TRACER ANALYSIS IN A FRACTURED GEO'MERMAL RESERVOIR: FIELD RESULTS FROM WAIRAKEI, NEW ZEALAND

A REPORT

SUBMITTED TO THE DEPARMENT OF PETROLEUM ENGINEERING

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

by

Martin Peter Fossum

June 1982

ACKNOWLEDGMENTS

I would like to thank the Department of Petroleum Engineering and the Stanford Geothermal Project for their generous support. Professor Roland Horne provided invaluable guidance without which this work would not have been possible.

Approved: _____
Principal Research Advisor

Date: _____

To Mo and my parents.

TABLE OF CONTENTS

Chapter I

INTRODUCTION

Interwell tracers have been used extensively in oil reservoirs to detect reservoir heterogeneities. High permeability production zones can channel a disproportionate amount of fluid between an injector and producer. This reduces the vertical sweep efficiency in waterfloods and causes early breakthrough of water in producing wells.

Analogously, tracer analysis is important in determining geothermal reservoir characteristics. Most importantly, it provides a means of identifying cold reinjection water flowing to producer wells.

In Japan, rapid interference from reinjection wells has caused thermal drawdown in four of five liquid-dominated fields.' Although maintaining reservoir pressure is one motive for total injection of waste water, the reduction in enthalpy and subsequent decrease in steam discharge rates may imply that maintenance of discharge enthalpy is more important.

Tracer analysis will be instrumental in identifying the relative importance of the flow paths which make enthalpic reduction possible. Unfortunately, models currently used for oil reservoirs cannot be applied to geothermal system. Oil field models assume flow through porous media. Geothermal reservoirs, however, are usually highly fractured. Since the preferred flow channels of reinjected water will be

through intervening fractures between producer and injector, a method of determining the extent and type of fracturing is needed.

A tracer model to characterize flow in a fractured geothermal system is developed in this paper. It can be used to:

(a) Investigate possible damage from premature reservior cooling;

(b) Determine natural flow patterns within the reservoir;

(c) Predict interference from current injection wells; and

(d) Plan reinjection programs to minimize thermal drawdown.

Recent field test data from tracer tests in Wairakei, New Zealand, are analyzed and compared. Although the main surface fault system is well identified at Wairakei, determining whether interwell flow is primarily within the fault or through other fractures is important before a reinjection program can be evaluated.

This study illustrates how tracers can be used to understand not only the nature of fracture systems, but also their relative influence as preferred pathways for well-to-well flow. It is a vital first step in designing a reinjection program.

Failure to recognize preferential flowpaths before reinjection could result in substantial financial losses for the operator and permanent damage to the reservoir. Proper application of tracer techniques can reduce the probability of this damage occurring.

Chapter II

LITERATURE SURVEY

Interwell tracer surveys are often used in the oil industry. In order to quantify these survey results, Brigham and Smith[1] formulated a model to predict the time of tracer breakthrough, peak concentration of the tracer, and the general form of the breakthrough curve in a five-spot flood. Baldwin[2] developed a similar model by dividing the five-spot pattern into a series of diverging-converging radial elements between the injector and the producer. Yuen[3] then constructed a computer algorithm to test for the degree of heterogeneity between different reservoir layers. Recent work by Abbaszadeh-Dehghani[4] has extended and refined this approach. Field results were compiled by Wagner.[5]

However, since most geothermal reservoirs are highly fractured, the above-mentioned models are not applicable. The five-spot model assumes flow through a porous medium. Tester, Bivins and Potter[6] used tracer profiles and residence time distributions to analyze a hydraulically fractured granitic geothermal reservoir. This was the first attempt at formulating an approach to the problem of flow in fractures. A multi-zone model assuming independent, one-dimensional dispersion in several separate zones was used. This, however, still assumed flow in porous media.

Recent reinjection results in Japan[7] have showed that thermal draw-down from reinjected waste water is often encountered. Drawing on this experience in highly fractured fields, Horne and Rodriguez[9] developed a preliminary model to characterize flow mechanisms in fractures. The model yields information on effective tracer dispersion in the fractured system. These results can be used to appropriately interpret tracer tests in fractured reservoirs and thereby assist in the planning of reinjection programs.

The work in this report continues the development of a tracer return interpretation procedure. It is based on the mathematical methods of Tester, Bivens and Potter[6] (also used by Abbaszadeh-Dehghani[4]) and the physical description of flow in fractures offered by Horne and Rodriguez.[9]

Chapter III

GEOLOGY

The Wairakei geothermal field is located in a large thermal area on New Zealand's North Island. The field is the longest operated liquid-dominated geothermal reservoir in the world. Figures 3.1 and 3.2 give a general idea of the location and overall geology. 17

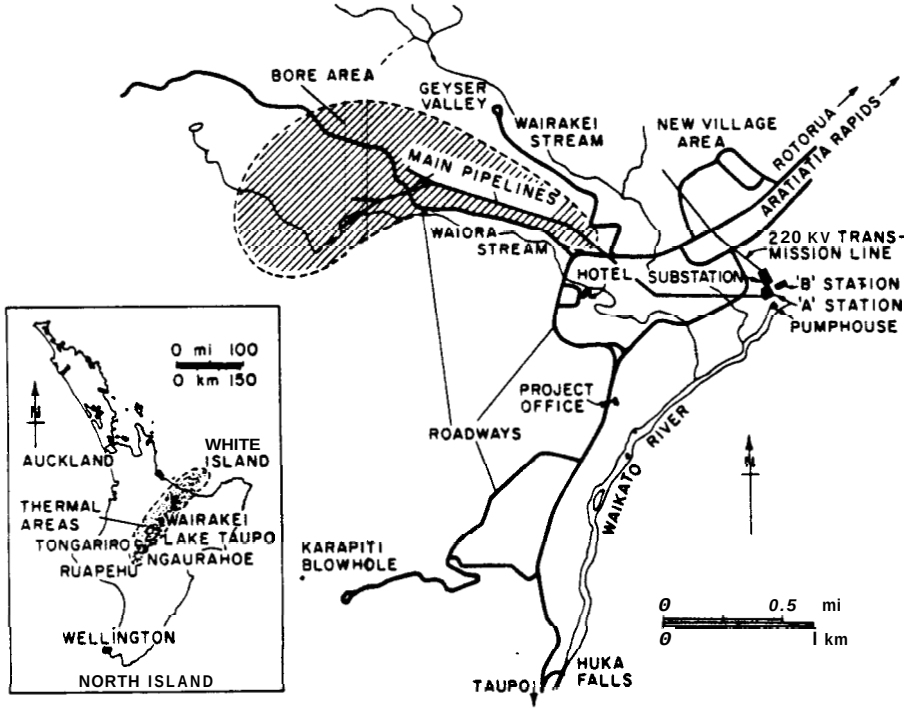


Fig. 3.1: LOCATION OF WAIRAKEI GEOTHERMAL FIELD, NORTH ISLAND, NEW ZEALAND

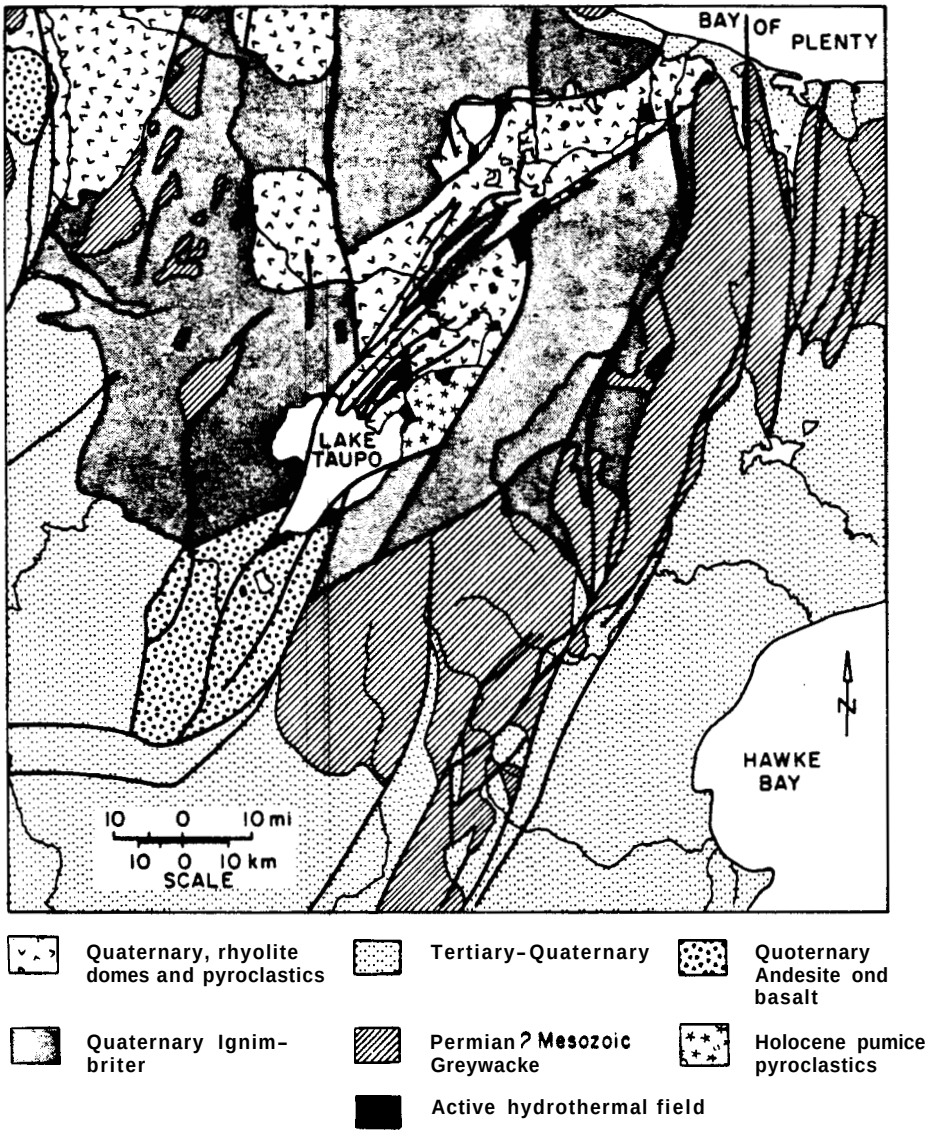


Fig. 3.2: GENERALIZED GEOLOGIC MAP **OF TAUPO** VOLCANIC ZONE, NEW ZEALAND

Of particular importance **is** the cross-section taken east-west across the thermal anomaly. Figures 3.3a and 3.3b show the major faults, Waiora, Kaiapo and Wairakei which serve as feed channels for the hot water aquifer. The thin vertical lines represent the average depth of drilled wells in each fault block. The block which contains the three wells studied is located between the Kaiapo and Wairakei faults. There is also some transverse faulting which will be important in analyzing **flow** paths. **This is** illustrated **in** detail in the empirical section.

