CURVE FITTING BY EXPONENTIALS

AND RELATED FUNCTIONS

by

SAAD SALEM BEN HAMEID

A thesis submitted for the

degree of Doctor of Philosophy

in the University of Glasgow

ProQuest Number: 10662664

ProQuest 10662664

ACKNOWLEDGEMENTS

My most sincere thanks are due to Professor D.C. Gilles, who supervised this work, for his guidance, encouragement and interest.

I am grateful to the Libyan Ministry of Education and the University of Libya for their financial support during the period of this research, and also to all staff members of the Computing Department, University of Glasgow for their kind co-operation.

Finally, I would like to thank Dr. H.I. Glass and his colleagues of the Department of Medical Physics, Hammersmith Hospital, London, for their assistance and providing the experimental data which has been investigated in this thesis.

S.S. Ben Hameid

July, 1969.

C O N T E N T S

CHAPTER 1

INTRODUCTION

1.1. Outline of the Problem

There are many problems in science in which experimental data may for theoretical reasons best be represented by linear combinations of exponential of the form

$$y(t_i) \sim f(t_i) = \sum_{r=1}^{n} c_r \exp(-\alpha_r t_i)$$

$$(1.1.1)$$

in which y describes a decay-type process and t represents measurement made at equally or unequally spaced time intervals.

These problems are classified into three cases (Willers, 1948):

(i)   whenever decay-type processes are involved, such as radioactive processes, e.g. the absorption of gamma rays by lead, quoted as an example in Guest (1961).   The problems also arise where radioactive material is used to trace the behaviour of a biological system.

(ii) in decay-type process with damped oscillations, such as charging processes.

(iii) with phenomena, which arise by separations of purely periodic processes whose periods do not have integral ratios, such as brightness fluctuations of variable stars, or oscillation processes.

The features of these cases are, in the first case the exponents α_r are all real, while in the second and the third case, they may be complex or purely imaginary.

In many cases we do not know the zero line to which the process decays, so there will be a term $t = 0$, which belongs to the value $\alpha = 0$, so we add an extra term (constant) to (1.1.1), which becomes

$$y_i \sim f_i = c_0 + \sum_{r=1}^{n} c_r \exp(-\alpha_r t_i), \text{ for}$$

$i = 1(1)N$ and $N \geqslant 2n + 1$

It is desired to fit the set of equations (1.1.1) to N observations of y at times t_1, $t_2 \ldots t_N$ where $N \geqslant 2n$. In determining the parameters c_r and α_r, where these parameters or their combinations have

physical or biological significance, it is not sufficient that the set of equations (1.1.1) merely approximate closely to the data, but it is necessary that the parameters be accurately estimated and that estimates of the errors of the parameter be obtained.

Our interest is the investigation of the decay-type problem. This problem is demanding much attention in the medical field especially in the use of tracers in biological systems (Berman and Schoenfeld, 1956; Glass and Nordin, 1963; Glass and De Garreta, 1967; Robertson, 1962). An experiment usually involves injecting radioactive tracers in biological system, and then determining the amount of radioactive substance (y_i) as a function of time (t_i). From the collected data one tries to draw conclusions about the system, that is, the number of compartments (exponents) involved, how they are interconnected and how accurately they can be estimated.

A two compartment system which deals with the use of labelled uric acid to estimate the compartment size and turnover of uric acid in man

(Glass et al., 1968) has been investigated.   The
details of this study will be given later.

The major difficulties in exponential curve
fitting problems are that we are dealing with a series
of non-linear equations and that the data are only
approximating the function $y(t)$ over a finite range
in t.   Lanczos (1957), has pointed out that there
are a number of simple and straightforward mathematical
solutions to the exponential fitting problem, but
unfortunately enormous practical problems arise when
they are applied to experimental data from physical
or biological sources.   The main reason for this is
the exceedingly non-orthogonal behaviour of the
exponential functions.   In most cases the accuracy
required is far beyond that usually available.   The
high correlation between the exponential functions
makes the matrix involved in estimating the required
parameter ill-conditioned (see Chapter 5).

In practice it is found that in most biological
systems the non-orthogonality of the exponential functions
is evident, i.e. that the data is represented by the
equation (1.1.1) in which the α's do not differ
from one another by order of magnitude.   If this is

the case that the α's do differ from one another,
then the exponents are reasonably orthogonal, and the
solution can be determined to a reasonable degree of
accuracy.   Indeed the "peeling off technique" is
quite satisfactory in this case (see Section 1.3) and
can be applied.

1.2.   The Two Compartment System

Let V_i be the amount of
substance in compartment i and
$\lambda_{ij} V_j \delta t$ be the amount of
substance going from compartment
i to compartment j in time δt,
further let v_i be the amount
of labelled substance in
compartment i.
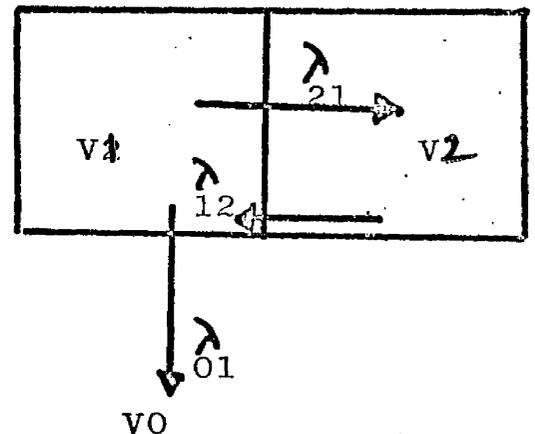
Then if the fraction of substance going from
compartment i to j is the same for labelled or
unlabelled substance, $\lambda_{ij} v_j \delta t$ is the amount of
labelled substance going from compartment i to j
in time δt.

Hence considering labelled substance, we have

$$v_1 + \lambda_{12}v_2 \delta t - \lambda_{21}v_1 \delta t - \lambda_{01}v_1 \ t = v_1 + \delta v_1,$$

$$v_2 + \lambda_{21}v_1 \delta t - \lambda_{12}v_2 \delta t = v_2 + \delta v_2$$

and $\quad v_0 + \lambda_{01}v_1 \delta t \ = v_0 + \delta v_0,$

from which we obtain

$$\frac{dv_1}{dt} = \lambda_{12}v_2 - (\lambda_{01} + \lambda_{21})v_1,$$

$$\frac{dv_2}{dt} = \lambda_{21}v_1 - \lambda_{12}v_2,$$

$$\frac{dv_0}{dt} = \lambda_{01}v_1$$

These equations form a set of first order differential equations and their solution may therefore be written as

$$v_i(t) = A_{i1} \exp (- \alpha_1 t) + A_{i2} \exp (- \alpha_2 t),$$

$$(1.2.1)$$

where

$$A_{21} = \frac{(\lambda_{01} + \lambda_{21} - \alpha_1)A_{11}}{\lambda_{12}}$$

$$(1.2.2)$$

and

$$A_{22} = \frac{(\lambda_{01} + \lambda_{21} - \alpha_1)A_{12}}{\lambda_{12}}$$

$$(1.2.3)$$

Usually the experimental data are measures of the concentration of unlabelled substance $v_0(t)$ as a function of time. Then it is required to fit a sum of exponential to the data, and so obtain values of the constants A_{01}, A_{02}, α_1 and α_2 and from these to determine the parameters of the model λ_{01}, λ_{12} and λ_{21}. A more detailed description and discussion of the model will be given in Chapter 7.

In this thesis we need to provide some fit if at all possible. Frequently an experimenter would know ratios of exponentials involved and would know

the accuracy of observations, in which case our objectives are:

(a)   to devise a satisfactory method to fit exponentials,

(b)   to determine accuracy of results corresponding to given accuracy of data,

(c)   to see any modifications, e.g. change of time interval could make fitting more accurate,

(d)   to see the significance of weighting factors in least-squares applications,

(e)   to determine the number of exponents from a particular set of data.

1.3. Previous Methods

A number of fitting procedures are possible, each resulting in different solution.   The main idea is to establish a practical procedure which will produce the best approximation for fitting the data of each kind of experiment taking into consideration the type of experiment and also the statistical errors of the fitted parameters.

We will now describe two existing methods.

(i) Graphical Peeling Off Method

This method (Perl, 1960) is the most common one and is used to resolve a decay curve into its components using a graphical procedure. Here the data are plotted on semi-log paper and the curve resolved by repeated subtraction of straight lines.

The method is certainly the easiest to perform, it may be considerably refined by using a least-squares procedure to fit straight lines, and some error estimation becomes available. The difficulty in this method is inherent in the subtraction procedure where some of the new generated y's become negative and their logs become imaginary, although when the decay ratios of the exponentials are widely separate (low correlation), so that the effective range of values of t determine only one component, this does not usually occur.

(ii) Prony's Method

Another mathematical approach to this problem has been suggested by Prony (Hildebrand, 1956). In this method the function is given in equidistant

values (v_i, t_i), where $t_i = t_0 + ih$, for $i = 0(1)N$.

If we write $x_r = \exp(-\alpha_r h)$, $r = 1(1)n$, then the set of equations (1.1.1) will become

$$f_i = a_1 x_1^i + a_2 x_2^i \ldots \ldots \ldots + a_n x_n^i,$$

$$(1.3.1)$$

where $\quad a_r = c_r \exp(-\alpha_r t_0)$.

If further, we let x_r be the roots of the equation

$$x^n + v_1 x_r^{n-1} \ldots \ldots \ldots \ldots \ldots + v_n = 0,$$

$$(1.3.2)$$

then the observations f_i are related by the difference equations

$$f_i v_n + f_{i+1} v_{n-1} + \ldots \ldots \ldots + f_{i+n+1} v_1 + f_{i+n} = 0$$

$$(1.3.3)$$

where $\quad i = 0(1)N - n$.

There are thus $(N - n + 1)$, such equations
which can be solved for the n quantities v_1, v_2....,
v_n, using the method of least-squares provided
$N \geqslant 2n - 1$.

The algebraic equation $(1.3.2)$ may then be
solved to determine the values of x_r and hence
α_r ($\alpha_r = - \ln x_r/h$), and finally the values of the
coefficients from $(1.3.1)$ by using least-squares
method.

The limitations of this method are that it
requires equal intervals of time and, more important,
that the errors in the observations will add errors
in the difference equations. The results from
Prony's method are usually erratic (Hildexbrand,
1956), the α's were often negative or complex
instead of positive values needed by the experiment.

1.4. Least-Squares Solution

Let us assume that we have N+1 pairs of
measured values (y_i, t_i), where $i = 0, 1, ..., N$.
We know it is rarely possible to fit these values,
so we expect to have a set of residuals or

differences

$$v_i = y_i - f(t_i) \neq 0$$

(1.4.1)

between the observed values and the calculated ones.

Some measure of the residuals and an estimate of the unknown parameters are required to fit the measured values. There are different approaches to this problem such as the least-squares, the maximum likelihood and the minimum variance methods. These methods are described in many text books including Guest (1961), Fraser (1951), Scheffe (1958).

If we consider that v's are elements of a vector, then as Froberg (1964) has mentioned, we want to find an approximation that makes these elements small, so the most natural measure is some characteristic of the vector norm. Then we get the principle that the approximation should be chosen so that the norm of this residual's vector is minimized. The norm we should use depends on the nature of the problem. If we use the Euclidean norm, we get the least-squares method

but if on the other hand we use the maximum norm,
we obtain the Chebysev approximation method.

For simplicity and because of statistical
considerations (see 2.3) we will use the least-squares
procedure in our investigation. The least-squares
principle which was proposed by Legendre (1805) and
subsequently by Gauss (1809) and Laplace (1811),
states that "of all possible solutions, the most
satisfactory is that which renders the sum squares
of the errors a minimum".

If we let the set of equations (1.1.1) take
the form

$$y_i \sim f_i = \sum_{r=1}^{n} c_r \phi_r(\alpha_r, t_i)$$

(1.4.2)

and assume that the functions $\phi_r(\alpha_r, t_i)$ are
completely known, $\overset{then}{}$ For N values of t, we have a
set of N linear equations. If N is equal to n,
then there is a unique solution (Lanczos, 1957)
providing the determinant of the system is not
equal to zero. There is one set of values c_1,
c_2, ..., c_n that will satisfy the above equations

If N is less than n then there are less equations than unknown parameters and in general there is an infinite number of solutions, each of which represents a curve passing through the given points. In this case the problem is incompletely determined and we can not find a single best fit, unless we have some extra information.

If N is greater than n, then in general there is no solution. In this case the least-squares makes it possible to have a best solution for the system. The problem in these circumstances is exactly the linear regression problem which has been studied in many text books, e.g. Guest (1961).

On the other hand if the functions $\phi_r(\alpha_r, t_i)$ in (1.4.2) are not known, where these functions are non-linear in the unknown parameters α_{rj} then the solution of these equations introduces a more difficult problem. This can be solved by changing the variables or using approximation methods based on Taylor's series expansion (Guest, 1961).

We may note that in the case of changing the variables, the transformed observations must be weighted even if the original observations were of

all equal weights (Chapter 4). We shall refer to
these weights and their estimation in Chapter 4.

The important features of the non-linear work
will be presented in Chapter 2 where two methods
based on some modifications of Newton-Raphson procedure
are discussed and applied to the exponential fitting.
We are able to provide an estimation of the standard
errors of the estimated parameters. The study of
special cases and uniqueness for some other cases
are included in Chapter 3.

The most important point in our objective stated
in Section 1.1, is the modification in spacing of the
observations, which is analysed in Chapter 5.

A description of the programs involved in the
calculations of the numerical work is given in
Chapter 6.

The practical application of this research
is given in Chapter 7.

CHAPTER 2

LEAST–SQUARES SOLUTION

2.1. **General Formulation for** $y_i \sim f_i = \sum_{r=0}^{n} c_r \phi_r(\alpha_r, t_i)$

The fitting of exponentials may be treated as a case of the general problem of fitting observations to a function

$$y_i \sim f_i = \sum_{r=0}^{n} c_r \phi_r(\alpha_r, t_i),$$

involving a linear combination of functions $\phi_r(\alpha_r, t_i)$ each containing a parameter α. Proceeding in the normal way of least-squares, we set

$$S = \sum_{i=1}^{N} w_i \left[y_i - \sum_{r=0}^{n} c_r \phi_r(\alpha_r, t_i) \right]^2$$

$$(2.1.1)$$

and to determine the unknown parameters c_r and α_r, we require to minimize S obtaining

$$\frac{\partial S}{\partial c_s} = -2 \sum_{i=1}^{N} w_i \left[y_i - \sum_{r=0}^{n} c_r \phi_r(\alpha_r, t_i) \right] \phi_s(\alpha_s, t_i) = 0$$

$$(2.1.2)$$

and

$$\frac{\partial S}{\partial \alpha_s} = -2 \ c_s \sum w_i \left[y_i - \sum c_r \phi_r(\alpha_r, t_i) \right] \phi_s'(\alpha_s, t_i) = 0$$

$$(2.1.3)$$

where $\phi_s'(\alpha_s, t_i) = \frac{\partial}{\partial \alpha_s} \phi_s(\alpha_s, t_i)$, for $s = 0(1)n$.

We have a set of non-linear equations to solve for the parameters $\alpha_0, \alpha_1, \ldots, \alpha_n$. The presence of the α's in non-linear form presents great difficulties in solving the problem. The equations must be solved iteratively. From the review of Powell (Walsh, 1966), there is a number of iterative techniques for solving the non-linear problems. The nature of the problem and what information we have got about it, will determine our choice of an iterative procedure which could be possible.

The solution of the non-linear problem is no longer necessarily unique and although in most practical cases, we have found the solution unique, we cannot prove this. Some comments on uniqueness are given in Chapter 3.

We are going to use the three methods in our solution; two of them are based on the use of generalized Newton-Raphson iterative technique, which is given in many text books including Modern Computing Methods (1962). The third method is developed by Berman et al. (1962).

(i)   Two Stage Minimization Method

In this method, we first minimize S (equation 2.1.1) with respect to the c's (equation 2.1.2). On substituting the values of the c's in S, we obtain S^* which is a function of the α's only. This may therefore be minimized with respect to the α_r, giving

$$\frac{\partial S^*}{\partial \alpha_r} = \frac{\partial S}{\partial \alpha_r} + \frac{\partial S}{\partial c_s} \times \frac{\partial c_s}{\partial \alpha_r} = \frac{\partial S}{\partial \alpha_r} \quad ,$$

since

$$\frac{\partial S}{\partial c_s} = 0, \text{ for all } s.$$

We therefore have a set of non-linear equations to be solved only for the α_r, namely

$$\sum_{r=0}^{n} c_r \sum_{i=1}^{N} w_i \, \phi_r'(\alpha_r, t_i) \, \phi_s(\alpha_s, t_i) = \sum_{i=1}^{N} w_i \, \phi_r'(\alpha_r, t_i) y_i$$

where the dash denoted differentiation with respect to α_r, and in which the c_r are functions of the α_r.

These equations may be solved iteratively by Newton-Raphson as will be shown in detail later.

(ii)   <u>One Stage Minimization Method</u>

In this method we estimate the coefficients in the same way as in the first method, then applying Newton-Raphson method to $2(n+1)$ equations, namely

$$\frac{\partial S}{\partial c_s} = -2\sum_{i=1}^{N} w_i \left[y_i - \sum_{r=0}^{n} c_r \, \phi_r(\alpha_r, t_i) \right] \phi_s(\alpha_s, t_i),$$

$$(2.1.4)$$

and

$$\frac{\partial S}{\partial \alpha_s} = -2 c_s \sum_{i=1}^{N} w_i \left[y_i - \sum_{r=0}^{n} c_r \phi_r(\alpha_r, t_i) \right] \phi_s'(\alpha_s, t_i),$$

$$(2.1.5)$$

for $s = 0(1)n$

giving

$$\frac{\partial S}{\partial c_s} + \sum_{r=0}^{n} \frac{\partial^2 S}{\partial c_s \partial c_r} \delta c_r + \sum_{r=0}^{n} \frac{\partial^2 S}{\partial c_s \partial \alpha_r} \delta \alpha_r = 0,$$

$$(2.1.6)$$

and

$$\frac{\partial S}{\partial \alpha_s} + \sum_{r=0}^{n} \frac{\partial^2 S}{\partial \alpha_s \partial c_r} \delta c_r + \sum_{r=0}^{n} \frac{\partial^2 S}{\partial \alpha_s \partial \alpha_r} \delta \alpha_r = 0,$$

$$(2.1.7)$$

for $s = 0(1)n$.

By solving these equations, we can determine the corrections δc's and $\delta \alpha$'s. Then adding the $\delta \alpha$'s to the initial values of the α's and iterating until the maximum correction is very small.

(iii) Berman's Method

In this method (Berman et al., 1962), the coefficients are estimated from the set of equation (2.1.1) by using the least-squares method as we did in the previous method, then determining the correction for the parameter by using a Taylor expansion and the least-squares method in the following way.

Considering $f_i = \sum\limits_{r=0}^{n} c_r \phi_r(\alpha_r, t_i)$, the solution corresponding to a least-square fit, f_i^0 may be related to the calculated function f_i by

$$f_i^{\,0} = f_i + \sum_{j=0}^{2(n+1)} \frac{\partial f_i}{\partial x_j} \delta x_j + \tfrac{1}{2} \sum_{\substack{j=0 \\ k=0}}^{2(n+1)} \frac{\partial^2 f_i}{\partial x_j \partial x_k}(\delta x_j \delta x_k)$$

$$+ \ldots\ldots$$

$$(2.1.8)$$

where

$$x_0 \text{ to } x_n = c_0 \text{ to } c_n \text{ and } x_{n+1} \text{ to } x_{2(n+1)} = \alpha_0 \text{ to } \alpha_n$$

Approximating (2.1.8) with only the first order term, we get

$$f_i{}^0 - f_i = \sum_{j=0}^{2(n+1)} \frac{\partial f_i}{\partial x_j} \, \delta x_j,$$

then applying the least-squares procedure to the new sum squares of residuals

$$R = \sum_{i=1}^{N} w_i \left[y_i - f_i - \sum_{j=0}^{2(n+1)} \frac{\partial f_i}{\partial x_j} \delta x_j \right]^2,$$

$$(2.1.9)$$

where y_i is the best available estimate for $f_i{}^0$.

A set of normal equations may be generated from equation (2.1.9) and a least-squares solution for the δx_j may be obtained.

2.2. Formulation for Exponentials

We will now give details of the methods as they formally apply to exponentials, expressing these in matrix terms which are appropriate for computation.

The sum of squares of the residuals which will now be minimized ~~are~~ is

$$S = \sum_{i=1}^{N} w_i \Big[y_i - \sum_{r=0}^{n} c_r \exp(-\alpha_r t_i) \Big]^2 , \text{ where } \alpha_0 \geqslant 0,$$

$$= \sum_{i=1}^{N} w_i v_i^2 \; . \qquad (2.2.1)$$

It will be convenient to use the following notations:-

$$a_{r,s} = \sum_{i=1}^{N} w_i \exp(-(\alpha_r + \alpha_s) t_i), \quad a_r' = \sum_{i=1}^{N} w_i y_i \exp(-\alpha_r t_i),$$

$$b_{r,s} = \sum_{i=1}^{N} w_i t_i \exp(-(\alpha_r + \alpha_s) t_i), \quad b_r = \sum_{i=1}^{N} w_i t_i y_i \exp(-\alpha_r t_i),$$

$$d_{r,s} = \sum_{i=1}^{N} w_i t_i^2 \exp(-(\alpha_r + \alpha_s)t_i), \quad d_r = \sum_{i=1}^{N} w_i t_i^2 y_i \exp(-\alpha_r t_i),$$

$$g_r = \sum_{i=1}^{N} w_i v_i \exp(-\alpha_r t_i) \quad , \quad p_r = \sum_{i=1}^{N} w_i v_i t_i \exp(-\alpha_r t_i),$$

$$\text{and} \quad q_r = \sum_{i=1}^{N} w_i v_i t_i^2 \exp(-\alpha_r t_i), \text{ for } r \text{ and } s = 0(1)n.$$

We may note that when $r \neq s$

$$\frac{\partial a_{r,s}}{\partial \alpha_r} = - b_{r,s} \quad , \quad \frac{\partial a_r}{\partial \alpha_r} = - b_r \quad , \quad \frac{\partial b_{r,s}}{\partial \alpha_r} = - d_{r,s}$$

$$\text{and} \quad \frac{\partial b_r}{\partial \alpha_r} = - d_r$$

(i) Using Two Stage Minimization

The unknown parameters $c_0, c_1, \ldots, c_n$ now can be determined from the set of equations (2.1.2), which can be written as

$$Ac = a, \qquad (2.2.2)$$

where A is a square symmetric positive definite matrix with elements $a_{r,s}$, c and a are vectors with elements c_r and a_r respectively.

Applying the generalized Newton-Raphson procedure to the set of equations (2.1.3), taking into account the variations of the c_r with respect to the α_r, we obtain

$$\sum_{s=0}^{n} b_{r,s} c_s - (\sum_{s=0}^{n} d_{r,s} c_s - d_r)\delta\alpha_r + \sum_{s=0}^{n} d_{r,s} c_s \delta\alpha_s + \sum_{k=0}^{n} \sum_{s=0}^{n} b_{r,}$$

$$\frac{\partial c_s}{\partial \alpha_k} \times \delta\alpha_k = b_r,$$

$$(2.2.3)$$

for $r = 0(1)n$.

To determine $\dfrac{\partial c_s}{\partial \alpha_k}$, we differentiate equation (2.1.2) with respect to the α's, then we get

$$\sum_{s=0}^{n} a_{r,s} \frac{\partial c_s}{\partial \alpha_k} + \sum_{s=0}^{n'} c_s \frac{\partial a_{r,s}}{\partial \alpha_k} = \frac{\partial a_r}{\partial \alpha_k},$$

$$(2.2.4)$$

if $r \neq k$, then (2.2.4) becomes

$$\sum_{s=0}^{n} a_{r,s} \frac{\partial c_s}{\partial \alpha_k} = b_{r,k} c_k ,$$

but if $r = k$, then (2.2.4) becomes

$$\sum_{s=0}^{n} a_{k,s} \frac{\partial c_s}{\partial \alpha_k} - \sum_{s=0}^{n} b_{k,s} c_s - b_{k,k} c_k - b_k = 0 .$$

. (2.2.5)

It will be convenient to express the set of equation (2.2.5) in matrix terms.

$$(CK - Z - BH) \delta\alpha = - P,$$

(2.2.6)

where C, B and H are square matrices with elements $d_{r,s}$, $b_{r;s}$ and $h_{r,s}$ respectively; $h_{r,s} = \frac{\partial c_s}{\partial \alpha_r}$ and H is determined from the set of equation (2.2.5) which is (in matrix terms) AH = BK - V, where K and V are diagonal matrices with elements c_r and p_r respectively; Z is a diagonal matrix with element

q_r, $\delta\alpha$ (the correction vector) and P are vectors with elements $\delta\alpha_r$ and P_r .

Equation (2.2.6) may be solved to determine the correction vector $\delta\alpha$ and this will be added to the

the initial

~~finished~~ vector of/values of the α's and the iteration

is repeated until the maximum absolute value of $\delta\alpha$ is

very small.

(ii) Using One Stage Minimization Method

The coefficients are determined from equation (2.2.2). For the corrections of the coefficients and the exponents, we have

$$\frac{\partial S}{\partial c_r} = 2(a_{r,0}\,c_0 + a_{1,1}c_1 + \ldots\ldots + a_{r,n}c_n - a_r),$$

$$(2.2.7)$$

$$\frac{\partial^2 S}{\partial c_r^2} = 2\,b_{r,r} \quad, \quad \frac{\partial^2 S}{\partial c_r \partial c_s} = 2\,b_{r,s},$$

$$\frac{\partial^2 S}{\partial c_r \partial\alpha_r} = 2(b_r - c_r b_{r,r} - (b_{r,0}c_0 + b_{r,1}c_1 + \ldots +$$

$$b_{r,n}c_n)\,),$$

$$\frac{\partial^2 s}{\partial c_s \partial \alpha_s} = -2\, c_s b_{r,s},$$

$$\frac{\partial s}{\partial \alpha_r} = -2\, c_r(b_{r,0}c_0 + b_{r,1}c_1 + \dots + b_{r,n}c_n - b_r),$$

$$(2.2.8)$$

$$\frac{\partial^2 s}{\partial \alpha_r^2} = 2\, c_r(\ d_{n,r}c_r + d_{r,0}c_0 + d_{r,1}c_1 + \dots + d_{r,n}c_n - d_r)$$

and

$$\frac{\partial^2 s}{\partial \alpha_r \partial \alpha_s} = 2\, c_r c_s d_{r,s}, \text{ for } r \text{ and } s = 0(1)n.$$

We may substitute these values into the equations (2.1.6) and (2.1.7), then we obtain

$$x\, \delta\, x = z, \qquad\qquad (2.2.9)$$

where X is a square symmetric matrix with elements $x_{j,k} = \dfrac{\partial^2 S}{\partial x_j \partial x_k}$, x is a vector with elements c_1, c_1, ..., c_n, α_0, α_1, ..., α_n, δ x is a column vector for the corrections with elements δc_0, δc_1, ..., δc_n, $\delta \alpha_0$, $\delta \alpha_1$, ..., $\delta \alpha_n$ and z is the column vector with elements $z_k = -\dfrac{\partial S}{\partial x_k}$ for k and j = 0(1)(2n+1), and can determine the corrections δc_r and $\delta \alpha_r$ from these equations and proceed as we did in (i).

(iii) Using Berman's Method

. The coefficients are determined from (2.2.2) as in the previous methods.   To determine the corrections, we have

$$f_i = \sum_{r=0}^{n} c_r \exp(-\alpha_r t_i), \text{ for } i = 1(1)N$$

Differentiating f_i with respect to c_r and α_r, we get

$$\frac{\partial f_i}{\partial c_r} = \exp(-\alpha_r t_i) \text{ and } \frac{\partial f_i}{\partial \alpha_r} = -c_r t_i \exp(-\alpha_r t_i).$$

Substituting these values in equation (2.1.5), we obtain

$$R = \sum_{i=1}^{N} w_i \left[y_i - \sum_{r=0}^{n} (1 - t_i \delta\alpha_r)c_r + \delta c_r \exp(-\alpha_r, t_i) \right]^2$$

$$(2.2.10)$$

Applying the least-squares method to equation (2.2.10), we get

$$\frac{\partial R}{\partial(\delta c_r)} = \sum_{i=1}^{N} w_i \left[y_i - \sum_{s=0}^{n} (1 - t_i \delta\alpha_s)c_s + \delta c_s \exp(-\alpha_s t_i) \right] \times$$

$$\exp(-\alpha_r t_i) = 0$$

$$\frac{\partial R}{\partial(\delta\alpha_r)} = 2c_r \sum_{i=1}^{N} w_i t_i \left[y_i - \sum_{s=0}^{n} (1 - t_i \delta\alpha_s)c_s + \delta c_s \exp(-\alpha_s t_i) \right.$$

$$\exp(-\alpha_r t_i) = 0$$

that is to say

$$\sum_{s=0}^{n} \delta c_s \sum_{i=1}^{N} w_i \exp(-(\alpha_r + \alpha_s)t_i) - \sum_{s=0}^{n} c_s \delta\alpha_s \sum_{i=1}^{N} w_i t_i \times$$

$$\exp(-(\alpha_r + \alpha_s)t_i) = -\tfrac{1}{2} \frac{\partial S}{\partial c_r},$$

$$\sum_{s=0}^{n} \delta c_s \sum_{i=1}^{N} w_i t_i \exp(-(\alpha_r + \alpha_s)t_i) - \sum_{s=0}^{n} c_s \delta \alpha_s x$$

$$\sum_{i=1}^{N} w_i t_i^2 \exp(-(\alpha_r + \alpha_s)t_i) = \frac{1}{2c_r} \frac{\partial S}{\partial \alpha_r},$$

for all values of r, where $\dfrac{\partial S}{\partial c_r}$ and $\dfrac{\partial S}{\partial \alpha_r}$ are given by (2.2.7) and (2.2.8). The above equations may be expressed in matrix terms such as

$$Q \, \delta q = h. \qquad\qquad (2.2.11)$$

where Q is a square symmetric matrix equal to

$$\begin{bmatrix} A & B \\ B & C \end{bmatrix},$$

δq is a column vector for the corrections, with elements δc_0, δc_1, ... δc_n, $-c_0 \delta \alpha_0$, $-c_1 \delta \alpha_1$, ..., $-c_n \delta \alpha_n$ and h is the column vector with elements

$$-\tfrac{1}{2} \frac{\partial S}{\partial c_0}, \quad -\tfrac{1}{2} \frac{\partial S}{\partial c_1}, \quad \ldots, \quad -\tfrac{1}{2} \frac{\partial S}{\partial c_n}, \quad \tfrac{1}{2}c_0 \frac{\partial S}{\partial \alpha_o},$$

$$\tfrac{1}{2}c_1 \frac{\partial S}{\partial \alpha_1}, \quad \ldots, \quad \frac{1}{2c_n} \frac{\partial S}{\partial \alpha_n}.$$

Berman has used the approximations $\dfrac{\Delta f_i}{\Delta c_r}$ and $\dfrac{\Delta f_i}{\Delta \alpha_r}$ instead of $\dfrac{\partial f_i}{\partial c_r}$ and $\dfrac{\partial f_i}{\partial \alpha_r}$ for his solution. These approximations in fact are not reliable in the sense that the errors in the observations will add errors in the difference equations and this could cause the ill-conditioning of the matrix which was involved in the solution of the parameters.

It can be shown that all the three methods ultimately converge at the same rate but that the initial rates of convergence vary considerably. This will be illustrated by the artificial and experimental data which we have fitted in Chapter 7.

2.3. Uncertainties in the Parameters

The least-squares solution of c has the following properties (Plackett, 1960, and Draper, 1966):

(i)   The vector c which will minimize the sum squares of the residuals does not depend on any distribution properties of the errors.

(ii)   The elements of c could be expressed as a

linear function of the observations y_1, y_2,

..., y_N .

(iii) If M is the variance-covariance matrix of

the vector c, then

$$M = \frac{S}{N - (n+1)} A^{-1},$$

where the diagonal terms of M provide the

variance of c, the off diagonal terms provide

the covariance between the elements of c and

$N-(n+1)$ is the number of degrees of freedom.

(iv)   The least-square estimate of c is better

than any other unbiased estimate of c, in

the sense that it has the least variance.

To justify (iii), let us consider equation

(2.2.2) which is

$$Ac = a = EWy,$$

where $A = EWE'$ and E is a matrix with elements

$e_{si} = \exp(-\alpha_s t_i)$ for $s = 0(1)n$, E' is the

transpose of E, W is a diagonal matrix for the

weights with element w_i and y is a column vector for the observation y_i. From (ii) we have

$$c_r = A_{r0}^{-1} a_0 + A_{r1}^{-1} a_1 + \ldots\ldots + A_{rn}^{-1} a_n,$$

$$= \sum_{i=1}^{N} w_i y_i \left[A_{r0}^{-1} u_0 + A_{r1}^{-1} u_1 + \ldots\ldots + A_{rn}^{-1} u_n \right],$$

$$= \sum_{i=1}^{N} w_i y_i z_r,$$

where

$$u_s = \exp(-\alpha_s t_i) \quad \text{and} \quad z_r = \sum_{s=0}^{n} A_{rs}^{-1} u_s.$$

Then the variance of c_r is $\sum_{i=1}^{N} w_i^{\frac{1}{2}} z_r^2$ times variance of $w_i^{\frac{1}{2}} y_i$ if z_r is independent of y's and $w_i^{\frac{1}{2}} y_i$ is calculated with equal weights.

Now $\quad c = A^{-1} a$

$$= A^{-1} EWy = \left(A^{-1} EW^{\frac{1}{2}} \right) \left(W^{\frac{1}{2}} y \right),$$

then $\displaystyle\sum_{i=1}^{N} w_i z_r^2$ is equal to the diagonal elements

of $(A^{-1}EW^{\frac{1}{2}})(A^{-1}EW^{\frac{1}{2}})'$ which is equal $(A^{-1})'$, hence

$$\text{variance } (c_r) = A_{rr}^{-1} \frac{S}{N - (n+1)} \quad ,$$

$$\text{and covariance } (c_r, c_s) = A_{rs}^{-1} \frac{S}{N - (n+1)} \quad .$$

Some writers, e.g. Berman (1962), have applied similar results to the case of exponential fitting, treating the coefficients and the exponents separately. It is clear, however, that the coefficients c_r and the exponents α_r are related and that any estimates of uncertainties should be based on that fact. In these circumstances, more accurate estimates can be obtained from considering the matrix which is involved in determining the corrections δc_r and $\delta \alpha_r$.

The two stage minimization method does not provide these estimates, since it treats the coefficients and the corrections for the exponents separately (the matrix to determine these corrections is not symmetric), but the one stage minimization gives these estimates, where variance $(\delta x_j) = X_{jj}^{-1}$ $S/(N-(2n+1))$ and covariance $(\delta x_j, \delta x_k) = X_{jk}^{-1} S/(N-(2n+1))$, for j and $k = 0(1)2n+1$. We use these

estimates for the measure of the uncertainties of the
parameters when we use the two stage minimization
method. In the case of Berman's method we obtain
the estimation of the uncertainties where variance
$(\delta q_j) = Q_{jj}^{-1} S/(N-(2n+1))$ and covariance $(\delta q_j, \delta q_k)$
$= Q_{jk}^{-1} S/(N-(2n+1))$, for j and k = 0(1)2n+1.

The accuracy of our estimate as detailed above
has been demonstrated by fitting artificial data
(Chapter 7).

CHAPTER   3

STUDY   OF   PARTICULAR   CASES

3.1.   Introduction

It has been pointed out (Cornfield ~~and~~ et al.,
1960), that the solution of the non-linear equations
for the fitting of exponentials by least-squares are
not necessarily unique.   The analysis showed that,
in the case of two exponentials, there existed at least
one other stationary point which was not a true
minimum.

In this chapter we will discuss the problem of
uniqueness.   To our knowledge the problem has not
been solved at the present time, and the work given
here may assist towards the general solution.   In
particular the result of Cornfield is explained.

3.2.   Single exponential

We shall begin with the study of the simple case
of fitting a single exponential with no constant term,

i.e. $y_i \sim a \exp(-b\, t_i);$ (3.2.1)

Clearly we can take a logorithmic transformation

with appropriate weights (see also 4.2), i.e. we could fit

$$\ln y = \ln a - bt,$$

a normal linear regression. The solution for this problem is unique; this suggests that for general case, the solution may be unique; at least with weights w_i proportional to $1/y_i^2$.

An alternative approach for uniqueness is as follows. The sum squares of residuals of (3.3.1) will be

$$S = \sum_{i=1}^{N} w_i \left(y_i - a \exp(-bt_i) \right)^2. \qquad (3.2.2)$$

By the usual least-squares method we may obtain the coefficient a, i.e.

$$a = \sum_{i=1}^{N} w_i y_i \exp(-bt_i) \bigg/ \sum_{i=1}^{N} w_i \exp(-2bt_i), \qquad (3.2.3)$$

This value of a may be substituted in (3.2.2) giving $\overset{*}{S}(b) = S(b, a(b))$, which is a function of b only. The minimum function $\overset{*}{S}(b)$ with respect to b may be studied.

