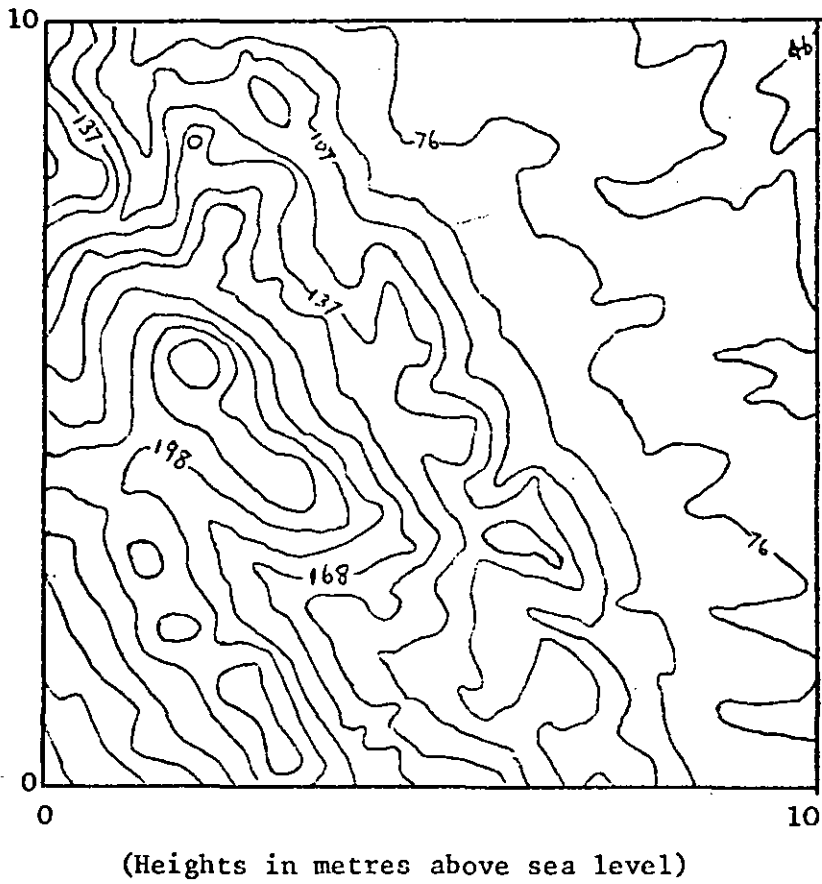


FIGURE 4.9: Shkapovskii data - Cross-section NW to SE



**FIGURE 4.10:** Original Charnwood contour map (from Ordnance Survey)



CHARNWOOD DATA (TREND).

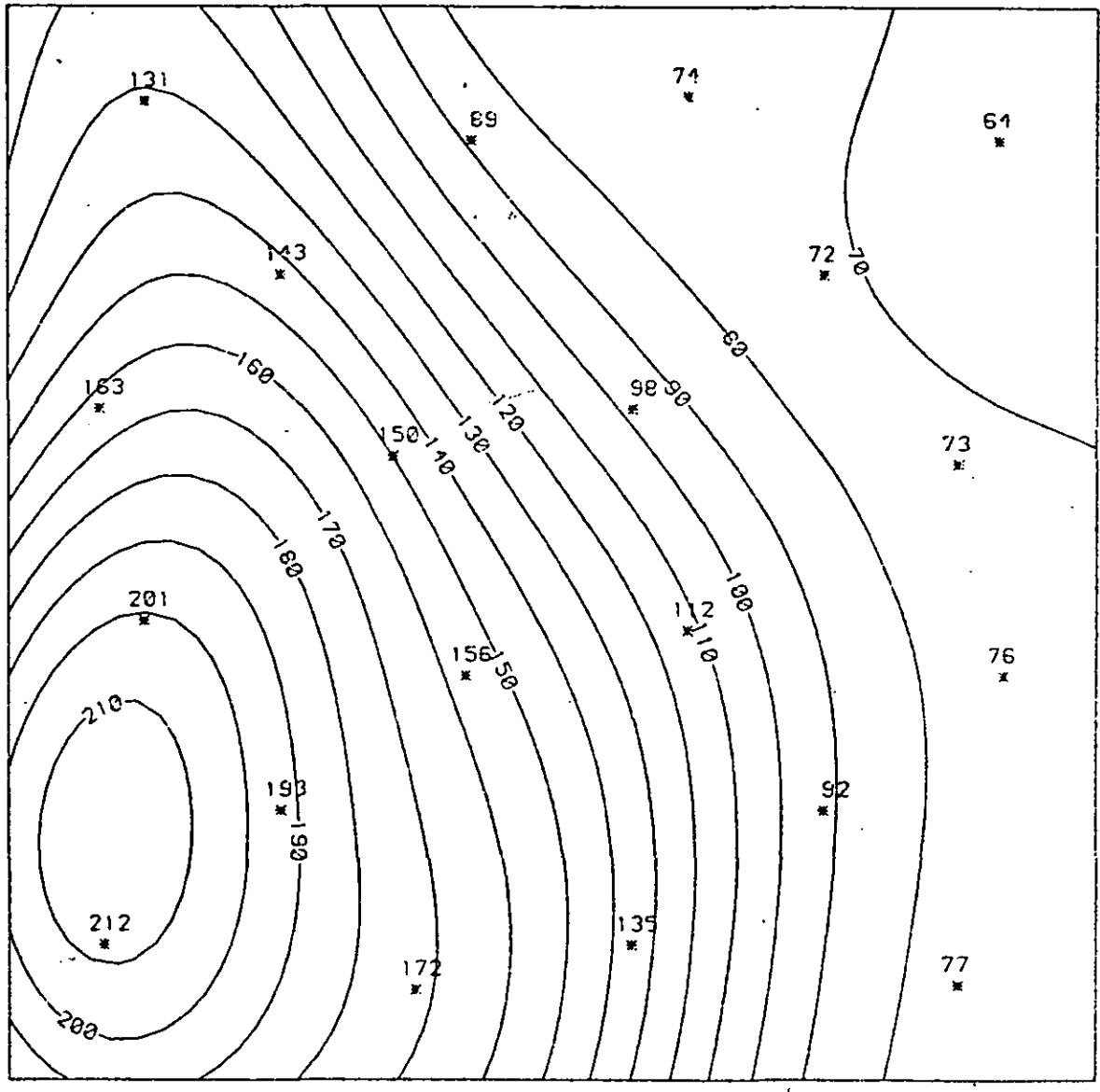


FIGURE 4.11: Trend fitted to Charnwood data, with average points

CHARNWOOD DATA.

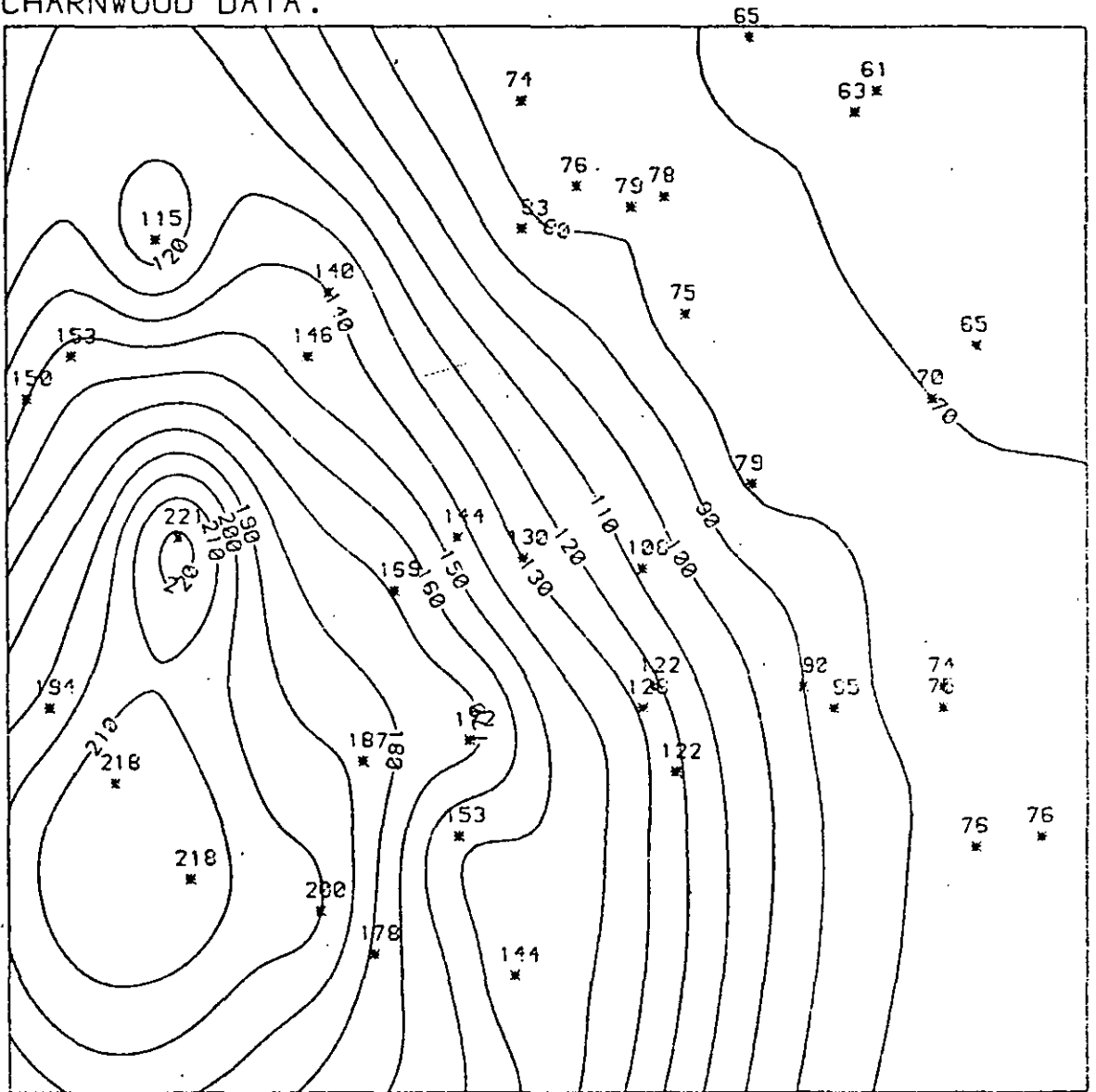


FIGURE 4.12: Trend plus residual map for Charnwood data, with data points

250 CHARNWOOD DATA (BOTTOM LEFT TO TOP RIGHT).

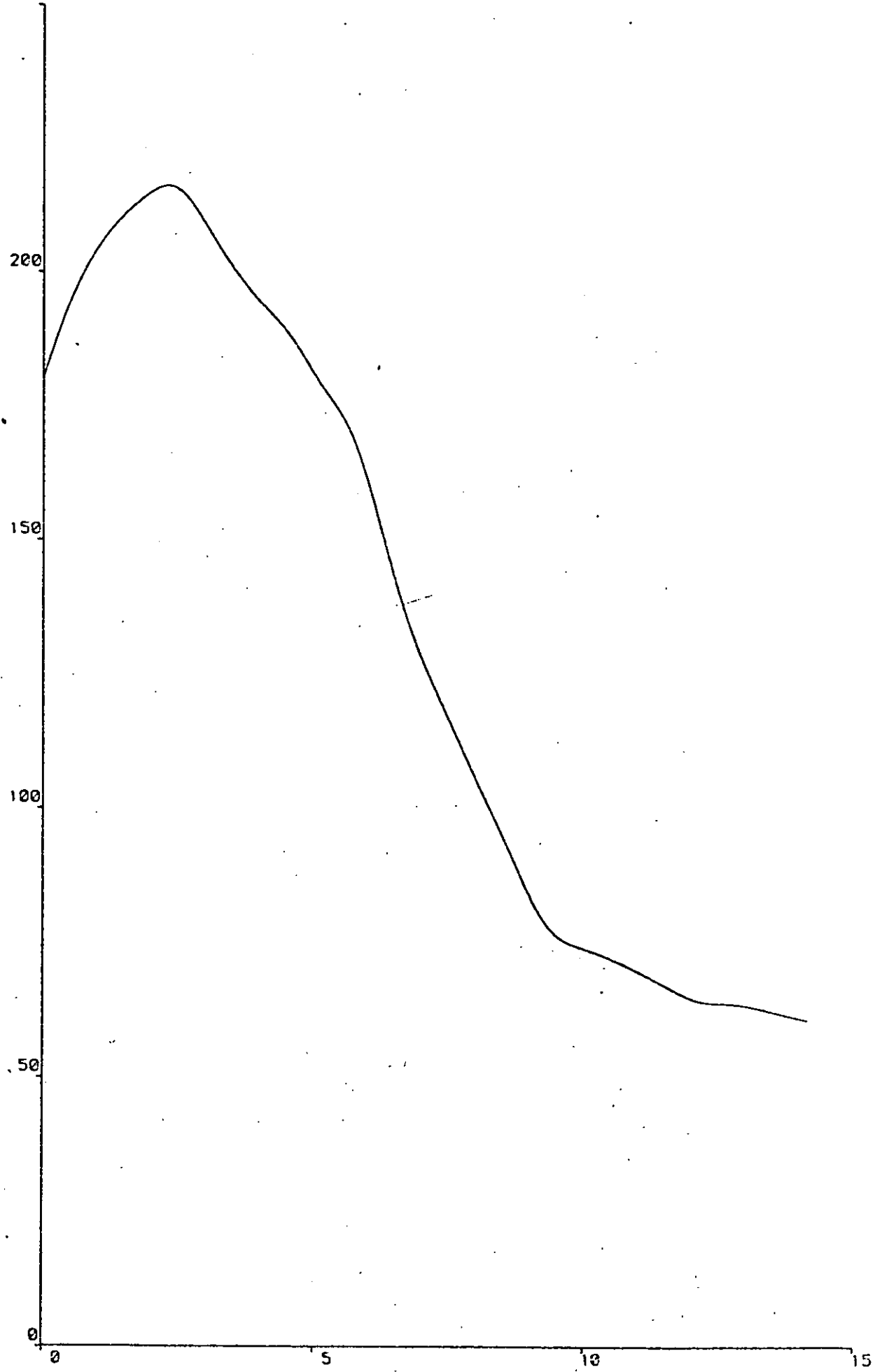


FIGURE 4.13: Charnwood data cross-section from SW to NE

## SYMMETRIC IRRIDIUM TIP DATA (TREND)

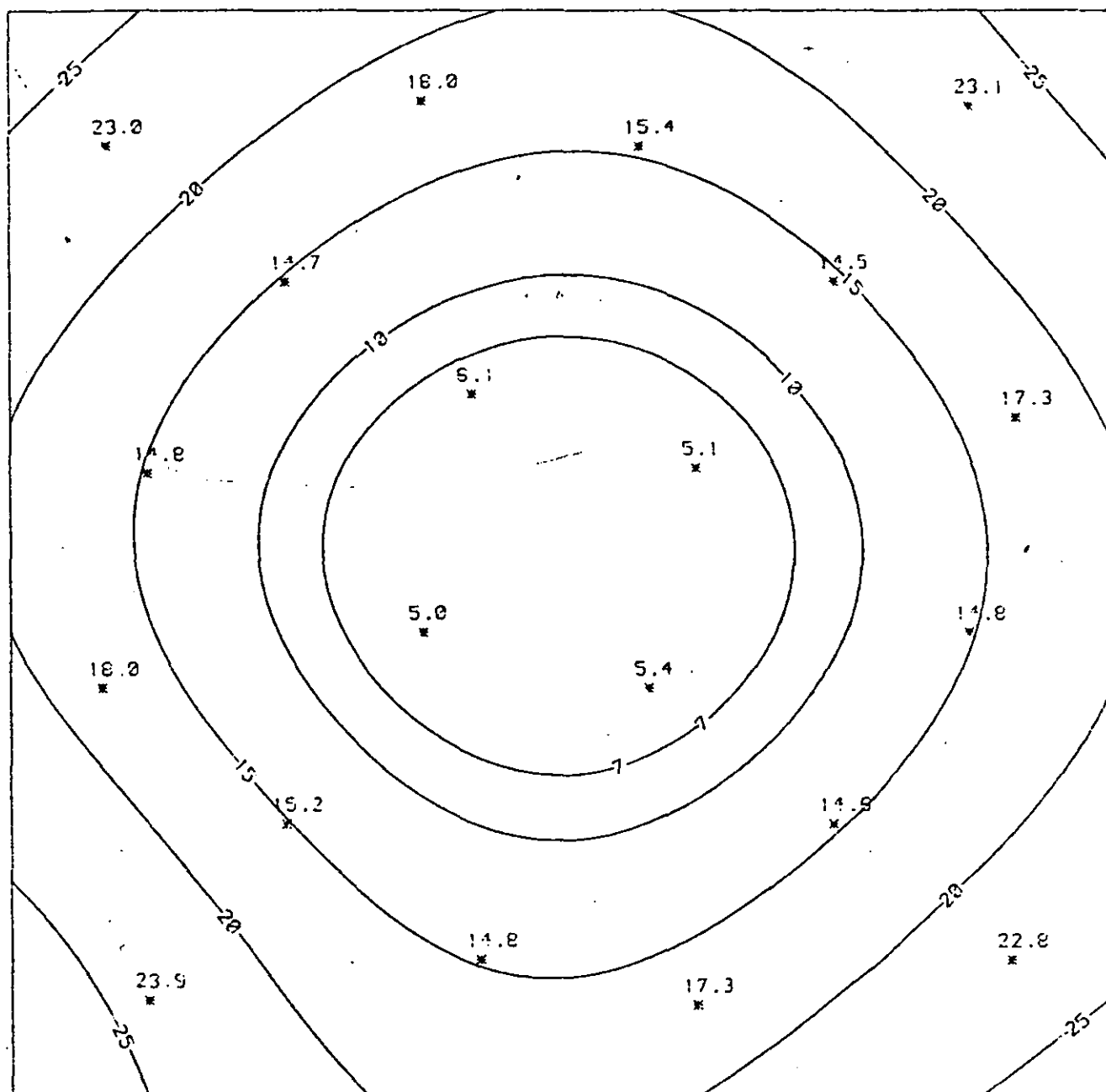


FIGURE 4.14

SYMMETRIC IRRIDIUM TIP DATA

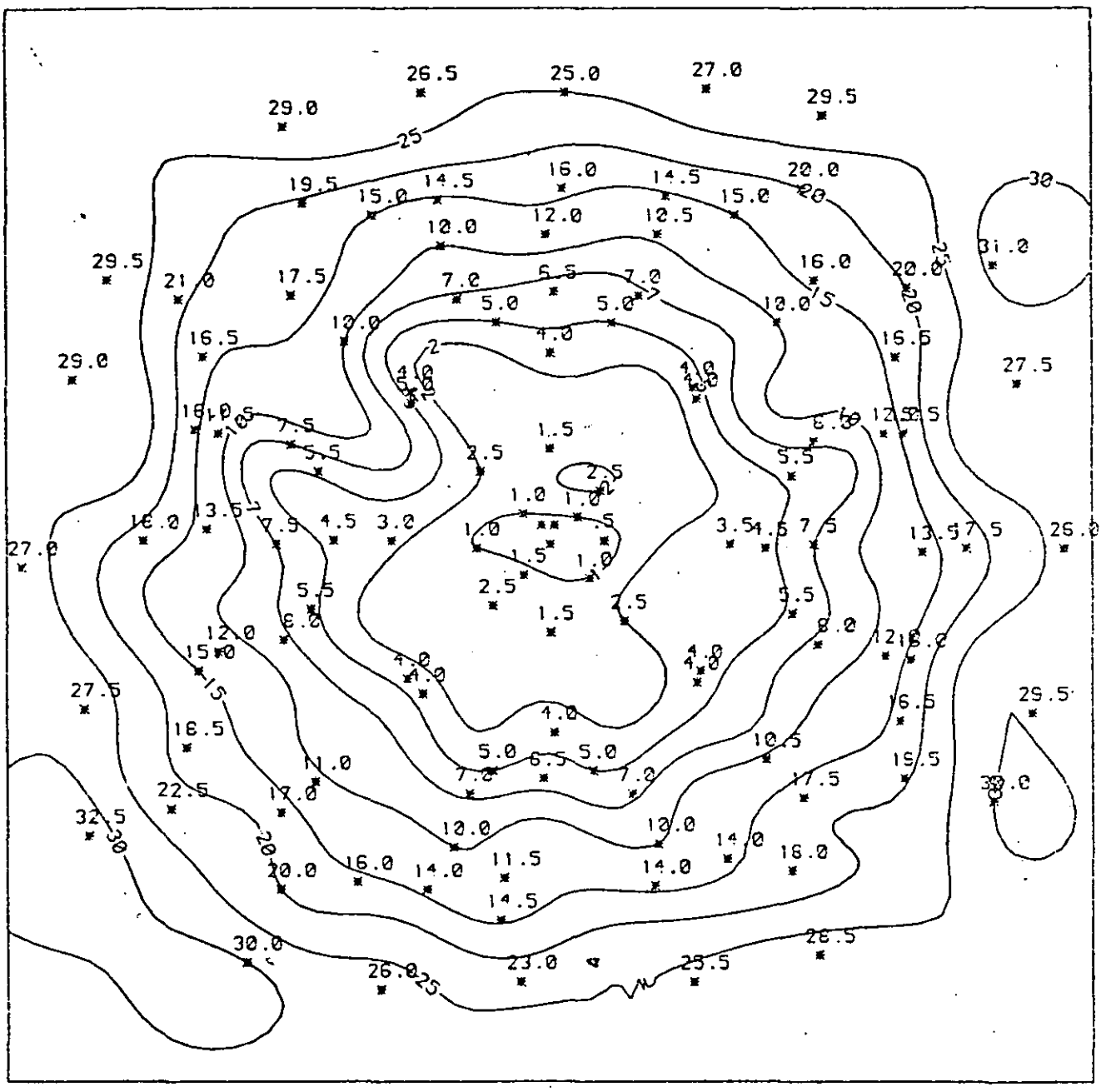


FIGURE 4.15

SYMMETRIC IRRIDIUM TIP (LEFT TO RIGHT).

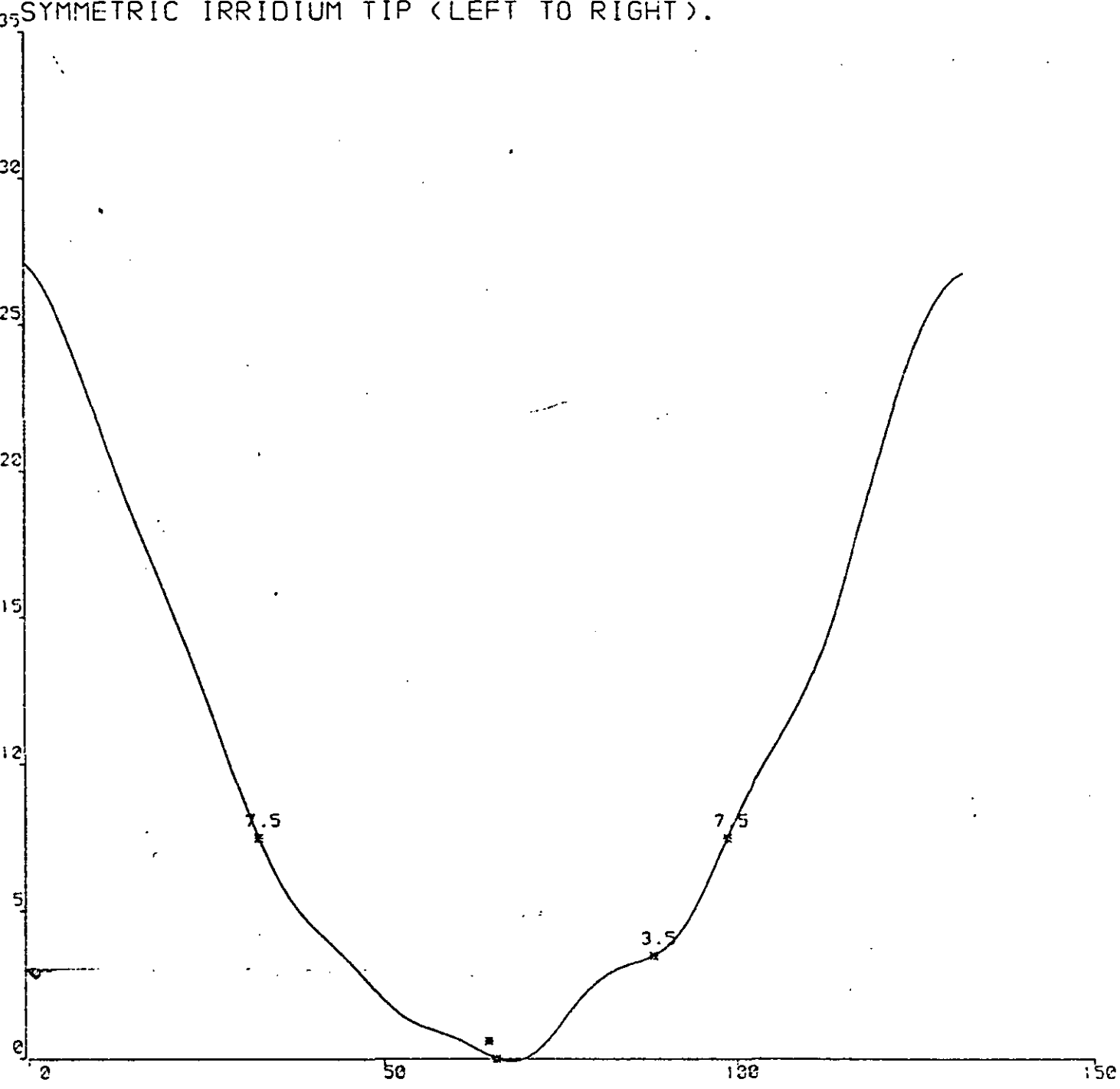


FIGURE 4.16

SYMMETRIC IRRIDIUM TIP (TOP TO BOTTOM).

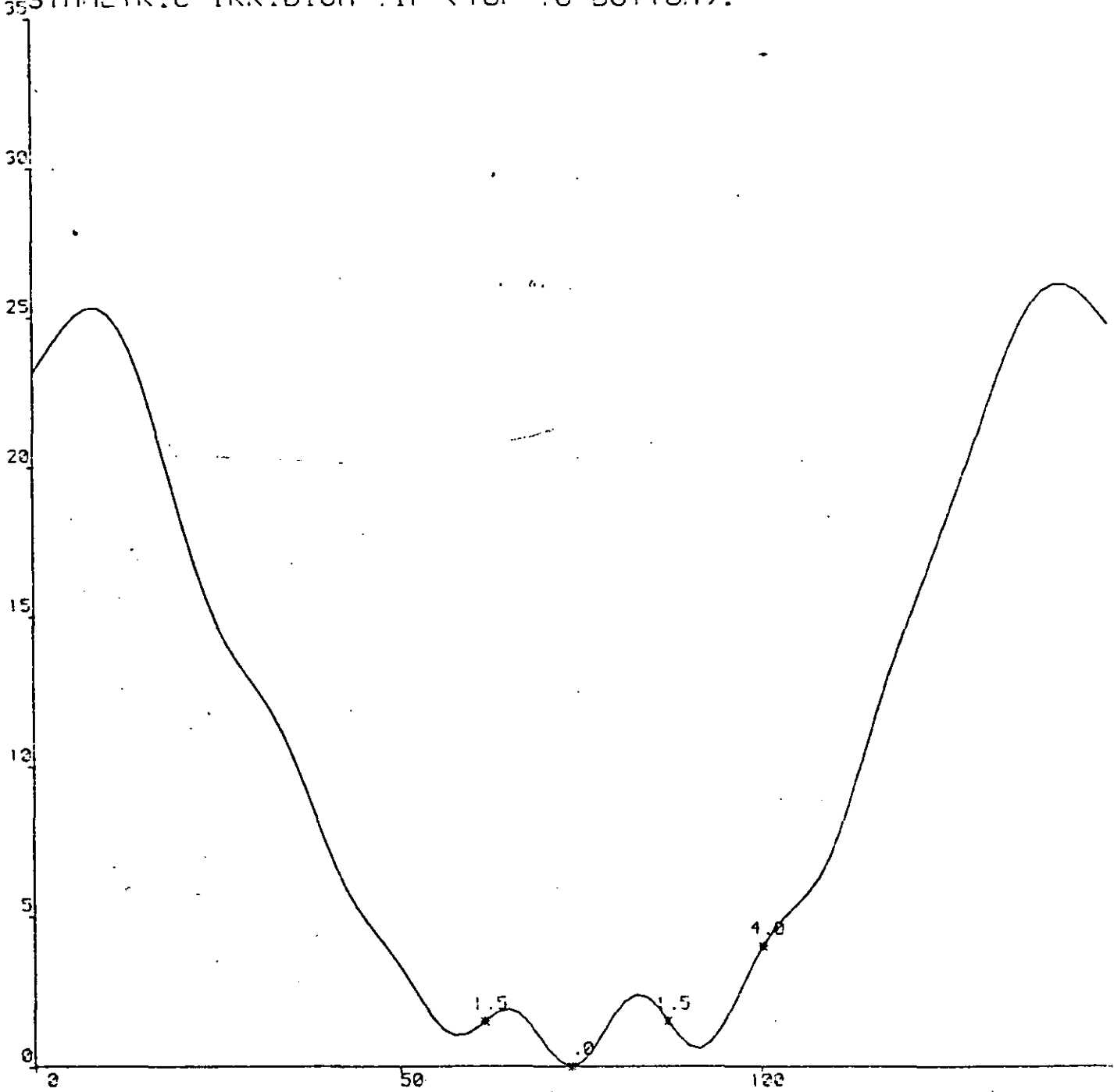


FIGURE 4.17

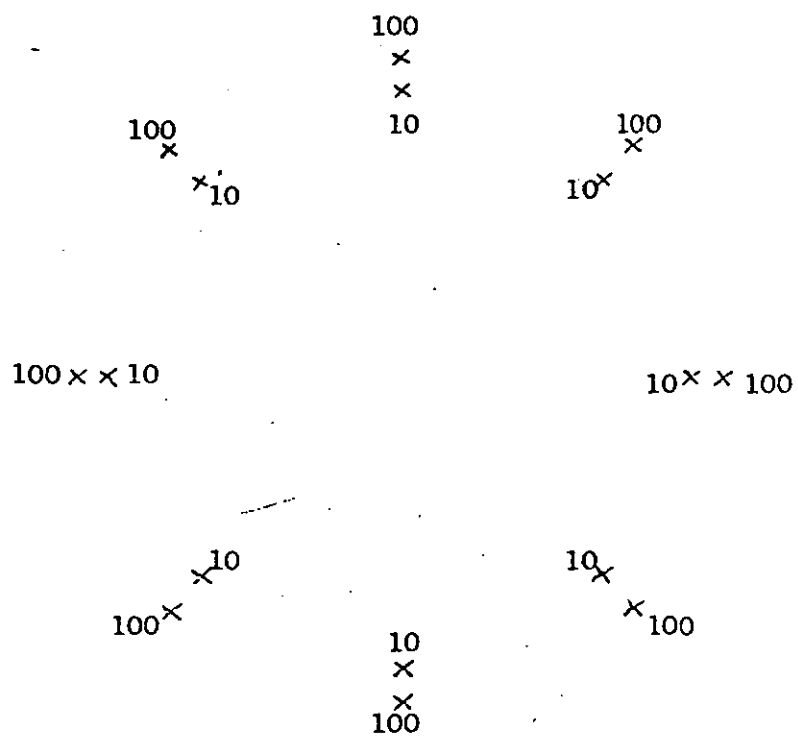


FIGURE 4.18: "Hole" test data for contouring



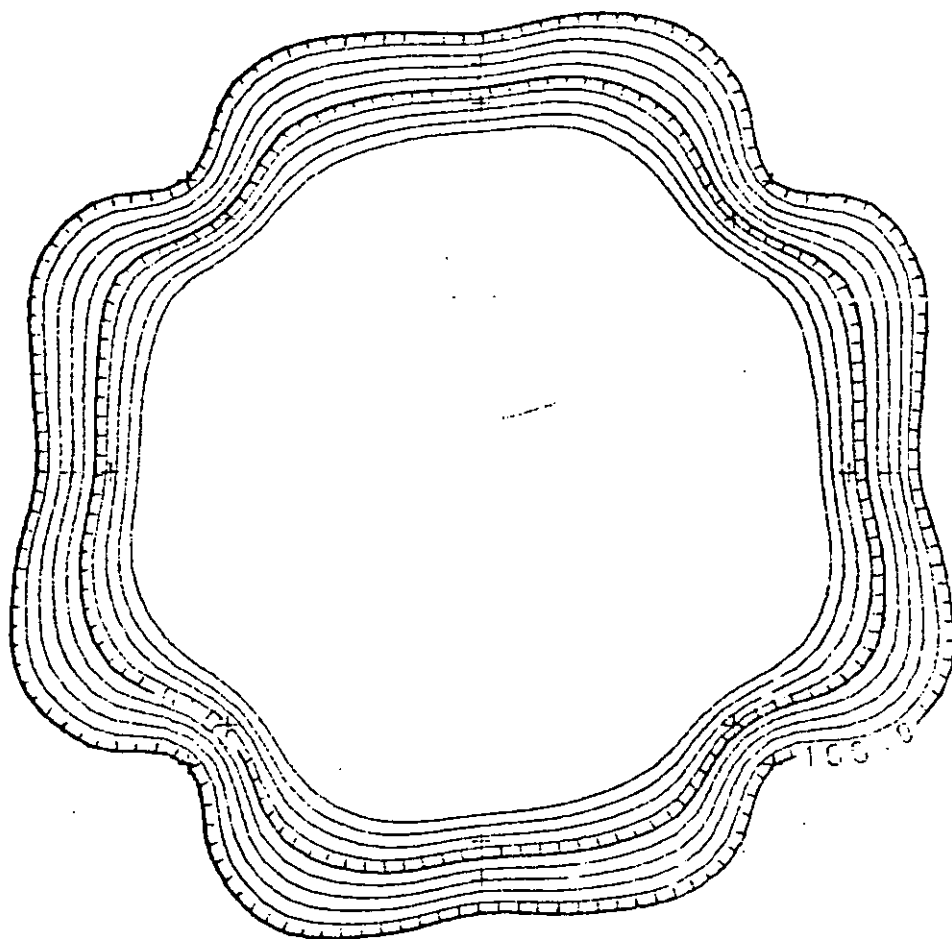


FIGURE 4.19: "Hole" data - GPCP 20x20 grid

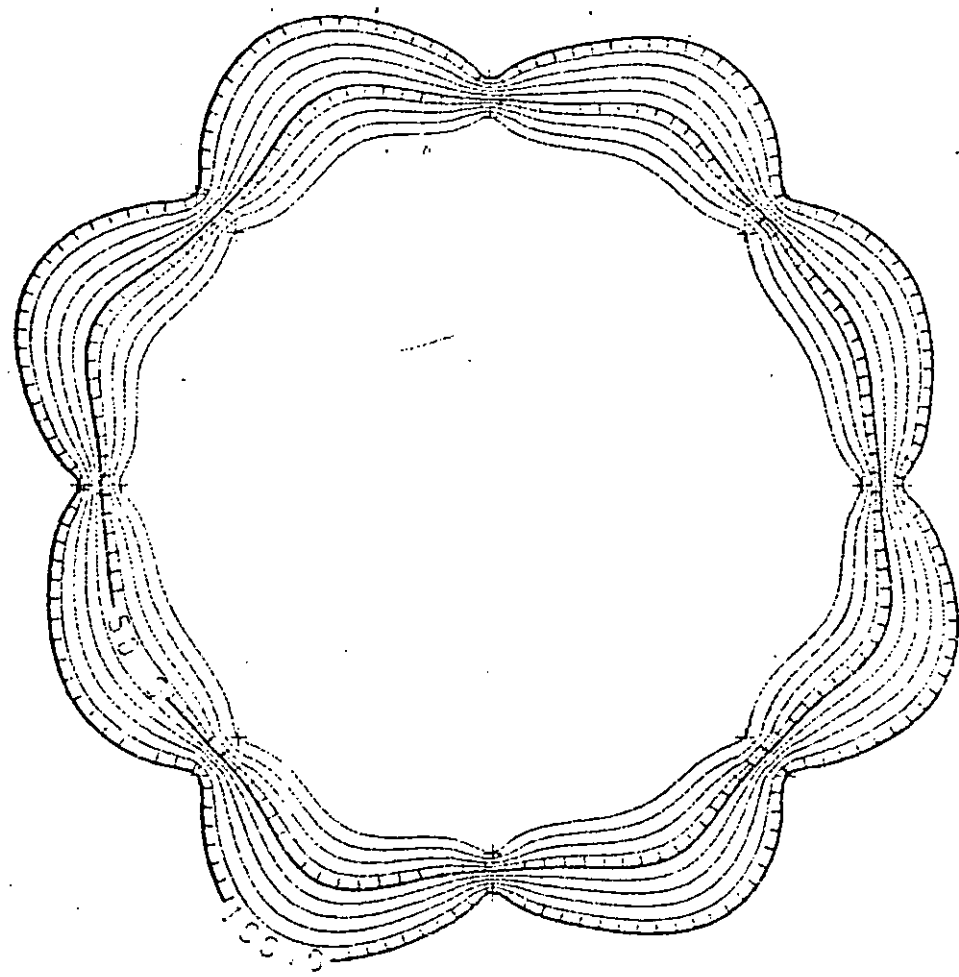


FIGURE 4.20: "Hole" data - GPCP 100x100 grid

"Hole" test data.