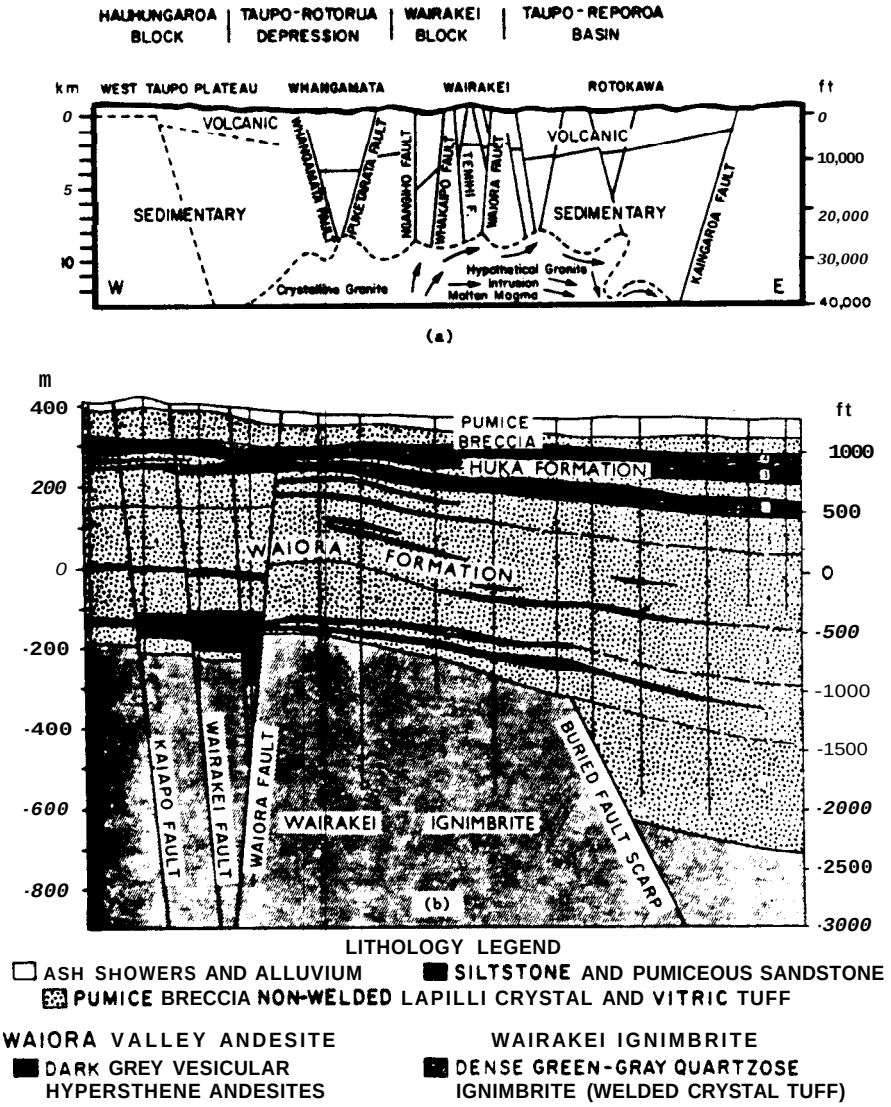


Fig. *3.3:* **(a) GEOLOGIC CROSS-SECTION OF TAUPO THERMAL ZONE AT WAIRAKEI: (b) DETAILED CROSS-SECTION OF WAIRAKEI RESERVOIR**

The reservoir itself is made up of a pumice breccia aquifer (Waiora Formation). Its thickness ranges between *460* **and 900 meters. It is overlain by an impermeable lacustrine mudstone which varies in depth from 60 to 150 meters (Huka formation).**

Most of the field's production comes from the contact area between the Waiora Formation and the underlying Wairaki ignimbrite. This interface lies between 570 and 680 meters.

Chapter IV

THEORY

In order to model the Wairakei field test data, a suitable transfer function had to be derived. Tester, Bivens and Potter[6] proposed a method to characterize the flow paths between injector and producer by analyzing N measured values of the exit tracer concentration C_i. By minimizing the objective function R,

$$R = \sum_{i=1}^{N} (\tilde{C} - C_i)^2 \qquad (1)$$

one can obtain estimates of the parameters of $\tilde{C}$ (where $\tilde{C}$ is the proposed transfer function). The transfer function in turn yields information about the flow channels in the fracture system. The function $\tilde{C}$ represents the produced tracer concentration in terms of dimensionless time $(\bar{u}t/L)$ and the Peclet number, $Pe = uL/\eta$. η is the longitudinal dispersion coefficient.

$$\tilde{C} = \tilde{C}(t; L/\bar{u}, Pe) \qquad (2)$$

The above function will describe flow through one fracture only. A second model represents the tracer profile in terms of M component channels:

$$\tilde{C} = \sum_{j=1}^{M} \varepsilon_j \tilde{C}_j((L/\bar{u})_j, Pe_j) \qquad (3)$$

where ε_j is the fraction of flow in path j.

Since almost all of the field tests observed had at most two peaks, M = 2 was chosen for comparative purposes. Thus:

$$\tilde{C} = \varepsilon_1 \tilde{C}_1((L/\bar{u})_1, Pe_1) + \varepsilon_2 \tilde{C}_2((L/\bar{u})_2, Pe_2) \qquad (4)$$

The two-component signal approach has been successfully applied to match observed tracer profile peaks. It can also be used to analyze the shape of a single peak tracer response. The one- and two-component models were tested to determine whether the tracer return to a producing well is a result of flow through one channel or two. An important point to be stressed is that a single peak profile may not be a result of only one flow path, and therefore may be better modeled assuming multiple flow channels.

Minimizing the objective function R with respect to the linear parameters ε_j and E, and the nonlinear parameters, Pe_j and $(L/\bar{u})_j$, yields estimates of the Peclet numbers and average residence times of the respective flow paths. As noted by Horne and Rodriguez,[9] however, the resultant values depend on the particular transfer function chosen to characterize tracer transport.

Rodriguez,[15] in an unpublished paper, presented an analysis of the primary mechanisms governing tracer dispersion in a single fracture

under laminar flow conditions. This work was based on a paper by Taylor.[16] The conclusions are reproduced here in order to clarify the selection of the transfer function.

The dispersion of the tracer solution, assuming neither losses nor generation of byproducts is influenced by two mechanisms:

(1) Molecular diffusion and

(2) Convective dispersion

Molecular diffusion results when the original fluid comes in contact with the tracer. This creates a concentration gradient which serves as the driving force for the dispersion. Convective dispersion occurs when viscous forces acting on the fluid create a velocity gradient in the transverse direction. Therefore, the fluid in the center of the fracture moves faster than that at the edge.

Since the residence times of a tracer solution in a fracture are small, the question of whether molecular diffusion affects the total dispersion of the fluid is important.

Molecular diffusion can be analyzed by assuming that the convective dispersion occurs before any diffusion effects appear. This gives rise to the tracer distribution shown below.

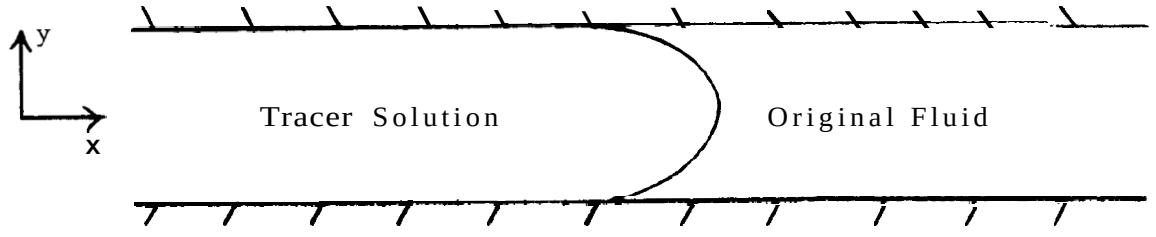


Fig. *4.1*

The profile can be approximated by a series of slabs:

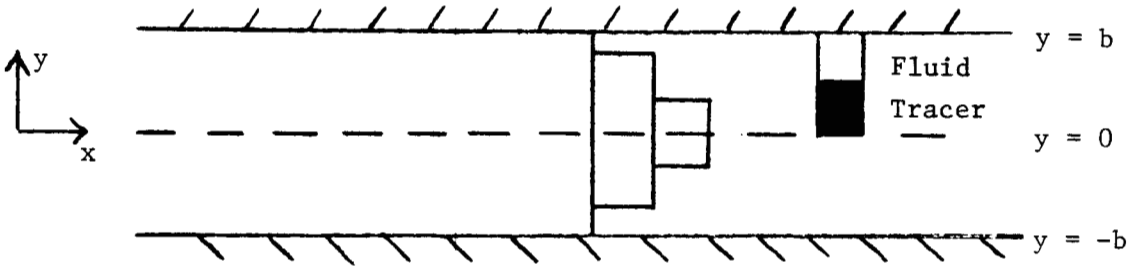


Fig. *4.2*

With this formulation, the problem is symmetric in y and only one half of the fracture thickness needs to be considered.

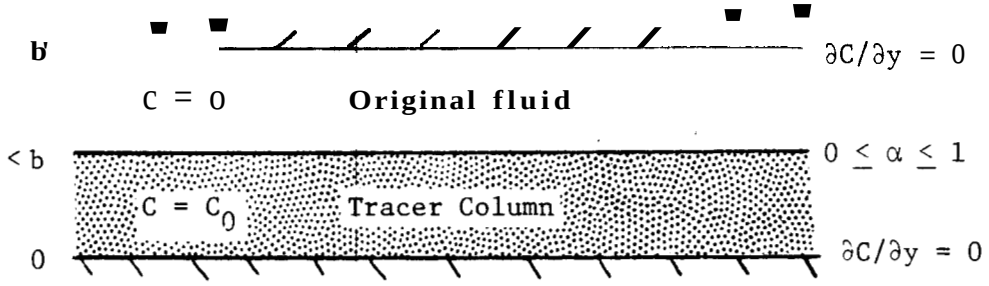


Fig. *4.3*

The one dimensional molecular diffusion problem, as sketched above, is mathematically represented by the following equations:

$$\frac{\partial^2 C}{\partial y^2} = \frac{\partial C}{\partial t} \qquad 0 < y < 1 \ , \ t > 0 \tag{5}$$

where D is the molecular diffusivity.

Initial Conditions :

$$C(y,0) = \begin{cases} C_0 & 0 \leq y \leq \alpha b \\ 0 & \alpha b \leq y \leq b \end{cases} \tag{6}$$

Boundary Conditions:

$$\frac{\partial C}{\partial y} = 0 \qquad \text{at } y = 0$$

$$\frac{\partial C}{\partial y} = 0 \qquad \text{at } y = b \tag{7}$$

If the following dimensionless parameters are used,

$$y_d = y/b$$

$$C_d = C/C_0 \tag{8}$$

$$t_d = (D/b^2)t$$

Equations 5 through 8 can be written as:

$$\frac{\partial^2 C_d}{\partial y_d^2} = \frac{\partial C_d}{\partial t_d} \tag{9}$$

$$C_d(y,0) = \begin{cases} 1 & 0 \le y_d \le \alpha \\ 0 & \alpha \le y_d \le 1 \end{cases} \tag{10}$$

$$\frac{\partial C_d}{\partial y_d} = 0 \quad \text{at} \quad y_d = 0 \tag{11}$$

$$\frac{\partial C_d}{\partial y_d} = 0 \quad \text{at} \quad y_d = 1 \tag{12}$$

By applying the method of separation of variables, a solution to Eqs. 9 through 12 is given by:

$$C(y_d,t_d) = \alpha + \frac{2}{\pi} \sum_{n=1}^{\infty} e^{-n^2\pi^2 t_d} \frac{\sin(n\pi\alpha)\cos(n\pi y_d)}{n} \tag{13}$$

One can use the difference between the concentration at the centerline and the wall as an indication of the tendency for the concentration to become uniform.

$$\delta(t_d) = C_d(0,t_d) - C_d(1,t_d) \tag{14}$$

Substituting Eq. 13 into Eq. 14:

$$\delta(t_d) = \frac{4}{\pi} \sum_{n=1}^{\infty} e^{-(2n-1)^2\pi^2 t_d} \frac{\sin\{(2n-1)\pi\alpha\}}{(2n-1)} \tag{15}$$

Figure 4.4[9] shows $\delta(t_d)$ vs t_d for different values of slug widths ***(a)***.