We have

$$\frac{d\overset{*}{S}}{db} = \frac{\partial S}{\partial b} + \frac{\partial S}{\partial a} \times \frac{da}{db} = \frac{\partial S}{\partial b}, \quad \text{since } \frac{\partial S}{\partial a} = 0,$$

so that $\overset{*}{S}$ has a minimum for the same value of b as S.   Further

$$\frac{d^2\overset{*}{S}}{db^2} = \frac{\partial^2 S}{\partial b^2} + \frac{\partial^2 S}{\partial b \partial a} \quad x \quad \frac{da}{db} ,$$

and

$$\frac{\partial S}{\partial b} = 2a \sum w_i(y_i - a \exp(-bt_i)) \, t_i \exp(-bt_i) , \quad (3.2.7)$$

$$\frac{da}{db} = -\sum w_i t_i \exp(-bt_i) \left[y_i - 2a \exp(-bt_i) \right] \bigg/ \sum w_i \exp(-2bt_i)$$

$$\frac{\partial^2 S}{\partial b \partial a} = \sum w_i t_i \exp(-bt_i) \left[y_i - 2a \exp(-bt_i) \right] ,$$

$$\frac{\partial^2 S}{\partial b^2} = 2a \sum w_i t_i^2 \exp(-bt_i) \left[2a \exp(-bt_i) - y_i \right] .$$

Hence

$$\frac{d^2\overset{*}{S}}{db^2} = 2a \sum w_i t_i^2 \exp(-bt_i) \left[2a \exp(-bt_i) - y_i \right]$$
$$- \sum w_i t_i \exp(-bt_i) \left[2a \exp(-bt_i) - y_i \right]^2 \bigg/ \sum w_i \exp(-bt_i)$$

It is required to show that $\dfrac{d^2\overset{*}{S}}{db^2}$ is positive

when b has the root given by equation (3.2.4) or

alternatively that $\dfrac{dS}{db}$ changes sign from -ve to +ve.

We find that we can only establish this for the case when the observations y_i are given by an exponential, i.e. when

$$y_i = \hat{a} \exp(-b_o t_i)$$

In this case equation (3.3.2) will be

$$a = \hat{a} \; \sum u_i \exp((b - b_o)t_i) \Big/ \sum u_i,$$

where $u_i = w_i \exp(-2bt_i)$.

Substituting the value of y_i and a in equation (3.2.4) we obtain

$$\frac{1}{2a} \frac{\partial S}{\partial b} = \hat{a}\left[\sum u_i t_i \exp((b-b_o)t_i) - \bar{t} \sum u_i \exp((b-b_o)t_i)\right]$$

$$= \hat{a} \sum u_i(t_i - \bar{t})(z_i - \bar{z}) = \hat{a} \, \mathrm{Cov}(t_i, z_i),$$

where $z_i = \exp((b-b_o)t_i), \bar{z} = \sum u_i z_i \big/ \sum u_i, \bar{t} = \sum u_i t_i \big/ \sum u_i$ and $\mathrm{Cov}(t_i, z_i)$ is the covariance between t_i, z_i.

Now if $b = b_o$ (the optimum value of b), then $\frac{\partial S}{\partial b} = 0$; if b is less than b_o then $\frac{\partial S}{\partial b}$ is less than zero, since z_i decreases with t_i and if b is greater than b_o, then $\frac{\partial S}{\partial b}$ is greater than zero, since z_i increases with t_i. Thus there is $b = b_o$, say, for which $S(b_o)$ is less than $S(b_r)$ where b_r does not equal b_o.

In the general case, where the y's are not given exactly by exponential value but have some error, this suggests that only one solution exists and is unique. We examined this numerically by calculating the fit of various sets of data which had been deliberately artificially rounded to ranging accuracy.

Ten artificial decay observations we generated from the curve

$$y_i = a \exp(-b_o t_i),$$

where $a = 10.0$ and $b_o = 0.4$, for $t = 0$ (1) 9.

The observations were then rounded to (i) 5, (ii) 4, (iii) 3, (iv) 2, (v) 1 place (s) after the decimal point. These five sets of data (table (1)) were then fitted using the method above. The results are given in tables (2) and Figures(1).

We may note from these results when b is equal to zero, that the coefficient is equal to the mean of the observations, the sums squares of the residuals is equal to the sum square of the deviations from the mean, the first and second dirivatives are less than zero. When b tends to infinity the coefficient tends to y_i, the sum squares of the residuals is equal $\sum_{i=2}^{10} y^2_i$, the first and the second derivatives tend to

t	y(5dp)	y(4dp)	y(3dp)	y(2dp)	y(1dp)
0.0	10.00000	10.0000	10.000	10.00	10.0
1.0	6.70320	6.7032	6.703	6.70	6.7
2.0	4.49329	4.4933	4.493	4.49	4.5
3.0	3.01194	3.0119	3.012	3.01	3.0
4.0	2.01897	2.0190	2.019	2.02	2.0
5.0	1.35335	1.3534	1.353	1.35	1.4
6.0	0.90718	0.9072	0.907	0.91	0.9
7.0	0.60810	0.6081	0.608	0.61	0.6
8.0	0.40762	0.4076	0.408	0.41	0.4
9.0	0.27324	0.2732	0.273	0.27	0.3

Table (1)

Ten observations generated from the single

exponential curve $y(t) = 10.0 \exp(-0.4t)$,

with 5,4,3,2and 1 decimal place(s) accuracy .

Tables 2

The tabulation of the coefficient , the sum squares of residuals , the first derivative and the second derivative for various values of the exponent (EXPO) , using the data of Table (1) . The minimum sum squares of residuals of 4.8 10^{-11} of Table 2(i) in the case of data accurate to 5 decimal places as one would expect , much smaller than for example the corresponding 3.6 10^{-3} of Table 2(v) for data accurate to 1 decimal place .

EXPO.	COEFFICIENT			S.S.OF RES.			FIRST DER.			SECOND DER.		
0.000	+2.9777	n	+0	+9.2869	n	+1	-4.7052	n	+2	-3.2755	n	+2
0.100	+5.2921	n	+0	+4.7943	n	+1	-3.9485	n	+2	+1.6106	n	+3
0.200	+7.4248	n	+0	+1.7383	n	+1	-2.1509	n	+2	+1.7165	n	+3
0.300	+8.9766	n	+0	+3.3834	n	+0	-7.6325	n	+1	+1.0453	n	+3
0.400	+1.0000	n	+1	+4.8539	n	-11	+4.8338	n	-6	+5.2580	n	+2
0.500	+1.0651	n	+1	+2.0735	n	+0	+3.6740	n	+1	+2.4017	n	+2
0.600	+1.1055	n	+1	+6.6639	n	+0	+5.2661	n	+1	+9.4733	n	+1
0.700	+1.1293	n	+1	+1.2260	n	+1	+5.8058	n	+1	+2.1549	n	+1
0.800	+1.1421	n	+1	+1.8102	n	+1	+5.8176	n	+1	-1.4911	n	+1
0.900	+1.1474	n	+1	+2.3811	n	+1	+5.5700	n	+1	-3.2369	n	+1
1.000	+1.1477	n	+1	+2.9203	n	+1	+5.2032	n	+1	-3.9809	n	+1
1.100	+1.1446	n	+1	+3.4202	n	+1	+4.7914	n	+1	-4.1926	n	+1
1.200	+1.1393	n	+1	+3.8784	n	+1	+4.3739	n	+1	-4.1240	n	+1
1.300	+1.1326	n	+1	+4.2955	n	+1	+3.9712	n	+1	-3.9133	n	+1
1.400	+1.1252	n	+1	+4.6735	n	+1	+3.5934	n	+1	-3.6369	n	+1
1.500	+1.1173	n	+1	+5.0152	n	+1	+3.2446	n	+1	-3.3376	n	+1
1.600	+1.1094	n	+1	+5.3234	n	+1	+2.9259	n	+1	-3.0386	n	+1
1.700	+1.1015	n	+1	+5.6013	n	+1	+2.6364	n	+1	-2.7525	n	+1
1.800	+1.0939	n	+1	+5.8517	n	+1	+2.3747	n	+1	-2.4852	n	+1
1.900	+1.0866	n	+1	+6.0771	n	+1	+2.1387	n	+1	-2.2394	n	+1
2.000	+1.0796	n	+1	+6.2802	n	+1	+1.9261	n	+1	-2.0154	n	+1
2.100	+1.0731	n	+1	+6.4631	n	+1	+1.7349	n	+1	-1.8126	n	+1
2.200	+1.0670	n	+1	+6.6278	n	+1	+1.5630	n	+1	-1.6296	n	+1
2.300	+1.0613	n	+1	+6.7762	n	+1	+1.4084	n	+1	-1.4651	n	+1
2.400	+1.0560	n	+1	+6.9100	n	+1	+1.2694	n	+1	-1.3173	n	+1
2.500	+1.0511	n	+1	+7.0306	n	+1	+1.1444	n	+1	-1.1846	n	+1
2.600	+1.0466	n	+1	+7.1393	n	+1	+1.0320	n	+1	-1.0656	n	+1
2.700	+1.0424	n	+1	+7.2374	n	+1	+9.3088	n	+0	-9.5889	n	+0
2.800	+1.0386	n	+1	+7.3258	n	+1	+8.3986	n	+0	-8.6314	n	+0
2.900	+1.0352	n	+1	+7.4056	n	+1	+7.5792	n	+0	-7.7722	n	+0
3.000	+1.0320	n	+1	+7.4777	n	+1	+6.8412	n	+0	-7.0009	n	+0
3.100	+1.0290	n	+1	+7.5427	n	+1	+6.1764	n	+0	-6.3084	n	+0
3.200	+1.0264	n	+1	+7.6014	n	+1	+5.5772	n	+0	-5.6861	n	+0
3.300	+1.0240	n	+1	+7.6544	n	+1	+5.0371	n	+0	-5.1269	n	+0
3.400	+1.0217	n	+1	+7.7023	n	+1	+4.5500	n	+0	-4.6239	n	+0
3.500	+1.0197	n	+1	+7.7456	n	+1	+4.1106	n	+0	-4.1715	n	+0
3.600	+1.0179	n	+1	+7.7847	n	+1	+3.7142	n	+0	-3.7642	n	+0
3.700	+1.0162	n	+1	+7.8200	n	+1	+3.3564	n	+0	-3.3976	n	+0
3.800	+1.0147	n	+1	+7.8519	n	+1	+3.0335	n	+0	-3.0673	n	+0
3.900	+1.0133	n	+1	+7.8808	n	+1	+2.7419	n	+0	-2.7696	n	+0
4.000	+1.0121	n	+1	+7.9069	n	+1	+2.4785	n	+0	-2.5013	n	+0

Table 2(i)

EXPO.	COEFFICIENT	S.S.OF RES.	FIRST DER.	SECOND DER.
0.000	+2.9777 x +0	+9.2868 x +1	=4.7052 x +2	=3.2754 x +‖
0.100	+5.2921 x +0	+4.7942 x +1	=3.9485 x +2	+1.6106 x +‖
0.200	+7.4248 x +0	+1.7382 x +1	=2.1509 x +2	+1.7165 x +‖
0.300	+8.9766 x +0	+3.3833 x +0	=7.6324 x +1	+1.0453 x +‖
0.400	+9.9999 x +0	+3.8614 x =8	=7.3073 x =5	+5.2579 x +‖
0.500	+1.0651 x +1	+2.0735 x +0	+3.6739 x +1	+2.4017 x +‖
0.600	+1.1054 x +1	+6.6637 x +0	+5.2660 x +1	+9.4729 x +‖
0.700	+1.1293 x +1	+1.2260 x +1	+5.8057 x +1	+2.1547 x +‖
0.800	+1.1421 x +1	+1.8102 x +1	+5.8175 x +1	=1.4911 x +‖
0.900	+1.1474 x +1	+2.3810 x +1	+5.5699 x +1	=3.2368 x +‖
1.000	+1.1477 x +1	+2.9202 x +1	+5.2030 x +1	=3.9808 x +‖
1.100	+1.1446 x +1	+3.4201 x +1	+4.7912 x +1	=4.1925 x +‖
1.200	+1.1393 x +1	+3.8783 x +1	+4.3738 x +1	=4.1239 x +‖
1.300	+1.1326 x +1	+4.2954 x +1	+3.9711 x +1	=3.9132 x +‖
1.400	+1.1252 x +1	+4.6734 x +1	+3.5933 x +1	=3.6368 x +‖
1.500	+1.1173 x +1	+5.0150 x +1	+3.2445 x +1	=3.3375 x +‖
1.600	+1.1094 x +1	+5.3233 x +1	+2.9258 x +1	=3.0385 x +‖
1.700	+1.1015 x +1	+5.6011 x +1	+2.6364 x +1	=2.7524 x +‖
1.800	+1.0939 x +1	+5.8515 x +1	+2.3747 x +1	=2.4851 x +‖
1.900	+1.0866 x +1	+6.0769 x +1	+2.1386 x +1	=2.2393 x +‖
2.000	+1.0796 x +1	+6.2800 x +1	+1.9261 x +1	=2.0153 x +‖
2.100	+1.0731 x +1	+6.4629 x +1	+1.7349 x +1	=1.8125 x +‖
2.200	+1.0670 x +1	+6.6276 x +1	+1.5629 x +1	=1.6296 x +‖
2.300	+1.0613 x +1	+6.7760 x +1	+1.4083 x +1	=1.4650 x +‖
2.400	+1.0560 x +1	+6.9098 x +1	+1.2693 x +1	=1.3172 x +‖
2.500	+1.0511 x +1	+7.0304 x +1	+1.1444 x +1	=1.1846 x +‖
2.600	+1.0466 x +1	+7.1391 x +1	+1.0320 x +1	=1.0656 x +‖
2.700	+1.0424 x +1	+7.2371 x +1	+9.3085 x +0	=9.5886 x +‖
2.800	+1.0386 x +1	+7.3256 x +1	+8.3983 x +0	=8.6311 x +‖
2.900	+1.0352 x +1	+7.4054 x +1	+7.5790 x +0	=7.7719 x +‖
3.000	+1.0320 x +1	+7.4774 x +1	+6.8410 x +0	=7.0007 x +‖
3.100	+1.0290 x +1	+7.5425 x +1	+6.1762 x +0	=6.3082 x +‖
3.200	+1.0264 x +1	+7.6012 x +1	+5.5771 x +0	=5.6860 x +‖
3.300	+1.0240 x +1	+7.6542 x +1	+5.0369 x +0	=5.1267 x +‖
3.400	+1.0217 x +1	+7.7021 x +1	+4.5498 x +0	=4.6238 x +‖
3.500	+1.0197 x +1	+7.7454 x +1	+4.1105 x +0	=4.1713 x +‖
3.600	+1.0179 x +1	+7.7844 x +1	+3.7141 x +0	=3.7641 x +‖
3.700	+1.0162 x +1	+7.8198 x +1	+3.3563 x +0	=3.3975 x +‖
3.800	+1.0147 x +1	+7.8517 x +1	+3.0334 x +0	=3.0672 x +‖
3.900	+1.0133 x +1	+7.8805 x +1	+2.7418 x +0	=2.7696 x +‖
4.000	+1.0121 x +1	+7.9066 x +1	+2.4785 x +0	=2.5013 x +‖

Table 2(ii)

EXPO.	COEFFICIENT		S.S.OF RES.		FIRST DER.		SECOND DER.	
0.000	+2.9776 ×	+0	+9.2869 ×	+1	-4.7050 ×	+2	-3.2760 ×	+
0.100	+5.2920 ×	+0	+4.7944 ×	+1	-3.9484 ×	+2	+1.6105 ×	+
0.200	+7.4246 ×	+0	+1.7384 ×	+1	-2.1509 ×	+2	+1.7164 ×	+
0.300	+8.9765 ×	+0	+3.3841 ×	+0	-7.6332 ×	+1	+1.0453 ×	+
0.400	+9.9998 ×	+0	+4.4461 ×	-7	-7.6760 ×	-3	+5.2580 ×	+
0.500	+1.0651 ×	+1	+2.0728 ×	+0	+3.6732 ×	+1	+2.4018 ×	+
0.600	+1.1054 ×	+1	+6.6624 ×	+0	+5.2655 ×	+1	+9.4742 ×	+
0.700	+1.1293 ×	+1	+1.2258 ×	+1	+5.8053 ×	+1	+2.1557 ×	+
0.800	+1.1421 ×	+1	+1.8100 ×	+1	+5.8172 ×	+1	-1.4904 ×	+
0.900	+1.1474 ×	+1	+2.3807 ×	+1	+5.5696 ×	+1	-3.2362 ×	+
1.000	+1.1477 ×	+1	+2.9200 ×	+1	+5.2029 ×	+1	-3.9803 ×	+
1.100	+1.1446 ×	+1	+3.4199 ×	+1	+4.7911 ×	+1	-4.1922 ×	+
1.200	+1.1393 ×	+1	+3.8780 ×	+1	+4.3736 ×	+1	-4.1236 ×	+
1.300	+1.1326 ×	+1	+4.2951 ×	+1	+3.9710 ×	+1	-3.9129 ×	+
1.400	+1.1252 ×	+1	+4.6731 ×	+1	+3.5932 ×	+1	-3.6366 ×	+
1.500	+1.1173 ×	+1	+5.0147 ×	+1	+3.2444 ×	+1	-3.3373 ×	+
1.600	+1.1094 ×	+1	+5.3230 ×	+1	+2.9257 ×	+1	-3.0384 ×	+
1.700	+1.1015 ×	+1	+5.6008 ×	+1	+2.6363 ×	+1	-2.7523 ×	+
1.800	+1.0939 ×	+1	+5.8511 ×	+1	+2.3746 ×	+1	-2.4851 ×	+
1.900	+1.0866 ×	+1	+6.0766 ×	+1	+2.1386 ×	+1	-2.2392 ×	+
2.000	+1.0796 ×	+1	+6.2797 ×	+1	+1.9260 ×	+1	-2.0153 ×	+
2.100	+1.0731 ×	+1	+6.4625 ×	+1	+1.7348 ×	+1	-1.8125 ×	+
2.200	+1.0670 ×	+1	+6.6273 ×	+1	+1.5629 ×	+1	-1.6296 ×	+
2.300	+1.0613 ×	+1	+6.7757 ×	+1	+1.4083 ×	+1	-1.4650 ×	+
2.400	+1.0560 ×	+1	+6.9094 ×	+1	+1.2693 ×	+1	-1.3172 ×	+
2.500	+1.0511 ×	+1	+7.0300 ×	+1	+1.1444 ×	+1	-1.1846 ×	+
2.600	+1.0466 ×	+1	+7.1387 ×	+1	+1.0320 ×	+1	-1.0656 ×	+
2.700	+1.0424 ×	+1	+7.2368 ×	+1	+9.3084 ×	+0	-9.5885 ×	+0
2.800	+1.0386 ×	+1	+7.3252 ×	+1	+8.3983 ×	+0	-8.6310 ×	+0
2.900	+1.0352 ×	+1	+7.4051 ×	+1	+7.5789 ×	+0	-7.7719 ×	+0
3.000	+1.0320 ×	+1	+7.4771 ×	+1	+6.8410 ×	+0	-7.0007 ×	+0
3.100	+1.0290 ×	+1	+7.5421 ×	+1	+6.1762 ×	+0	-6.3081 ×	+0
3.200	+1.0264 ×	+1	+7.6008 ×	+1	+5.5770 ×	+0	-5.6859 ×	+0
3.300	+1.0240 ×	+1	+7.6539 ×	+1	+5.0369 ×	+0	-5.1267 ×	+0
3.400	+1.0217 ×	+1	+7.7017 ×	+1	+4.5498 ×	+0	-4.6238 ×	+0
3.500	+1.0197 ×	+1	+7.7450 ×	+1	+4.1105 ×	+0	-4.1713 ×	+0
3.600	+1.0179 ×	+1	+7.7841 ×	+1	+3.7141 ×	+0	-3.7641 ×	+0
3.700	+1.0162 ×	+1	+7.8194 ×	+1	+3.3563 ×	+0	-3.3974 ×	+0
3.800	+1.0147 ×	+1	+7.8513 ×	+1	+3.0334 ×	+0	-3.0672 ×	+0
3.900	+1.0133 ×	+1	+7.8802 ×	+1	+2.7418 ×	+0	-2.7695 ×	+0
4.000	+1.0121 ×	+1	+7.9063 ×	+1	+2.4785 ×	+0	-2.5013 ×	+0

Table 2(iii)

EXPO.	COEFFICIENT		S.S.OF RES.		FIRST DER.		SECOND DER.	
0.000	+2.9770	$\times$ +0	+9.2829	$\times$ +1	−4.7028	$\times$ +2	−3.2722	$\times$ +2
0.100	+5.2908	$\times$ +0	+4.7927	$\times$ +1	−3.9464	$\times$ +2	+1.6096	$\times$ +3
0.200	+7.4230	$\times$ +0	+1.7381	$\times$ +1	−2.1500	$\times$ +2	+1.7153	$\times$ +3
0.300	+8.9745	$\times$ +0	+3.3864	$\times$ +0	−7.6326	$\times$ +1	+1.0446	$\times$ +3
0.400	+9.9978	$\times$ +0	+5.5706	$\times$ −5	−5.3115	$\times$ −2	+5.2545	$\times$ +2
0.500	+1.0649	$\times$ +1	+2.0670	$\times$ +0	+3.6665	$\times$ +1	+2.4006	$\times$ +2
0.600	+1.1052	$\times$ +1	+6.6495	$\times$ +0	+5.2581	$\times$ +1	+9.4735	$\times$ +1
0.700	+1.1291	$\times$ +1	+1.2238	$\times$ +1	+5.7982	$\times$ +1	+2.1605	$\times$ +1
0.800	+1.1419	$\times$ +1	+1.8073	$\times$ +1	+5.8106	$\times$ +1	−1.4835	$\times$ +1
0.900	+1.1472	$\times$ +1	+2.3774	$\times$ +1	+5.5638	$\times$ +1	−3.2288	$\times$ +1
1.000	+1.1475	$\times$ +1	+2.9161	$\times$ +1	+5.1978	$\times$ +1	−3.9733	$\times$ +1
1.100	+1.1445	$\times$ +1	+3.4155	$\times$ +1	+4.7867	$\times$ +1	−4.1858	$\times$ +1
1.200	+1.1392	$\times$ +1	+3.8733	$\times$ +1	+4.3699	$\times$ +1	−4.1181	$\times$ +1
1.300	+1.1325	$\times$ +1	+4.2900	$\times$ +1	+3.9677	$\times$ +1	−3.9081	$\times$ +1
1.400	+1.1251	$\times$ +1	+4.6677	$\times$ +1	+3.5904	$\times$ +1	−3.6325	$\times$ +1
1.500	+1.1172	$\times$ +1	+5.0090	$\times$ +1	+3.2420	$\times$ +1	−3.3338	$\times$ +1
1.600	+1.1093	$\times$ +1	+5.3171	$\times$ +1	+2.9236	$\times$ +1	−3.0354	$\times$ +1
1.700	+1.1014	$\times$ +1	+5.5947	$\times$ +1	+2.6345	$\times$ +1	−2.7497	$\times$ +1
1.800	+1.0938	$\times$ +1	+5.8449	$\times$ +1	+2.3730	$\times$ +1	−2.4828	$\times$ +1
1.900	+1.0865	$\times$ +1	+6.0702	$\times$ +1	+2.1372	$\times$ +1	−2.2373	$\times$ +1
2.000	+1.0796	$\times$ +1	+6.2731	$\times$ +1	+1.9248	$\times$ +1	−2.0136	$\times$ +1
2.100	+1.0730	$\times$ +1	+6.4559	$\times$ +1	+1.7338	$\times$ +1	−1.8111	$\times$ +1
2.200	+1.0669	$\times$ +1	+6.6205	$\times$ +1	+1.5620	$\times$ +1	−1.6284	$\times$ +1
2.300	+1.0612	$\times$ +1	+6.7689	$\times$ +1	+1.4075	$\times$ +1	−1.4640	$\times$ +1
2.400	+1.0560	$\times$ +1	+6.9025	$\times$ +1	+1.2686	$\times$ +1	−1.3163	$\times$ +1
2.500	+1.0511	$\times$ +1	+7.0230	$\times$ +1	+1.1437	$\times$ +1	−1.1838	$\times$ +1
2.600	+1.0466	$\times$ +1	+7.1317	$\times$ +1	+1.0314	$\times$ +1	−1.0649	$\times$ +1
2.700	+1.0424	$\times$ +1	+7.2297	$\times$ +1	+9.3034	$\times$ +0	−9.5825	$\times$ +0
2.800	+1.0386	$\times$ +1	+7.3181	$\times$ +1	+8.3939	$\times$ +0	−8.6258	$\times$ +0
2.900	+1.0351	$\times$ +1	+7.3979	$\times$ +1	+7.5750	$\times$ +0	−7.7673	$\times$ +0
3.000	+1.0319	$\times$ +1	+7.4699	$\times$ +1	+6.8375	$\times$ +0	−6.9966	$\times$ +0
3.100	+1.0290	$\times$ +1	+7.5349	$\times$ +1	+6.1731	$\times$ +0	−6.3046	$\times$ +0
3.200	+1.0264	$\times$ +1	+7.5936	$\times$ +1	+5.5742	$\times$ +0	−5.6828	$\times$ +0
3.300	+1.0239	$\times$ +1	+7.6466	$\times$ +1	+5.0344	$\times$ +0	−5.1239	$\times$ +0
3.400	+1.0217	$\times$ +1	+7.6944	$\times$ +1	+4.5476	$\times$ +0	−4.6213	$\times$ +0
3.500	+1.0197	$\times$ +1	+7.7377	$\times$ +1	+4.1085	$\times$ +0	−4.1691	$\times$ +0
3.600	+1.0179	$\times$ +1	+7.7767	$\times$ +1	+3.7123	$\times$ +0	−3.7622	$\times$ +0
3.700	+1.0162	$\times$ +1	+7.8120	$\times$ +1	+3.3547	$\times$ +0	−3.3957	$\times$ +0
3.800	+1.0147	$\times$ +1	+7.8439	$\times$ +1	+3.0319	$\times$ +0	−3.0656	$\times$ +0
3.900	+1.0133	$\times$ +1	+7.8728	$\times$ +1	+2.7405	$\times$ +0	−2.7682	$\times$ +0
4.000	+1.0121	$\times$ +1	+7.8988	$\times$ +1	+2.4773	$\times$ +0	−2.5000	$\times$ +0

Table 2(iv)

EXPO.	COEFFICIENT			S.S.OF RES.			FIRST DER.			SECOND DER.		
0.000	+2.9800	x	+0	+9.2716	x	+1	-4.7024	x	+2	-3.2333	x	+2
0.100	+5.2940	x	+0	+4.7834	x	+1	-3.9430	x	+2	+1.6114	x	+3
0.200	+7.4257	x	+0	+1.7327	x	+1	-2.1461	x	+2	+1.7148	x	+3
0.300	+8.9767	x	+0	+3.3662	x	+0	-7.6036	x	+1	+1.0435	x	+3
0.400	+9.9995	x	+0	+3.6245	x	-3	+1.3736	x	-1	+5.2461	x	+2
0.500	+1.0650	x	+1	+2.0859	x	+0	+3.6787	x	+1	+2.3953	x	+2
0.600	+1.1054	x	+1	+6.6784	x	+0	+5.2662	x	+1	+9.4422	x	+1
0.700	+1.1292	x	+1	+1.2274	x	+1	+5.8033	x	+1	+2.1423	x	+1
0.800	+1.1420	x	+1	+1.8113	x	+1	+5.8149	x	+1	-1.4943	x	+1
0.900	+1.1473	x	+1	+2.3819	x	+1	+5.5672	x	+1	-3.2357	x	+1
1.000	+1.1476	x	+1	+2.9209	x	+1	+5.2006	x	+1	-3.9780	x	+1
1.100	+1.1445	x	+1	+3.4205	x	+1	+4.7891	x	+1	-4.1893	x	+1
1.200	+1.1392	x	+1	+3.8785	x	+1	+4.3720	x	+1	-4.1210	x	+1
1.300	+1.1326	x	+1	+4.2954	x	+1	+3.9696	x	+1	-3.9106	x	+1
1.400	+1.1251	x	+1	+4.6733	x	+1	+3.5920	x	+1	-3.6348	x	+1
1.500	+1.1173	x	+1	+5.0148	x	+1	+3.2434	x	+1	-3.3358	x	+1
1.600	+1.1093	x	+1	+5.3230	x	+1	+2.9248	x	+1	-3.0372	x	+1
1.700	+1.1015	x	+1	+5.6007	x	+1	+2.6355	x	+1	-2.7513	x	+1
1.800	+1.0938	x	+1	+5.8510	x	+1	+2.3739	x	+1	-2.4843	x	+1
1.900	+1.0865	x	+1	+6.0764	x	+1	+2.1379	x	+1	-2.2386	x	+1
2.000	+1.0796	x	+1	+6.2794	x	+1	+1.9255	x	+1	-2.0147	x	+1
2.100	+1.0731	x	+1	+6.4622	x	+1	+1.7343	x	+1	-1.8120	x	+1
2.200	+1.0669	x	+1	+6.6269	x	+1	+1.5624	x	+1	-1.6291	x	+1
2.300	+1.0612	x	+1	+6.7752	x	+1	+1.4079	x	+1	-1.4646	x	+1
2.400	+1.0560	x	+1	+6.9089	x	+1	+1.2689	x	+1	-1.3169	x	+1
2.500	+1.0511	x	+1	+7.0295	x	+1	+1.1440	x	+1	-1.1842	x	+1
2.600	+1.0466	x	+1	+7.1382	x	+1	+1.0316	x	+1	-1.0653	x	+1
2.700	+1.0424	x	+1	+7.2362	x	+1	+9.3051	x	+0	-9.5858	x	+0
2.800	+1.0386	x	+1	+7.3246	x	+1	+8.3953	x	+0	-8.6285	x	+0
2.900	+1.0351	x	+1	+7.4044	x	+1	+7.5762	x	+0	-7.7695	x	+0
3.000	+1.0319	x	+1	+7.4764	x	+1	+6.8385	x	+0	-6.9985	x	+0
3.100	+1.0290	x	+1	+7.5414	x	+1	+6.1738	x	+0	-6.3061	x	+0
3.200	+1.0264	x	+1	+7.6001	x	+1	+5.5749	x	+0	-5.6840	x	+0
3.300	+1.0239	x	+1	+7.6531	x	+1	+5.0349	x	+0	-5.1249	x	+0
3.400	+1.0217	x	+1	+7.7010	x	+1	+4.5480	x	+0	-4.6221	x	+0
3.500	+1.0197	x	+1	+7.7442	x	+1	+4.1088	x	+0	-4.1698	x	+0
3.600	+1.0179	x	+1	+7.7833	x	+1	+3.7126	x	+0	-3.7627	x	+0
3.700	+1.0162	x	+1	+7.8186	x	+1	+3.3549	x	+0	-3.3962	x	+0
3.800	+1.0147	x	+1	+7.8505	x	+1	+3.0321	x	+0	-3.0660	x	+0
3.900	+1.0133	x	+1	+7.8793	x	+1	+2.7406	x	+0	-2.7635	x	+0
4.000	+1.0121	x	+1	+7.9054	x	+1	+2.4774	x	+0	-2.5003	x	+0

Table 2(v)

Figures 1

The results of Tables 2 being plotted , where Figure 1(i) corresponds to Table 2(i) and so on . The curves A , S , F1 and F2 represent the coefficient , the sum squares of residuals , the first and the second derivatives respectively (the scaling factors are 1/2 , 1/20 , 1/100 and 1/400 for A,S,F1 and F2 respectively).
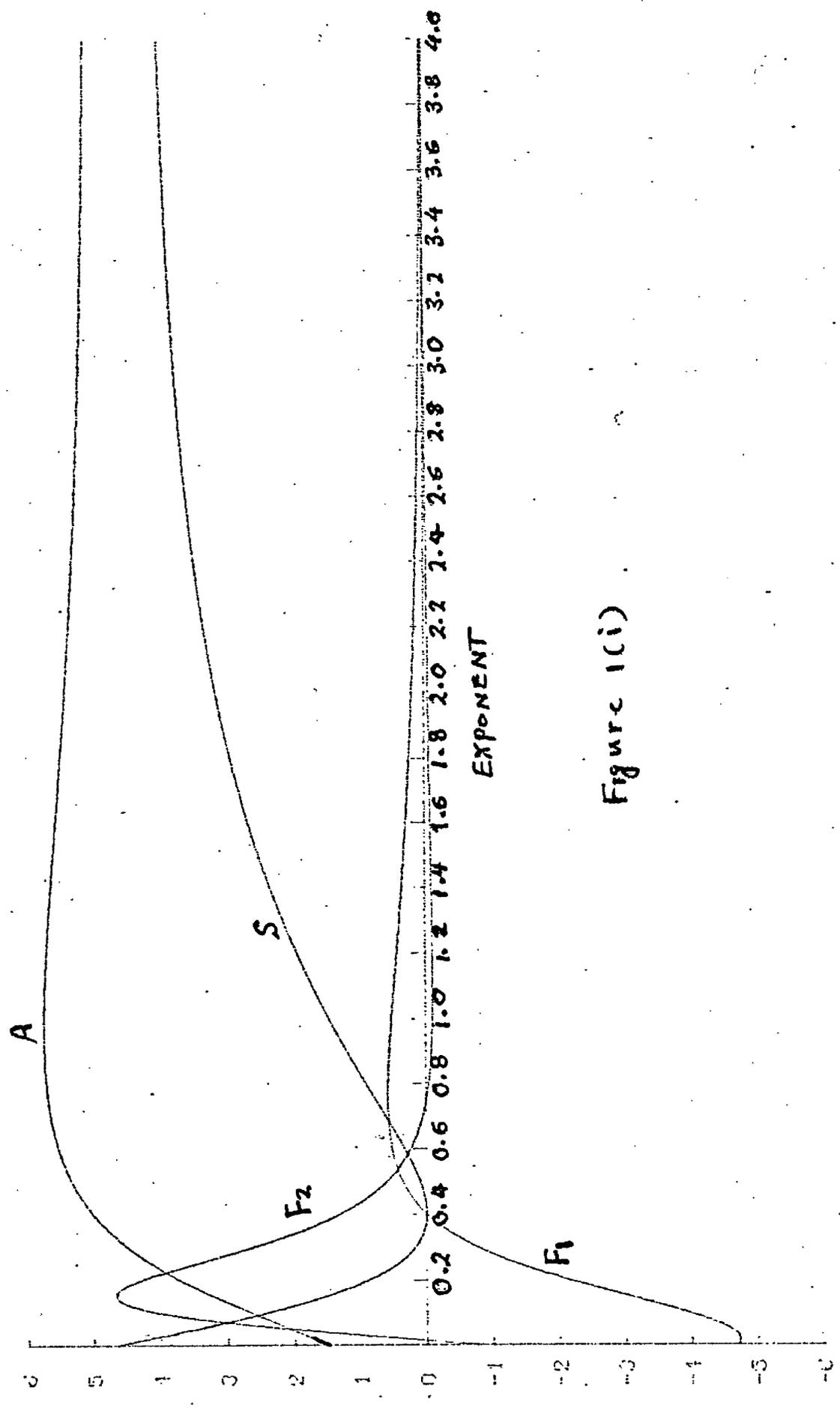
Figure 1(i)