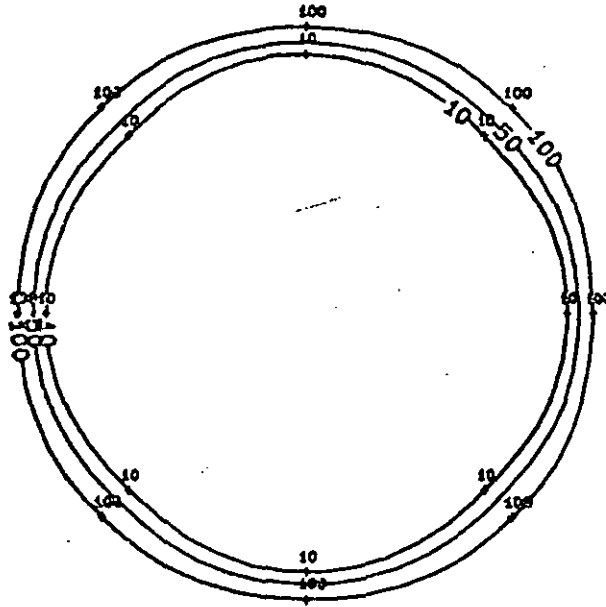


FIGURE 4.21

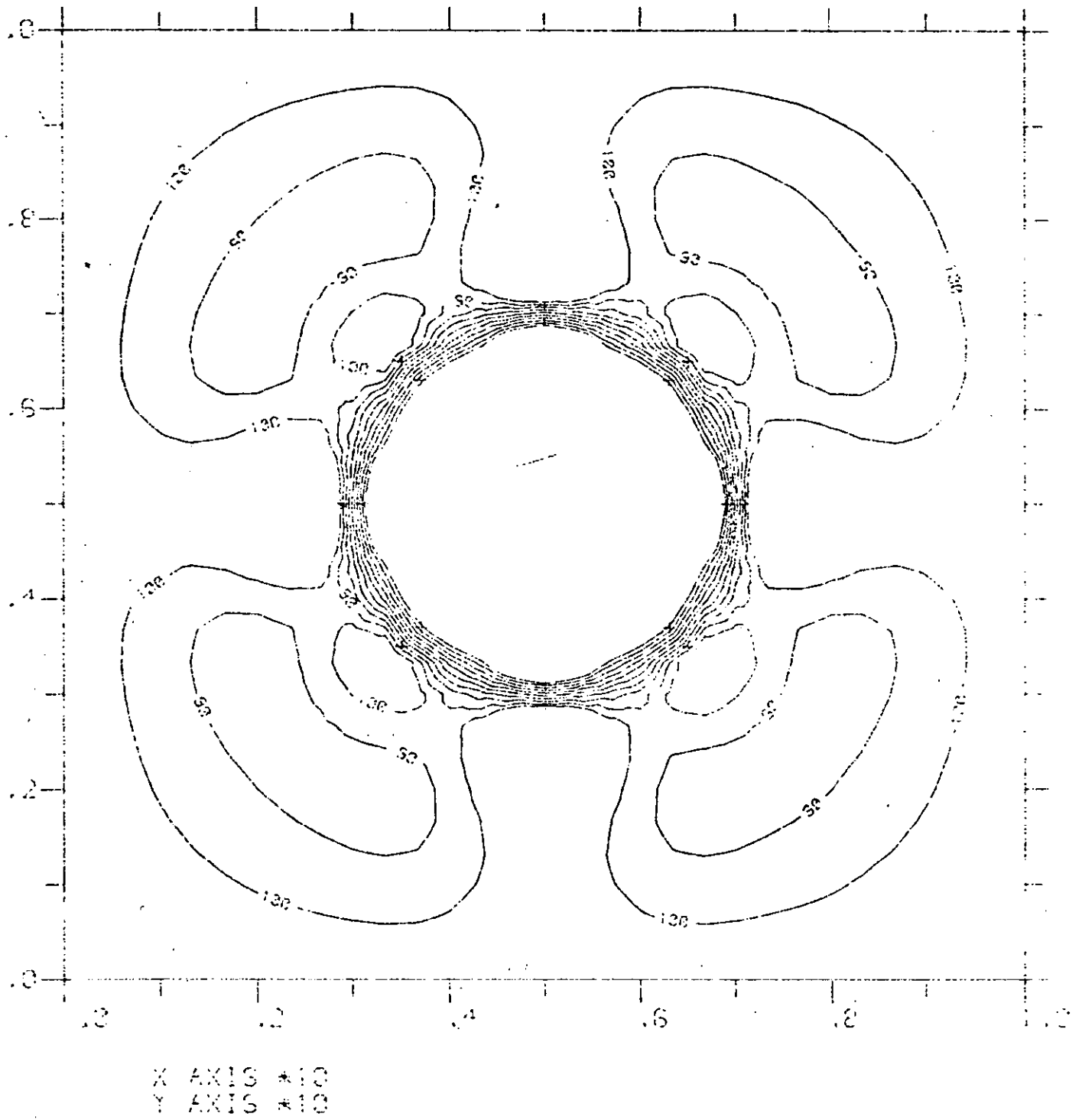


FIGURE 4.22: "Hole" data - GINOSURF

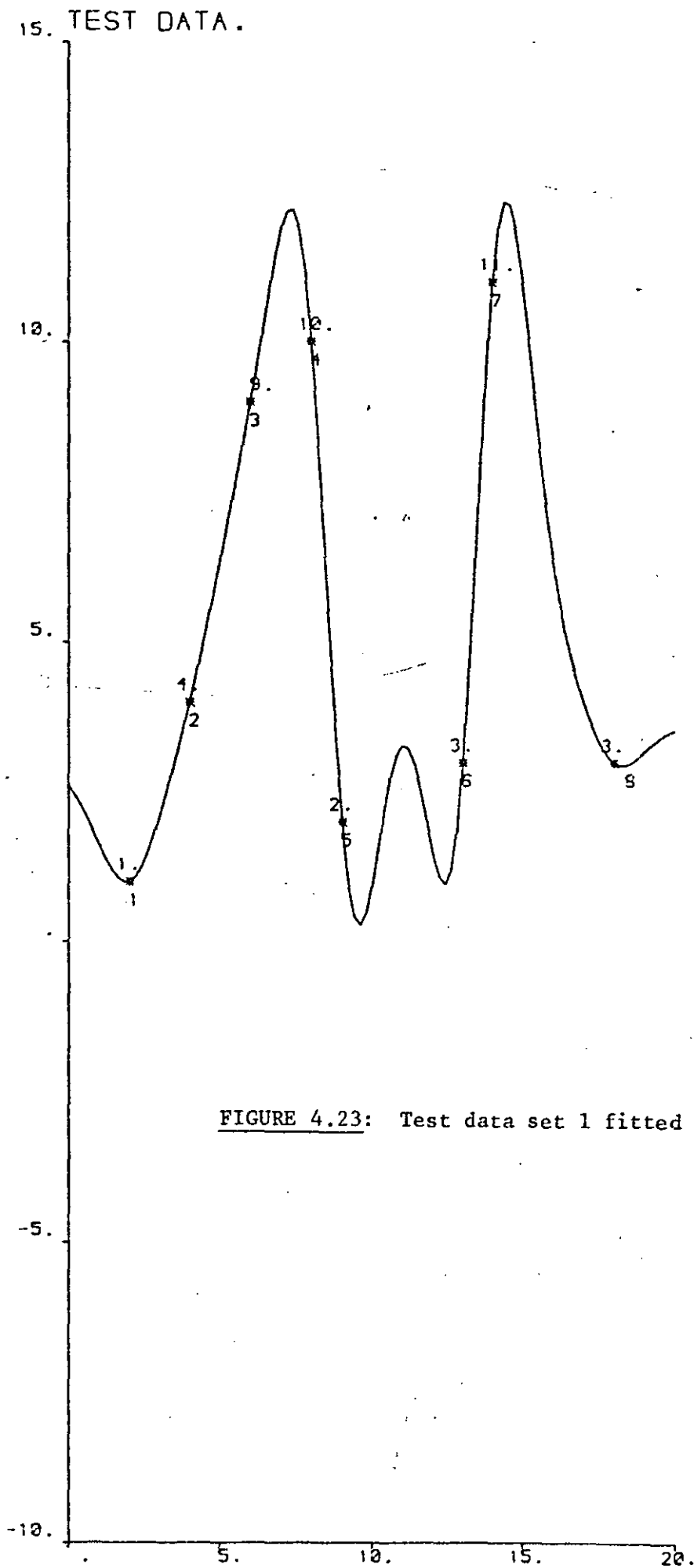


FIGURE 4.23: Test data set 1 fitted by SIMP

20. TEST DATA 2.

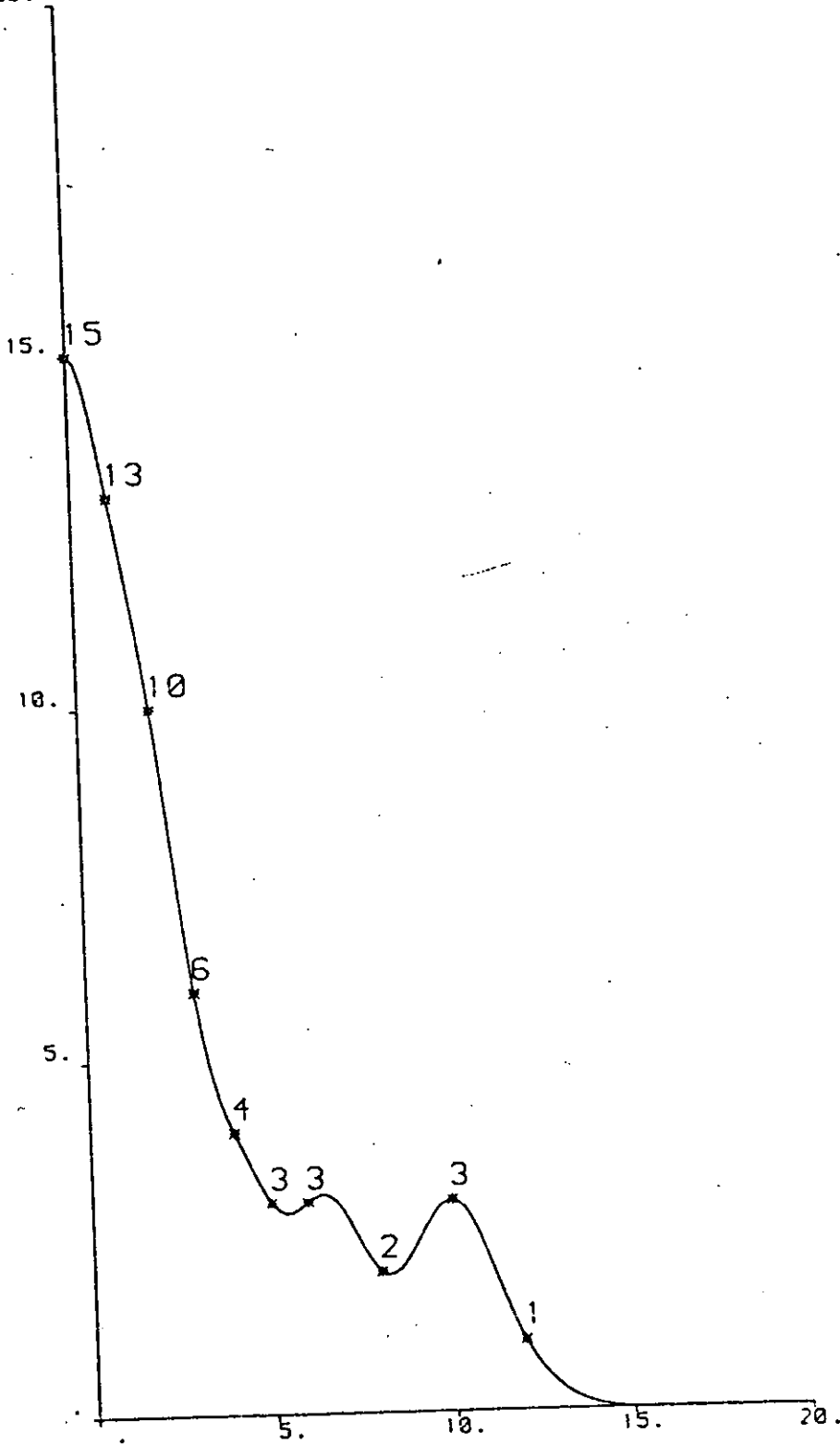


FIGURE 4.24: Test data set 2 fitted by SIMP

CHAPTER 5

APPLICATION TO THE OPTIMISATION OF FUNCTIONS

OF SEVERAL VARIABLES

## 5.1 GENERAL OUTLINE

As mentioned earlier, the problem of global optimisation of a function of several variables has been studied by many people. Dixon et al (1976) make the point that what is perhaps needed is a technique which will obtain a good estimate of the optimum value of the function in a reasonably small number of function evaluations, assuming that each function evaluation is very expensive and that it is therefore worthwhile undertaking a considerable amount of calculation between function evaluations if this leads to a reduction in the number required. A method aimed at meeting this objective has been developed using the concept of stochastic interpolation introduced earlier in this work.

Conventional optimisation techniques normally require a reasonable number of function evaluations to reach a successful conclusion. This is especially the case if a method depending on the computation of the derivatives of the function in several dimensions is to be used. Ideally we want to be able to use the minimum number of function evaluations to give the maximum information about the overall form of the function. We should also like to be able to estimate the derivatives of the function at a point based on the existing known values at other points.

Given an objective function  $F()$ , whose derivative is not easily computable, and a "region of interest"  $R$  within which the search for an optimum value is to be conducted, the outline algorithm is as follows:

1. Carry out  $N$  function evaluations at a set of initial points  $\{x_1, x_2, \dots, x_N\}$  scattered throughout  $R$ .
2. Fit a stochastic interpolating function  $f()$  to this set of



function values  $\{z_1, z_2, \dots, z_N\}$ . The function  $f()$  will have derivatives which are easily computable, using equation (2.35).

3. Find the optimum value in the initial set and optimise  $f()$  starting from this point by means of a conventional optimisation technique using the derivatives of  $f()$ .
4. Evaluate the true function value  $F(\underline{x}^*)$  at the optimum point  $\underline{x}^*$  of the interpolating function  $F()$  found in the previous step, and compare this with  $f(\underline{x}^*)$ . If they are not sufficiently close, refit the interpolating function using the new point and repeat step 3 above.
5. If they are sufficiently close, either terminate with this as the estimated optimum value, or proceed to refine the solution locally using a conventional technique on  $F()$ .

Several interesting problems must be dealt with before this simple outline scheme can be put into practice. The distribution of the initial set of  $N$  function evaluations in such a way as to gain the maximum information about  $F()$  in the region  $R$  is not a trivial problem, and will be discussed in the next section.

Fitting the stochastic interpolating function  $f()$  to the  $N$  data points can be carried out using the concept of the two stage model developed in the previous chapter, possibly extended to more than two dimensions. However, two changes seemed to be appropriate in this case. Firstly, since the data points will be chosen to be as widely scattered as possible throughout  $R$ , it does not seem sensible to generate the average points which define the long-range trend by a clustering algorithm

as described previously. Instead, the number of average points  $n_A$  to be used is defined in advance and the locations of these points are generated using the same algorithm which generates the positions of the initial  $N$  data points. Values of  $F()$  are not computed at these points - instead weighted average values are calculated from nearby data points, using as weighting function the auto-correlation function  $g()$ .

$$z_j^A = \frac{\sum_{i=1}^N z_i g(r_{ij})}{\sum_{i=1}^N g(r_{ij})}, \quad (5.1)$$

where  $r_{ij}$  is the distance from average point  $j$  to initial data point  $i$ .

The long-range trend is fitted to these averaged values, giving rise to the parameter  $\rho_L$ , the long-range correlation distance. Values of the trend are computed at each of the initial data points, and subtracted from the known function values there, giving a set of residual values. The short-range residual function is fitted to these, with a parameter  $\rho_s$ , the short-range correlation distance. Thus the interpolating function  $f(\underline{x})$  at any point is the sum of the trend function (based on the  $n_A$  average points) and the residual function (based on the  $N$  residual values at the original data points).

This technique for fitting trend and residual functions to the data is in most essentials similar to that described in Chapter 4, and suffers from the same limitations, especially in the matter of disentangling trend from residual. The "smoothing" to obtain the average values is carried out using the auto-correlation function, which depends on the unknown correlation distance  $\rho$ . This difficulty is resolved in SIMP by setting the value of  $\rho$  equal to half the "average inter-point distance",  $\bar{d}$ , a formula for which is given as equation 5.20. Some simulation experiments have been carried out to validate this estimation procedure

(see Appendix B) and the results seem to indicate that the overall structure of the data can be reproduced, even if the actual estimates of the correlation distances are low.

The interpolating function  $f()$  is optimised using a "variable metric" algorithm, as described in Zoutendijk (1976) pp.370f. This requires the first derivatives of  $f()$ , which in practice are the sums of the derivatives of the trend and the residual functions.

Having reached a local optimum of  $f()$ , at a point  $\underline{x}^*$  say, the true function value  $F(\underline{x}^*)$  is evaluated here and included in the set of known data points.  $N$  is incremented by 1,  $\underline{x}_N$  is set equal to  $\underline{x}^*$ , and  $z_N$  become  $F(\underline{x}^*)$ . If  $|F(\underline{x}^*) - f(\underline{x}^*)| < \epsilon$ , a given tolerance, the procedure terminates. Otherwise the interpolating function is refitted including the new point, and optimisation of  $f()$  begins again starting at the current best point in the data set.

An additional feature which is simple to include is an estimate of the integral of  $F()$  over the region  $R$ . This is discussed in a later section.

## 5.2 CONVERGENCE OF THE OPTIMISATION ALGORITHM AND RELATED QUESTIONS

The optimisation algorithm outlined previously is based on the concept of iterative optimisation of an interpolating function, a new data point being evaluated each time, until ultimate agreement is reached between the (optimum) interpolated value and the true function value. This leads to three related questions being raised:

1. Is this algorithm guaranteed to converge to a result in a finite number of iterations?
2. Does the introduction of a new data point in the very near neighbourhood of an existing data point always lead to improved accuracy in the interpolating function in that neighbourhood?
3. What is the behaviour of the gradient of the interpolating function as data points become very close to each other, in particular when the method converges on an optimum value?

To answer these questions, let us assume that we have a known data point (Point 1) and another location (Point 3) where the function value is unknown, and let us suppose that the distance  $h_{13}$  between these two points is sufficiently small so that the correlation between the values at the two points

$$g(h_{13}) = 1 + \frac{h_{13}^2}{2} g''(0) + o(h_{13}^4) , \quad (5.2)$$

may be closely enough approximated by

$$g(h_{13}) = 1 - \frac{h_{13}^2}{2} \sigma_s^2 . \quad (5.3)$$

where  $\sigma_s^2 = -g''(0)$  .

Now the variance of the unknown value at Point 3 given the known value at Point 1 is

$$\begin{aligned} \text{Var}[Z_3|Z_1] &= \sigma^2(1 - g^2(h_{13})) \\ &= \sigma^2 \sigma_s^2 h_{13}^2 + O(h_{13}^4) . \end{aligned} \quad (5.4)$$

This result enables us to examine the question of convergence. The optimisation algorithm will terminate when a point  $\underline{x}^*$  is found which is a local optimum point of the interpolating function and where

$$|F(\underline{x}^*) - f(\underline{x}^*)| < \epsilon . \quad (5.5)$$

For a given  $\epsilon$  value and a certain probability level  $\alpha$ , we may define a distance  $\delta$  so that

$$P[|F(\underline{x}) - f(\underline{x})| < \epsilon] \geq \alpha$$

for points within a distance  $\delta$  of a known data point. For small  $\delta$ , we know from (5.4) that the variance of the interpolating function is approximated closely by  $\sigma^2 \sigma_s^2 \delta^2$ , and so

$$P[|F(\underline{x}) - f(\underline{x})| \geq \epsilon] = 2(1 - \Phi(\frac{\epsilon}{\sigma \sigma_s \delta})) , \quad (5.6)$$

where  $\Phi()$  is the Standard Normal Integral Function.

Thus  $\delta$  is chosen so that

$$\Phi(\frac{\epsilon}{\sigma \sigma_s \delta}) = \frac{1}{2}(1 + \alpha) . \quad (5.7)$$

Therefore, if the new point  $\underline{x}^*$  is within a distance  $\delta$  of an existing data point, the probability is at least  $\alpha$  that the termination criterion will be met. The probability is zero that an infinite number of data points will be generated in a finite region without satisfying the termination criterion.

Let us now consider the situation when two known data points are closely adjacent. Introduce a second data point (Point 2) at a distance  $h_{12}$  from Point 1 and a distance  $h_{23}$  from Point 3 (see Figure 5.1). We should like to be able to show that, for suitably small distances,

$$\text{Var}[Z_3|Z_1, Z_2] < \text{Var}[Z_3|Z_1] \quad (5.8)$$

Now, if  $\underline{X}$  and  $\underline{Y}$  are two vectors which are jointly multivariate Normal, it can be seen that (e.g. Whittle, 1963, p.46-47)

$$\text{Var}(\underline{X}|\underline{Y}) = S_{XX} - S_{XY}S_{YY}^{-1}S_{XY} \quad (5.9)$$

where  $S_{XX}$  = covariance matrix for  $\underline{X}$

$S_{XY}$  = cross-covariance matrix between  $\underline{X}$  and  $\underline{Y}$

$S_{YY}$  = covariance matrix for  $\underline{Y}$

Since  $S_{YY}$  and hence  $S_{YY}^{-1}$  are non-negative definite, we can see that

$$\text{Var}(\underline{X}|\underline{Y}) \leq \text{Var}(\underline{X}) \quad (5.10)$$

So if we set  $\underline{X}$  equal to the residual error in  $Z_3$  predicted from  $Z_1$ , and  $\underline{Y}$  to  $Z_2$ , the relationship (5.8) follows immediately.

Thus, for an interpolating function based on realisations of a regular stochastic process with continuous derivatives up to at least second order, the addition of a new data point within a close distance of an existing point will reduce the variance of the interpolation errors within a certain region.

In terms of modelling a real function  $F()$ , let us assume that  $F()$  has continuous derivatives up to at least second order, so that within some small region,

$$(F'(\underline{x}))^2 < \text{some finite limit.}$$

We may assume that, within the given small region,  $F()$  is a realisation

of a stochastic process whose gradient has finite variance, i.e. a regular process with continuous derivatives up to at least second order. For such functions, therefore, as the number of data used to interpolate them increases, ultimately the residual errors in the interpolation will tend to decrease everywhere.

We shall now turn our attention to the question of the behaviour of the gradient of the interpolating function in the near neighbourhood of two adjacent data points. Having pivoted on Point 1 and Point 2, the estimated value at the unknown Point 3 is

$$f(\underline{x}_3) = \mu - \gamma_1 g(h_{13}) - \gamma_2 g(h_{23}) \quad (5.11)$$

where

$$\gamma_1 = \frac{z_1 - g(h_{12})z_2}{1 - g^2(h_{12})}$$

$$\gamma_2 = \frac{z_2 - g(h_{12})z_1}{1 - g^2(h_{12})}$$

and  $z_1$  and  $z_2$  are the values (less the mean  $\mu$ ) at the two known points.

Therefore,

$$\frac{\partial}{\partial x_k} f(\underline{x}_3) = -\gamma_1 \frac{\partial}{\partial x_k} g(h_{13}) - \gamma_2 \frac{\partial}{\partial x_k} g(h_{23}) \quad (5.12)$$

This gives a general form of the slope of the interpolating function in the near neighbourhood of these two data points. We shall examine two special cases:

1. Point 3 has its  $x_k$  co-ordinate in the interval between the  $x_k$  co-ordinates of Points 1 and 2. Let us further assume it is midway between the points, so that

$$\frac{\partial}{\partial x_k} g(h_{13}) = -\frac{\partial}{\partial x_k} g(h_{23}) = c' \quad (5.13)$$

Therefore

$$\begin{aligned} \frac{\partial}{\partial x_k} f(\underline{x}_3) &= -(\gamma_1 - \gamma_2)c' \\ &= \frac{c'}{1 - g^2(h_{12})} (z_1 - z_2)(1 + g(h_{12})) \end{aligned} \quad (5.14)$$

If  $h_{12}$  is sufficiently small,  $1+g(h_{12}) \approx 2$  and  $1-g^2(h_{12}) \approx \sigma_s^2 h_{12}^2$ .

$$\text{So } \frac{\partial}{\partial x_k} f(\underline{x}_3) \approx \frac{2c'(z_1 - z_2)}{\sigma_s^2 h_{12}^2} \quad (5.15)$$

If Point 3 actually lies on the straight line joining Points 1 and 2, and the difference between the two points is in the  $x_k$  dimension only, then

$$\begin{aligned} c' &= \frac{\partial}{\partial h_{13}} \left(1 - \frac{h_{13}^2}{2} \sigma_s^2\right) \\ &= h_{13} \sigma_s^2 = \frac{h_{12}}{2} \sigma_s^2 \end{aligned} \quad (5.16)$$

$$\text{And then } \frac{\partial}{\partial x_k} f(\underline{x}_3) \approx \frac{z_1 - z_2}{h_{12}} \quad (5.17)$$

In this case, we have shown the gradient of the interpolating function midway between the two points to be equal to the chord slope joining the points, which is an extremely reasonable result.

2. If Point 3 is well beyond Points 1 and 2 in the  $x_k$  direction, so that

$$\frac{\partial}{\partial x_k} g(h_{13}) \approx \frac{\partial}{\partial x_k} g(h_{23}) = c' \quad (5.18)$$

$$\begin{aligned} \text{Then } \frac{\partial}{\partial x_k} f(\underline{x}_3) &= \frac{c'(z_1 + z_2)(1 - g(h_{13}))}{\sigma_s^2 h_{12}^2} \\ &= \frac{c'(z_1 + z_2)(\frac{1}{2} h_{12}^2 \sigma_s^2)}{\sigma_s^2 h_{12}^2} \\ &= \frac{c'}{2} (z_1 + z_2) \end{aligned} \quad (5.19)$$

This again is quite reasonable, since it implies that the interpolating function slope is equal to the slope of the auto-correlation function times the average of the deviations from the mean at the two data points.



### 5.3 DISTRIBUTION OF INITIAL POINTS

This is an important part of the algorithm, and may well repay further study. Even if the region of interest  $R$  is of simple form, it is by no means obvious how to arrange  $N$  points so as to survey the  $m$ -dimensional region as efficiently as possible, especially if  $N < 2^m$ .

Intuitively, the points should be spaced apart as far as possible from one another, without lying on the boundaries of  $R$ . They should also be spread evenly throughout  $R$  so as to maximise the information gained about the form of  $F()$ . For this reason scattering points randomly through  $R$  is not recommended, since it leads to an uneven distribution and parts of  $R$  which are not close to a data point. It also means that the results cannot be reproduced exactly.

The approach which has been adopted is to set up the positions of the  $N$  data points using an 'ad hoc' technique, and then adjust the positions to minimise a 'repulsive' function which will tend to spread the points more evenly through  $R$ .

We shall assume that  $R$  is of simple rectangular form:  $\underline{x} \in R$  if  $a_k \leq x_k \leq b_k$ ,  $k=1, \dots, m$ . With this assumption, two 'ad hoc' methods for initialising the points have been developed.

#### Method 1

This is a recursive algorithm. At any stage, we have  $N'$  points to be positioned in a region  $R'$ . If  $N'$  is odd, place one point in the centre of  $R'$ . Divide  $R'$  into two equal regions along its longest dimension, and position half the remaining points in each such region using the same algorithm.

## Method 2

Define for each dimension  $k$  a "high value"  $x_k^u = \alpha b_k + (1-\alpha)a_k$ , and a "low value"  $x_k^L = \alpha a_k + (1-\alpha)b_k$ , where  $\alpha$  is taken to be 0.75, for example. The position of each point may be represented by a word of  $m$  bits, where a 0 in position  $k$  represents  $x_k^L$  and a 1 represents  $x_k^u$ . Such "words" are permuted systematically so as to produce a set of points which will explore reasonably well the total region  $R$ .

Method 1 above works quite well for  $N \geq 2^m$ , but otherwise gives rise to sets of points lying in a subspace of  $R$ . Method 2 is preferred in this case. Whichever is used, the distribution of points is unlikely to be ideal. The locations are updated to minimise a "repulsive" function which aims to spread them as widely as possible throughout  $R$ .

The basis of the repulsive function was taken to be the same as the correlation function  $g()$ , as given in equation (2.37). Intuitively, this corresponds to positioning the initial set of data points so as to minimise the total correlation between them. However, at this stage we do not know the value of  $\rho$ , the correlation distance, to be used. Therefore, let us define the average distance between the points,  $\bar{d}$ , by dividing up the total volume of  $R$  between the  $N$  points and finding the dimensions of the equivalent hypercube for each point.

$$\text{i.e.} \quad \bar{d} = \left( \prod_{k=1}^m (b_k - a_k) / N \right)^{1/m} \quad (5.20)$$

Good results are obtained if we set  $\rho = \frac{1}{2}\bar{d}$ .

As well as repelling the points from one another, we also need to repel them from the boundaries of  $R$ . Otherwise they are obviously

giving information about the space beyond R, which is not required. The way in which this is achieved is to imagine that each point has an "image" in each of the  $2m$  boundaries of R, and that it is also repelled from these image points. Thus the total repulsive function to be minimised is

$$\begin{aligned}
 H = & \sum_{i=1}^N \sum_{j=1}^N \exp -d_{ij}^2 / 2\rho^2 + \sum_{i=1}^N \sum_{k=1}^{2m} \exp -(x_{ik} - a_k)^2 / 2\rho^2 \\
 & + \sum_{i=1}^N \sum_{k=1}^{2m} \exp -(x_{ik} - b_k)^2 / 2\rho^2
 \end{aligned} \tag{5.21}$$

where  $d_{ij}^2 = \sum_{k=1}^m (x_{ik} - x_{jk})^2$ . (See Figure 5.2).

This function is adjusted by moving one point at a time, in one dimension only (based on the first derivatives of H). This terminates when the changes in successive values of H are less than a preset tolerance. Figure 5.3 shows an example of positioning 20 points in a square two-dimensional region of interest.

#### 5.4 ANISOTROPIC CORRELATION

In all the work carried out so far, we have assumed that the auto-correlation function between two points  $\underline{x}_i$  and  $\underline{x}_j$  is of the form

$$g(\underline{r}) = \exp(-d_{ij}^2/2\rho^2)$$

where

$$d_{ij}^2 = \sum_{k=1}^m (x_{ik} - x_{jk})^2 . \quad (5.22)$$

It is very simple to extend this definition to the case where the correlation is not isotropic:

Let

$$d_{ij}^2 = \sum_{k=1}^m \alpha_k (x_{ik} - x_{jk})^2 , \quad (5.23)$$

where the  $m$  coefficients  $\alpha_k$ ,  $k=1, \dots, m$  are the "anisotropy factors" for the model.

With this addition, everything carries through as before, with minor changes. For example, the derivatives of the interpolation function become

$$\begin{aligned} \frac{\partial}{\partial x_k} f(\underline{x}) &= \sum_{i=1}^N \gamma_i \alpha_k g'(r_i) \\ &= \sum_{i=1}^N \gamma_i \alpha_k (x_{ik} - x_k) \exp(-r_i^2/2\rho^2) / \rho^2 \end{aligned}$$

where

$$r_i = \sqrt{\sum_{k=1}^m \alpha_k (x_{ik} - x_k)^2} . \quad (5.24)$$

What we are essentially doing here is to rescale the coordinates  $\underline{x}$  to produce an isotropic model. Matérn (1960, p.17) shows that  $\exp(-\underline{u}'A\underline{u})$  is a suitable auto-correlation function where  $A$  is an orthonormal matrix, and this corresponds to a more general case with the directions of anisotropy not aligned along the coordinate axes.

In geostatistics (see David, 1977, p.134 and Journel & Huijbregts,

1978; p.177-183) there is a distinction made between "geometrical" or "affine" anisotropy, which is the kind of model outlined above in which different scaling factors are applied in different directions, and "zonal" or "stratified" anisotropy, in which there exist strata or zones in space with different properties. We shall not consider the latter case at the present time.

The main problem that remains is that of estimating the anisotropy factors  $\alpha_k$ ,  $k=1, \dots, m$  from the  $N$  given data points. Two methods are outlined below, but they are by no means ideal or foolproof. More work needs to be done on this problem.

#### METHOD I

1. Take a set of pairs of neighbouring points, with say the  $\ell^{\text{th}}$  such pair being points  $i$  and  $j$ .
2. Estimate the correlation  $c_\ell$  between the values at the two points using the method outlined on page 25.
3. Form the anisotropic auto-correlation function,

$$c_\ell = \exp \left[ - \sum_{k=1}^m \alpha_k \Delta x_{\ell k}^2 / 2\rho^2 \right] \quad (5.25)$$

where  $\Delta x_{\ell k}^2 = (x_{ik} - x_{jk})^2$ .

Taking logs and setting  $y_\ell = -\log c_\ell$

$$y_\ell = \sum_{k=1}^m \alpha_k \Delta x_{\ell k}^2 / 2\rho^2 \quad (5.26)$$

4. If we use  $L$  such pairs of points, then we may estimate the coefficients  $\alpha_k / 2\rho^2$  by multiple linear regression applied to the  $L$  values  $y_\ell$ ,  $\ell=1, \dots, L$ , dependent on the  $L$  values of the  $m$  variables  $\Delta x_{\ell k}^2$ ,  $\ell=1, \dots, L$ ,  $k=1, \dots, m$ . Obviously our estimates of the anisotropy factors are therefore conditioned by the

value of  $\rho$ , which is at this stage unknown, but their ratios may be estimated and they may be normalised to give a mean value of 1.0.

## METHOD II

1. For each dimension  $k$ , find pairs of points  $i$  and  $j$  such that the separation between them is mainly in the  $x_k$ -direction, i.e.,

$$d_{ij} \leq \lambda |x_{ik} - x_{jk}|$$

where  $\lambda$  can be taken to be, for example, 1.5. (This is similar to the idea of "angle classes" used in geostatistics - see Journel & Huijbregts, 1978, p.211).

2. For the  $\ell^{\text{th}}$  pair, compute  $\Delta \ell = |Z_i - Z_j| / |x_{ik} - x_{jk}|$ .
3. Find the median,  $\tilde{\Delta}(k)$ , over all such pairs, and repeat from step 1 for all values of  $k$  from 1 to  $m$ .
4. Assume for each  $k$  that the median "gradient"  $\tilde{\Delta}(k)$  is approximately proportional to  $\sqrt{\alpha_k}$ , the linear anisotropy factor. Hence the ratios of the anisotropy factors may be estimated.

Step 4 above can be justified from the work of Hawkins & Cressie (1981). They define  $Y = \sqrt{|Z_i - Z_j|}$  and show that a reasonable estimator of the variogram is proportional to the median of  $Y_\ell^4$ . The variogram is proportional to  $1-g(r) = 1 - \exp(-\alpha_k r^2 / 2\rho^2)$  if the distance is all in the  $k^{\text{th}}$  dimension. Thus, for reasonably small  $r$ , we may say approximately that  $\tilde{Y}^4$  is proportional to  $\alpha_k r^2$ .

Both these methods have been tested on simulated anisotropic data - the results are presented in Appendix B. In both cases the anisotropy of the data is clearly underestimated, although it appears that Method II is better than Method I. It seems to be possible to detect the presence of

anisotropy, without necessarily obtaining a very precise estimate of its value. It is therefore important, in the optimisation application, to ensure that the variables used are scaled beforehand so as to reduce, as far as possible, the effect of anisotropy.

### 5.5 ESTIMATION OF THE FUNCTION INTEGRAL

It is a simple matter to calculate the integral of the interpolating function  $f(\underline{x})$  over the region  $R$ , once it has been fitted to the data points. This may be of some value in estimating the integral of the true function  $F(\underline{x})$ , and can be produced as a "by-product" from the optimisation process.

If the interpolating function is

$$f(\underline{x}) = \sum_{i=1}^N \gamma_i c_i(\underline{x}) + \mu$$

$$\text{then } \int_R f(\underline{x}) d\underline{x} = \sum_{i=1}^N \int_R \gamma_i c_i(\underline{x}) d\underline{x} + \mu \int_R d\underline{x} \quad (5.27)$$

With the assumed form of covariance function, this becomes

$$\int_R f(\underline{x}) d\underline{x} = \sum_{i=1}^N \gamma_i \int_R \exp\left[-\sum_{k=1}^m \alpha_k (x_{ik} - x_k)^2 / 2\rho^2\right] dx_1, dx_2, \dots, dx_m + \mu \int_R d\underline{x} \quad (5.28)$$

Now if  $R$  is a rectangular region, bounded between limits  $a_k$  and  $b_k$  in the  $k^{\text{th}}$  dimension, then

$$\begin{aligned} & \int_R \exp\left[-\sum_{k=1}^m \alpha_k (x_{ik} - x_k)^2 / 2\rho^2\right] dx_1, dx_2, \dots, dx_m \\ &= \prod_{k=1}^m \int_{a_k}^{b_k} \exp\left[-\alpha_k (x_{ik} - x_k)^2 / 2\rho^2\right] dx_k \end{aligned} \quad (5.29)$$

And

$$\begin{aligned} & \int_{a_k}^{b_k} \exp\left[-\alpha_k (x_{ik} - x_k)^2 / 2\rho^2\right] dx_k \\ &= \rho \sqrt{\frac{2\pi}{\alpha_k}} \left[ \Phi\left(\frac{\sqrt{\alpha_k} (b_k - x_{ik})}{\rho}\right) - \Phi\left(\frac{\sqrt{\alpha_k} (a_k - x_{ik})}{\rho}\right) \right] \end{aligned} \quad (5.30)$$

where  $\Phi(\cdot)$  is the Standard Normal Integral function.

In practice, the full integral is the sum of the integrals of the long-range trend and the short-range residual. (See Matérn, 1960, p.20 & Matheron, 1971, p.59).



## 5.6 RESULTS WITH TEST FUNCTIONS

Tests of this optimisation technique have been carried out on a variety of functions, most of which have been used by other workers to test global optimisation algorithms. None of them is in fact "expensive" to compute, but it is hoped that good results with these test functions will show promise for general applications.

The functions and the results obtained are described together, and then the overall results are summarised.

### 1. Simple Test Function (2d)

This merely illustrates the use of the interpolating function for both optimisation and integration.

$$F_1(x_1, x_2) = (\sin x_1 - \frac{1}{2}x_1)(2x_2 - x_2^2) . \quad (5.31)$$

The region R is  $0 \leq x_1 \leq \pi/2$ ;  $0 \leq x_2 \leq 3$ .

This function has a maximum value of 0.3424 at  $(\pi/3, 1)$  and the total integral over R is 0.

10 initial data points were generated, and a stochastic interpolating function was fitted with no trend or anisotropy factors assumed. Figure 5.4 illustrates how the algorithm conducted its search from the current highest point. After four additional function evaluations a point was found at which real and interpolated function values agreed to within a specified tolerance ( $10^{-4}$ ), giving an estimated maximum value of 0.3417 at (1.0717, 0.9621). The estimated integral was -0.0101.

## 2. Branin's RCOS (2d)

This function is described in de Biase and Frontini (1978).

$$F_2(x_1, x_2) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 8.75 \cos x_1 + 10. \quad (5.32)$$

The region R is  $-5 \leq x_1 \leq 10$  ;  $0 \leq x_2 \leq 15$  .

This function has three global minima with values 1.25 within R, and the total integral over R is  $12.238 \times 10^3$  .

20 initial data points were generated, and anisotropy factors were estimated as (0.7658, 1.2342). The trend function was fitted using 12 average points. Figure 5.5 is a contour map of the interpolating function fitted to the 20 data points - the approximate locations of the three global minima are apparent. Figure 5.6 shows the average points and the trend function fitted. Figures 5.7 and 5.8 are cross-sections of the interpolating function along the diagonals of R.

The lowest of the 20 initial function values was 2.7060, and after 10 extra function evaluations the optimisation algorithm terminated with a value of 1.2754 at (-3.1403, 12.4313). This is close to one of the global minima at  $(-\pi, 12.275)$ . Figure 5.9. shows the region close to the minimum and the path traced by the algorithm during optimisation. Figure 5.10 is a plot of current function value versus number of evaluations, and shows how the algorithm converges. The estimated integral was  $11.342 \times 10^3$  .

## 3. Goldstein and Price's Function (2d)

This function is described in de Biase and Frontini (1978).

$$F_3(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)^2] \\ [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]. \quad (5.33)$$

The region  $R$  is  $-2 \leq x_i \leq 2$ ,  $i=1,2$ . This function has a global minimum with value 3.0 at  $(0,-1)$ .

40 initial points were generated, and anisotropy factors were estimated as  $(0.7662, 1.2338)$ . The trend was fitted with 20 average points and Figures 5.11 and 5.12 show contour maps of the full interpolating function and the trend function, respectively. Figures 5.13 and 5.14 are cross-sections along the diagonals of  $R$ . The lowest initial function value was 547.19, and after 13 extra function evaluations the algorithm terminated with a value of 6.226 at  $(-0.0152, -0.9614)$ . Figure 5.15 illustrates the progress of the optimisation (note the bad guess at value number 50).

#### 4. Rosenbrock's Banana Valley Function (2d)

This function is commonly used to test hill-climbing optimisation techniques, as it involves searching along the bottom of a steep curved valley.

$$F_4(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (5.34)$$

The region  $R$  was chosen as  $-5 \leq x_i \leq 5$ ,  $i=1,2$ . The minimum point is at  $(1,1)$  with a value of 0.

20 points were generated initially, with the trend fitted to 10 average points. Figure 5.16 shows a contour map of the interpolating function. Figure 5.17 is a cross-section along the tangent to the valley at  $(1,1)$ . The initial lowest point was 49.975, and after an extra 8 function evaluations a value of 0.0219 was reached at  $(1.0227, 1.0312)$ . Figure 5.18 illustrates the progress of the algorithm.

5. Shekel Function (4d)

de Biase and Frontini (1978) describe this family of test functions.

A 4-variable example was chosen:

$$F_5(x_1, x_2, x_3, x_4) = \sum_{i=1}^7 \frac{1}{\sum_{k=1}^4 (x_k - a_{ik})^2 + c_i} \quad (5.35)$$

with the following constants:

<u>i</u>	<u>c<sub>i</sub></u>	<u>a<sub>ik</sub></u>
1	0.1	0.7, 0.7, 0.5, 0.1
2	0.2	0.3, 0.3, 0.8, 0.2
3	0.3	0.8, 0.6, 0.8, 0.6
4	0.4	0.4, 0.2, 0.9, 0.9
5	0.5	0.1, 0.6, 0.8, 0.1
6	0.6	0.8, 0.2, 0.7, 0.7
7	0.7	0.4, 0.5, 0.1, 0.9

The region R was taken as:  $0 \leq x_k \leq 1, k=1, \dots, 4$ .

20 initial points were generated, and 10 average points were used to fit the trend. The highest initial value was 11.6759, and after an extra 17 function evaluations a value of 16.525 was reached.

Contour maps were generated on two orthogonal planes. Figure 5.19 shows a map in the  $x_1, x_2$  plane with  $x_3=0.5$  and  $x_4=0.1$ , while Figure 5.20 shows a map in the  $x_3, x_4$  plane with  $x_1=0.7$  and  $x_2=0.7$ . In both cases the contour maps were generated after optimisation and the path taken by the algorithm, projected on to the appropriate plane, has been drawn. Figure 5.21 illustrates the progress of the algorithm in terms of function value versus number of evaluations. Also shown is the performance of the NAG routine EO4CGF, starting from a pair of randomly chosen initial points.

6. Price's Function (9d)

This function is described in Price (1977), and is particularly difficult to optimise because of the presence of exponential terms.

$$F_6(x_1, \dots, x_9) = \sum_{k=1}^4 (\alpha_k^2 + \beta_k^2) + (x_1 x_3 - x_2 x_4)^2. \quad (5.36)$$

where  $\alpha_k = (1-x_1 x_2) x_3 [\exp [x_5 (g_{1k} - g_{3k} \times 10^{-3} x_1 - g_{5k} \times 10^{-3} x_8)] - 1]$   
 $-g_{5k} + g_{4k} x_2$

and  $\beta_k = (1-x_1 x_2) x_4 [\exp [x_6 (g_{1k} - g_{2k} - g_{3k} \times 10^{-3} x_7 - g_{4k} \times 10^{-3} x_9)] - 1]$   
 $-g_{5k} x_1 + g_{4k}$

and the constants  $g_{ik}$  are given by the matrix:

0.485	0.752	0.869	0.982
0.369	1.254	0.703	1.455
5.2095	10.0677	22.9274	20.2153
23.3037	101.779	111.461	191.267
28.5132	111.8467	134.3884	211.4823

The region R was taken to be  $-3 \leq x_k \leq 3$ ,  $k=1, \dots, 9$ .

The logarithm of the function itself was modelled and optimised. 40 points were generated initially and 20 average points were used to fit the trend. The lowest initial value was 10.4759 (actual function value 35450.76), and after 13 extra function evaluations a value of 5.3768 (actual function value 216.33) was reached. Figure 5.22 illustrates the progress of the optimisation algorithm. For comparison purposes, the NAG routine EO4CGF was started from a number of randomly chosen starting points. Also shown on Figure 5.22 is the progress of this routine starting from a "good" point and a "bad" point.

### Summary of Results

Table 5.1 summarises the parameters of the interpolating functions fitted to the six test functions. To compare efficiency of optimisation with a standard technique (in terms of number of function evaluations), each of these functions has also been optimised using the NAG library routine EO4CGF, starting from the best of the  $N$  initial points.  $N_E$  is the number of extra function evaluations required by the experimental technique to reach a conclusion, and  $N_N$  is the number of function evaluations required to reach the same level of the objective function by EO4CGF. These results are tabulated in Table 5.2.

Discussion of the total computer time used by the optimisation algorithm is made difficult by the fact that some means must be found to compare algorithms running on different machines in different languages. Therefore, a suggestion of Dixon & Szegö (1978, p.2-3) has been adopted, and a "standard" time computed against which other times may be computed. The results for our test functions may also be compared with those given in Dixon & Szegö (1978), p.9-10. Table 5.3 contains these times, both in mill units and standardised, for both SIMP and the NAG routine EO4CGF (the latter's time relating only to the optimisation stage starting from the best point of SIMP's initial  $N$  trials, while the former's time includes the initial  $N$  trials).

The optimisation algorithm using the stochastic interpolating function concept appears to be effective, at least in terms of total function evaluations. It seems to give better results than standard techniques in higher dimensions, where the overheads involved in estimating derivatives are greater. The stochastic interpolating function gets round this

problem by estimating the derivatives from the existing data points, rather than requiring further function evaluations.

This work has concerned itself mainly with reaching a true function value which is consistent with an optimum value of the interpolating function. This does not guarantee, of course, that it is in fact an optimum value of the true function. Two possibilities exist for verifying the location of a true optimum value:

- a) Centre a new, smaller region of interest  $R'$  around the point found by the algorithm, and generate a new set of points spread throughout  $R'$ , including any of the existing data points which lie inside  $R'$ . Refit the interpolating function inside the smaller region and repeat the optimisation algorithm.
- b) Use a conventional local optimisation algorithm on the true function, starting from the estimated location of the optimum point.