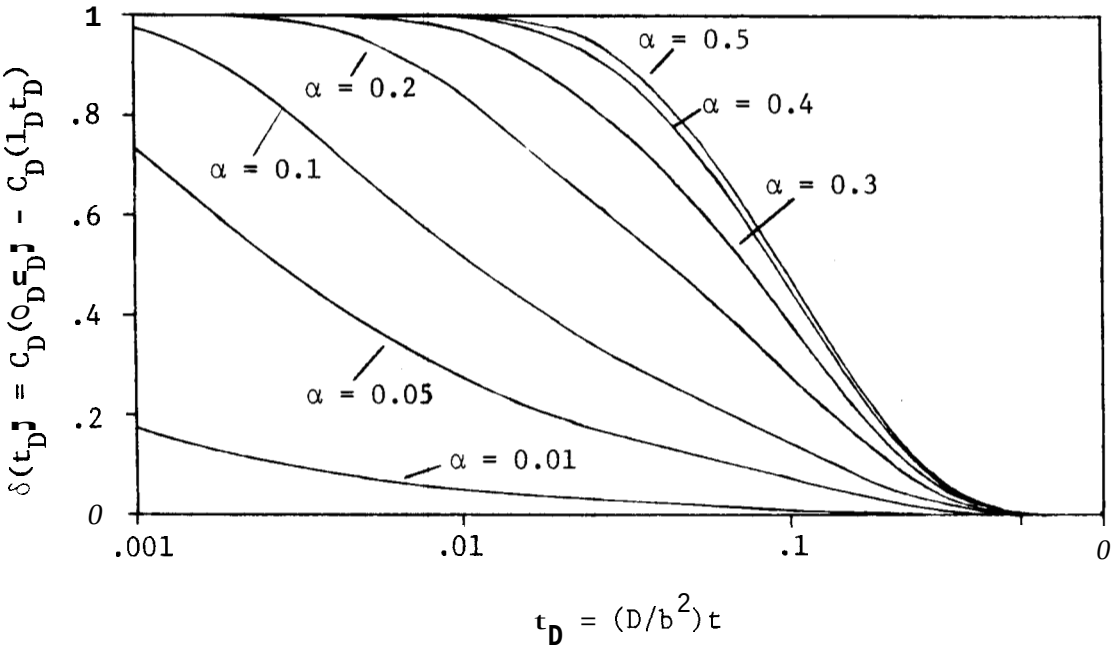


Fig. *4.4:* CONCENTRATION DIFFERENCE BETWEEN WALL AND CENTERLINE AS A FUNCTION OF TIME AND TRACER PENETRATION ACROSS THE FRACTURE

As the graph shows, for all slug widths ***(a)***, the concentration across the fracture width becomes uniform when t_d = **0.5**. If we assume average values for D and b, t_d = **0.5** can be used as an estimate of the actual time it takes for the concentration to become uniformly distributed. Letting 2b = **0.1** cm and **D** = 1×10^{-5} cm^2/sec,

$$t = \frac{(.05)^2\ 0.5}{1 \times 10^{-5}} = 125 \text{ sec}$$

Thus in a 1 mm fracture, the tracer will be totally dispersed across the width of the fracture within 125 seconds. Therefore molecular diffusion in the transverse direction is an important component in the dispersion of the tracer solution in the fracture.

Analagously, for a 500 meter long fracture it can be shown that for molecular diffusion in the axial direction,

$$t_d = \frac{1 \times 10^{-5} \times (90 \text{ min}) \times 60}{(500 \times 100)^2} = 2.16 \times 10^{-11}$$

Since this figure is 10 orders of magnitude less than that required for dispersion across the fracture, molecular diffusion in the direction of flow is an unimportant factor.

Thus, for field cases, convective dispersion will be dominated by molecular diffusion in the transverse direction. This "Taylor Dispersion" results in the tracer front propagating at the average speed of flow even though the centerline fluid in the fracture is moving faster.

The differential equation that applies is:

$$D \frac{\partial^2 C}{\partial y^2} - u(y) \frac{\partial C}{\partial x} = \frac{\partial C}{\partial t} \qquad (16)$$

This assumes molecular diffusion in the direction perpendicular to flow and convection of tracer in the direction of flow at a distance x from the injector.

An expression for u(y) has been derived by Horne.[8]

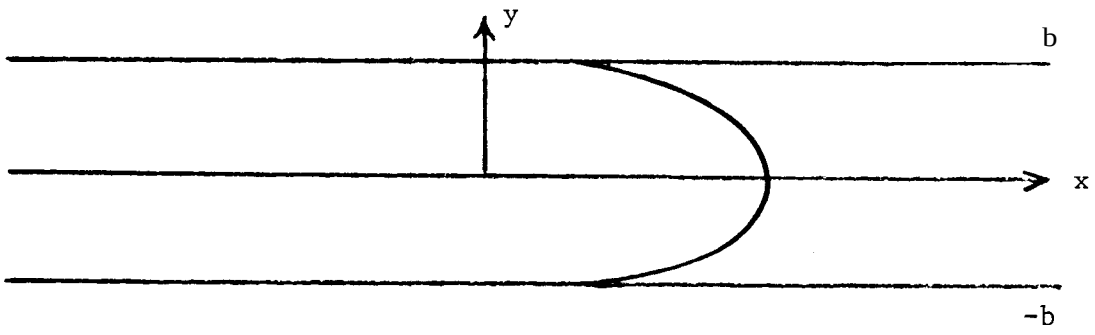


Fig. 4.5

For one-dimensional flow in the x direction, the momentum equation for flow between parallel plates assuming a boundary layer is;

$$\frac{\partial P}{\partial x} = \mu \frac{\partial^2 u}{\partial y^2} \tag{17}$$

Boundary conditions:

$$u = 0 \quad \text{if } y = -b \tag{18}$$

$$y = b \tag{19}$$

Integrating and applying the above conditions yields:

$$u(y) = \frac{3}{2} \bar{u} \left(1 - \frac{y^2}{b^2} \right) \tag{20}$$

Substituting this velocity profile into the **PDE**, gives:

$$D \frac{\partial^2 C}{\partial y^2} - \frac{3\bar{u}}{2} \left(1 - \frac{y}{b^2}\right) \frac{\partial C}{\partial x} = \frac{\partial C}{\partial t} \tag{21}$$

Initial Condition:

$$C(x,y,0) = 0 \tag{22}$$

Boundary Conditions:

$$\frac{\partial C}{\partial y} = 0 \quad \text{at} \quad y = 0$$

$$\frac{\partial C}{\partial y} = 0 \quad \text{at} \quad y = l \tag{23}$$

$$C(0,y,t) = C_0 \quad \text{at} \quad x = l$$

Using the dimensionless parameters:

$$y_d = y/b$$

$$x_d = x/L \tag{24}$$

and

$$u_d = (b^2 u/DL)^2 \tag{25}$$

the equations are written:

$$\frac{\partial^2 C_d}{\partial y^2} - \frac{3u_d}{2} (1 - y_d)^2 \frac{\partial C_d}{\partial x_d} = \frac{\partial C_d}{\partial t_d} \tag{26}$$

Initial Condition:

$$C_d(x_d, y_d) = 0 \tag{27}$$

Boundary Conditions:

$$\frac{\partial C_d}{\partial y_d} = 0 \qquad \text{at} \quad y_d = 0 \tag{28}$$

$$\frac{\partial C_d}{\partial y_d} \qquad \text{at} = 0 \quad y_d = 1 \tag{29}$$

$$C_d(0, y_d, t_d) = 1 \qquad \text{at} \quad x_d = 0 \tag{30}$$

To model flow in the x direction, consider a moving x'_d coordinate which moves at the average dimensionless velocity $\bar{u}_d$:

$$x'_d = x_d - \bar{u}_d t_d \tag{31}$$

The effective velocity $u'_d(y)$ related to the moving plane x'_d at speed $u(y)$ is:

$$u'_d(y) = u_d(y) - \bar{u}_d(y) \tag{32}$$

$$= \frac{3}{2} u_d(1 - y^2) - \bar{u}_d \tag{33}$$

Thus,

$$u'_d(y) = \frac{3}{2} \bar{u}_d \left(\frac{1}{3} - \frac{y}{b^2}\right) \tag{34}$$

Substituting,

$$\frac{\partial^2 C_d}{\partial y_d^2} - \frac{3}{2} u_d \left(\frac{1}{3} - y_d^2\right) \frac{\partial C_d}{\partial x'_d} = 0 \tag{35}$$

Initial Conditions:

$$C_d(x_d, y_d, 0) = 0 \tag{36}$$

Boundary Conditions:

$$\frac{\partial C_d}{\partial y_d} = 0 \qquad \text{at} \quad y_d = 0 \tag{37}$$

$$\frac{\partial C_d}{\partial y_d} = 0 \qquad \text{at} \quad y_d = 1 \tag{38}$$

$$C(-ut_d, y_d) = 1 \tag{39}$$

If, on average,

$$\frac{\partial C_d}{\partial x'_d} = \frac{\partial \overline{C}_d}{\partial x'_d} \tag{40}$$

where $\overline{C}_d$ is the average concentration across the fracture thickness, then $\frac{\partial C_d}{\partial x'_d}$ is only a function of x'_d. Substituting,

$$\frac{\partial^2 C_d}{\partial y_d} - \frac{3}{2} u_d \left(\frac{1}{3} - y_d^2\right) \frac{\partial C_d}{\partial x'_d} = 0 \tag{41}$$

A solution to Eq. 41 satisfying the Boundary Conditions is:

$$C_d(x'_d, y_d) = \frac{1}{4} \bar{u}_d \frac{\partial \bar{C}}{\partial x'_d} \left\{ y_d^2 - \frac{y_d^4}{2} \right\} + C_{d_o} \tag{42}$$

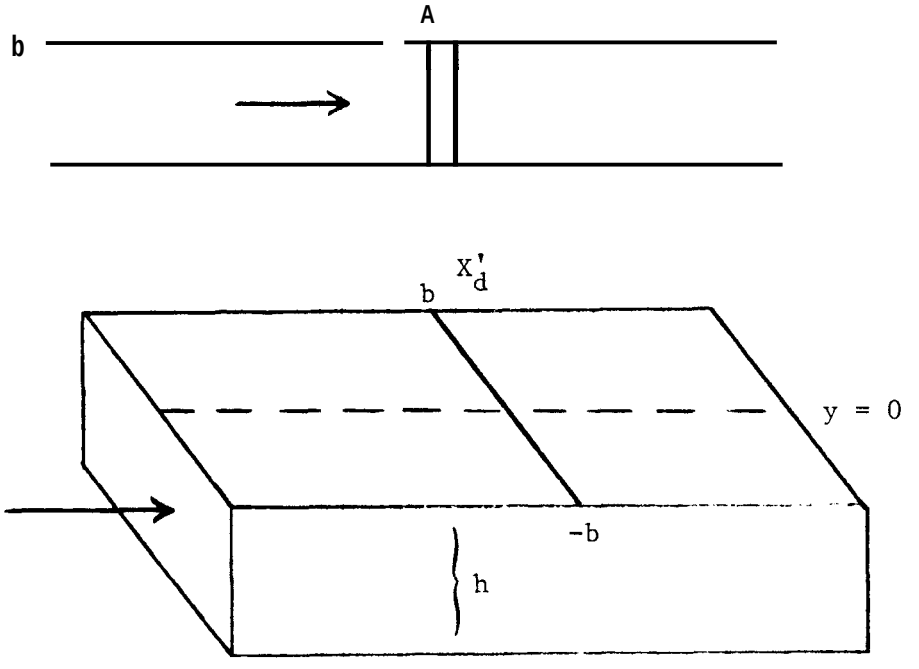


Fig. 4.6

The rate of mass transfer of C_d across the plane AB in Fig. 4.6 at x'_d is:

$$h \int_0^b u(y)\, c\, dy = bh(\overline{uc}) = q \tag{43}$$

where h is the height of the fracture and q is the volumetric flow rate.