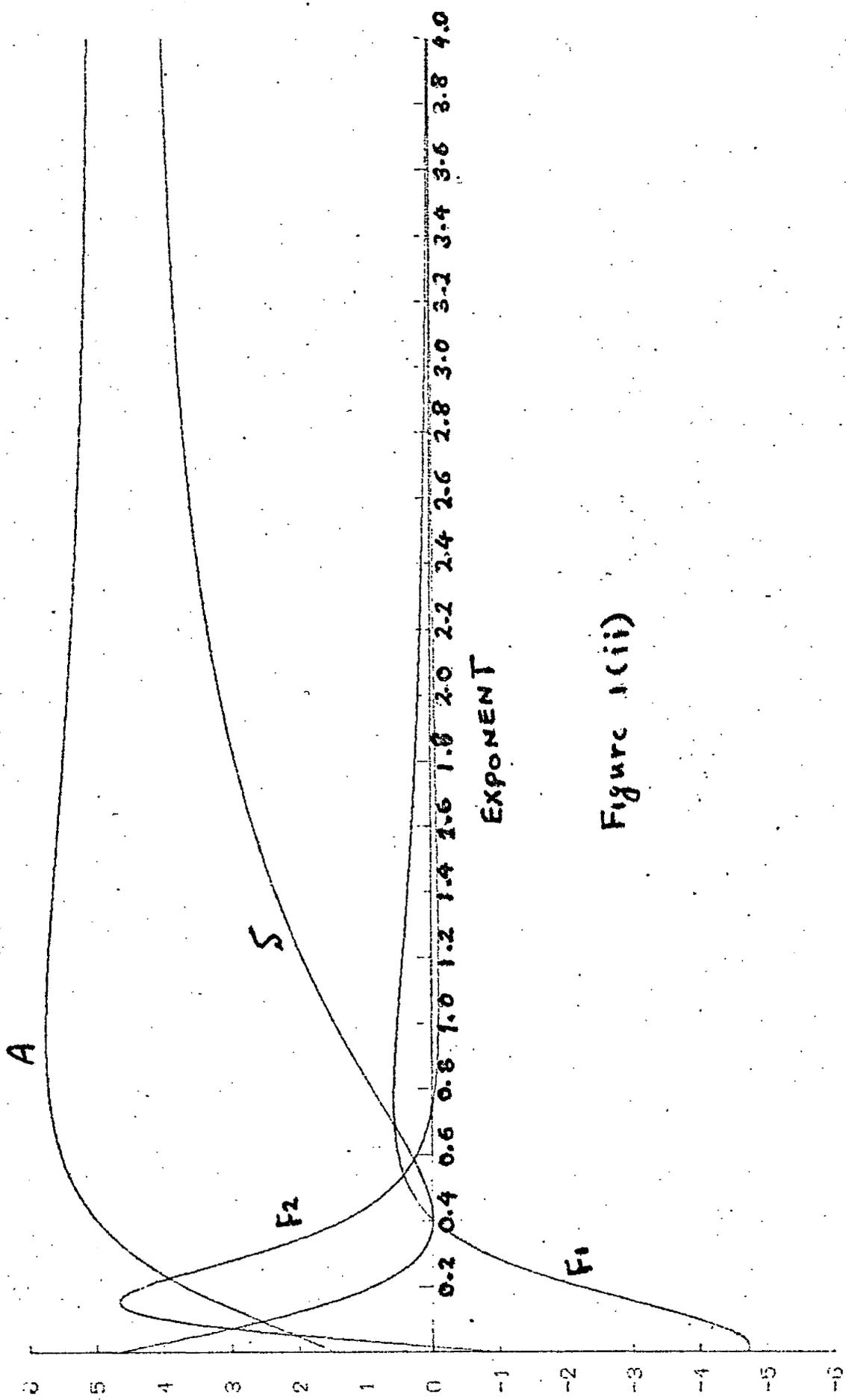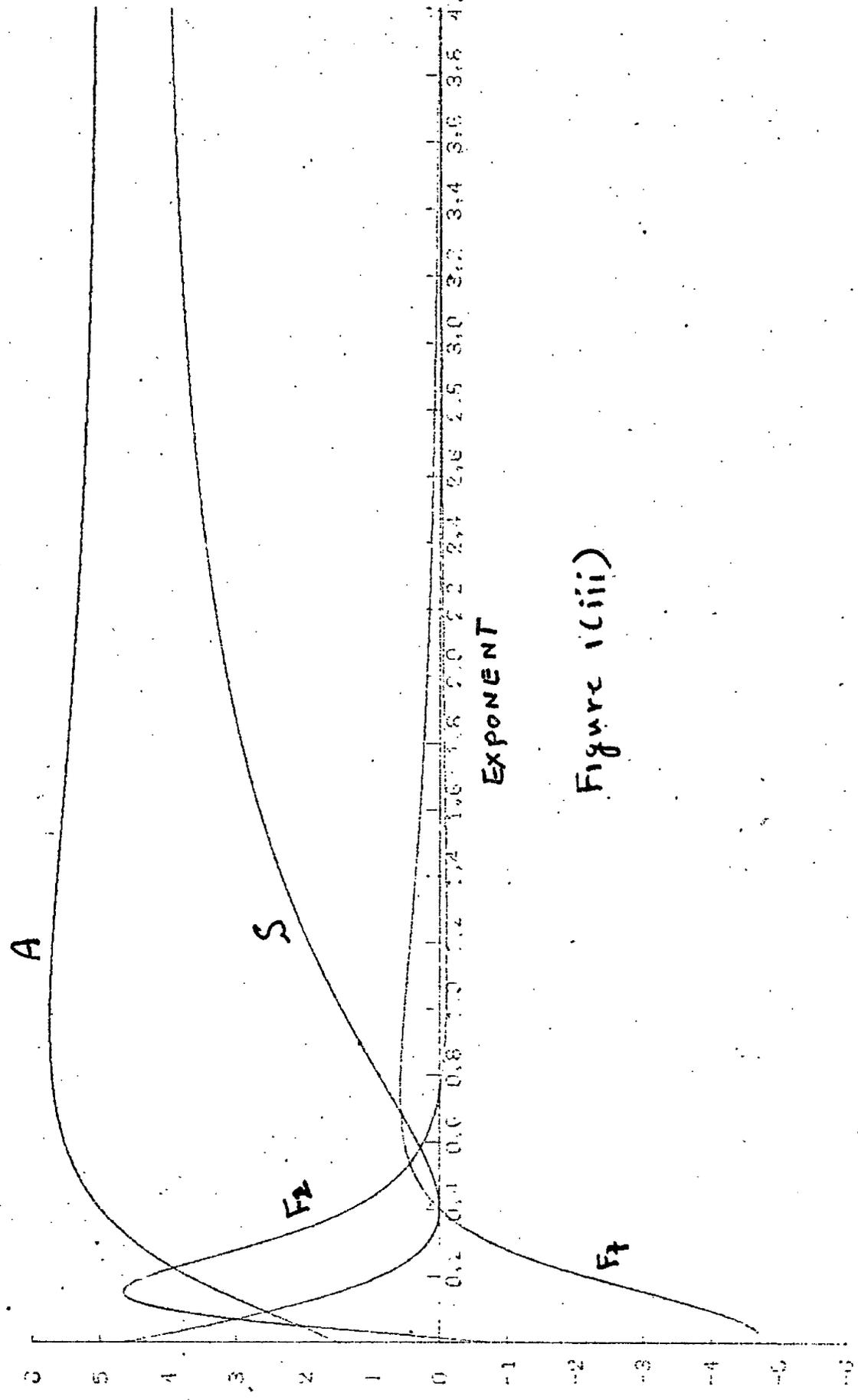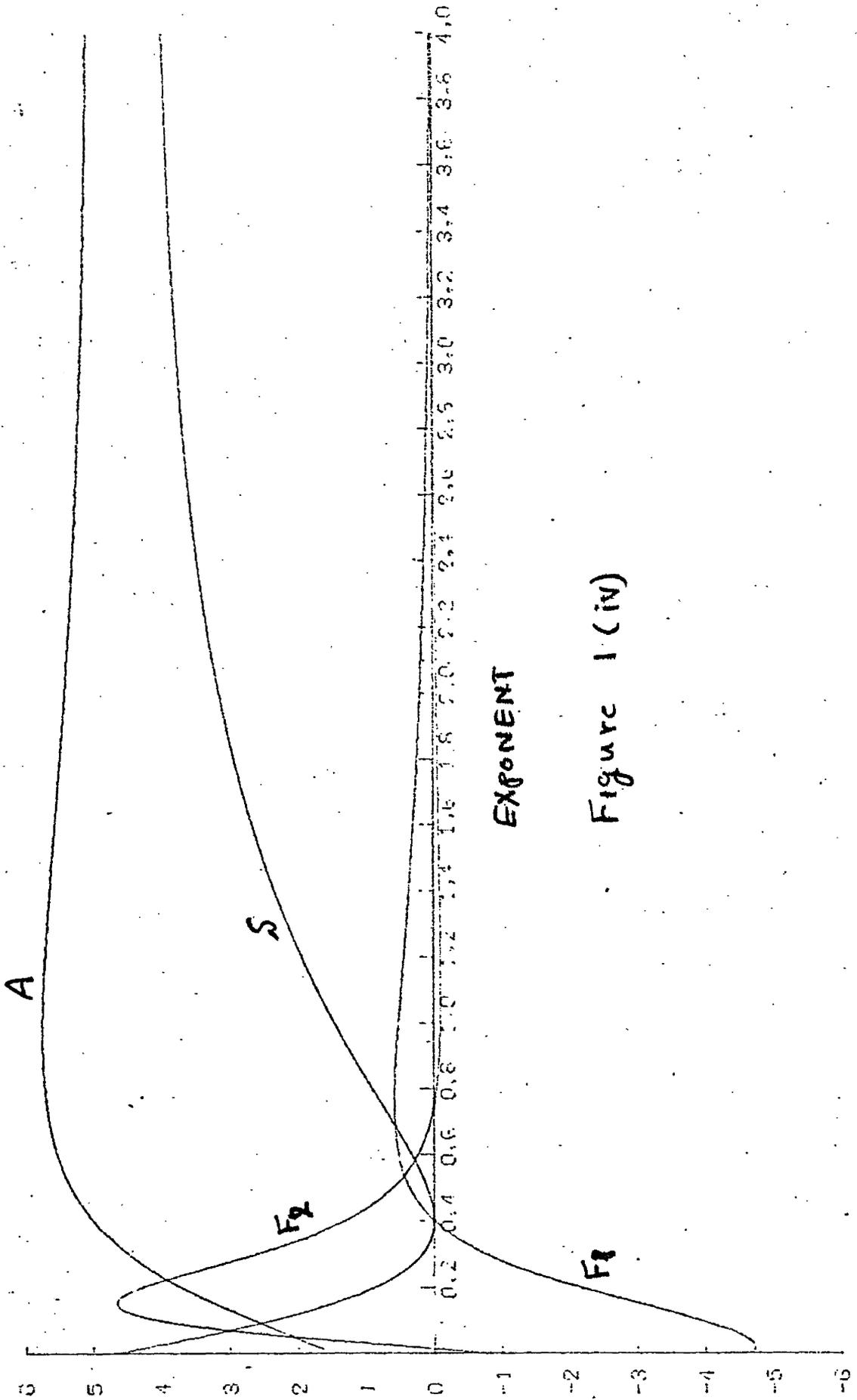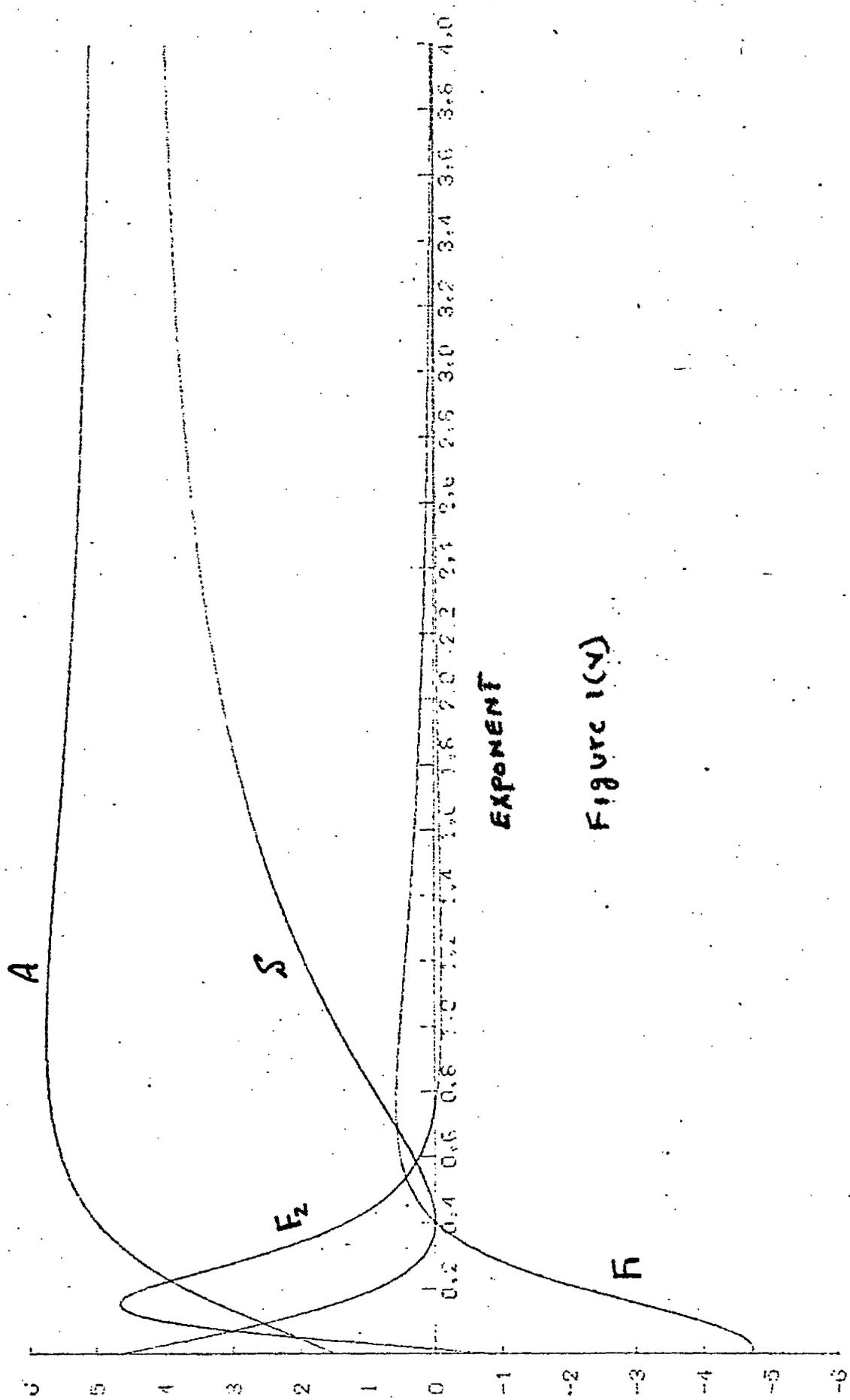EXPONENT

Figure 1(ii)

Figure 1(iii)

EXPONENT

EXPONENT

Figure 1 (iv)

EXPONENT

Figure 1(Y)

zero.   If t_1 is greater than zero, then when b tends to infinity, the coefficient will tend to infinity and the sum squares of the residuals will tend to

$$\sum_{i=2}^{N} y_i^2.$$

3.3.   Single exponential plus constant term

In this case we have the curve,

$$y_i = a_o + a_1 \exp(-bt_i) + u_i,$$

Again by the least-squares method we may determine a_o and a_1 from

$$S = \sum w_i \left[y_i - a_o - a_1 \exp(-bt_i) \right]^2, \quad (3.3.1)$$

i.e. from the equations.

$$\frac{\partial S}{\partial a_o} = -2 \sum w_i \left[y_i - a_o - a_1 \exp(-bt_i) \right] = 0,$$

$$\frac{\partial S}{\partial a_1} = -2 \sum w_i \left[y_i - a_o - a_1 \exp(-bt_i) \right] \exp(-bt_i) = 0. \quad (3.3.2)$$

Those values of a_o and a_1 may be substituted in (3.3.1), giving $S^*(b) = S(b, a_o(b), a_1(b))$, which is a function of b only.   The minimum function $S^*(b)$ with respect to b may be studied.

We have

$$\frac{d\overset{*}{S}}{db} = \frac{\partial S}{\partial b} + \frac{\partial S}{\partial a_o} \times \frac{da_o}{db} + \frac{\partial S}{\partial a_1} \times \frac{da_1}{db} = \frac{\partial S}{\partial b},$$

since $\dfrac{\partial S}{\partial a_o} = \dfrac{\partial S}{\partial a_1} = 0$, so that $\overset{*}{S}$ has a minimum for

the same value of b as S.

Further

$$\frac{d^2\overset{*}{S}}{db^2} = \frac{\partial^2 S}{\partial b^2} + \frac{\partial^2 S}{\partial b \partial a_o} \times \frac{da_o}{db} + \frac{\partial^2 S}{\partial b \partial a_1} \times \frac{da_1}{db}.$$

$$\frac{1}{2a_1} \frac{\partial S}{\partial b} = \sum w_i \left[(y_i - a_o) - a_1 \exp(-bt_i) \right] t \exp(-bt_i); \quad (3.3.3$$

From equation (3.3.2) we have

$$a_i = \sum w_i (y_i - a_o) \exp(-bt_i) \Big/ \sum w_i \exp(-2bt_i).$$

Now suppose in the perfect fit that

$$y_i - a_o = \overset{\wedge}{a_1} \exp(-b_o t_i),$$

then a_1 will become

$$a_1 = \overset{\wedge}{a_1} \sum u_i \exp((b - b_o)t_i) \Big/ \sum u_i,$$

where $u_i = w_i \exp(-2bt_i)$.

Substituting the values of a_1 and $y_i - a_o$ in

equation (3.3.2) we obtain

$$\frac{1}{2a_1}\frac{\partial S}{\partial b} = \hat{a}_1 \left(\sum u_i z_i t_i - \bar{z}\,\bar{t} \sum u_i \right),$$

$$= \hat{a}_1 \text{ Cov}(t_i, z_i) \sum u_i,$$

where $z_i = \exp((b-b_0)t_i)$, $\bar{z} = \sum u_i z_i \Big/ \sum u_i$

$$\text{and } \bar{t} = \sum u_i t_i \Big/ \sum u_i.$$

Hence we obtain the same result as in 3.2.

3.4. Two exponentials

Cornfield ~~and~~ et al. (1960) have pointed out that the least-squares method has several unusual features when applied to linear combinations of exponentials. They have proved that in the case of fitting two exponentials, there exists at least one other stationary point which was not a true minimum but a saddle point. This can be conveniently studied in same manner as Section 3.2.

For the data

$$y_i = a \exp(-ut_i) + b \exp(-vt_i), \qquad (3.4.1)$$

we wish to minimize

$$S = \sum w_i \left[y_i - a \exp(-ut_i) - b \exp(-vt_i) \right]^2 .$$

We may solve for a and b by the normal least-squares procedure for linear regression, so that

a = det a/det .nd b = det b/det, where

$$\det a = \sum w_i y_i \exp(-ut_i) \times \sum w_i \exp(-2vt_i) - \sum w_i y_i \exp(-vt_i)$$
$$\times \sum w_i \exp(-(u+v)t_i),$$

$$\det b = \sum w_i y_i \exp(-vt_i) \times \sum w_i \exp(-2ut_i) - \sum w_i y_i \exp(-ut_i)$$
$$\times \sum w_i \exp(-(u+v)t_i),$$

$$\text{and } \det = \sum w_i \exp(-2ut_i) \sum w_i \exp(-2vt_i)$$
$$- \left[\sum w_i \exp(-(u+v)t_i) \right]^2 .$$

Substituting these values of a and b in S, giving S (u,v), which is then a function of the exponents u and v only.   This is clearly symmetric in u and v so that the normal gradient ($\frac{\partial S}{\partial n}$) to the line u = v must be zero.   However as v tends to u, the expression for a and b become undetermined, and we must therefore proceed in the usual way to determine the limit of these expressions.   We find after much algebra

$$\frac{1}{2}\frac{\partial}{\partial u}(\det b) = \sum w_i y_i \exp(-ut_i)\sum w_i t_i \exp(-2ut_i)$$
$$- \sum w_i y_i t_i \exp(-ut_i) \times \sum w_i \exp(-2ut_i),$$

$$\frac{1}{2}\frac{\partial^2}{\partial u^2}(\det a + \det b) = \sum w_i y_i \exp(-ut_i)\sum w_i t_i \exp(-2ut_i)$$
$$- \sum w_i y_i t_i \exp(-ut_i)\sum w_i t_i \exp(-2ut_i)$$

$$\frac{1}{2}\frac{\partial^2}{\partial u^2}(\det) = \sum w_i \exp(-2ut_i)\sum w_i t_i^2 \exp(-2ut_i)$$
$$- \left[\sum w_i t_i \exp(-2ut_i)\right]^2,$$

and the sum squares of the residuals will then tend
in this case to

$$S = \sum_{i=1}^{N} w_i \left[y_i - (k_1 - k_2 t_i)\exp(-ut_i)\right]^2,$$

where $k_1 = \dfrac{\partial^2}{\partial u^2}(\det a + \det b)\Big/\dfrac{\partial^2}{\partial u^2}(\det)$,

and $k_2 = \dfrac{\partial}{\partial u}(\det b)\Big/\dfrac{\partial^2}{\partial u^2}(\det)$.

If u_o say, is the value of u which gives the
minimum value of S, then $\dfrac{\partial}{\partial u}$ (det b) tends to zero

and $S(u,u)$ tends to $S(u_o)$. This implies that the

exponential $(k_1 - k_2 t_i)\exp(-ut_i)$ tends to the single

exponential $k_1 \exp(-ut_i)$.

We can also show that if v tends to zero then the original sum squares of residuals of the double exponentials will tend to $S = \sum w_i \left[y_i - a \exp(-ut_i) - b \right]^2$ i.e. a sum squares of residuals of single exponential plus constant term.

We have shown that under certain conditions, namely that y is exactly the sum of two exponentials $y_i = a \exp(-u_o t_i) + b \exp(-v_o t)$, then there is a minimum on the line $u = v$ at $u = u_o$ and a minimum on the line $v = o$ at $u = u_1$. It can also be shown that on any line $u \cos \theta + v \sin \theta = o$, there is only one minimum. We have not been able to demonstrate the unique existance of the absolute minimum $u = u_o$, $v = v_o$; but these results support this view.

As previously, these results suggest that for y subject to error, i.e. $y_i = a \exp(-u_o t_i) + b \exp(-v_o t_i) + e_i$, with some conditions on the errors e_i, there is a unique absolute minimum within the region bounded by the lines $u = v$, $v = o$. This hypothesis was again tested numerically. A set of data (table (*3)) was generated from the curve

$$y(t) = 6.0 \exp(-vt) + 4.0 \exp(-ut),$$

for $v = 5$, $u = 1$ and $t = 0$ (0.3) 2.7.

The observations were then rounded to 6, 5, 4, 3 and 2 decimal places.

The sum squares of the residuals was then calculated for $v = 0(0.5)7.0$ and $u = 0(0.5)7.0$ (Figure 2(i)) and was also plotted (Figure 2 (ii) for the data with 6 decimal places accuracy).

We may note that the minimum value of the sum squares of the residuals is at $v = 5$, $u = 1$ and $v = 1$, $u = 5$, and it is equal to 8.26×10^{-5} (for data with 6 decimal places), 2.48×10^{-4} (for data with 2 decimal places). The minimum on the line $u = v$ at $u = 1.5$ is 1.44 and the minimum on the line $v = 0$ at $u = 2.0$ is 0.54.

t	y (6dp)	y (5dp)	y (4dp)	y (3dp)	y (2dp)
0.0	10.000000	10.00000	10.0000	10.000	10.00
0.3	5.337428	5.33743	5.3374	5.337	5.34
0.6	3.492020	3.49202	3.4920	3.492	3.49
0.9	2.484856	2.48486	2.4848	2.485	2.48
1.2	1.816079	1.81608	1.8161	1.816	1.82
1.5	1.340992	1.34099	1.3410	1.341	1.34
1.8	0.992288	0.99229	0.9923	0.992	0.99
2.1	0.734846	0.73485	0.7348	0.735	0.73
2.4	0.544332	0.54433	0.5443	0.544	0.54
2.7	0.394242	0.39424	0.3942	0.394	0.39

Table (3)

Ten observations generated from the

double exponential curve

$$y(t) = 6.0 \exp(-5t) + 4.0 \exp(-t) ,$$

with 6 , 5 , 4 , 3 and 2 Decimal places .

Figures 2

In Figure 2(i) , the sum squares of residuals is tabulated for v=0(0.5)7.0 and u=0(0.5)7.0 , using the data of Table (3) of 6 decimal places accuracy .

In Figure 2(ii) , some of the tabulated values are plotted (the scaling factors for v and u are equal to 1/10) . The minimum value of the sum squares of residuals is equal to 8.26 $\times$-5 at v=0.5 , u=0.1 and v=0.1 , u=0.5 .
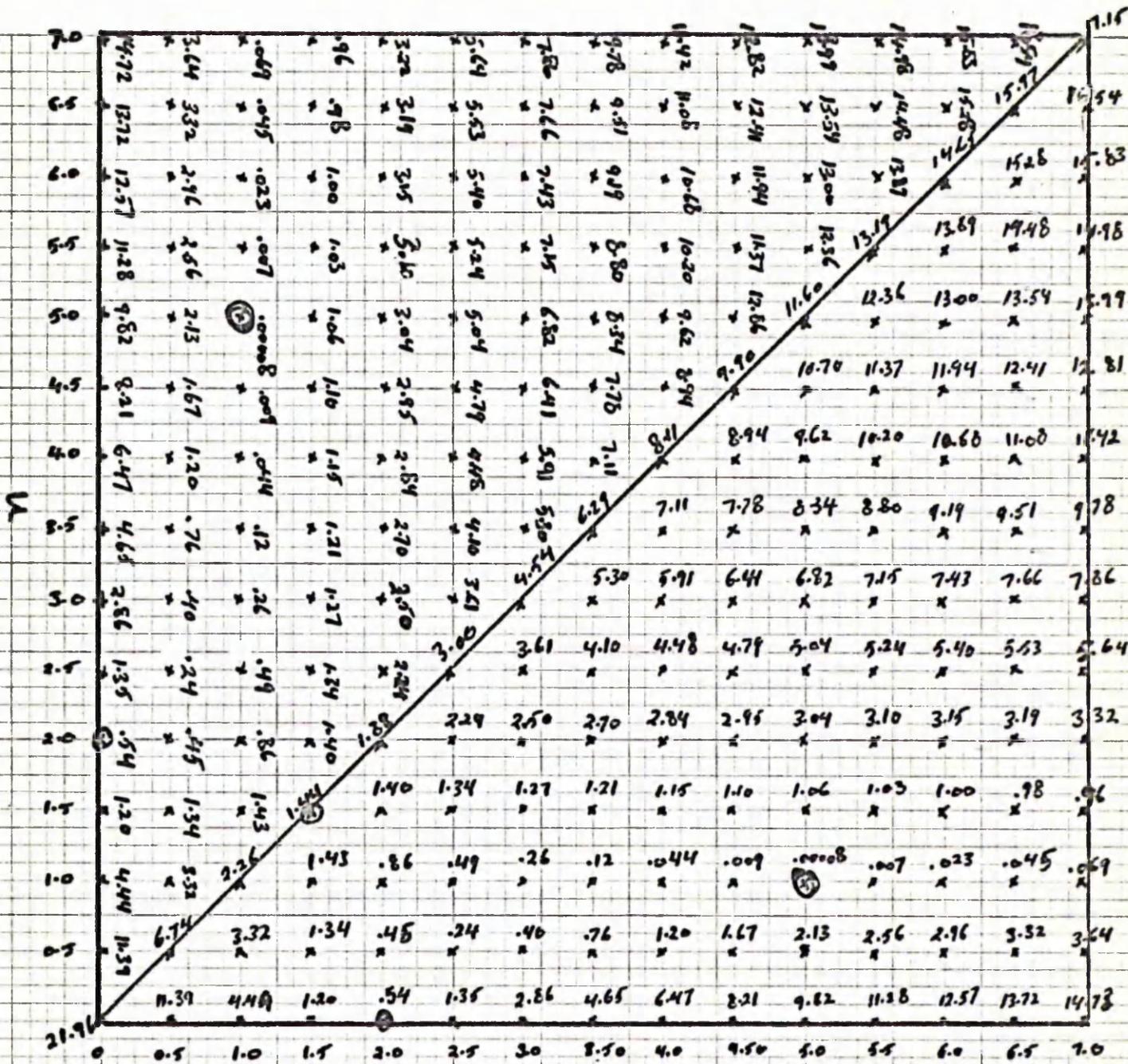
Figure 2(i)
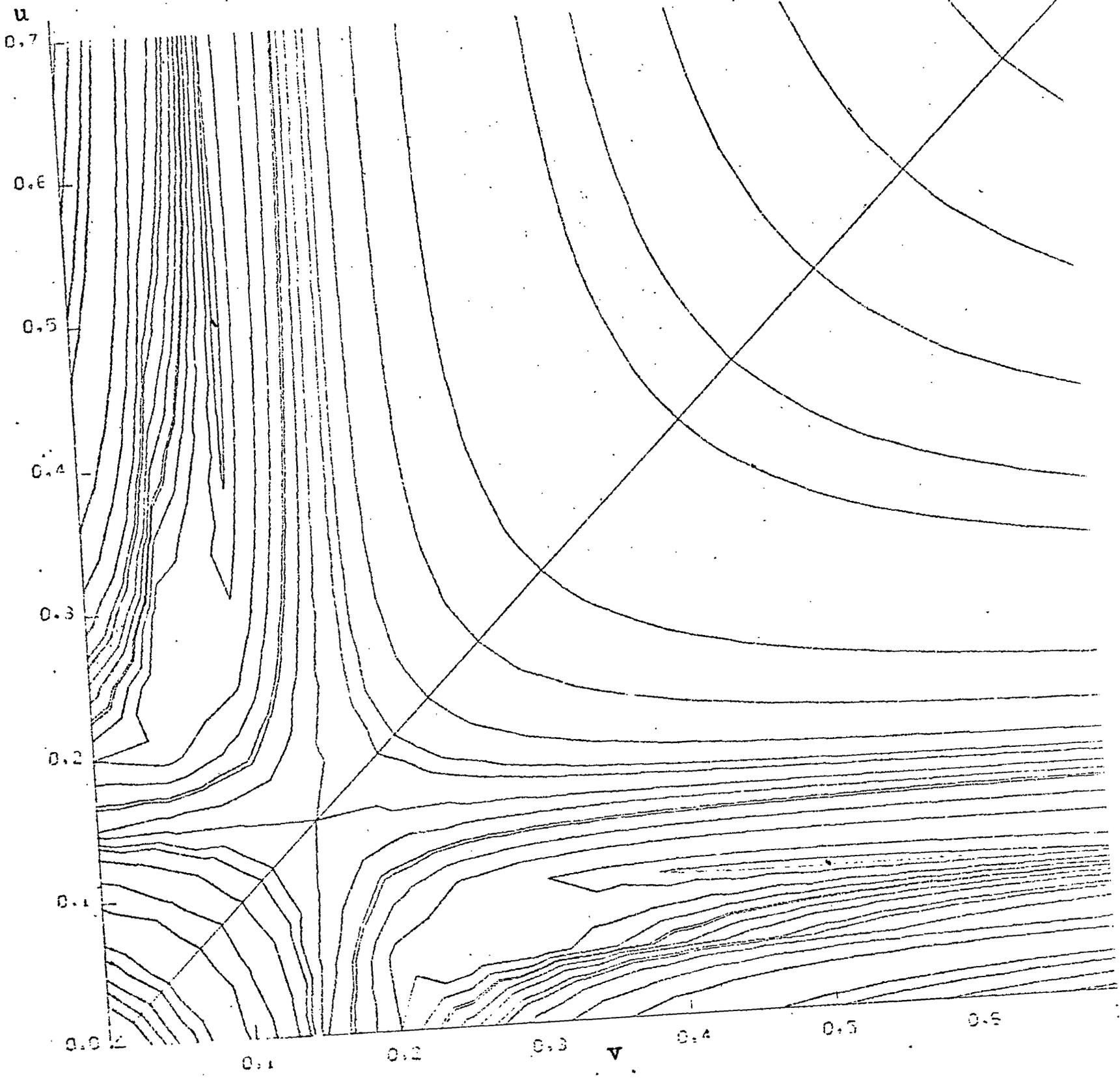
figure 2(ii)

CHAPTER 4

WEIGHTING FACTORS

4.1. Introduction

Frequently the conditions of an experiment are such that certain values of the observations (y_i) may be less reliable than others. In this case greater weights should be given to the measurements of higher reliability. This can be done either by transformations of the data or by associating weight with each observation, the weights form a square diagonal matrix of order N (the number of observations), with positive non-null diagonal elements, the jth element being the weight associated with the observation y_j.

During the investigation of many sets of experimental and artificial data, several points concerning the weighting factors arose. The relative accuracies of data in exponential curve fitting are due to the different sampling techniques and measurement procedure used. In practical cases two different procedures are adopted, which we designate as sampling over fixed counts (see Section 4.2) or sampling over

fixed periods of time (see Section 4.3). Artificial data have also been introduced to the study of the exponential curve fitting problem, notably by Lanczos (1957). Since the accuracy of the results depends significantly upon the weights used (see Chapter 7), it is essential that simulated artificial data should correspond to the correct sampling procedure. This may not in general be the case, and ~~this~~ artificial data may be ~~used~~ subject only to normal round-off error; this is therefore also discussed (Section 4.4).

4.2. Sampling over fixed number of counts

When the time interval is chosen so that the number of observations (pulses) is the same in all cases, we have distribution of $\frac{N}{t}$, where $\frac{1}{t}$ is the measured time to observe N pulses.

The probability that the Nth pulse arrives in time t to t + dt is equal to the probability of N-1 pulses arriving in time t_0 to t multiplied by the probability of a pulse arriving in time t to t + dt, which is

$$\frac{(Pt)^{N-1} \exp(-pt) \times Pdt}{(N-1)!}$$

We may note that the mean $= \int_{0}^{\infty} \dfrac{(Pt)^N \exp(-pt)}{(N-1)!} \, dt$, which

is equal to $\dfrac{P^N N!}{(N-1)! \; p^{N+1}} = \dfrac{N}{P}$.

The distribution of $\dfrac{N}{t} = g$ is $- B(g)dg$,

where $B(g)dg = -\dfrac{(Pt)^{N-1} \exp(-pt)}{(N-1)!} \, Pdt$.

Mean of $\dfrac{N}{t} = \int_{0}^{\infty} g \; B(g)dg = \int_{0}^{\infty} \dfrac{N}{t} \times \dfrac{P^N t^{N-1} \exp(-pt) \, dt}{(N-1)!} = \dfrac{N}{N-1} P$,

and the variance of $\dfrac{N}{t} = -\int_{0}^{\infty} g^2 \; B(g)dg - \left(\dfrac{NP}{N-1}\right)^2$,

$$= \int_{0}^{\infty} \left(\dfrac{N}{t}\right)^2 \dfrac{P^N t^{N-1} \exp(-pt)}{(N-1)!} \, dt - \left(\dfrac{NP}{N-1}\right)^2,$$

$$= \left(\dfrac{NP}{N-1}\right)^2 \times \dfrac{1}{(N-2)}.$$

Thus for constant N, the square root of the variance is proportional to the mean, i.e. the relative error is

constant and is accurately equal to 1/square root of $(N-2)$.

So in this case when we have measurements which are collected over a fixed number of counts, we multiply each term by weight proportional to the inverse of the variance.

Hence in our curve fitting problem we minimize

$$S = \sum_{i=1}^{N} \frac{1}{y_i^2} \left[y_i - \sum_{r=0}^{n} c_r \exp(-\alpha_r, t_i) \right]^2,$$

Data collected in this way may also be dealt with by applying a logarithmic transformation. The sum squares of the residuals will then be

$$S = \sum_{i=1}^{N} w_i' (\ln y_i - \ln f_i)^2, \qquad (4.2.1)$$

and to first order this sum is equal $\sum_{i=1}^{N} w_i (y_i - f_i)^2$, where $w_i' = w_i y_i^2$. And in particular if the original data had equal weights, then w_i' are unequal.

The detailed derivation is as follows.

If we consider $y_i - f_i = v_i$, where the v's are assumed to be very small compared with the y's, then $\ln y_i - \ln f_i = -\ln(f_i/y_i)$. Taking only first order terms in series expansion of $-\ln(f_i/y_i)$, we get

$$\ln y_i - \ln f_i = v_i/y_i + \tfrac{1}{2}(v_i/y_i)^2 + \ldots\ldots \quad ,$$

$$\sim v_i/y_i = (y_i - f_i)/y_i$$

therefore equation $(4.2.1)$ will become

$$S = \sum_{i=1}^{N} w_i' \frac{(y_i - f_i)^2}{y_i^2} , \qquad (4.2.2)$$

If we did not apply the transformation, the sum squares of the residual was

$$S = \sum_{i=1}^{N} w_i (y_i - f_i)^2 , \qquad (4.2.3)$$

Comparing the equations $(4.2.2)$ and $(4.2.3)$, then we obtain

$$\sum \frac{w_i'}{y_i^2} (y_i - f_i)^2 = \sum w_i (y_i - f_i)^2 ,$$

i.e. $w_i' = w_i y_i^2$.

Hence the effect of the transformation on the data collected over fixed number of counts is that the new transformed weights (w_i') are equal to unity, since $w_i = 1/y_i^2$ and $w_i' = w_i y_i^2 = 1$.

4.3. <u>Sampling over fixed periods of time</u>

Suppose the probability that an event takes place in time t to $t + dt$ is $p(t)dt$.

Let $p(t + (r-1)dt) = p_r$, for $r=1(1)n$, then in time t to $t + T$, where $T = n\, dt$, the distribution of events given by g.f (generating functions), (Aitken, 1962) is

$$g.f = (dt)^n (p_1 T + q_1)(p_2 T + q_2) + \ldots + (p_n T + q_n) ,$$

where $\quad q(t)dt = 1 - p(t)dt.$

(i) Assume that p is constant over the interval t to $t + T$. The distribution is then the Binomial Distribution with $g.f = (1 + Pdt(T-1))^n$, and the f.m.g.f (factorial moment generating function) is

where $a = T-1$

$f.m.g.f = (1 + \frac{PTa}{n})^n$, since $T = n\, dt$. The f.m.g.f

will tend to exp(pTa), as n tends to infinity, and this is a Poisson Distribution, then the mean is equal to pT and the variance $p^2T^2 + pT - p^2T^2 = pT$.

(ii) Assume that p is not constant. This is a Poisson Binomial Distribution, with factorial moment generating functions (Aitken, 1962).

f.m.g.f $= (1 + p_1 dt\ a)(1 + p_2\ dt\ a)+ \ldots +(1 + p_n\ dt\ a)$,

$$= 1 + a \sum_{i=1}^{n} p_i\ dt + a^2 \sum_{i=1}^{n-1} p_i\ dt \sum_{j=i+1}^{n} p_j\ dt + \ldots$$

Hence the mean $= \sum_{i=1}^{n} p_i\ dt$,

and u_2 (second factorial moment about the origin) =

$$2 \sum_{i=1}^{n-1} p_i\ dt \sum_{j=i+1}^{n} p_j\ dt.$$

Let us consider the case where $p = \exp(-bt)$,

then we have

$$u_1 \text{ (the mean)} = \int_t^{t+T} \exp(-bt)\, dt,$$

$$= \frac{1}{b}\left[\exp(-bt) - \exp(-b(t+T))\right],$$

$$u_2' = 2\int_t^{t+T} \exp(-bt)\, dt \int_t^{t+T} \exp(-bt)\, dt,$$

$$= 2\int_t^{t+T} \exp(-bt)\, dt \frac{1}{b}\left[\exp(-bt) - \exp(-b(t+T))\right],$$

$$= \frac{1}{b^2}\left[\exp(-bt) - \exp(-b(t+T))\right]^2,$$

$$= u_1^2,$$

since u_2 (the variance) $= u_2' - u_1^2 + u_1,$

then $u_2 = u_1$, this means that the variance is equal to the mean.

We may note that in the finite case for the Poisson Binomial Distribution

$$u_2' = 2 \, (p_1 p_2 + p_1 p_3 + \cdots + p_1 p_n + p_2 p_3 + \cdots + p_2 p_n$$

$$+ \cdots + p_{n-1} \, p_n) ,$$

$$= (p_1 + p_2 + \cdots + p_n)^2 - (p_1^2 + p_2^2 + \cdots + p_n^2),$$

$$= \left(\sum_{i=1}^{n} p_i \right)^2 - \sum_{i=1}^{n} p_i^2 ,$$

whence

$$u_2 = \left(\sum_{i=1}^{n} p_i \right)^2 - \sum_{i=1}^{n} p_i^2 - \left(\sum_{i=1}^{n} p_i \right)^2 + \sum_{i=1}^{n} p_i ,$$

$$= \sum_{i=1}^{n} (p_i - p_i^2)$$

$$= \sum_{i=1}^{n} p_i \, (1 - p_i),$$

$$= \sum_{i=1}^{n} p_i \, q_i$$

In the limiting case, as dt tends to zero, we have

$$u_2 = \int_{t}^{t+T} (\exp(-bt)dt - \exp(-2\,bt)(dt)^2)$$

$$= \int_{t}^{t+T} \exp(-bt)\,dt$$

$$= \frac{1}{b} (1 - \exp(-bT)) \exp(-bt),$$

as previously obtained.