TABLE 5.1

Summary of Interpolating Functions Fitted to Test Data

FUNCTION	m	N	n <sub>A</sub>	$\mu$	$\rho_L$	$\rho_s$	$\sigma$	$\sigma_R$
1. Simple Test	2	10	-	-0.2526	-	1.0	-	-
2. Branin's RCOS	2	20	12	31.593	3.0415	1.699	46.94	30.85
3. Goldstein & Price	2	40	20	10.22 10	0.527	0.276	17.693 $\times 10^5$	77.18 $\times 10^4$
4. Banana Valley	2	20	10	5472.7	1.336	0.356	13.59 $\times 10^3$	8.14 $\times 10^3$
5. Shekel	4	20	10	8.633	0.364	0.251	1.605	1.079
6. Price (Log)	9	40	20	12.994	2.830	0.429	0.925	0.838

- m - Number of variables  
 N - Number of initial points generated  
 n<sub>A</sub> - Number of average points for trend  
 $\mu$  - Grand mean fitted  
 $\rho_L$  - Correlation distance for trend  
 $\rho_s$  - Correlation distance for residual  
 $\sigma$  - Standard deviation of initial data  
 $\sigma_R$  - Standard deviation of residuals



TABLE 5.2

Results of Optimisation of Test Functions

FUNCTION	m	N	$f_I$	$f_F$	$N_E$	$N_N$
1. Simple Test	2	10	0.3296	0.3417	4	-
2. Branin's RCOS	2	20	2.7060	1.2754	10	14
3. Goldstein & Price	2	40	547.19	6.266	13	11
4. Banana Valley	2	20	49.975	0.0219	8	9
5. Shekel	4	20	11.676	16.525	17	89
6. Price (Log)	9	40	10.476	5.377	14	76

m - Number of variables

N - Number of initial points generated

$f_I$  - Best function value from initial points

$f_F$  - Best function value at end of algorithm

$N_E$  - Number of extra function evaluations to reach  $f_F$

$N_N$  - Number of function evaluations by EO4CGF to reach  $f_F$

TABLE 5.3

Timing of Optimisation Results

FUNCTION	SIMP		EO4CGF	
	MILL UNITS	STANDARDISED	MILL UNITS	STANDARDISED
1. STANDARD FUNCTION	21	1.0	58	1.0
2. Branin's RCOS	208	10	3	0.05
3. Goldstein & Price	668	32	4	0.07
4. Banana Valley	108	5	29	0.5
5. Shekel	384	18	12	0.2
6. Price (log)	1500	71	36	0.6

*(Standard function consists of 1000 evaluations of a particular Shekel function - see Dixon & Szegő, 1978, p.2-3).*

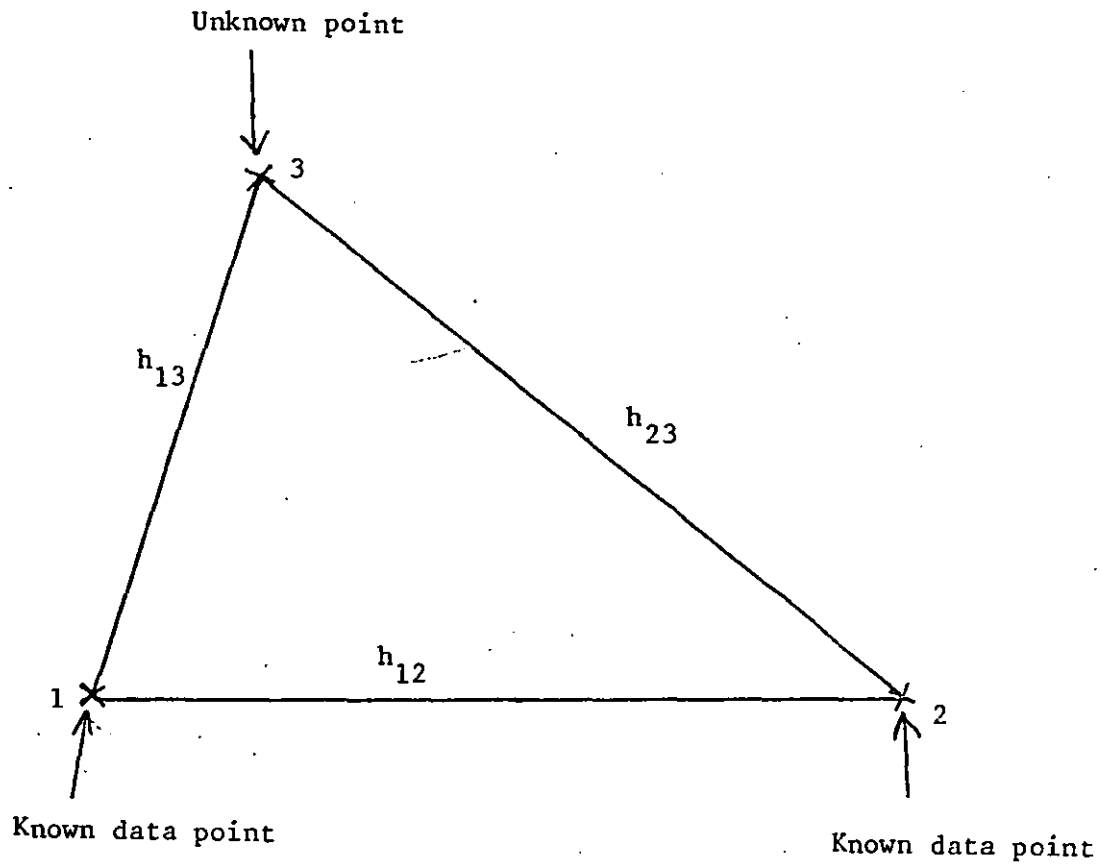


FIGURE 5.1: Configuration of two known data points and one unknown point in close proximity

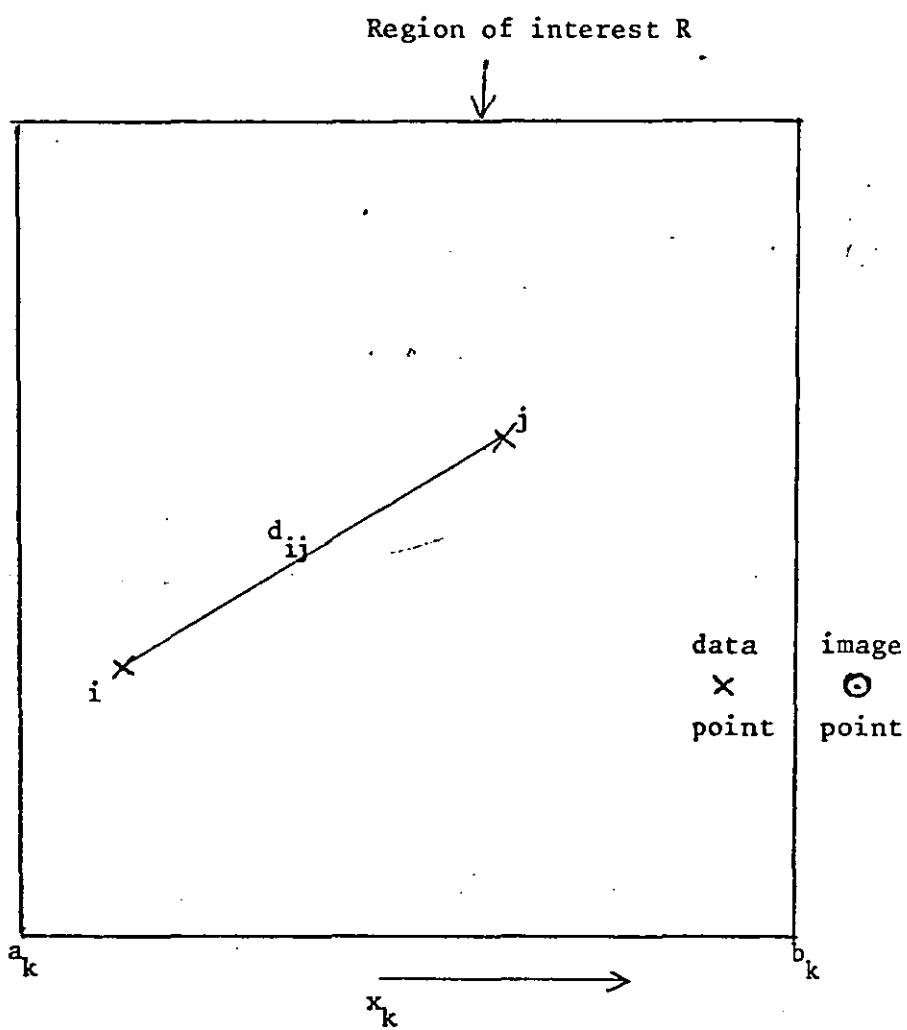
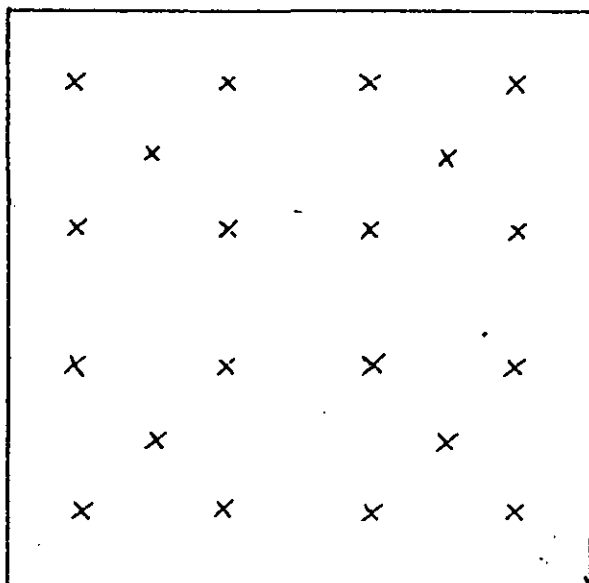
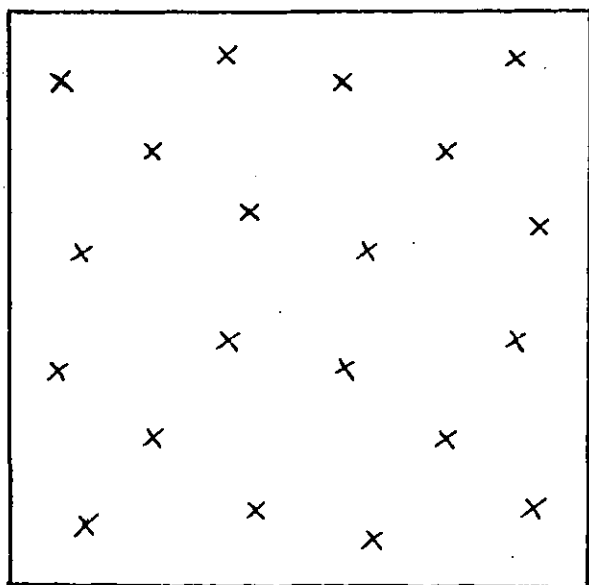


FIGURE 5.2: Illustration of components of repulsive function

$N = 20$  $m = 2$ 

INITIAL DISTRIBUTION  
OF POINTS

$H = 14.7284$



POINTS AFTER ADJUSTMENT

$H = 12.9927$

FIGURE 5.3: Example of distribution of initial points

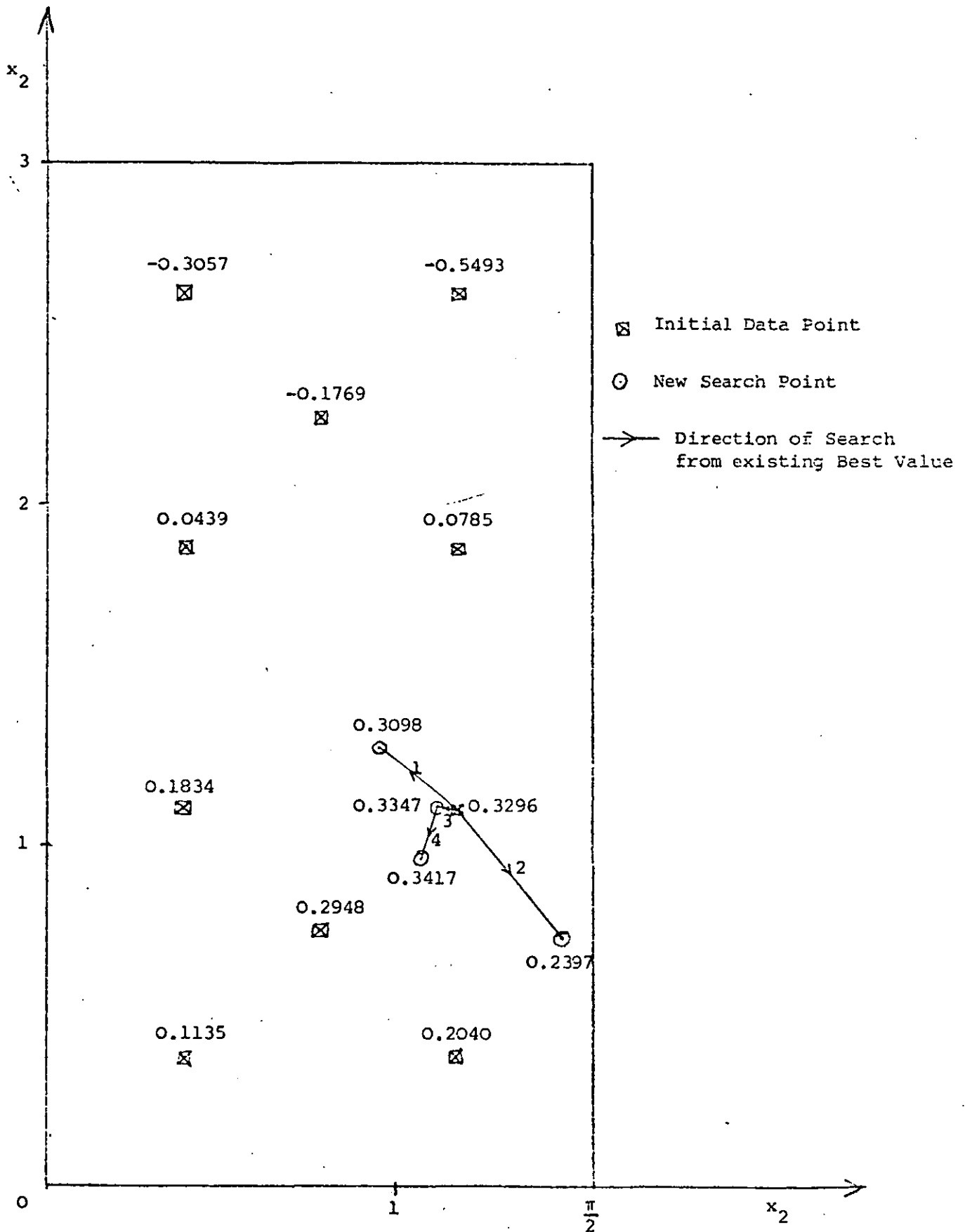


FIGURE 5.4: Test function 1 with 10 initial points

## BRANIN'S RCOS.

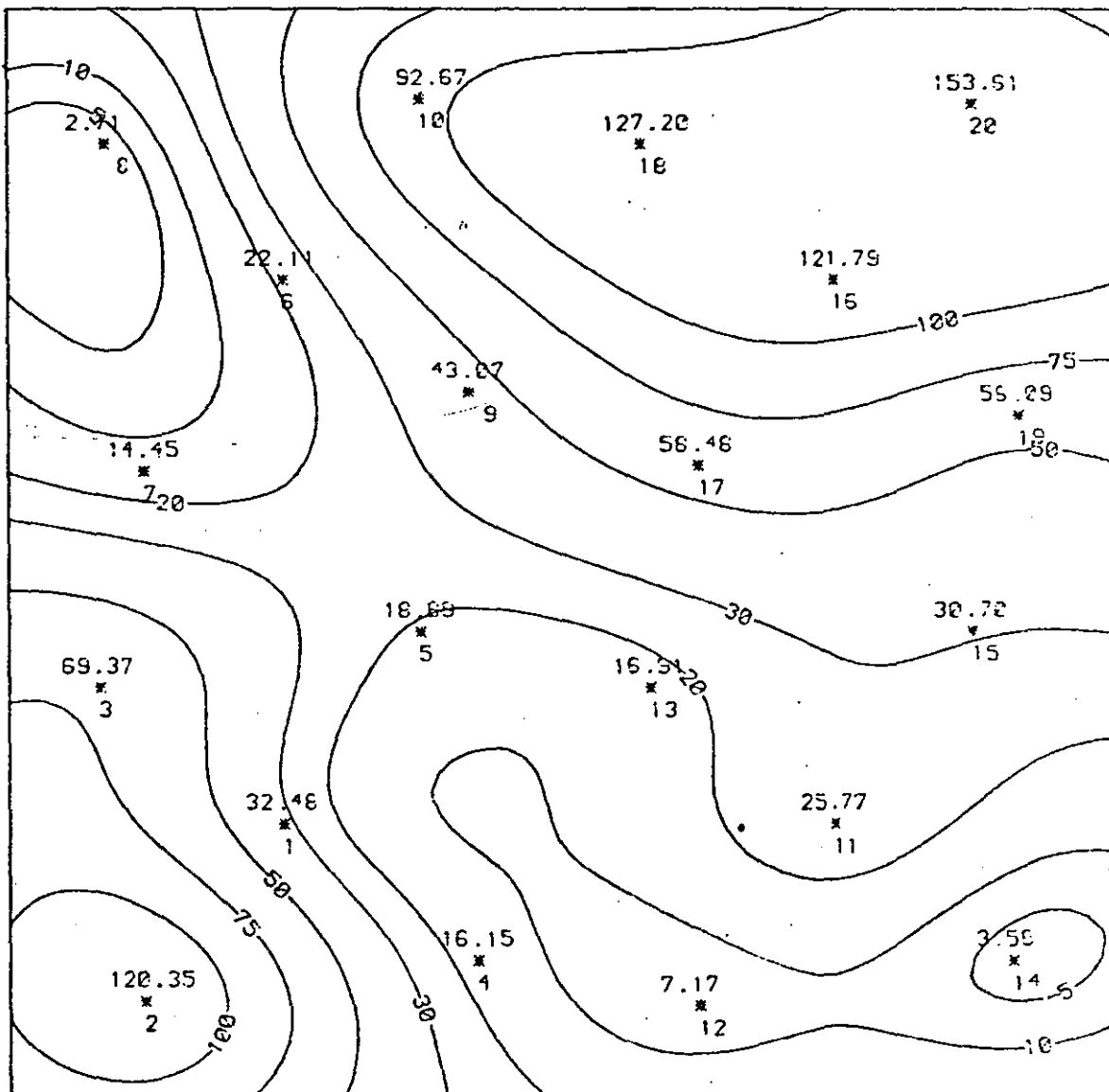


FIGURE 5.5: Branin's RCOS - interpolating function contours

## BRANIN'S RCOS (TREND) . .

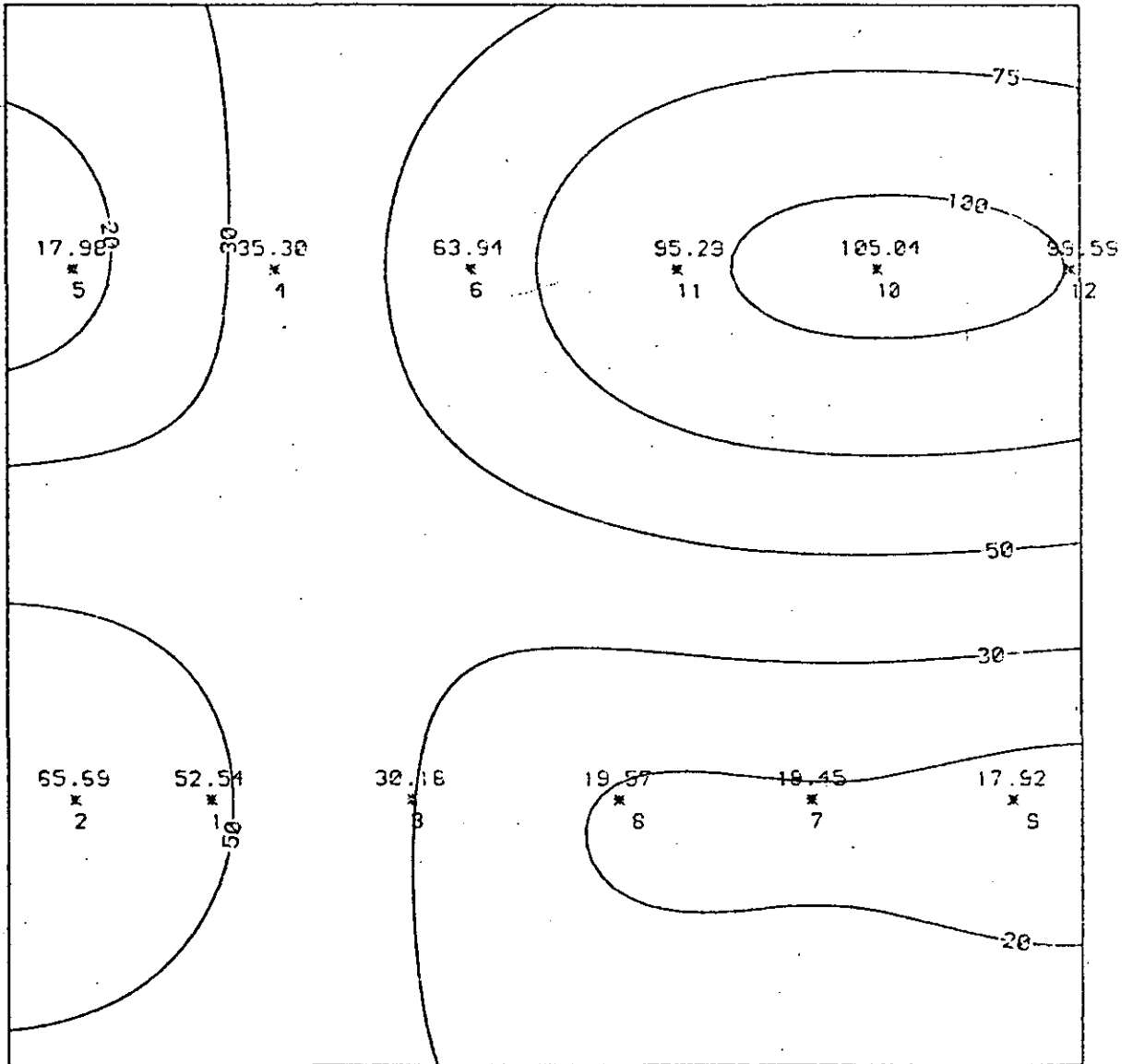


FIGURE 5.6: Branin's RCOS - trend function contours



BRANIN'S RCOS (BOTTOM LEFT TO TOP RIGHT).

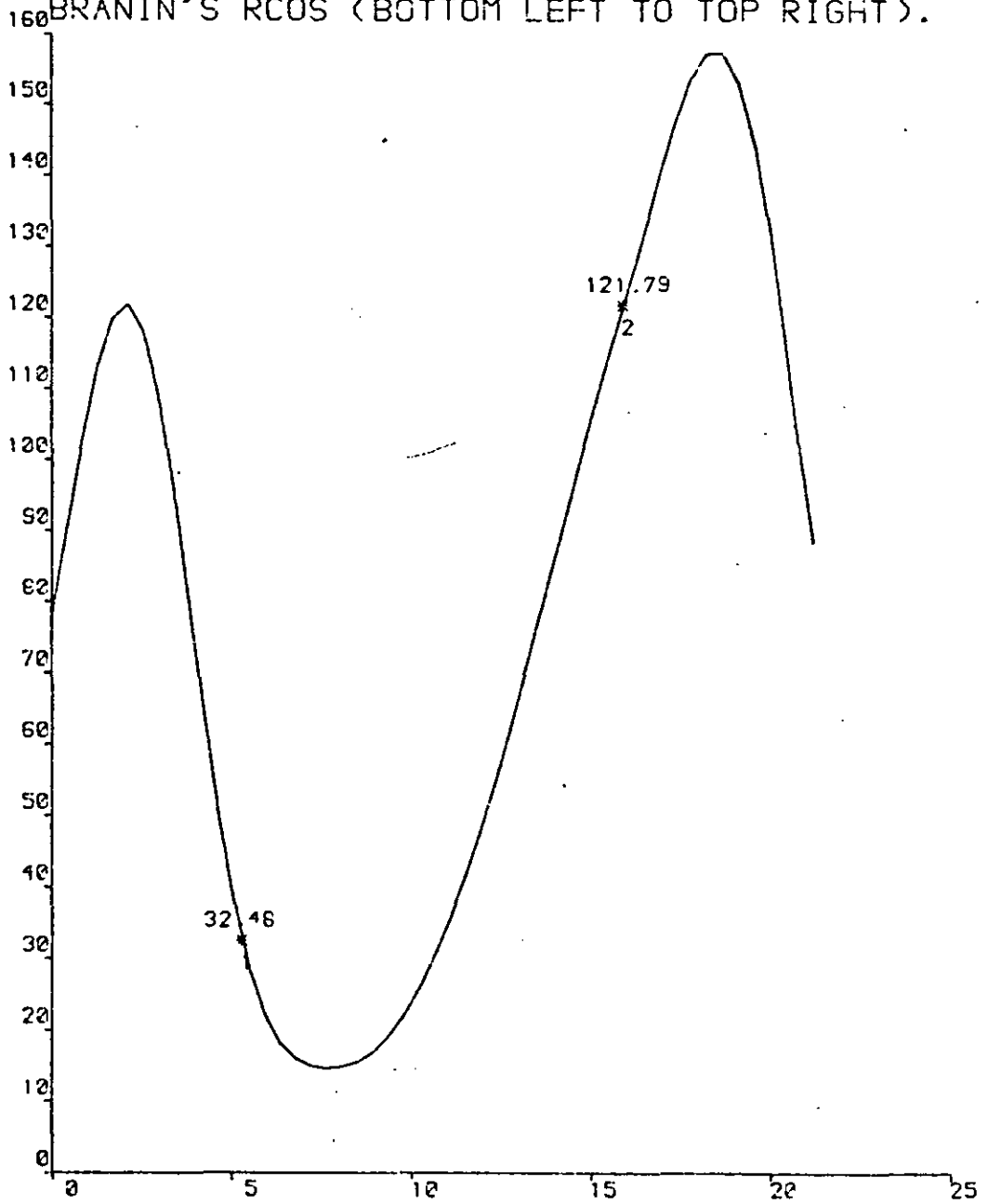


FIGURE 5.7: Branin's RCOS - section through interpolating function

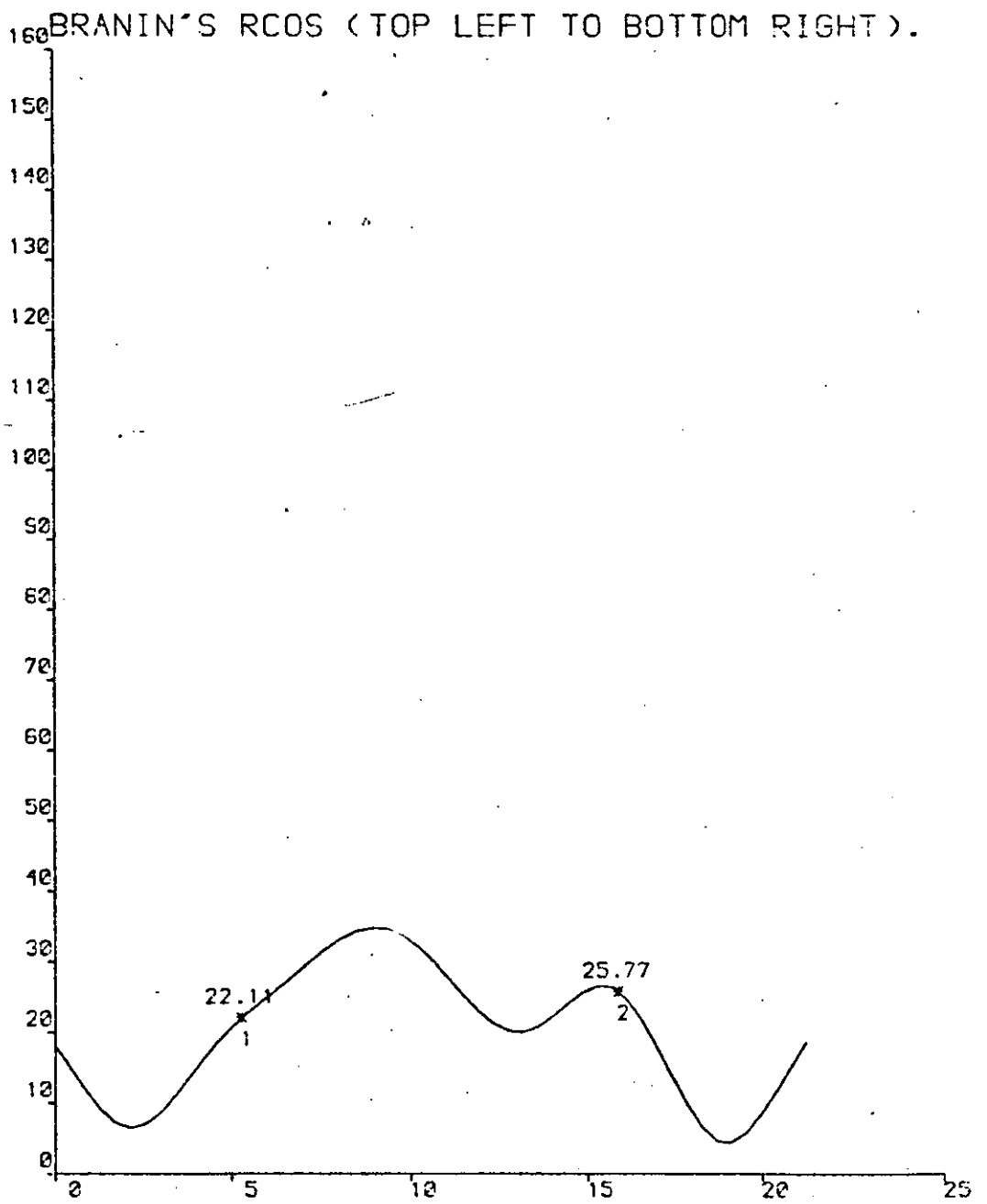
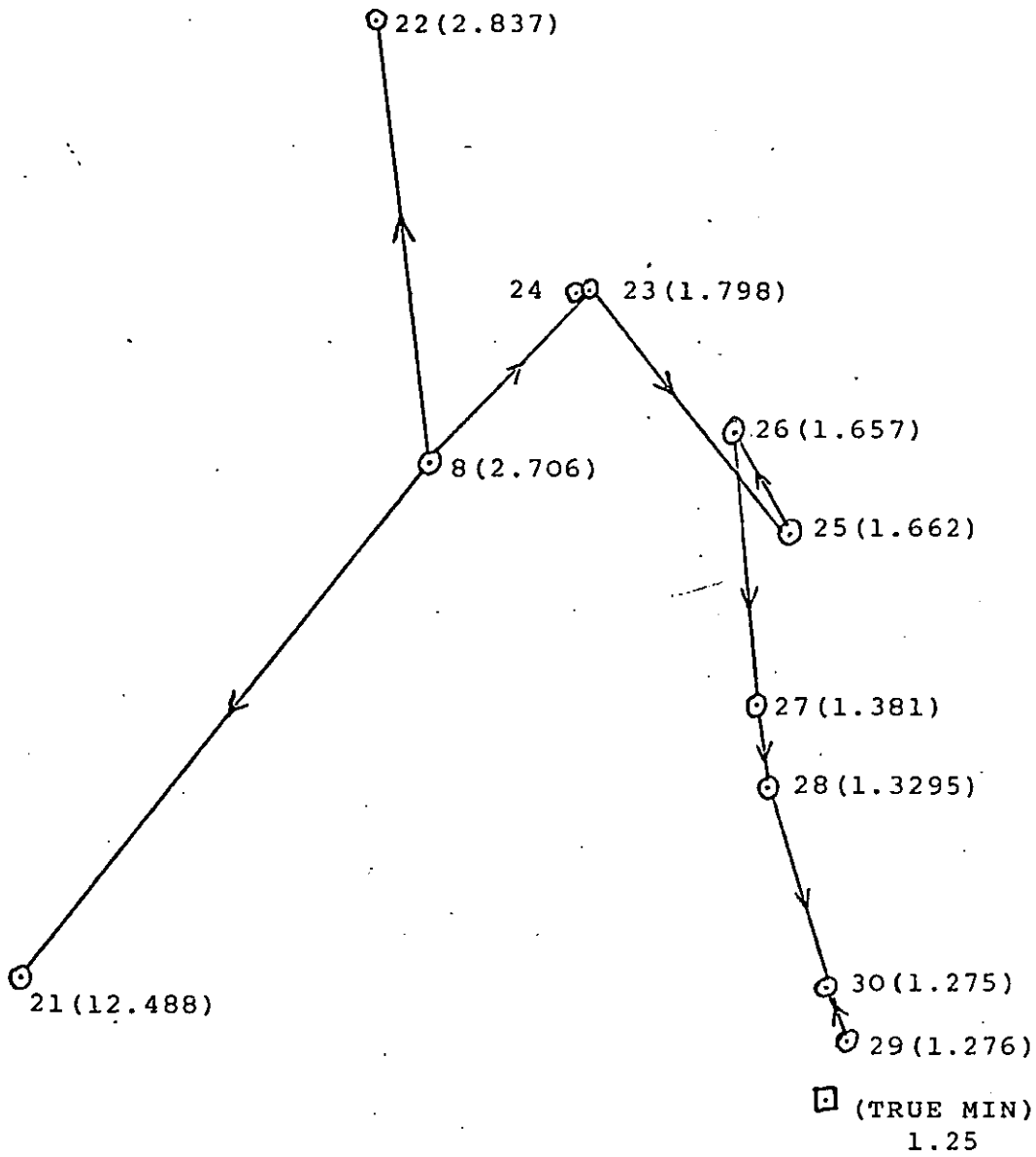


FIGURE 5.8 : Branin's RCOS - section through interpolation function

BRANIN'S RCOS

PATH TAKEN BY OPTIMISATION ALGORITHM

FIGURE 5.9

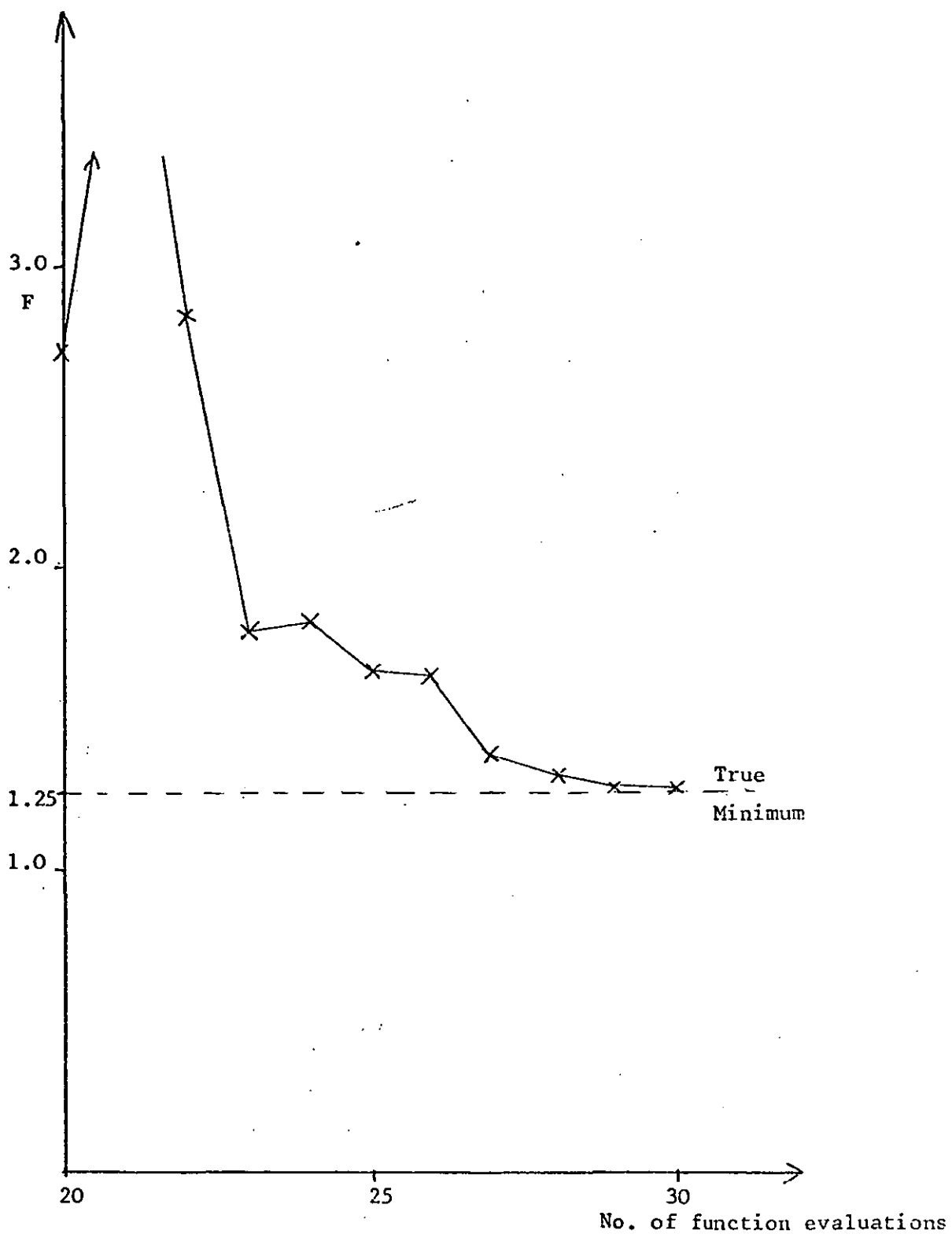
Branin's RCOS

FIGURE 5.10: Convergence of algorithm to true minimum

GOLDSTEIN & PRICE.

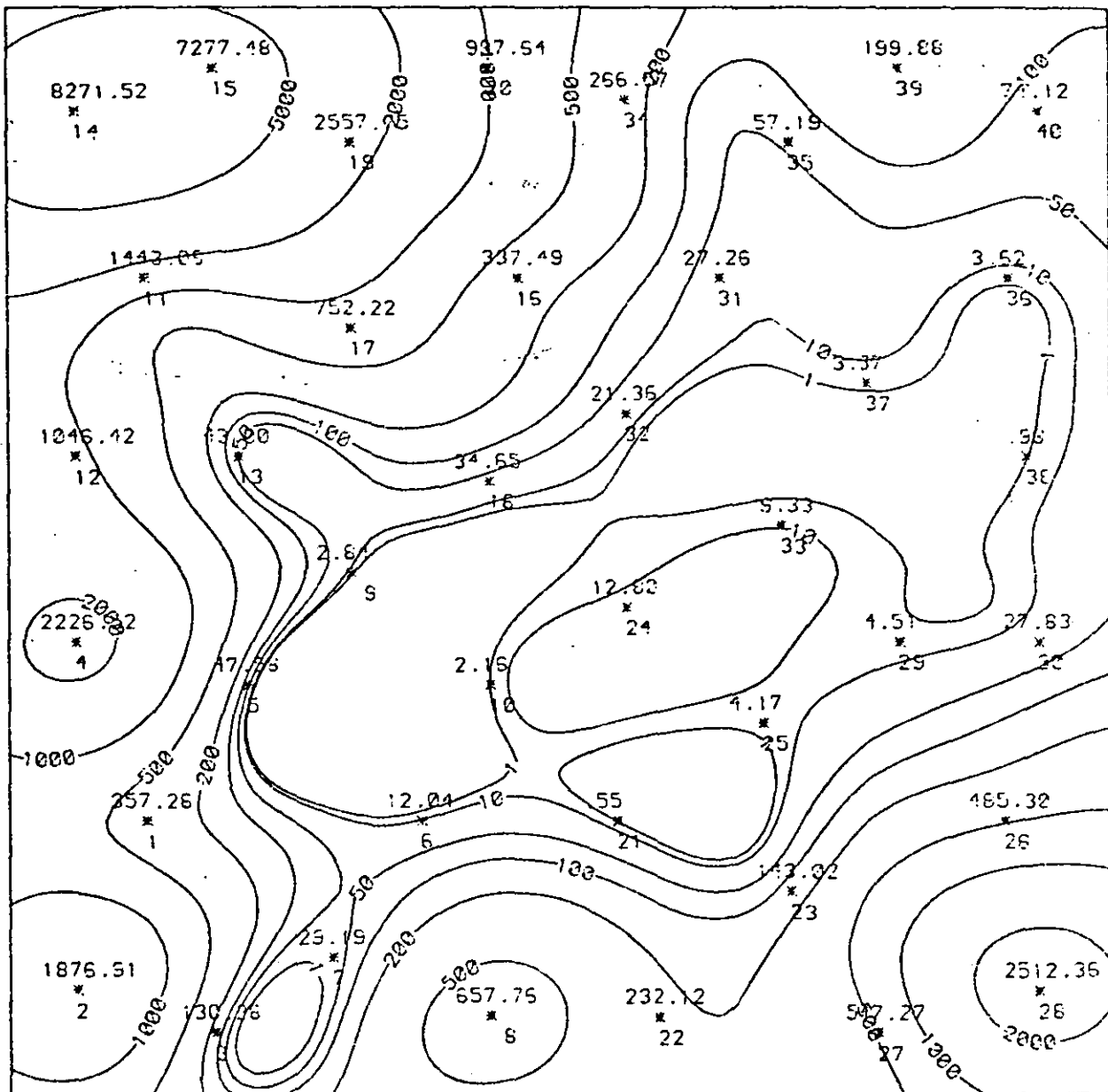


FIGURE 5.11: Goldstein & Price - interpolating function contours

GOLDSTEIN & PRICE (TREND).

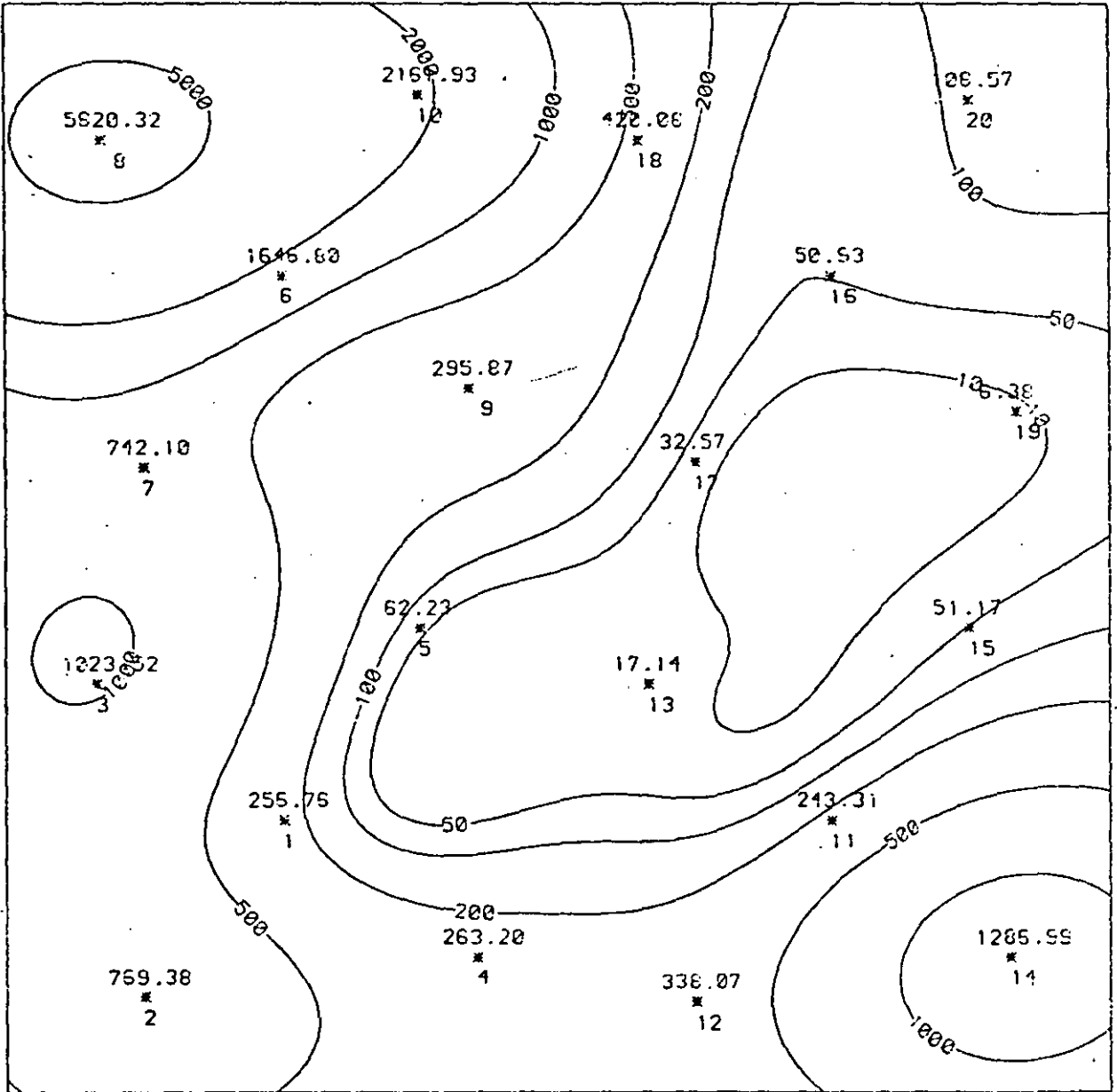


FIGURE 5.12: Goldstein & Price - trend function contours

G & P (TOP LEFT TO BOTTOM RIGHT).