Equations 42 through 43 can be expressed as:

$$q_d = \int_0^1 u_d C_d dy_d \tag{44}$$

where:

$$q_d = \frac{b}{C_0 h} q \tag{45}$$

and, recalling Eq. *34:*

$$u'_d = \frac{3}{2} u_d \left(\frac{1}{3} - y_d^2\right) \tag{46}$$

Substituting Eqs. *46* and *42* into Eq. *44* yields:

$$q_d = -\frac{2}{105} \bar{u}_d^2 \frac{\partial \bar{C}_d}{\partial x'_d} \tag{47}$$

Using a material balance, one can see that the change in volumetric rate out of the system must equal the rate of accumulation $\partial \bar{C}_d / \partial t_d$.

So,

$$-\frac{\partial q_d}{\partial x'_d} = \frac{\partial \bar{C}_d}{\partial t_d} \tag{48}$$

Thus,

$$\eta_d \frac{\partial^2 C_d}{\partial x'^2_d} = \frac{\partial \bar{C}_d}{\partial t_d} \tag{49}$$

where :

$$\eta_d = \frac{2}{105} \bar{u}_d^2 \tag{50}$$

This can also be written in dimensional form:

$$\eta = \frac{2}{105} \left(\frac{b^2 \bar{u}}{DL} \right)^2 \tag{51}$$

and then the differential equation can be written:

$$\eta \frac{\partial^2 C}{\partial x^2} = \frac{\partial C}{\partial t} \tag{52}$$

where η is the effective longitudinal dispersion coefficient for the fracture.

As the above derivation by Rodriguez'[5] shows, the concentration C is dispersed relative to a plane which moves with a mean velocity $\bar{u}$, even though the maximum velocity is at the center of the fracture (at y = 0) and is equal to 3/2 $\bar{u}$. Therefore, the longitudinal diffusion process follows the same law as molecular diffusion but with a dispersion coefficient η.

A solution to Eq. 52 with a material of mass s, concentrated at a point x = 0, at time t = 0, is:

$$C(t;\eta,L) = \frac{sx}{2\sqrt{\pi \eta t}} e^{-\frac{(x-\bar{u}t)^2}{4\eta t}} \tag{53}$$

The exit concentration as a function of time is given by substituting x = L into (53).

$$C(t;\eta,L) = \frac{sL}{2J\eta\sqrt{\eta t}} e^{-\frac{(L-ut)^2}{4\eta t}} \tag{54}$$

The above equation can he rewitten in terms of dimensionless time $(\bar{u}/L)t$ and the Peclet number $\bar{u}L/\eta$. This equation then represents the concentration of tracer as it passes the production well recording point.

Thus, the general transfer function for one flow path is:

$$\tilde{C}(t;(L/\bar{u}),Pe) = \frac{s}{2\sqrt{\pi(\eta/uL)(u/L)}} e^{-\frac{(1-(\bar{u}/L)t)^2}{4(\eta/uL)(\bar{u}/L)t}} \tag{55}$$

and for two flow paths:

$$C(t;(L/u)_j,Pe_j) = \frac{s}{2\sqrt{\pi(\eta/\bar{u}L)_1(\bar{u}/L)_1}} e^{-\frac{(1-(\bar{u}/L)_1 t)^2}{4(\eta/\bar{u}L)_1(\bar{u}/L)_1 t}} + \frac{s}{2\sqrt{\pi(\eta/\bar{u}L)_2(\bar{u}/L)_2}} e^{-\frac{(1-(u/L)_2 t)^2}{4(\ /uL)_2(u/L)_2 t}} \tag{56}$$

In the New Zealand field case, a quantity of 1–131 radioactive tracer was injected instantaneously into the flowstream. The above transfer function and its parameters will give information about the fracture system and degree of dispersion by analyzing the shape of the tracer profile. For example, Figs. *4.7*, *4.8* and *4.9* indicate possible tracer return profiles.

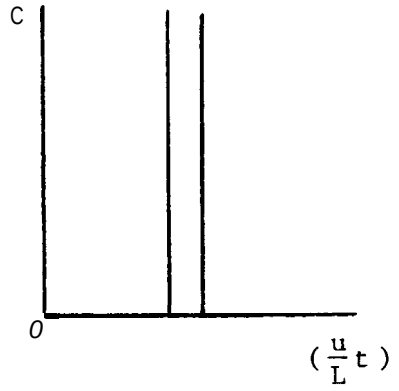


Fig. *4.7*

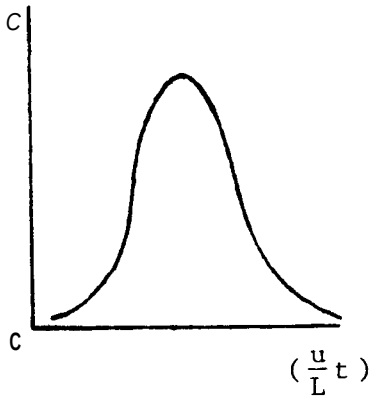


Fig. *4.8*

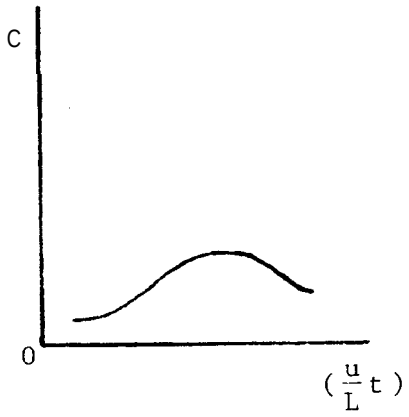


Fig. *4.9*

Figure *4.7* exhibits piston–like flow without dispersion. Figure *4.9* shows more dispersion than **Fig.** *4.8.* The parameter $\frac{\eta}{uL} = Pe^{-1}$ determines the degree of dispersion and hence the shape of the profile. If $\frac{\eta}{uL}$ is small (Peclet number large), a profile like Fig. *4.8* will emerge. **If,** however, $\frac{\eta}{uL}$ is large, the curve will he more like Fig. *4.9.* Physically, this implies that when the Peclet number is large, the tracer profile does not change much during the time interval necessary for the fluid to reach the detection point. If $\frac{\eta}{uL}$ approaches a value close to 0.01, the profile flattens out and its shape becomes skewed.

The transfer function derived in this way can then be used to analyze the observed tracer response in order to estimate the Peclet numbers, flow fractions and residence times in each of the flow paths.

Chapter V

OPTIMIZATION: TEST PROGRAM/VARPRO

The parameters of $\tilde{C}$ were estimated using a non-linear least-squares method of curve-fitting. The main program incorporates the subroutine Varpro (from the Stanford Center for Information Technology) which computes the optimal values for both linear and non-linear parameters of a given fitting function. The input data required are:

1) N observed values (C_i);

2) Time, t; and

3) Estimates of the non-linear parameters.

These are entered as:

i) $\alpha_1 = Pe_1^{-1}$

ii) $\alpha_2 = \left(\frac{u}{L}\right)_1$

iii) $a_3 = Pe_2^{-1}$

iv) $\alpha_4 = \left(\frac{u}{L}\right)_2$

Varpro is based on a paper by Golub and Pereya.[10] They showed that a least-squares fit of non-linear models of the form:

$$\tilde{C}(\varepsilon,\alpha_i,t) = \sum_{j=1}^{M} \varepsilon_j \tilde{C}_j(\alpha_i;t) \qquad i = 1,2,3,4$$

where:

t = independent variable

C_j = observed dependent variable

E_j = linear parameter

a_i = non-linear parameter

can be accomplished by first optimizing with respect to ε_j.

The functional,

$$R(\varepsilon_j,\alpha_j) = \sum_{i=1}^{N} (C_i - \tilde{C}(\varepsilon_j,\alpha_j,t))^2$$

is minimized by first assuming values for α_j. Then linear least-squares are applied to the residual R to obtain E_j. After the linear parameters are computed, the residual can be modified by substituting the optimal estimates for ε_j and then minimizing with respect to a_j. When the optimal non-linear parameters have been computed, the linear parameters can be recovered. A proof of this technique is given in the Golub paper.[10]

The numerical routine utilizes a Taylor expansion of the transfer function $\tilde{C}$ by expanding around the α_j. Linear least-squares are then used to determine the optimum values for the parameter increments, $\delta\alpha_j$.

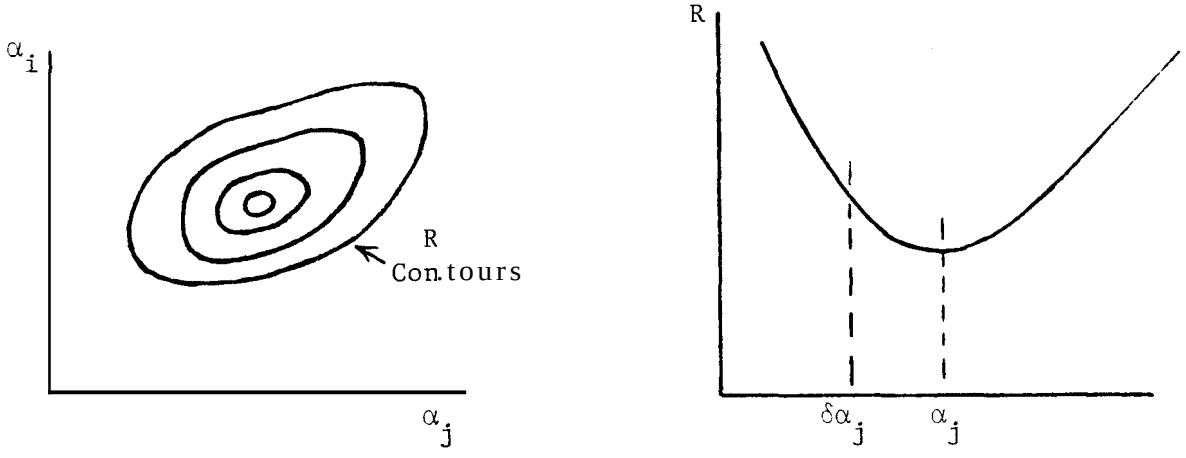
$$\tilde{C}(\alpha_j,\varepsilon_j;t_i) - \tilde{C}_0 = \sum_{j=1}^{M} \left(\frac{\partial C_{0_j}}{\partial \alpha_j} \delta\alpha_j \right) \qquad i = 1,N$$

The derivatives are evaluated at the starting point $C_{\theta'}$. The residual can then be expressed as:

$$R(\varepsilon_j,\alpha_j) = \sum \left((C_i - C_o) - \sum_{i=1}^{N} \frac{\partial C_o}{\partial \alpha_j} \delta\alpha_j\right)^2$$

Applying least-squares then yields a set **of** normal equations.

A gradient-expansion method is used to search for those parameters a_j that minimize $R(\varepsilon_j,\alpha_j)$:



All parameters are incremented simultaneously so that the maximum variation **of** R is attained. The gradient (VR) determines the magnitude of the largest change, and giving it the opposite direction indicates the path of steepest descent. The main idea is to change δa_j so that $R(\varepsilon_j,\alpha+\delta\alpha_j) \leq R(\varepsilon_j,\alpha_j)$. This is documented in detail in the main program.