This does not work for the assumption that $p(t)$ is constant unless we use the following integration

method

Integral is

$$
u_2' = \int_t^{t+T} f(t_1)dt_1 \int_t^{t+T} f(t_2)\, dt_2
$$

$$
= \int_t^{t+T} dt_1 \int_t^{t+T} f(t_1)\, f(t_2)\, dt_2
$$

and is over the shaded area (Figure 3).

Clearly we may write

$$
u_2' = \int_t^{t+T} dt_1 \int_{t_1}^{t+T} f(t_1)f(t_2)dt_2 + \int_t^{t+T} dt_1 \int_{t_1}^{t+T} f(t_1)f(t_2)dt_2 .
$$

By changing the variables in second term, we get

$$
u_2' = \int_t^{t+T} dt_1 \int_{t_1}^{t+T} f(t_1)f(t_2)dt_2 + \int_t^{t+T} dt_2 \int_{t_2}^{t+T} f(t_1)f(t_2)dt_1
$$
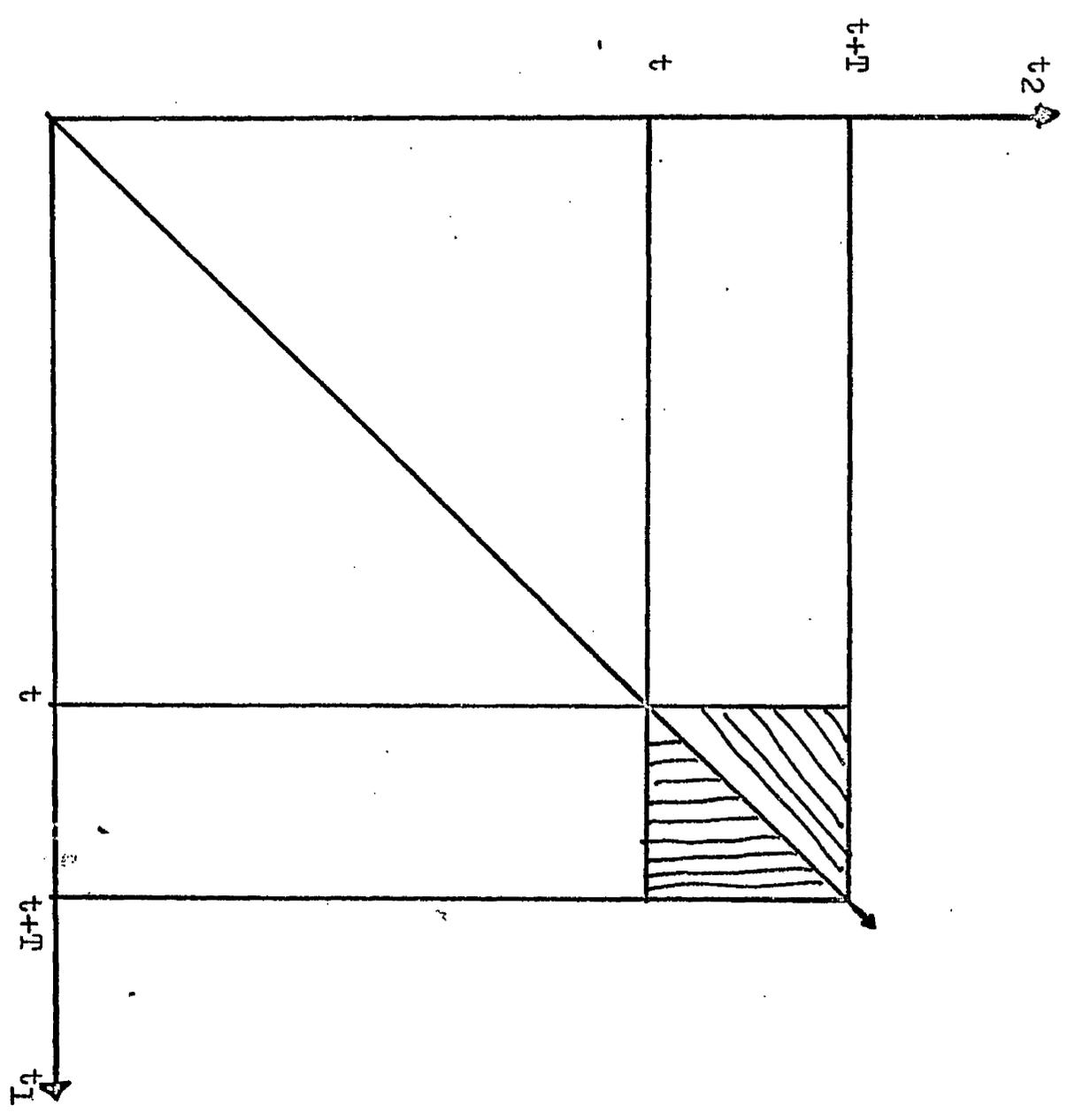
Figure (3)

$$= \int_t^{t+T} dt_1 \int_t^{t+T} dt_2 \; f(t_1) \; f(t_2)$$

$$= \int_t^{t+T} f(t_1) \; dt_1 \int_t^{t+T} f(t_2) \; dt_2$$

$$= \left[\int_t^{t+T} f(t) \; dt \right]^2$$

Now $u_1 = \displaystyle\int_t^{t+T} f(t) \; dt$, so that $u_2' = u_1^2$

whence $u_2 = u_2' - u_1^2 + u_1 = u_1$

Hence this is true for any form of $f(t)$.

So in this case when we have measurements which are collected over a fixed period of time, we multiply each term by weight proportional to the inverse of the variance, so that each variable is distributed with same variance.

Data collected in this way may also be dealt with by applying a square root transformation. The sum squares of the residuals will then be

$$S = \sum_{i=1}^{N} 2 w_i' (y_i^{\frac{1}{2}} - f_i^{\frac{1}{2}})^2 . \qquad (4.3.1)$$

But the original sum squares of the residuals is

$$S = \sum_{i=1}^{N} w_i (y_i - f_i)^2 ,$$

which could take the form

$$S = \sum w_i (y_i^{\frac{1}{2}} - f_i^{\frac{1}{2}})^2 (y_i^{\frac{1}{2}} + f_i^{\frac{1}{2}})^2 .$$

$$(4.3.2)$$

Comparing the equations $(4.3.1)$ and $(4.3.2)$ we obtain

$$\sum_{i=1}^{N} 2 w_i' (y_i^{\frac{1}{2}} - f_i^{\frac{1}{2}})^2 = \sum_{i=1}^{N} w_i (y_i - f_i^{\frac{1}{2}})^2 (y_i^{\frac{1}{2}} + f_i^{\frac{1}{2}})^2 ,$$

i.e. $2 w_i' = w_i \left(y_i^{\frac{1}{2}} + f_i^{\frac{1}{2}} \right)^2 = 2 w_i y_i$, if $y_i \sim f_i$.

Hence the effect of the transformation on the data collected over fixed periods of time is that w_i' is equal to unity since $w_i = 1/y_i$.

4.4. Sampling over fixed number of decimal places

In this case which is usually for data generated artificially, the observations have the same number of decimal places, and the error is rectangularly distributed with constant variance, then the observations will have uniform weights. So in the case we minimize

$$ S = \sum_{i=1}^{N} \left[y_i - \sum_{r=0}^{n} c_r \exp \left(-\alpha_r t_i \right) \right]^2 . $$

In this case, it is of no advantage to transform the data.

CHAPTER 5

OPTIMUM SPACING

5.1. Introduction-optimum spacing in polynomial regression

If in equations (2.1.2), the matrix of the normal equations and the right hand side are given exactly, there are some errors in the numerical solution for the coefficients. These kind of errors have been investigated by Wilkinson (1963). When a relatively small change in the matrix of the normal equations or the constant term (the right hand side), causes a relatively large change in the coefficient vector, then the set of equations (2.1.2) is said to be ill-conditioned with respect to that solution (Fox and Myers, 1968). Hartree (1958) has shown that the measure of ill-conditioning of a system equations can be determined by the ratio $\left| \dfrac{L_{max}}{L_{min}} \right|$, where L_{max} and L_{min} are respectively the largest and the smallest latent roots of the matrix of the equations. If this ratio equal to unity, then the system is well-conditioned, but if on the other hand this ratio is large compared with unity, then the system is ill-conditioned.

Plackett (1960), has shown that the ratio

$$\frac{\text{r.m.s. error of elements of c}}{\text{r.m.s. element of c}} \Big/ \frac{\text{r.m.s. error of element of A}}{\text{r.m.s. element of A}}$$

$$= \frac{\text{norm } (A) \times \text{norm } (A^{-1})}{(n+1)},$$

where A is the matrix of the normal equations and c is the vector of the coefficients. The norm condition-number $\text{norm } (A) \times \text{norm } (A^{-1})/(n+1)$ can be taken as a measure of ill-conditioning in the matrix A.

If the functions $\phi_r (\alpha_r, t_i)$ in (2.1.2) are highly correlated then the system of equations (2.1.2) will be ill-conditioned. The extent of this ill-conditioning may be reduced by re-spacing the observations.

This re-spacing has been examined in the case of polynomials, which are of course linear in the unknown parameters. Considering the polynomial of degree n

$$p(t) = a_o + a_1 t + a_2 t^2 + \ldots + a_n t^n,$$

for N observations $y_i = y(t_i)$, these observations being uncorrelated variables with random error v_i, namely

$$y(t_i) = p(t_i) + v_i, \quad i = 1(1)N, \quad N \quad n + 1.$$

The method of least-squares, gives to determine the coefficients $a_o, a_1, \ldots, a_n$

$$a = (T'WT)^{-1} T'WY = A^{-1} T'WY.$$

In this equation a is the column vector of the coefficient, A is the square symmetric matrix whose elements are $A_{r,s} = \sum_{i=1}^{N} w_i t_i^r t_i^s$, for r and s = o(1) n, A^{-1} is the inverse of A, T is the matrix of independent variables, T' is the transpose of T, Y is the column vector of dependent variables and W is the diagonal matrix of the weights.

De la Garza (1954) has proved that it is possible to choose n + 1 distinct values x_j with $|x_j| \leq 1$, such that $X'HX = T'WT$, and which make the maximum variance of the fitted values as small as possible. He has called the matrix $T'WT$ the information matrix and $X'HX$ is the information matrix of re-spacing. Guest (1958), further showed that the values of the independent variables which will minimize the maximum variance of the fitted values could be obtained from the zeros of the derivatives of Legendre polynomial.

Hoel (1958) has discussed how to choose the values of the independent variables so that the maximum variance of the fitted values will be as small as possible by minimizing the generalized variance given by

$$G.V = \left| X'HX \right|^{-1},$$

i.e. the reciprocal of Vandermonde determinant.
Hoel has shown that the generalized variance of the
fitted values of the polynomial $p(t)$ will be
minimized when the generalized variance of the
estimates of the coefficient of the polynomial is
minimized.

The previous methods and their properties do
not apply to re-space observations from exponential
functions because of the non-linearity.   We may
however use the correlation coefficients between the
functions as criteria for re-spacing these
observations.

5.2.   The correlation between the exponential function.

Suppose we wish to fit the curve

$$y_i = c_o + c_1 \exp(-u_1 t_i) + c_n \exp(-u_n t_i) + v_i,$$

where $i = 1(1)$ N and N $\geqslant$ 2n + 1.

If we put $x_r(t_i) = \exp(-u_r t_i)$, for $r = 1(1)$ n,
then we have from chapter 2

$$\frac{\partial S}{\partial c_o} = -2 \sum_{i=1}^{N} w_i \left[y_i - \sum_{r=0}^{N} c_r x_r(t_i) \right] x_o(t_i) = 0,$$

where $x_o(t_i) = 1$.   This equation could take the form

$$\sum_{r=o}^{N} c_r \sum_{i=1}^{N} w_i x_r(t_i) = \sum_{i=1}^{N} w_i y_i \text{ or } \bar{y} = \sum_{r=o}^{N} c_r \bar{x}_r, \text{ where}$$

$$\bar{x}_r = \sum w_i x_r(t_i) / \sum w_i \text{ and } \bar{y} = \sum w_i y_i / \sum w_i, \text{ are}$$

the means.   This leads naturally to the form

$$y_i - \bar{y} = c_1 (x_1(t_i) - \bar{x}_1) + \dots + c_n(x_n(t_i) - \bar{x}_n).$$

Now in the new form (about the mean) the normal equations will be

$$A^* c^* = a^*,$$

where $A^*_{r;s} = \sum w_i(x_r(t_i) - \bar{x}_r)(x_s(t_i) - \bar{x}_s) / \sum w_i,$

$$a^*_r = \sum w_i(y_i - \bar{y})(x_r(t_i) - \bar{x}_r) / \sum w_i,$$

and $c^*_r = c_r$ for r and s = 1(1) n.   We may

note that c_o is given by $c_o = \bar{y} - \sum_{r=1}^{n} c_r \bar{x}_r.$

If D is a diagonal matrix with elements

$$D_r = (\sum w_i(x_r(t_i) - \bar{x}_r) / \sum w_i)^{\frac{1}{2}},$$

$$M = D^{-1} A^* D^{-1}.$$

This matrix is the correlation matrix of $x_r(t_i)$.   It is a symmetric matrix with diagonal elements equal to unity, and non-diagonal elements

$P_{r,s}$, the correlation coefficient between $x_r(t_i)$ and $x_s(t_i)$.

We may note that $A_{r,s}$ is the covariance between $x_r(t_i)$ and $x_s(t_i)$, D_r^2 is the variance of $x_r(t_i)$ and a_r is the covariance between $x_r(t_i)$ and the y's.

Now the equations to determine the coefficient c_1, c_2, ..., c_n will become

$$D^{-i} A D^{-1} D c = D^{-1} a \text{ or } Mb = D^{-1}a, \text{ if we put}$$

$b = Dc.$

The coefficient c_r thus depend upon the nature of the correlation matrix M. If this matrix is ill-conditioned, then the solution of the coefficients c_r will be badly determined. It can be shown that this is the case if the functions $x_r(t_i)$ are highly correlated. In this case, the best defined solution can be obtained by choosing the values of t_i such that the matrix M is as well-conditioned as possible.

In the case of two variables, the latent roots of the matrix M can be shown to be $1 \pm P$, where P is the correlation coefficient. If we assume the RHS is subject to error $e = k_1 v_1 + k_2 v_2$, where v_1 and v_2

are the latent vectors of M. Then the error in b is given by

$$M \delta b = e,$$

i.e.
$$\delta b = M^{-1} e,$$

$$= M^{-1} (k_1 v_i + k_2 v_2),$$

$$\sim \frac{1}{L_1} k_1 v_1, \text{ if } L_1 \leq L_2,$$

$$\sim \frac{1}{L_1} e.$$

Then $|\delta b| \leq (|\frac{1}{L_1}| + |\frac{1}{L_2}|) e$ and approximately of order $\frac{1}{L_1} e$, and this is equal $\frac{1}{1-p} e$, since $L_1 = 1-p$ in the case of two variables.

For functions $x_r(t_i)$ which are positively correlated, the measure of ill-conditioning is therefore

$$\frac{L_{max}}{L_{min}} = \frac{1 + P}{1 - P}, \text{ where } 0 < P < 1,$$

and clearly we wish to choose the values of t_i so that P be as small as possible. For given forms of distribution of t_i and given function $x_r(t_i)$, the optimum interval for the two variables case can therefore be determined.

These calculations were therefore carried out for the exponentials with $x_r(t_i) = \exp(-u_r t_i)$, for different combination of u_1 and u_2 and different number of observations either equally spaced, i.e.

$$t_i = 0, \ h, \ 2h, \ 3h, \ \ldots, \ (N-1)h,$$

or exponentially spaced

$$t_i = 0, \ h, \ 2h, \ 4h, \ \ldots, \ 2^{N-1}h.$$

In each case, the correlation coefficient, interval h; and its minimum value and the corresponding value of h (or the optimum interval) could be determined. These are given in tables 4 to 15, and graphically in figures 4 to 7.

In the case of exponentials with equal intervals, it can be shown that the correlation coefficient between $\exp(-u_1 t_i)$ and $\exp(-u_2 t_i)$ is

$$P = \mathrm{Cov}(u_1, u_2)/(\mathrm{var}(u_1)\mathrm{var}(u_2))^{\frac{1}{2}},$$

where $\mathrm{Cov}(u_1, u_2) = \dfrac{1-\exp(-(u_1+u_2)hN)}{1-\exp(-(u_1+u_2)h)} - \dfrac{1}{N}\dfrac{(1-\exp(-u_1 Nh))(1-\exp(-u_2 Nh))}{(1-\exp(-u_1 h))(1-\exp(-u_2 h))}$

$$\mathrm{var}(u_1) = \frac{1-\exp(-2u_1 Nh)}{1-\exp(-2u_2 h)} - \frac{1}{N}\left(\frac{1-\exp(-u_1 Nh)}{1-\exp(-u_2 Nh)}\right)^2,$$

and $\mathrm{var}(u_2) = \dfrac{1-\exp(-2u_2 Nh)}{1-\exp(-2u_2 h)} - \dfrac{1}{N}\left(\dfrac{1-\exp(-u_2 Nh)}{1-\exp(-u_2 h)}\right)^2.$

When N tends to infinity and h does not tend to

Tables (4) to (15)

The tabulation of the optimum interval(MININT) and the corresponding minimum correlation coefficient(MINCOR) between the exponential functions $\exp(-u_1 ih)$ and $\exp(-u_2 ih)$, using equal intervals $(i=0,1,\ldots,N-1)$ and exponential intervals $(i=0,1,2,2^2,\ldots,2^{N-2})$ for different combinations of u_1 (EX1) and u_2 (EX2) , and different number of observations (N) .

Table (4) for EX1=1 (1)9 , EX2=2(1)10 and N=5 .

Table (5) for EX1=1 (1)9 , EX2=2(1)10 and N=10 .

Table (6) for EX1=1 (1)9 , EX2=2(1)10 and N=20 .

Table (7) for EX1=1 (1)9 , EX2=2(1)10 and N=30 .

Table (8) for EX1=1 (1)9 , EX2=2(1)10 and N=40 .

Table (9) for EX1=1 (1)9 , EX2=2(1)10 and N=50 .

Table (10) for EX1=1 , EX2=20(10)100 and N=5 .

Table (11) for EX1=1 , EX2=20(10)100 and N=10 .

Table (12) for EX1=1 , EX2=20(10)100 and N=20 .

Table (13) for EX1=1 , EX2=20(10)100 and N=30 .

Table (14) for EX1=1 , EX2=20(10)100 and N=40 .

Table (15) for EX1=1 , EX2=20(10)100 and N=50 .

NUMBER OF OBSERVATIONS =   5

EX1	EX2	EQUAL INTERVAL		EXPON INTERVAL	
		MININT	MINCOR	MININT	MINCOR
1	2	0.7545	0.9740	0.5352	0.9620
1	3	0.6001	0.9405	0.4306	0.9118
1	4	0.5064	0.9130	0.3680	0.8695
1	5	0.4420	0.8911	0.3237	0.8353
1	6	0.3947	0.8735	0.2938	0.8076
1	7	0.3580	0.8592	0.2678	0.7846
1	8	0.3264	0.8473	0.2493	0.7654
1	9	0.3048	0.8373	0.2303	0.7491
1	10	0.2829	0.8286	0.2173	0.7351
2	3	0.4420	0.9907	0.3131	0.9865
2	4	0.3776	0.9740	0.2693	0.9620
2	5	0.3344	0.9567	0.2408	0.9362
2	6	0.3025	0.9405	0.2179	0.9118
2	7	0.2743	0.9260	0.2026	0.8896
2	8	0.2562	0.9130	0.1875	0.8696
2	9	0.2383	0.9014	0.1735	0.8515
2	10	0.2220	0.8911	0.1663	0.8354
3	4	0.3143	0.9953	0.2219	0.9932
3	5	0.2789	0.9855	0.2012	0.9789
3	6	0.2548	0.9741	0.1811	0.9620
3	7	0.2331	0.9624	0.1683	0.9447
3	8	0.2166	0.9511	0.1592	0.9279
3	9	0.2044	0.9406	0.1505	0.9119
3	10	0.1924	0.9307	0.1418	0.8969
4	5	0.2458	0.9972	0.1729	0.9959
4	6	0.2218	0.9907	0.1614	0.9865
4	7	0.2069	0.9827	0.1505	0.9748
4	8	0.1931	0.9741	0.1399	0.9621
4	9	0.1788	0.9653	0.1289	0.9490
4	10	0.1693	0.9567	0.1224	0.9362
5	6	0.2024	0.9981	0.1463	0.9972
5	7	0.1865	0.9936	0.1335	0.9907
5	8	0.1722	0.9876	0.1233	0.9820
5	9	0.1640	0.9810	0.1190	0.9723
5	10	0.1566	0.9741	0.1150	0.9622
6	7	0.1699	0.9986	0.1219	0.9980
6	8	0.1610	0.9953	0.1171	0.9932
6	9	0.1529	0.9908	0.1127	0.9866
6	10	0.1452	0.9855	0.1085	0.9790
7	8	0.1512	0.9990	0.1116	0.9985
7	9	0.1426	0.9964	0.1069	0.9948
7	10	0.1338	0.9928	0.1025	0.9896
8	9	0.1321	0.9992	0.1014	0.9989
8	10	0.1234	0.9972	0.0966	0.9959
9	10	0.1183	0.9994	0.0906	0.9991

Table (4)

		EQUAL INTERVAL		EXPON INTERVAL	
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	2	0.5231	0.9595	0.5040	0.9583
1	3	0.4228	0.9068	0.3944	0.9022
1	4	0.3632	0.8630	0.3266	0.8540
1	5	0.3202	0.8279	0.2812	0.8140
1	6	0.2900	0.7996	0.2487	0.7807
1	7	0.2653	0.7763	0.2215	0.7524
1	8	0.2465	0.7570	0.2028	0.7281
1	9	0.2273	0.7405	0.1853	0.7070
1	10	0.2155	0.7264	0.1705	0.6884
2	3	0.3068	0.9856	0.2976	0.9853
2	4	0.2648	0.9595	0.2534	0.9583
2	5	0.2360	0.9323	0.2212	0.9296
2	6	0.2151	0.9068	0.1992	0.9022
2	7	0.1998	0.8837	0.1795	0.8770
2	8	0.1843	0.8630	0.1653	0.8540
2	9	0.1719	0.8445	0.1542	0.8331
2	10	0.1648	0.8280	0.1440	0.8141
3	4	0.2181	0.9927	0.2121	0.9925
3	5	0.1974	0.9774	0.1884	0.9768
3	6	0.1762	0.9595	0.1692	0.9583
3	7	0.1662	0.9413	0.1561	0.9391
3	8	0.1573	0.9237	0.1446	0.9203
3	9	0.1485	0.9070	0.1338	0.9023
3	10	0.1395	0.8913	0.1238	0.8852
4	5	0.1702	0.9956	0.1651	0.9955
4	6	0.1591	0.9856	0.1512	0.9853
4	7	0.1482	0.9731	0.1390	0.9723
4	8	0.1371	0.9596	0.1273	0.9583
4	9	0.1253	0.9458	0.1192	0.9439
4	10	0.1212	0.9324	0.1131	0.9296
5	6	0.1437	0.9970	0.1363	0.9970
5	7	0.1299	0.9900	0.1240	0.9898
5	8	0.1219	0.9808	0.1165	0.9803
5	9	0.1177	0.9705	0.1103	0.9696
5	10	0.1139	0.9598	0.1049	0.9583
6	7	0.1204	0.9979	0.1152	0.9978
6	8	0.1158	0.9927	0.1087	0.9925
6	9	0.1116	0.9857	0.1031	0.9853
6	10	0.1076	0.9776	0.0981	0.9769
7	8	0.1105	0.9984	0.1022	0.9984
7	9	0.1060	0.9945	0.0969	0.9943
7	10	0.1016	0.9890	0.0920	0.9886
8	9	0.1006	0.9988	0.0913	0.9987
8	10	0.0956	0.9956	0.0863	0.9955
9	10	0.0893	0.9990	0.0806	0.9990

Table (5)

NUMBER OF OBSERVATIONS = 20

		EQUAL INTERVAL		EXPON INTERVAL	
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	2	0.3581	0.9511	0.5429	0.9600
1	3	0.2942	0.8865	0.4249	0.9066
1	4	0.2561	0.8321	0.3548	0.8608
1	5	0.2257	0.7880	0.3062	0.8231
1	6	0.2094	0.7521	0.2695	0.7918
1	7	0.1935	0.7223	0.2432	0.7654
1	8	0.1773	0.6972	0.2201	0.7428
1	9	0.1687	0.6759	0.2040	0.7232
1	10	0.1615	0.6575	0.1885	0.7060
2	3	0.2109	0.9827	0.3195	0.9859
2	4	0.1810	0.9511	0.2718	0.9600
2	5	0.1658	0.9179	0.2396	0.9326
2	6	0.1541	0.8866	0.2143	0.9066
2	7	0.1421	0.8581	0.1957	0.8826
2	8	0.1300	0.8322	0.1780	0.8608
2	9	0.1230	0.8090	0.1660	0.8411
2	10	0.1195	0.7883	0.1563	0.8232
3	4	0.1543	0.9912	0.2273	0.9928
3	5	0.1362	0.9728	0.2030	0.9778
3	6	0.1232	0.9511	0.1827	0.9600
3	7	0.1186	0.9290	0.1671	0.9417
3	8	0.1145	0.9075	0.1558	0.9237
3	9	0.1100	0.8872	0.1457	0.9066
3	10	0.1055	0.8681	0.1360	0.8904
4	5	0.1204	0.9947	0.1767	0.9957
4	6	0.1149	0.9827	0.1619	0.9859
4	7	0.1093	0.9678	0.1499	0.9735
4	8	0.1035	0.9515	0.1391	0.9600
4	9	0.0976	0.9349	0.1285	0.9462
4	10	0.0916	0.9185	0.1207	0.9326
5	6	0.1067	0.9965	0.1472	0.9971
5	7	0.0995	0.9881	0.1354	0.9902
5	8	0.0920	0.9770	0.1242	0.9811
5	9	0.0843	0.9645	0.1176	0.9708
5	10	0.0765	0.9512	0.1120	0.9600
6	7	0.0891	0.9975	0.1228	0.9979
6	8	0.0797	0.9912	0.1158	0.9928
6	9	0.0727	0.9827	0.1100	0.9859
6	10	0.0681	0.9728	0.1050	0.9773
7	8	0.0711	0.9981	0.1091	0.9984
7	9	0.0657	0.9933	0.1038	0.9945
7	10	0.0620	0.9865	0.0991	0.9890
8	9	0.0616	0.9985	0.0983	0.9988
8	10	0.0602	0.9947	0.0936	0.9957
9	10	0.0585	0.9988	0.0865	0.9990

Table (6)

EX1	EX2	EQUAL INTERVAL		EXPON INTERVAL	
		MININT	MINCOR	MININT	MINCOR
1	2	0.2847	0.9482	0.5548	0.9606
1	3	0.2354	0.8794	0.4352	0.9081
1	4	0.2080	0.8212	0.3627	0.8633
1	5	0.1853	0.7736	0.3131	0.8264
1	6	0.1699	0.7346	0.2753	0.7957
1	7	0.1605	0.7022	0.2495	0.7699
1	8	0.1510	0.6748	0.2247	0.7479
1	9	0.1414	0.6513	0.2090	0.7288
1	10	0.1319	0.6308	0.1941	0.7121
2	3	0.1687	0.9817	0.3260	0.9861
2	4	0.1508	0.9483	0.2777	0.9606
2	5	0.1319	0.9129	0.2451	0.9337
2	6	0.1223	0.8795	0.2186	0.9081
2	7	0.1180	0.8491	0.2002	0.8847
2	8	0.1135	0.8217	0.1830	0.8633
2	9	0.1089	0.7971	0.1693	0.8439
2	10	0.1042	0.7747	0.1596	0.8264
3	4	0.1221	0.9907	0.2328	0.9929
3	5	0.1160	0.9713	0.2072	0.9781
3	6	0.1094	0.9485	0.1871	0.9607
3	7	0.1025	0.9250	0.1703	0.9426
3	8	0.0955	0.9020	0.1590	0.9250
3	9	0.0884	0.8800	0.1490	0.9082
3	10	0.0813	0.8591	0.1395	0.8923
4	5	0.1049	0.9944	0.1812	0.9957
4	6	0.0953	0.9818	0.1650	0.9861
4	7	0.0856	0.9658	0.1530	0.9739
4	8	0.0758	0.9483	0.1423	0.9607
4	9	0.0707	0.9305	0.1320	0.9471
4	10	0.0659	0.9129	0.1230	0.9337
5	6	0.0803	0.9962	0.1503	0.9971
5	7	0.0715	0.9873	0.1387	0.9904
5	8	0.0654	0.9755	0.1274	0.9814
5	9	0.0618	0.9622	0.1197	0.9713
5	10	0.0606	0.9483	0.1141	0.9607
6	7	0.0626	0.9973	0.1251	0.9980
6	8	0.0611	0.9907	0.1179	0.9929
6	9	0.0595	0.9817	0.1120	0.9861
6	10	0.0580	0.9713	0.1070	0.9782
7	8	0.0590	0.9980	0.1111	0.9985
7	9	0.0572	0.9929	0.1057	0.9946
7	10	0.0553	0.9858	0.1010	0.9892
8	9	0.0548	0.9964	0.1003	0.9988
8	10	0.0525	0.9944	0.0957	0.9957
9	10	0.0495	0.9967	0.0906	0.9990

Table (7)

| EX1 | EX2 | EQUAL INTERVAL | | EXPON INTERVAL | |
		MININT	MINCOR	MININT	MINCOR
1	2	0.2447	0.9468	0.5603	0.9610
1	3	0.2043	0.8759	0.4399	0.9089
1	4	0.1743	0.8156	0.3664	0.8645
1	5	0.1626	0.7663	0.3163	0.8280
1	6	0.1504	0.7258	0.2790	0.7977
1	7	0.1380	0.6918	0.2523	0.7722
1	8	0.1257	0.6629	0.2279	0.7504
1	9	0.1219	0.6383	0.2113	0.7316
1	10	0.1185	0.6171	0.1966	0.7151
2	3	0.1473	0.9812	0.3297	0.9862
2	4	0.1242	0.9468	0.2809	0.9610
2	5	0.1188	0.9106	0.2477	0.9343
2	6	0.1130	0.8763	0.2207	0.9069
2	7	0.1069	0.8449	0.2023	0.8857
2	8	0.1006	0.8164	0.1853	0.8645
2	9	0.0942	0.7905	0.1708	0.8454
2	10	0.0879	0.7670	0.1612	0.8280
3	4	0.1129	0.9905	0.2354	0.9930
3	5	0.1034	0.9706	0.2092	0.9783
3	6	0.0934	0.9470	0.1892	0.9610
3	7	0.0834	0.9226	0.1718	0.9431
3	8	0.0743	0.8987	0.1604	0.9257
3	9	0.0696	0.8760	0.1505	0.9090
3	10	0.0648	0.8545	0.1411	0.8932
4	5	0.0859	0.9942	0.1833	0.9958
4	6	0.0737	0.9812	0.1664	0.9862
4	7	0.0670	0.9647	0.1545	0.9741
4	8	0.0621	0.9468	0.1439	0.9610
4	9	0.0608	0.9286	0.1337	0.9476
4	10	0.0594	0.9106	0.1240	0.9343
5	6	0.0631	0.9961	0.1517	0.9972
5	7	0.0610	0.9870	0.1402	0.9904
5	8	0.0594	0.9749	0.1291	0.9816
5	9	0.0576	0.9613	0.1208	0.9715
5	10	0.0557	0.9470	0.1151	0.9610
6	7	0.0586	0.9972	0.1268	0.9980
6	8	0.0564	0.9905	0.1189	0.9930
6	9	0.0541	0.9813	0.1130	0.9862
6	10	0.0517	0.9706	0.1079	0.9783
7	8	0.0533	0.9979	0.1120	0.9985
7	9	0.0504	0.9927	0.1067	0.9946
7	10	0.0474	0.9854	0.1019	0.9893
8	9	0.0464	0.9984	0.1012	0.9988
8	10	0.0430	0.9942	0.0966	0.9958
9	10	0.0386	0.9987	0.0916	0.9991

Table (8)

NUMBER OF OBSERVATIONS = 50

		EQUAL INTERVAL		EXPON INTERVAL	
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	2	0.2166	0.9460	0.5636	0.9612
1	3	0.1763	0.8738	0.4427	0.9094
1	4	0.1610	0.8124	0.3685	0.8653
1	5	0.1455	0.7620	0.3182	0.8290
1	6	0.1298	0.7202	0.2811	0.7989
1	7	0.1224	0.6854	0.2540	0.7736
1	8	0.1186	0.6561	0.2297	0.7519
1	9	0.1146	0.6310	0.2126	0.7332
1	10	0.1105	0.6091	0.1950	0.7169
2	3	0.1248	0.9809	0.3318	0.9863
2	4	0.1182	0.9460	0.2827	0.9612
2	5	0.1107	0.9093	0.2492	0.9346
2	6	0.1026	0.8743	0.2218	0.9094
2	7	0.0944	0.8421	0.2035	0.8863
2	8	0.0863	0.8128	0.1866	0.8653
2	9	0.0783	0.7861	0.1717	0.8462
2	10	0.0727	0.7620	0.1621	0.8290
3	4	0.1018	0.9903	0.2368	0.9930
3	5	0.0886	0.9700	0.2103	0.9784
3	6	0.0758	0.9460	0.1904	0.9612
3	7	0.0691	0.9212	0.1726	0.9434
3	8	0.0629	0.8970	0.1613	0.9260
3	9	0.0614	0.8739	0.1514	0.9095
3	10	0.0601	0.8521	0.1420	0.8938
4	5	0.0705	0.9941	0.1845	0.9958
4	6	0.0624	0.9809	0.1673	0.9863
4	7	0.0608	0.9642	0.1553	0.9742
4	8	0.0591	0.9460	0.1447	0.9612
4	9	0.0573	0.9276	0.1346	0.9479
4	10	0.0553	0.9093	0.1246	0.9346
5	6	0.0599	0.9961	0.1525	0.9972
5	7	0.0577	0.9868	0.1411	0.9905
5	8	0.0552	0.9745	0.1300	0.9817
5	9	0.0526	0.9607	0.1214	0.9717
5	10	0.0500	0.9462	0.1156	0.9612
6	7	0.0541	0.9972	0.1278	0.9980
6	8	0.0509	0.9903	0.1195	0.9930
6	9	0.0476	0.9809	0.1135	0.9863
6	10	0.0443	0.9700	0.1065	0.9784
7	8	0.0464	0.9979	0.1126	0.9985
7	9	0.0425	0.9926	0.1072	0.9947
7	10	0.0386	0.9852	0.1025	0.9893
8	9	0.0375	0.9984	0.1017	0.9988
8	10	0.0353	0.9941	0.0971	0.9958
9	10	0.0324	0.9937	0.0922	0.9991

Table (9)

NUMBER OF OBSERVATIONS = 5

		EQUAL INTERVAL		EXPON INTERVAL	
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	20	0.1747	0.7820	0.1450	0.6581
1	30	0.1358	0.7626	0.1152	0.6259
1	40	0.1168	0.7523	0.1001	0.6087
1	50	0.1060	0.7464	0.0805	0.5950
1	60	0.0914	0.7410	0.0561	0.5855
1	70	0.0729	0.7357	0.0606	0.5794
1	80	0.0623	0.7325	0.0580	0.5752
1	90	0.0606	0.7305	0.0553	0.5720
1	100	0.0588	0.7291	0.0523	0.5691

Table (10)

NUMBER OF OBSERVATIONS = 10

		EQUAL INTERVAL		EXPON INTERVAL	
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	20	0.1432	0.6495	0.1047	0.5761
1	30	0.1144	0.6176	0.0711	0.5183
1	40	0.0988	0.6003	0.0567	0.4818
1	50	0.0784	0.5865	0.0485	0.4560
1	60	0.0649	0.5773	0.0395	0.4352
1	70	0.0603	0.5714	0.0332	0.4186
1	80	0.0576	0.5672	0.0298	0.4052
1	90	0.0548	0.5639	0.0279	0.3942
1	100	0.0517	0.5610	0.0261	0.3847

Table (11)

NUMBER OF OBSERVATIONS = 20

		EQUAL INTERVAL		EXPON INTERVAL	
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	20	0.1135	0.5556	0.1147	0.6032
1	30	0.0912	0.5117	0.0851	0.5523
1	40	0.0701	0.4847	0.0619	0.5189
1	50	0.0606	0.4631	0.0551	0.4965
1	60	0.0567	0.4571	0.0485	0.4794
1	70	0.0527	0.4490	0.0407	0.4645
1	80	0.0485	0.4423	0.0345	0.4522
1	90	0.0436	0.4361	0.0307	0.4422
1	100	0.0378	0.4305	0.0291	0.4340

Table (12)

		EQUAL INTERVAL		EXPON INTERVAL	
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	20	0.0936	0.5175	0.1172	0.6123
1	30	0.0720	0.4652	0.0594	0.5637
1	40	0.0604	0.4360	0.0643	0.5312
1	50	0.0555	0.4177	0.0566	0.5097
1	60	0.0507	0.4046	0.0506	0.4937
1	70	0.0455	0.3941	0.0437	0.4798
1	80	0.0397	0.3852	0.0363	0.4677
1	90	0.0355	0.3783	0.0317	0.4580
1	100	0.0322	0.3728	0.0299	0.4502

Table (13)

NUMBER OF OBSERVATIONS = 40

		EQUAL INTERVAL		EXPON INTERVAL	
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	20	0.0843	0.4955	0.1183	0.6167
1	30	0.0622	0.4399	0.0912	0.5692
1	40	0.0563	0.4090	0.0659	0.5373
1	50	0.0507	0.3887	0.0573	0.5161
1	60	0.0450	0.3735	0.0514	0.5005
1	70	0.0386	0.3614	0.0449	0.4872
1	80	0.0347	0.3523	0.0371	0.4751
1	90	0.0312	0.3450	0.0327	0.4657
1	100	0.0302	0.3394	0.0303	0.4579

Table (14)

NUMBER OF OBSERVATIONS = 50

	EQUAL INTERVAL		EXPON INTERVAL	
EX1 - EX2	MININT	MINCOR	MININT	MINCOR
1 - 20	0.0723	0.4815	0.1190	0.6193
1 - 30	0.0590	0.4242	0.0923	0.5725
1 - 40	0.0525	0.3915	0.0667	0.5408
1 - 50	0.0461	0.3693	0.0577	0.5198
1 - 60	0.0394	0.3526	0.0519	0.5044
1 - 70	0.0348	0.3402	0.0456	0.4915
1 - 80	0.0312	0.3307	0.0375	0.4794
1 - 90	0.0300	0.3233	0.0332	0.4702
1 - 100	0.0269	0.3173	0.0304	0.4624

Table (15)

Figures (4) to (7)

The plotting of the correlation coefficient (p) between the exponential functions $\exp(-u_1 ih)$ and $\exp(-u_2 ih)$ as a function of the interval (h) and the number of observations (N) , where $i=0(1$ $N-1$.