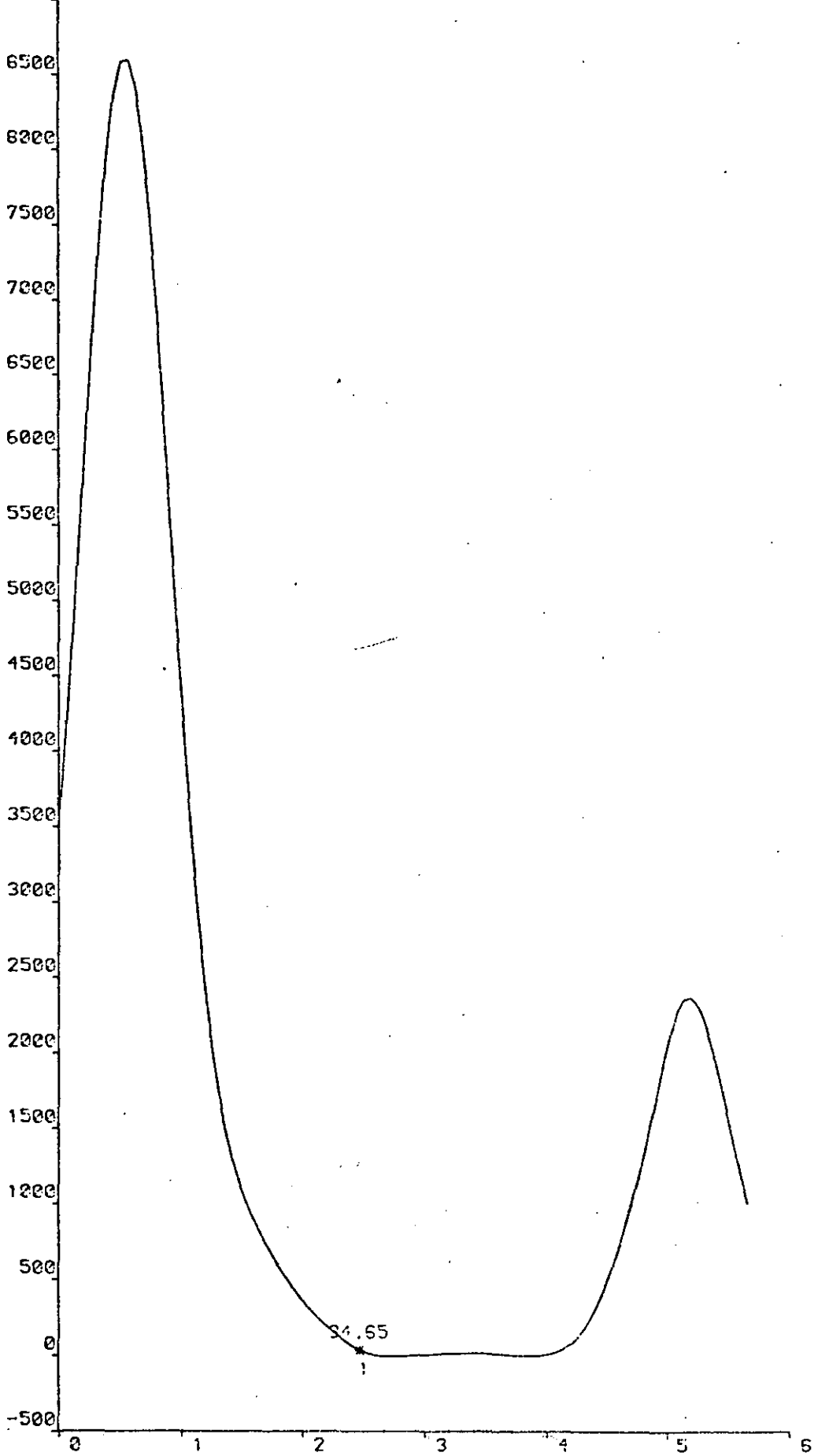


FIGURE 5.13: Goldstein & Price - section through interpolation function

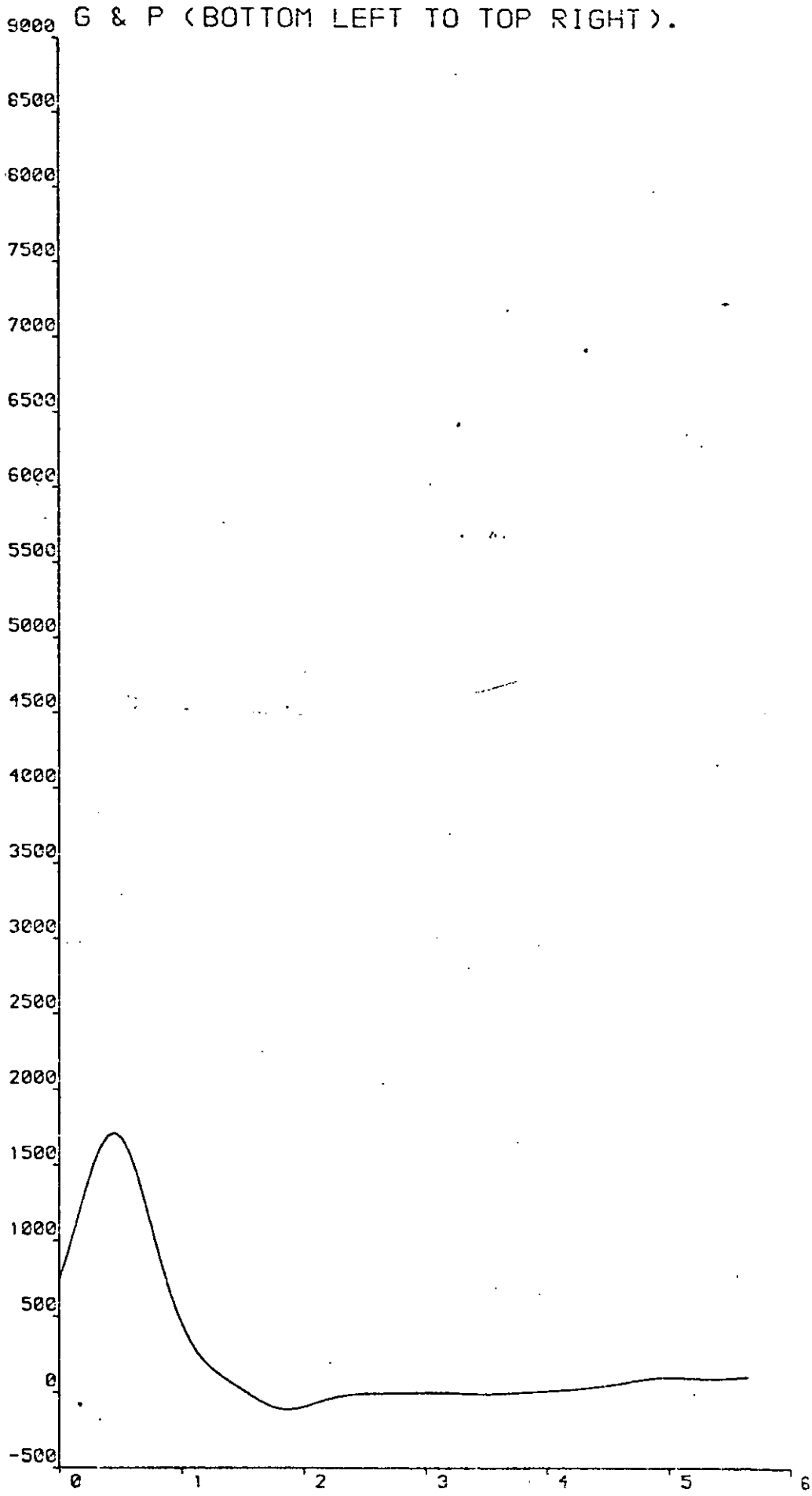
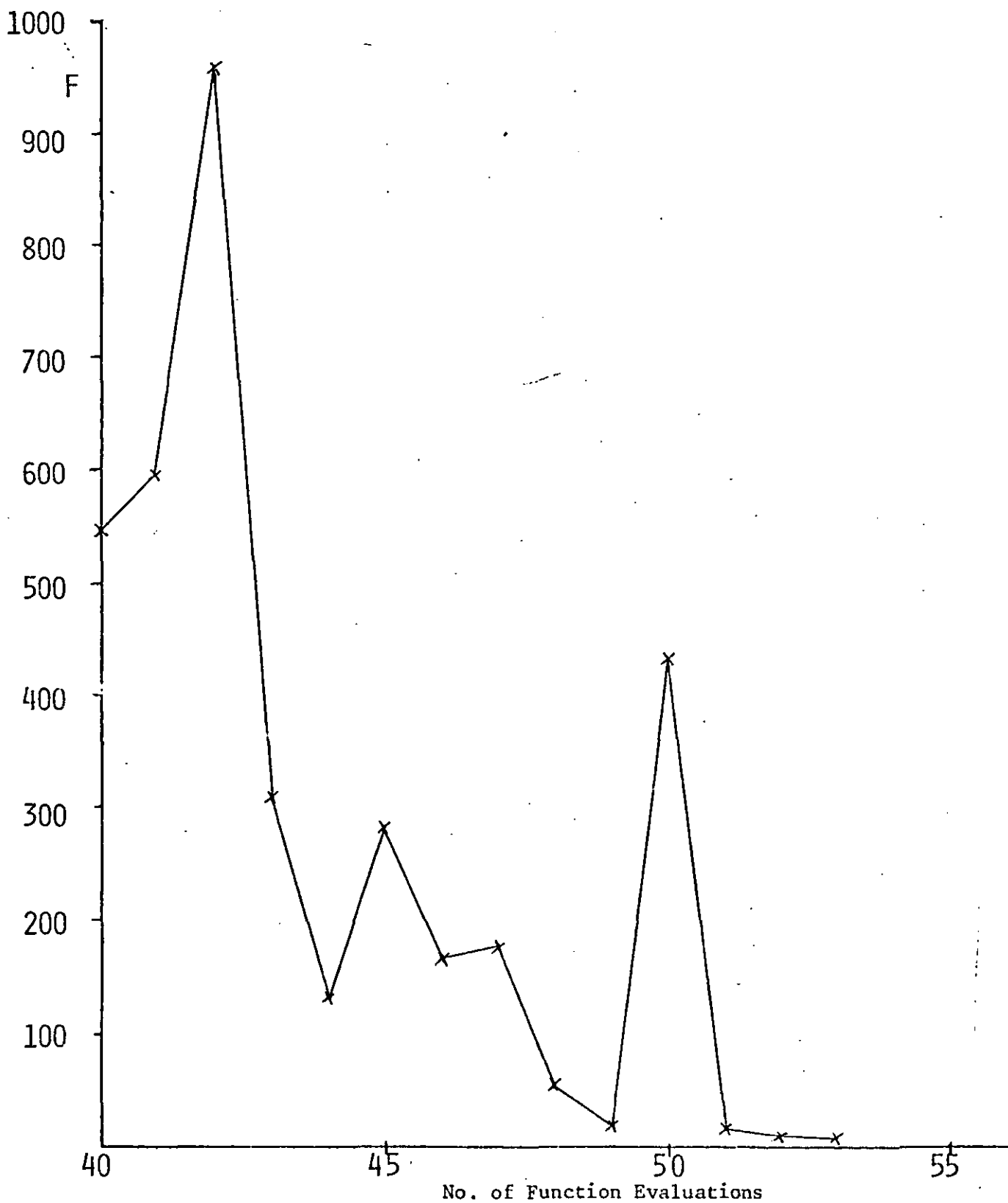


FIGURE 5.14: Goldstein & Price - section through interpolating function



GOLDSTEIN AND PRICEFIGURE 5.15

## BANANA VALLEY FUNCTION.

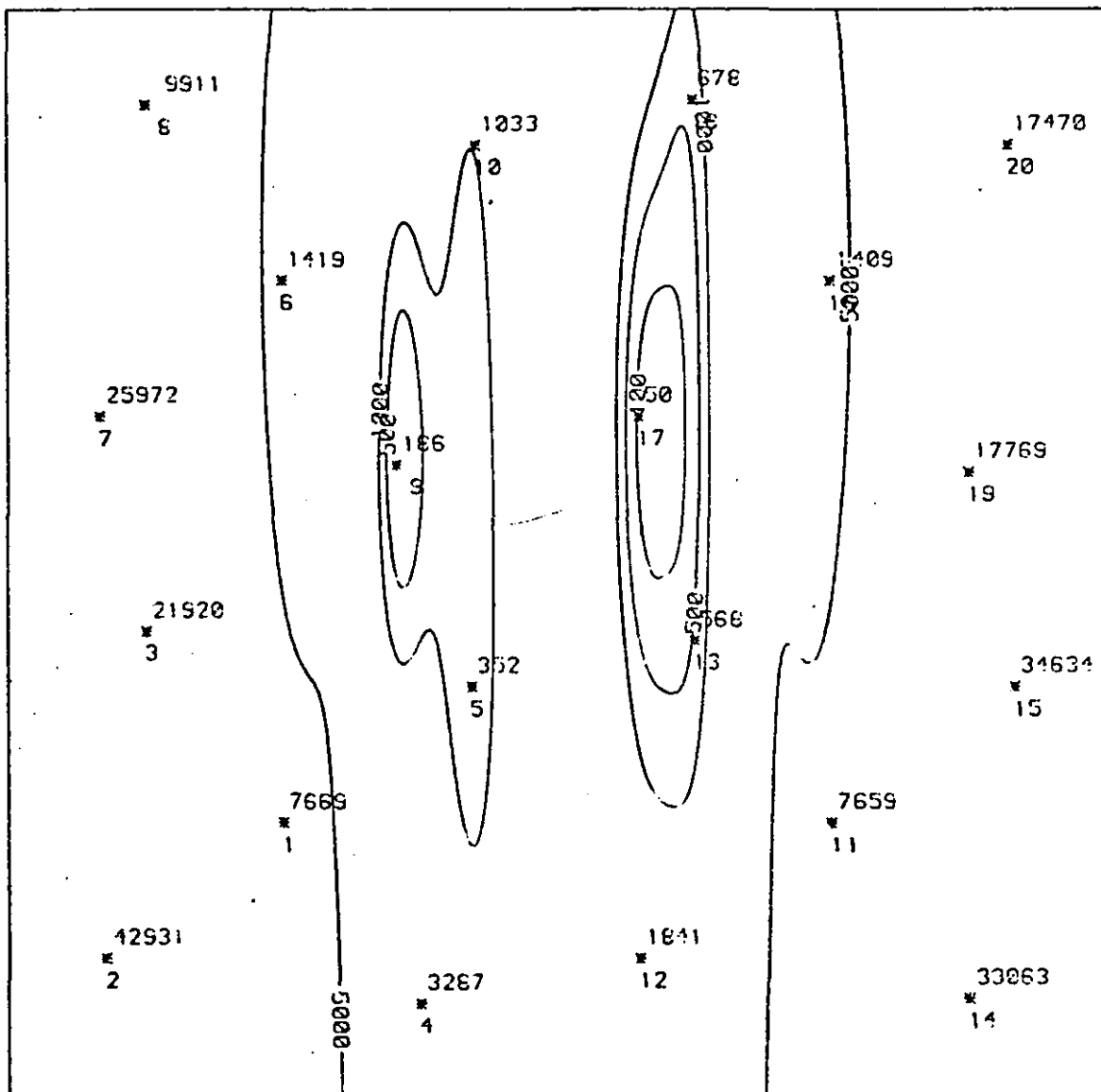


FIGURE 5.16: Banana Valley interpolating function contours

SECTION AFTER OPTIMISATION.

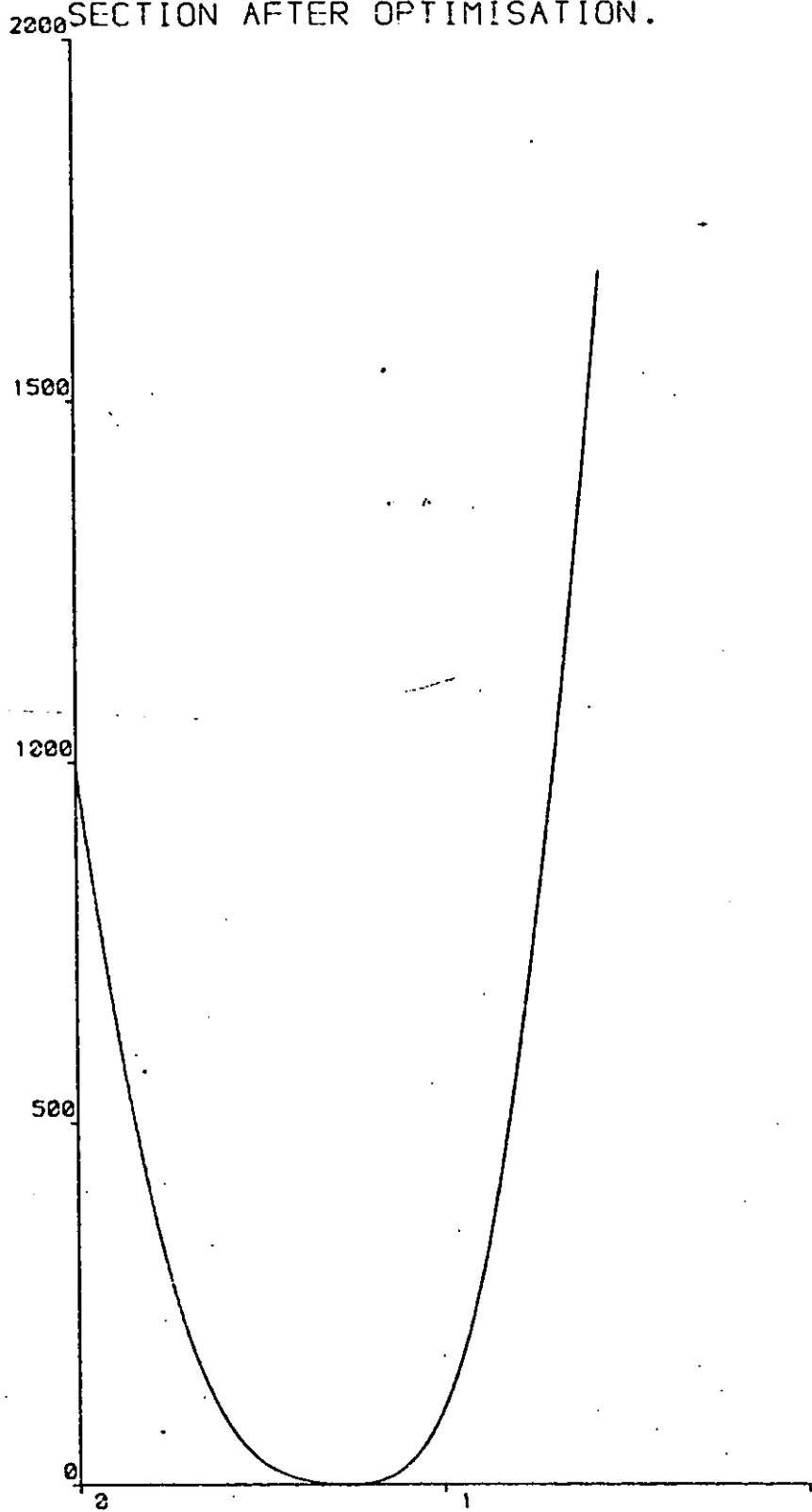


FIGURE 5.17: Banana Valley cross-section along tangent to valley after optimisation

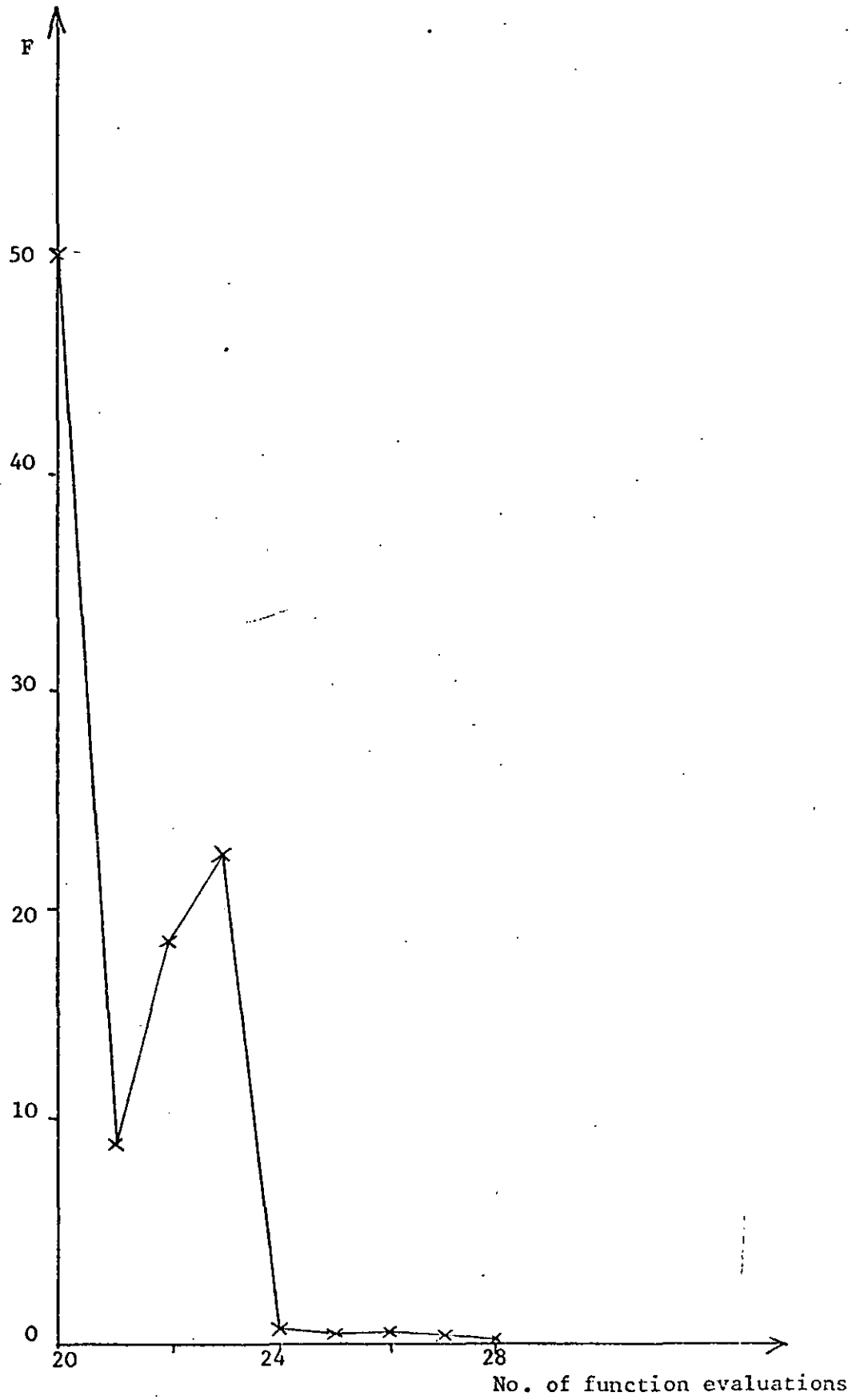


FIGURE 5.18: Banana Valley progress of optimisation

4-D SHEKEL  $(x_1, x_2)$ .

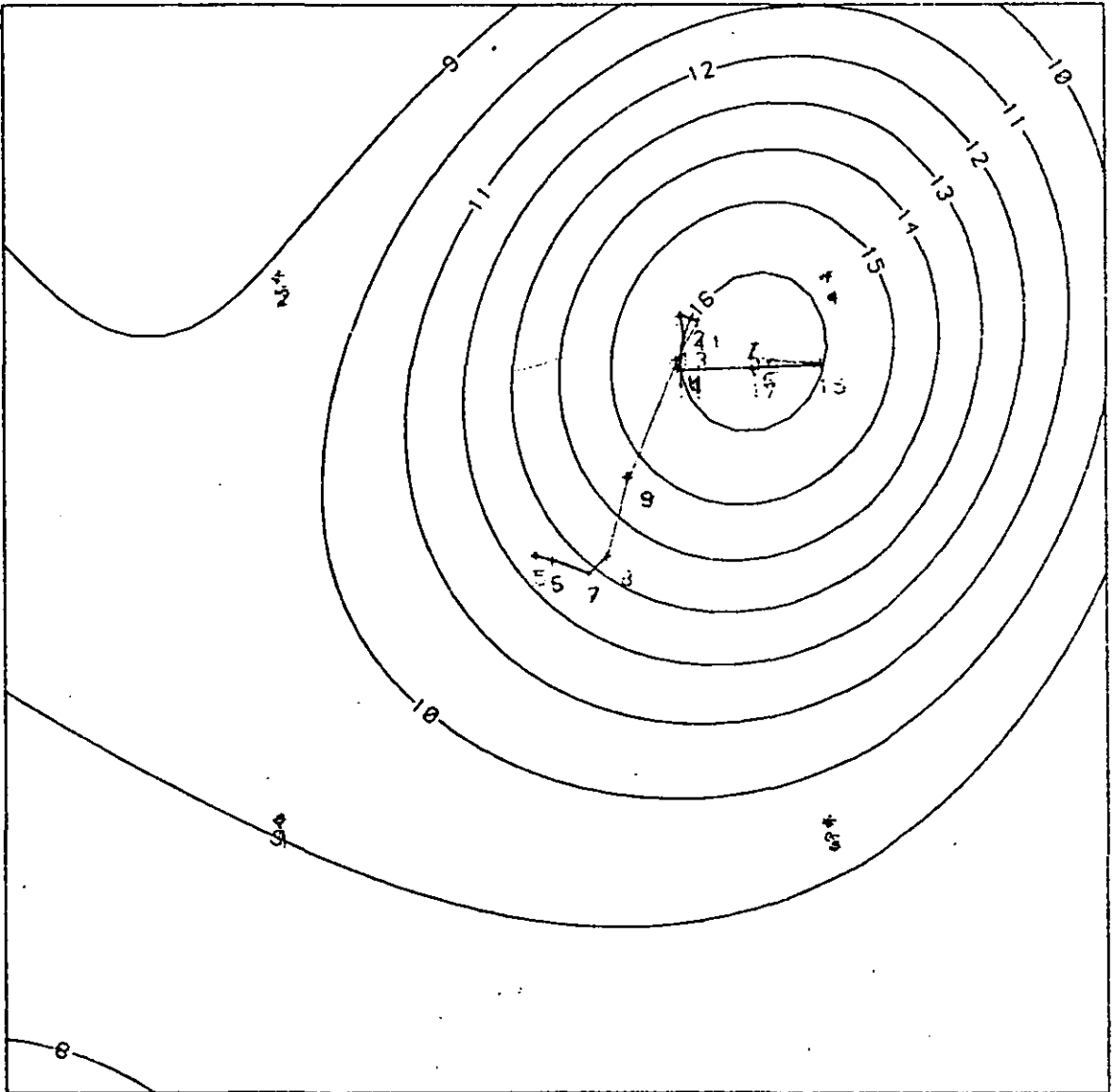
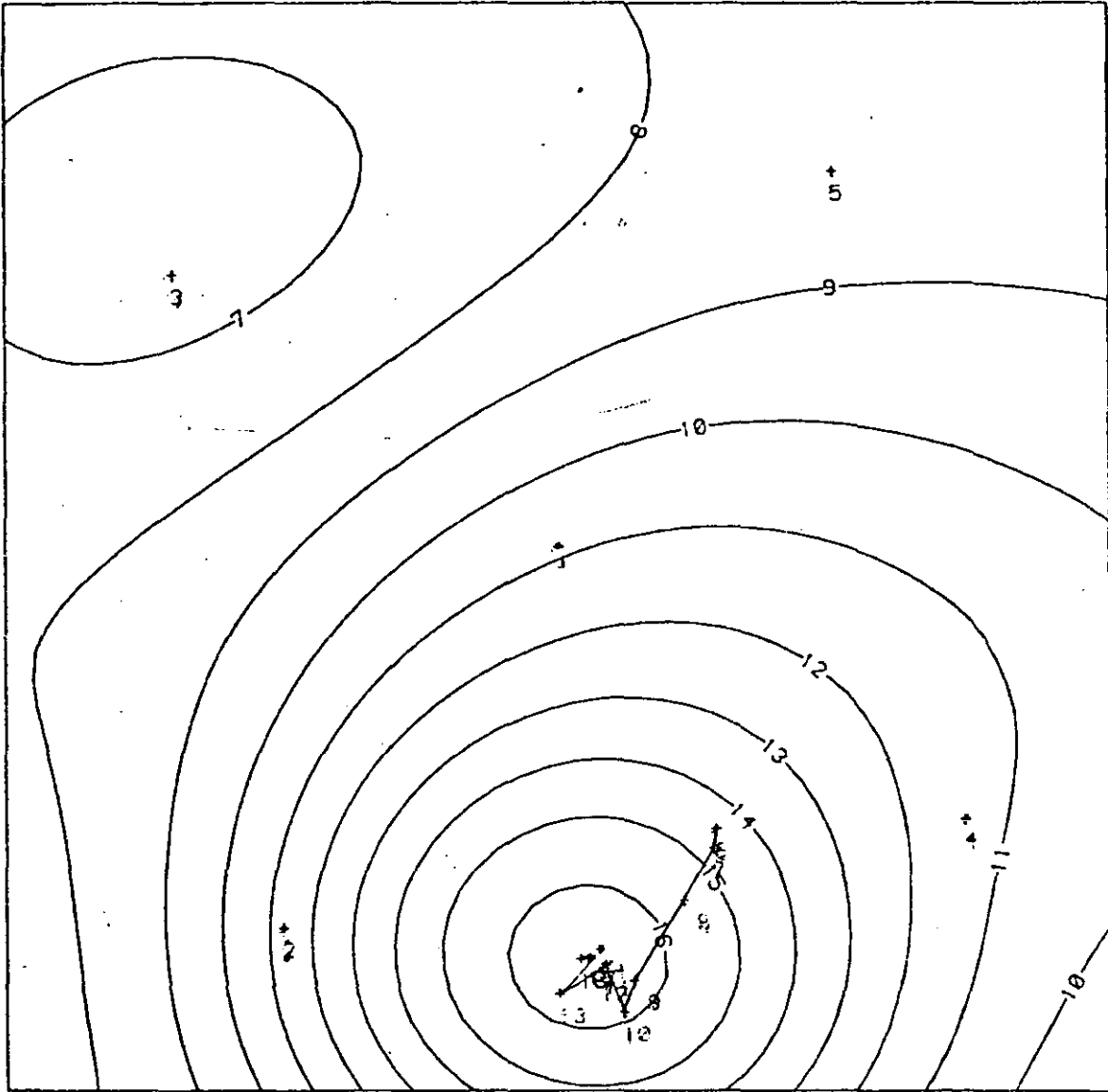
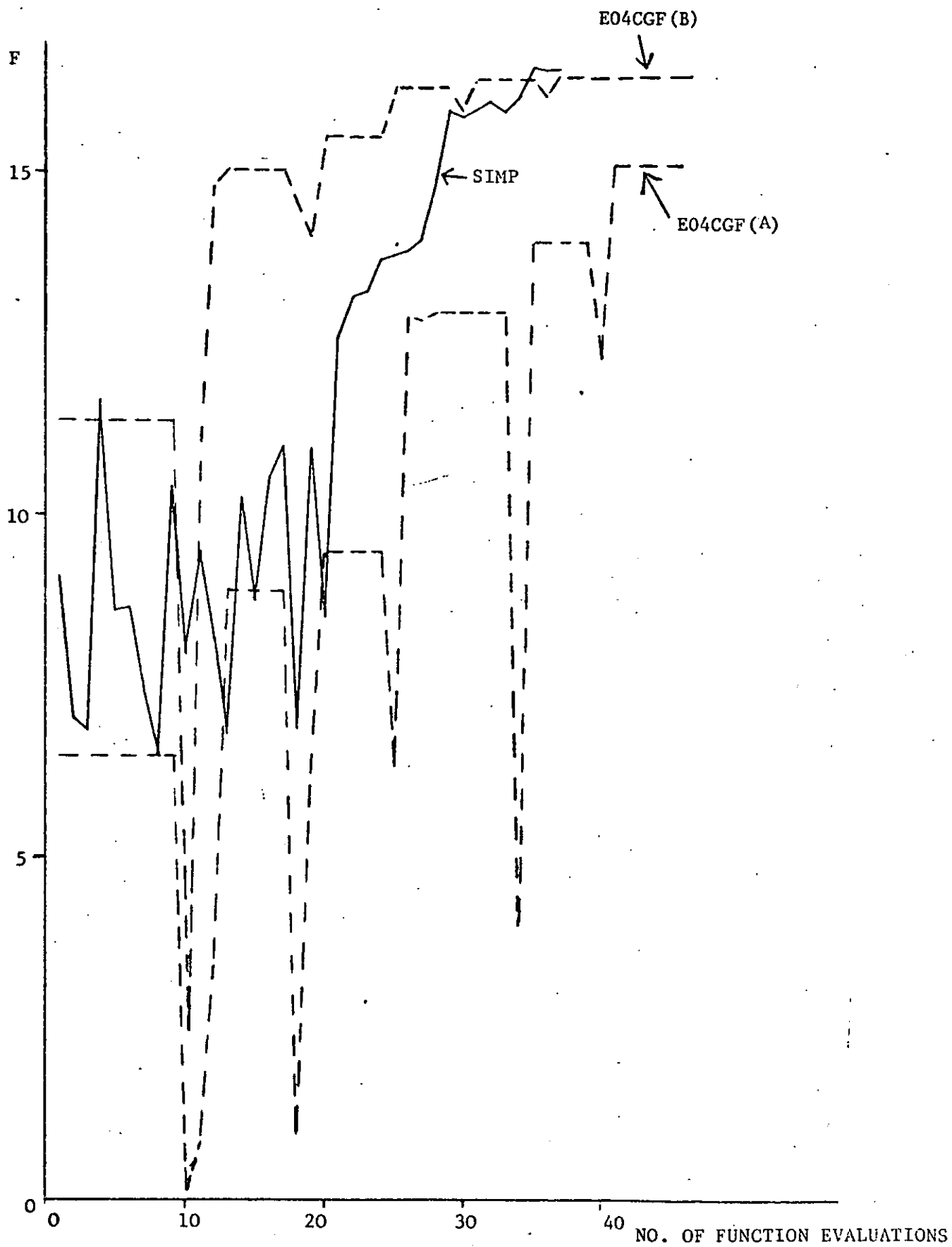


FIGURE 5.19: Shekel function - interpolating function contours in  $(x_1, x_2)$  plane

4-D SHEKEL (X3, X4)



**FIGURE 5.20:** Shekel function - interpolating function contours in  $(x_3, x_4)$  plane

4-d SHEKELFIGURE 5.21

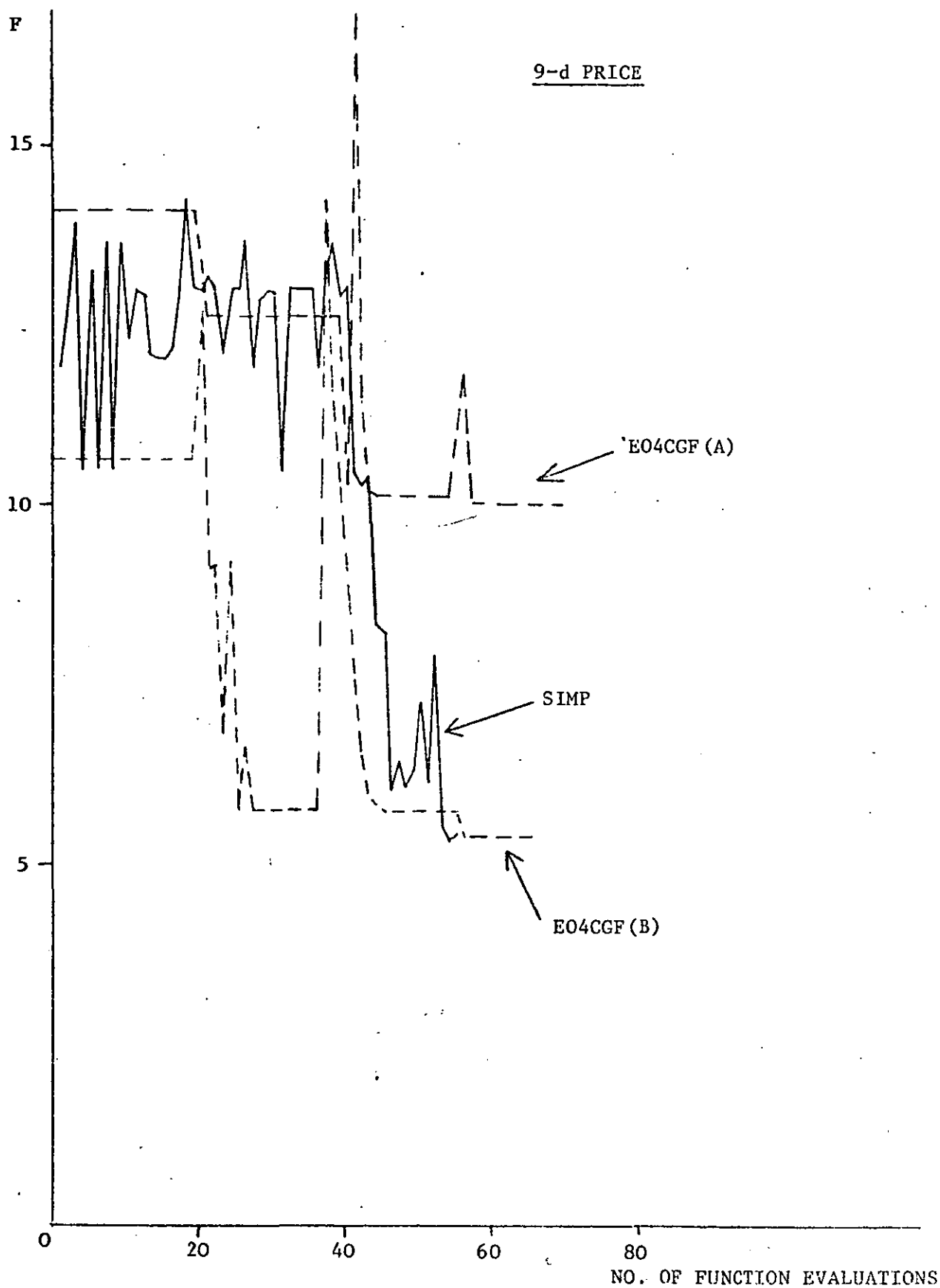


FIGURE 5.22



CHAPTER 6

A MODEL FOR THE OCCURRENCE OF OILFIELDS

## 6.1 DESCRIPTION OF THE BASIC MODEL

Consider an oil province, or region of interest, which has total area  $\Omega$ . Let  $Z(\underline{x})$  be a stationary, homogeneous and isotropic random process defined over the region. We shall suppose that an oilfield is a connected subset of the region of interest over which  $Z(\underline{x}) \geq u$ , some fixed limit. Conceptually,  $Z(\underline{x})$  may be assumed to represent "potential oil reserves per unit area", and these become classified as actual reserves once they exceed a given threshold. (See Figure 6.1).

Let  $f_z(\cdot)$  be the probability density function of  $Z(\underline{x})$  at any fixed point, and

$$R(z) = \int_z^{\infty} f_z(v) dv = 1 - F(z) . \quad (6.1)$$

For this basic model, the following random variables will be of interest:

1. The number of oilfields ( $N$ ) contained within the region  $\Omega$ .
2. The area ( $A$ ) of a "randomly selected" oilfield.
3. The volume ( $V$ ) of a "randomly selected" oilfield.

## 6.2 EXPECTATIONS OF OILFIELD VARIABLES

To generate even approximate formulae for the expectations of oilfield random variables we shall need to consider carefully what we mean by a "randomly selected" oilfield. We shall suppose that the  $N$  oilfields in  $\Omega$  have areas  $A_1, \dots, A_N$ , and that we "randomly select" an oilfield by generating a uniform random integer between 1 and  $N$ .