A detailed test program used to verify that Varpro and access subroutines were functioning properly is included in the remainder **of** this section.

```
C
C
C                  THE MAIN PROGRAM
C
C       A TEST CASE FOR FITTING A LINEAR COMBINATION OF
C             EXPONENTIAL FUNCTIONS
C
C
C
C
C          THIS IS A TEST PROGRAM WHICH UTILIZES THE NONLINEAR LEAST-
C         SQUARES SUBROUTINE VARPRO. THE TEST FUNCTION
C
C         ETA(A,ALF;T)=BETA1 + BETA2*EXP(-ALF1*T) + BETA3*EXP(-ALF2*T)
C
C                      BETAI = LINEAR PARAMETERS
C                      ALFI  = NONLINEAR PARAMETERS
C                       T    = INDEPENDENT VARIABLE
C
C         IS USED TO FIT THE DATA FROM A CHEMISTRY EXPERIMENT IN
C         A PAPER BY OSBORNE.( REFERENCED IN THE INTRODUCTION)
C         THE OBJECTIVE OF THIS PRELIMINARY RUN IS TO MATCH THE
C         RESULTS OBTAINED IN OSBORNE'S PAPER. THESE RESULTS
C         ARE ATTACHED FOR COMPARISON.
C
C         THE FLOW SEQUENCE OF THE MAIN PROGRAM IS AS FOLLOWS:
C
C               * DATA IS READ FROM THE ORVYL FILE TRACERDATA
C
C               * SUBROUTINE ADA COMPUTES THE INCIDENCE MATRIX
C                 AND THE DERIVATIVES OF ETA.
C
C               * SUBROUTINE VARPRO COMPUTES THE OPTIMAL ESTIMATES
C                 OF THE LINEAR AND NONLINEAR PARAMETERS.
C
C               * RESULTS ARE PUT INTO THE ORVYL FILE TESTCASE
C
C ***************************************************************
C ***************************************************************
C
C                  PROGRAM BEGINS
C
C ***************************************************************
C ***************************************************************
C
C
C$ASSIGN 1 TO FILE TRACERDATA
C$ASSIGN 2 TO FILE RESULTS INOUT FORMAT=F LENGTH=80
C$ASSIGN 3 TO FILE TESTCASE INOUT FORMAT=F LENGTH=8000
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMENSIONS FOR THE INCEDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(33),T(33),ALF(2),BETA(10),W(100),A(33,7),
     *INC(12,3)
C
C
C         SET PARAMETERS FOR VARPRO.
```

```
C
C
      EXTERNAL ADA
      NMAX=33
      LPP2=7
      IPRINT=1
      IV=1
      N=33
      NL=2
      L=3
      DO 1 I=1,33
1     W(I)=1.0
C
C
C        READ DATA FROM TRACERDATA. SEQUENTIAL ORDERING AND
C        PROPER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (1,10) NL
 10   FORMAT(I3)
      WRITE(2,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C       ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (1,15)(ALF(I),I=1,2)
 15   FORMAT(F4.2)
      WRITE(2,20)(ALF(I),I=1,2)
 20   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F4.2))
C
C
C       L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      READ(1,22) L
 22   FORMAT(I3)
      WRITE(2,25)L
 25   FORMAT(1H0,10X,'NUMBER OF LINEAR PARAMETERS'//(I3))
C
C
C       N IS THE NUMBER OF OBSERVATIONS
C
C
      READ(1,30) N
 30   FORMAT(I4)
      WRITE(2,35) N
 35   FORMAT(1H0,10X,'NUMBER OF OBSERVATIONS'//(I3))
C
C
C       IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      READ(1,38) IV
 38   FORMAT(I4)
```

```
      WRITE(2,40) IV
 40   FORMAT(1H0,10X,' # OF INDEPENDENT VARIABLES T'//(I4))
C
C
C     Y IS THE N-VECTOR OF OBSERVATIONS
C
C
      READ(1,45)(Y(I),I=1,N)
 45   FORMAT(F5.3)
      WRITE(2,50)(I,Y(I),I=1,N)
 50   FORMAT(1H0,'OBSERVATIONS'//(I3,5X,F5.3))
C
C
C     T IS THE INDEPENDENT VARIABLE
C
C
      READ(1,55)(T(I),I=1,N)
 55   FORMAT(F7.3)
     WRITE(2,60)(I,T(I),I=1,N)
 60   FORMAT(1H0,'INDEPENDENT VARIABLES'//(I3,5X,F6.1))
C
C
      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
C
      STOP
      END
C
C
C
C
C
C**********************************************************************
C**********************************************************************
C
C                    SUBROUTINES
C
C**********************************************************************
C**********************************************************************
C
C
C
C
      SUBROUTINE ADA (L,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
C$ASSIGN 2 TO FILE RESULTS INOUT FORMAT=F LENGTH=80
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALF(2),A(33,7),T(33),INC(12,8),B(100,10),Y(33
     *)
C
C
C
C
C
C     THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C     INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C     IN THE J-TH FUNCTION PHI(J).  (THE PROGRAM SETS ALL OTHER
C     INC(K,J) TO ZERO.)
C
```

```
C
      INC(1,1)=0.0
      INC(1,2)=1.0
      INC(1,3)=0.0
      INC(2,1)=0.0
      INC(2,2)=0.0
      INC(2,3)=1.0
C
C
      WRITE(2,70)((INC(I,J),J=1,3),I=1,2)
 70   FORMAT(1H0,' INCIDENCE MATRIX   INC(I,J)= '//(I3))
C
C
C      THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
C      THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C      B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C      J=1,L. THE CONSTANT FUNCTIONS PHI WHICH DO NOT
C      DEPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C      APPEAR FIRST.
C
C
      IF(ISEL.EQ.2)  GO TO 105
      IF(ISEL.EQ.3)  GO TO 165
       DO 90 I=1,N
 90    A(I,1)=1.0
      WRITE(2,100)(A(I,1),I=1,N)                    5.4
 100  FORMAT(1H0,'COLUMN #1 OF A(I,J) MATRIX'//(F3.0))
C
      CONTINUE
C
 105      DO 110 I=1,N
 110  A(I,2)=DEXP(-ALF(1)*T(I))
      WRITE(2,120)(A(I,2),I=1,N)
 120  FORMAT(1H0,'COLUMN #2 OF A(I,J) MATRIX'//(F8.5))
C
C
      DO 130 I=1,N
 130  A(I,3)=DEXP(-ALF(2)*T(I))
      WRITE(2,140)(A(I,3),I=1,N)
 140  FORMAT(1H0,'COLUMN #3 OF A(I,J) MATRIX'//(F8.5))
      IF (ISEL.EQ.2) GO TO 200
C
      DO 150 I=1,N
 150   A(I,4)=0.0
      WRITE(2,160)(A(I,4),I=1,N)
 160  FORMAT(1H0,'COLUMN #4 OF A(I,J) MATRIX'//(F3.1))
C
C
 165     DO 170 I=1,N
         B(I,2)=DEXP(-ALF(1)*T(I))
         B(I,3)=DEXP(-ALF(2)*T(I))
 170   A(I,5)=-T(I)*B(I,2)
      WRITE(2,180)(A(I,5),I=1,N)
 180  FORMAT(1H0,'COLUMN #5 OF A(I,J) MATRIX'//(F9.5))
C
C
      DO 190 I=1,N
 190   A(I,6)=-T(I)*B(I,3)
       WRITE(2,195)(A(I,6),I=1,N)
 195   FORMAT(1H0,'COLUMN #6 OF A(I,J) MATRIX'//(F9.5))
```

```
 200      CONTINUE
C
C
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
C
C
C
      SUBROUTINE VARPRO (L, NL, N, NMAX, LPP2, IV, T, Y, W, ADA, A,
     X IPRINT, ALF, BETA, IERR)
C
C          GIVEN A SET OF N OBSERVATIONS, CONSISTING OF VALUES Y(1),
C        Y(2), ..., Y(N) OF A DEPENDENT VARIABLE Y, WHERE Y(I)
C        CORRESPONDS TO THE IV INDEPENDENT VARIABLE(S) T(I,1), T(I,2),
C        ..., T(I,IV), VARPRO ATTEMPTS TO COMPUTE A WEIGHTED LEAST
C        SQUARES FIT TO A FUNCTION ETA (THE 'MODEL') WHICH IS A LINEAR
C        COMBINATION
C                                  L
C       ETA(ALF, BETA; T)  =  SUM  BETA  * PHI (ALF; T) + PHI    (ALF; T
C                             J=1      J      J                 L+1
C
C        OF NONLINEAR FUNCTIONS PHI(J) (E.G., A SUM OF EXPONENTIALS AND
C        OR GAUSSIANS).  THAT IS, DETERMINE THE LINEAR PARAMETERS
C        BETA(J) AND THE VECTOR OF NONLINEAR PARAMETERS ALF BY MINIMIZ-
C        ING
C
C                          2     N                                    2
C           NORM(RESIDUAL)  =  SUM  W  * (Y  - ETA(ALF, BETA; T )) .
C                              1=1   I     I                       I
C
C        THE (L+1)-ST TERM IS OPTIONAL, AND IS USED WHEN IT IS DESIRED
C        TO FIS ONE OR MORE OF THE BETA'S (RATHER THAN LET THEM BE
C        DETERMINED).  VARPRO REQUIRES FIRST DERIVATIVES OF THE PHI'S.
C
C                                 NOTES:
C
C        A)  THE ABOVE PROBLEM IS ALSO REFERRED TO AS 'MULTIPLE
C        NONLINEAR REGRESSION'.  FOR USE IN STATISTICAL ESTIMATION,
C        VARPRO RETURNS THE RESIDUALS, THE COVARIANCE MATRIX OF TIIE
C        LINEAR AND NONLINEAR PARAMETERS, AND THE ESTIMATED VARIANCE OF
C        THE OBSERVATIONS.
C
C        B) AN ETA OF THE ABOVE FORM IS CALLED 'SEPARABLE'.  THE
C        CASE OF A NONSEPARABLE ETA CAN BE HANDLED BY SETTING L = 0
C        AND USING PHI(L+1).
C
C        C) VARPRO MAY ALSO BE USED TO SOLVE LINEAR LEAST SQUARES
C        PROBLEMS (IN THAT CASE NO ITERATIONS ARE PERFORMED).  SET
C        NL = 0.
```

```
C
C        D)  THE MAIN ADVANTAGE OF VARPRO OVER OTHER LEAST SQUARES
C        PROGRAMS IS THAT NO INITIAL GUESSES ARE NEEDED FOR THE LINEAR
C        PARAMETERS.  NOT ONLY DOES THIS MAKE IT EASIER TO USE, BUT IT
C        OFTEN LEADS TO FASTER CONVERGENCE.
C
C
C     DESCRIPTION OF PARAMETERS
C
C        L        NUMBER OF LINEAR PARAMETERS BETA (MUST BE .GE. 0).
C        NL       NUMBER OF NONLINEAR PARAMETERS ALF (MUST BE .GE. 0).
C        N        NUMBER OF OBSERVATIONS.  N MUST BE GREATER THAN L + NL