Figure (4) for $u_1 = 1$, $u_2 = 2$, N=5(5)25 and infinity .

Figure (5) for $u_1 = 1$, $u_2 = 4$, N=5(5)50 and infinity .

Figure (6) for $u_1 = 1$, $u_2 = 10$, N=5(5)50 and infinity .
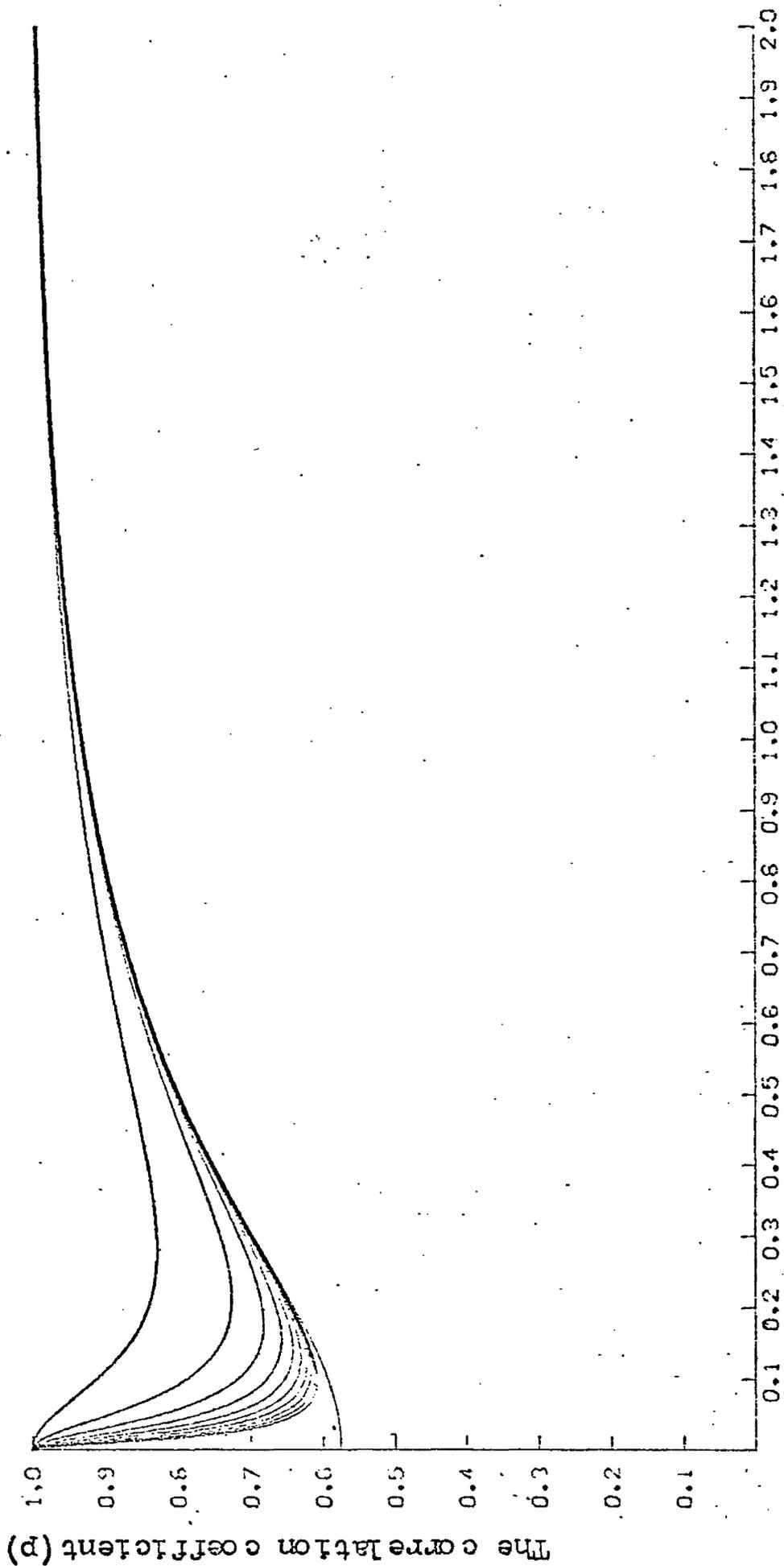
Figure (7) for $u_1 = 1$, $u_2 = 50$, N=5(5)50 and infinity .

The interval(h)

figure (4)

The correlation coefficient (p)

The interval(h)

figure (5)

The correlation coefficient (p)
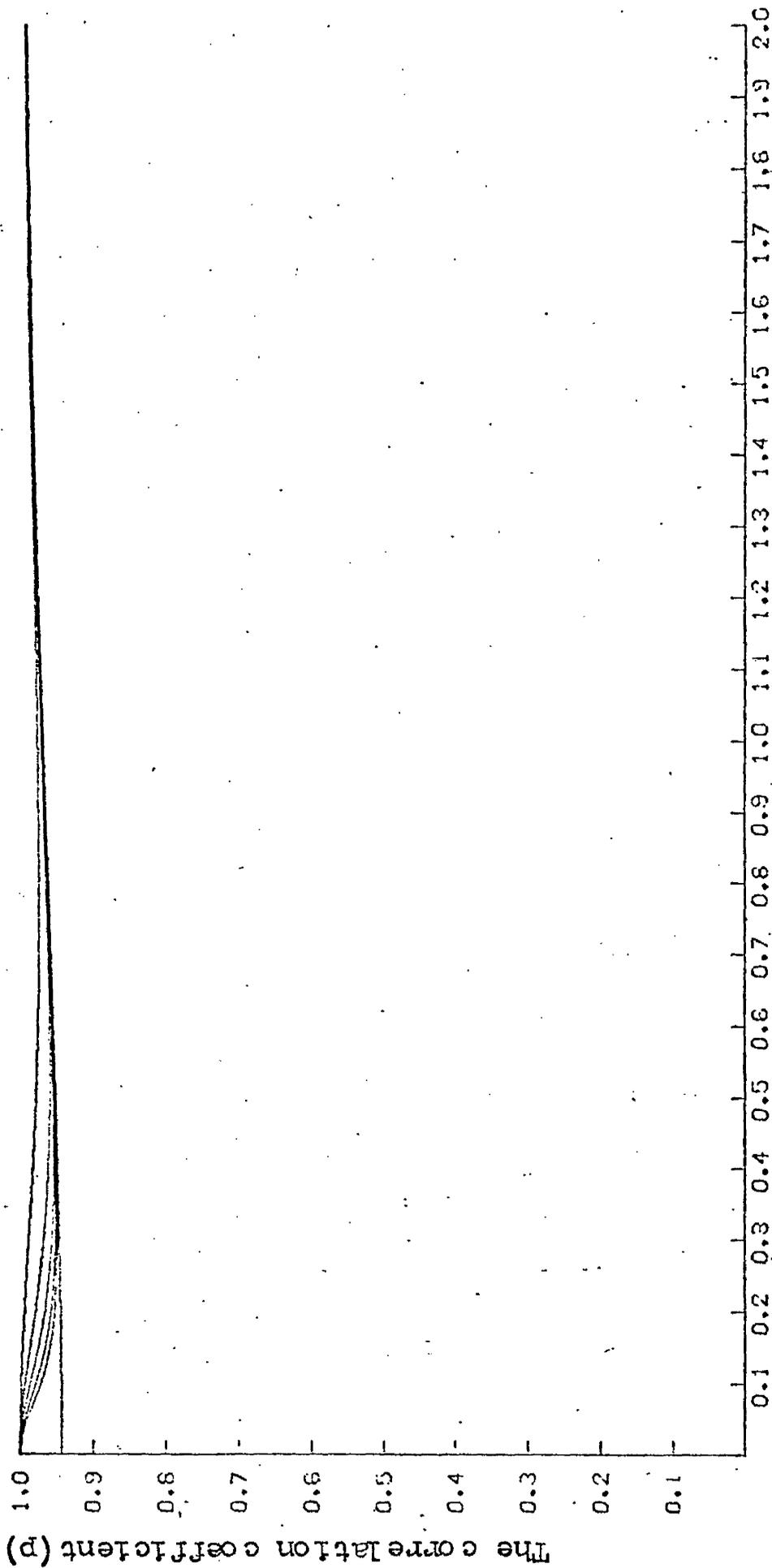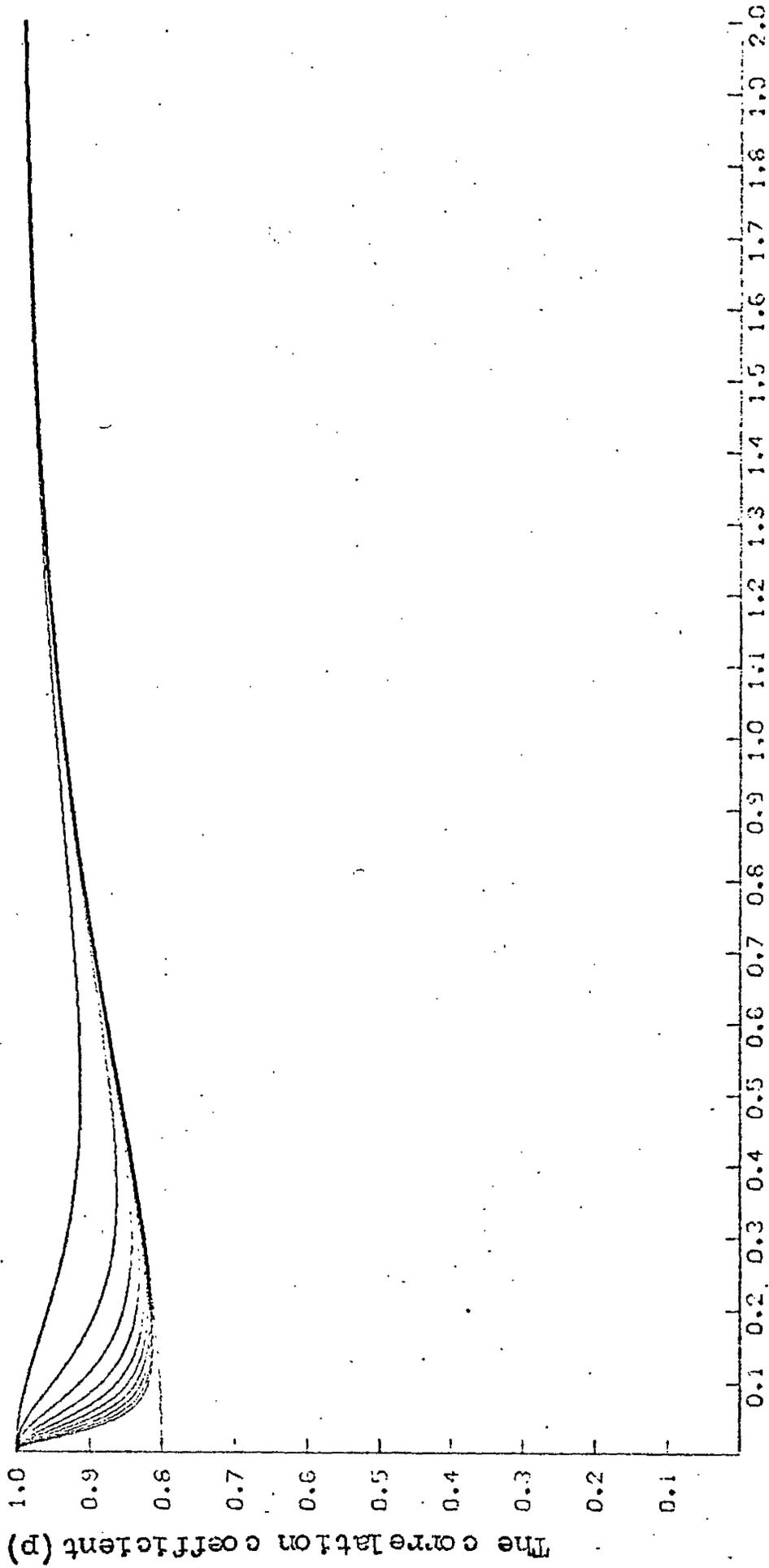
The Interval(h)

figure (6)

The interval(h)

figure (7)

The correlation coefficient (p)
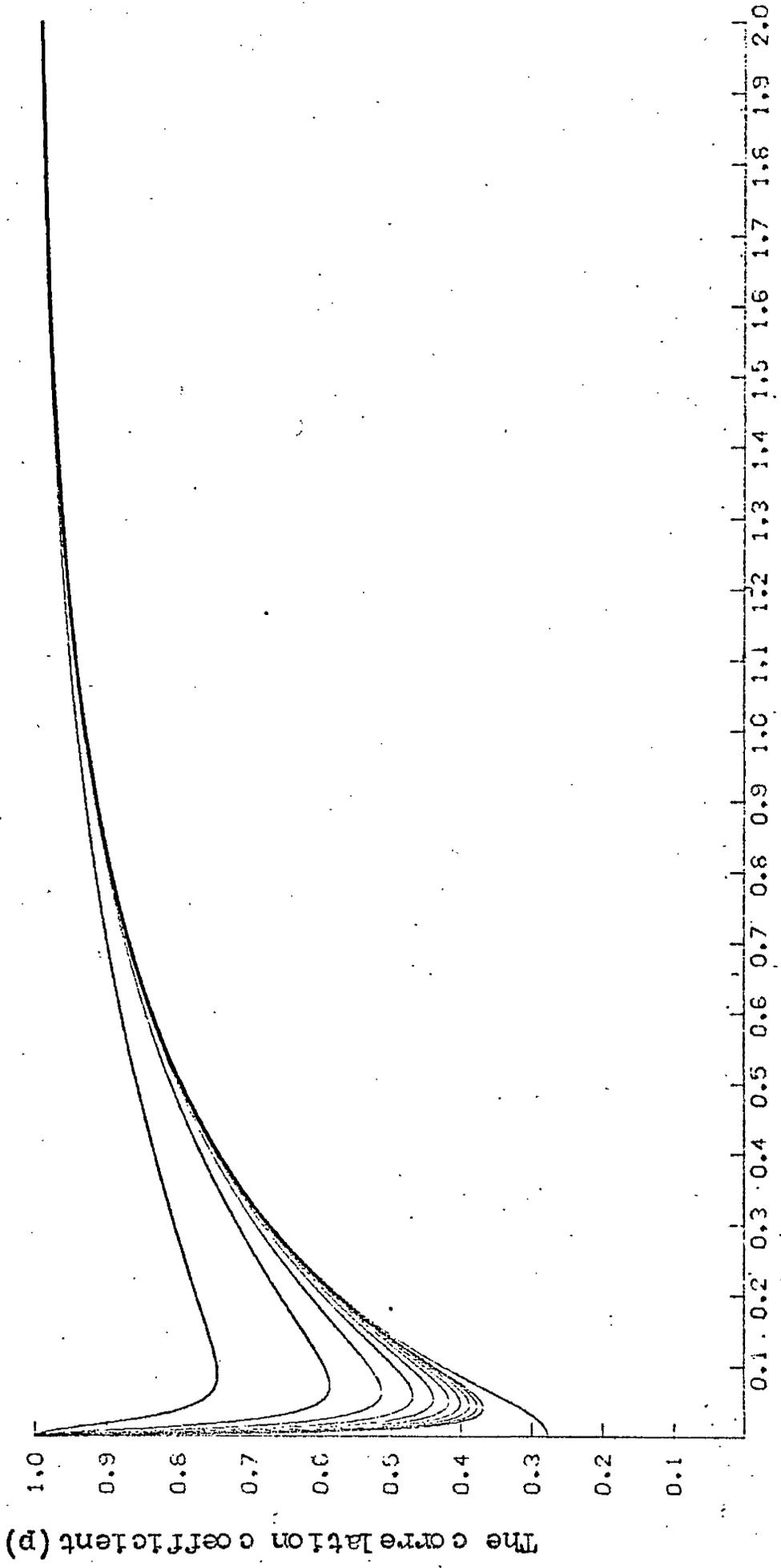
infinity, then p will tend to

$$\frac{1}{1-\exp(-(u_1+u_2)h)} \times \left[(1-\exp(-u_1 h))(1-\exp(-u_2 h)) \right]^{\frac{1}{2}},$$

and if h, tends to zero, then p will tend to its minimum value which is $2(u_1 \times u_2)^{\frac{1}{2}} / (u_1 + u_2)$. In both cases i.e. when N tends to infinity, h tends to infinity and when N finite, h tends to infinity; P will tend to unity.

5.3.   Use of tables

Suppose we have two exponents $u_1 = 1$ and $u_2 = 2$ $(u_1/u_2 = \frac{1}{2})$, and we want to determine the minimum correlations between $\exp(-u_1 ih)$ and $\exp(-u_2 ih)$, and the corresponding optimum interval h, for $i = 0$, (1) 4 (i.e., 5 observations).   From table (4), the optimum interval (MININT) is 0.7545 and the minimum correlation coefficient (MINCOR) is 0.9740, for u_1(EXI) = 1 and u_2(EX$_2$) = 2, for equal spacing; and MININT = 0.5352 and MINCOR = 0.9620, for exponential spacing.

As the number of observations increase we may note from these tables that the best minimum correlation coefficients are those corresponding to

equal spacing.   In the case of using exponential
spacing, MINCOR decreases as N increases (less than
20), and then starts increasing with N.

If it happens that we want to determine MININT
for combinations (ratios) of EX1 and EX2 which are not
in the tables, for example EX1 = .1 and EX2 = .2
for N = 5, we still can use these tables (in this
case table (4)).   From this table MININT for
EX1/EX2 = $\frac{1}{2}$ (where the absolute values for EX1 and
EX2 are 1 and 2) is 0.7545, then MININT for EX1 /
EX2 = $\frac{1}{2}$ (where the absolute values for EX1 and EX2
are .1 and .2) is 0.7545x10 = 7.545 and the
corresponding MINCOR is the same, i.e. is equal to
0.9740.

In the general case when we have more .than
two exponentials, we can still use the correlation
coefficient criteria in re-spacing the observations
by using MININT which correspond to the largest ratio
of the exponents.   This will be shown for Lanczos
data in chapter 7.

CHAPTER 6

EXPONENTIAL CURVE FITTING PROGRAMS

Programs for the fitting of exponentials in accordance with the details of previous chapters have been written in Algol for the English Electric (now ICL) KDF9 computer. The full specification and coding of these programs will be found in the appendix, together with details of other programs written for the calculations of this thesis.

6.1. General exponential curve fitting programs

Three programs are available for the fitting of an exponential curve of the form

$$y \sim c_1 \exp(-\alpha_1 t) + c_2 \exp(-\alpha_2 t) + \ldots + c_n \exp(-\alpha_n t),$$

to $m \ (\geqslant 2n)$ observations y_i, t_i. The programs determine the coefficients c_1, c_2, $\ldots$, c_n, the exponents α_1, α_2, $\ldots$, α_n and their errors by the two stage minimization method, one stage minimization method, and Berman's method respectively (chapter 2).

Each program allows for sets of data to be fitted and different weighting factors to be used.

The programs require initial estimates of
the exponents $\alpha_1, \alpha_2, \ldots, \alpha_n$. If the
exponent α_1 is set to zero, then the fit includes
a constant term, i.e. the fitted curve is

$$y \sim c_1 + c_2 \exp(-\alpha_2 t) + \ldots + c_n \exp(-\alpha_n t).$$

Failures can occur in the iterations if the
initial estimates of the exponents differ considerably
from the final values, or if "overfitting" is
attempted, i.e. the attempted fitting of the data
by an exponential curve with too many terms. Thus
checks are included in the programs to ensure that
the exponents, α_r, are always positive, and the
linear equations to determine the corrections to
both the coefficient c_r and the exponents α_r are
not singular. A check is also provided to see if
sometimes it may happen that the diagonal elements
of the matrix to determine the errors in the
parameters may be negative.

It may happen in the fitting of exponents that
the number of exponents is not known exactly. In
this case, the fitting may be performed for different

values of n, and an analysis of variance using F-test (or variance ratio test) performed to determine the exact degree in a manner similar to that used for the fitting of polynomials (Guest, 1961).

A typical scheme of analysis of variance table for this case is shown in table (16). From F-distribution tables (table (18) of the Biom£trika tables), we could determine the probability of obtaining value of F greater than the calculated

value $\dfrac{s_o - s_1}{2} \times \dfrac{m-2}{s_1}$ for 2 and m-2 degrees of freedom.

If the probability is less or equal to 0.05, then one exponential fit is significant. We do the same thing for $\dfrac{s_1 - s_2}{2} \times \dfrac{m - 4}{s_2}$ and see if the probability corresponds to this value is not greater than 0.05.

Smillie and Anstey (1964) have developed a method to calculate the probabilities in an F-distribution by considering a random variable F in F-distribution with n_1 and n_2 degrees of freedom,

Source of variation	Sum squares	Degree of freedom	variance	F
Total $\sum w_i y_i^2$	s_o	m		
Regression $c_1\exp(-\alpha_1 t_i)$	$s_1 - s_o$	2	$\dfrac{s_1-s_o}{2}$	$\dfrac{s_1-s_o}{2}\,x\,\dfrac{m-2}{s_1}$
Deviation $\sum w_i\left[y_i - c_1\exp(-\alpha_1 t_i)\right]^2$	s_1	$m-2$	$\dfrac{s_1}{m-2}$	
Regression $c_1\exp(-\alpha_1 t_i)+c_2\exp(-\alpha_2 t_i)$	$s_2 - s_1$	2	$\dfrac{s_2-s_1}{2}$	$\dfrac{s_2-s_1}{2}\,x\,\dfrac{m-4}{s_2}$
Deviation $\sum w_i\left[y_i-c_1\exp(-\alpha_1 t_i)-c_2\exp(-\alpha_2 t_i)\right]^2$	s_2	$m-4$	$\dfrac{s_2}{m-4}$	

Table (16)

The analysis of variance table.

where n_1 corresponds to the greater mean square (variance) and n_2 corresponds to the lesser mean square. Considering also the variable

$$u = \frac{\left(1 - \frac{2}{9\,n_2}\right)\, F^{\frac{1}{3}} - \left(1 - \frac{2}{9\,n_1}\right)}{\left(\frac{2}{9\,n_2}\, F^{\frac{2}{3}} + \frac{2}{9\,n_1}\right)^{\frac{1}{2}}}$$

which is approximately normally distributed with zero mean and unit variance. If F_o is the calculated value of F and u_o is the corresponding value of u, then the probability of obtaining F greater than F_o is approximately equal to the probability of obtaining a value of the standard normal variable greater than u_o. This probability may be approximated by

$$\tfrac{1}{2} \left(1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4\right)^{-4}$$

where
$$x = u_o / (2)^{\frac{1}{2}}$$

$$a_1 = 0.278393, \quad a_2 = 0.230389, \quad a_3 = 0.000972$$

and $a_4 = 0.078108.$

.The method has been programmed as a procedure in the general curve fitting programs.

6.2. The estimation of the initial valves of the exponents program

As we know Newton-Raphson procedure needs a good initial guess for the exponents to converge to the required solution.   We tried to use the graphical procedure (peeling off 1.3) to estimate the initial values of the exponents from a set of data to be fitted.   The procedure gave a good result only when the exponents were very different. The difficulty in this method is inherent in the subtraction step which produces a negative value of the new y's (ln of negative value is imaginary).

We have written a program to calculate the initial values of the exponents.   The results from this program were better than the results from the previous procedure.

The program calculates the initial values of the exponents for the following cases.

(a) One exponential: in this case the curve is

of the form

$$y_i = c \exp(-\alpha t_i).$$

Evaluating $S = \sum w_i \left[y_i - c \exp(-\alpha t_i) \right]^2$, for

different values of α and choosing the value of

α which gives the minimum value of S as initial

value of the exponent in the general curve

fitting program.

(b) One exponential plus constant term:- now

we have the curve

$$y_i = c_1 \exp(-\alpha_1 t_i) + c_2 \exp(-\alpha_2 t_i)$$

where $\alpha_1 = o$.

Evaluating $S = \sum w_i \left[y_i - c_1 \exp(-\alpha_1 t_i) - c_2 \exp(-\alpha_2 t_i) \right]^2$

for $\alpha_1 = o$ and different values of α_2, until

we get the minimum value of S. we use this

value of α_2 and $\alpha_1 = o$ as initial values for

the general curve fitting program.

(c) Two exponentials, we have the form

$$y_i = c_1 \exp(-\alpha_1 t_i) - c_2 \exp(-\alpha_2 t_i)$$

We put $\alpha_1 = r \cos x$ and $\alpha_2 = r \sin x$, then evaluating

$$S = \sum w_i \left[y_i - c_1 \exp(-\alpha_1 t_i) - c_2 \exp(-\alpha_2 t_i) \right]^2$$

for different values of r and x ($0 < x < \pi/4$) until we obtain the minimum value of S.   Then *we use* the values of α_1 and α_2 in the general curve fitting program.

We may note that if we put $x = \pi/4$ then it is exactly the same as case (a) and if we put $x = o$, it is the same as case (b).

(d) Two exponentials plus constant term: here we have

$$y = c_1 \exp(-\alpha_1 t_i) + c_2 \exp(-\alpha_2 t_i) + c_3 \exp(-\alpha_3 t_i)$$

where $\alpha_1 = 0$, $\alpha_2 = r \cos x$ and $\alpha_3 = r \sin x$.   Then evaluating

$$S = \sum w_i \left[y_i - c_1 \exp(-\alpha_1 t_i) - c_2 \exp(-\alpha_2 t_i) - c_3 \exp(-\alpha_3 t_i) \right]^2$$

for different values of r and x and ($0 < x < \pi/4$) until we get the minimum value of S.   *We use* the estimated values of α_2, α_3 and $\alpha_1 = 0$ in the general curve fitting program.

6.3. The optimum interval and the minimum correlation program

The program generate tables of the minimum correlation coefficient between $\exp(-u_1 t_i)$ and $\exp(-u_2 t_i)$ and the corresponding optimum interval for different number of observations and two types of intervals according to the details of chapter 5.

CHAPTER 7

NUMERICAL RESULTS

The programs described in the previous chapter have been applied to data of various types, generally with success, in that the data have been adequately represented by exponentials. We will comment here on three applications, to the artificial data described by Lanczos in his criticism of exponential fitting (Lanczos, 1957), artificial data generated by Glass appropriate to an experimental situation and biological data in collaboration with Glass of this University and of the Department of Medical Physics at Hamersmith Hospital.

7.1. The artificial data generated by Lanczos

Lanczos (1957) has given an example of a set of 24 decay observations to show the numerical difficulties which may occur on account of the non-orthogonal behaviour of the exponential functions. These observations which are presented in table 17 (i), were generated from the curve

t	y (6dp)	y (5dp)	y (4dp)	y (2dp)
0.00	2.513400	2.51340	2.5134	2.51
0.05	2.044333	2.04433	2.0443	2.04
0.10	1.668404	1.66840	1.6684	1.67
0.15	1.366418	1.36642	1.3664	1.37
0.20	1.123232	1.12323	1.1232	1.12
0.25	0.926890	0.92689	0.9269	0.93
0.30	0.767934	0.76793	0.7679	0.77
0.35	0.638878	0.63888	0.6389	0.64
0.40	0.533784	0.53378	0.5338	0.53
0.45	0.447936	0.44794	0.4479	0.45
0.50	0.377585	0.37758	0.3776	0.38
0.55	0.319739	0.31974	0.3197	0.32
0.60	0.272013	0.27201	0.2720	0.27
0.65	0.232496	0.23250	0.2325	0.23
0.70	0.199659	0.19966	0.1996	0.20
0.75	0.172270	0.17227	0.1723	0.17
0.80	0.149341	0.14934	0.1493	0.15
0.85	0.130070	0.13007	0.1301	0.13
0.90	0.113812	0.11381	0.1138	0.11
0.95	0.100042	0.10004	0.1000	0.10
1.00	0.088332	0.08833	0.0883	0.09
1.05	0.078335	0.07834	0.0783	0.08
1.10	0.069767	0.06977	0.0698	0.07
1.15	0.062393	0.06239	0.0624	0.06

Table 17(i)

Lanczos data (24 observations) generated from

the ~~double~~ exponential curve

$$y(t) = 0.0951 \exp(-t) + 0.8607 \exp(-3t) + 1.5576 \exp(-5t) ,$$

with 6, 5, 4 and 2 decimal places accuracy .

$$y(t)=0.0951\exp(-t)+0.8607\exp(-3t)+1.5576\exp(-5t),$$

and are considered to be accurate to half unit of the second decimal.

The method which Lanczos has used in fitting this data is exactly the same as Prony's method which has been described in section 1.3. Lanczos could not find a three exponentials fit to his data, but he managed to fit the data to two exponentials.

$$y(t) = 2.202 \exp(-4.45t) + 0.305 \exp(-1.58t),$$

with sum squares of residual equal to 3.89×10^{-3}

In the application of the least-squares procedure to this data, it is appropriate to use unit weights because the only error is due to round-off error (section 4.4). Using the three methods of section 2.2 we were unable to determine a three exponentials solution. However we were able to fit the data to two exponentials, namely

$$y(t)=(2.105\pm.052)\exp(-(4.51\pm.06)t)+(0.403\pm.053)\exp(-(1.81\pm.11)t),$$

using one stage minimization method, with sum squares of residual equal to 1.14×10^{-4} after 5 iterations;

$y(t)=(2.105\pm.052)\exp(-(4.57\pm.06)+)+(0.403\pm.053)\exp(-(1.81\pm.11)t)$

using two stage minimization method, with sum squares

of residual equal to 1.14×10^{-4} after 4 iterations;

$y(t)=(2.105\pm.051)\exp(-(4.57\pm.06)t)+(0.403\pm.052)\exp(-(1.81\pm.11)t)$,

using Berman's method, with sum squares of residuals

equal to 1.14×10^{-4} after 4 iterations.

Our fit is better than the one obtained by

Lanczos, in the sense that the sum squares of

residuals obtained from out fitting is less than

the one given by Lanczos method.

If we take the data to greater accuracy to

6, 5 and 4 decimal places respectively (table 17 (i))

instead of the 2 decimal places of Lanczos, we are

able to obtain a three exponentials solution from

each method.   These solutions are presented in

table 17 (ii).

We may note from this table that in the two

stage minimization method (T) the convergence was

very slow and 85 iterations were needed to give

accurate solution (using data accurate to 6dp),

where the one stage minimization method (0) took 3

iterations and Berman's method (B) took 2 iterations.

	N.D.P	COEFFICIENTS			EXPONENTS			N.I
(T)	6	.0954±.0002;	.8615±.0005;	1.5565±.0006;	1.0025±.0010;	3.003±.001;	5.0011±.0004;	85;
	5	.096±.002 ;	.864±.005 ;	1.554±.006 ;	1.002±.009 ;	3.00±.01 ;	5.001±.004 ;	45;
	4	.09±.02 ;	.86±.04 ;	1.56±.06 ;	.97±.09 ;	3.0±.1 ;	5.0±.04 ;	81;
(O)	6	.0955±.0002;	.8618±.0005;	1.5562±.0006;	1.0018±.0009;	3.002±.001;	5.0008±.0004;	3;
	5	.096±.002 ;	.863±.005 ;	1.554±.006 ;	1.004±.009 ;	3.01±.01 ;	5.002±.004 ;	3;
	4	.09±.02 ;	.86±.04 ;	1.56±.06 ;	.97±.09 ;	3.0±.1 ;	5.0±.04 ;	4;
(B)	6	.0955±.0002;	.8618±.0005;	1.5562±.0006;	1.0018±.0009;	3.002±.001;	5.0008±.0004;	2;
	5	.096±.002 ;	.863±.005 ;	1.554±.006 ;	1.004±.009 ;	3.01±.01 ;	5.002±.004 ;	3;
	4	.09±.02 ;	.86±.04 ;	1.56±.06 ;	.97±.09 ;	3.0±.1 ;	5.0±.04 ;	3;

Table 17(11)

the fitting of Lanczos data (Table 17(1)) , using

two stage minimization method(T) , one stage minimization method

(0) and Berman's method (B) .

The reason for the difficulty in fitting this data is that the exponentials, exp $(-t)$, exp $(-3t)$ and exp $(-5t)$, are highly correlated. This may be seen by examination of the correlation matrix for the three functions for the values of t appropriate to Lanczos data (namely 24 values at interval 0.05), which is

$$\begin{bmatrix} 1.00000 & 0.96400 & 0.89329 \\ 0.96400 & 1.00000 & 0.97908 \\ 0.89329 & 0.97908 & 1.00000 \end{bmatrix}$$

It might be observed that the determinant of this matrix is 3.805×10^{-4} and the latent roots are 2.8914, 0.10740 and .001193, so that the matrix is obviously ill-conditioned. The smallest latent root is .001193; very approximately therefore we see that we shall lose about 3 figures in the data (the reciprocal of this smallest latent root is about 900). This agrees with the difficulty in fitting the data to 2 decimal places or at best 3 significant figures, and the accuracy (2 significant figures) in our fit of the data to 4 decimals.

Table (18)

This table is the same as Table (4) , but the number of observations here is equal to 24 .

| | | EQUAL INTERVAL | | EXPON INTERVAL | |
EX1	EX2	MININT	MINCOR	MININT	MINCOR
1	2	0.3216	0.9496	0.5490	0.9603
1	3	0.2661	0.8630	0.4301	0.9073
1	4	0.2303	0.8266	0.3589	0.8621
1	5	0.2095	0.7809	0.3097	0.8247
1	6	0.1910	0.7434	0.2725	0.7938
1	7	0.1739	0.7123	0.2464	0.7677
1	8	0.1658	0.6861	0.2225	0.7453
1	9	0.1577	0.6639	0.2066	0.7260
1	10	0.1496	0.6445	0.1914	0.7090
2	3	0.1901	0.9822	0.3227	0.9860
2	4	0.1670	0.9497	0.2745	0.9603
2	5	0.1529	0.9155	0.2424	0.9332
2	6	0.1384	0.8831	0.2165	0.9074
2	7	0.1247	0.8534	0.1980	0.8836
2	8	0.1209	0.8268	0.1806	0.8621
2	9	0.1171	0.8030	0.1677	0.8425
2	10	0.1133	0.7816	0.1580	0.8248
3	4	0.1375	0.9909	0.2301	0.9929
3	5	0.1227	0.9720	0.2052	0.9780
3	6	0.1177	0.9498	0.1850	0.9603
3	7	0.1126	0.9271	0.1687	0.9421
3	8	0.1073	0.9049	0.1574	0.9244
3	9	0.1019	0.8838	0.1474	0.9074
3	10	0.0964	0.8638	0.1378	0.8913
4	5	0.1145	0.9946	0.1791	0.9957
4	6	0.1077	0.9823	0.1635	0.9860
4	7	0.1005	0.9668	0.1515	0.9737
4	8	0.0931	0.9500	0.1408	0.9603
4	9	0.0856	0.9327	0.1303	0.9467
4	10	0.0781	0.9156	0.1219	0.9332
5	6	0.0969	0.9964	0.1488	0.9971
5	7	0.0875	0.9877	0.1371	0.9903
5	8	0.0778	0.9762	0.1256	0.9813
5	9	0.0717	0.9633	0.1187	0.9711
5	10	0.0671	0.9497	0.1131	0.9603
6	7	0.0744	0.9974	0.1239	0.9979
6	8	0.0688	0.9909	0.1169	0.9929
6	9	0.0630	0.9822	0.1111	0.9860
6	10	0.0614	0.9720	0.1060	0.9780
7	8	0.0622	0.9980	0.1101	0.9985
7	9	0.0607	0.9931	0.1048	0.9946
7	10	0.0593	0.9862	0.1001	0.9891
8	9	0.0589	0.9955	0.0993	0.9988
8	10	0.0573	0.9946	0.0947	0.9957
9	10	0.0552	0.9988	0.0896	0.9990

Table (18)

We may, for this data apply the techniques of chapter 5 to find the optimum interval. A table corresponding to the tables of chapter 5 is given here (table 18) for 24 observations. We note from the table that for the ratios 1/3, 1/5 and 3/5 the minimum correlation coefficients are 0.8830, 0.7809 and 0.9720 respectively, with corresponding intervals 0.27, 0.21 and 0.12. Of these minimum correlation coefficients, the largest, and therefore the one causing the most trouble, is corresponding to the ratio 3/5, the ratio nearest to unity; accordingly this suggests that the interval 0.12 should be chosen instead of 0.05. Even in this case, however we would expect difficulty in fitting the data, since even using the two exponential fit, $1/(1-p)$ is equal to 35 approximately, and we would therefore expect a loss in accuracy of nearly 2 significant figures.