If we repeat this experiment a sufficiently large number of times, the "average area" so obtained will tend to the value  $\sum A_i / N$ . It is thus tempting to relate this to the "expected area" of an oilfield in some way. R.E. Miles (1974, p.202ff) defines an "ergodic distribution" - in the same way, if we let  $\Omega$  increase in area, we may define the "ergodic expectation"

$$E[A] = \lim_{\Omega \rightarrow R^2} \frac{\sum A_i}{N} \quad (6.2)$$

Define  $A(\Omega)$  to be the total area of the region  $\Omega$ , and then

$$\lim_{\Omega \rightarrow R^2} \frac{\sum A_i}{N} = \frac{\lim_{\Omega \rightarrow R^2} \sum A_i / A(\Omega)}{\lim_{\Omega \rightarrow R^2} N / A(\Omega)} \quad (6.3)$$

Define  $T$  to be the area inside  $\Omega$  above the level  $Z(\underline{x}) = a$  per unit (gross) area of  $\Omega$ , and  $n$  to be the number of oilfields per unit (gross) area of  $\Omega$ . Then using multidimensional ergodic theorems (see e.g. Wiener, 1939, p.1-18 and Adler, 1981, p.142ff) we can see that

$$\lim_{\Omega \rightarrow R^2} \frac{\sum A_i}{A(\Omega)} = E[T] \quad (6.4)$$

and

$$\lim_{\Omega \rightarrow R^2} \frac{N}{A(\Omega)} = E[n] \quad (6.5)$$

Thus the "ergodic expectation" of the area of a "randomly selected"

oilfield is given by

$$E[A] = \frac{E[T]}{E[n]} \quad (6.6)$$

To derive an expression for  $E[T]$  we must define an indicator variable  $X(\underline{x})$  such that

$$\begin{aligned} X(\underline{x}) &= 0 \quad \text{if } Z(\underline{x}) < u \\ &= 1 \quad \text{if } Z(\underline{x}) \geq u \end{aligned}$$

Assume that  $\Omega$  is a unit square, whence

$$E[T] = E \int_{\Omega} X(\underline{x}) d\underline{x} = \int_{\Omega} P[Z(\underline{x}) \geq u] d\underline{x} = R(u) \int_{\Omega} d\underline{x} = R(u) \quad (6.7)$$

To obtain an expression for  $E[n]$  is not so straightforward.

Adler (1981, p.70ff) has shown that the extension to two or more dimensions of the theory of "level-crossings" in one dimension is non-trivial. He defines the "IG characteristic"  $\Gamma$  of an excursion set in the more general case, and shows that in two dimensions the expectation of  $\Gamma$  for a zero-mean, homogeneous, Gaussian process is (p.115):

$$E[\Gamma] = (2\pi\sigma^2)^{-3/2} \sigma_{\omega}^2 u e^{-u^2/2\sigma^2} \quad (6.8)$$

where  $\sigma_{\omega}^2$  is the variance of the gradient of  $Z(\underline{x}) = -\sigma^2 g''(0)$ .

It appears from Adler's point set representation for  $\Gamma$  (see p.78ff) that this will give a good approximation to  $n$ , the number of oilfields (excursion sets) in a unit area. The main difference will arise when sets with "holes" appear (see Figure p.76). It would thus seem reasonable to take as an approximation:

$$E[n] \approx (2\pi\sigma^2)^{-3/2} \sigma_{\omega}^2 u e^{-u^2/2\sigma^2} \quad (6.9)$$

from which it follows that

$$E[A] \approx \frac{(2\pi\sigma^2)^{3/2} R(u)}{\sigma_{\omega}^2 u e^{-u^2/2\sigma^2}} \quad (6.10)$$

As well as areas and numbers of oilfields, we are also interested in the reserves of oil contained in them. To include this in the stochastic model requires that we give a physical interpretation to the variable  $Z(\underline{x})$ . We shall do this by describing  $Z(\underline{x})$  as the "potential oil reserves per unit area", and assuming that "potential reserves" only become "actual reserves" once they exceed a certain threshold  $u$ . That is to say, the reserves of oil per unit area =  $Z(\underline{x})$  if  $Z(\underline{x}) \geq u$ , and = 0 if  $Z(\underline{x}) < u$ .

As before, we are interested in  $E[V]$ , the "average reserves" of an oilfield "randomly selected" from the  $N$  oilfields in  $\Omega$  with reserves  $V_1 \dots V_N$ . Once again we shall define this in terms of the "ergodic expectation" so that

$$E[V] = \lim_{\Omega \rightarrow R^2} \frac{\sum V_i}{N} \quad (6.11)$$

If we define  $Y$  to be the total reserves inside  $\Omega$  per unit (gross) area, then

$$\begin{aligned} \lim_{\Omega \rightarrow R^2} \frac{\sum V_i}{N} &= \frac{\lim \sum V_i / A(\Omega)}{\lim N / A(\Omega)} \\ &= \frac{E[Y]}{E[n]} \end{aligned} \quad (6.12)$$

To derive an expression for  $E[Y]$ , assume that  $\Omega$  is a unit square, and define a variable  $W(\underline{x})$  such that:

$$\begin{aligned} W(\underline{x}) &= Z(\underline{x}) \quad \text{if } Z(\underline{x}) \geq u \\ &= 0 \quad \text{if } Z(\underline{x}) < u \end{aligned}$$

$$\begin{aligned} \text{Then} \quad E[Y] &= E \int_{\Omega} W(\underline{x}) d\underline{x} = \int_{\Omega} E[W(\underline{x})] d\underline{x} \\ &= E[W(\underline{x})] \int_{\Omega} d\underline{x} = E[W(\underline{x})] \end{aligned} \quad (6.13)$$

$$\text{Now} \quad E[W(\underline{x})] = \int_0^{\infty} z f_z(z) dz \quad (6.14)$$

So in the Normal case:

$$E[Y] = \frac{\sigma}{\sqrt{2\pi}} e^{-u^2/2\sigma^2} \quad (6.15)$$

and

$$\begin{aligned} E[V] &= E[Y]/E[n] \\ &\approx \frac{2\pi\sigma^4}{u\sigma^2} \quad (6.16) \end{aligned}$$

Thus we have derived approximate formulae for the expected numbers, area and volumes of oilfields in a certain sense. The question arises as to the accuracy of these approximations. Adler (1981, p.136ff) shows that as the level  $u$  increases, the excursion sets tend to become ~~convex~~<sup>X</sup> figures with no holes - thus for large  $u$  equation (6.9) will be a good approximation. To investigate the accuracy of the approximation over a range of values of  $u$  (or of  $\epsilon=u/\sigma$ ), some simulation experiments were carried out, in particular to validate equation (6.10).

It was felt to be relatively simple to simulate a closed contour  $Z(\underline{x})=u$  and compute the area within it, whereas the validation of (6.9) or (6.16) would have required an order of magnitude more work. The technique used was to choose a point in space, specify that  $Z(\underline{x})=u$  at that point, and track the contour from there back to the starting point using a triangular grid of simulated values of the correlated variable  $Z(\underline{x})$ . A number of realisations of closed contours were generated in this way, and the results as regards their areas are shown in Appendix B. The results of these simulations are in reasonably good agreement with equation (6.10), even for quite small values of  $\epsilon$  ( $=u/\sigma$ ).

### 6.3 VARIABLE THRESHOLD

A useful generalisation of the model is to assume that the threshold level  $u$  is not a constant but varies from point to point of the area of interest. This corresponds to the intuitive concept that oilfields tend to be found in clusters, and some areas are more likely than others to hold oil. This is essentially equivalent to assuming that  $Z(\underline{x})$  is in fact composed of two components: a slowly-varying trend plus a residual, while the threshold  $u$  remains a constant.

We shall consider  $u$  to be a stationary random process, with a much longer-range correlation structure than  $Z(\underline{x})$  (see Figure 6.2). Let us define  $\epsilon = u/\sigma$ ,  $\sigma_s^2 = -g''(0)$  and  $R^*(\epsilon) = R(\epsilon\sigma)$ .

We can rewrite the approximate formulae for the expectations of the quantities of interest with respect to the value of  $\epsilon$  which is appropriate to a particular point.

$$E[A] \approx \frac{R^*(\epsilon) (2\pi)^{3/2}}{\epsilon\sigma_s^2 e^{-\epsilon^2/2}} \quad (6.17)$$

$$E[V] \approx \frac{2\pi\sigma}{\epsilon\sigma_s^2} \quad (6.18)$$

$$E[n] \approx \frac{\epsilon\sigma_s^2 e^{-\epsilon^2/2}}{(2\pi)^{3/2}} \quad (6.19)$$

$$E[Y] \approx \frac{e^{-\epsilon^2/2}\sigma}{\sqrt{2\pi}} \quad (6.20)$$

To fit a model of this type to data and produce the above estimates for various points within a given oil province, we need to estimate the following parameters: values of  $\epsilon$  at points of interest, the variance  $\sigma^2$  of  $Z(\underline{x})$  and the variance  $\sigma_s^2$  of the gradient.

#### 6.4 FITTING THE MODEL TO OILFIELD DATA

In order to illustrate a simple application of this model, data for the British sector of the North Sea up to about mid-1974 has been used.

This consists of:

1. Locations of 104 exploration wells and whether or not a commercial oilfield was discovered.
2. Areas and (estimated) recoverable reserves of commercial oilfields.

This data is only approximate as well as rather out-of-date, but it is used to illustrate how the model can be fitted over an oil province to give a reasonable picture without any geological information being included.

The first problem is the estimation of  $\epsilon$ . This was carried out at the location of each oilfield, based on the number of successful and unsuccessful exploration wells drilled in the near neighbourhood.

Let  $n_s$  = number of successful wells,  
 $n_f$  = number of unsuccessful wells,  
 and  $\eta = n_s / (n_s + n_f)$ .

If wells were drilled at random, we should expect  $\eta$  to give us an estimate of  $R^*(\epsilon)$ , and hence an estimated value of  $\epsilon$ . But obviously exploration wells are not drilled wholly at random, but tend to be drilled in the more likely places first. We should take some account of this in the estimation of  $\epsilon$ . How this should be done is very much open to debate - we shall make a very crude assumption for this illustrative example.

Suppose the probability that a well is drilled at some point  $\underline{x}$ , conditional upon the value  $Z(\underline{x})=z$ , is proportional to  $F(z)$ .



$$P[\text{well drilled at } \underline{x} | Z(\underline{x}) = z] = \alpha F(x) ,$$

$$\text{and } P[\text{well drilled at } \underline{x} \text{ and } Z(\underline{x}) = z] = \alpha F(z) f(z) dz .$$

Therefore the total probability is  $\int_{-\infty}^{\infty} \alpha F(z) f(z) dz = 1 .$

$$\text{Let } v = F(z) \text{ and thus } dv = f(z) dz .$$

$$\text{Therefore } \alpha \int_0^1 v dv = 1 \Rightarrow \alpha = 2 .$$

And so the probability that an oilfield will be discovered at the point  $\underline{x}$  is

$$2 \int_{\epsilon^{\sigma}}^{\infty} F(z) f(z) dz = 1 - [\Phi(\epsilon)]^2 \quad (6.21)$$

Let us therefore assume that  $\eta$  is an estimate of  $1 - [\Phi(\epsilon)]^2$  and

$$\hat{R}(\epsilon) = 1 - \sqrt{1 - \eta} . \quad (6.22)$$

This method was used to estimate  $\epsilon$  at the location of each oilfield ("near neighbourhood" being defined as within 3 British North Sea blocks of the oilfield). Table 6.1 shows the data for each oilfield and the estimated values of  $\epsilon$ . These values of  $\epsilon$  were input to the automatic contouring algorithm (with grand mean equal to 3.0 for the boundary regions) and the resulting map is shown in Figure 6.3. This map gives a fairly good indication of the general shape of the northern North Sea basins in the British sector.

The other two parameters,  $\sigma^2$  and  $\sigma_s^2$  were estimated from the North Sea data somewhat approximately. Using equation (6.17) for the mean oilfield area together with the estimated  $\epsilon$  values, it was possible to produce a value of  $\sigma_s^2$  for each field. An average value of  $0.3676 \text{ miles}^{-2}$  was used for later calculations. Using this value together with equation (6.18) for each oilfield gave a set of values of  $\sigma$ . These values were well

scattered, with a few very large ones. The arithmetic mean was  $32.59 \times 10^6$  STB/sq.mile, with a median value of  $18.94 \times 10^6$  STB/sq.mile. The geometric mean of  $20.30 \times 10^6$  STB/sq.mile was chosen as the best compromise for the overall value of  $\sigma$ .

Using these estimated parameter values, it is possible to substitute into equations (6.17) to (6.20) to show the relationships between  $\epsilon$  and the functions of interest. Figure 6.4 shows the expected volumes of discovered oilfields and the expected reserves per (British) North Sea block as functions of  $\epsilon$ . Figure 6.5 shows the expected areas of discovered fields and the expected number discovered per block as functions of  $\epsilon$ . These graphs, in association with the  $\epsilon$  contour map of Figure 6.3, give an impression of the model's predictions about prospects in the British North Sea.

### 6.5 MEAN AND VARIANCE OF OIL RESERVES

From the results of fitting a model of this form to the reserves of an oil province, we wish to be able to gain an impression of the estimated total reserves and the uncertainty in this estimate. The latter we may categorise by the variance in the total reserves, and may be considered to be made up of uncertainty from three sources:

1. For a fixed spatial distribution of  $\epsilon$  values, the variance in the total reserves =  $\text{Var}[R]$  say.
2. Given  $\epsilon$  values at fixed points (oilfields) and correlation parameters for the  $\epsilon$  distribution, the uncertainty due to the fact that the  $\epsilon$  values form a stochastic process.
3. Errors due to uncertainty in the correlation parameters of the  $\epsilon$  process.

The first source of variation may be computed, and the second estimated by simulation, but the third is more difficult to quantify. Let us suppose that our region of interest  $\Omega$  is divided into  $M$  blocks, and in each such block we may assume that the value of  $\epsilon$  is essentially constant (thus we are making a step-function approximation to the true  $\epsilon$  surface). Let  $Y_i$  be the reserves in the  $i^{\text{th}}$  block, with value  $\epsilon_i$  and  $u_i = \epsilon_i \sigma$ . Then the total reserves

$$R = \sum_{i=1}^M Y_i$$

$$\therefore \text{Var}[R] = \sum_{i=1}^M \sum_{j=1}^M \text{Covar}[Y_i, Y_j] \quad (6.23)$$

Consider

$$\text{Covar}[Y_1, Y_2] = A_1 A_2 \int_{u_1}^{\infty} \int_{u_2}^{\infty} z_1 z_2 f(z_1, z_2) dy_1 dz_2 - E[Y_1]E[Y_2] \quad (6.24)$$

where  $A_1, A_2$  are the areas of the two blocks.

The first term above

$$= \frac{A_1 A_2}{\sqrt{2\pi\tau}} \int_{u_1}^{\infty} z_1 G(z_1) dz_1 \quad (6.25)$$

where  $G(z_1) = \frac{1}{\sqrt{2\pi\tau}} \int_{u_2}^{\infty} z_2 \exp \left[ -\frac{(\sigma_2^2 z_1^2 - 2\sigma_{12}^2 z_1 z_2 + \sigma_1^2 z_2^2)}{2\tau^4} \right] dz_2$

and  $\sigma_1^2, \sigma_2^2$  and  $\sigma_{12}^2$  are the variances and covariances of  $Z(x)$  in the two blocks

and  $\tau^4 = \sigma_1^2 \sigma_2^2 - \sigma_{12}^4$ .

Now,

$$\begin{aligned} G(z_1) &= \frac{1}{\sqrt{2\pi\tau}} e^{-z_1^2/2\sigma_1^2} \left[ \int_{u_2}^{\infty} \left( z_2 - \frac{\sigma_{12}^2}{\sigma_1^2} z_1 \right) \exp \left[ -\frac{\sigma_1^2 \left( z_2 - \frac{\sigma_{12}^2}{\sigma_1^2} z_1 \right)^2}{2\tau^4} \right] dz_2 \right. \\ &\quad \left. + \frac{\sigma_{12}^2}{\sigma_1^2} z_1 \int_{u_2}^{\infty} \exp \left[ -\frac{\sigma_1^2 \left( z_2 - \frac{\sigma_{12}^2}{\sigma_1^2} z_1 \right)^2}{2\tau^4} \right] dz_2 \right] \\ &= e^{-z_1^2/2\sigma_1^2} \frac{\tau}{\sigma_1} \left[ \frac{s}{\sqrt{2\pi}} e^{-w^2/2s^2} + \frac{\sigma_{12}^2}{\sigma_1^2} z_1 R^* \left( \frac{w}{s} \right) \right] \quad (6.26) \end{aligned}$$

where  $w = u_2 - \frac{\sigma_{12}^2}{\sigma_1^2} z_1$

and  $s^2 = \frac{\tau^4}{\sigma_1^2}$

These formulae enable us to estimate  $\text{Var}[R]$ , given a set of values of  $\epsilon$  for each block. This calculation was applied to the North Sea data, using the  $\epsilon$  values at the oilfields as fixed and estimating the correlation distance for the  $\epsilon$  process ( $\rho_\epsilon$ ) by maximum likelihood methods as 0.623 units (1 unit = 1 block length = 24.75 miles). Other parameters used were:

For the  $\epsilon$  process:

$$\mu_\epsilon = 3.0 \text{ (fixed so that at the boundaries of the oil province the oilfields vanish)}$$

$$\sigma_\epsilon = 0.551 \text{ (estimated from data).}$$

For the  $Z(x)$  process:

$$\rho = 0.0666 \text{ units (= 1.65 miles)}$$

$$\sigma = 12.433 \times 10^9 \text{ STB/sq.unit}$$

$$(\text{= } 20.30 \times 10^6 \text{ STB/sq. mile})$$

On this basis, assuming that the  $\epsilon$  values are fixed as in Figure 6.3, the mean reserves value was computed as  $38.86 \times 10^9$  STB with a variance of  $721.1 \times 10^{18}$  STB<sup>2</sup>, or a standard deviation of  $26.85 \times 10^9$  STB.

To explore the uncertainty inherent in the definition of the  $\epsilon$  values, random realisations of  $\epsilon$  values were produced, consistent with the values at the oilfields and with the parameters  $\rho_\epsilon$ ,  $\mu_\epsilon$  and  $\sigma_\epsilon$ . The details of these experiments are described in Appendix B. On this basis, the mean reserves (over 10 realisations) was computed as  $48.95 \times 10^9$  STB, with a variance of  $917.6 \times 10^{18}$  STB<sup>2</sup> (standard deviation of  $30.29 \times 10^9$  STB).

The sensitivity of this model to the value of  $\rho_\epsilon$  was explored by varying this parameter. Values of  $\rho_\epsilon$  equal to 0.44 and 0.8 were used, as these gave likelihood values approximately 50% of that for the maximum likelihood estimate of 0.623. The results for these values are also shown in Appendix B.

To put these estimates into perspective, it should be noted that they include the reserves from existing fields (approximately  $13.4 \times 10^9$  STB). The Department of Energy (1976) estimated a possible total of oil reserves from existing licences of 3,190 million tons (equivalent to about  $22.3 \times 10^9$  STB). Odell and Rosing's model produced a total for the whole North Sea of between 79 and  $138 \times 10^9$  STB. However, for various reasons this latter estimate is probably optimistic.

TABLE 6.1

British North Sea Oilfield Data and Estimated Parameters

Oilfield	Block	Longitude	Latitude	Area (sq.mile)	Reserves (10 <sup>6</sup> STB)	n <sub>s</sub>	n <sub>F</sub>	ε
Montrose	22/18	1°24'E	57°23'N	12	500	1	5	1.36
Josephine	30/13	2°32'E	56°34'N	5	300	3	10	1.16
Forties	21/10	0°59'E	57°44'N	40	1800	1	8	1.58
Auk	30/16	2°2'E	56°25'N	16	150	2	9	1.31
Brent	211/29	1°41'E	61°6'N	66	2000	5	2	0.09
Argyll	30/24	2°46'E	56°10'N	5	100	2	10	1.36
Beryl	9/13	1°32'E	59°33'N	10	800	2	5	1.02
Cormorant	211/26	1°6'E	61°8'N	5	400	6	3	0.20
Thistle	211/18	1°32'E	61°22'N	22	800	6	4	0.34
Piper	15/17	0°16'E	58°28'N	16	800	1	14	1.83
Maureen	16/29	1°43'E	58°7'N	4	300	2	2	0.55
Dunlin	211/23	1°36'E	61°16'N	20	1250	6	4	0.34
Alwyn	3/14	1°40'E	60°33'N	24	500	2	5	1.02
Hutton	211/28	1°24'E	61°4'N	12	800	6	3	0.20
Heather	2/5	0°57'E	60°57'N	9	500	2	1	0.20
Ninian	3/8	1°29'E	60°48'N	40	1100	4	4	0.55
Andrew	16/28	1°24'E	58°3'N	5	200	2	4	0.90
Magnus	211/12	1°17'E	61°37'N	10	400	3	5	0.81
Claymore	14/19	0°16'W	58°26'N	16	700	1	8	1.58

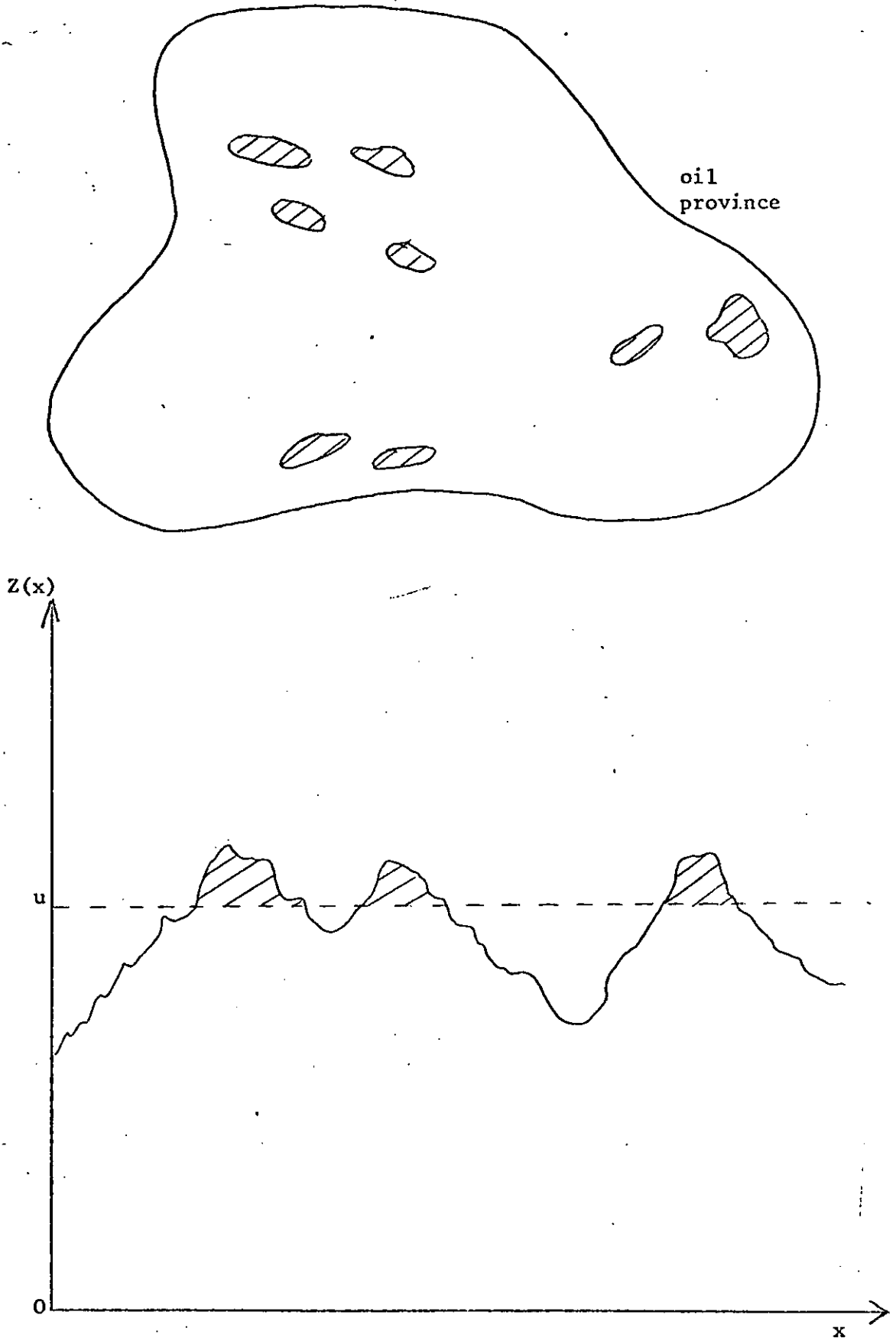


FIGURE 6.1: Model of the occurrence of oilfields

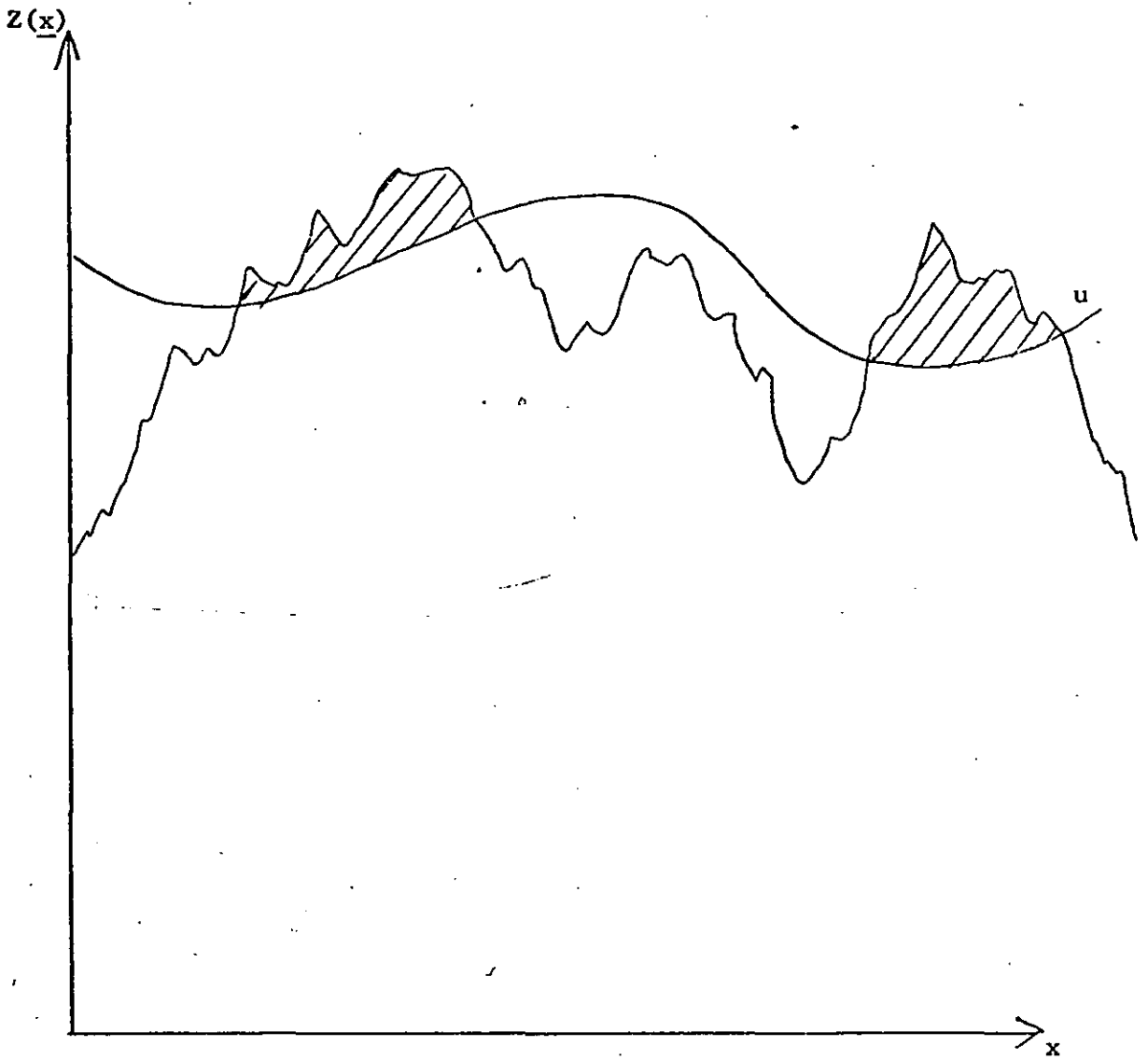


FIGURE 6.2: Oilfield occurrence with variable cut-off level



North Sea epsilon contours.

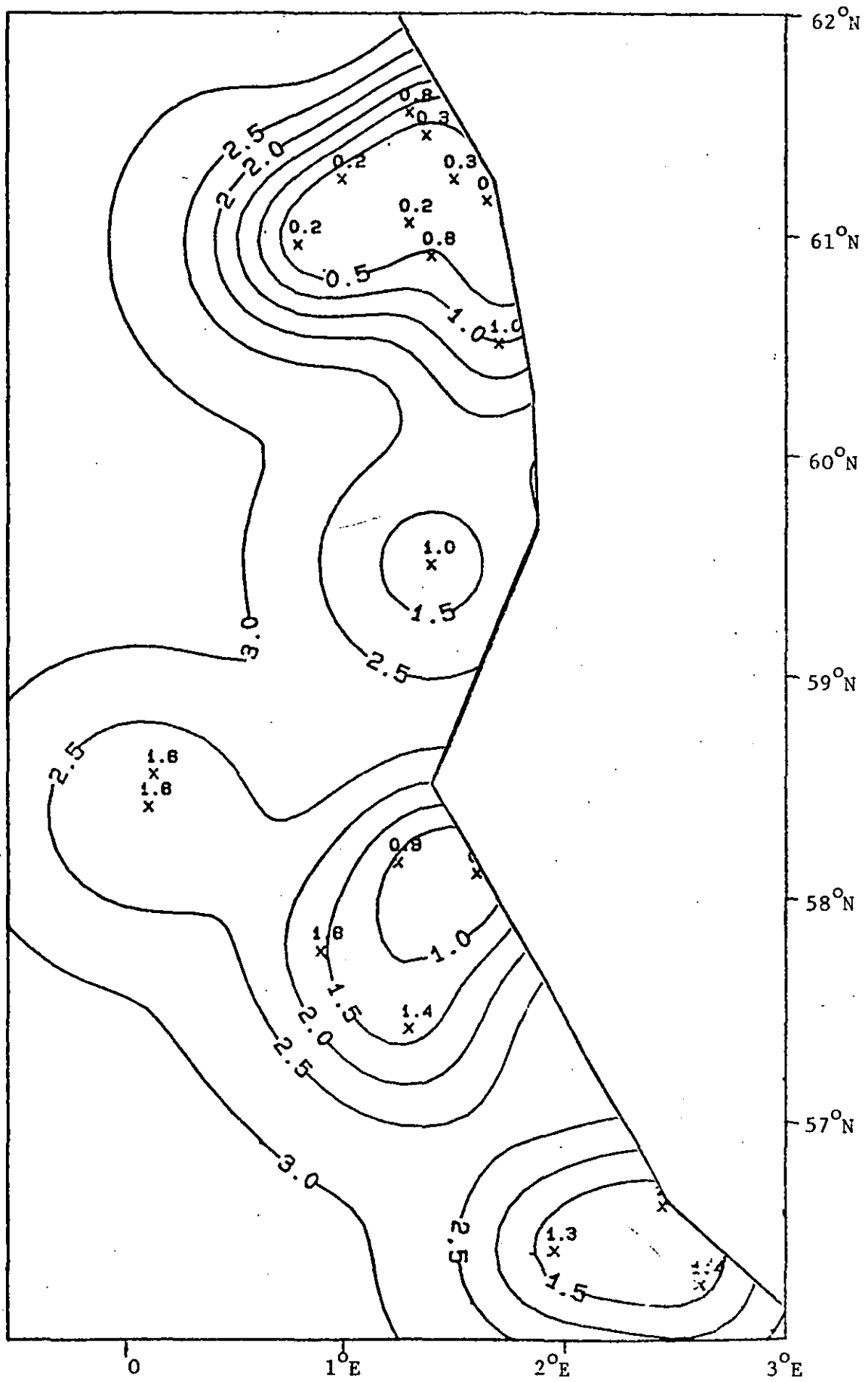


FIGURE 6.3: Contours of  $\epsilon$  and existing oilfield values

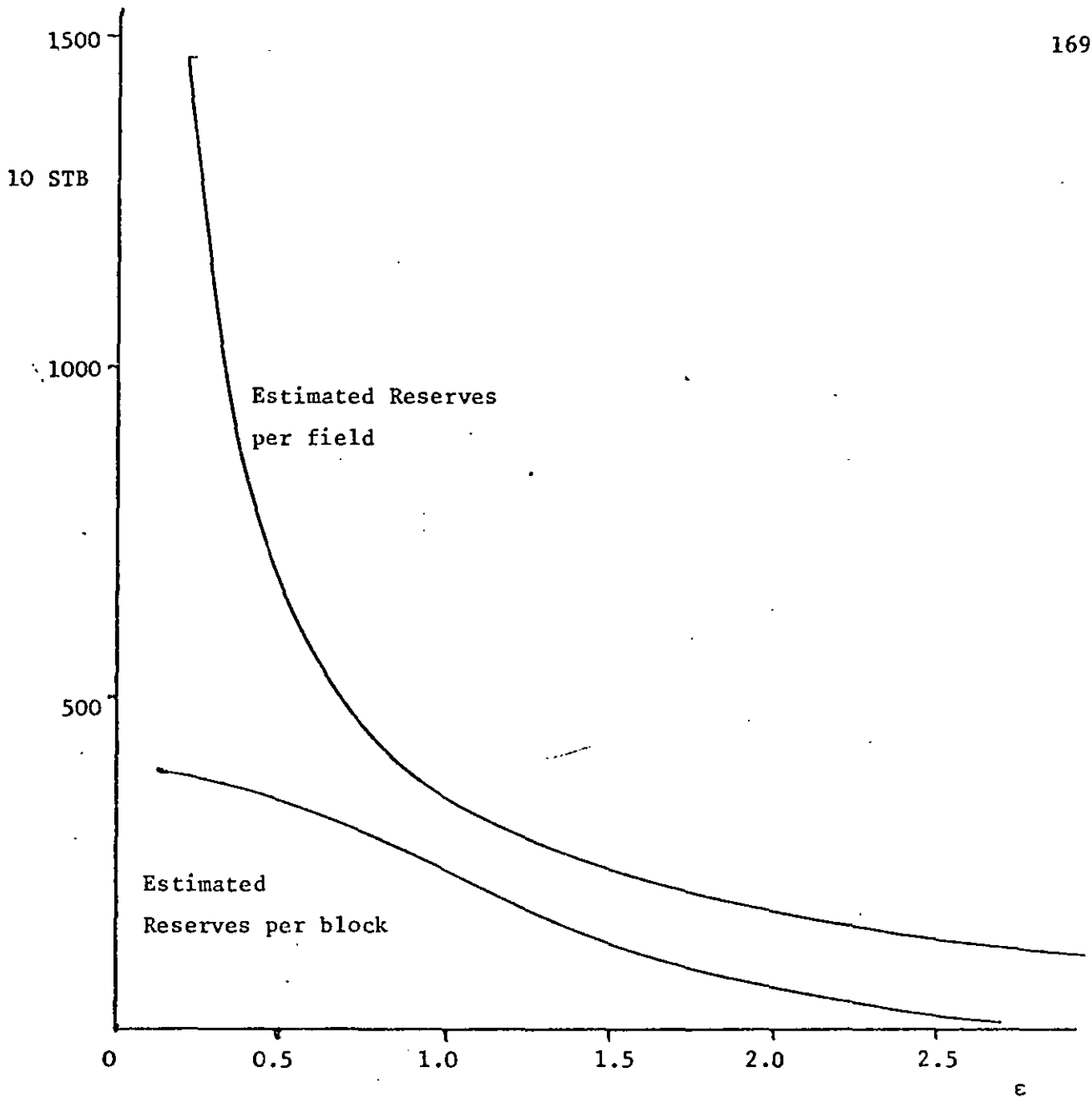
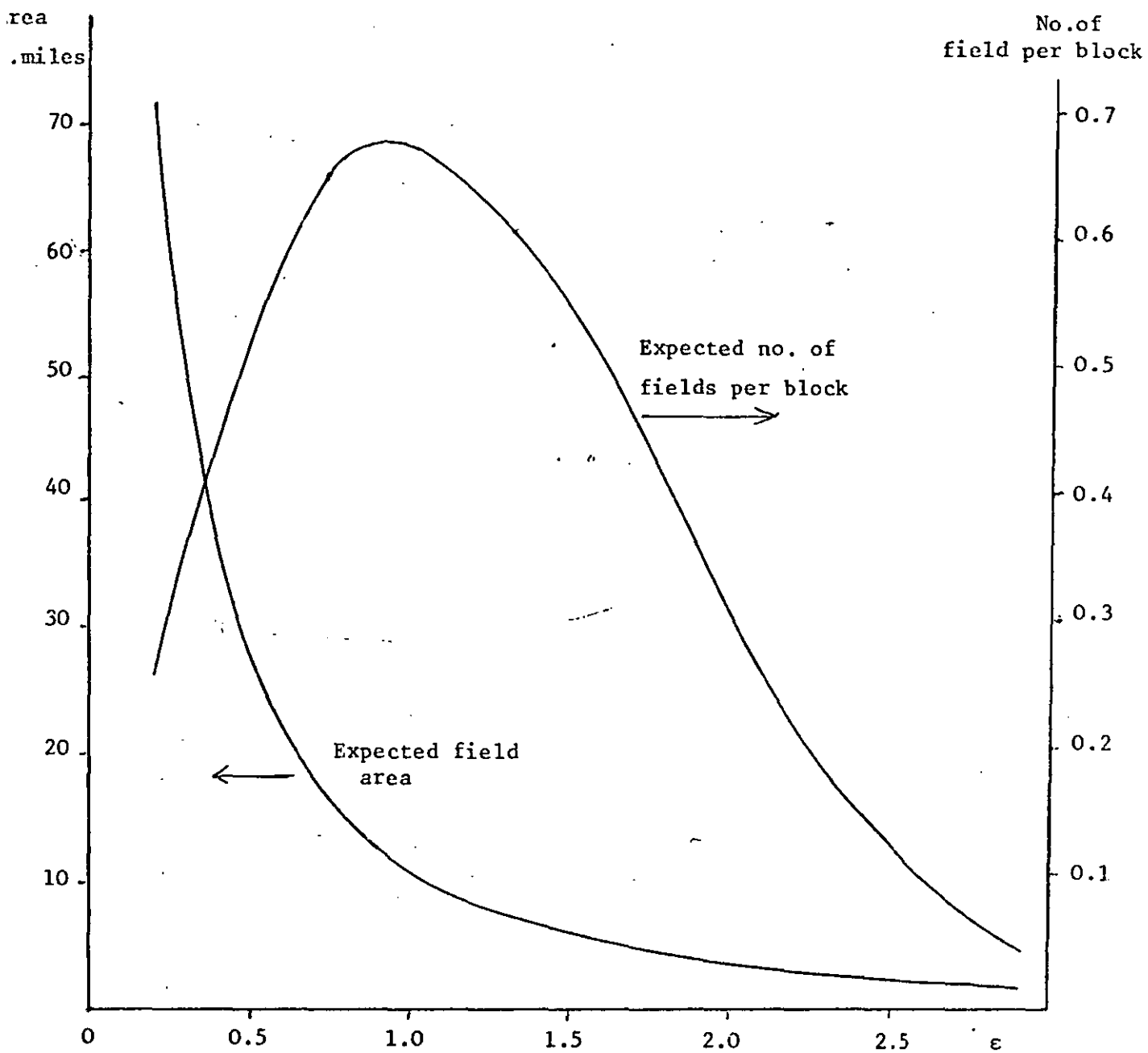


FIGURE 6.4: Estimated reserves as a function of  $\epsilon$



**FIGURE 6.5:** Areas and numbers of fields as functions of  $\epsilon$

CHAPTER 7

CONCLUSIONS

The aim of this work has been to explore possible applications for the theory of correlated random functions of several variables. This theory is well-developed, but there appeared to be interesting fields of study to which it could be applied, but so far had not.

The first such field was contouring, in particular contouring from sparse and arbitrary data points. Current methods of interpolating from the data points to any other point seemed to lack an underlying fundamental model, and could therefore be classified as "ad hoc" techniques. Typically, they also tended to lead to discontinuities in derivatives at various points, and to run into difficulties with data points which are unevenly scattered.

Treating the data points as realisations of a stationary stochastic process gives the desired fundamental conceptual model from which a sensible interpolation technique may proceed. The model has few parameters, and when these have been estimated an interpolating function is produced which is continuous in all derivatives everywhere, and passes exactly through all the data.

One obvious fact about contouring is that there is no "right" contour map for a given set of data - an infinite number of maps will represent the given data exactly. But it is interesting to see just what effect a change in the interpolating algorithm can have on the visual impact of a contour map.

If it is desired to model a long-range trend in the data, as well as local variations, it was felt best to use a stochastic process model for this also. The alternative is to fit some function (e.g. polynomial) to act as the trend function, but this seems arbitrary unless there is a

priori knowledge that the trend is of this form.

The contouring problem having led to a satisfactory model for data in two dimensions, it was felt useful to extend the model to several dimensions, and to explore situations in which it is desirable to extract as much information as possible about the overall form of a function of several variables, with data only available at a finite set of points. The main application to be investigated was that of optimisation, in particular for cases where the number of function evaluations needed to be minimised. Use of the interpolating function led to an algorithm which compared well with conventional techniques, in particular in higher dimensions since derivatives need not be explicitly computed.

This work has led to the development of a piece of software named SIMP (Stochastic Interpolation and Modelling Program) which combines most of these concepts and algorithms into a single package. Data may be presented as either a set of arbitrary point values, or as a function of several variables with points to be generated systematically to explore a specified space. The program will fit a stationary stochastic model to the data and may use it to generate contour maps and cross-sections, or to optimise the given function.

Finally, the concept of a stationary stochastic process has been used to develop a model for the occurrence of oilfields in an oil-bearing region, in terms of the excursions of such a process above a given level. This model seems to have intuitive appeal, requires only a few parameters to be estimated and leads to estimates of useful quantities such as expected reserves and area per oilfield, and expected number of oilfields per unit area. Encouraging results have been obtained from fitting this model to some data for the British North Sea.

## REFERENCES

- ADLER, R.J. (1976): *"On generalising the notion of upcrossings to random fields"*, Adv.Appl.Prob., Vol.8, pp.789-805.
- ADLER, R.J. & HASOFER, A.M., (1976): *"Level crossings for random fields"*, Annals of Probability (1976), Vol.4, No.1, pp.1-12.
- ADLER, R.J. (1981): *"The geometry of random fields"*, Wiley, Chichester.
- AKIMA, H. (1975): *"Comments on 'Optimal contour mapping using universal kriging' by Ricardo A. Olea"*, J.Geophys.Res., Vol.80(5), pp.832-834.
- BARTLETT, M.S. (1966): *"An introduction to stochastic processes"*, Cambridge.
- BEALE, E.M.L., (1970): *"Computational methods for least squares"*, Chapter 9 of *"Integer and Nonlinear Programming"* ed. J. Abadie, North Holland.
- BELYAEV, Y.K., (1966): *"On the number of intersections of a level by a Gaussian stochastic process, I"*, Theory of Probability and Its Applics., Vol. XI, No.1, pp.106-113.
- de BIASE, L. & FRONTINI, F., (1978): *"A stochastic method for global optimisation"*, Compstat 1978, Physica-Verlag, pp.355-361.
- BRODLIE, K.W. (1980): *"A review of methods for curve and function drawing"*, 'Mathematical Methods in Computer Graphics Design'. Academic Press, pp.1-37.
- CALCOMP Inc. (1971): *"GPCP User's Manual"*.
- CHAMBERS, J.M. (1977): *"Computational Methods for Data Analysis"*, Wiley, New York.