C                 (I.E., THE NUMBER OF OBSERVATIONS MUST ESCEED THE
C                 NUMBER OF PARAMETERS).
C        IV       NUMBER OF INDEPENDENT VARIABLES T.
C        T        REAL N BY IV MATRIX OF INDEPENDENT VARIABLES.  T(I, J)
C                 CONTAINS THE VALUE OF THE I-TH OBSERVATION OF THE J-TH
C                 INDEPENDENT VARIABLE.
C        Y        N-VECTOR OF OBSERVATIONS, ONE FOR EACH ROW OF T.
C        W        N-VECTOR OF NONNEGATIVE WEIGHTS.  SHOULD BE SET TO 1'S
C                 IF WEIGHTS ARE NOT DESIRED.  IF VARIANCES OF THE
C                 INDIVIDUAL OBSERVATIONS ARE KNOWN, W(I) SHOULD BE SET
C                 TO 1./VARIANCE(I).
C        INC      NL X (L+1) INTEGER INCIDENCE MATRIX.  INC(K, J) = 1 IF'
C                 NON-LINEAR PARAMETER ALF(K) APPEARS IN THE J-TH
C                 FUNCTION PHI(J).  (THE PROGRAM SETS ALL OTHER INC(K, J
C                 TO ZERO.)  IF PHI(L+1) IS INCLUDED IN THE MODEL,
C                 THE APPROPRIATE ELEMENTS OF THE (L+1)-ST COLUMN SHOULD
C                 BE SET TO 1'S.  INC IS NOT NEEDED WHEN L = 0 OR NL = (
C                 CAUTION:  THE DECLARED ROW DIMENSION OF INC (IN ADA)
C                 MUST CURRENTLY BE SET TO 12.  SEE 'RESTRICTIONS' BELOI
C        NMAX     THE DECLARED ROW DIMENSION OF THE MATRICES A AND T.
C                 IT MUST BE AT LEAST MAX(N, 2*NL+3).
C        LPP2     L+P+2, WHERE P IS THE NUMBER OF ONES IN THE MATRIX INC
C                 THE DECLARED COLUMN DIMENSION OF A MUST BE AT LEAST
C                 LPP2.  (IF L = 0, SET LPP2 = NL+2.  IF NL = 0, SET LPP2
C                 L+2.)
C        A        REAL MATRIX OF SIZE MAX(N, 2*NL+3) BY L+P+2.  ON INPUT
C                 IT CONTAINS THE PHI(J)'S AND THEIR DERIVATIVES (SEE
C                 BELOW).  ON OUTPUT, THE FIRST L+NL ROWS AND COLUMNS OF
C                 A WILL CONTAIN AN APPROXIMATION TO THE (WEIGHTED)
C                 COVARIANCE MATRIX AT THE SOLUTION (THE FIRST L ROWS
C                 CORRESPOND TO THE LINEAR PARAMETERS, THE LAST NL TO TH:
C                 NONLINEAR ONES), COLUMN L+NL+1 WILL CONTAIN THE
C                 WEIGHTED RESIDUALS (Y - ETA), A(1, L+NL+2) WILL CONTAI
C                 THE (EUCLIDEAN) NORM OF THE WEIGHTED RESIDUAL, AND
C                 A(2, L+NL+2) WILL CONTAIN AN ESTIMATE OF THE (WEIGHTED
C                 VARIANCE OF THE OBSERVATIONS, NORM(RESIDUAL)**2/
C                 (N - L - NL).
C        IPRINT   INPUT INTEGER CONTROLLING PRINTED OUTPUT.  IF IPRINT I
C                 POSITIVE, THE NONLINEAR PARAMETERS, THE NORM OF THE
C                 RESIDUAL, AND THE MARQUARDT PARAMETER WILL BE OUTPUT
C                 EVERY IPRINT-TH ITERATION (AND INITIALLY, AND AT THE
C                 FINAL ITERATION).  THE LINEAR PARAMETERS WILL BE
C                 PRINTED AT THE FINAL ITERATION.  ANY ERROR MESSAGES
C                 WILL ALSO BE PRINTED.  (IPRINT = 1 IS RECOMMENDED AT
C                 FIRST.)  IF IPRINT = 0, ONLY THE FINAL QUANTITIES WILL
C                 BE PRINTED, AS WELL AS ANY ERROR MESSAGES.  IF IPRINT
C                 - 1, NO PRINTING WILL BE DONE.  THE USER IS THEN
C                 RESPONSIBLE FOR CHECKING THE PARAMETER IERR FOR ERROR.':
```

```
C         ALF      NL-VECTOR OF ESTIMATES OF NONLINEAR PARAMETERS
C                  (INPUT).  ON OUTPUT IT WILL CONTAIN OPTIMAL VALUES OF
C                  THE NONLINEAR PARAMETERS.
C         BETA     L-VECTOR OF LINEAR PARAMETERS (OUTPUT ONLY).
C         IERR     INTEGER ERROR FLAG (OUTPUT):
C                  .GT. 0 - SUCCESSFUL CONVERGENCE, IERR IS THE NUMBER Of'
C                       ITERATIONS TAKEN.
C                  -1  TERMINATED FOR TOO MANY ITERATIONS.
C                  -2  TERMINATED FOR ILL-CONDITIONING (MARQUARDT
C                       PARAMETER TOO LARGE.)  ALSO SEE IERR = -8 BELOW.
C'                 -4  INPUT ERROR IN PARAMETER N, L, NL, LPP2, OR NMAX.
C                  -5  INC PIATRIX IMPROPERLY SPECIFIED, OR P DISAGREES
C                       WITH LPP2.
C                  -6  A WEIGHT WAS NEGATIVE.
C                  -7  'CONSTANT' COLUMN WAS COMPUTED MORE THAN ONCE.
C                  -8  CATASTROPHIC FAILURE - A COLUMN OF THE A MATRIX HA
C                       BECOME ZERO.  SEE 'CONVERGENCE FAILURES' BELOW.
C
C                  (IF IERR .LE. -4, THE LINEAR PARAMETERS, COVARIANCE
C                  MATRIX, ETC. ARE NOT RETURNED.)
C
C     SUBROUTINES REQUIRED
C
C          NINE SUBROUTINES, DPA, ORFAC1, ORFAC?, BACSUR, POSTPR, COV,
C        XNORM, INIT, AND VARERR ARE PROVIDED.  IN ADDITION, THE USER
C        MUST PROVIDE A SUBROUTINE (CORRESPONDING TO THE ARGUMENT ADA)
C        WHICH, GIVEN ALF, WILL EVALUATE THE FUNCTIONS PHI(J) AND THEIP
C        PARTIAL DERIVATIVES D PHI(J)/D ALF(K), AT THE SAMPLE POINTS
C        T(I).  THIS ROUTINE MUST BE DECLARED 'EXTERNAL' IN THE CALLIN(
C        PROGRAM.  ITS CALLING SEQUENCE IS
C
C        SUBROUTINE ADA (L+1, NL, N, NMAX, LPP2, IV, A, INC, T, ALF,
C        ISEL)
C
C          THE USER SHOULD MODIFY THE EXAMPLE SUBROUTINE 'ADA' (GIVEN
C        ELSEWHERE) FOR HIS OWN FUNCTIONS.
C
C          THE VECTOR SAMPLED FUNCTIONS PHI(J) SHOULD BE STORED IN THE
C        FIRST N ROWS AND FIRST L+1 COLUMNS OF THE MATRIX A, I.E.,
C        A(I, J) SHOULD CONTAIN PHI(J, ALF; T(I,1), T(I,2), ...,
C        T(I,IV)), I = 1, ..., N; J = 1, ..., L (OR L+1).  THE (L+1)-ST
C        COLUMN OF A CONTAINS PHI(L+1) IF PHI(L+1) IS IN THE MODEL,
C        OTHERWISE IT IS RESERVED FOR WORKSPACE.  THE 'CONSTANT' FUNC-
C        TIONS (THESE ARE FUNCTIONS PHI(J) WHICH DO NOT DEPEND UPON ANY
C        NONLINEAR PARAMETERS ALF, E.G., T(I)**J) (IF ANY) MUST AFPEAR
C        FIRST, STARTING IN COLUMN 1.  THE COLUMN N-VECTORS OF NONZERO
C        PARTIAL DERIVATIVES D PHI(J) / D ALF(K) SHOULD BE STORED
C        SEQUENTIALLY IN THE MATRIX A IN COLUMNS L+2 THROUGH L+P+1.
C        THE ORDER IS
C
C          D PHI(1)  D PHI(2)          D PHI(L)  D PHI(L+1)  D PHI(1)
C          --------, --------, ...,   --------, ----------, --------,
C          D ALF(1)  D ALF(1)          D ALF(1)   D ALF(1)   D ALF(2)
C
C          D PHI(2)        D PHI(L+1)        D PHI(1)         D PHI(L+1)
C          --------, ...,  ----------, ...,  ---------, ...,  ----------,
C          D ALF(2)         D ALF(2)         D ALF(NL)        D ALF(NL)
C
C        OMITTING COLUMNS OF DERIVATIVES WHICH ARE ZERO, AND OMITTING
C        PHI(L+1) COLUMNS IF PHI(L+1) IS NOT IN THE MODEL.  NOTE THAT
```

```
C        THE LINEAR PARAMETERS BETA ARE NOT USED IN THE MATRIX A.
C        COLUMN L+P+2 IS RESERVED FOR WORKSPACE.
C
C        THE CODING OF ADA SHOULD BE ARRANGED SO THAT:
C
C        ISEL = 1  (WHICH OCCURS THE FIRST TIME ADA IS CALLED) MEANS:
C                  A.  FILL IN THE INCIDENCE MATRIX INC
C                  B.  STORE ANY CONSTANT PHI'S IN A.
C                  C.  COMPUTE NONCONSTANT PHI'S AND PARTIAL DERIVA-
C                      TIVES.
C             = 2  MEANS COPIPUTE ONLY THE NONCONSTANT FUNCTIONS PHI
C             = 3  MEANS COMPUTE ONLY THE DERIVATIVES
C
C        (WHEN THE PROBLEM IS LINEAR (NL = 0) ONLY ISEL = 1 IS USED, AN
C        DERIVATIVES ARE NOT NEEDED.)
C
C      RESTRICTIONS
C
C           THE SUBROUTINE5 DPA, INIT (AND ADA) CONTAIN THE LOCALLY
C        DIMENSIONED MATRIX INC, WHOSE DIMENSIONS ARE CURRENTLY SET FOR
C        MAXIMA OF L+l = 8, NL = 12.  THEY MUST BE CHANGED FOR LARGER
C        PROBLEMS.  DATA PLACED IN ARRAY A IS OVERWRITTEN ('DESTROYED')
C        DATA PLACED IN ARRAYS T, Y AND INC IS LEFT INTACT.  THE PROGRA
C        RUNS IN WATFIV, EXCEPT WHEN L = 0 OR NL = 0.
C
C           IT IS ASSUMED THAT THE MATRIX PHI(J, ALF; T(I)) HAS FULL
C        COLUMN RANK.  THIS MEANS THAT THE FIRST L COLUMNS OF THE MATRI
C        A MUST BE LINEARLY INDEPENDENT.
C
C           OPTIONAL NOTE:  AS WILL BE NOTED FROM THE SAMPLE SUBPROGRA
C        ADA, TtiE DERIVATIVES D PHI(J)/D ALF(K) (ISEL = 3) MUST BE
C        COMPUTED INDEPENDENTLY OF THE FUNCTIONS PHI(J) (ISEL = 2),
C        SINCE THE FUNCTION VALUES ARE OVERWRITTEN AFTER ADA IS CALLED
C        WITH ISEL = 2.  THIS IS DONE TO MINIMIZE STORAGE, AT THE POS-
C        SIBLE ESPENSE OF SOME RECOMPUTATION (SINCE THE FUNCTIONS AND
C        DERIVATIVES FREQUENTLY HAVE SOME COMMON SUBEXPRESSIONS).  TO
C        REDUCE THE AMOUNT OF COMPUTATION AT THE EXPENSE OF SOME
C        STORAGE, CREATE A MATRIX B OF DIMENSION NMAX BY L+1 IN ADA, AN
C        AFTER THE COMPUTATION OF THE PIII'S (ISEL = 2), COPY THE VALUES
C        INTO B.  THESE VALUES CAN THEN BE USED TO CALCULATE THE DERIV-