We may note that in this case (using the optimum interval which is 0.12), the correlation matrix is

$$\begin{bmatrix} 1.00000 & 0.90648 & 0.79950 \\ 0.90648 & 1.00000 & 0.97198 \\ 0.79950 & 0.97198 & 1.00000 \end{bmatrix}$$

t	y(6dp)	y(5dp)	y(4dp)	y(2dp)
0.00	2.513400	2.51340	2.5134	2.51
0.12	1.539665	1.53966	1.5397	1.54
0.24	0.962896	0.96290	0.9629	0.96
0.36	0.616108	0.61611	0.6161	0.62
0.48	0.404072	0.40407	0.4041	0.40
0.60	0.272013	0.27201	0.2720	0.27
0.72	0.188110	0.18811	0.1881	0.19
0.84	0.133664	0.13366	0.1337	0.13
0.96	0.097547	0.09755	0.0975	0.10
1.08	0.073039	0.07304	0.0730	0.07
1.20	0.056022	0.05602	0.0560	0.06
1.32	0.043931	0.04393	0.0439	0.04
1.44	0.035142	0.03514	0.0351	0.04
1.56	0.028608	0.02861	0.0286	0.03
1.68	0.023646	0.02365	0.0236	0.02
1.80	0.019800	0.01980	0.0198	0.02
1.92	0.016760	0.01676	0.0168	0.02
2.04	0.014316	0.01432	0.0143	0.01
2.16	0.012319	0.01232	0.0123	0.01
2.28	0.010666	0.01066	0.0107	0.01
2.40	0.009280	0.00928	0.0093	0.01
2.52	0.008105	0.00811	0.0081	0.01
2.64	0.007102	0.00710	0.0071	0.01
2.76	0.006239	0.00624	0.0062	0.01

Table 19(i)

The modified Lanczos data , using the optimum
interval equals to 0.12   .

	N.D.P	COEFFICIENTS			EXPONENTS			N
(T)	6	.09509±.00001 ;	.8606±.0001 ;	1.5577±.0001 ;	.99995±.00006 ;	2.9999±.0002 ;	4.9993±.0009 ;	1
	5	.0954±.002 ;	.863±.001 ;	1.555±.002 ;	1.0010±.0007 ;	3.003±.002 ;	5.002±.001 ;	2
	4	.094±.001 ;	.86±.01 ;	1.56±.01 ;	.996±.005 ;	2.99±.02 ;	4.998±.008 ;	2
(O)	6	.09509±.00001 ;	.8606±.0001 ;	1.5577±.0001 ;	.99994±.00006 ;	2.9998±.0002 ;	4.9992±.0009 ;	2
	5	.0954±.002 ;	.863±.001 ;	1.555±.002 ;	1.0010±.0007 ;	3.003±.002 ;	5.002±.001 ;	3
	4	.094±.001 ;	.86±.01 ;	1.56±.01 ;	.996±.005 ;	2.99±.02 ;	4.998±.008 ;	3
(B)	6	.09509±.00001 ;	.8606±.0001 ;	1.5577±.0001 ;	.99994±.00006 ;	2.9998±.0002 ;	4.9992±.0009 ;	2
	5	.0954±.002 ;	.863±.001 ;	1.555±.002 ;	1.0010±.0007 ;	3.003±.002 ;	5.002±.001 ;	2
	4	.094±.001 ;	.86±.01 ;	1.56±.01 ;	.996±.006 ;	2.99±.02 ;	4.998±.008 ;	3

Table 19(ii)

The fitting of the modified Lanczos data (Table 19(i)) , using (T) , (O) and (B) methods .

It might be observed that the determinant of this matrix is 3.195×10^{-3}, with latent roots equal to 2.7872, 0.20730 and .008529, accordingly since the reciprocal of the smallest latent root is about 180, we would actually expect a loss of slightly more than 2 significant figures, in gaining approximately 1 significant figure by change of interval.

The data with 6, 5, 4 and 2 decimal places is given in table 19 (i). The three programs gave satisfactory three exponential fits for the data to 4 or more decimal places (table 19 (ii)), but again we could not find a three exponentials solution to fit the data to 2 decimal place accuracy. In this case all three programs gave the same two exponentials solution, namely

$$y(t) = (2.229 \pm .036) \exp(-(4.44 \pm .05)t) + (0.280 \pm .037) \exp(-(1.49 \pm .10)t)$$

with sum squares of residuals equal to 2.57×10^{-4} after 4 iterations in each program.

As a matter of interest, we also fitted this data using fewer observations, namely 12 observations at interval 0.17 and 10 observations at interval 0.20, both these sets of data covering the same

t	y (6dp)	y (5dp)	y (4dp)	y (2dp)
0.0	2.513400	2.51340	2.5134	2.51
0.2	1.123232	1.12323	1.1232	1.12
0.4	0.533784	0.53378	0.5338	0.53
0.6	0.272013	0.27201	0.2720	0.272
0.8	0.149341	0.14934	0.1493	0.15
1.0	0.088332	0.08833	0.0883	0.09
1.2	0.056022	0.05602	0.0560	0.06
1.4	0.037778	0.03778	0.0378	0.04
1.6	0.026806	0.02681	0.0268	0.03
1.8	0.019800	0.01980	0.0198	0.02

Table 20(i)

The modefied Lanczos data , using the optimum
interval equals to 0.2 corresponds to 10 obsevations .

Method	N.D.P	COEFFICIENTS			EXPONENTS			N.I
(T)	6	.09504±.00008	.8606±.0004	1.5578±.0005	.9997±.0004	2.9996±.0007	4.9999±.0003	3;
	5	.0951±.0004	.863±.002	1.557±.003	1.000±.002	3.000±.004	5.000±.002	4;
	4	.102±.003	.91±.02	1.50±.02	1.03±.01	3.08±.02	5.04±.01	6;
	2	2.21±.01	.30±.01		4.50±.02	1.50±.04		5;
(O)	6	.09504±.00008	.8605±.0004	1.5578±.0005	.9997±.0004	2.9996±.0007	4.9999±.0003	2;
	5	.0950±.0004	.861±.002	1.558±.003	1.000±.002	3.000±.004	5.000±.002	3;
	4	.102±.003	.91±.02	1.50±.02	1.03±.01	3.08±.02	5.04±.01	5;
	2	2.21±.01	.30±.01		4.50±.02	1.50±.04		4;
(B)	6	.09504±.00008	.8605±.0004	1.5578±.0005	.9997±.0004	2.9996±.0007	4.9999±.0003	2;
	5	.0950±.0004	.861±.002	1.558±.003	1.000±.002	3.000±.004	5.000±.002	2;
	4	.102±.003	.91±.02	1.50±.02	1.03±.01	3.08±.02	5.04±.01	3;
	2	2.21±.01	.30±.01		4.50±.02	1.50±.04		4;

Table 20(11)

The fitting of data Table 20(1), using (T), (O) and (B) methods.

t	y (6dp)	y (5dp)	y (4dp)	y (2dp)
0.00	2.513400	2.51340	2.5134	2.51
0.17	1.262821	1.26282	1.2628	1.26
0.34	0.662601	0.66260	0.6626	0.66
0.51	0.365099	0.36510	0.3651	0.36
0.68	0.212077	0.21208	0.2121	0.21
0.85	0.130070	0.13007	0.1301	0.13
1.02	0.084145	0.08414	0.0841	0.08
1.19	0.057224	0.05722	0.0572	0.06
1.36	0.040696	0.04070	0.0407	0.04
1.53	0.030072	0.03007	0.0301	0.03
1.70	0.022938	0.02294	0.0229	0.02
1.87	0.017943	0.01794	0.0179	0.02

Table 20(iii)

The modified Lanczos data , using the optimum

interval equals to 0.17 corresponds to 12 observations .

	N.D.P	COEFFICIENTS			EXPONENTS			N.I
(T)	6	.09512±.00007 ;	.8606±.0004 ;	1.5576±.0004 ;	1.0003±.0003 ;	3.0003±.0006 ;	5.0001±.0003 ;	2 ;
	5	.0952±.0007	.861±.004	1.557±.004	1.000±.003	3.001±.006	5.001±.003 ;	4 ;
	4	.098±.005	.87±.03	1.54±.04	1.01±.02	3.02±.05	5.01±.02 ;	4 ;
	2	2.20±.04	.31±.04 ;	;	4.50±.05	1.59±.10 ;	;	3 ;
(O)	6	.09514±.00007 ;	.8607±.0004 ;	1.5575±.0004 ;	1.0001±.0003 ;	3.0002±.0006 ;	5.0000±.0003 ;	2 ;
	5	.0952±.0007	.861±.004	1.557±.004	1.000±.003	3.001±.006	5.001±.003 ;	3 ;
	4	.098±.005	.87±.03	1.54±.04	1.01±.02	3.02±.05	5.01±.03 ;	4 ;
	2	2.20±.04	.31±.04 ;	;	4.50±.05	1.59±.10 ;	;	4 ;
(B)	6	.09514±.00007 ;	.8607±.0004 ;	1.5575±.0004 ;	1.0002±.0003 ;	3.0002±.0006 ;	5.0000±.0003 ;	2 ;
	5	.0952±.0007	.861±.004	1.557±.004	1.000±.003	3.001±.006	5.001±.003 ;	2 ;
	4	.098±.005	.87±.03	1.54±.04	1.01±.02	3.02±.05	5.01±.02 ;	3 ;
	2	2.20±.04	.31±.04 ;	;	4.50±.05	1.59±.10 ;	;	3 ;

Table 20(iv)

The fitting of data Table 20(iii) , using (T) ,(O) and (B) methods

range as that of Lanczos, but using a larger interval. The data and the results are given in tables 20. In both these cases, the results are more accurate than those of Lanczos, showing the importance of using the appropriate intervals.

For this particular example, however, because of the strong correlation between the exponentials, exp $(-3t)$ and exp $(-5t)$, the data will never be fitted accurately and we must expect to lose about 2 significant figures. Accordingly for data of this type the observations must be correct to 3 - 4 significant figures in order to give a solution at all accurate.

7.2. Artificial experimental data

Glass and De Garreta (1967) made a quantitive analysis of exponential curve fitting for biological applications by generating a large number of sets of artificial data with controlled random error which was derived from the formula

$$\text{Error} = \frac{\text{RND}}{2} \text{ x } \frac{p}{100} \text{ x exact value,}$$

where RND is a random normal deviate and p is the percentage error. The exact values were generated from the two exponential functions.

$$y(t) = c_1 \exp(-u_1 t) + c_2 \exp(-u_2 t),$$

where $t = 0(2)20$, for three ratios u_1/u_2 equal to $1/2$, $1/3$ and $1/4$ (the absolute values of u_2 were 0.2, 0.3, 0.4, the value of u_1 being fixed at 0.1).

The values of the coefficients were $c_1 = c_2 = 0.5$ and the values of p were 2%, 3%, 5% and 10% for each combination of u_1 and u_2. The observations were calculated by adding the error to the exact values. The data thus corresponds to the case of constant error.

Ten sets of data were fitted by the program using two stage minimization method, for each combination of the exponents and error. The data were fitted without weights and with the appropriate weighting factors (proportional to the inverse of the square of observations).

Glass and De Garreta have compared our results with results obtained from a program developed by Marquardt (1967) which is written in Fortran IV. This program was used by Glass on IBM 7090 computer, at Imperial College, London University. In this program no weights are used and it has the facility

| Weights | | Marquardt Program | Program I | |
Data error	Exponent ratio	Nil Mean parameter error %	Nil Mean parameter error %	W.F.3 Mean parameter error %
2%	4:1	9.84	10.28	6.34
3%	4:1	14.32	14.72	9.68
5%	4:1	36.38	31.98	21.66
10%	4:1	34.62	26.64	23.18
2%	3:1	10.28	8.36	13.42
3%	3:1	32.04	26.92	23.28
5%	3:1	60.54	23.48	32.72
10%	3:1	80.42	25.88	61.40
2%	2:1	84.56	42.54	36.06
3%	2:1	104.52	28.32	22.96
5%	2:1	95.80	50.32	52.34

Table (21)

Comparison between mean parameter error obtained from
Marquardt program and the ones obtained from program I (using
two stage minimization method without weights and with weights
proportional to the invers of the squares of the observations).

to estimate the errors in the fitted parameters (that time our program did not estimate these errors). A simple statistical procedure was used by Glass to estimate the parameters error by calculating the standard deviation and the coefficient variation for the estimates of each of the four parameters obtained from each set of data. Glass and De Garreta have pointed out that our results were better than those obtained by Marquardt in both cases without using weights or with weights where in the last case the effect of the weights is to reduce the parameter error by approximately one-third. The results are given in table (21).

7.3. Experimental data

The programs have been used to analyse radioactive uric acid turnover data in two compartment model. This work is described in detail elsewhere (Glass et al. 1968, Appendix A) and will be described briefly here.

The two compartments model of section 1.2. is appropriate to patients suffering from gout, the two compartments being uric acid in solution, and

deposited uric acid in crystallized form.   From
an analysis of the injected dose, it is required
to determine the number of compartments of the
model (corresponding to the number of exponents)
and to estimate the volumes of the compartments
and the turnover rates.

The analysis follows that of the two
compartment model described earlier (section 1.2).
Applying the boundary conditions that $v_2 = 0$ when
$t = o$, we find that

$$A_{21} = -A_{22}, \quad v_1 = A_{11} + A_{12},$$

and $A_{11}(\lambda_{01} + \lambda_{21} - \alpha_1) = -A_{12}(\lambda_{01} + \lambda_{21} - \alpha_2).$ (7.3.1)

We may note from the solution of (1.1.2) and
(1.1.3) that

$$\alpha_1 + \alpha_2 = \lambda_{01} + \lambda_{21} + \lambda_{12}, \qquad (7.3.2)$$

$$\alpha_1 \alpha_2 = \lambda_{01} \lambda_{12} , \qquad (7.3.3)$$

hence we obtain using (7.3.1), (7.3.2) and (7.3.3)

$$\lambda_{12} = (A_{11} \alpha_2 + A_{12} \alpha_1)/(A_{11} + A_{12}),$$

$$\lambda_{01} = \alpha_1 \alpha_2 (A_{11} + A_{12})/(A_{11} \alpha_2 + A_{12} \alpha_1),$$

$$\lambda_{21} = A_{11} A_{12} (\alpha_1 - \alpha_2)^2/(A_{11} + A_{12})(A_{11} \alpha_2 + A_{12} \alpha_1)$$

Usually the experimental data are the measures
of the concentration of the labelled substance as
a function of time of the form

$$y(t_i) = c_1 \exp(-\alpha_1 t_i) + c_2 \exp(-\alpha_2 t_i).$$

Then it is required to calculate c_1, c_2, α_1 and α_2 using an exponential curve fitting program. From these values, the parameters of the model may be calculated. These parameters are

$$\lambda_{12} = (c_1 \alpha_2 + c_2 \alpha_1)/(c_1 + c_2),$$ fraction per unit time passing from first compartment to second compartment,

$$\lambda_{01} = \alpha_1 \alpha_2 (c_1 + c_2)/(c_1 \alpha_2 + c_2 \alpha_1),$$ fraction per unit time passing out of system from first compartment,

$$\lambda_{21} = c_1 c_2 (\alpha_1 - \alpha_2)^2/(c_1 + c_2)(c_1 \alpha_2 + c_2 \alpha_1),$$

fraction per unit time passing from first compartment to second compartment,

$$V_1 = 100/(c_1 + c_2),$$ volume of first compartment, e.g. m.g.,

$$V_2 = V_1 \lambda_{21}/\lambda_1 2,$$ volume of second compartment,

and the turnover $= V_1 \lambda_{01}$, units per unit time, e.g. mg/day.

The data analyses were for twenty patients, including four normals, and are given in tables 22. In each case, the data were fitted with one (table 23 (i)) and two (table 23 (ii)) exponentials, using the two stage minimization method program which is superior to the other methods. The goodness of fit of each of the functions to the data was then

Tables 22

The data provided by Glass for twenty patients
G(1) to G(20) .

t	G(1)	G(2)	G(3)	G(4)	G(5)	G(6)
0.25	0.0719		0.08812	0.07501	0.04270	0.05593
0.50	0.0455	0.03553	0.06557	0.05743	0.02916	0.03901
0.75	0.03185	0.0174	0.04951	0.05463	0.02423	0.0296
1.25	0.0239	0.01071	0.03121	0.03637	0.01805	0.02505
1.75	0.0185	0.00731	0.02287	0.03068	0.01948	0.02228
2.25	0.01263	0.005029	0.01899	0.02042	0.01537	0.01641
2.75	0.00922	0.003294	0.01358	0.01918	0.01361	0.01027
3.50	0.00526	0.001672		0.01278	0.007108	0.005182
4.50	0.00364	0.0008133	0.00455	0.008545	0.005178	0.003418
5.50	0.001965	0.00044348		0.007243	0.00397	0.002218
6.50	0.00145	0.0002331		0.007045	0.003136	0.001365
7.50	0.000763	0.0001415		0.002914	0.002237	0.0008244
8.50	0.00051	0.00008176	0.000473	0.00199	0.001709	0.000578
9.50	0.000226	0.0000513	0.000283	0.001764	0.001453	0.0003911
10.50						

Table 22(1)

t	G(7)	G(8)	G(9)	G(10)	G(11)	G(12)
0.25	0.05592	0.04766		0.07229	0.08471	0.05321
0.75	0.04205	0.03815	0.03205	0.05462	0.04647	0.04929
1.25	0.03534	0.02685	0.02544	0.04498	0.03766	0.04327
1.75	0.02918	0.01835	0.02083	0.03499	0.02661	0.03561
2.25	0.02220	0.01489	0.01357	0.02802	0.01616	0.0244
2.75	0.02264	0.01154		0.02263	0.01129	0.01961
3.50	0.01493	0.007897	0.00791	0.01581	0.00826	0.01737
4.50	0.01170	0.004943	0.004434	0.01112	0.003611	0.01079
5.50		0.003171	0.00357	0.007711	0.001908	0.007
6.50	0.005615	0.001752	0.002636	0.005024	0.0008782	0.004626
7.50	0.003779	0.001241	0.001946	0.003525	0.0004368	0.003209
8.50	0.002246	0.0006559	0.001443	0.002398	0.0002414	0.002467
9.50	0.001556	0.0004702	0.001313	0.001753	0.0001687	0.001551
10.50			0.001004	0.001195	0.0001149	0.001016

Table 22(ii)

t	G(13)	G(14)
0.667	0.04022	0.03777
1.167	0.0316	0.03285
1.667	0.02683	0.02751
2.167	0.02328	0.02159
2.667	0.0181	0.01591
3.417	0.01195	0.01186
4.417	0.008425	0.008318
5.417	0.006253	0.005481
6.417	0.004163	0.003751
7.417	0.002843	0.002695
8.417	0.001765	0.001563
9.417	0.001192	0.001021
10.417	0.0008856	0.0007135

Table 22(iii)

t	G(15)	G(16)	G(17)	G(18)	G(19)	G(20)
0.25		0.04337	0.06837	0.07077	0.02912	0.03389
0.50	0.1004					
0.75	0.04195	0.03299	0.05089	0.07959	0.02622	0.027575
1.25	0.0275	0.02988	0.04199	0.06212	0.02542	0.025345
1.75	0.01864	0.02228	0.02951	0.06611	0.02038	0.01786
2.25	0.0146	0.01772	0.02564	0.04536	0.01685	0.015235
2.75	0.008891	0.01579	0.01935	0.04156	0.01416	0.01215
3.50	0.003544	0.01315	0.01271	0.031	0.01156	0.008502
4.5	0.001837	0.00823	0.0076/48	0.02408	0.007831	0.0046505
5.5	0.0009579	0.005757	0.004405	0.01905	0.005857	0.00269
6.5	0.0006235	0.004155	0.003353	0.01274	0.004396	0.0017035
7.5	0.0003939	0.00323	0.00206	0.01002	0.002811	0.001052
8.5	0.000208	0.002488	0.001242	0.007518	0.002038	0.0007569
9.5	0.0001245	0.001709	0.0008288	0.006179	0.001529	0.00055645
10.5		0.000889	0.0005046	0.003921	0.001164	0.0003471

Table 22 (iv)

examined using F ratio criterion (section 6.1) to determine whether a statistically significant improvement in fit was obtained by fitting a double rather than a single exponential function to the data.   The data were also fitted both with a single and a double exponential functions, with first point omitted (except $G(9)$, $G(13)$ and $G(14)$), since there were several reasons why this point may be suspected.

In the course of this investigation it was found that of major importance was the weight to be associated with each data point.   The weights depend on the sampling technique and measurement procedure (chapter 4), i.e. there are two main considerations which determine the correct weighting factor which should be used, namely the primary sampling and the measurement technique.   In this investigation there were three stages in the sampling process (Glass, personal communication, 1966);   firstly the collection of urine over fixed time period, secondly taking a fraction from each collection sample and diluting this, and thirdly, sampling from this second sample for radioactive assay purpose.   Fixed time sampling requires that

PATIENT	C	α	PROB.F
G(1)	.0737±.0037	.598±.028	3.81 ₁₀-7
G(2)	.0496±.0030	.809±.036	6.98 ₁₀-7
G(3)	.1008±.0040	.606±.025	3.98 ₁₀-7
G(4)	.0779±.0042	.395±.021	4.79 ₁₀-7
G(5)	.0400±.0021	.353±.018	4.58 ₁₀-7
G(6)	.0606±.0019	.508±.014	2.09 ₁₀-7
G(7)	.0574±.0017	.373±.012	6.61 ₁₀-7
G(8)	.0532±.0017	.539±.016	3.50 ₁₀-7
G(9)	.0426±.0032	.446±.029	2.45 ₁₀-6
G(10)	.0758±.0016	.424±.009	1.25 ₁₀-7
G(11)	.0904±.0114	.711±.080	3.23 ₁₀-7
G(12)	.0638±.0021	.393±.012	2.24 ₁₀-7
G(13)	.0518±.0011	.399±.007	2.11 ₁₀-7
G(14)	.0514±.0011	.413±.008	2.23 ₁₀-7
G(15)	.137±.011	.848±.050	1.25 ₁₀-6
G(16)	.0446±.0012	.366±.010	1.43 ₁₀-7
G(17)	.0751±.0043	.493±.026	1.12 ₁₀-7
G(18)	.0886±.0039	.286±.013	3.71 ₁₀-7
G(19)	.0341±.0008	.320±.008	1.49 ₁₀-7
G(20)	.0404±.0012	.464±.012	1.96 ₁₀-7

TABle 23(1)

Single exponential fitting

(all data points in , uising weighting factor 2)

PATIENT	C1	C2	$\alpha 1$	$\alpha 2$	PROB.F
G(1)	.04124±.0035	.0517±.0039	1.84±.33	.495±.016	3.57 $_{10-5}$
G(2)	.0379±.0025	.0244±.0033	1.53±.16	.610±.024	3.12 $_{10-5}$
G(3)	.040±029	.067±.030	1.08±.51	.506±.069	2.8 $_{10-1}$
G(4)	.042±.029	.044±.031	.75±.39	.303±.078	1.28 $_{10-1}$
G(5)	.032±.042	.036±.002	4.9±5.6	.33±.078	3.21 $_{10-2}$
G(6)	.0019	.588	.7799	.5039	1.29 $_{10-2}$

NO STE : ELEMENS OF MATRIX ARE NEGATIVE

PATIENT	C1	C2	$\alpha 1$	$\alpha 2$	PROB.F
G(7)	.039±.433	.055±.002	7.7±45.1	.362±.013	1.66 $_{10-1}$
G(8)	.033±.04	.025±.014	.446±.057	.93±.32	2.19 $_{10-2}$
G(9)	.047±.003	.0032±.0032	.619±.073	.111±.100	1.55 $_{10-3}$
G(10)	.033±.010	.048±.011	.76±.15	.35±.24	1.11 $_{10-4}$
G(11)	.090±.011	---------	.711±.080	--------	6.88 $_{10-6}$
G(12)	-.036±.080	.068±.003	5.9±9.3	.410±.014	8.44 $_{10-2}$
G(13)	.0024	.0513	2.137	.397	5.36 $_{10-2}$

NO STE : ELEMENTS OF MATRIX ARE NEGATIVE

PATIENT	C1	C2	$\alpha 1$	$\alpha 2$	PROB.F
G(14)	.049±.063	.025±.064	.42±.14	.28±1.3	4.56 $_{10-2}$
G(15)	.169±.057	.089±.008	3.03±.85	.683±.026	1.14 $_{10-4}$
G(16)	.045±.001	---------	.366±.010	---------	1.65 $_{10-7}$
G(17)	.075±.004	---------	.493±.026	---------	5.99 $_{10-7}$
G(18)	.886±.004	--------	.286±.03	---------	3.71 $_{10-7}$
G(19)	-.009±.002	.037±.001	2.36±.98	.337±.008	6.22 $_{10-3}$
G(20)	-.012±.005	.048±.005	1.53±.75	.502±.024	3.45 $_{10-2}$

Table 23(ii)

Double exponential fitting

(all data points in , using weighting factor 2)

weighting factor (2), i.e each weight is inversely proportional to the observation should be used (section 4.3). The final (third stage) sampling was effectively a fixed count procedure (section 4.2) and for this data weights inversely proportional to the square of the observation should be used.

However three samples were analysed from each sample of urine at this final stage. These three samples were in close agreement and the error of sampling at this stage was clearly small. The major source of error was therefore in the primary collection of urine, and accordingly the data were analysed using weights inversely proportional to the observations.

Our results were compared by Glass with the clinical findings on each patient and the correlation between the existence of two components and the presence of gout was examined. The data were also examined for comparison using weights inversely proportional to the square of the observations (fixed counts sampling); but the results from this did not correlate as well as with clinical diagnosis.

The results from fitting the data where the first point was omitted, also did not correlate with clinical diagnosis. This means that early sampling was important.

A program which was developed by Berman (1962) for simulation and analysis of biological models which also allowed it to be used for exponential curve fitting was then used by Glass to analyse the data in terms of one compartment model, and of a two compartment model in those cases where the two components had been indicated by our results. Berman's program provides the estimates of the parameter of the model which are the turnover rates and pool sizes and also it provides the error in these parameters and it has the possibility of using weights. The program is written in Fortran II and it was run by Glass on the IBM 7090 computer at Imperial College, London University.

A comparison between our results and those obtained from Berman's program has been made (Glass et al. 1968, Appendix A). This indicates that our results of the two exponential components were more

compatible with the clinical diagnosis of gout
than those obtained using Berman's program.

7.4. Conclusion

The work of this thesis enables exponential
curve fitting to be satisfactorarily carried out
whenever possible.   From estimates of parameters,
however rough, the optimum interval of tabulations
may be found in the manner of chapter 5, and an
estimate of the accuracy of the fit obtained.   If
this estimate shows that the parameters can be
determined to sufficient accuracy, then the programs
described earlier can be used to satisfactorarily
determine the parameters and give accurate measures
of their errors.

REFERENCES

AITKEN, A.C., 8th. ed. (1962). Statistical mathematics, Oliver and Boyed, Edinburgh.

BERMAN, M., and SCHOENFELD, R. (1956). Invariants in experimental data on linear kinetics and the formulation of models, J. Appl. Phys., 27, 1361.

BERMAN, M., SHAN, E., and WEISS, M.F. (1962). The routine fitting of kinetics data to models : a mathematical formulation for digital computers, Biophys. J., 2, 275.

CORNFIELD, J., STEINFELD, J., and GREENHOUSE, S.W. (1960). Models for interpretation of experiments using tracer compounds, Biometrics, 16, 212.

DE LA GARZA, A. (1954). Spacing of information in polynomial regression, Ann. Math. Statist, 25, 123.

DRAPER, N.R., and SMITH, H. (1966). Applied regression analysis, John Wiley and Sons, Inc., New York.

FOX, L., and MAYERS, D.F. (1968). Computing methods for scientists and engineers, Clarendon Press, Oxford.

FRASER, D.A.S. (1958). Statistics : an introduction, John Wiley and Sons, Inc., New York.

FROBERG, C.E. (1964). Introduction to numerical analysis, Addison-Wesley Publishing Co., Inc., London.

GUEST, P.G. (1958). The spacing of observations in polynomial regression, Ann. Math. Statist., 29, 294.

GUEST, P.G. (1961). Numerical methods of curve fitting, Cambridge University Press.

GLASS, H.I., and DE GARRETA, A.C. (1967). Quantitative analysis of exponential curve fitting for biological applications, Phy. Med. Biol., 12, 379.

GLASS, H.I., and NORDIN, B.E.C. (1963). The analysis of radio-isotope clearance data obtained using bone seeking isotope, Phys. Med. Biol., 8, 387.

GLASS, H.I., SCOTT, J.T., GILLES, D.C., ARNOT, R.N., HOLLOWAY, V., AND BEN HAMEID, S.S. (1968). The interpretation of radioactive uric acid turnover data. (Appendix A).

HARTREE, D.R., 2nd. ed. (1958). Numerical analysis, Oxford University Press.

HILDERBRAND, F.B. (1956). Introduction to numerical analysis, McGraw-Hill, New York.

HOEL, P.G. (1958). Efficiency problems in polynomial estimation, Ann. Math. Statist., 29, 1134.

LANCZOS, C. (1957). Applied analysis, Sir Isaac Pitman and Sons, Inc., New York.

MARQUARDT, D.W. (1964). Least squares estimation of non-linear parameters, a computer program I.B.M. SHARE library No. 3094.

MODERN COMPUTING METHODS. (1961). Notes on applied science, No. 16, H.M. Stationary Office, London.

PEARSON, E.S., and HARLEY, H.O., ed. (1954). Biometrika tables for statistician, Cambridge University Press.

PERL, W. (1960). A method for curve fitting by exponential functions, Int. J. Appl. Radiat., 8, 211.

PLACKETT, R.L. (1960). Principles of regression analysis, Clarendon Press, Oxford.

ROBERTSON, J.S. (1962). Mathematical treatment of uptake and release of indicator substances in relation to flow analysis in tissues and organs, Handbook of Physiology, Section 2: Circulation, American Physiological Society, $\underline{1}$, 617.

SCHEFFE, H. (1958). The analysis of variance, John Wiley and Sons, Inc., New York.

SMILLIE, K.W., and ANSTEY, T.H. (1964). A note on the calculation of probabilities in an F-Distribution, A.C.M., $\underline{7}$, 725.

WALSH, J., ed. (1966). Numerical analysis : an introduction, Academic Press, London.

WILKINSON, J.H. (1963). Rounding errors in algebraic processes, Notes on applied science, No. 32, H.M. Stationary Office, London.

WILLERS, F.A. (1948). Practical analysis : graphical and numerical methods, Dover Publications, Inc., New York.

APPENDIX   A

The Interpretation of Radioactive Uric Acid Turnover Data

Radioaktive Isotope in Klinik und Forschung

Band VIII

Vorträge am Gasteiner internationalen Symposion 1968

Herausgegeben von

Professor Dr. K. Fellinger und Doz. Dr. R. Höfer

Departments of Medical Physics and Rheumatology
Royal Postgraduate Medical School, Hammersmith Hospital,
and
Computer Department, Glasgow University

The Interpretation of Radioactive Uric Acid Turnover Data

By

H. I. Glass, J. T. Scott, D. C. Gilles, R. N. Arnot,
V. P. Holloway and S. S. Ben Hameid

Abstract

In certain cases, a mathematically significant improvement in fit was obtained by describing the urine radioactive uric acid clearance curves with a double exponential rather than a mono-exponential funtion. This optimum fit was determined by significance tests carried out on results obtained by fitting one and two exponentials to the same data using the Gilles exponential curve fitting programme. A model is proposed suggesting that the uric acid may also exist in a second pool, or physiological form. The parameters of such a model, the turnover rates and pool sizes, and their errors have been calculated using the Berman programme in a group of twenty patients, including four normals. The values obtained by compartmental analysis agree with the results quoted by other workers, however, the normal turnover rate appears to be about 15 % higher than that found by other workers. In two cases of tophaceous gout, the second pool amounted to approximately 20 % of the total uric acid pool. The accuracy with which the pool size and turnover rate may be estimated from serum measurements, has also been investigated.

Extrait

Dans certains cas, une amélioration mathématique significative dans l'ajustement des courbes a été obtenue lorsque l'on décrit la courbe de clearance de l'acide urique radioactif par une fonction à deux exponentielles plutôt qu'à une seule exponentielle. Cet ajustement optimum a été obtenu par des tests significatifs en traitant les résultats obtenus par ajustement d'une ou de deux exponentielles aux mêmes valeurs, en utilisant le programme d'ajustement des courbes exponentielles de Gilles. Un modèle est proposé, qui suggère que l'acide urique peut aussi exister dans un second compartiment ayant une signification physiologique. Les paramètres d'un tel modèle, les taux de renouvellement et la taille des compartiments, ainsi que leurs erreurs ont été calculés avec le programme de Berman chez un groupe de 20 patients, comprenant 4 normaux. Les valeurs obtenues par analyse compartimentale s'accordent avec les résultats cités par d'autres auteurs, cependant les taux de renouvellement apparemment comme étant 15 % plus élevés (que ceux trouvés par d'autres auteurs). Dans deux cas de goutte tophique le deuxième compartiment atteignait approximativement 20 % du compartiment total d'acide urique. La précision avec laquelle la taille du compartiment et la constante de renouvellement peuvent être estimées, à partir des mesures dans le sérum, a également été étudiée.

Auszug

In einer Reihe von Fällen hat es sich gezeigt, daß die Clearancekurve radioaktiver Harnsäure im Harn durch eine doppelte Exponentialfunktion eher zu erklären wäre als durch eine monoexponentielle Funktion. Die optimale Analyse einer solchen Kurve wurde mit Hilfe von Signifikanztesten durchgeführt, wobei unter Verwendung des Programmes nach GILLES einmal eine Exponentialfunktion und dann zwei Exponentialfunktionen an den gleichen experimentellen Daten angewendet wurden. Es wird nunmehr ein mathematisches Modell vorgeschlagen, in dem Harnsäure noch in einem zweiten Pool, bzw. in einer zweiten physiologischen Form vorliegen kann. Die Parameter eines solchen Modelles, Umsatzraten, Poolgrößen und deren Fehler wurden unter Verwendung des BERMAN Programmes in einer Gruppe von 20 Patienten, unter denen sich auch 4 Normale befanden, berechnet. Die Werte, die durch die Kompartement-Analyse erhalten wurden, stimmten mit Ausnahme der Umsatzrate, die etwa 15 % höher liegt, mit den Ergebnissen anderer Autoren überein. In 2 Fällen mit Gichttophi betrug dieser zweite Harnsäurepool etwa 20 % des Gesamtharnsäurepools. Schließlich wurde auch die Genauigkeit mit der Poolgröße und Umsatzraten aus Messungen der Serumaktivität bestimmt werden können, untersucht.

Introduction

Labelled uric acid can be used to estimate the pool size and turnover of uric acid in man. BENEDICT et al. (1949), GEREN et al. (1950), BISHOP et al. (1951) and SEEGMILLER et al. (1961) have used nitrogen-15 labelled uric acid. Carbon-14 labelled uric acid has been used by SORENSEN (1960) for the same purpose. A curve is obtained by plotting the relative specific activity of uric acid in the urine against time and the pool size is usually estimated by extrapolating the curve to zero time. BISHOP indicated that several of his curves, and those obtained by BENEDICT, were not straight when plotted on semilogarithmic paper. This finding was confirmed by SORENSEN. BISHOP analysed the curves empirically into two exponential components in order to extrapolate accurately to zero time. SORENSEN (1962) proposed a two compartment system to account for this phenomenon and pointed out its association with a patient suffering from tophaceous gout. The observation by BISHOP of curved rather than straight clearance curves was not confined to patients with tophaceous gout. These observations posed several problems, not the least of which is to determine the optimum method of analysing the data. If a mathematically significant improvement in fit is obtained by analysing the data in terms of a double rather than a single exponential mathematical function, a method of estimating pool size and turnover rate in this situation is required. The correlation between the presence of uric acid in more than one "pool" or physiological form and the clinical findings was used in this investigation to assess various analytical procedures. The problems of careful urine collection pose many difficulties and the possibility of carrying out turnover studies from serum samples only has been investigated.

Methods

100 µc of C-14 labelled uric acid, of specific activity between 0.05 mc/mg and 0.12 mc/mg, were completely dissolved in 100 ml of isotonic saline by heating but the solution was not allowed to boil. After cooling, 50 ml were injected into a sealed sterile bottle through a 0.22 µ millipore filter and the solution was used immediately. Just after injection, the uric acid content of the dose solution was measured by a slightly modified version of the PRAETORIUS method (1949) and standards for urine counting were prepared as described below.

The patients were kept on a low purine, low protein diet for one week prior to the injection and throughout the experimental period.

A blood sample, to be used for background counting, was taken and the patient voided urine. 30 µc in 30 ml were injected intravenously over a period of ten minutes.

Blood samples were taken at increasing intervals over an eleven day period. Serum was separated, assayed for uric acid content by the PRAETORIUS method, and frozen at −20° C.

Urine was collected under toluene in six twelve-hour periods followed by eight twenty-four-hour periods, care being taken to ensure that each patient emptied his bladder at the end of each collection period. The importance of this point will be discussed below. The uric acid content of each urine collection was measured, and uric acid was isolated for radioactive assay.