- COTTAFAVA, G. & le MOLI (1969): "*Automatic contour map*", Communications of the ACM, Vol.12, No.7, July 1969.
- CRAMER, H. & LEADBETTER, M.R. (1967): "*Stationary and related stochastic processes*", Wiley, New York.
- DAVID, M. (1977): "*Geostatistical Ore Reserve Estimation*", Elsevier, Amsterdam.
- DEPARTMENT OF ENERGY (1976): "*Development of the oil and gas resources of the United Kingdom*", H.M.S.O., London.
- DIXON, L.C.W., GOMULKA, J. & HERSOM, S.E. (1976): "*Reflections on the global optimisation problem*", 'Optimisation in Action', Academic Press, pp.398-433.
- DIXON, L.C.W. & SZEGÖ, G.P. (1978): "*Towards global optimization 2*", North Holland, Amsterdam.
- FLETCHER, R. (1980): "*Practical methods of optimization, Vol.1*", Wiley, Chichester.
- HAWKINS, D.M. & CRESSIE, N. (1981): "*Robust kriging - a proposal*", Technical Report, National Research Institute for Mathematical Sciences, Pretoria.
- HUIJBREGTS, C. & MATHERON, G. (1971): "*Universal kriging*", Decision making in the mineral industry, Special Vol.12, pp.159-169.
- JOURNEL, A.G. & HUIJBREGTS, C.J. (1978): "*Mining geostatistics*", Academic Press, London.

- JUCHA, S. & NITKIEWICA, K. (1975): "*Probabilistic methods of estimating prospective hydrocarbon reserves*", NAFTA, August, 1975, No.8, pp.313-318.
- MATERN, B. (1960): "*Spatial variation*", Meddelanden fran Statens Skogsforskningsinstitut 495.
- MATHERON, G. (1963): "*Principles of Geostatistics*", Econ.Geol. Vol.58, pp.1246-1266.
- MATHERON, G. (1971): "*The theory of regionalised variables and its applications*", Ecole Nationale Superieure des Mines de Paris.
- McLAIN, D.H. (1974): "*Drawing contours from arbitrary data points*", The Computer Journal, Vol.17, pp.318-324.
- McLAIN, D.H. (1976): "*Two dimensional interpolation from random data*", The Computer Journal, Vol.19, pp.178-181.
- MEISNER, J. & DEMIRMEN, F. (1981): "*The creaming method: a Bayesian procedure to forecast future oil and gas discoveries in mature exploration provinces*", J.R. Statist.Soc. A, Vol.144, Part 1, pp.1-31.
- MILES, R.E. (1974): "*A synopsis of Poisson Flats in Euclidean Spaces*", in "*Stochastic Geometry*", ed. E.F. Harding & D.G. Kendall, pp.202-227, Wiley, London.
- NELDER, J.A. & MEAD, R. (1964): "*A simplex method for function minimization*" Computer Journal Vol.7, pp.308-312.

- NEWTON, R. (1973): "*A statistical prediction technique for deriving contour maps from geophysical data*", *Mathematical Geology*, Vol.5, No.2, pp.179-189.
- ODELL, P.R. & ROSING, K.E. (1974): "*The North Sea oil province - a simulation model of development*", *Energy Policy*, December 1974, pp.316-329.
- OLEA, R.A. (1974): "*Optimal contour mapping using universal kriging*", *J.Geophys.Res.*, Vol. 79(5), pp.695-702.
- PRICE, W.L. (1977): "*A controlled random search procedure for global optimisation*", *The Computer Journal*, Vol.20, No.4, pp.367-370.
- RIPLEY, B.D. (1981): "*Spatial statistics*", Wiley, New York.
- SABIN, M.A. (1980): "*Contouring - a review of methods for scattered data*", 'Mathematical Methods in Computer Graphics and Design', Academic Press, pp.63-86.
- SCHAGEN, I.P. (1979): "*Interpolation in two dimensions - a new technique*", *J.Inst.Math.Applics.*, Vol.23, pp.53-59.
- SCHAGEN, I.P. (1980): "*The use of stochastic processes in interpolation and approximation*", *Intern. J. Computer Math.*, Section B, Vol.8, pp.63-76.
- SCHAGEN, I.P. (1980): "*Stochastic interpolating functions - applications in optimisation*", *J.Inst.Maths.Applics.*, Vol.26, pp.93-101.
- SCHAGEN, I.P. (1980): "*Stochastic interpolation applied to the optimisation of expensive objective functions*", *Compstat 1980*, Physica-Verlag pp.362-367.

- SCHAGEN, I.P. (1980): "A stochastic model for the occurrence of oilfields and its application to some North Sea data", *Appl. Statistics* (1980), Vol.28, No.3, pp.282-291.
- SHVIDLER, M.L. (1964): "Filtration flows in heterogeneous media", Consultants Bureau, New York (trans. from Russian).
- SOLIS, F.J. & WETS, R.J.-B. (1981): "Minimization by random search techniques", *Mathematics of Operations Research*, Vol.6, No.1, pp.19-30.
- SUTCLIFFE, D.C. (1976): "An algorithm for drawing the curve  $f(x,y)=0$ ", *The Computer Journal*, Vol.19, No.3, pp.246-249.
- SUTCLIFFE, D.C. (1980): "Contouring over rectangular and skewed rectangular grids", 'Mathematical Methods in Computer Graphics and Design', Academic Press, pp.39-62.
- SWITZER, P. (1965): "A random set process in the plane with a Markovian property", *Ann.Math.Stat.*, Vol.36.
- WATSON, G.S. (1971): "Trend surface analysis", *J.Inst.Assoc. for Math. Geol.* Vol.3, No.3, pp.215-226.
- WATSON, G.S. (1972): "Trend surface analysis and spatial correlation", *Geological Soc. of America Special paper 146*, pp.39-46.
- WHITTLE, P. (1954): "On stationary processes in the plane", *Biometrika* 41, pp.431-49.
- WHITTLE, P. (1963): "Prediction and regulation", English Universities Press, London.

WIENER, N. (1939): "*The Ergodic Theorem*", Duke Math. Journal, Vol.5,  
pp.1-18.

YAGLOM, A.M. (1964): "*An introduction to the theory of stationary random functions*", Prentice-Hall, 1964.

ZOUTENDIJK, G. (1976): "*Mathematical Programming Methods*", North Holland.

APPENDIX A

SIMP - A STOCHASTIC INTERPOLATION

AND MODELLING PROGRAM

## INTRODUCTION

The program SIMP has been written in ALGOL-68R and is running on the Loughborough University of Technology 1904S computer system. Its purpose is to fit a stochastic interpolating function to multi-dimensional data and apply it in various ways. The data to be fitted can be defined in one of two forms:

1. As a user-defined function, with  $N$  data points spread evenly throughout the region of interest, and the function evaluated at these points.
2. As  $N$  input data values, at arbitrary input co-ordinates.

The program can produce one or all of the following results:

1. Contour maps of the interpolating function in any two dimensions.
2. Cross-sections of the interpolating function along any line in the region of interest.
3. Optimisation of the user-defined function (providing one is supplied) by iterative optimisation of the interpolating function.

The whole system actually operates in two stages - the main program (SIMP) which fits the model to the data and carries out all the computation for the results to be obtained. If contour maps or cross-sections are required, the requisite data is generated by SIMP and output to a file. A secondary program (SIMPLOT) can be activated to produce the actual plots from this file. All the data relevant to the plotting is input to this program (e.g. scaling factors, number sizes, colours etc.).

The program SIMPLOT is also written in ALGOL-68R, but interfaces with the FORTRAN graphics library GINO-F via a second intermediate file and a FORTRAN program which interprets coded instructions on the file as calls to GINO-F subroutines. This is part of a system developed in the Computer Studies Department for plotting from ALGOL-68R programs. Each GINO-F subroutine call required is activated by an identically-named procedure call in SIMPLOT.

The program SIMP also possesses the ability to dump the stochastic model and the data on to a "model file" and to restart a later run from this file. This is designed to save computing time when several sets of results are required from the same set of data, as the model fitting phase need only be carried out once. Figure A1 illustrates the overall system.



## SIMP - An outline of the structure and the data input

The main program SIMP operates in five phases, as follows:

1. Data generation. (Procedure SETUPDATA).
2. Model fitting. (Procedure FITMODEL).
3. Optimisation. (Procedure OPTIMISE).
4. Contouring. (Procedure CONTOUR).
5. Cross-sectioning. (Procedure SECTION).

The operation of these five phases is controlled by commands read into the program as data. These consist of eight-character strings punched in the first eight columns of data cards. The first such card controls the "mode" of the program. The three possible commands are:

- NORMAL - Model neither dumped to file or restarted from file.
- DUMP - Model dumped to model file after phase 2 (model fitting).
- GETMODEL - Model and data acquired from model file and phases 1 and 2 omitted.

Thus in "NORMAL" mode or "DUMP" mode phases 1 and 2 are carried out (in that order) before any other phase is invoked, while in "GETMODEL" mode they are omitted.

Each phase of the program will now be described, in terms of the overall action performed and the data input required.

### Phase 1 - Data Generation

This phase of the program defines the dimensionality of the problem, the limits of the region of interest, and the values at the initial set of

data points. These latter are either generated from a user-defined function or input as data. The five commands to control this phase and their associated data values are described below:

1. NDIMS - This command specifies the number of dimensions in which the data is defined. The value is read in as an integer following the command, and stored in the variable NDIMS.
2. LIMITS - This command specifies the lower and upper limits of the region of interest in each dimension. Following the command, the values are read in for each dimension from 1 to NDIMS, lower limit followed by upper limit each time.
3. GENERATE - This command specifies that the initial data values are to be generated from a user-defined function, and spread evenly throughout the region of interest. Following the command, the integer value NPOINTS is read in, which is the number of initial data points.
4. DATA - This command is an alternative to the "GENERATE" command. It specifies that the initial data values will be read in as input. Following the command the integer NPOINTS is read, and after that NPOINTS sets of data values. Each set I consists of NDIMS values of the co-ordinates of the point XPT[I], plus the value of the function of interest Z[I].

5. FITMODEL - This command specifies that Phase 1 of the program has ended and Phase 2 should be entered.

Notes: If the command "NDIMS" is omitted, a default value of NDIMS=2 will be used. One and only one of the commands "GENERATE" and "DATA" should be used.

### Phase 2 - Model Fitting

This phase of the program fits a two-stage stochastic model to the data points input in the first phase. Data may be input to this phase to control the fitting of model parameters, but some or all of this data may be omitted, in which case default values will be used. The following commands are understood by this phase of the program:

- AVERAGE - This command is followed by the number of average points (NOAV) to be used in the two-stage model (default value of NOAV=0).
- MUFACT - This command is followed by a value for the real variable MUFACT, which governs the estimation of the grand mean. (Default value = 0). If  $\bar{Z}$  is the arithmetic mean of the given data values, and  $\tilde{Z}$  is the median, then the grand mean  $\mu$  is estimated as
- $$\mu = \text{MUFACT} * \bar{Z} + (1 - \text{MUFACT}) * \tilde{Z} .$$
- ROSMIN - This command is followed by the minimum value for the short-range correlation distance. (Default = 0).
- ROSMAX - Maximum value of the short-range correlation distance. (Default =  $10^{12}$ ).

- MEAN - Input grand mean value. If this command is not read, then the grand mean will be estimated using MUFACT.
- ROLMIN - Minimum value of the long-range correlation distance. (Default = 0).
- ROLMAX - Maximum value of the long-range correlation distance. (Default =  $10^{12}$ ).
- ANISOMIN - Minimum anisotropy factor value. (Default = 0).
- ANISOMAX - Maximum anisotropy factor value. (Default =  $10^{12}$ ).
- ANISO - This command is followed by NDIMS values of input anisotropy factors. If this command is not read, then the anisotropy factors will be estimated.
- ENDMODEL - This command terminates the data for phase 2. The stochastic model is fitted to the data using the parameters input (or default values).

Note: All commands (except ENDMODEL) are optional and may occur in any order.

The model fitting which is carried out after the input of these parameters consists of the following steps:

- a) Compute grand mean and anisotropy factors (if not input as data).
- b) Distribute NOAV average points evenly through the region of interest and calculate a weighted average value at each such point.
- c) Estimate the long-range correlation distance from the average points.

- d) Compute the trend value (based on the average points) for each initial data point and hence the residual error at each point.
- e) Estimate the short-range correlation distance based on the residual error values.

The model to fit the data is then totally defined by these estimated parameters. If the program is in "DUMP" mode, all the details of the data and the fitted model will be output to a file.

### Phase 3 - Optimisation

This phase of the program is initiated by reading the command "OPTIMISE" in the first 8 columns of a card. Following this the program reads the value of the variable EPSILON, which is the termination tolerance for the optimisation algorithm. No other data input is required for this phase.

The following steps are carried out:

- a) Optimise interpolating function fitted to current set of data points, using a "variable metric" algorithm.
- b) Evaluate user-defined function at optimum point so found.
- c) If the user-defined function and the interpolating function values are within the tolerance EPSILON of one another, then the algorithm terminates. Otherwise the model is re-fitted, using the new point just evaluated in the current set, and the algorithm is repeated from step a).

In addition to optimisation, this phase of the program also computes the integral of the interpolating function over the region of interest.

#### Phase 4 - Contouring

This phase of the program is initiated by reading the command "CONTOUR". Further commands and data input for this phase are as follows:

- LEVELS - This command is followed by the integer NLEVELS (number of contour levels required), and then NLEVELS values of contour levels.
- REGULAR - This is designed for the input of contour levels which are evenly spaced. The data to follow consists of the integer NLEVELS (number of contour levels required) and two real values SLEVEL (first contour level) and DLEVEL (increment between successive contour levels). This command is an alternative to "LEVELS".
- STEPMAX - This command is followed by the value of the maximum step length for producing contour lines. (Default = 0.1).
- TITLE - On the card following the command, a title of up to 80 characters will be read.
- PLANE - This specifies the plane in which contouring is to be carried out. Two integers IX and IY are read (dimensions corresponding to X and Y on the 2-d plot). Following this, for every dimension not equal to IX or IY, a fixed value is read in (Default IX=1, IY=2).

- PROJDIST - This command is followed by a real value which determines the maximum projected distance which a data point may be from the contouring plane and still be marked on the plot. (Default =  $10^{12}$ ).
- TRENDMAP - This command initiates the generation of contour lines of the long-range trend. Average points will be marked on the plot (if within PROJDIST of the contouring plane).
- FULLMAP - This command initiates the generation of contour lines of the full interpolating function. Data points will be marked on the plot (if within PROJDIST of the contouring plane).
- ENDMAP - End of Phase 4.

Note: These commands may occur in any order, repeated any number of times, although it is obviously sensible to ensure that the relevant data has been input before using "TRENDMAP" or "FULLMAP".

#### Phase 5 - Cross-sectioning

This phase of the program is initiated by reading the command "SECTION". Following this command the following data is expected:

1. NSTEPS - An integer corresponding to the number of steps, or points along the cross-section at which the interpolation function will be evaluated.
2. Co-ordinates  $X1[K]$ ,  $K=1, NDIMS$  and  $X2[K]$ ,  $K=1, NDIMS$  of the two points between which the cross-section is to be produced.

This program outputs to the plotting file not only the values at the steps along the cross-section line, but also the values at any data points which lie on the line, so that these may (if desired) be included on the plot.

#### Termination

Phases 3,4 and 5 may be re-run in any order as often as required. The command "FINISH" terminates the program when all the required computation has been done.



EXAMPLES OF INPUT DATA

Some examples of input data are included here as a guide to using the program in different cases.

1. User-defined function with optimisation and contouring

A 4-dimensional Shekel function is defined by the user, given by the procedure with declaration:

```
PROC obj = (REF REAL x, INT ndims) REAL :
```

The following data cards are input.

<u>Command</u> (Cols. 1-8)	<u>Rest of Card</u>	<u>Comments</u>
NORMAL		"Normal" mode
NDIMS	4	} Phase 1 Data input
LIMITS	0 1 0 1 0 1 0 1	
GENERATE	20	
FITMODEL		} Phase 2 Model fitting
AVERAGE	10	
ENDMODEL		} Phase 3 Optimisation
OPTIMISE	0.001	
CONTOUR		} Phase 4 Contouring (2 maps)
PLANE	1 2 0.5 0.1	
STEPMAX	0.025	
REGULAR	12 7 1	
PROJDIST	0.25	
FULLMAP		
PLANE	3 4 0.7 0.7	
FULLMAP		
ENDMAP		
FINISH		



### 3. Input data values with contouring and sectioning

In a previous run a model has been defined using 72 input data points, and this has been stored on a file using the "DUMP" command. Data cards for a later run are as follows:

<u>Command</u> (Cols. 1-8)	<u>Rest of Card</u>				<u>Comments</u>
GETMODEL					"GETMODEL" mode
CONTOUR					} Phase 4 Contouring
STEPMAX	0.05				
REGULAR	6	100	100		
FULLMAP					
ENDMAP					
SECTION	100				} Phase 5 Cross-section
	0	0	12	8	
SECTION	100				} Phase 5 Cross-section (repeated)
	0	8	12	0	
FINISH					

## A BRIEF DESCRIPTION OF THE MAIN VARIABLES AND PROCEDURES IN SIMP

Before describing the procedures which make up the program, it is worth also describing some of the more important variables, especially those that make up the "model" which is fitted to the given data.

### Integers

- ndims - Number of dimensions in problem.
- npoints - Number of given data points (N).
- noav - Number of average points for estimation of trend ( $n_A$ ).

### Reals

- mu - Grand mean ( $\mu$ ).
- ro - Short-range correlation distance ( $\rho_S$ ).
- rolong - Long-range correlation distance ( $\rho_L$ ).
- avsep - Average separation between data points.

### Real Arrays

- [1:npoints, 1:ndims] REAL xpt - Co-ordinates of data points.
- [1:noav, 1:ndims] REAL xav - Co-ordinates of average points.
- [1:npoints] REAL z - Function values at data points.
- [1:noav] REAL zav - Trend values at average points.
- [1:npoints] REAL ze - Residual values at data points.
- [1:npoints] REAL gamma - Values of  $\underline{\gamma}$  at data points.
- [1:npoints] REAL delta - Values of  $\underline{\gamma}$  for trend at average points.
- [1:ndims] REAL aniso - Anisotropy factors for each dimension.
- [1:ndims] REAL xlower - Lower boundaries of region of interest in each dimension.
- [1:ndims] REAL xupper - Upper boundaries of region of interest in each dimension.

An outline flowchart of the main routine of SIMP is included as Figure A2. Seven procedures are called from the main routine:

1. getmodel - procedure to input model values from file.
2. dump - procedure to dump model values to file after model has been fitted to data.
3. setupdata - procedure to set up number of dimensions, boundaries of region of interest, and values of given function at data points (Phase 1).
4. fitmodel - procedure to fit model to data values (Phase 2).
5. optimise - optimisation procedure (Phase 3).
6. contour - contouring procedure (Phase 4).
7. section - cross-sectioning procedure (Phase 5).

A brief description of every other procedure in the program follows, in the order in which they occur in the program.

Procedures marked with an asterisk have been listed at the end of this appendix - these consist of those procedures of particular novelty or relevance to the techniques used in the program.

1. min - Returns minimum value of a list of reals.
2. max - Returns maximum value of a list of reals.
3. outarray - Writes array of reals out to file.
4. inarray - Reads in array of reals from file.
5. obj - User-defined objective function.
6. distance - Calculates distance between two points, including anisotropy factors.
7. pivot - Pivots on diagonal element of symmetric matrix.
8. set up - Sets up correlation matrix for a set of points. (Calls distance).

- \*9. cutup ✓ - Ad hoc method to define initial locations of points in region of interest, for large number of points. (Calls cutup).
- \*10. explore ✓ - Ad hoc method to define initial locations of points in region of interest, for small number of points. (Calls distance).
- \*11. hcalc ✓ - Computes value and derivatives of repulsive function for a given configuration of points inside the region of interest. (Calls setup).
- \*12. updateh ✓ - Updates value of repulsive function when the position of a point has been changed.
- \*13. spreadem ✓ - Takes initial configuration of points in region of interest and varies it to reduce the value of the repulsive function. (Calls hcalc, updateh).
- \*14. initialpts ✓ - Defines positions of initial data points in region of interest. (Calls cutup, explore, spreadem).
- 15. statistics - Estimates mean, median and standard deviation of a list of reals and prints them out.
- 16. cubic - Finds a root of a cubic equation between 0 and 1 by Newton's method.
- 17. covarest - Estimates correlation between the function values at two points. (Calls cubic).
- 18. regress - Multiple regression procedure. (Calls pivot).

- \*19. anisocalc - Estimates anisotropy factors from initial set of data points. (Calls distance, covarest, regress).
- \*20. averagepts✓ - Defines positions of average points in region of interest and calculates weighted average values for them. (Calls initialpts, distance).
- \*21. errorsum ✓ - Computes error sum of squares for a given value of  $\rho$  and set of data points. (Calls setup, pivot).
- \*22. minest ✓ - Estimates minimum value of a function of one variable given 3 points.
- \*23. estimated ro ✓ - Estimates  $\rho$  by minimising error sum of squares. (Calls errorsum, minest).
- \*24. Quick ro - Estimates  $\rho$  based on correlations between adjacent pairs of data points. (Calls covarest).
- \*25. point value - Estimates function value at unknown point given a set of data points and values of  $\mu$  and  $\rho$ . (Calls distance).
- \*26. interpolator - Estimates total function value using two-stage model, as a sum of trend and residual terms. (Calls point value).
- \*27. prepare - Computes  $\underline{y}$  values for model, given data values and  $\mu$  and  $\rho$ . (Calls setup and pivot).
- 28. normal - Approximates the Standard Normal Integral.

- 29. `integral` - Estimates the integral of the interpolating function over the region of interest. (Calls `normal`).
- 30. `onedimmax` - Searches for maximum of function along a given line. (Calls `interpolator`).
- 31. `changedir` - Changes direction matrix for variable metric optimisation algorithm.
- 32. `variable, metric` - Optimises interpolating function starting from current best point using variable metric algorithm. (Calls `interpolator`, `onedimmax`, `distance`, `changedir`).
- 33. `gammachange` - Updates model  $\gamma$  values to allow for new data point. (Calls `distance`, `prepare`).
- \*34. `find` - Searches along a given straight line to find point at which interpolating function equals contour level. (Calls `interpolator`).
- \*35. `divide` - Searches between two data points to find reference point. (Calls `interpolator`, `find`).
- \*36. `select` - Selects reference points for a given contour level and set of data points. (Calls `divide`).
- \*37. `outpoint` - Outputs point on contour to plot file, and checks to see if any reference point should be deleted.
- \*38. `anglediff` - Computes the difference between two angles.



- \*39. smooth - Checks if contour is reasonably smooth at current point. (Calls anglediff).
- \*40. stepalong - Generates new point on contour from existing point. (Calls interpolator, find, smooth).
- \*41. outofarea - Checks if contour segment has left the area of interest.
- \*42. backtostart - Checks if contour segment has returned to its starting point. (Calls anglediff).
- \*43. countourtrace - Traces the contour segments for a given contour level. (Calls select, outpoint, stepalong, outofarea, backtostart).
- 44. contourstart - Initialises contour map and defines data points to be plotted. (Calls distance, interpolator).
- 45. contourmap - Produces contour lines for all levels for a given model and set of data. (Calls contourstart, statistics, countourtrace).

SIMPLOT - AN OUTLINE OF THE STRUCTURE AND THE DATA INPUT

SIMPLOT has two sources of input data. One is the plot file created by running SIMP, which contains contour maps and/or cross-sectional data generated from the model fitted to the given data points. The other is from the standard input file and contains control commands and data to supervise the actual creation of the relevant plots.

Some of the commands govern the input of parameters for the plotting (which may have default values) and others instruct the program to create a plot from the data on the plot file. The commands are read in the first 8 columns of a data card, and a description of them and their associated data values follows:

- TITLESIZ           - Followed by values of SIZETITLE, DXTITLE and  
DYTITLE. SIZETITLE is the size of the characters  
for plotting the title in mms. (Default is 3mms).  
DXTITLE and DYTITLE are the co-ordinates (in mms)  
of the start of the title relative to the top left-  
hand corner of the map or section plot. (Default  
values are 0 and 2mms).
- SCALE               - Followed by values of SCALE and SPACING. SCALE is  
the scaling factor in both X and Y directions, in  
mms/data unit (default is 10), generally used for  
contour maps. SPACING is the distance in mms on  
the contour lines between adjacent markings of the  
contour level (default is 300).
- LEVELSIZ           - Followed by values of LEVELSIZE and CONTOURDECS.

LEVELSIZE is the size in mms of the contour level markings (for contour maps) or of the axis scale markings (for sections). (Default is 2).

CONTOURDECS is the number of decimal places to which these are plotted (default is 0).

#### COLOUR

- Followed by values of CONTOURPEN, CPSTART and CPINC. CONTOURPEN is an integer in the range 1 to 4 which controls the pen colour for a subset of the contours (1 is black, 2 is red, 3 is blue and 4 is green). Contour levels, starting at number CPSTART and incrementing by CPINC, will be plotted in this colour. (For example, if the data input were: 2 3 5, contour levels numbered 3, 8, 13 etc. on the list would be coloured red). (Default values are 1 1 1).

#### POINTS

- Followed by values of PTPEN, PTSYMB, and PTSIZE. PTPEN is an integer from 1 to 4 indicating pen colour for marking the positions on the plot of the data points. PTSYMB is an integer indicating the type of symbol used to mark the data points (see GINO-F documentation for routine SYMBOL for details). If PTSYMB<0 then the data points are not plotted. PTSIZE is the size in mms of these symbols. (Default values are 1 8 2).

#### PTNO

- Followed by values of PTNOSIZE, PTNODX, and PTNODY. PTNOSIZE is the size in mms of the characters used

to mark the data point value against the plotted symbol (default is 2). If  $PTVALSIZE \leq 0$  this will not be plotted.  $PTVALDECS$  is the number of decimal places to which it will be plotted.

$PTVALDX$  AND  $PTVALDY$  are the distances in mms of the start of this number from the plotted symbol, in the X and Y directions. (Default values are 2 0 -2 2). Figure A3 illustrates all these various parameters with reference to plotting a typical data point.

- TITLE** - On the following card, up to 80 characters of title may be input.
- DRAWMAP** - This command instructs the program to plot the next contour map on the plot file.
- PLOTSIZE** - Followed by values of  $XMAX$  and  $YMAX$ , which are the maximum overall plot sizes in mms in the X and Y directions. This command should be given, once only, before any plotting is carried out.
- FACTOR** - Followed by the value of  $FACTOR$ , a scaling factor by which the function values on the file are multiplied for the plot. Contour levels are multiplied by the same factor (default is 1).
- DRAWSECT** - This command instructs the program to plot the next cross-section on the plot file.

- SCALEXY - Followed by values of SCALEX and SCALEY, distance scaling factors in mms/data unit in the X and Y directions, generally used for cross-sections. (Default is 10 10).
- AXES - Followed by values for the axes of a cross-section plot: XL, XU, DX, XSTART, YL, YU and DY. XL are the lower, upper and incremental values to be marked on the X axis, and YL, YU and DY are similar values for the Y axis. XSTART is the X-value of the start of the cross-section.
- FINISH - Terminates the run.

Notes: The above commands are all optional, with the exception of "FINISH". The commands "DRAWMAP" and "DRAWSECT" should be mixed in a way which reflects the structure of the plot file. (For example, if the input to SIMP contained the commands "CONTOUR", "SECTION" and "CONTOUR" in that order, then the input to SIMPLOT should contain "DRAWMAP", "DRAWSECT" and "DRAWMAP" in the same order. Each command (except "FINISH" and "PLOTSIZE") may be repeated as often as required.

EXAMPLES OF INPUT DATA TO SIMPLOT

These examples correspond to the three examples given for SIMP, and illustrate how the results of these runs could be plotted.

Example 1

<u>Command</u>	<u>Rest of Card</u>	<u>Comments</u>
TITLE		
4-D SHEKEL (X1,X2)		(Title starts in column 1)
PLOTSIZE	500 200	
LEVELSIZ	2 -1	
POINTS	2 3 2	Data points in red, with "+" symbol and values not plotted.
PTVALUE	-1 0 0 0	
SCALE	150 200	
DRAWMAP		
TITLE		
4-D SHEKEL (X3,X4)		
DRAWMAP		
FINISH		

Example 2

<u>Command</u>	<u>Rest of Card</u>	<u>Comments</u>
TITLE		
6-HUMP CAMEL FUNCTION (TREND)		
PLOTSIZE	500 250	
POINTS	2 8 2	
LEVELSIZ	2 -1	
PTVALUE	2 1 -2 2	
FACTOR	-1	
SCALE	40 150	



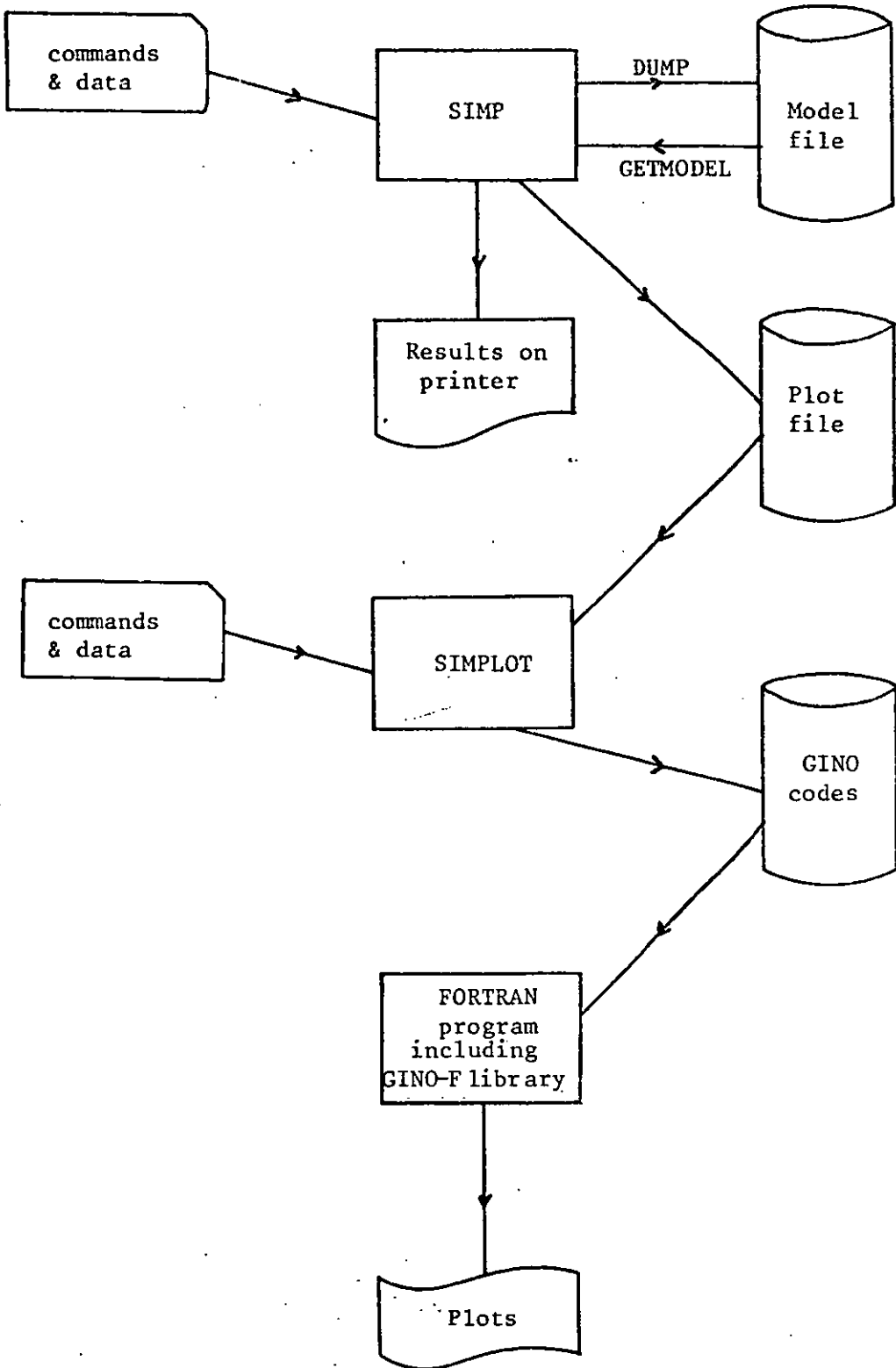


FIGURE A1: Diagram to illustrate overall system



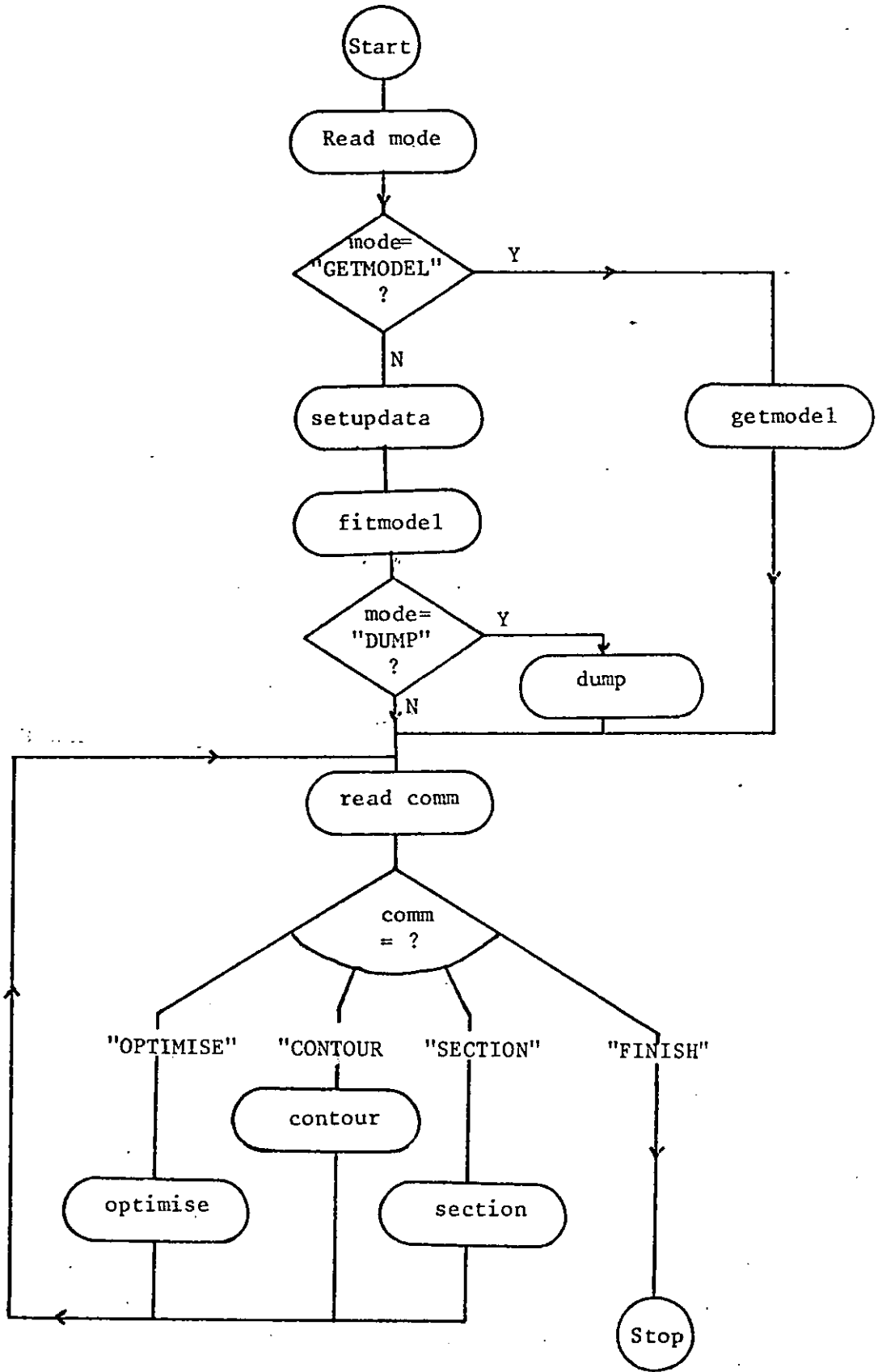
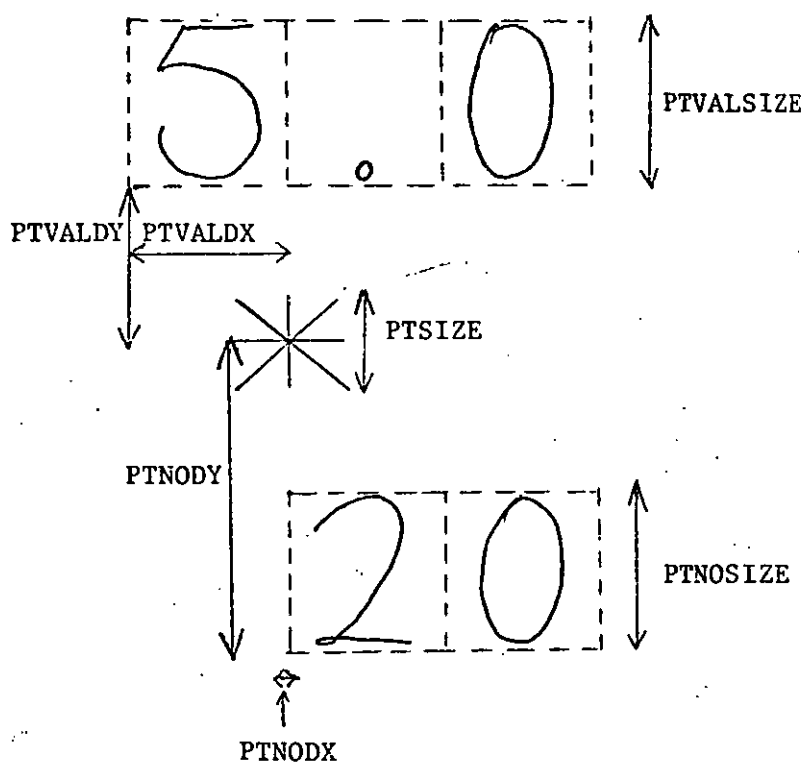


FIGURE A2: Outline flowchart of main routine of SIMP



**FIGURE A3:** Illustration of parameters for data point plotting in SIMPLOT

```

'PROC' CUTUP = ('REF'[,]'REAL' XPT,'REF'[]'REAL' XLOWER,XUPPER,
               'INT' NPOINTS,NDIMS,'REF''INT' NOLEFT,IPOINT) 'VOID':
'C'
  PROCEDURE TO DEFINE LOCATIONS OF LARGE NO. OF INITIAL PTS.
'C'
'BEGIN'
'REAL' LARGEST,SAVEUP,SAVELOW;
'INT' ICUT,REMAINDER,NOLEFTSAVE;
'IF' NOLEFT > 0
'THEN'
  LARGEST := 0.0;
  'FOR' I 'TO' NDIMS 'DO' 'BEGIN'
    'IF' XUPPER[I] - XLOWER[I] > LARGEST
    'THEN'
      LARGEST := XUPPER[I] - XLOWER[I]; ICUT := I
    'FI'
  'END';
SAVEUP := XUPPER[ICUT] ; SAVELOW := XLOWER[ICUT];
REMAINDER := NOLEFT - 2*(NOLEFT/'2);
'IF' REMAINDER = 1
'THEN'
  IPOINT 'PLUS' 1; NOLEFT 'MINUS' 1;
  'FOR' I 'TO' NDIMS 'DO'
    XPT[IPOINT,I] := 0.5*(XUPPER[I]+XLOWER[I])
  'FI';
NOLEFT := NOLEFT/'2;
NOLEFTSAVE := NOLEFT;
XUPPER[ICUT] := 0.5*(SAVEUP+SAVELOW);
CUTUP (XPT,XLOWER,XUPPER,NPOINTS,NDIMS,NOLEFT,IPOINT);
XUPPER[ICUT] := SAVEUP; XLOWER[ICUT] := 0.5*(SAVEUP+SAVELOW);
NOLEFT := NOLEFTSAVE;
CUTUP (XPT,XLOWER,XUPPER,NPOINTS,NDIMS,NOLEFT,IPOINT);
XUPPER[ICUT] := SAVEUP; XLOWER[ICUT] := SAVELOW
'FI'
'END';

```

```

'PROC' EXPLORE = ('REF'[,]'REAL' XPT,'REF'[]'REAL' XLOWER,XUPPER,
                 'REF''INT' IPOINT,'INT' NPOINTS,NDIMS) 'VOID':
'C'
  PROCEDURE TO DEFINE LOCATIONS OF INITIAL POINTS.
'C'
'BEGIN'
  [1:NDIMS]'REAL' XDUM,HIGH,LOW,ANISODUM;
  'BOOL' FINISHED,POINTOK;
  'INT' R,NHIGHS,IFIX,ISWITCH;
  'REAL' FACTOR := 4.0;
  'FOR' I 'TO' NDIMS 'DO' 'BEGIN'
    HIGH[I] := ((FACTOR-1.0)*XUPPER[I] + XLOWER[I])/FACTOR;
    LOW[I] := ((FACTOR-1.0)*XLOWER[I] + XUPPER[I])/FACTOR;
    ANISODUM[I] := 1.0
  'END';
  'FOR' R 'TO' NDIMS-1 'WHILE' IPOINT < NPOINTS 'DO' 'BEGIN'
    XDUM := HIGH;
    NHIGHS := NDIMS; FINISHED := 'FALSE'; IFIX := 1;
    'WHILE' 'NOT' FINISHED 'DO' 'BEGIN'
      POINTOK := 'TRUE';
      'FOR' K 'TO' IPOINT 'WHILE' POINTOK 'DO' 'BEGIN'
        POINTOK := POINTOK 'AND'
          DISTANCE(XDUM,XPT[K,],ANISODUM,NDIMS) > 1.0E-6
      'END';
      'IF' POINTOK
        'THEN'
          IPOINT 'PLUS' 1;
          XPT[IPOINT,] := XDUM;
          FINISHED := (IPOINT = NPOINTS)
        'FI';
      'FOR' J 'TO' NDIMS - R 'DO' 'BEGIN'
        ISWITCH := IFIX + R - 1 + J;
        'IF' ISWITCH > NDIMS 'THEN' ISWITCH 'MINUS' NDIMS 'FI';
        XDUM[ISWITCH] := HIGH[ISWITCH] + LOW[ISWITCH] - XDUM[ISWITCH]
      'END';
      IFIX 'PLUS' 1;
      'IF' IFIX > NDIMS 'THEN' IFIX 'MINUS' NDIMS 'FI';
      NHIGHS := 0;
      'FOR' I 'TO' NDIMS 'DO' 'BEGIN'
        'IF' XDUM[I] > LOW[I] 'THEN' NHIGHS 'PLUS' 1 'FI'
      'END';
      FINISHED := FINISHED 'OR' NHIGHS = NDIMS
    'END'
  'END'
'END';