C        ATIVES (ISEL = 3).  (THIS MAKES USE OF THE FACT THAT WHEN A
C        CALL TO ADA WITH ISEL = 3 FOLLOWS A CALL WITH ISEL = 2, THE
C        ALFS ARE THE SAME.)
C
C           TO CONVERT TO OTHER MACHINES, CHANGE THE OUTPUT UNIT IN THI
C        DATA STATEMENTS IN VARPRO, DPA, POSTPR, AND VARERR.  THE
C        PROGRAM HAS BEEN CHECKED FOR PORTABILITY BY THE BELL LABS PFOR
C        VERIFIER.  FOR MACHINES WITHOUT DOUBLE PRECISION HARDIJARE, IT
C        MAY BE DESIRABLE TO CONVERT TO SINGLE PRECISION.  THIS CAN BE
C        DONE BY CHANGING (A) THE DECLARATIONS 'DOUBLE PRECISION' TO
C        'REAL', (B) THE PATTERN '.D' TO '.E' IN THE 'DATA' STATEMENT I
C        VARPRO, (C) DSIGN, DSQRT AND DABS TO SIGN, SQRT AND ABS,
C        RESPECTIVELY, AND (D) DEXP TO EXP IN THE SAMPLE PROGRAPIS ONLY.
C
C      NOTE ON INTERPRETATION OF COVARIANCE MATRIX
C
C           FOR USE IN STATISTICAL ESTIMATION (MULTIPLE NONLINEAR
C        REGRESSION) VARPRO RETURNS THE COVARIANCE MATRIX OF TtiE LINEAR
C        AND NONLINEAR PARAMETERS.  THIS MATRIX WILL BE USEFUL ONLY IF
C        THE USUAL STATISTICAL ASSUMPTIONS HOLD:  AFTER WEIGHTING, THE
```

```
C        ERRORS IN THE OBSERVATIONS ARE INDEPENDENT AND NORMALLY DISTRI
C        BUTED, WITH MEAN ZERO AND THE SAME VARIANCE.  IF THE ERRORS DO
C        NOT HAVE MEAN ZERO (OR ARE UNKNOWN), THE PROGRAM WILL ISSUE A
C        WARNING MESSAGE (UNLESS IPRINT .LT. 0) AND THE COVARIANCE
C        MATRIX WILL NOT BE VALID.  IN THAT CASE, THE MODEL SHOULD BE
C        ALTERED TO INCLUDE A CONSTANT TERM (SET PHI(1) = 1.).
C
C           NOTE ALSO THAT, IN ORDER FOR THE USUAL ASSUMPTIONS TO HOLD,
C        THE OBSERVATIONS MUST ALL BE OF APPROXIMATELY THE SAME
C        MAGNITUDE (IN THE ABSENCE OF INFORMATION ABOUT THE ERROR OF
C        EACH OBSERVATION), OTHERWISE THE VARIRNCES WILL NOT BE THE
C        SAME.  IF THE OBSERVATIONS ARE NOT THE SAME SIZE, THIS CAN BE
C        CURED BY WEIGHTING.
C
C           IF THE USUAL ASSUMPTIONS HOLD, THE SQUARE ROOTS OF THE
C        DIAGONALS OF THE COVARIANCE MATRIX A GIVE THE STANDARD ERROR
C        S(I) OF EACH PARAMETER.  DIVIDING A(I,J) BY S(I)*S(J) YIELDS
C        THE CORRELATION MATRIX OF THE PARAMETERS.  PRINCIPAL ASES AND
C        CONFIDENCE ELLIPSOIDS CAN BE OBTAINED BY PERFORMING AN EIGEN-
C        VALUE/EIGENVECTOR ANALYSIS ON A.  ONE SHOULD CALL THE EISPACK
C        PROGRAM TRED2, FOLLOWED BY TQL2 (OR USE THE EISPAC CONTROL
C        PROGRAM).
C
C     CONVERGENCE FAILURES
C
C           IF CONVERGENCE FAILURES OCCUR, FIRST CHECK FOR INCORRECT
C        CODING OF THE SUBROUTINE ADA.  CHECK ESPECIALLY THE ACTION OF
C        ISEL, AND THE COMPUTATION OF THE FARTIAL DERIVATIVES.  IF THE:'
C        ARE CORRECT, TRY SEVERAL STARTING GUESSES FOR ALF.  IF ADA
C        IS CODED CORRECTLY, AND IF ERROR RETURNS IERR = -2 OR -8
C        PERSISTENTLY OCCUR, THIS IS A SIGN OF ILL-CONDITIONING, WHICH
C        MAY BE CAUSED BY SEVERAL THINGS.  ONE IS POOR SCALING OF THE
C        PARAMETERS; ANOTHER IS AN UNFORTUNATE INITIAL GUESS FOR THE
C        PARAMETERS, STILL ANOTHER IS A POOR CHOICE OF THE MODEL.
C
C     ALGORITHM
C
C           THE RESIDUAL R IS MODIFIED TO INCORPORATE, FOR ANY FIXED
C        ALF, THE OPTIMAL LINEAR PARAMETERS FOR THAT ALF.  IT IS THEN
C        POSSIBLE TO MINIMIZE ONLY ON THE NONLINEAR PARAMETERS.  AFTER
C        THE OPTIMAL VALUES OF THE NONLINEAR PARAMETERS HAVE BEEN DETER
C        MINED, THE LINEAR PARAMETERS CAN BE RECOVERED BY LINEAR LEAST
C        SQUARES TECHNIQUES (SEE REF. 1).
C
C           THE MINIMIZATION IS BY A MODIFICATION OF OSBORNE'S (REF. 3)
C        MODIFICATION OF THE LEVENBERG-MARQUARDT ALGORITHM.  INSTEAD OF
C        SOLVING THE NORMAL EQUATIONS WITH MATRIX
C
C                  T      2
C              (J J + NU  * D),      WHERE  J = D(ETA)/D(ALF),
C
C        STABLE ORTHOGONAL (HOUSEHOLDER) REFLECTIONS ARE USED ON A
C        MODIFICATION OF THE MATRIX
C                                    (   J   )
C                                    (-------) ,
C                                    ( NU*D  )
C
C        WHERE D IS A DIAGONAL MATRIX CONSISTING OF THE LENGTHS OF THE
C        COLUMNS OF J.  THIS MARQUARDT STABILIZATION ALLOWS THE ROUTINE
C        TO RECOVER FROM SOME RANK DEFICIENCIES IN THE JACOBIAN.
```

```
C        OSBORNE'S EMPIRICAL STRATEGY FOR CHOOSING THE MARQUARDT PARAM-
C        ETER HAS PROVEN REASONABLY SUCCESSFUL IN PRACTICE.  (GAUSS-
C        NEWTON WITH STEP CONTROL CAN BE OBTAINED BY MAKING THE CHANGE
C        INDICATED BEFORE THE INSTRUCTION LABELED 5).  A DESCRIPTION CA
C        BE FOUND IN REF. (3), AND A FLOW CHART IN (2), P. 22.
C
C        FOR REFERENCE, SEE
C
C        1.  GENE H. GOLUB AND V. PEREYRA, 'THE DIFFERENTIATION OF
C            PSEUDO-INVERSES AND NONLINEAR LEAST SQUARES PROBLEMS WHOSE
C            VARIABLES SEPARATE,' SIAM J. NUMER. ANAL. 10, 413-432
C            (1973).
C        2.  ------, SAME TITLE, STANFORD C.S. REPORT 72-261, FEB. 1972
C        3.  OSBORNE, MICHAEL R., 'SOME ASPECTS OF NON-LINEAR LEAST
C            SQUARES CALCULATIONS,' IN LOOTSMA, ED., 'NUMERICAL METHODS
C            FOR NON-LINEAR OPTIMIZATION,' ACADEMIC PRESS, LONDON, 1972
C        4.  KROGH, FRED, 'EFFICIENT IMPLEMENTATION OF A VARIABLE PRO-
C            JECTION ALGORITHM FOR NONLINEAR LEAST SQUARES PROBLEMS,'
C            COMM. ACM 17, PP. 167-169 (MARCH, 1974).
C        5.  KAUFMAN, LINDA, 'A VARIABLE PROJECTION METHOD FOR SOLVING
C            SEPARABLE NONLINEAR LEAST SQUARES PROBLEMS', B.I.T. 15,
C            49-57 (1975).
C        6.  DRAFER, N., AND SMITH, H., APPLIED REGRESSION ANALYSIS,
C            WILEY, N.Y., 1966 (FOR STATISTICAL INFORMATION ONLY).
C        7.  C. LAWSON AND R. HANSON, SOLVING LEAST SQUARES PROBLEMS,
C            PRENTICE-HALL, ENGLEWOOD CLIFFS, N. J., 1974.
C
C                    JOHN BOLSTAD
C                    COMPUTER SCIENCE DEPT., SERRA HOUSE
C                    STANFORD UNIVERSITY
C                    JANUARY, 1977
C
C     ..................................................................
C
      DOUBLE PRECISION A(NMAX, LPP2), BETA(L), ALF(NL), T(NMAX, IV),
     2 W(N), Y(N), ACUM, EPSl, GNSTEP, NU, PRJRES, R, RNEW, XNORM
      INTEGER B1, OUTPUT
      LOGICAL SKIP
      EXTERNAL ADA
      DATA EPSl /1.D-6/, ITMAX /50/, OUTPUT /6/
C
C           THE FOLLOWING TWO PARAMETERS ARE USED IN THE CONVERGEHCE
C           TEST:  EPSl IS AN ABSOLUTE AND RELATIVE TOLERANCE FOR THE
C           NORM OF THE PROJECTION OF THE RESIDUAL ONTO THE RANGE OF TH
C           JACOBIAN OF THE VARIABLE PROJECTION FUNCTIONAL.
C           ITMAX IS THE MAXIMUM NUMBER OF FUNCTION AND DERIVATIVE
C           EVALUATIONS ALLOWED.  CAUTION:  EPSl MUST NOT BE
C           SET SMALLER THAN 10 TIMES THE UNIT ROUND-OFF OF THE MACHINE
C
      IERR = 1
      ITER = 0
      LP1 = L + 1
      B1 = L + 2
      LNL2 = L + NL + 2
      NLPl = NL + 1
      SKIP = .FALSE.
      MODIT = IPRINT
      IF (IPRINT .LE. 0) MODIT = ITMAX + 2
      NU = 0.
C               IF GAUSS-NEWTON IS DESIRED REMOVE THE NEXT STATEMENT.
```

```
      NU = 1.
C
C              BEGIN OUTER ITERATION LOOP TO UPDATE ALF.
C              CALCULATE THE NORM OF THE RESIDUAL AND THE DERIVATIVE OF
C              THE MODIFIED RESIDUAL THE FIRST TIME, BUT ONLY THE
C              DERIVATIVE IN SUBSEQUENT ITERATIONS.
C
    5 CALL DPA (L, NL, N, NMAX, LPP2, IV, T, Y, W, ALF, ADA? IERR,
     X IPRINT, A, BETA, A(1, LP1), R)
      GNSTEP = 1.0
      ITERIN = 0
      IF (ITER .GT. 0) GO TO 10
         IF (NL .EQ. 0) GO TO 90
         IF (IERR .NE. 1) GO TO 99
C
         IF (IPRINT .LE. 0) GO TO 10
         WRITE (OUTPUT, 207) ITERIN, R
         WRITE (OUTPUT, 200) NU
         WRITE (3,207) ITERIN,R
         WRITE (3,200) NU
C                                  BEGIN TWO-STAGE ORTHOGONAL FACTORIZATION
   10 CALL ORFAC1(NLP1, NMAX, N, L, IPRINT, A(1, B1), PRJRES, IERR)
      IF (IERR .LT. 0) GO TO 99
      IERR = 2
      IF (NU .EQ. 0.) GO TO 30
C
C              BEGIN INNER ITERATION LOOP FOR GENERATING NEW ALF AND
C              TESTING IT FOR ACCEPTANCE.
C
   25    CALL ORFAC2(NLP1, NMAX, NU, A(1, B1))
C
C              SOLVE A NL X NL UPPER TRIANGULAR SYSTEM FOR DELTA-ALF.
C              THE TRANSFORMED RESIDUAL (IN COL. LNL2 OF A) IS OVER-
C              WRITTEN BY THE RESULT DELTA-ALF.
C
   30    CALL BACSUB (NMAX, NL, A(1, B1), A(1, LNL2))
         DO 35 K = 1, NL
   35       A(K, B1)  = ALF(K) + A(K, LNL2)
C           NEW ALF(K) = ALF(K) + DELTA ALF(K)
C
C              STEP TO THE NEW POINT NEW ALF, AND COMPUTE THE NEW