Assay of ¹⁴C Uric Acid in Urine

Two standards were prepared by diluting 1.0 ml of the dose solution to 100 ml, then adding 5 and 15 ml of the ¹/₁₀₀ solution to 250 mg of inactive uric acid which had been dissolved in 10 ml of 0.73 % (w/v) lithium carbonate, and diluted to 35 ml with distilled water. The tubes were placed in a boiling water bath and the uric acid reprecipitated by the dropwise addition of 2N acetic acid. After cooling to room temperature, the precipitated uric acid was collected in a sintered glass funnel and washed with distilled water until the eluate was neutral. The crystals were dried overnight at 80° C and stored in a desiccator until assayed. Any breakdown products of uric acid in the dose solution which were produced during preparations, were eliminated by this procedure. Immediate preparation of the standards ensured that no significant breakdown of the uric acid could occur in the dose solution since the time of the injection.

Urine samples of known volumes increasing from 10 to 200 ml over the period of investigation were removed from the twelve and twenty-four hour collections. To these samples 250 mg of inactive uric acid were added and the samples were then processed using the method described by SORENSEN (1960).

Approximately 25 mg of the pure uric acid extracted by this method were weighed in the counting vials, and after gently tapping the vials to break up any accumulations of uric acid crystals, 20 ml of scintillator gel were added by pipette (HELF et al., 1960). The vials were shaken vigorously to disperse the suspension evenly, cooled at + 4° C and counted in an automatic liquid scintillator counter. Duplicates were made of at least four samples. Two 25 mg samples of each standard were prepared in the same way, and two background samples of 20 ml of the gel were used. A total of 30,000 counts were obtained from the majority of samples, the lowest number of counts being 7,000 when the sample to background ratio was 7:1.

The percentage of the injected dose contained in each standard sample and the weight of uric acid originating from urine in each sample was calculated. The percentage of the dose/mg of uric acid in each urine collection was then estimated and the results plotted against the midtime of the collection period (Figures 1 and 2).

Assay of ¹⁴C in Serum

A standard solution was made by diluting 1.0 ml of the dose solution to 500 ml. Duplicate serum, standard and background samples were prepared by adding 1.0 ml of serum, standard solution and inactive serum, respectively, to 10 ml of scintillator (BRAY, 1960). The samples were shaken and kept for three hours to allow the protein precipitate to settle. They were then cooled for twenty minutes at + 4° C, and counted in an automatic liquid scintillation counter. 10,000 counts were obtained in samples taken up to two days, and the total counts decreased to a minimum of 1,000 for the
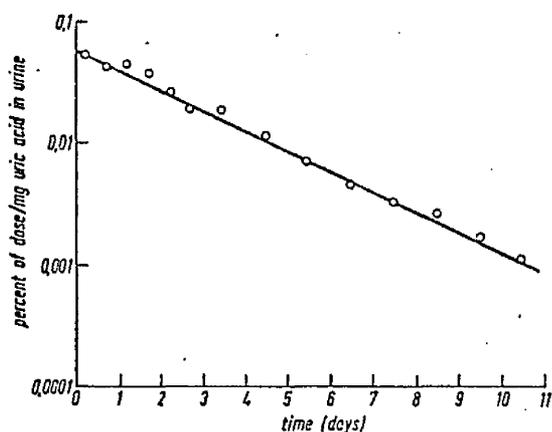
Fig. 1. Graph showing percentage of the injected dose per mg. of uric acid in urine, plotted against the mid-time of each collection period. These data were fitted by one exponential.
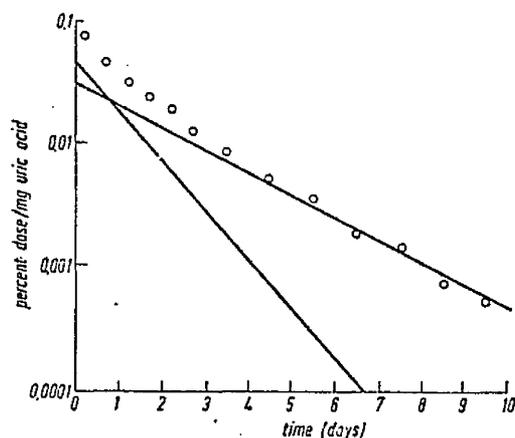

Fig. 2. Graph showing percentage of the injected dose per mg. of uric acid in urine, plotted against the mid-time of each collection period. These data were fitted by two exponentials.
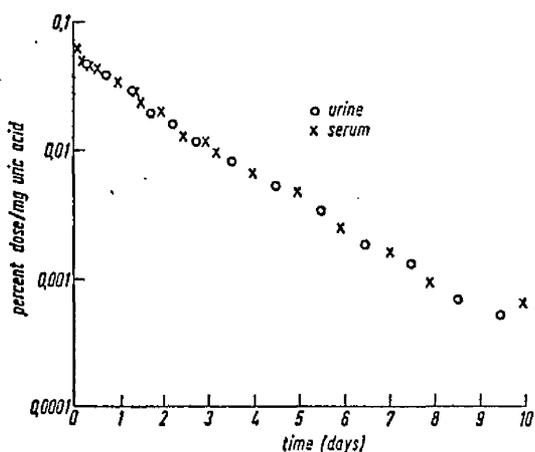

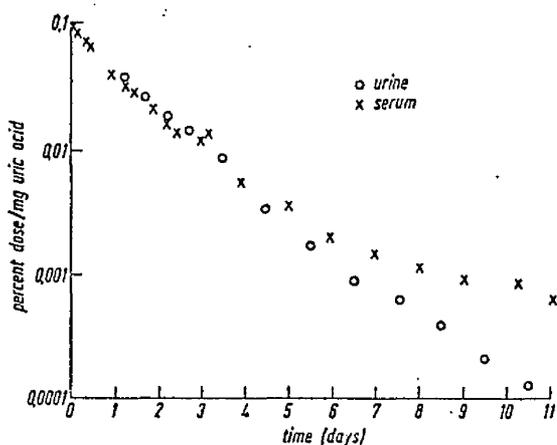Fig. 3. Graph showing good agreement between urine and serum measurement.

Fig. 4. Graph showing the greatest disagreement obtained between urine and serum measurements. The increased concentration of activity in serum samples towards the end of the investigation period could be due to re-appearance of C-14 in breakdown products of uric acid.

last sample. Internal standards were added and the samples were reshaken, allowed to stand for three hours, and cooled before recounting.

The percentage of the dose/mg of uric acid in each serum sample was calculated and the results plotted against the time of sampling (Figures 3 and 4).

Data Analysis

Two exponential curve fitting programmes with facilities for applying different weighting factors were available for the data analysis. One of these was developed by GILLES and BEN HAMEID (1966) for use on the KDF9 computer (Computer Department, Glasgow University). This programme, which is written in Algol, is a least squares iterative procedure based on the Newton-Raphson method. The second programme used was that devised by BERMAN et al. (1962) for simulation and analysis of

biological models, which also allowed it to be used for exponential curve fitting. An advantage of this programme over the Gilles programme is that error estimates of the curve parameters are also obtained. The BERMAN programme is written in Fortran II, and was run on an I.B.M. 7090 computer at Imperial College, London University.

There were three specific points of interest in the urine data analysis problem. The first of these concerns the choice of curve fitting programme. The data were therefore submitted to both programmes, using initially a weighting factor on each data point inversely proportional to the square of the observation. In each case the data were fitted both with a single and a double exponential function. The goodness of fit of each of the functions to the data was then examined using an F ratio criterion to determine whether a statistically significant improvement in fit was obtained by fitting a double rather than a single exponential function to the data.

The second point concerns the weighting factor to be applied. The data were then re-examined using both programmes and substituting a weighting factor proportional to the reciprocal of the observation, since the primary sampling procedure consisted in collecting samples over a fixed time, and therefore such a weighting factor may be more relevent to this situation than the one used previously.

Thirdly, using the weighting factor proportional to the inverse of the observation, the data were re-examined leaving out the first data point. There were several reasons why this data point may be suspect. The initial sampling period was over the first twelve hours after the injection. If the patient's bladder was not completely empty at the time of the injection of labelled uric acid, the initial concentration may be too low. If the time for complete mixing in the first pool, assuming more than one does exist, is slow, then the measured urine concentration may be too high. Finally the points on the clearance curve have been associated with the mid-time of the sampling period. In the most rapidly changing curve examined, this plotting procedure would have introduced a 4 % error in the time associated with the first data point.

The results were compared with the clinical findings on each patient, and the correlation between the existance of two components and the presence of gout was examined.

The BERMAN programme was then used to analyse the data in terms of a one compartment model, and of a two compartment model in those cases where two components had been indicated by curve analysis using the GILLES programme and a weighting factor inversely proportional to the observation (see below). This analysis provided estimates of the parameters of the model and the errors in these parameters. This was a model fitting as opposed to an exponential curve fitting procedure. The error in the data due to sampling assay and manipulative procedures has been estimated to be ± 5 %. The estimate was calculated from the computer derived errors in the curve parameters in those cases which were unequivocally found to be monoexponential. Since the average value of the ratio of the exponents in those cases where two exponentials were found, was greater than 4.5 : 1, and the data accuracy was of the order of ± 5 %, acceptable errors may be expected in the derived parameters with the number of data points used in these investigations (GLASS and DE GARETTA, 1967).

Results

The results of the mathematical analysis are summarised in Table 1. The comparison between the two computer programmes indicates that results compatible with the premise of associating two exponential components with the clinical diagnosis of gout were more apparent from the analysis of the data carried out by the GILLES programme than that obtained using the BERMAN programme (Table 1, columns 3 and 4). When

15*

the weighting factor was changed to one inversely proportional to the observation, this correlation increased (Table 1, columns 2 and 3, 4 and 5) but more so in the case of the GILLES programme results. When the results were re-examined with the first data point omitted, the correlation decreased. The importance of the early samples is apparent.

Although the correlation between the diagnosis of gout and the detection of the two exponential functions in the data does not hold absolutely throughout the series, the data from nine out of the thirteen untreated gout patients were better fitted by a double exponential function. Of the remaining two patients who yielded data which were fitted with two exponentials, the data from one of these, number 6, yielded one exponential only if the first sample point was not included in the analysis. This was a normal subject.

Table 1: Number of Components Derived from Data

Patient Number	Berman all points in W.F. ($1/Y$)	Berman all points in W.F. ($1/Y^2$)	Gilles all points in W.F. ($1/Y$)	Gilles all points in W.F. ($1/Y^2$)	Gilles 1st point omitted W.F. ($1/Y$)	Diagnosis**
3	1	1	1	1	1	Normal (Rheumatoid Arthritis
11	1	2	1	1	1	Normal (Rheumatic Fever)
6	1	1	2	2*	1	Normal (Myocardial Infarct)
15	—	—	—	—	1	Normal (Ankylosing Spondilitis)
18	1	1	1	1	1	Gout on allopurinol
7	1	1	1	1*	2	Hypertension and Hyper- uricaemia
10	2	2	2	2	2	Hypertension
4	1	1	1	1	1	Gout, non tophaceous and Hypertension
12	1	1	1	1	1	Gout, non tophaceous and Hypertension
17	1	1	1	1	1	Gout, non tophaceous
13	1	1	1	2	2	Gout, non tophaceous
1	2	1	2	2	2	Gout, non tophaceous
16	1	1	1	1	1	Gout, tophaceous and Hypertension
19	1	1	1	2*	1	Gout, tophaceous and Hypertension
14	1	1	1	2	2	Gout, tophaceous and Hypertension
2	2	2	2	2	2	Gout, tophaceous and Hypertension
8	1	1	2	2	2	Gout, tophaceous
5	1	1	2	2*	1	Gout, tophaceous
20	1	1	1	2*	1	Gout, tophaceous
9	—	—	2	2	2	Gout, tophaceous

The total exchangeable pools and fractional turnover rates, calculated by the BERMAN programme, are presented in Table 2. A two compartment model was used whenever an improved fit had been obtained with two exponentials by the GILLES programme with all data points and a weighting factor inversely proportional to the observation. However, in seven cases, this meant that the BERMAN programme was using data in which it had not indicated an improvement of fit with two components. In five of these cases, the parameter error estimates were, not surprisingly, excessive. For these five cases, the value of pool size and fractional turnover rate obtained from the single compartment model were inserted in the table. The turnover rate (mg/day) was calculated from the fractional turnover rate and the pool size. When two compartment analysis was used, the size of the first pool only was used to estimate turnover rate.

Table 2: Total Exchangeable Uric Acid Pool and Turnover Rate

Patient Number	Total Exchangeable Pool (mg)	Turnover Rate (Day⁻¹)	Turnover Rate (mg/day)	Diagnosis
3	992 ± 37	.607 ± .023	602	Normal (Rheumatoid Arthritis)
11	1081 ± 41	.735 ± .026	795	Normal (Rheumatic Fever)
6	1650 ± 47	.508 ± .014	838	Normal (Myocardial Infarct)
15	1019 ± 26	.709 ± .018	723	Normal (Ankylosing Spondylitis)
18	1031 ± 36	.305 ± .010	314	Gout on allopurinol
7	1742 ± 52	.373 ± .011	650	Hypertension and Hyperuricaemia
10	1362 ± 18	.450 ± .006	552	Hypertension
4	1284 ± 64	.394 ± .019	506	Gout, non tophaceous and Hypertension
12	1569 ± 49	.393 ± .012	616	Gout, non tophaceous and Hypertension
17	1324 ± 22	.494 ± .008	659	Gout, non tophaceous
13*	1928 ± 34	.400 ± .007	771	Gout, non tophaceous
1	1388 ± 20	.733 ± .024	789	Gout, non tophaceous
16	2282 ± 53	.361 ± .008	824	Gout, tophaceous and Hypertension
19*+	2933 ± 71	.320 ± .008	939	Gout, tophaceous and Hypertension
14*	1943 ± 35	.413 ± .007	802	Gout, tophaceous and Hypertension
2	1948 ± 23	.961 ± .027	1542	Gout, tophaceous and Hypertension
8	1933 ± 49	.576 ± .022	1000	Gout, tophaceous
5	2471 ± 138	.562 ± .267	861	Gout, tophaceous
20*+	2475 ± 68	.464 ± .012	1149	Gout, tophaceous
9	3199 ± 1477	.480 ± .051	948	Gout, tophaceous

* The number of components depends on the inclusion of the first data point.
** In this report, all patients with clinical or radiological first evidence of urate deposition have been classified as having tophaceous gout.
* Owing to excessive parameter error estimates, pool and turnover estimates were obtained from the single exponential analysis.
+ These patients would have a single pool if the first point were omitted.

The patients have been divided into five groups and the average pool sizes and turnover rates have been calculated from the values in Table 2 and presented in Table 3 (the association of high turnover rates with the presence of tophi is considered to be coincidental). The range in each group is also given, although because of the small number of patients in each group these must be interpreted with caution. The larger than normal pool sizes in the gout patients is apparent. The pool size of the gout patient on treatment with allopurinol is within the normal range, but the turnover rate is well below normal.

Table 3: Summary of Total Exchangeable Pools and Turnover Rates

	Number in Group	Total Exchangeable Pool (mg)		Turnover Rate (mg/day)	
		Average	Range	Average	Range
Normals	4	1186	992 — 1650	740	723 — 838
Gout on allopurinol	1	1031	—	314	—
Hypertension	2	1552	1362, 1742	601	552, 650
Gout, non tophaceous	5	1499	1284 — 1928	668	506 — 789
Gout, tophaceous	8	2398	1933 — 3199	1008	802 — 1542

On comparison with the values of SEEGMILLER et al. (1961), the pool size for normals is slightly higher (1186 mg compared with 1071 mg). The pool size in one patient with hyperuricaemia (No. 7) is higher than normal which also agrees with SEEGMILLER's findings on three patients. The average pool size in the gout group who have normal turnover rates is 1499 compared with 1400 mg obtained by SEEGMILLER, and in the gout group with high turnover rates the average pool sizes are again comparable (2398 mg compared with 2450 mg). In this latter group the turnover rate was also comparable but slightly lower than SEEGMILLER's results (1008 mg/day compared with 1191 mg/day). The value obtained for the turnover rate in four normals is 740 mg/day which is close to an average value for three normals of 758 mg/day quoted by BENEDICT et al. (1949). Both these figures are higher than the average normal value of 622 mg/day measured by SEEGMILLER using ^{15}N labelled uric acid and the values of 671 mg/day measured by SORENSEN (1963). The reason for this difference is not apparent.

In the data from five patients the GILLES programme indicated that two exponentials best fitted the data and the model parameter error estimates obtained by the BERMAN programme were not considered excessive. In these five patients, the presence of two pools is considered highly probable. The two compartments' sizes, the transfer rates between the compartments, and the errors in these qantities, are presented in Table 4. In the case of the three patients with tophaceous gout, the second pools represent 10 %, 18 % and 38 % of the total uric acid pool. In the other two patients the second pool represents 10 % and 22% of the total.

Table 4: Two Compartment Model Pool Sizes, Turnover and Transfer Rates
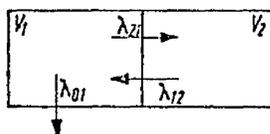
Patient	Pool Size (V_1) (mg)	Pool Size (V_2) (mg)	Turnover Rate ($\lambda_{01} \cdot V_1$) mg/day	Transfer Rate between Compartments ($\lambda_{12} V_2, \lambda_{21} V_1$) mg/day
10	1227 ± 17	135 ± 15	522 ± 11	77 ± 21
8	1736 ± 66	197 ± 64	1000 ± 55	148 ± 114
1	1076 ± 35	312 ± 32	789 ± 38	390 ± 101
2	1606 ± 46	342 ± 35	1542 ± 70	329 ± 76
9	1975 ± 208	1224 ± 1283	948 ± 115	195 ± 39

The possibility of obtaining reliable estimates of pool size and turnover rates from radioactivity measurements in the serum was investigated in thirteen patients and the results are compared with the results obtained from the calculations based on all the urine data in each patient, in Table 5. The serum concentration measured in samples taken between days 2 and 9 inclusive were used. These limits were taken to minimize any second pool effects by eliminating the values taken before day 2, and to minimize the effect due to the appearance of breakdown products of uric acid in the serum by ignoring data taken after day 9. This criterion was evolved after inspection of the serum clearance curves. A monoexponential function was fitted to this data and the pool size and turnover rates were estimated.

Table 5: Comparison of Pool Size and Turnover Estimated from Urine and Serum

Patient Number	Pool Size (mg)		Turnover (mg/day)	
	Urine	*Serum*	*Urine*	*Serum*
11	1081	975	795	906
18	1031	1474	314	392
10	1362	1446	552	665
7	1742	1455	650	611
12	1569	1570	616	631
16	2282	2689	824	789
19	2933	3092	939	1014
17	1324	1804	659	837
13	1928	2421	771	968
14	1943	3044	802	893
8	1933	1668	1000	938
20	2475	3084	1149	1545
9	3199	4006	948	1735
	Standard Error of Estimate		Standard Error of Estimate	
	$= 12.3\%$ on serum measurement		$= 19.6\%$ on serum measurement	

There is a very high correlation (0.96) between the same quantities estimated by the different methods but the standard error of the estimates are 12.3 % and 19.6 % for pool size and turnover rate respectively. This would imply that although the serum method can probably be used to obtain useful estimates of the quantities in large scale population studies, the application of the serum results in a particular individual is less useful than a study performed by analysis of urine samples.

Discussion

Obvious difficulties arise in interpreting the radioactive uric acid clearance data in terms of pool size and turnover rates when the clearance curve is not monoexponential. In general, attempting to treat such data as a monoexponential function will yield an overestimate of the first pool size and turnover rate. In most cases no significant errors will be introduced but in cases associated with large pools and high turnover rates, the errors may be 20 % or more. This is the difference in the estimate of both pool size and turnover rate if the data from the patient 2, for example, are analysed first as a monoexponential function and then as a double exponential function.

During the course of this investigation, several factors which are of importance in analysing this type of data emerged. The first point concerned the weighting factor to be associated with each data point prior to curve analysis. The weighting factor to

be used depends on the sampling technique and measurement procedure. That is to say, there are two main considerations which determine the correct weighting factor which should be used. One is in the primary sampling and the other in the measurement techniques and manipulation. There were three stages in the sampling process in this investigation; firstly, the collection of urine over a fixed time period, secondly, taking a fraction from each collection sample, and thirdly, taking a fraction of the second samples for radioactive assay purposes. Fixed time sampling requires that a weighting factor inversely proportional to the observation should be applied. The data were analysed using this weighting factor. The data were also examined, for comparison, using a weighting factor inversely proportional to the square of the specific activity, and as shown this yielded results which gave less correlation with the clinical diagnosis. The effect of the secondary sampling procedure is probably such as to make the correct weighting factor which should be applied inversely proportional to the observation raised to a power whose value is between unity and 1.5. It should be stressed that the choice of the correct weighting factor is independent of the final method of assessing the results.

The method of obtaining urine samples by collection over fixed time periods is an important consideration when the results are to be examined by exponential curve analysis. The concentration of radioactive uric acid in the urine collection represents the average concentration over the whole period since the last occasion on which the bladder was emptied. If the concentration is to be associated with a particular time value, it is important that the bladder should be emptied at the end of each sampling period. If this is not done then the time to which each concentration estimate refers may be subject to a comparatively large error especially during the early part of the investigation. One further more elementary point concerns assigning the time value to the mid-point of the collection period. This assumes a linear function, and it would be more correct to select a time corresponding to the average value of the function during this period. This error is small and corresponds to an error of only 4 % in the time associated with the first data point, in the case of the most rapidly changing urine concentration curve observed.

The question still remains whether the initial curvature of the urine clearance data is due to a slow mixing effect or a second pool or possibly both of these. The effect of slow mixing would be most apparent on the first data point, for which, as mentioned previously, there are other grounds for suspecting its reliability. However, in none of the five cases in which it is suggested above that it is highly probable that a second compartment exists, was this result dependent on the inclusion of the first data point in the analysis. This would imply that even if mixing in the first compartment was not complete for twelve hours, the final result suggesting the presence of a second compartment, would have been unaffected. It seems unlikely that the immediate uric acid pool would be associated with a physiological pool whose mixing time would greatly exceed twelve hours. The decision to include the first data point was based on the greater correlation obtained when results were compared with the clinical findings.

The final outstanding problem posed by this investigation is associated with the choice of the curve fitting programme. The difference in the mathematical methods used in these two programmes is assumed to account for the results obtained. This point is being examined at present. ~~The absence of parameter error estimates in the Gilles programme is a disadvantage~~. It must be stressed however that the theory of estimating confidence regions of non linear parameters is not well developed and any estimates obtained from the curve fitting procedures are only approximate. No mathematical criteria of selecting a particular curve fitting procedure for any specific problem can be given by us at this time.

Conclusion

The existence of a second pool of uric acid, or alternatively, the existence of uric acid in a physiologically different form, is highly probable in five patients. In these patients no inconsistency exists between the number of exponentials present, the magnitude of the error estimates, or the effect of the presence or omission of the first data point. The exchange of uric acid between serum and tophi has been suggested by Seegmiller (1963) and Sorensen (1962) and it would seem reasonable to assume that the second pool here represents either microtophi or the exchangeable surface of tophaceous deposits. The two compartments analysis procedure provides a method of obtaining a quantitative estimate of the exchangeable uric acid associated with tophi or microtophi.

Acknowledgements

References

Benedict, J. D., P. H. Forsham and D. Stetton jr.: J. biol. Chem. 181 (1949), 183.

Berman, M., E. Shahn and M. F. Weiss: Biophys. J. 2 (1962), 275.

Bishop, C., W. Garner and J. H. Talbott: J. clin. Invest. 30 (1951), 879.

Bray, G. A.: Analyt. Biochem. 1 (1960), 279.

Gilles, D. C. and S. S. Ben Hameid: Personal communication, 1966.

Glass, H. I. and A. C. de Garreta: Phys. in Med. Biol. 12 (1967), 379.

Geren, W., A. Bendich, O. Bodansky and G. B. Brown: J. biol. Chem. 183 (1950), 21.

Helf, S., C. G. White and R. N. Shelley: Anal. Chem. 32 (1960), 238.

Praetorius, E.: Scand. J. clin. Lab. Invest. 1 (1949), 222.

Seegmiller, J. E., A. I. Grayzel, L. Laster and L. Liddle: J. clin. Invest. 40 (1961), 1304.

Seegmiller, J. E., L. Laster and R. R. Howell: New Engl. J. Med. 268 (1963), 712.

Sorensen, L. B.: Scand. J. clin. Lab. Invest. 12 (1960), 54.

Sorensen, L. B.: Arch. intern. Med. 109 (1962), 379.

Sorensen, L. B.: Med. Clin. N. Amer. 47 (1963), 169.

APPENDIX   B

PROGRAM I : Exponential Curve Fitting Program using

Two Stage Minimization Method .

PROGRAM II: Exponential Curve Fitting Program using

One Stage Minimization Method .

PROGRAM III: Exponential Curve Fitting Program using

Berman Method .

INPUT DATA

The input data for the three programs are the same , and they are :

(i) N1 ;   (number of sets of data to be fitted)

(ii) Data set number ;

(iii) Case ; (weighting factor : is either 1 i.e the observations have a uniform weights , 2 i.e the observations are taken at fixed interval of time or 3 i.e the observations are taken at fixed number of counts)

(iv) N ;   (number of trials : N is always equal tp 1 unless it is required to fit the data (the same set) to different number of exponents in this case steps (iiiv) and (ix) should be repeated with new values)

(v) m ;   (number of observations)

(vi) t_1 ;t_2;...;t_m;

(vii) y_1;y_2; ...;y_m;

(viii) n ; (number of exponents)

(ix)   1; 2 ;....; n; (initial values of exponents)

OUTPUT

The following results are the output from each program :

(i)   Data set number ;

(ii)   Weighting factor ;

(iii)   Number of exponents ;

(iv)   Coefficients ;

(v)   Standard errors of coefficients and exponents ;

(vi)   Exponents ;

(vii)   Maximum absolute value of correction in exponents ;

(viii)   Sum squares of residuals ;

(ix)   Variance ratio (F) ;

(x)   Probability of F ;