```

```
'PROC' HCALC = ('REF' [] 'REF' [] 'REAL' S, 'REF' [,] 'REAL' XPT, DH, D2H,
               'REF' [] 'REAL' XLOWER, XUPPER, 'REAL' RO, 'INT' NPOINTS,
               NDIMS) 'REAL':
```

```
'C'
```

```
PROCEDURE TO CALCULATE SEPARATION MEASURE "H" FOR SET OF PTS.
```

```
'C'
```

```
'BEGIN'
```

```
'INT' IDUM, JDUM;
```

```
'REAL' HPART, HNEW := 0.0;
```

```
[1:NDIMS] 'REAL' ANISODUM;
```

```
'FOR' I 'TO' NDIMS 'DO' ANISODUM[I] := 1.0;
```

```
SETUP(S, XPT, ANISODUM, RO, NPOINTS, NDIMS);
```

```
'FOR' I 'TO' NPOINTS 'DO' 'BEGIN'
```

```
'FOR' K 'TO' NDIMS 'DO' 'BEGIN'
```

```
HPART := EXP(-((XPT[I,K]-XLOWER[K])^2/(0.5*RO^2)));
```

```
HNEW 'PLUS' HPART;
```

```
DH[I,K] := HPART*4.0*(XPT[I,K]-XLOWER[K])/RO^2;
```

```
D2H[I,K] := 4*HPART*(4*(XPT[I,K]-XLOWER[K])^2/RO^4-1/RO^2);
```

```
HPART := EXP(-((XPT[I,K]-XUPPER[K])^2/(0.5*RO^2)));
```

```
DH[I,K] 'PLUS' 4.0*HPART*(XPT[I,K]-XUPPER[K])/RO^2;
```

```
D2H[I,K] 'PLUS' 4*HPART*(4*(XPT[I,K]-XUPPER[K])^2/RO^4
          - 1/RO^2);
```

```
HNEW 'PLUS' HPART
```

```
'END';
```

```
'FOR' J 'TO' NPOINTS 'DO' 'BEGIN'
```

```
'IF' I # J
```

```
'THEN'
```

```
'IF' I > J
```

```
'THEN' IDUM := I; JDUM := J
```

```
'ELSE' IDUM := J; JDUM := I
```

```
'FI';
```

```
HPART := S[IDUM][JDUM];
```

```
HNEW 'PLUS' HPART;
```

```
'FOR' K 'TO' NDIMS 'DO' 'BEGIN'
```

```
DH[I,K] 'PLUS' 2.0*HPART*(XPT[I,K]-XPT[J,K])/RO^2;
```

```
D2H[I,K] 'PLUS' 2*HPART*((XPT[I,K]-XPT[J,K])^2/RO^4
          - 1/RO^2)
```

```
'END'
```

```
'FI'
```

```
'END'
```

```
'END';
```

```
HNEW
```

```
'END';
```

```

'PROC' UPDATEH = ('REF'[] 'REF'[] 'REAL' S, 'REF'[, ] 'REAL' XPT, DH, D2H,
                 'REF'[] 'REAL' XLOWER, XUPPER, 'REF' 'REAL' HNEW,
                 'REAL' RO, NEWX, 'INT' ISHIFT, KSHIFT, NPOINTS, NDIMS)
                 'VOID':

'C'
  PROCEDURE TO UPDATE "H" VALUE FOR CHANGED POSITION OF PT.
'C'
'BEGIN'
'REAL' HPART, OLDHPART, DHPART, D2HPART;
'INT' IDUM, JDUM;
[1:NPOINTS]'REAL' NEWS;
'FOR' I 'TO' NPOINTS 'DO'
  'IF' I # ISHIFT
  'THEN'
    'REAL' DIST := 0.0;
    'FOR' K 'TO' NDIMS 'DO'
      'IF' K # KSHIFT
      'THEN' DIST 'PLUS' (XPT[I, K]-XPT[ISHIFT, K])^2
      'ELSE' DIST 'PLUS' (XPT[I, K]-NEWX)^2
      'FI';
    NEWS[I] := EXP(-(DIST/(2.0*RO^2)))
  'FI';
NEWS[ISHIFT] := 1.0;
OLDHPART := EXP(-((XPT[ISHIFT, KSHIFT]-XLOWER[KSHIFT])^2/(0.5*RO^2)));
HPART := EXP(-((NEWX-XLOWER[KSHIFT])^2/(0.5*RO^2)));
HNEW 'PLUS' HPART - OLDHPART;
DH[ISHIFT, KSHIFT] 'PLUS' 4*(HPART*NEWX-OLDHPART*XPT[ISHIFT, KSHIFT]
+ XLOWER[KSHIFT]*(OLDHPART-HPART))/RO^2;
D2H[ISHIFT, KSHIFT] 'PLUS' 4*(4*HPART*(NEWX-XLOWER[KSHIFT])^2
- 4*OLDHPART*(XPT[ISHIFT, KSHIFT]-XLOWER[KSHIFT])^2
+ RO^2*(OLDHPART-HPART))/RO^4;
OLDHPART := EXP(-((XPT[ISHIFT, KSHIFT]-XUPPER[KSHIFT])^2/(0.5*RO^2)));
HPART := EXP(-((NEWX-XUPPER[KSHIFT])^2/(0.5*RO^2)));
HNEW 'PLUS' HPART - OLDHPART;
DH[ISHIFT, KSHIFT] 'PLUS' 4*(HPART*NEWX-OLDHPART*XPT[ISHIFT, KSHIFT]
+ XUPPER[KSHIFT]*(OLDHPART-HPART))/RO^2;
D2H[ISHIFT, KSHIFT] 'PLUS' 4*(4*HPART*(NEWX-XUPPER[KSHIFT])^2
- 4*OLDHPART*(XPT[ISHIFT, KSHIFT]-XUPPER[KSHIFT])^2
+ RO^2*(OLDHPART-HPART))/RO^4;
'FOR' J 'TO' NPOINTS 'DO'
  'IF' J # ISHIFT
  'THEN'
    'IF' ISHIFT > J
    'THEN' IDUM := ISHIFT; JDUM := J
    'ELSE' IDUM := J; JDUM := ISHIFT
    'FI';
    OLDHPART := S[IDUM][JDUM];
    HPART := NEWS[J];
    HNEW 'PLUS' 2.0*(HPART - OLDHPART);
    DHPART := 2*(HPART*NEWX - OLDHPART*XPT[ISHIFT, KSHIFT]
+ XPT[J, KSHIFT]*(OLDHPART-HPART))/RO^2;
    DH[ISHIFT, KSHIFT] 'PLUS' DHPART;
    DH[J, KSHIFT] 'MINUS' DHPART;
    D2HPART := 2*(HPART*(NEWX-XPT[J, KSHIFT])^2 - OLDHPART*
(XPT[ISHIFT, KSHIFT]-XPT[J, KSHIFT])^2)/RO^4;
    D2HPART 'PLUS' 2.0*(OLDHPART-HPART)/RO^2;
    D2H[ISHIFT, KSHIFT] 'PLUS' D2HPART;
    D2H[J, KSHIFT] 'PLUS' D2HPART;

```

```

'FOR' K 'TO' NDIMS 'DO'
  'IF' K # KSHIFT
  'THEN'
    DHPART := 2.0*(HPART-OLDHPART)*(XPT[ISHIFT,K]-XPT[J,K])/RO^2;
    D2HPART := 2*(HPART-OLDHPART)*(XPT[ISHIFT,K]-
      XPT[J,K])^2/RO^4;
    D2HPART 'PLUS' 2.0*(OLDHPART-HPART)/RO^2;
    DH[ISHIFT,K] 'PLUS' DHPART;
    DH[J,K] 'MINUS' DHPART;
    D2H[ISHIFT,K] 'PLUS' D2HPART;
    D2H[J,K] 'PLUS' D2HPART;
  'FI';
  S[IDUM][JDUM] := NEWS[J]
'FI';
XPT[ISHIFT,KSHIFT] := NEWX
'END';

```

```

'PROC' SPREADEM = ('REF'[,]'REAL' XPT,'REF'[]'REAL' XLOWER,XUPPER,
                  'REAL' AVSEP,'INT' NPOINTS,NDIMS) 'VOID':

```

```

'C'

```

```

PROCEDURE TO MOVE INITIAL CONFIGURATION OF PTS INTO BETTER
POSITIONS USING REPULSIVE FUNCTION.

```

```

'C'

```

```

'BEGIN'

```

```

'REAL' HNEW,HOLD,DHMAX,HPART,STEP;

```

```

'REAL' RO := 0.5*AVSEP;

```

```

'REAL' SHIFT := 0.5*RO;

```

```

'INT' NITS := 0;

```

```

'INT' IMAX,KMAX,IDUM,JDUM;

```

```

'INT' IMAXLAST, KMAXLAST;

```

```

IMAXLAST := 0; KMAXLAST := 0;

```

```

'REAL' LASTSTEP := 0.0;

```

```

[1:NDIMS]'REAL' ANISODUM;

```

```

[1:NPOINTS,1:NDIMS]'REAL' DH,D2H;

```

```

[1:NPOINTS]'REF'[]'REAL' S;

```

```

'FOR' I 'TO' NPOINTS 'DO' S[I] := [1:I]'REAL';

```

```

'FOR' I 'TO' NDIMS 'DO' ANISODUM[I] := 1.0;

```

```

'BOOL' FINISHED := 'FALSE';

```

```

'BOOL' DEBUG := 'FALSE';

```

```

HOLD := 1.0&6;

```

```

'REAL' NEWX;

```

```

HNEW := HCALC(S,XPT,DH,D2H,XLOWER,XUPPER,RO,NPOINTS,NDIMS);

```

```

OUTF(STANDOUT,$LL"INITIAL H VALUE ="2X<4.4>$,HNEW);

```

```

'WHILE' 'NOT' FINISHED 'DO''BEGIN'

```

```

  NITS 'PLUS' 1;

```

```

  DHMAX := 0.0;

```

```

  'FOR' I 'TO' NPOINTS 'DO'

```

```

    'FOR' K 'TO' NDIMS 'DO'

```

```

      'IF' 'ABS'DH[I,K] > DHMAX

```

```

        'AND' (XPT[I,K]-XLOWER[K]) > -0.01*(XUPPER[K]-XLOWER[K])*

```

```

        'SIGN'(DH[I,K]) 'AND' (XUPPER[K]-XPT[I,K]) > 0.01*

```

```

        (XUPPER[K]-XLOWER[K])*'SIGN'(DH[I,K])

```

```

        'THEN' IMAX := I; KMAX := K; DHMAX := 'ABS'DH[I,K]

```

```

      'FI';

```

```

  'IF' DEBUG

```

```

  'THEN'

```

```

    OUTF(STANDOUT,$L<5>,>2X<2.5&2>,>2X<5>,>1X<5>,>2X<2.5&2>,$,
          (NITS,HNEW,IMAX,KMAX,DHMAX))

```

```

  'FI';

```

```

FINISHED := HOLD - HNEW < 1.0/NPOINTS 'OR' NITS > 2*NDIMS*NPOINTS;

```

```

HOLD := HNEW;

```

```

'IF' DHMAX <= 1.0&-6

```

```

'THEN'

```

```

  'FOR' I 'TO' NPOINTS 'WHILE' FINISHED 'DO'

```

```

    'FOR' K 'TO' NDIMS 'WHILE' FINISHED 'DO'

```

```

      'IF' D2H[I,K] < 0.0

```

```

      'THEN'

```

```

        FINISHED := 'FALSE';

```

```

        IMAX := I; KMAX := K;

```

```

        DHMAX := 1.0&-6; DH[I,K] := DHMAX

```

```

      'FI'

```

```

  'FI';

```



```

'IF' 'NOT' FINISHED
'THEN'
  'IF' D2H [IMAX, KMAX] > 0.0
  'THEN'
    STEP := MAX((-SHIFT, MIN((SHIFT, DH [IMAX, KMAX]/D2H [IMAX,
      KMAX]))));
    FINISHED := 'ABS'STEP < 0.001*(XUPPER [KMAX]-XLOWER [KMAX])
  'ELSE' STEP := DH [IMAX, KMAX]*1.0*SHIFT/DHMAX
  'FI';
  'IF' STEP > 0.0
  'THEN' STEP := MIN((STEP, (XUPPER [KMAX]-XPT [IMAX, KMAX])/2.0))
  'ELSE' STEP := MAX((STEP, (XLOWER [KMAX]-XPT [IMAX, KMAX])/2.0))
  'FI';
  'IF' IMAX = IMAXLAST 'AND' KMAX = KMAXLAST
  'AND' STEP*LASTSTEP < 0
  'THEN' STEP := -0.5*LASTSTEP
  'FI';
  LASTSTEP := STEP;
  IMAXLAST := IMAX; KMAXLAST := KMAX;
  'IF' DEBUG 'THEN'
  OUTF (STANDOUT, $2X<2.5&2>, 2X<5.4>$, (D2H [IMAX, KMAX], STEP));
  'SKIP' 'FI';

  NEWX := MAX((XLOWER [KMAX], MIN((XUPPER [KMAX], XPT [IMAX, KMAX]
    + STEP)))));
  UPDATEH (S, XPT, DH, D2H, XLOWER, XUPPER, HNEW, RO, NEWX, IMAX, KMAX,
    NPOINTS, NDIMS)
  'FI'
'END';
OUTF (STANDOUT, $L2X"FINAL H VALUE ="2X<4.4>, 2X"AFTER"<4>" ITERATIONS"$,
  (HNEW, NITS))
'END';

```

```

'PROC' INITIALPTS = ('REF'[,]'REAL' XPT,'REF'[]'REAL' XLOWER,XUPPER,
                    'REF''REAL' AVSEP,'INT' NPOINTS,NDIMS) 'VOID':
  'C'
    PROCEDURE TO DEFINE FINAL POSITIONS OF INITIAL PTS.
  'C'
'BEGIN'
  'INT' IPOINT,NOLEFT;
  IPOINT := 0; NOLEFT := NPOINTS;
  AVSEP := 1.0;
  'FOR' I 'TO' NDIMS 'DO' AVSEP 'TIMES' (XUPPER[I]-XLOWER[I]);
  'IF' NPOINTS > 0 'THEN'
  AVSEP := EXP(LN(AVSEP/NPOINTS)/NDIMS);
  'SKIP'
  'ELSE' AVSEP := EXP(LN(AVSEP)/NDIMS)
  'FI';
  OUTF(STANDOUT,$LL"AVERAGE SEPARATION BETWEEN PTS ="2X<6.4>L$,AVSEP);
  'IF' NPOINTS >= 2^NDIMS
  'THEN'
    CUTUP(XPT,XLOWER,XUPPER,NPOINTS,NDIMS,NOLEFT,IPOINT)
  'ELSE'
    EXPLORE(XPT,XLOWER,XUPPER,IPOINT,NPOINTS,NDIMS);
    'IF' IPOINT < NPOINTS
    'THEN'
      NOLEFT := NPOINTS - IPOINT;
      CUTUP(XPT,XLOWER,XUPPER,NPOINTS,NDIMS,NOLEFT,IPOINT)
    'FI'
  'FI';
  PRINT((NEWLINE,"STARTING POSITIONS OF INITIAL PTS -",NEWLINE));
  'FOR' I 'TO' NPOINTS 'DO'
    OUTF(STANDOUT,$L<5>,N(NDIMS)(2X<4.4>)$,(I,XPT[I,]));
  SPRADEM(XPT,XLOWER,XUPPER,AVSEP,NPOINTS,NDIMS);
  'SKIP'
'END';

```

```

'PROC' ANISOCALC = ('REF'[,]'REAL' XPT,'REF'[]'REAL' Z,ANISO,
                  'REF''REAL' MU,RO,SD,AVSEP,ANISOMIN,ANISOMAX,
                  'INT' NPOINTS,NDIMS) 'VOID':
'C'
  PROCEDURE TO ESTIMATE ANISOTROPY FACTORS.
'C'
'BEGIN'
  'INT' NCOUNT := 0; 'INT' NREF := 4;
  'REAL' DIST,MAXDIST,ALPHAMEAN;
  [1:NDIMS]'REAL' ALPHA;
  [1:NPOINTS]'REAL' ZSCALE;
  [1:NREF*NPOINTS]'REAL' Y;
  [1:NREF*NPOINTS,1:NDIMS]'REAL' DELTAX2;
  'BOOL' DEBUG := 'FALSE';
  MAXDIST := 1.5*AVSEP;
  'CLEAR' DELTAX2;
  'FOR' I 'TO' NDIMS 'DO' ALPHA[I] := 1.0;
  'FOR' I 'TO' NPOINTS 'DO' 'BEGIN'
    ZSCALE[I] := (Z[I]-MU)/SD;
    'FOR' J 'TO' I-1 'WHILE' NCOUNT < NREF*I 'DO' 'BEGIN'
      'IF' ZSCALE[I]*ZSCALE[J] > 0.0
        'THEN'
          DIST := DISTANCE(XPT[I,],XPT[J,],ALPHA,NDIMS);
          'IF' DIST < MAXDIST
            'THEN'
              NCOUNT 'PLUS' 1;
              Y[NCOUNT] := -LN(COVAREST(ZSCALE[I],ZSCALE[J]));
              'IF' Y[NCOUNT] < 1.0
                'THEN'
                  'FOR' K 'TO' NDIMS 'DO'
                    DELTAX2[NCOUNT,K] := (XPT[I,K]-XPT[J,K])^2
                  'ELSE' NCOUNT 'MINUS' 1
                'FI'
              'FI'
            'END'
          'END'
        'END'
      REGRESS(DELTA2,Y,ALPHA,NDIMS,NCOUNT);
      ALPHAMEAN := 0.0; NCOUNT := 0;
      'FOR' I 'TO' NDIMS 'DO'
        'IF' ALPHA[I] > 0.0
          'THEN'
            ALPHAMEAN 'PLUS' ALPHA[I]; NCOUNT 'PLUS' 1
          'FI';
        'IF' NCOUNT = 0
          'THEN'
            NCOUNT := 1;
            ALPHAMEAN := 1.0/AVSEP^2
          'FI';
        ALPHAMEAN 'DIV' NCOUNT;
        RO := SQRT(1.0/(2.0*ALPHAMEAN));
        OUTF(STANDOUT,$LL"ESTIMATED RO ="2X<4.4>L$,RO);
        PRINT((NEWLINE,NEWLINE,"ALPHA VALUES AND ANISOTROPY FACTORS -",
              NEWLINE));
        'FOR' I 'TO' NDIMS 'DO' 'BEGIN'
          'IF' ALPHA[I] > 0.0
            'THEN' ANISO[I] := ALPHA[I]/ALPHAMEAN
            'ELSE' ANISO[I] := ANISOMIN
          'FI';
          ANISO[I] := MAX((ANISOMIN,MIN((ANISO[I],ANISOMAX))));
          OUTF(STANDOUT,$L<5>,2X<2.5&2>,2X<4.4>$,(I,ALPHA[I],ANISO[I]))
        'END'
      'END';

```

```
'PROC' AVERAGEPTS = ('REF'[,]'REAL' XPT,XAV,'REF'[]'REAL' ZAV,Z,
                     ANISO,XLOWER,XUPPER,'INT' NOAV,NPOINTS,NDIMS)
                     'VOID':
```

```
'C'
```

```
PROCEDURE TO DEFINE POSITIONS AND VALUES OF AVERAGE PTS.
```

```
'C'
```

```
'BEGIN'
```

```
'REAL' AVSEP,RO,WTSUM,DIST,COVAR;
INITIALPTS (XAV,XLOWER,XUPPER,AVSEP,NOAV,NDIMS);
PRINT((NEWLINE,NEWLINE,"POSITIONS AND VALUES OF AVERAGE PTS -",
      NEWLINE));
'CLEAR' ZAV;
RO := 0.5*AVSEP;
'FOR' I 'TO' NOAV 'DO' 'BEGIN'
  WTSUM := 0.0;
  'FOR' J 'TO' NPOINTS 'DO' 'BEGIN'
    DIST := DISTANCE(XPT[J,],XAV[I,],ANISO,NDIMS);
    COVAR := EXP(-(DIST^2/(2.0*RO^2)));
    WTSUM 'PLUS' COVAR;
    ZAV[I] 'PLUS' Z[J]*COVAR
  'END';
  ZAV[I] 'DIV' WTSUM;
  OUTF(STANDOUT,$L<3>,2X<2.5&2>,2XN(NDIMS)(1X<3.4>)$,
      (I,ZAV[I],XAV[I,]))
'END';
'END';
```

```
'PROC' ERRORSUM = ('REF'[,]'REAL' X,'REF'[]'REAL' Z,ANISO,
                  'REAL' RO,MU,'INT' NPOINTS,NDIMS) 'REAL':
```

```
'C'
```

```
PROCEDURE TO COMPUTE ERROR SUM OF SQUARES FOR GIVEN RO VALUE.
```

```
'C'
```

```
'BEGIN'
```

```
[1:NPOINTS]'REF'[]'REAL' S;
'FOR' I 'TO' NPOINTS 'DO' S[I] := [1:I]'REAL';
[1:NPOINTS]'BOOL' PIVOTTED;
'REAL' ESTIMATE,ERRORS; ERRORS := 0.0;
'INT' NOUSED := 0;
SETUP(S,X,ANISO,RO,NPOINTS,NDIMS);
'FOR' I 'TO' NPOINTS 'DO' 'BEGIN'
  ESTIMATE := MU;
  'FOR' J 'TO' I-1 'DO'
    'IF' PIVOTTED[J] 'THEN' ESTIMATE 'MINUS' S[I][J]*(Z[J]-MU) 'FI';
  ERRORS 'PLUS' (Z[I]-ESTIMATE)^2/S[I][I];
  PIVOT(S,PIVOTTED,I,NPOINTS);
  'IF' PIVOTTED[I] 'THEN' NOUSED 'PLUS' 1 'FI'
'END';
ERRORS/NOUSED
'END';
```

```

'PROC' MINEST = ('REF'[] 'REAL' XX,YY, 'REF' 'BOOL' MINFOUND) 'REAL':
'C'
  PROCEDURE TO ESTIMATE MIN OF FUNCTION FROM 3 PTS.
'C'
'BEGIN'
  'REAL' A,B,C,WORKER,MINX;
  A := ((XX[2]-XX[3])*(YY[1]-YY[2]) - (XX[1]-XX[2])*(YY[2]-YY[3]))/
        ((XX[1]-XX[2])*(XX[2]-XX[3])*(XX[1]-XX[3]));
  B := (YY[1] - YY[2] - A*(XX[1]-XX[2])*(XX[1]+XX[2]))/(XX[1]-XX[2]);
  C := YY[1] - A*XX[1]^2 - B*XX[1];
  MINFOUND := A > 0.0;
  'IF' MINFOUND
  'THEN' MINX := -B/(2.0*A)
  'ELSE'
    'REAL' MINY := 1.0&18; WORKER := 0.0;
    'FOR' I '_TO' 3 'DO' 'BEGIN'
      WORKER 'PLUS' XX[I]/3.0;
      'IF' YY[I] < MINY
      'THEN' MINY := YY[I] ; MINX := XX[I]
    'FI'
  'END';
  MINX := 3.0*MINX - 2.0*WORKER
'FI';
MINX
'END';

```

```

'PROC' ESTIMATED RO = ('REF'[,]'REAL' X,'REF'[]'REAL' Z,ANISO,
                      'REAL' MU,STARTRO,'INT' NPOINTS,NDIMS) 'REAL':
'C'
  PROCEDURE TO ESTIMATE RO BY MINIMISING ERROR SUM OF SQUARES.
'C'
'BEGIN'
  'INT' IDROP,NITS; NITS := 0;
  'REAL' RO,CLOSENESS,MAXESUM; CLOSENESS := 0.025*STARTRO;
  'BOOL' FOUNDIT,MINFOUND; FOUNDIT := 'FALSE';
  [1:3]'REAL' ESUM,TRIALRO;
  PRINT( (NEWLINE,NEWLINE,
          "VALUES OF RO AND ERROR VARIANCE FOR ESTIMATION -",NEWLINE));
  'FOR' I 'TO' 3 'DO' 'BEGIN'
    TRIALRO[I] := STARTRO*(2.0^(I-2));
    ESUM[I] := ERRORSUM(X,Z,ANISO,TRIALRO[I],MU,NPOINTS,NDIMS);
    OUTF(STANDOUT,$L<3>,>2X<4.4>,>2X<2.5&2>,$,(I,TRIALRO[I],ESUM[I]))
  'END';
  'WHILE' NITS < 20 'AND' 'NOT' FOUNDIT 'DO' 'BEGIN'
    RO := MINEST(TRIALRO,ESUM,MINFOUND);
    NITS 'PLUS' 1;
    RO := MAX((RO,0.1*STARTRO/NITS));
    MAXESUM := 0.0;
    'FOR' I 'TO' 3 'DO'
      'IF' ESUM[I] > MAXESUM
        'THEN' IDROP := I; MAXESUM := ESUM[I]
      'FI';
    FOUNDIT := MINFOUND 'AND' 'ABS'(RO-TRIALRO[IDROP]) < CLOSENESS;

  'IF' 'NOT' FOUNDIT
  'THEN'
    MAXESUM := ERRORSUM(X,Z,ANISO,RO,MU,NPOINTS,NDIMS);
    'IF' MAXESUM > ESUM[IDROP]
    'THEN'
      RO := 0.5*(RO+TRIALRO[IDROP]);
      MAXESUM := ERRORSUM(X,Z,ANISO,RO,MU,NPOINTS,NDIMS)
    'FI';
    ESUM[IDROP] := MAXESUM;
    TRIALRO[IDROP] := RO;
    OUTF(STANDOUT,$L<3>,>2X<4.4>,>2X<2.5&2>,>1X<3>,$,
          (NITS,RO,MAXESUM,IDROP))
  'FI'
'END';
RO
'END';

```

```

'PROC' QUICK RO = ('REF'[,]'REAL' X,'REF'[]'REAL' Z,ANISO,'REAL' MU,SD,
                  'INT' NPOINTS,NDIMS) 'REAL':
'C'
  PROCEDURE FOR QUICK ESTIMATION OF RO.
'C'
'BEGIN'
[1:NPOINTS]'REAL' ZSCALE,ROVAL; 'CLEAR' ROVAL;
[1:NPOINTS]'INT' POINTER; 'CLEAR' POINTER;
'INT' NEAREST,ICOUNT;
'REAL' XCOVAR,RO,ROMEAN,ROSD,DIST,MINDIST;
'INT' NFAIL := 0;
ICOUNT := 0;
'FOR' I 'TO' NPOINTS 'DO' ZSCALE[I] := (Z[I]-MU)/SD;
'FOR' I 'TO' NPOINTS 'DO' 'BEGIN'
  NEAREST := I; MINDIST := 1.0&12;
  'FOR' J 'TO' NPOINTS 'DO'
    'IF' I # J
      'THEN'
        DIST := DISTANCE(X[I,],X[J,],ANISO,NDIMS);
        'IF' DIST < MINDIST 'AND' POINTER[J] # I
          'AND' ZSCALE[I]*ZSCALE[J] > 0
            'THEN'
              NEAREST := J; MINDIST := DIST
        'FI'
      'FI';
  'IF' NEAREST # I
    'THEN'
      ICOUNT 'PLUS' 1;
      XCOVAR := COVAREST(ZSCALE[I],ZSCALE[NEAREST]);
      'IF' XCOVAR < 1.0&-5 'THEN' NFAIL 'PLUS' 1 'FI';
      POINTER[I] := NEAREST;
      RO := MIN((4.0*MINDIST,MINDIST*SQRT(-1.0/(2.0*LN(XCOVAR)))));
      ROVAL[ICOUNT] := RO;
      OUTF(STANDOUT,$L2(2X<4>),2(2X<2.5&2>),3(2X<4.4>)$,(I,NEAREST,
        ZSCALE[I],ZSCALE[NEAREST],MINDIST,XCOVAR,RO))
    'FI'
  'END';
PRINT((NEWLINE,"STATISTICS FOR ESTIMATED RO VALUES -",NEWLINE));
STATISTICS(ROVAL,ICOUNT,ROMEAN,RO,ROSD);
'IF' NFAIL > ICOUNT/2 'THEN' RO := ROMEAN 'FI';
RO
'END';

```

```

'PROC' POINT VALUE = ('REF'[,]'REAL' XPT,'REF'[]'REAL' X,GAMMA,ANISO,
                    DFDX,'REAL' MU,RO,'INT' NPOINTS,NDIMS) 'REAL':
'C'
  PROCEDURE TO ESTIMATE FUNCTION VALUE AT UNKNOWN PT.
'C'
'BEGIN'
'REAL' ESTIMATE,DIST2,COVAR;
ESTIMATE := MU;
'FOR' I 'TO' NPOINTS 'DO' 'BEGIN'
  DIST2 := DISTANCE(X,XPT[I,],ANISO,NDIMS)^2;
  'IF' DIST2 < 25.0*RO^2
  'THEN'
    COVAR := EXP(-(DIST2/(2.0*RO^2)))*GAMMA[I];
    ESTIMATE 'PLUS' COVAR;
    'FOR' K 'TO' NDIMS 'DO'
      DFDX[K] 'PLUS' COVAR*ANISO[K]*(XPT[I,K]-X[K])/RO^2
  'FI'
'END';
ESTIMATE
'END';

```

```

'PROC' INTERPOLATOR = ('REF'[,]'REAL' XPT,XAV,'REF'[]'REAL' X,GAMMA,
                    DELTA,ANISO,DFDX,'REF''REAL' MODG,'REAL' MU,
                    RO,ROLONG,'INT' NPOINTS,NOAV,NDIMS) 'REAL':
'C'
  INTERPOLATING FUNCTION USING 2-STAGE MODEL.
'C'
'BEGIN'
'REAL' INTERP;
'CLEAR' DFDX;
INTERP := POINTVALUE(XAV,X,DELTA,ANISO,DFDX,MU,ROLONG,NOAV,NDIMS)
        + POINTVALUE(XPT,X,GAMMA,ANISO,DFDX,0.0,RO,NPOINTS,NDIMS);
MODG := 0.0;
'FOR' I 'TO' NDIMS 'DO' MODG 'PLUS' 'ABS' DFDX[I];
INTERP
'END';

```



```
'PROC' PREPARE = ('REF'[ , ] 'REAL' XPT, 'REF' [ ] 'REAL' Z, ANISO, GAMMA,
                 'REAL' RO, MU, 'INT' NPOINTS, NDIMS) 'VOID':
```

```
'C'
```

```
PROCEDURE TO COMPUTE GAMMA VALUES.
```

```
'C'
```

```
'BEGIN'
```

```
[1:NPOINTS]'REF' [ ] 'REAL' S;
```

```
[1:NPOINTS]'BOOL' PIVOTTED;
```

```
'INT' IDUM, JDUM;
```

```
'FOR' I 'TO' NPOINTS 'DO' S[I] := [1:I]'REAL';
```

```
SETUP (S, XPT, ANISO, RO, NPOINTS, NDIMS);
```

```
'FOR' I 'TO' NPOINTS 'DO' PIVOT (S, PIVOTTED, I, NPOINTS);
```

```
'FOR' I 'TO' NPOINTS 'DO' 'BEGIN'
```

```
GAMMA[I] := 0.0;
```

```
'FOR' J 'TO' NPOINTS 'DO' 'BEGIN'
```

```
'IF' I > J
```

```
'THEN' IDUM := I; JDUM := J
```

```
'ELSE' IDUM := J; JDUM := I
```

```
'FI';
```

```
'IF' PIVOTTED[J]
```

```
'THEN' GAMMA[I] 'MINUS' S[IDUM][JDUM]*(Z[J]-MU)
```

```
'FI'
```

```
'END';
```

```
'IF' 'NOT' PIVOTTED[I] 'THEN' GAMMA[I] := 0.0 'FI'
```

```
'END'
```

```
'END';
```

```
'PROC' FIND = ('REF'[ , ] 'REAL' XPT, XAV, 'REF' [ ] 'REAL' X, GAMMA, DELTA,
              ANISO, 'REAL' ANGLE, CONTOUR, TOL, DLLIM, MU, RO, ROLONG,
              'INT' IX, IY, NPOINTS, NOAV, NDIMS) 'BOOL':
```

```
'C'
```

```
PROCEDURE TO FIND CONTOUR VALUE ALONG GIVEN LINE.
```

```
'C'
```

```
'BEGIN'
```

```
'INT' ITS := 0;
```

```
'REAL' MODG, DL, DZDL, VALUE;
```

```
[1:NDIMS]'REAL' DFDX;
```

```
VALUE := INTERPOLATOR (XPT, XAV, X, GAMMA, DELTA, ANISO, DFDX, MODG, MU, RO,
                      ROLONG, NPOINTS, NOAV, NDIMS);
```

```
'WHILE' 'ABS' (VALUE-CONTOUR) > TOL 'AND' ITS <= 10 'DO' 'BEGIN'
```

```
ITS 'PLUS' 1;
```

```
DZDL := DFDX[IX]*COS(ANGLE) + DFDX[IY]*SIN(ANGLE);
```

```
'IF' 'ABS' DZDL < 1.0E-9 'THEN' DZDL := 1.0E-9 'FI';
```

```
DL := (CONTOUR-VALUE)/DZDL;
```

```
DL := DL/(1.0 + 'ABS' DL/DLLIM);
```

```
X[IX] 'PLUS' DL*COS(ANGLE);
```

```
X[IY] 'PLUS' DL*SIN(ANGLE);
```

```
VALUE := INTERPOLATOR (XPT, XAV, X, GAMMA, DELTA, ANISO, DFDX, MODG, MU, RO,
                      ROLONG, NPOINTS, NOAV, NDIMS)
```

```
'END';
```

```
'ABS' (VALUE-CONTOUR) < TOL
```

```
'END';
```

```

'PROC' DIVIDE = ('REF'[,]'REAL' XPT,XAV,XYZ,'REF'[]'REAL' X,GAMMA,
                DELTA,ANISO,'REAL' CONTOUR,TOL, DLLIM,MU,RO,ROLONG,
                'INT' IX,IY,NPOINTS,NOAV,NDIMS) 'BOOL':
'C'
  PROCEDURE TO DIVIDE LINE TO FIND REF. PT.
'C'
'BEGIN'
'REAL' FRACTION,VALUE,ANGLE,DZDL,MODG,AX,AY;
'INT' IDIR,NITS,IXX,I;
[1:3]'REAL' XYZL,XYZU;
[1:NDIMS]'REAL' DFDX;
FRACTION := (CONTOUR - XYZ [3, 2])/(XYZ [3, 1]-XYZ [3, 2]);
FRACTION := MAX((0.001,MIN((0.999,FRACTION))));
AX := XYZ [1, 2] + FRACTION*(XYZ [1, 1]-XYZ [1, 2]);
AY := XYZ [2, 2] + FRACTION*(XYZ [2, 1]-XYZ [2, 2]);
FRACTION := XYZ [1, 1] - XYZ [1, 2];
'IF' 'ABS' FRACTION < 1.0E-9 'THEN' FRACTION := 1.0E-9 'FI';
ANGLE := ARCTAN((XYZ [2, 1]-XYZ [2, 2])/FRACTION);
IXX := -'SIGN' FRACTION;
IDIR := 'SIGN'(XYZ [3, 2]-XYZ [3, 1]);
I := (3+IDIR)'/2;
XYZU := XYZ [, I]; XYZL := XYZ [, 3-I];
IDIR := IDIR*IXX; DZDL := -IDIR; NITS := 0;
'WHILE' DZDL*IDIR < 0 'AND' NITS <= 10 'DO' 'BEGIN'
  NITS 'PLUS' 1;
  X[IX] := AX;
  X[IY] := AY;
  VALUE := INTERPOLATOR(XPT,XAV,X,GAMMA,DELTA,ANISO,DFDX,MODG,MU,
                        RO,ROLONG,NPOINTS,NOAV,NDIMS);
  DZDL := DFDX[IX]*COS(ANGLE) + DFDX[IY]*SIN(ANGLE);
  'IF' VALUE > CONTOUR
  'THEN' XYZU[1] := AX; XYZU[2] := AY; XYZU[3] := VALUE
  'ELSE' XYZL[1] := AX; XYZL[2] := AY; XYZL[3] := VALUE
  'FI';
  AX := 0.5*(XYZL[1]+XYZU[1]); AY := 0.5*(XYZL[2]+XYZU[2])
'END';
X[IX] := AX; X[IY] := AY;
FIND(XPT,XAV,X,GAMMA,DELTA,ANISO,ANGLE,CONTOUR,TOL, DLLIM,MU,RO,
     ROLONG,IX,IY,NPOINTS,NOAV,NDIMS)
'END';

```

```

'PROC' SELECT = ('REF' [,] 'REAL' XPT, XAV, 'REF' [] 'REAL' X, GAMMA, DELTA,
                ANISO, XREF, YREF, XDATA, YDATA, ZDATA, 'REAL' CONTOUR,
                TOL, DLLIM, MU, RO, ROLONG, MEDIAN, 'REF' 'INT' NREF,
                'INT' IX, IY, NDATA, NPOINTS, NOAV, NDIMS) 'VOID':
'C'
  PROCEDURE TO DEFINE REFERENCE PTS.
'C'
'BEGIN'
  'REAL' DIST, DMIN;
  'INT' K; NREF := 0;
  [1:3, 1:2] 'REAL' XYZ;
  [1:NDATA] 'BOOL' IBEEN; 'CLEAR' IBEEN;
  'BOOL' FINISHED := 'FALSE';
  'FOR' I 'FROM' NDATA 'BY' -1 'TO' 1 'WHILE' 'NOT' FINISHED 'DO'
  'BEGIN'
    'IF' (ZDATA[I]-CONTOUR)*(MEDIAN-CONTOUR) < 1.0&-12
      'OR' I > NDATA - 4
    'THEN'
      DMIN := 1.0&12;
      FINISHED := 'TRUE';
      'FOR' J 'TO' NDATA 'DO'
        'IF' I # J 'AND' 'NOT' IBEEN[J] 'AND'
          (ZDATA[I]-CONTOUR)*(ZDATA[J]-CONTOUR) < 1.0&-12
        'THEN'
          DIST := SQRT((XDATA[I]-XDATA[J])^2 + (YDATA[I]-YDATA[J])^2);
          FINISHED := 'FALSE';
          'IF' DIST < DMIN
            'THEN' K := J; DMIN := DIST
          'FI'
        'FI';
      'IF' 'NOT' FINISHED
      'THEN'
        'IF' I <= NDATA - 4 'THEN' IBEEN[K] := 'TRUE' 'FI';
        XYZ [1,1] := XDATA [I]; XYZ [2,1] := YDATA [I]; XYZ [3,1] := ZDATA [I];
        XYZ [1,2] := XDATA [K]; XYZ [2,2] := YDATA [K]; XYZ [3,2] := ZDATA [K];
        'IF' DIVIDE (XPT, XAV, XYZ, X, GAMMA, DELTA, ANISO, CONTOUR, TOL, DLLIM,
                    MU, RO, ROLONG, IX, IY, NPOINTS, NOAV, NDIMS)
        'THEN'
          NREF 'PLUS' 1;
          XREF [NREF] := X [IX]; YREF [NREF] := X [IY]
        'FI'
      'FI'
    'FI'
  'END'
'END';

```

```
PROC' OUTPOINT = ('REF'[]'REAL' XREF,YREF,STORE,'REF'[]'BOOL' DELETED,
  'REAL' AX,AY,DTOL,'REF''INT' NSTORE,'INT' NREF) 'VOID':
```

```
'C'
  PROCEDURE TO OUTPUT CONTOUR PT.
'C'
EGIN'
'REAL' DIST;
STORE[2*NSTORE + 1] := AX; STORE[2*NSTORE + 2] := AY;
NSTORE 'PLUS' 1;
'IF' NSTORE = 4 'OR' 'ABS'(AX+999.0) < 0.001
'THEN'
  OUTF(PLOTFILE,$L8(1X<3.4>)$,STORE);
  'CLEAR' STORE;
  NSTORE := 0
'FI';
'FOR' I 'TO' NREF 'DO''BEGIN'
  'IF' 'NOT' DELETED[I]
  'THEN'

    DIST := SQRT((AX-XREF[I])^2 + (AY-YREF[I])^2);
    DELETED[I] := DIST < DTOL
  'FI'
'END'
END';
```

```
PROC' ANGLEDIFF = ('REAL' ANGLE1,ANGLE2) 'REAL':
'C' DIFFERENCE BETWEEN 2 ANGLES. 'C'
BEGIN'
MIN(('ABS'(ANGLE1-ANGLE2),'ABS'(ANGLE1-ANGLE2+2*PI),
  'ABS'(ANGLE1-ANGLE2-2*PI)))
END';
```

```
PROC' SMOOTH = ('REF''REAL' ANGLE,'REAL' X1,Y1,X2,Y2,ANGTOL) 'BOOL':
'C' PROCEDURE TO CHECK THAT CONTOUR IS SMOOTH. 'C'
BEGIN'
'REAL' ANGNUM,DANGLE,WORKER;
WORKER := X1 - X2;
'IF' 'ABS'WORKER < 1.0E-9 'THEN' WORKER := 1.0E-9 'FI';
ANGNUM := ARCTAN((Y1-Y2)/WORKER);
'IF' X1 < X2 'THEN' ANGNUM 'PLUS' PI 'FI';
DANGLE := ANGLEDIFF(ANGLE,ANGNUM);
'IF' DANGLE < ANGTOL
'THEN' ANGLE := ANGNUM; 'TRUE'
'ELSE' 'FALSE'
'FI'
END';
```

```

'PROC' STEPALONG = ('REF'[,]'REAL' XPT,XAV,'REF'[]'REAL' X,GAMMA,
                  DELTA,ANISO,'REF''REAL' ANGLE,'REAL' CONTOUR,
                  ANGTOL,TOL,DLIM,STEPMAX,MU,RO,ROLONG,'BOOL'
                  NEWSTART,'INT' IX,IY,NPOINTS,NOAV,NDIMS) 'VOID':
  'C'
    PROCEDURE TO FIND NEXT PT ON CONTOUR.
  'C'
  'BEGIN'
    'REAL' ANGL1,ANG2,WORKER,XSAVE,YSAVE,STEP,VALUE,THETA,ORTHOG,MODG;
    'INT' NITLIM := 2;
    XSAVE := X[IX]; YSAVE := X[IY];
    [1:NDIMS]'REAL' DFDX;
    VALUE := INTERPOLATOR(XPT,XAV,X,GAMMA,DELTA,ANISO,DFDX,MODG,MU,
                          RO,ROLONG,NPOINTS,NOAV,NDIMS);
    STEP := STEPMAX;
    'IF' 'ABS' DFDX[IY] < 1.0E-9 'THEN' DFDX[IY] := 1.0E-9 'FI';
    ANGL1 := ARCTAN(-DFDX[IX]/DFDX[IY]);
    ANGL2 := ANGL1 + PI;
    'IF' ANGLEDIFF(ANGL1,ANGLE) < ANGLEDIFF(ANGL2,ANGLE)
    'THEN' THETA := ANGL1
    'ELSE' THETA := ANGL2
    'FI';
    'IF' NEWSTART 'THEN' ANGLE := THETA 'FI';
    ORTHOG := THETA + 0.5*PI;
    X[IX] := XSAVE + STEP*COS(THETA); X[IY] := YSAVE + STEP*SIN(THETA);
    'INT' NITS := 0;
    'WHILE' ('NOT' FIND(XPT,XAV,X,GAMMA,DELTA,ANISO,ORTHOG,CONTOUR,TOL,
                       DLLIM,MU,RO,ROLONG,IX,IY,NPOINTS,NOAV,NDIMS)
            'OR' 'NOT' SMOOTH(ANGLE,X[IX],X[IY],XSAVE,YSAVE,ANGTOL))
            'AND' NITS < NITLIM 'DO' 'BEGIN'

      NITS 'PLUS' 1;
      STEP 'DIV' 2;
      X[IX] := XSAVE + STEP*COS(THETA); X[IY] := YSAVE + STEP*SIN(THETA)
    'END';
    'IF' NITS >= NITLIM 'THEN' ANGLE := THETA 'FI'
  'END';

'PROC' OUTOFAREA = ('REF''REAL' AX,AY,'REAL' BORD,XLAST,YLAST,
                  'INT' SIGN) 'BOOL':
  'C' PROCEDURE TO CHECK IF CONTOUR HAS LEFT SPECIFIED AREA. 'C'
  'BEGIN'
    'IF' (AX-BORD)*SIGN > 0
    'THEN'
      AY := YLAST + (AY-YLAST)*(BORD-XLAST)/(AX-XLAST);
      AX := BORD;
      'TRUE'
    'ELSE' 'FALSE'
    'FI'
  'END';

```

```

'PROC' BACKTOSTART = ('REAL' AX,AY,XSTART,YSTART,STEPMAX,
                    SAVEDANGLE) 'BOOL':
  'C' PROCEDURE TO CHECK IF CONTOUR HAS RETURNED TO START. 'C'
'BEGIN'
  'REAL' DIST,ENDANGLE;
  DIST := SQRT((AX-XSTART)^2 + (AY-YSTART)^2);
  'IF' DIST < 1.5*STEPMAX
  'THEN'
    ENDANGLE := XSTART - AX;
    'IF' 'ABS'ENDANGLE < 1.0E-9 'THEN' ENDANGLE := 1.0E-9 'FI';
    ENDANGLE := ARCTAN((YSTART-AY)/ENDANGLE);
    'IF' XSTART < AX 'THEN' ENDANGLE 'PLUS' PI 'FI';
    ANGLEDIFF(ENDANGLE,SAVEDANGLE) < 0.5*PI
  'ELSE' 'FALSE'
  'FI'
'END';

```

```

'PROC' CONTOURTRACE = ('REF'[,]'REAL' XPT,XAV,'REF'[']'REAL' X,GAMMA,
                    DELTA,ANISO,XDATA,YDATA,ZDATA,XLOWER,XUPPER,
                    'REAL' CONTOUR,ANGTOL,TOL,DLIM,STEPMAX,MU,
                    RO,ROLONG,MEDIAN,'INT' IX,IY,NPOINTS,NOAV,
                    NDATA,NDIMS) 'VOID':
  'C'
  PROCEDURE TO TRACE CONTOUR LINES OF GIVEN LEVEL.
  'C'
'BEGIN'
  'REAL' ANGLE,SAVEDANGLE,DTOL,XSTART,YSTART,XLAST,YLAST;
  'REAL' STEP;
  'REAL' DUMANG;
  [1:8]'REAL' STORE; 'CLEAR' STORE;
  'INT' NSTORE,ICON,NREF,IREF;
  [1:NDATA]'REAL' XREF,YREF;
  DTOL := STEPMAX;
  SELECT (XPT,XAV,X,GAMMA,DELTA,ANISO,XREF,YREF,XDATA,YDATA,ZDATA,
          CONTOUR,TOL,DLIM,MU,RO,ROLONG,MEDIAN,NREF,IX,IY,NDATA,
          NPOINTS,NOAV,NDIMS);
  [1:NREF]'BCOL' DELETED; 'CLEAR' DELETED;
  NSTORE := 0; ICON := 0;

```

```

'BOOL' BORDERHIT, FINISHED, NEWSTART, JUSTHIT, ALLGONE;
OUTF (PLOTFILE, $L"VALUE"3X<4.4>$, CONTOUR);
OUTF (STANDOUT, $LL"CONTOUR LEVEL"2X<4.4>L$, CONTOUR);
OUTF (STANDOUT, $L"NO. OF REFERENCE PTS ="<5>L$, NREF);
ALLGONE := NREF = 0; IREF := 1;
'WHILE' 'NOT' ALLGONE 'DO' 'BEGIN'
  ICON 'PLUS' 1;
  OUTF (PLOTFILE, $L"LINE"4X<4>$, ICON);
  BORDERHIT := FINISHED := 'FALSE';
  X[IX] := XSTART := XLAST := XREF[IREF];
  X[IY] := YSTART := YLAST := YREF[IREF];
  OUTF (STANDOUT, $L"SEGMENT"<4>, 2X"STARTS AT"2 (2X<3.4>)$,
        (ICON, XSTART, YSTART));
  ANGLE := 0.0; NEWSTART := 'TRUE';
  OUTPOINT (XREF, YREF, STORE, DELETED, X[IX], X[IY], DTOL, NSTORE, NREF);
  STEP := STEPMA/5;
  'WHILE' 'NOT' FINISHED 'DO' 'BEGIN'
    STEPALONG (XPT, XAV, X, GAMMA, DELTA, ANISO, ANGLE, CONTOUR, ANGTOL,
              TOL, DLLIM, STEP, MU, RO, ROLONG, NEWSTART, IX, IY, NPOINTS,
              NOAV, NDIMS);
    STEP := STEPMA;
    'IF' NEWSTART
      'THEN' SAVEDANGLE := ANGLE; NEWSTART := 'FALSE' 'FI';
    JUSTHIT := OUTOFAREA (X[IX], X[IY], XLOWER[IX], XLAST, YLAST, -1)
      'OR' OUTOFAREA (X[IX], X[IY], XUPPER[IX], XLAST, YLAST, 1)
      'OR' OUTOFAREA (X[IY], X[IX], XLOWER[IY], YLAST, XLAST, -1)
      'OR' OUTOFAREA (X[IY], X[IX], XUPPER[IY], YLAST, XLAST, 1);
    XLAST := X[IX]; YLAST := X[IY];
    OUTPOINT (XREF, YREF, STORE, DELETED, X[IX], X[IY], DTOL, NSTORE, NREF);
    'IF' JUSTHIT
      'THEN' FINISHED := BORDERHIT; BORDERHIT := 'TRUE' 'FI';
    'IF' 'NOT' FINISHED 'AND' JUSTHIT
      'THEN'
        OUTPOINT (XREF, YREF, STORE, DELETED, -999, 0, DTOL, NSTORE, 0);
        OUTF (PLOTFILE, $L"JOIN"4X<4>$, ICON);
        PRINT (" BORDER REACHED");
        OUTPOINT (XREF, YREF, STORE, DELETED, XSTART, YSTART, DTOL, NSTORE, 0);
        DUMANG := ANGLE - PI;
        'IF' DUMANG < -PI/2 'THEN' DUMANG 'PLUS' 2*PI 'FI';
        ANGLE := SAVEDANGLE - PI;
        'IF' ANGLE < -PI/2 'THEN' ANGLE 'PLUS' 2*PI 'FI';
        SAVEDANGLE := DUMANG;
        XLAST := XSTART; YLAST := YSTART;
        XSTART := X[IX]; YSTART := X[IY];
        X[IX] := XLAST; X[IY] := YLAST
      'FI';
    'IF' 'NOT' JUSTHIT 'AND' BACKTOSTART (X[IX], X[IY], XSTART,
                                         YSTART, STEPMA, SAVEDANGLE)
      'THEN'
        OUTPOINT (XREF, YREF, STORE, DELETED, XSTART, YSTART, DTOL, NSTORE, 0);
        FINISHED := 'TRUE';
        PRINT (" RETURNED TO START")
      'FI'
    'END';
  OUTF (XREF, YREF, STORE, DELETED, -999, 0, DTOL, NSTORE, 0);
  ALLGONE := 'TRUE';
  'FOR' I 'TO' NREF 'DO'
    'IF' 'NOT' DELETED [I]
      'THEN'
        IREF := I; ALLGONE := 'FALSE'
      'FI'
  'END'
'END';

```

APPENDIX B

RESULTS OF SIMULATION EXPERIMENTS



## INTRODUCTION

Simulation experiments were performed to validate certain of the results and the techniques developed. The areas of investigation of these experiments were as follows:

- a) Estimation of  $\rho$  - comparison of "pair-point" and "maximum likelihood" techniques.
- b) Estimation of parameters for "two-stage" model
- c) Estimation of anisotropy - comparison of techniques
- d) Areas of closed contours of a correlated stationary process
- e) Uncertainty in oil province reserves due to different realisations of  $\epsilon$ .

Although simulation experiments should be treated with a certain degree of caution, and not used indiscriminantly, they can provide a valuable check on the validity or otherwise of techniques which have been developed, in particular in the field of estimation.

### a) Estimation of $\rho$

These simulation experiments were carried out in a two-dimensional region of area  $10 \times 10$  units, with values of the true correlation distance ( $\rho$ ) of 1, 2 and 4 units, and using  $N$  randomly positioned data points, where  $N$  took values 10, 20 and 40.

For each combination of  $N$  and  $\rho$  values, the following procedure was carried out:

1. Generate  $N$  data points, with data values given by the true  $\rho$ ,  $\mu$  and  $\sigma$  values.
2. Estimate  $\hat{\mu}$  by the median of the data values
3. Estimate  $\hat{\rho}$  using the "pair-point" method

4. Estimate  $\hat{\rho}$  using the "maximum-likelihood" method
5. Repeat the whole process 10 times from step 1.
6. Compute the average and standard deviation of the  $\hat{\rho}$  estimates by both methods.

(For these experiments, the true values of  $\mu$  and  $\sigma$  were fixed as 10 and 2 respectively).

The details of the generation of the N data point values are as follows:

- 1.1  $i:=1$
- 1.2 For the  $i^{\text{th}}$  data point, compute its position using a uniform random distribution within the region of interest.
- 1.3 Estimate its mean value  $\hat{z}_i$  and residual variance  $\sigma_i^2$ , given the preceding  $i-1$  data points (see equ. (2.48)).
- 1.4 Set  $z_i = \hat{z}_i + \sigma_i Z$ , where Z is a Standard Normal random variate.
- 1.5  $i:=i+1$
- 1.6 Repeat from step 1.2 until  $i>N$ .

As well as computing the average  $\hat{\rho}$  value (for both techniques) from the 10 iterations, it was possible to estimate the mean square error by:-

$$\text{MSE} = (\hat{\rho} - \rho)^2 + S_{\rho}^2 \quad (\text{B.1})$$

where  $\hat{\rho}$  = average  $\rho$  estimate

$S_{\rho}^2$  = computed variance of  $\rho$  estimates.

The results of these 9 experiments are tabulated in Table B.1, and are shown in graphical form in Figure 2.5.

#### b) Estimation of "two-stage" model parameters

The estimation of the parameters of the "two-stage" model is a more complex process, and only a very limited set of experiments was carried out, to verify that at least reasonably sensible results were being obtained.

A two-dimensional  $10 \times 10$  region with  $\mu=10$  was used once more, with the following parameters for the long-range and short-range processes:

$$\text{Long-range: } \rho_L = 2.5; \sigma_L = 3$$

$$\text{Short-range: } \rho_s = 1 ; \sigma_s = 1$$

The procedure for each experiment was as follows:

1. Generate random coordinates for  $N$  data points, with data values  $(z_i)$  given by  $\mu$ ,  $\rho_L$  and  $\sigma_L$ .
2. Using the same coordinates, generate short-range residuals  $(z_i^e)$  given by  $\rho_s$  and  $\sigma_s$ , with mean 0.
3. For each point  $i$ ,  $z_i := z_i + z_i^e$ .
4. Estimate  $\hat{\mu}$  by the median of these data values.
5. Position  $n_A$  "average points" evenly in the region of interest, and compute weighted average values  $(z_j^A)$  at these points.
6. Estimate  $\hat{\rho}_L$  from these average points by the "maximum likelihood" method.
7. Estimate  $\hat{z}_i^e$  at each data point by subtracting the fitted trend process using  $\hat{\mu}$  and  $\hat{\rho}_L$ .
8. Estimate  $\hat{\rho}_s$  using the "pair-point" method
9. Estimate  $\hat{\rho}_s$  using the "maximum-likelihood" method.
10. Repeat the whole process 10 times from step 1.
11. Compute the average and standard deviation of:  $\hat{\rho}_L, \hat{\rho}_s$  by both methods,  $\hat{\rho}_L / \hat{\rho}_s$  (maximum likelihood).

The experiment was repeated twice, once with  $N=20$  and  $n_A=10$ , and the second time with  $N=40$  and  $n_A=20$ . The results are shown in Table B.2. It is noticeable that the correlation distances are consistently under-estimated, but that the ratio of  $\rho_L$  to  $\rho_s$  is reproduced to a reasonable extent, especially with more data points.

c) Estimation of Anisotropy

Two methods have been developed for estimating anisotropy factors (see Section 5.4). In addition, it will be useful to compare the results obtained when points are randomly scattered with those obtained with regularly positioned data points. Other variables of interest are the number of data points (N) and the ratio of the anisotropy factors ( $\alpha_1$  and  $\alpha_2$ ).

As before, the basic experiment involved a 2-dimensional 10×10 area. The values of the other parameters were fixed:  $\mu=10$ ,  $\sigma=2$ ,  $\rho=2$ . 16 experiments were carried out, varying the four factors as follows:

1. Method I/Method II
2. Random points/regular points
3.  $N=20/N=40$
4.  $\alpha_2/\alpha_1=4.0/\alpha_2/\alpha_1=25.0$ .

Each experiment involved generating N data points in the region of interest using the given parameters and anisotropy factors, and then using the appropriate method to estimate the anisotropy factors. The ratio  $\hat{\alpha}_2/\hat{\alpha}_1$ , was computed, and this procedure repeated 10 times. The results of each such experiment were quoted as:

1. The geometric mean of  $\hat{\alpha}_2/\hat{\alpha}_1$
2.  $K = e^s$

where s is the estimated standard deviation of  $\log(\hat{\alpha}_2/\hat{\alpha}_1)$ .

(This seemed the most appropriate way to quote the results, as we are interested in the ratios of anisotropies in the two directions).

Table B.3 gives these results for each experiment. The anisotropy ratio is generally underestimated quite significantly, although Method II seems to do a better job than Method I.

#### d) Areas of closed contours

In Chapter 6, a formula is derived (6.10) for the approximate average area of an oilfield, considered as a closed contour at a certain level  $u$ . If we set  $\epsilon = u/\sigma$ , then equation (6.17) gives the average area in terms of  $\sigma, \sigma_s$  and  $\epsilon$ . It was felt necessary to validate these formulae, especially in the case where  $\epsilon$  is small and the approximation may not be accurate. Also of interest is the shape of the distribution of these areas, which should be approximately negative exponential, at least for large  $\epsilon$ .

In order to investigate these distributions, random realisations of such contours were generated, and their areas computed. A triangular grid was used to track the positions of the points on the contour, and realisations of the correlated stochastic process were generated at the required nodes of the grid, using the method described in section a) of this appendix. Figure B.1 illustrates this procedure for a simple example. The closed contour  $Z(\underline{x})=u$  is thus approximated by a set of straight line segments, each produced by linear interpolation across a triangle.

This procedure was carried out using the following parameters:  $\rho=2$  ( $=1/\sigma_s$ ),  $\sigma=2$ , and for a range of values of  $\epsilon$  ( $=u/\sigma$ ) from 0.5 to 2.5. Between 25 and 32 realisations were generated for each value of  $\epsilon$ , and the results are tabulated in Table B.4. The average area ( $\bar{A}$ ) and standard deviation ( $\hat{\sigma}_A$ ) were estimated from the results and an approximate 95% confidence interval for the true mean area was computed, assuming approximate normality for the distribution of  $\bar{A}$ .

From Figure B.3 it can be seen that  $E[A]$ , computed from equation (6.17), falls inside this confidence interval in every case, and that the agreement is particularly close between  $E[A]$  and  $\bar{A}$  for large  $\epsilon$ . Figure B.3 shows

histograms of the distribution of areas for the various values of  $\epsilon$  - the negative exponential assumption is seen to be quite reasonable, even for small  $\epsilon$ . Figures B.4 and B.5 show some example realisations of closed contours for  $\epsilon=1.0$  and  $\epsilon=2.0$  respectively. These bear out the assumption that for large  $\epsilon$  the contours will tend to take elliptical form (see Adler, 1981, p.136ff), although this is obviously not the case for smaller values of  $\epsilon$ . The results of these simulation experiments seem to confirm that the assumptions made in Chapter 6 are not unreasonable.

c) Uncertainty in oil reserves due to variation in  $\underline{\epsilon}$

In Chapter 6 it was shown that the uncertainty in oil reserves in a given area, according to our model, can be considered to be due to at least two sources. One is the variance in the reserves given a certain set of  $\underline{\epsilon}$  values in the blocks under consideration, and the other is the variance due to variations in the  $\underline{\epsilon}$  values themselves. To evaluate the latter, random realisations of  $\underline{\epsilon}$  for the blocks were generated, and for each such set of  $\underline{\epsilon}$  values the mean and variance of the reserves were computed.

Given the known  $\underline{\epsilon}$  value at the locations of the existing fields, it is simple to estimate the mean values  $\underline{\epsilon}$  at the block locations, as well as the residual covariance matrix  $S^*$  for the blocks. The simplest way of generating a random  $\underline{\epsilon}$  realisation consistent with this is to use the Cholesky decomposition method (see Ripley, 1981, p.17), and find a lower triangular matrix  $L$  such that

$$LL' = S^* .$$

Then, 
$$\underline{\epsilon} = \bar{\underline{\epsilon}} + L\underline{e} \tag{B.2}$$

where the  $\underline{e}$  values are independent Standard Normal random variables.

This Cholesky method is to be preferred to the method outlined earlier

for generating random realisations in this case, as the locations of the blocks are fixed for each iteration. Thus the Cholesky decomposition can be carried out once for all.

Three different values of the correlation parameter for the  $\underline{\epsilon}$  values ( $\rho_{\underline{\epsilon}}$ ) were used - 0.44, 0.623 and 0.8 block units. In each case the assumed values of  $\mu_{\underline{\epsilon}}$  and  $\rho_{\underline{\epsilon}}$  were 3.0 and 0.551 respectively. For the stochastic process used to model the oilfields, the values of  $\rho$  and  $\sigma$  were 0.0666 block units and  $12.433 \times 10^6$  STB. For each value of  $\rho_{\underline{\epsilon}}$ , the following calculations were carried out: using the mean  $\underline{\epsilon}$  values ( $\bar{\underline{\epsilon}}$ ) the mean and variance of the oil reserves were computed; then 10 iterations were performed, each time generating a random  $\underline{\epsilon}$  realisation and computing the mean and variance of the oil reserves. If  $\bar{R}$  is the mean reserves for one such iteration, and  $\sigma_R^2$  the corresponding variance, then for that iteration we may set  $S^2 = \bar{R}^2 + \sigma_R^2$ . We may estimate  $E[R^2]$  by averaging this  $S^2$  value over the 10 iterations, and similarly estimate  $E[R]$  by averaging  $\bar{R}$ . Hence the variance, including the uncertainty in  $\underline{\epsilon}$ , can be computed from  $E[R^2] - (E[R])^2$ .

Table B.5 gives the results of these experiments. As would be expected, increasing  $\rho_{\underline{\epsilon}}$  increases the mean and variance of the oil reserves. Allowing uncertainty in the  $\underline{\epsilon}$  values tends to increase the mean reserves, because a block with high  $\epsilon$  value will have little contribution to the expected reserves, and therefore can only give an increased contribution if  $\epsilon$  is allowed to vary.

TABLE B.1

Results of  $\rho$  Estimation Testing

		<u>N=10</u>			<u>N=20</u>			<u>N=40</u>		
		<u>Mean</u>	<u>S.d.</u>	<u>M.S.E.</u>	<u>Mean</u>	<u>S.d.</u>	<u>M.S.E.</u>	<u>Mean</u>	<u>S.d.</u>	<u>M.S.E.</u>
$\rho=1.0$	P.-P.	1.93	1.26	2.46	1.70	0.66	0.93	1.23	0.42	0.23
	M.L.	0.63	0.31	0.23	0.57	0.21	0.23	0.98	0.30	0.09
$\rho=2.0$	P.-P.	2.53	2.12	4.78	1.77	0.69	0.53	1.83	0.41	0.19
	M.L.	0.91	0.56	1.50	1.16	0.34	0.82	1.25	0.16	0.59
$\rho=4.0$	P.-P.	2.42	0.80	3.14	1.84	0.60	5.04	1.73	0.78	5.76
	M.L.	1.34	0.67	7.51	1.94	0.36	4.40	1.86	0.50	4.84

P.-P. -  $\hat{\rho}$  by pair-point method (10 iterations)

M.L. -  $\hat{\rho}$  by maximum likelihood method (10 iterations)



TABLE B.2Results of Parameter Estimation for "Two-Stage" Model

True values:  $\mu = 10$        $\rho_L = 2.5$        $\rho_S = 1.0$  .  
     $\sigma_L = 3.0$        $\sigma_S = 1.0$

N = 20, n<sub>A</sub> = 10

<u>Parameter</u>	<u>Mean</u>	<u>S.d.</u>	<u>M.S.E.</u>
$\hat{\rho}_L$ (ML)	1.838	0.575	0.769
$\hat{\rho}_S$ (ML)	0.523	0.182	0.260
$\hat{\rho}_S$ (Pair-point)	0.517	0.205	0.275

Median value of  $\hat{\rho}_L / \hat{\rho}_S$  (ML) = 3.861.

N = 40, n<sub>A</sub> = 20

<u>Parameter</u>	<u>Mean</u>	<u>S.d.</u>	<u>M.S.E.</u>
$\hat{\rho}_L$ (ML)	1.464	0.194	1.1108
$\hat{\rho}_S$ (ML)	0.524	0.267	0.299
$\hat{\rho}_S$ (Pair-point)	0.403	0.254	0.421

Median value of  $\hat{\rho}_L / \hat{\rho}_S$  (ML) = 2.942

TABLE B.3

Results of Anisotropy Estimation

		METHOD I		METHOD II	
		RANDOM PTS.	REGULAR PTS.	RANDOM PTS.	REGULAR PTS.
$\alpha_2/\alpha_1$ =4.0	N=20	0.986 (4.111)	0.651 (2.425)	5.204 (2.839)	1.361 (2.592)
	N=40	1.546 (2.411)	1.213 (1.732)	1.7696 (2.073)	2.162 (1.769)

		METHOD I		METHOD II	
		RANDOM PTS.	REGULAR PTS.	RANDOM PTS.	REGULAR PTS.
$\alpha_2/\alpha_1$ =25.0	N=20	0.923 (3.928)	0.924 (3.696)	5.862 (2.134)	2.444 (1.590)
	N=40	2.017 (1.865)	2.027 (1.633)	4.114 (2.081)	4.268 (1.550)

The top value in each cell is the geometric mean of  $\hat{\alpha}_2/\hat{\alpha}_1$ .

The value in brackets =  $e^s$ , where  $s$  is the estimated standard deviation of  $\log(\hat{\alpha}_2/\hat{\alpha}_1)$ .

TABLE B.4

Areas of Oilfield Realisations

$\epsilon$	$E[A]$	<u>No. of Realisations</u>	<u>Av. <math>A(\bar{A})</math></u>	$\hat{\sigma}_A$	<u>Approximate 95% C.I. for mean</u>
0.5	44.045	26	65.58	95.47	28.89-102.28
1.0	16.485	32	21.00	18.34	14.23-27.76
1.5	8.640	28	9.27	9.11	5.89-12.64
2.0	5.307	25	4.86	5.21	2.82-6.91
2.5	3.559	25	3.18	3.60	1.61-4.76

TABLE B.5

Variation in Oil Reserves

<u><math>\epsilon</math> fixed</u>	$\rho_{\epsilon}=0.44$	$\rho_{\epsilon}=0.623$	$\rho_{\epsilon}=0.8$
Mean ( $10^9$ STB)	24.63	38.86	47.79
Variance ( $10^{18}$ STB <sup>2</sup> )	505.5	721.1	867.1
Standard deviation ( $10^9$ STB)	22.48	26.85	29.45
<u>Averaging <math>\epsilon</math> realisations</u>			
$E[R^2]$ ( $10^{18}$ STB <sup>2</sup> )	1608.3	3313.45	4095.1
$E[R]$ ( $10^9$ STB)	31.06	48.95	55.535
$\text{Var}[R]$ ( $10^{18}$ STB <sup>2</sup> )	643.6	917.6	1010.9
Standard deviation ( $10^9$ STB)	25.37	30.29	31.80

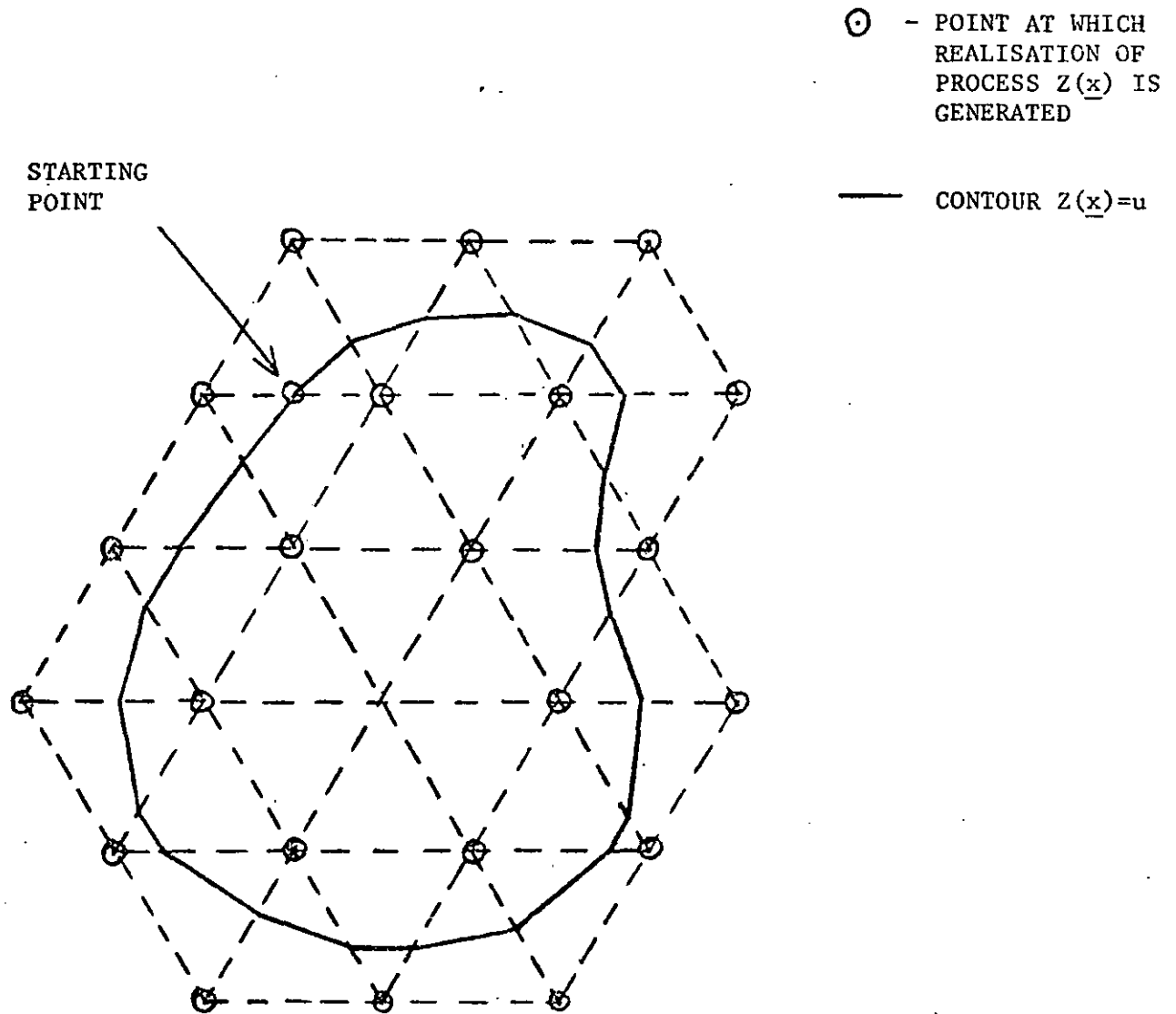


FIGURE B.1: Example of generation of random closed contour

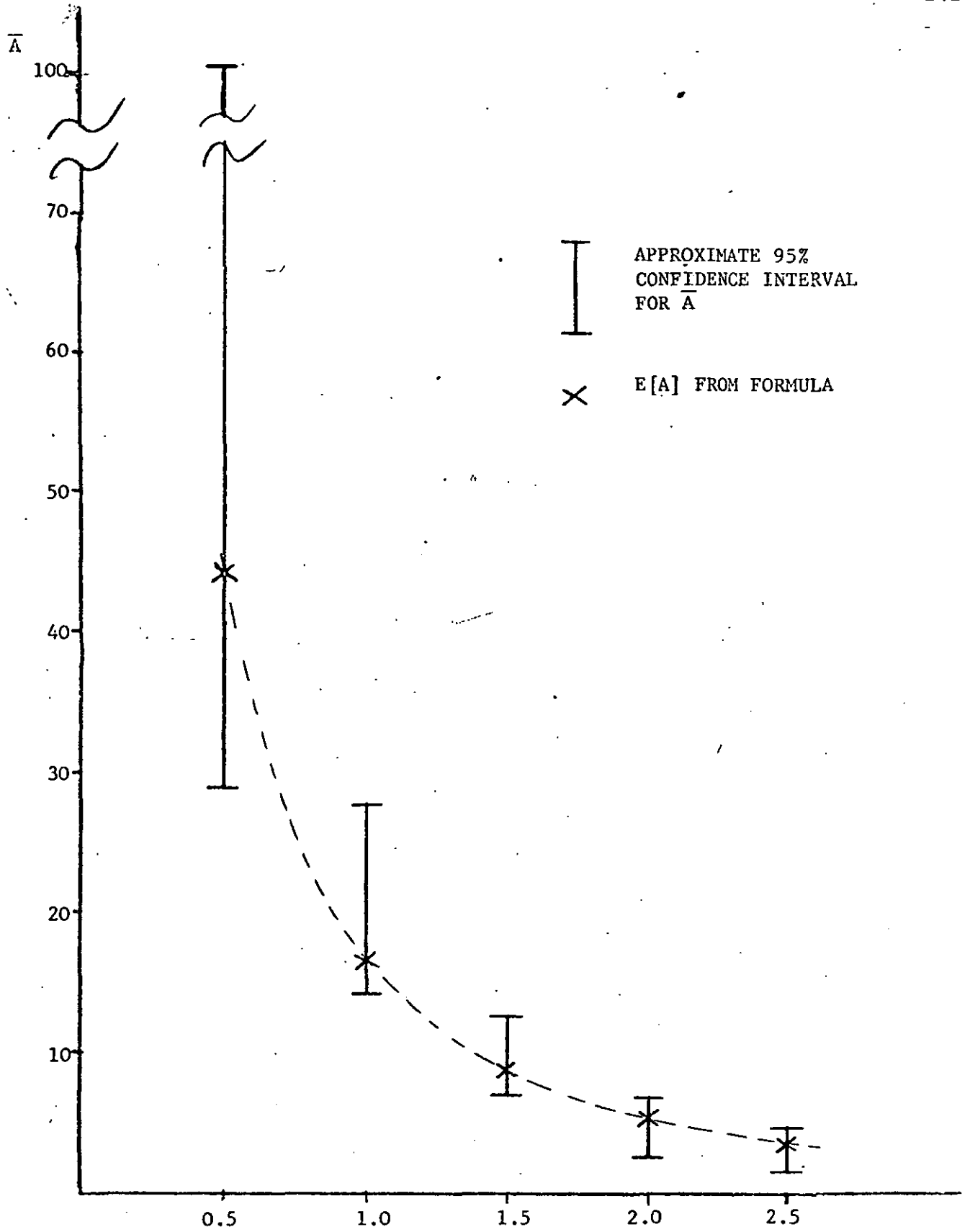
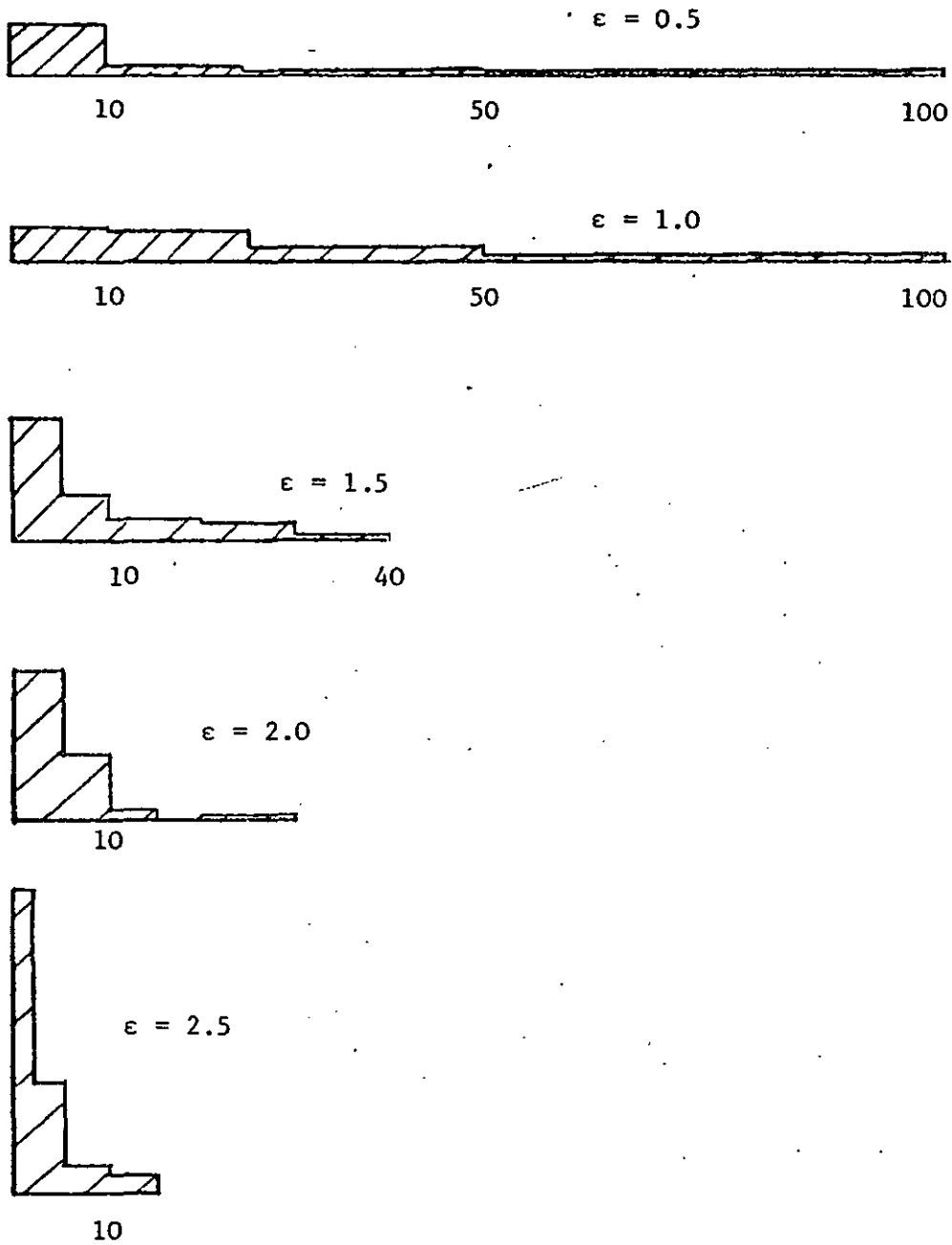


FIGURE B.2: Results of random realisations of closed contours



**FIGURE B.3:** Area histograms for different values of  $\epsilon$

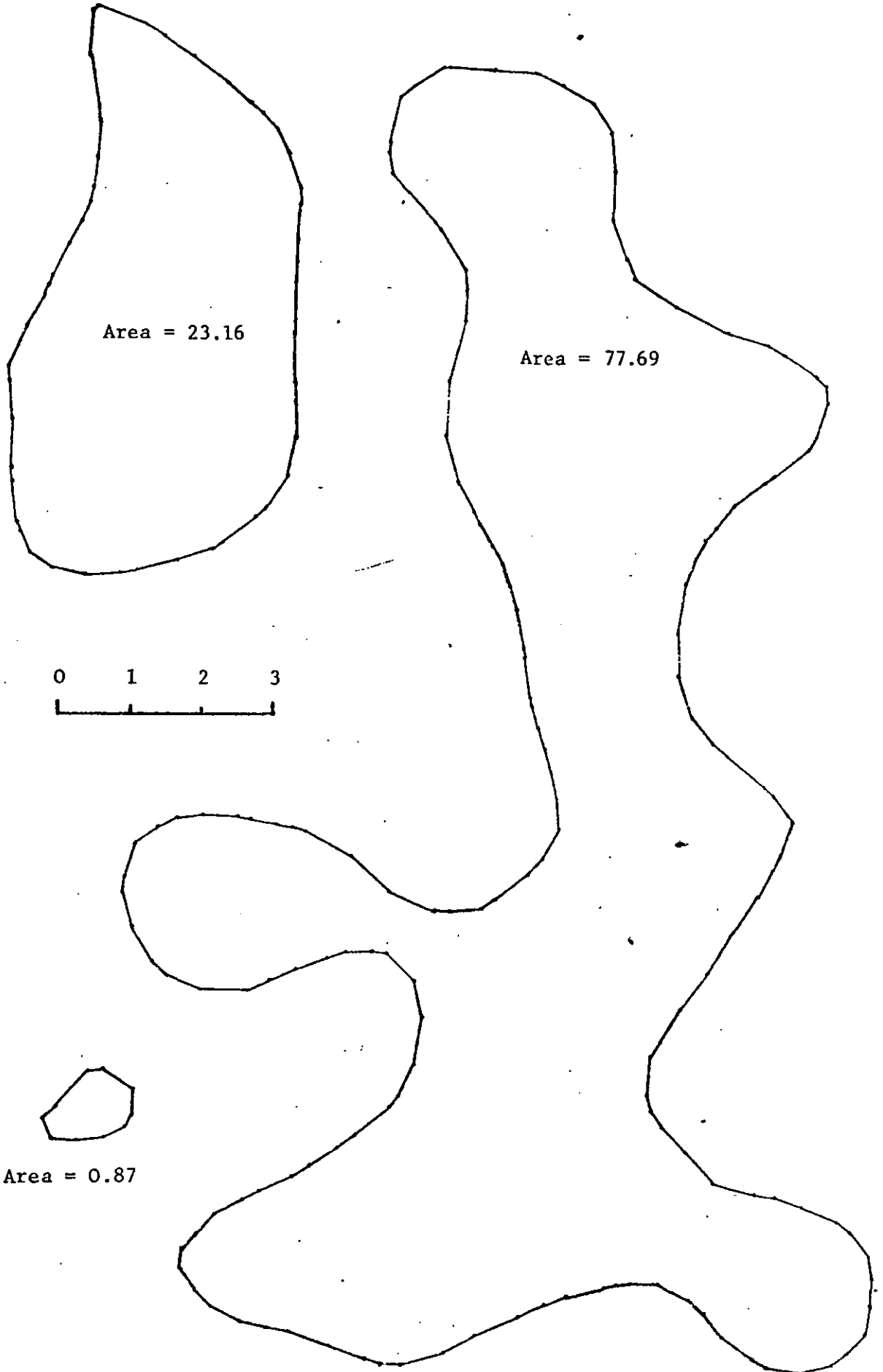


FIGURE B.4: Examples of contours for  $\epsilon = 1.0$



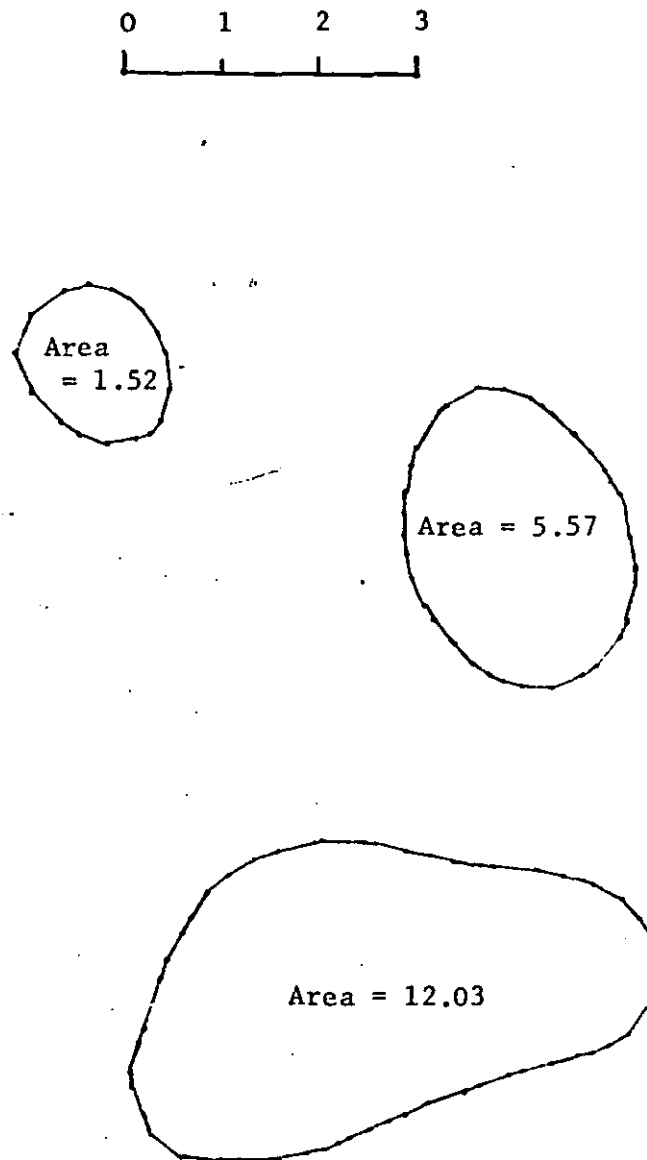


FIGURE B.5: Examples of contours for  $\epsilon = 2.0$