(xi)   Number of iterations ;

```
begin  library A0,A6,A14;

       comment This program is for Exponential Curve Fitting
       using TWO STAGE MINIMIZATION  method;
       procedure  EXP MAT(a,t,n,m,E);
       value a,t,n,m;  integer n,m; array a,t,E;
       begin  integer i,j;
              for i:= 1 step 1 until n do
              for j:= 1 step 1 until m do
              E[i,j]:=exp(-a[i]xt[j]);
         end EXP MAT;
       procedure MAT TRANS(E,n,m,D);
       value E,n,m;  integer n,m; array E,D;
       begin  integer i,j;
              for i:= 1 step 1 until n do
              for j:= 1 step 1 until m do
              D[j,i]:=E[i,j];
         end MAT TRANS;
       procedure DIAG(t,m,T);
       value t,m; integer m; array t,T;
       begin  integer i,j;
              for i:= 1 step 1 until m do
              for j:= 1 step 1 until m do
              T[i,j]:= if i=j then t[j] else 0.0;
         end DIAG;
       procedure  MM MULT(E,D,n,m,G);
       value E,D,n,m;  integer n,m; array E,D,G;
```

```
begin  integer i,j,k;  real g;
       for i:= 1 step 1 until n do
       begin  for k:= 1 step 1 until n do
              begin g:=0.0;
                    for j:= 1 step 1 until m do
                    g:=E[i,j]×D[j,k]+g;
                    G[i,k]:=g;
              end;
       end;
   end MM MULT;
procedure SMM MULT(E,T,n,m,ET);
value E,T,n,m; integer n,m; array E,T,ET;
begin integer i,j,k; real g;
       for i:=1 step 1 until n do
       begin for k:=1 step 1 until m do
             begin g:=0.0;
                   for j:=1 step 1 until m do
                   g:=E[i,j]×T[j,k]+g;
                   ET[i,k]:=g;
             end;
       end;
   end SMM MULT;
procedure MV MULT(E,y,n,m,a);
value E,y,n,m;  integer n,m; array E,y,a;
begin  integer i,j;  real g;
```

```
            for i:=1 step 1 until n do
            begin  g:=0.0;
                   for j:=1 step 1 until m do
                   g:=E[i,j]×y[j]+g;
                   a[i]:=g;
            end;
      end MV MULT;
procedure  MAT SUB(G,A1,n,X);
value G,A1,n; integer n; array G,A1,X;
begin  integer i,j;
       for i:=1 step 1 until n do
       for j:=1 step 1 until n do
       X[i,j]:=G[i,j]-A1[i,j];
    end MAT SUB;
real procedure  MAX EL(b,n);
value b,n; integer n; array b;
begin  integer i; real eps;
       eps := 0.0;
       for i:=1 step 1 until n do
       eps := if abs(b[i])>eps then abs(b[i]) else eps;
       MAXEL := eps;
    end MAX EL;
real procedure PROB(F,n1,n2);
value F,n1,n2; integer n1,n2; real F;
begin real prob,a0,b0,u0,x0,a1,a2,a3,a4;
```

```
        a0:=1.0-2.0/(9.0×n2);   b0:=1.0-2.0/(9.0×n1);
        u0:= (a0×F↑(1/3)-b0)/((1.0-a0)×F↑(2/3)+1.0-b0)↑(1/2);
        a1:=0.278393; a2:=0.230389;  a3:=0.000972;
        a4:=0.078108;  x0:=u0/sqrt(2.0);
        prob:=0.5/((((a4×x0+a3)×x0+a2)×x0+a1)×x0+1.0)↑4;
        PROB := prob;
    end PROB;
procedure SOLVE(C,n,R,X,FAIL);
value C,n,R;  integer n; label FAIL; array C,R,X;
begin integer i,im1,j,jj,k;  real t,modt,p,modp;
        array ROWC,ROWR[1:n]; integer array J[1:n];
        for i:=1 step 1 until n do
        begin ROW:p:=modp:=0.0;
            for j:=1 step 1 until n do
            begin ROWC[j]:=t:=C[i,j];
                    modt := if t<0.0 then -t else t;
                    if modt>modp then
                    begin p:=t ;   modp:=modt;   jj:=j end
            end;
            if p = 0.0 then goto FAIL;
            for j:=1 step 1 until n do
            begin C[i,j]:=ROWC[j]:=ROWC[j]/p;
                    R[i,j]:=ROWR[j]:=R[i,j]/p;
            end;
            J[i]:=jj;   im1:=i-1;
```

```
                    for j:=1 step 1 until im1,i+1 step 1 until n do
                    begin  t:=C[j,jj];
                            for k:=1 step 1 until n do
                            begin C[j,k]:=C[j,k]-ROWC[k]×t;
                                   R[j,k]:=R[j,k]-ROWR[k]×t
                            end;
                    end;
                end;
            for i:=1 step 1 until n do
            begin jj:=J[i];
                    for j:=1 step 1 until n do
                    X[jj,j]:=R[i,j]
            end
        end SOLVE;
procedure INVERT(M,n,MI,FAIL);
value M,n; integer n;  label FAIL; array M,MI;
begin integer i,j; array R[1:n,1:n];
        for i:=1 step 1 until n do
        for j:=1 step 1 until n do
        R[i,j]:= if i=j then 1.0 else 0.0;
        SOLVE(M,n,R,MI,FAIL);
    end INVERT;
procedure WVEC(n,f,c);
value n,f,c; integer n,f; array c;
begin integer i;
```

```
        for i := 1 step 1 until n do
        begin write(70,f,c[i]);
                if i/4 =i÷4 and n > 4 then
                write text(70,[[2c24s]]);
            end;
        end WVEC;
procedure FORM(A,c,K,V,n,X);
value A,c,K,V,n; integer n; array A,c,K,V,X;
begin integer i,j;
        for i:= 1 step 1 until n do
        for j:= 1 step 1 until n do
        begin X[i,j]:=A[i,j];
                X[i,j+n]:=X[j+n,i]:=-K[i,j];
                X[i+n,j+n]:=c[i]×V[i,j];
            end;
        end FORM;
procedure FEEZ(Z,A,n,m,v,b,F2);
value n,m; integer n,m; label F2; array Z,A,v,b;
begin integer i,j;
        for i:= 2 step 1 until n do
        begin for j:= 2 step 1 until n do
                Z[i-1,j-1]:=Z[i,j];
                v[i-1]      :=v[i];
            end;
        INVERT(Z,n-1,A,F2);   MV MULT(A,v,n-1,n-1,b);
```

```
            for i:= n step -1 until 2 do
            b[i]:=b[i-1];

            b[1]:=0.0;

      end FEEZ;

integer n,m,N,P,h,N1,case,NI,I,i,j,k,n1,n2,f1,f2;  real prob,F;
open(20);  open(70);

N1:=read(20);    comment the number of sets of data;

f1:=format([ndds;]); f2:=format([4s+d.ddddsdddds₁₀+nd;]);

for I:=1 step 1 until N1 do

begin write text(70,[[p]DATA*SET*NUMBER[13s]]);

      copy text(20,70,[;;]);

      case := read(20);    comment the weighting factor;

      N    := read(20);    comment the number of trials;

      m    := read(20);    comment the number of observations;

      begin array t,y,f,r,w[1:m],T,W[1:m,1:m],S[0:N],MS,SM[1:N];

            integer array df[0:N];

            for j:=1 step 1 until m do
            t[j]:=read(20);  DIAG(t,m,T);

            for j:=1 step 1 until m do
            begin y[j]:=F:=read(20);
                  w[j]:=1.0/FT(case-1);

              end;

            comment t and y are the values of observations;
            DIAG(w,m,W);

            write text(70,[[4s]WEIGHTING*FACTOR[4s]]);
```

```
write (70,f1,case);

for k:=1 step 1 until N do

begin n:=read (20); comment the number of exponents

        h:=2×n;

        write text (70,[[3c]NUMBER *OF *EXPONENTS[8s]])

        write (70,f1,n);

        begin array E,EW,ET,ETT[1:n,1:m],D[1:m,1:n]

                ,B,C,K,V,Z,BK,CK,A,A1[1:n,1:n]

                ,X,X1[1:h,1:h],alpha,a,c,v,z,

                delta alpha[1:n];

                for i:=1 step 1 until n do

                alpha[i]:=read(20);

                comment the initial values of exponents

                NI:=P:=0;

        LL:     EXP MAT (alpha,t,n,m,E);

                MAT TRANS (E,n,m,D);

                SMM MULT (E,W,n,m,EW);

                MM MULT (EW,D,n,m,A);

                INVERT (A,n,A1,F2);

                MV MULT (EW,y,n,m,a);

                MV MULT (A1,a,n,n,c);

                MV MULT (D,c,m,n,f);

                for j:=1 step 1 until m do

                r[j]:=y[j]-f[j];

                SMM MULT (EW,T,n,m,ET);
```

```
MM MULT(ET,D,n,m,B);

SMM MULT(ET,T,n,m,ETT);

MM MULT(ETT,D,n,m,C);

MV MULT(ET,r,n,m,v);

MV MULT(ETT,r,n,m,z);

DIAG(c,n,K);

DIAG(v,n,V);

DIAG(z,n,Z);

MM MULT(B,K,n,n,BK);

MM MULT(C,K,n,n,CK);

MAT SUB(BK,V,n,K);

MAT SUB(CK,Z,n,V);

MM MULT(A1,K,n,n,Z);

MM MULT(B,Z,n,n,CK);

MAT SUB(V,CK,n,Z);

if alpha[1]=0.0 then

begin FEEZ(Z,CK,n,m,v,delta alpha,F2);

      df[k]:=m-h-1;  goto Q;

   end;

INVERT(Z,n,CK,F2);

MV MULT(CK,v,n,n,delta alpha);

df[k]:=m-h;

Q : for i:=1 step 1 until n do

begin alpha[i]:=alpha[i]-delta alpha[i];

      if alpha[i] < 0.0 then goto F1;
```

```
                end;
S[k]:=0.0;

for j:=1 step 1 until m do

S[k]:=r[j]×w[j]×r[j]+S[k];

NI:=NI+1;

if MAXEL(delta alpha,n) > 10^{-4}

and P=0 then goto LL;

FORM(A,c,K,V,n,X);

if alpha[1] = 0.0 then

begin  for i:= 1 step 1 until n do

       for j:= 1 step 1 until n-1 do

       X[i,j+n]:=X[j+n,i]:=X[i,j+1+n];

       for i:= n step 1 until h-2 do

       for j:= n step 1 until h-2 do

       X[i+1,j+1]:=X[i+2,j+2];

       INVERT(X,h-1,X1,F2);

       for i:= h step -1 until n+2 do

       X1[i,i]:=X1[i-1,i-1];

       X1[n+1,n+1]:= 0.0;

   end else    INVERT(X,h,X1,F2);
write text(70,[[2c]COEFFICIENTS[12s]]);
WVEC(n,f2,c);

MS[k] := S[k]/df[k];

write text(70,[[2c]STE*OF*PARAMETERS[7s
]);

for i:= 1 step 1 until h do
```

```
          begin if X1[i,i] < 0.0 then goto R;
              f[i]:=sqrt (MS[k]×X1[i,i]);
          end;
      WVEC (h,f2,f);  goto M;
R :   write text(70,[ ****NO*STE*ELEMENTS
          *OF*MATRIX*ARE*NEGATIVE]);
M :   write text (70,[[2c]ALPHA[15s]****]);
      WVEC (n,f2,alpha);
      write text(70,[[2c]MAX*ABS*DELTA*
      ALPHA[5s]]);
      write(70,f2,MAXEL(delta alpha,n) );
      write text(70,[[2c]SUM*SQUARES*
      OF*RESIDUALS]);
      write(70,f2,S[k]);
      P:=P+1 ;  if P=1 then goto LL;
      if k=1 then
      begin S[0]:=0.0;
          for j:=1 step 1 until m do
          S[0]:=y[j]×w[j]×y[j]+S[0];
          df[0]:=m;
        end;
      n2:=df[k];  n1:=df[k-1]-n2;
      SM[k]:=abs (S[k-1]-S[k])/n1;
      if SM[k]<MS[k] then
```

```
                              begin F:=MS[k]/SM[k];
                                   prob := PROB(F,n2,n1);
                               end else
                              begin F:=SM[k]/MS[k];
                                   prob := PROB(F,n1,n2);
                               end;
                              write text(70,[[2c]VARIANCE*
                              RATIO*(F)[6s]]);
                              write(70,f2,F);
                              write text(70,[[2c]PROBABILITY*
                              OF*F[8s]]);
                              write(70,f2,prob);
                              write text(70,[[2c]NUMBER*OF*
                              ITERATIONS[6s]]);
                              write(70,f1,NI-1);    goto EXIT;
                        F1:   write text(70,[[2c]FAILED*IN*
                              EXPONENTIATION]);     goto EXIT;
                        F2:   write text(70,[[2c]FAILED*IN*
                              MATRIX*INVERSION]);
             EXIT:   end;
          end;
       end;
    end; close(70); close(20);
 end →
```

```
begin   library A0,A6,A14;

        comment This program is for Exponential Curve Fitting

        using ONE STAGE MINIMIZATION  method;

        procedure  EXP MAT(a,t,n,m,E);

        value a,t,n,m;  integer n,m; array a,t,E;

        begin  integer i,j;
                for i:= 1 step 1 until n do
                for j:= 1 step 1 until m do
                E[i,j]:=exp(-a[i]xt[j]);

          end EXP MAT;

        procedure MAT TRANS(E,n,m,D);

        value E,n,m;  integer n,m; array E,D;

        begin   integer i,j;
                for i:= 1 step 1 until n do
                for j:= 1 step 1 until m do
                D[j,i]:=E[i,j];

         end MAT TRANS;

        procedure DIAG(t,m,T);

        value t,m; integer m; array t,T;

        begin  integer i,j;
                for i:= 1 step 1 until m do
                for j:= 1 step 1 until m do
                T[i,j]:= if i=j then t[j] else 0.0;

          end DIAG;

        procedure  MM MULT(E,D,n,m,G);
        value E,D,n,m;  integer n,m; array E,D,G;
```

```
begin  integer i,j,k;  real g;
      for i:= 1 step 1 until n do
      begin  for k:= 1 step 1 until n do
             begin g:=0.0;
                     for j:= 1 step 1 until m do
                     g:=E[i,j]×D[j,k]+g;
                     G[i,k]:=g;
             end;
      end;
   end MM MULT;
procedure SMM MULT(E,T,n,m,ET);
value E,T,n,m; integer n,m; array E,T,ET;
begin integer i,j,k; real g;
      for i:=1 step 1 until n do
      begin for k:=1 step 1 until m do
             begin g:=0.0;
                     for j:=1 step 1 until m do
                     g:=E[i,j]×T[j,k]+g;
                     ET[i,k]:=g;
             end;
      end;
   end SMM MULT;
procedure MV MULT(E,y,n,m,a);
value E,y,n,m;  integer n,m; array E,y,a;
begin  integer i,j;  real g;
```

```
              for i:=1 step 1 until n do
              begin  g:=0.0;
                      for j:=1 step 1 until m do
                      g:=E[i,j]xy[j]+g;
                      a[i]:=g;
              end;
        end MV MULT;
procedure  MAT SUB(G,A1,n,X);
value G,A1,n;  integer n; array G,A1,X;
begin  integer i,j;
              for i:=1 step 1 until n do
              for j:=1 step 1 until n do
              X[i,j]:=G[i,j]-A1[i,j];
        end MAT SUB;
real procedure  MAX EL(b,n);
value b,n; integer n; array b;
begin  integer i; real eps;
        eps := 0.0;
              for i:=1 step 1 until n do
              eps := if abs(b[i])>eps then abs(b[i]) else eps;
        MAXEL := eps;
        end MAX EL;
real procedure PROB(F,n1,n2);
value F,n1,n2; integer n1,n2; real F;
begin real prob,a0,b0,u0,x0,a1,a2,a3,a4;
```

```
        a0:=1.0-2.0/(9.0×n2);  b0:=1.0-2.0/(9.0×n1);

        u0:=(a0×F↑(1/3)-b0)/((1.0-a0)×F↑(2/3)+1.0-b0)↑(1/2);

        a1:=0.278393;  a2:=0.230389;  a3:=0.000972;

        a4:=0.078108;  x0:=u0/sqrt(2.0);

        prob:=0.5/((((a4×x0+a3)×x0+a2)×x0+a1)×x0+1.0)↑4;

        PROB := prob;

    end PROB;

procedure SOLVE(C,n,R,X,FAIL);

value C,n,R;  integer n; label FAIL; array C,R,X;

begin integer i,im1,j,jj,k;  real t,modt,p,modp;

        array ROWC,ROWR[1:n]; integer array J[1:n];

        for i:=1 step 1 until n do

        begin ROW:p:=modp:=0.0;

                for j:=1 step 1 until n do

                begin ROWC[j]:=t:=C[i,j];

                        modt := if t<0.0 then -t else t;

                        if modt>modp then

                        begin p:=t ;  modp:=modt;  jj:=j end

                end;

                if p = 0.0 then goto FAIL;

                for j:=1 step 1 until n do

                begin C[i,j]:=ROWC[j]:=ROWC[j]/p;

                        R[i,j]:=ROWR[j]:=R[i,j]/p;

                end;

                J[i]:=jj;  im1:=i-1;
```

```
                    for j:=1 step 1 until im1,i+1 step 1 until n d
                    begin  t:=C[j,jj];
                            for k:=1 step 1 until n do
                            begin C[j,k]:=C[j,k]-ROWC[k]×t;
                                  R[j,k]:=R[j,k]-ROWR[k]×t
                        end;
                end;
            end;
        for i:=1 step 1 until n do
        begin jj:=J[i];
                for j:=1 step 1 until n do
                X[jj,j]:=R[i,j]
            end
        end SOLVE;
    procedure INVERT(M,n,MI,FAIL);
    value M,n; integer n;  label FAIL; array M,MI;
    begin integer i,j; array R[1:n,1:n];
            for i:=1 step 1 until n do
            for j:=1 step 1 until n do
            R[i,j]:= if i=j then 1.0 else 0.0;
            SOLVE(M,n,R,MI,FAIL);
        end INVERT;
    procedure WVEC(n,f,c);
    value n,f,c; integer n,f; array c;
    begin integer i;
```

```
        for i := 1 step 1 until n do
        begin write(70,f,c[i]);
                if i/4 =i÷4 and n > 4 then
                write text(70,[[2c24s]]);
            end;
    end WVEC;
procedure FORM(A,c,K,V,n,X);
value A,c,K,V,n; integer n; array A,c,K,V,X;
begin integer i,j;
        for i:= 1 step 1 until n do
        for j:= 1 step 1 until n do
        begin X[i,j]:=A[i,j];
                X[i,j+n]:=X[j+n,i]:=-K[i,j];
                X[i+n,j+n]:=c[i]×V[i,j];
            end;
    end FORM;
procedure FEEZ(X,X1,n,h,m,d,b,F2);
value n,h,m; integer n,h,m; label F2; array X,X1,d,b;
begin integer i,j;
        for i:= 1 step 1 until n do
        for j:= 1 step 1 until n-1  do
        X[i,j+n]:=X[j+n,i]:=X[i,j+1+n];
        for i:= n step 1 until h-2 do
        begin for j:= n step 1 until h-2 do
                X[i+1,j+1]:=X[i+2,j+2];
```

```
                        d[i+1]:=d[i+2];
            end;
            INVERT(X,h-1,X1,F2);   MV MULT(X1,d,h-1,h-1,b);
            for i:= h step -1 until n+2 do
            begin b[i]:=b[i-1];
                    X1[i,i]:=X1[i-1,i-1];
                end;
            b[n+1]:=X1[n+1,n+1]:=0.0;
        end FEEZ;
integer n,m,N,P,h,N1,case,NI,I,i,j,k,n1,n2,f1,f2;  real prob
open(20);  open(70);
N1:=read(20);   comment the number of sets of data;
f1:=format([ndds;]); f2:=format([4s+d.ddddsdddds₁₀+nd;]);
for I:=1 step 1 until N1 do
begin write text(70,[[p]DATA*SET*NUMBER[13s]]);
        copy text(20,70,[;;]);
        case := read(20);     comment the weighting factor;
        N    := read(20);     comment the number of trials;
        m    := read(20);     comment the number of observations
        begin array t,y,f,r,w[1:m],T,W[1:m,1:m],S[0:N],MS,SM[1
                ; integer array df[0:N];
                for j:=1 step 1 until m do
                t[j]:=read(20);  DIAG(t,m,T);
                for j:=1 step 1 until m do
                begin y[j]:=F:=read(20);
```

```
                    w[j]:=1.0/F↑(case-1);
      end;
comment t and y are the values of observations;
DIAG(w,m,W);
write text (70,[[4s]WEIGHTING*FACTOR[4s]]);
write(70,f1,case);
for k:=1 step 1 until N do
begin n:=read(20); comment the number of exponents;
      h:=2×n;
      write text (70,[[3c]NUMBER*OF*EXPONENTS[8s]]);
      write(70,f1,n);
      begin array E,EW,ET,ETT[1:n,1:m],D[1:m,1:n]
            ,B,C,K,V,Z,BK,CK,A,A1[1:n,1:n]
            ,X,X1[1:h,1:h],alpha,a,c,v,z,
            delta alpha[1:n],d,b[1:h];
            for i:=1 step 1 until n do
            alpha[i]:=read(20);
            comment the initial values of exponents
            ;NI:=P:=0;
      LL:   EXP MAT(alpha,t,n,m,E);
            MAT TRANS (E,n,m,D);
            SMM MULT(E,W,n,m,EW);
            MM MULT(EW,D,n,m,A);
            INVERT(A,n,A1,F2);
            MV MULT(EW,y,n,m,a);
```

```
MV MULT(A1,a,n,n,c);
MV MULT(D,c,m,n,f);
for j:=1 step 1 until m do
r[j]:=y[j]-f[j];
SMM MULT(EW,T,n,m,ET);
MM MULT(ET,D,n,m,B);
SMM MULT(ET,T,n,m,ETT);
MM MULT(ETT,D,n,m,C);
MV MULT(ET,r,n,m,v);
MV MULT(ETT,r,n,m,z);
DIAG(c,n,K);
DIAG(v,n,V);
DIAG(z,n,Z);
MM MULT(B,K,n,n,BK);
MM MULT(C,K,n,n,CK);
MAT SUB(BK,V,n,K);
MAT SUB(CK,Z,n,V);
MV MULT(EW,r,n,m,z);
FORM(A,c,K,V,n,X);
for i:= 1 step 1 until n do
begin d[i]:=z[i];
      d[i+n]:=-c[i]×v[i];
   end;
if alpha[1]=0.0 then
begin FEEZ(X,X1,n,h,m,d,b,F2);
```

```
                        df[k]:=m-h-1;  goto Q;
                 end;
             INVERT(X,h,X1,F2);
             MV MULT(X1,d,h,h,b); df[k]:=m-h;
    Q :  for i:=1 step 1 until n do
         begin delta alpha[i]:=b[i+n];
                 alpha[i]:=alpha[i]+delta alpha[i]
                 if alpha[i] < 0.0 then goto F1;
             end;
         S[k]:=0.0;
         for j:=1 step 1 until m do
         S[k]:=r[j]×w[j]×r[j]+S[k];
         NI:=NI+1;
         if MAXEL(b,n) > 10-5 then goto LL;
         write text(70,[[2c]COEFFICIENTS[12s]]);
         WVEC(n,f2,c);
         MS[k] := S[k]/df[k];
         write text(70,[[2c]STE*OF*PARAMETERS
         [7s]]);
         for i:= 1 step 1 until h do
         begin if X1[i,i] < 0.0 then goto R;
                 f[i]:=sqrt(MS[k]×X1[i,i]);
             end;
         WVEC(h,f2,f);  goto M;
    R :   write text(70,[****NO*STE*ELEMENTS
             *OF*MATRIX*ARE*NEGATIVE]);
    M :   write text (70,[[2c]ALPHA[15s]****]);
```

```
WVEC (n,f2,alpha);
write text (70,[[2c]MAX*ABS*DELTA*
PARAMETER*]);
write (70,f2,MAXEL(b,n) );
write text (70,[[2c]SUM*SQUARES*
OF*RESIDUALS]);
write (70,f2,S[k]);
P:=P+1 ;  if P=1 then goto LL;
if k=1 then
begin S[0]:=0.0;
      for j:=1 step 1 until m do
      S[0]:=y[j]×w[j]×y[j]+S[0];
      df[0]:=m;
  end;
n2:=df[k];  n1:=df[k-1]-n2;
SM[k]:=abs (S[k-1]-S[k])/n1;
if SM[k]<MS[k] then
begin F:=MS[k]/SM[k];
      prob := PROB(F,n2,n1);
  end else
begin F:=SM[k]/MS[k];
      prob := PROB(F,n1,n2);
  end;
write text (70,[[2c]VARIANCE*
RATIO*(F)[6s]]);
```

```
                              write(70,f2,F);

                              write text(70,[[2c]PROBABILITY*

                              OF*F[8s]]);

                              write(70,f2,prob);

                              write text(70,[[2c]NUMBER*OF*

                              ITERATIONS[6s]]);

                              write(70,f1,NI-1);   goto EXIT;

                 F1:   write text(70,[[2c]FAILED*IN*

                              EXPONENTIATION]);

                              goto EXIT;

                 F2:   write text(70,[[2c]FAILED*IN*

                              MATRIX*INVERSION]);

             EXIT:   end;

          end;

       end;

    end; close(70); close(20);

end ->
```

```
begin  library A0,A6,A14;

       comment This program is for Exponential Curve Fitting
       using BERMAN  method;

       procedure  EXP MAT(a,t,n,m,E);

       value a,t,n,m;  integer n,m; array a,t,E;

       begin  integer i,j;
              for i:= 1 step 1 until n do
              for j:= 1 step 1 until m do
              E[i,j]:=exp(-a[i]×t[j]);

          end EXP MAT;

       procedure MAT TRANS(E,n,m,D);

       value E,n,m;  integer n,m; array E,D;

       begin  integer i,j;
              for i:= 1 step 1 until n do
              for j:= 1 step 1 until m do
              D[j,i]:=E[i,j];

         end MAT TRANS;

       procedure DIAG(t,m,T);

       value t,m; integer m; array t,T;

       begin  integer i,j;
              for i:= 1 step 1 until m do
              for j:= 1 step 1 until m do
              T[i,j]:= if i=j then t[j] else 0.0;

         end DIAG;

       procedure  MM MULT(E,D,n,m,G);

       value E,D,n,m;  integer n,m; array E,D,G;
```

```
begin  integer i,j,k;  real g;
      for i:= 1 step 1 until n do
      begin  for k:= 1 step 1 until n do
            begin g:=0.0;
                  for j:= 1 step 1 until m do
                  g:=E[i,j]×D[j,k]+g;
                  G[i,k]:=g;
            end;
        end;
   end MM MULT;
procedure SMM MULT(E,T,n,m,ET);
value E,T,n,m; integer n,m; array E,T,ET;
begin integer i,j,k; real g;
      for i:=1 step 1 until n do
      begin for k:=1 step 1 until m do
            begin g:=0.0;
                  for j:=1 step 1 until m do
                  g:=E[i,j]×T[j,k]+g;
                  ET[i,k]:=g;
            end;
        end;
   end SMM MULT;
procedure MV MULT(E,y,n,m,a);
value E,y,n,m;  integer n,m; array E,y,a;
begin  integer i,j; real g;
```

```
              for i:=1 step 1 until n do
              begin g:=0.0;
                      for j:=1 step 1 until m do
                      g:=E[i,j]×y[j]+g;
                      a[i]:=g;
              end;
        end MV MULT;
  procedure  MAT SUB(G,A1,n,X);
  value G,A1,n;  integer n; array G,A1,X;
  begin  integer i,j;
              for i:=1 step 1 until n do
              for j:=1 step 1 until n do
              X[i,j]:=G[i,j]-A1[i,j];
        end MAT SUB;
  real procedure  MAX EL(b,n);
  value b,n; integer n; array b;
  begin  integer i; real eps;
              eps := 0.0;
              for i:=1 step 1 until n do
              eps := if abs(b[i])>eps then abs(b[i]) else eps;
              MAXEL := eps;
        end MAX EL;
  real procedure PROB(F,n1,n2);
  value F,n1,n2; integer n1,n2; real F;
  begin real prob,a0,b0,u0,x0,a1,a2,a3,a4;
```

```
        a0:=1.0-2.0/(9.0×n2);  b0:=1.0-2.0/(9.0×n1);

        u0:=(a0×F↑(1/3)-b0)/((1.0-a0)×F↑(2/3)+1.0-b0)↑(1/2);

        a1:=0.278393; a2:=0.230389;  a3:=0.000972;

        a4:=0.078108;  x0:=u0/sqrt(2.0);

        prob:=0.5/((((a4×x0+a3)×x0+a2)×x0+a1)×x0+1.0)↑4;

        PROB := prob;

    end PROB;

procedure SOLVE(C,n,R,X,FAIL);

value C,n,R;  integer n; label FAIL; array C,R,X;

begin integer i,im1,j,jj,k;  real t,modt,p,modp;

        array ROWC,ROWR[1:n]; integer array J[1:n];

        for i:=1 step 1 until n do

        begin ROW:p:=modp:=0.0;

                for j:=1 step 1 until n do

                begin ROWC[j]:=t:=C[i,j];

                        modt:= if t<0.0 then -t else t;

                        if modt>modp then

                        begin p:=t ;  modp:=modt;  jj:=j end

                end;

                if p = 0.0 then goto FAIL;

                for j:=1 step 1 until n do

                begin C[i,j]:=ROWC[j]:=ROWC[j]/p;

                        R[i,j]:=ROWR[j]:=R[i,j]/p;

                end;

                J[i]:=jj;  im1:=i-1;
```

```
                    for j:=1 step 1 until im1,i+1 step 1 until n do
                    begin  t:=C[j,jj];
                            for k:=1 step 1 until n do
                            begin C[j,k]:=C[j,k]-ROWC[k]xt;
                                    R[j,k]:=R[j,k]-ROWR[k]xt
                            end;
                    end;
                end;
            for i:=1 step 1 until n do
            begin jj:=J[i];
                    for j:=1 step 1 until n do
                    X[jj,j]:=R[i,j]
            end
        end SOLVE;
    procedure INVERT(M,n,MI,FAIL);
    value M,n; integer n;  label FAIL; array M,MI;
    begin integer i,j; array R[1:n,1:n];
            for i:=1 step 1 until n do
            for j:=1 step 1 until n do
            R[i,j]:= if i=j then 1.0 else 0.0;
            SOLVE(M,n,R,MI,FAIL);
        end INVERT;
    procedure WVEC(n,f,c);
    value n,f,c; integer n,f; array c;
    begin integer i;
```

```
        for i := 1 step 1 until n do
        begin write(70,f,c[i]);
                if i/4 =i÷4 and n > 4 then
                write text(70,[[2c 24s]]);
            end;
    end WVEC;
procedure FORM(A,B,C,n,K);
value A,B,C,n; integer n; array A,B,C,K;
begin integer i,j;
        for i:= 1 step 1 until n do
        for j:= 1 step 1 until n do
        begin K[i,j]:=A[i,j];
                K[i,j+n]:=K[j+n,i]:=B[i,j];
                K[i+n,j+n]:=C[i,j];
            end;
    end FORM;
procedure FEEZ(X,X1,n,h,m,d,b,F2);
value n,h,m; integer n,h,m; label F2; array X,X1,d,b;
begin integer i,j;
        for i:= 1 step 1 until n do
        for j:= 1 step 1 until n-1 do
        X[i,j+n]:=X[j+n,i]:=X[i,j+1+n];
        for i:= n step 1 until h-2 do
        begin for j:= n step 1 until h-2 do
                X[i+1,j+1]:=X[i+2,j+2];
                d[i+1]:=d[i+2];
```

```
                end;
        INVERT(X,h-1,X1,F2);   MV MULT(X1,d,h-1,h-1,b);
        for i:= h step -1 until n+2 do
        begin b[i]:=b[i-1];
            X1[i,i]:=X1[i-1,i-1];
            end;
        b[n+1]:=X1[n+1,n+1]:=0.0;
    end FEEZ;
integer n,m,N,P,h,N1,case,NI,I,i,j,k,n1,n2,f1,f2;   real prob,F
open(20);   open(70);
N1:=read(20);    comment the number of sets of data;
f1:=format([ndds;]); f2:=format([4s+d.ddddsddddsₗₒ+nd;]);
for I:=1 step 1 until N1 do
begin write text(70,[[p]DATA*SET*NUMBER[13s]]);
        copy text(20,70,[;;]);
        case := read(20);    comment the weighting factor;
        N    := read(20);    comment the number of trials;
        m    := read(20);    comment the number of observations;
        begin array t,y,f,r,w[1:m],T,W[1:m,1:m],S[0:N],MS,SM[1:N]
            integer array df[0:N];
            for j:=1 step 1 until m do
            t[j]:=read(20);   DIAG(t,m,T);
            for j:=1 step 1 until m do
            begin y[j]:=F:=read(20);
                w[j]:=1.0/F↑(case-1);
```

```
          end;

comment t and y are the values of observations;

DIAG(W,m,W);

write text(70,[[4s]WEIGHTING*FACTOR[4s]]);

write(70,f1,case);

for k:=1 step 1 until N do

begin n:=read(20); comment the number of exponents;

        h:=2×n;

        write text(70,[[3c]NUMBER*OF*EXPONENTS[8s]]);

        write(70,f1,n);

        begin array E,EW,ET,ETT[1:n,1:m],D[1:m,1:n]

              ,B,C,A,A1[1:n,1:n],K,Z[1:h,1:h],alpha,

              a,c,v,z,delta alpha[1:n],b,d[1:h];

              for i:=1 step 1 until n do

              alpha[i]:=read(20);

              comment the initial values of exponents;

              NI:=P:=0;

        LL:   EXP MAT(alpha,t,n,m,E);

              MAT TRANS(E,n,m,D);

              SMM MULT(E,W,n,m,EW);

              MM MULT(EW,D,n,m,A);

              INVERT(A,n,A1,F2);

              MV MULT(EW,y,n,m,a);

              MV MULT(A1,a,n,n,c);

              MV MULT(D,c,m,n,f);
```

```
for j:=1 step 1 until m do
r[j]:=y[j]-f[j];
SMM MULT(EW,T,n,m,ET);
MM MULT(ET,D,n,m,B);
SMM MULT(ET,T,n,m,ETT);
MM MULT(ETT,D,n,m,C);
MV MULT(EW,r,n,m,v);
MV MULT(ET,r,n,m,z);
for i:= 1 step 1 until n do
begin b[i] := v[i];
      b[i+n]:=z[i];
   end;
FORM(A,B,C,n,K);
if alpha[1]≠ 0.0 then
begin FEEZ(K,Z,n,h,m,b,d,F2);
      df[k]:=m-h-1;   goto Q;
   end;
INVERT(K,h,Z,F2);
MV MULT(Z,b,h,h,d);   df[k]:=m-h;
Q :   for i:=1 step 1 until n do
begin delta alpha[i]:=d[i+n]:= -d[i+n]/
      c[i];
      alpha[i]:=alpha[i]+delta alpha[
      ];
      if alpha[i] < 0.0 then goto F1;
```

```
                                      end;
                              S[k]:=0.0;

                              for j:= 1 step 1 until m do
                              S[k]:=r[j]xw[j]xr[j]+S[k];
                              NI:=NI+1;

                              if MAXEL(d,n) > 10-5 then goto LL;
                              write text (70,[[2c]COEFFICIENTS[12s]]);
                              WVEC(n,f2,c);
                              MS[k] := S[k]/df[k];
                              write text (70,[[2c]STE*OF*PARAMETERS[7s
                              ]);

                              for i:= 1 step 1 until h do
                              begin if Z[i,i] < 0.0 then goto R;
                                    f[i]:=sqrt(MS[k]xZ[i,i]);
                                end;
                              for i:= 1 step 1 until n do
                              f[i+n]:=f[i+n]/c[i];
                              WVEC(h,f2,f);   goto M;
                      R :     write text (70,[*****NO*STE*ELEMENTS
                              *OF*MATRIX*ARE*NEGATIVE]);
                      M :     write text (70,[[2c]ALPHA[15s]****]);
                              WVEC(n,f2,alpha);
                              write text (70,[[2c]MAX*ABS*DELTA*
                              PARAMETERS]);
                              write(70,f2,MAXEL(d,h) );
```

```
write text(70,[[2c]SUM*SQUARES*
OF*RESIDUALS]);
write(70,f2,S[k]);
P:=P+1 ; if P=1 then goto LL;
if k=1 then
begin S[0]:=0.0;
      for j:=1 step 1 until m do
      S[0]:=y[j]×w[j]×y[j]+S[0];
      df[0]:=m;
  end;
n2:=df[k];  n1:=df[k-1]-n2;
SM[k]:=abs (S[k-1]-S[k])/n1;
if SM[k]<MS[k] then
begin F:=MS[k]/SM[k];
      prob := PROB(F,n2,n1);
  end else
begin F:=SM[k]/MS[k];
      prob := PROB(F,n1,n2);
  end;
write text(70,[[2c]VARIANCE*
RATIO*(F)[6s]]);
write(70,f2,F);
write text(70,[[2c]PROBABILITY*
OF*F[8s]]);
write(70,f2,prob);
write text(70,[[2c]NUMBER*OF*
```

```
                    ITERATIONS[6s]]);
              write(70,f1,NI-1);    goto EXIT;
        F1:   write text(70,[[2c]FAILED*IN*
              EXPONENTIATION]);
              goto EXIT;
        F2:   write text(70,[[2c]FAILED*IN*
              MATRIX*INVERSION]);
        EXIT:   end;
          end;
      end;
    end; close(70); close(20);
 end→
```

APPENDIX   C

The Initial Estimation of The Values of The

Exponents Program .

INPUT DATA

(i)   N ; (number of sets of data);

(ii)   Data set number ;

(iii)   m ; (number of observations);

(iv)   n ; (number of coefficients : n is either 1 case (a) , 2

               cases (b) and (c) or 3 case (d));

(v)   l ; (l is 1 case (a) or 2 case (b) or 3 case (c) or 4 case (d));

(vi)   d ; (first value of r);

(vii)   in; (increment in r);

(viii) li; (last value of r);

(ix)   case ; (weighting factor);

(x) $t_1 ; t_2 ; \ldots ; t_m ;$

(xi) $y ; y_2 ; \ldots ; y_m ;$

OUTPUT

(i)   Data set number ;

(ii)   Weighting factor ;

(iii) Coefficients ;

(iv)   Exponents ;

(v)   Minimum value of sum squares of residuals;

(vi)   x ; (the angle in degrees)

(vii) r ; (the modulus)

```
begin library A0,A6,A12,A14;
    comment This program is for estimating the parameters and
    evaluating the minimum;
    procedure VV MULT(a,t,n,m,A);
    value a,t,n,m; integer n,m; array a,t,A;
    begin integeri,j;
        for i:= 1 step 1 untiln do
        for j:= 1 step 1 until m do
        A[i,j]:=a[i]xt[j];
      end VV MULT;
    procedure EXP MAT(A,n,m,E);
    value A,n,m; integer n,m; array A,E;
    begininteger i,j;
        for i:= 1 step 1 until n do
        for j:= 1 step 1 until m do
        E[i,j]:=exp(-A[i,j]);
      end EXP MAT;
    procedur MAT TRANS(E,n,m,D);
    value E,n,m; integer n,m; array E,D;
    begin integer i,j;
        for i:= 1 step 1 until n do
        for j:= 1 step 1 until m do
        D[j,i]:=E[i,j];
      end MAT TRANS;
    procedure MM MULT(E,D,n,m,G);
    value E,D,n,m; integer n,m; array E,D,G;
      begin integer i,j,k; real g;
```

```
        for i:= 1 step 1 until n do
        begin for k:= 1 step 1 until n do
                begin g:= 0.0;
                        for j:= 1 step 1 until m do
                        g:=E[i,j]×D[j,k]+g;
                        G[i,k]:=g;
                end;
            end;
    end MM MULT;
procedure MV MULT(E,y,n,m,a);
value E,y,n,m; integer n,m; array E,y,a;
begininteger i,j; real g;
        for i:= 1 step 1 until n do
        begin g:=0.0;
                for j:= 1 step 1 until m do
                g:=E[i,j]×y[j]+g;
                a[i]:=g;
            end;
    end MV MULT;
procedure SOLVE(C,n,R,X);
value C,n,R; integer n; array C,R,X;
begin integer i,im1,j,jj,k;
        real t,modt,p,modp,r; array ROWC[1:n];
        integer array J[1:n];  .
        for i:= 1 step 1 untiln do
        begin ROW:p:=modp:=0.0;
```

```
                    for j:= 1 step 1 until n do
                    begin ROWC[j]:=t:=C[i,j];
                         modt:= if t < 0.0 then -t else t;
                         if modt > modp then
                         begin p:=t; modp:=modt;
                                jj:=j
                         end
                    end;
               for j:= 1 step 1 until n do
               C[i,j]:=ROWC[j]:=ROWC[j]/p;
               R[i]:=r:=R[i]/p;
               J[i]:=jj; im1:=i-1;
               for j:= 1 step 1until im1 ,i+1 step 1 until n do
               begin t:=C[j,jj];
                         for k:= 1 step 1 until n do
                         C[j,k]:=C[j,k]-ROWC[k]×t;
                         R[j]:=R[j]-r×t
               end;
          end;
          for i:= 1 step 1 until n do
          X[J[i]]:=R[i];
     end SOLVE;
procedure DIAG(t,m,t);
value t,m;integer m; array t,T;
begin intger i,j;
          for i:= 1 step 1 until m do
          for j:=1 step 1 until m do
```

```
                    T[i,j]:= if i=j then t[j] else 0.0;
            endDIAG;
    procedure SMMMULT(a,t,n,m,b);
    value a,t,n,m;integer n,m; array a,t,b;
    begin integer i,j,k; real g;
            for i:=1 step 1 until n do
            beginfor k:= 1 step 1 until m do
                    begin g:=0.0;
                            for j := 1 step 1 until m do
                            g:=a[i,j]×t[j,k]+g;
                            b[i,k]:=g;
                    end;
                end;
            end SMM MULT;
    procedure SELECT(l,a,b,x);
    value l,b,x; integer l;real x; array b,a;
    begin switch SW:=L1,L2,L3,L4; goto SW[l];
            L1: a[1]:=x; goto L;
            L2: a[1]:=0.0; a[2]:=x; goto L;
            L3: a[1]:=x×b[1]; a[2]:=x×b[2]; goto L;
            L4: a[1]:=0.0; a[2]:=x×b[1];a[3]:=x×b[2];
    L:end SELECT;
    procedure ESTIMATE(t,y,w,a,l,ang,n,m,minx,S,r,d,in,li);
    value t,y,w,l,n,m; integer l,ang,n,m;
    array t,y,w,a,r; realS,minx,d,in,li;
    begin integer i,j; real theta,T,x,s; arrayA,E[1:n,1:m],
            D[1:m,1:n],W[1:m,1:m],f[1:m],c[1:n],b[1:2];
```

```
T:=3.1415926536; S:=n+5; DIAG(w,m,W);
for j:= 1 step 1 until 20 do
begin theta:=2.0×j×T/180.0;
      b[1]:=cos(theta); b[2]:=sin(theta);
      forx :=d step in until li do
      begin SELECT(1,a,b,x);        VVMULT(a,t,n,m,A);
            EXPMAT(A,n,m,E);        MATTRANS(E,n,m,D);
            SMMMULT(E,W,n,m,A);     MMMULT(A,D,n,m,E);
            MVMULT(A,y,n,m,a);      SOLVE(E,n,a,c);
            MVMULT(D,c,m,n,f);
            s:=0.0;
            for i:= 1 step 1 until m do
            s:=(y[i]-f[i])↑2+s;
            if s < S then
            begin S:=s; minx:=x; ang:=2×j;
                  for i:= 1 step 1 until n do
                  r[i]:=c[i];
            end;
      end;
      if l < 3 and j=1 then
      begin ang:=0; goto M end;
end;
M:theta:=ang×T/180.0; b[1]:=cos(theta); b[2]:=sin(theta)
SELECT(1,a,b,minx);
end ESTIMATE;
integer n,m,i,j,k,l,case,ang,N,f1,f2; real minx,S,in,d,li;
```

```
open(20);  open(70);

f1:=format([ndds;]); f2:=format([4+d.dddsddds₁₀+nd;]);

N:=read(20);

for k:= 1 step 1 until N do

begin write text(70,[[2c]DATA*SET*NUMBER*****]);

      copy text(20,70,[;;]);

      m:=read(20); n:=read(20); l:=read(20);

      d:=read(20); in:=read(20); in:=read(20);

      li:=read(20); case:=read(20);

      begin array t,y,w[1:m],a,r[1:n];

            forj:= 1 step 1 until m do

            t[j]:=read(20);

            for j:= 1 step 1 until m do

            begin y[j]:=S:=read(20);

                  w[j]:=1.0/S↑(case-1);

               end;

            ESTIMATE(t,y,w,a,l,ang,n,m,minx,S,r,d,in,li);

            write text(70,[[2c]WEIGHTING*FACTOR**]);

            write(70,f1,case);

            write text(70,[[2c]THE*COEFFICIENTS]);

            for i:= 1 step 1 until n do

            write(70,f2,r[i]);

            write text(70,[[2c]THE*EXPONENTS***]);

            for i:= 1 step 1 until n do

            write(70,f2,a[i]);

            write text(70,[[2c]MINIMUM*OF*S****]);

               write(70,f2,S);
```

```
                    write text(70,[[2c]THE*ANGLE*********]);

                    write(70,f1,ang);

                    write text(70,[[2c]THE*MODULUS*******]);

                    write(70,f2,minx);

            end;

        new line(70,3);

    end;

  close(20);close(70);

end→
```

APPENDIX   D

The Optimum Interval and The Corresponding minimum
Correlation Coefficient Between The Exponential
Functions Program .

INPUT DATA

(i)   m ; (number of observations);

(ii) a ; (first value of EX1);

(iii) b ; (increment in EX1);

(iv) p ; (last value of EX1);

(v)   c ; (first value of EX2);

(vi) d ; (increment in EX2) ;

(vii) q ; (last value of EX2);

(viii)inc;(starting value of intervale   );

(ix) G ; (is either 0 If there is only one set of data to be

             run or 1 if there is more than one);

OUTPUT

    The program outputs tables like table (4) , for examle .

```
begin  library AC,A6,A12;
       comment this program calculates the minimum correlation
coefficients between exp(-kxt) and exp(-lxt), and the
corresponding optimum intervals of t  for different values
of k and l;
real procedure CORCOEF (k,l,m,t);
value m,k,l,t;  integer m,k,l; array t;
begin integer j; real X1,X2,S1,S2,S11,S22,S12,COV,V1,V2;
       S1 := S2 := S11 := S22 := S12 := 0.0;
       for j := 1 step 1 until m do
       begin X1 := exp(-kxt[j]);  X2 := exp(-lxt[j]);
             S1  := X1+S1;           S2  := X2+S2;
             S11:= X1xX1+S11; S12 := X1xX2+S12;
             S22:= X2xX2+S22;
          end;
       V1 := S11-S1xS1/m; V2 := S22-S2xS2/m; COV := S12-S1xS2/m
       CORCOEF := COV/sqrt(V1xV2);
    end CORCOEF;
procedure BESTINT (I,m,int,t);
value m,I,int; integer m,I; real int; array t;
begin  integer i;
       if I = 1 then
       begin for i := 2 step 1 until m do
             t[i]  := (i-1)xint;
          end else
       begin for i := 2 step 1 until m do
```

```
                    t[i] := 2↑(i-2)xint;

            end;

      end BESTINT;

   procedure MININTANDCOR (k,l,m,f,inc);

   value k,l,m,f,inc; integer k,l,m,f; real inc;

   begin integer I; real y0,y1,y2,s,h,mini,mincor,int;

            array t[1:m];

            t[1] := 0.0;  s := inc;

            for I := 1 step 1 until 2 do

            begin

            M     :  mincor:=1.0; int:=0.0;

            L     :  int:=int+inc;  BESTINT(I,m,int,t);

                     if CORCOEF(k,l,m,t) > mincor then goto P;

                     mincor:=CORCOEF(k,l,m,t);  goto L;

            P     :  h:=int-inc;  if abs(h-inc) > 10-6 then goto R;

                     inc  := 0.5Xinc;  goto M;

            R     :  BESTINT(I,m,h,t);      y0:=CORCOEF(k,l,m,t);

                     BESTINT(I,m,int,t);   y1:=CORCOEF(k,l,m,t);

                     mini:=h-inc;

                     BESTINT(I,m,mini,t); y2:=CORCOEF(k,l,m,t);

                     int:=y1-2.0Xy0+y2;    mini:=h-0.5Xincx(y1-y2)/int

                     BESTINT(I,m,mini,t);  mincor:=CORCOEF(k,l,m,t);

                     write(70,f,mini); write(70,f,mincor); inc := s;

            end;

      end MININTANDCOR;

   integer k,l,m,G,f1,f2,a,b,c,d,p,q;  real inc;
```

```
        open(20); open(70);
        f1 :=format([2sndd]); f2 :=format([3sd.dddd]);
NEXT :  m :=read(20); a :=read(20); b:=read(20); p:=read(20);
        c :=read(20); d :=read(20); q:=read(20); inc:=read(20);
        G:=read(20);
        write text(70,[[p]**NUMBER*OF*OBSERVATIONS*=*]);
        write(70,f1,m); new line(70,3);
        write text(70,[[13s]EQUAL*INTERVAL[5s]EXPON*INTERVAL[cc]]);
        write text(70,[ **EX1 **EX2 ***MININT ***MINCOR ***MININT ***MINCOR
        [c]]);
        for k := a    step b until p do
        for l := c+k step d until q do
        begin write(70,f1,k);  write(70,f1,l);
              MININTANDCOR(k,l,m,f2,inc); new line(70,1);
           end;
        if G ≠ 0 then goto NEXT;
        close(20);  close(70);
    end →
```