C              NORM OF RESIDUAL.  NEW ALF IS STORED IN COLUMN B1 OF A.
C
   40    CALL DPA (L, NL, N, NMAX, LPP2, IV, T, Y, W, A(1, B1), ADA,
     X   IERR, IPRINT, A, BETA, A(1, LP1), RNEW)
         IF (IERR .NE. 2) GO TO 99
         ITER = ITER + 1
         ITERIN = ITERIN + 1
         SKIP = MOD(ITER, MODIT) .NE. 0
         IF (SKIP) GO TO 45
            WRITE (OUTPUT, 203) ITER
            WRITE (OUTPUT, 216) (A(K, B1), K = 1, NL)
            WRITE (OUTPUT, 207) ITERIN, RNEW
             WRITE (3,203) ITER
             WRITE (3,216) (A(K,B1), K = 1, NL)
             WRITE (3,207) ITERIN,RNEW
C
   45    IF (ITER .LT. ITMAX) GO TO 50
            IERR = -1
            CALL VARERR (IPRINT, IERR, 1)
```

```
            GO TO 95
   50     IF (RNEW - R .LT. EPS1*(R + 1.D0)) GO TO 75
C
C              RETRACT THE STEP JUST TAKEN
C
               IF (NU .NE. 0.) GO TO 60
C                                                  GAUSS-NEWTON OPTION ONLY
               GNSTEP = 0.5*GNSTEP
               IF (GNSTEP .LT. EPS1) GO TO 95
               DO 55 K = 1, NL
   55             A(K, B1) = ALF(K) + GNSTEP*A(K, LNL2)
               GO TO 40
C                                                  ENLARGE THE MARQUARDT PARAMETF
   60          NU = 1.5*NU
               IF (.NOT. SKIP) WRITE (OUTPUT, 206) NU
               IF (NU .LE. 100.) GO TO 65
                  IERR = -2
                  CALL VARERR (IPRINT, IERR, 1)
                  GO TO 95
C                                                  RETRIEVE UPPER TRIANGULAR FORM
C                                                  AND RESIDUAL OF FIRST STAGE.
   65     DO 70 K = 1, NL
             KSUB = LP1 + K
             DO 70 J = K, NLP1
                JSUB = LP1 + J
                ISUB = NLP1 + J
   70           A(K, JSUB) = A(ISUB, KSUB)
          GO TO 25
C                                                  END OF INNER ITERATION LOOP
C              ACCEPT THE STEP JUST TAKEN
C
   75  R = RNEW
          DO 80 K = 1, NL
   80        ALF(K) = A(K, B1)
C                                                  CALC. NORM(DELTA ALF)/NORM(ALF
          ACUM = GNSTEP*XNORM(NL, A(1, LNL2))/XNORM(NL, ALF)
C
C              IF ITERIN IS GREATER THAN 1, A STEP WAS RETRACTED DURING
C              THIS OUTER ITERATION.
C
          IF (ITERIN .EQ. 1) NU = 0.5*NU
          IF (SKIP) GO TO 85
             WRITE (OUTPUT, 200) NU
             WRITE (OUTPUT, 208) ACUM
             WRITE (3,200) NU
             WRITE (3,208) ACUM
   85  IERR = 3
          IF (PRJRES .GT. EPS1*(R + 1.D0)) GO TO 5
C              END OF OUTER ITERATION LOOP
C
C              CALCULATE FINAL QUANTITIES -- LINEAR PARAMETERS, RESIDUALS,
C              COVARIANCE MATRIX, ETC.
C
   90  IERR = ITER
   95  IF (NL .GT. 0) CALL DPA(L, NL, N, NMAX, LPP2, IV, T, Y, W, ALF,
         X ADA, 4, IPRINT, A, BETA, A(1, LP1), R)
          CALL POSTPR(L, NL, N, NMAX, LNL2, EPS1, R, IPRINT, ALF, W, A,
         X A(1, LP1), BETA, IERR)
   99  RETURN
C
```

```
  200 FORMAT (9H      NU =, E15.7)
  203 FORMAT (12H0  ITERATION. I4, 24H    NONLINEAR PARAMETERS)
  206 FORMAT (25H      STEP RETRACTED, NU =, E15.7)
  207 FORMAT (1H0, 15, 20H  NORM OF RESIDUAL =, E15.7)
  208 FORMAT (34H      NORM(DELTA-ALF) / NORM(ALF) =, E12.3)
  216 FORMAT (1H0, 7E15.7)
      END
C
      SUBROUTINE ORFAC1(NLP1, NMAX, N, L, IPRINT, B, PRJRES, IERR)
C
C             STAGE 1:  HOUSEHOLDER REDUCTION OF
C
C                     (          )      ( DR'. R3 )     NL
C                     ( DR . R2 )  TO   (----, -- ),
C                     (          )      (  0  . R4 )   N-L-NL
C
C                       NL    1           NL    1
C
C          WHERE DR = -D(Q2)*Y IS THE DERIVATIVE OF THE MODIFIED RESIDUA
C          PRODUCED BY DPA, R2 IS THE TRANSFORMED RESIDUAL FROM DPA, ANT
C          DR' IS IN UPPER TRIANGULAR FORM (AS IN REF. (2), P. 18).
C          DR IS STORED IN ROWS L+l TO N AND COLUMNS L+2 TO L + NL + 1 O
C          THE MATRIS A (I.E., COLUPINS 1 TO NL OF THE PIATRIX B).  R2 IS
C          STORED IN COLUMN L + NL + 2 OF THE MATRIX A (COLUMN NL + 1 Of
C          B).  FOR K = 1, 2, ..., NL, FIND REFLECTION I - U * U' / BETA
C          WHICH ZEROES B(I, K), I = L+K+1, ..., N.
C
C     ..................................................................
C
      DOUBLE PRECISION ACUM, ALPHA, B(NMAX, NLP1), BETA, DSIGN, PRJRES,
     X U, SNORM
C
      NL = NLP1 - 1
      NL23 = 2*NL + 3
      LP1 = L + 1
C
      DO 30 K = 1, NL
         LPK = L + K
         ALPHA = DSIGN(XNORM(N+1-LPK, B(LPK, K)), B(LPK, K))
         U = B(LPK, K) + ALPHA
         B(LPK, K) = U
         BETA = ALPHA * U
         IF (ALPHA .NE. 0.0) GO TO 13
C                                                    COLUMN WAS ZERO
         IERR = -8
         CALL VARERR (IPRINT, IERR, LP1 + K)
         GO TO 99
C                                 APPLY REFLECTIONS TO REMAINING COLUMNS
C                                 OF B AND TO RESIDUAL VECTOR.
   13    KP1 = K + 1
         DO 25 J = KP1, NLPl
            ACUM = 0.0
            DO 20 I = LPK, N
   20          ACUM = ACUM + B(I, K) * B(I, J)
            ACUM = ACUM / BETA
            DO 25 I = LPK, N
   25          B(I, J) = B(I, J) - B(I, K) * ACUM
   30    B(LPK, K) = -ALPHA
C
      PRJRES = XNORM(NL, B(LP1, NLP1))
```

```
C
C            SAVE UPPER TRIANGULAR FORM AND TRANSFORMED RESIDUAL, FOR US
C            IN CASE A STEP IS RETRACTED.  ALSO COMPUTE COLUMN LENGTHS.
C
      IF (IERR .EQ. 4) GO TO 99
      DO 50 K = 1, NL
         LPK = L + K
         DO 40 J = K, NLP1
            JSUB = NLP1 + J
            B(K, J) = B(LPK, J)
 40         B(JSUB, K) = B(LPK, J)
 50      B(NL23, K) = XNORM(K, B(LP1, K))
C
 a9   RETURN
      END
C
      SUBROUTINE ORFAC2(NLP1, NMAX, NU, B)
C
C        STAGE 2:  SPECIAL HOUSEHOLDER REDUCTION OF
C
C                        NL      ( DR' . R3 )       (DR'' . R5 )
C                                (----- -- )       (----- -- )
C                    N-L-NL      (  0  . R4 )  TO   (  0  . R4 )
C                                (----- -- )       (----- -- )
C                        NL      (NU*D , 0  )       (  0  . R6 )
C
C                                   NL    1            NL    1
C
C        WHERE DR', R3, AND R4 ARE AS IN ORFAC1, NU IS THE MARQUARDT
C        PARAMETER, D IS A DIAGONAL MATRIX CONSISTING OF THE LENGTHS OF
C        THE COLUMNS OF DR', AND DR'' IS IN UPPER TRIANGULAR FORM.
C        DETAILS IN (1), PP. 423-424.  NOTE THAT THE (N-L-NL) BAND OF
C        ZEROES, AND R4, ARE OMITTED IN STORAGE.
C
C     ..................................................................
C
      DOUBLE PRECISION ACUM, ALPHA, B(NMAX, NLP1), BETA, DSIGN, NU, U,
     X XNORM
C
      NL = NLP1 - 1
      NL2 = 2*NL
      NL23 = NL2 + 3
      DO 30 K = 1, NL
         KP1 = K + 1
         NLPK = NL + K
         NLPKM1 = NLPK - 1
         B(NLPK, K) = NU * B(NL23, K)
         B(NL, K) = B(K, K)
         ALPHA = DSIGN(XNORM(K+1, B(NL, K)), B(K, K))
         U = B(K, K) + ALPHA
         BETA = ALPHA * U
         B(K, K) = - ALPHA
C                        THE K-TH REFLECTION MODIFIES ONLY ROWS K,
C                        NL+1, NL+2, ..., NL+K, AND COLUMNS K TO NL+1.
         DO 30 J = KP1, NLP1
            B(NLPK, J) = 0.
            ACUM = U * B(K,J)
            DO 20 I = NLP1, NLPKM1
 20            ACUM = ACUM + B(I,K) * B(I,J)
            ACUM = ACUM / BETA
```

```
            B(K,J) = B(K,J) - U * ACUM
            DO 30 I = NLP1, NLPK
   30          B(I,J) = B(I,J) - B(I,K) * ACUM
C
      RETURN
      END
C
      SUBROUTINE DPA (L, NL, N, NMAX, LPP2, IV, T, Y, W, ALF, ADA, ISEL
     X IPRINT, A, U, R, RNORM)
C
C        COMPUTE THE NORM OF THE RESIDUAL (IF ISEL = 1 OR 2), OR THE
C        (N-L) X NL DERIVATIVE OF THE MODIFIED RESIDUAL (N-L) VECTOR
C        Q2*Y (IF ISEL = 1 OR 3).  HERE Q * PHI = S, I.E.,
C
C         L     ( Q1 ) (                    )   ( S  . R1 .  F1  )
C               (----) ( PHI . Y . D(PHI)  ) = (--- . --   ----  )
C         N-L   ( Q2 ) (                    )   ( 0  . R2 .  F2  )
C
C                   N       L    1      P          L     1     P
C
C        WHERE Q IS N X N ORTHOGONAL, AND S IS L X L UPPER TRIANGULAR.
C        THE NORM OF THE RESIDUAL = NORM(R2), AND THE DESIRED DERIVATIV
C        ACCORDING TO REF. (5), IS
C                                                        -1
C                    D(Q2 * Y) = -Q2 * D(PHI)* S  * Q1* Y.
C
C     ..................................................................
C
      DOUBLE PRECISION A(NMAX, LPP2), ALF(NL), T(NMAX, IV), W(N), Y(N),
     X ACUM, ALPHA, BETA, RNORM, DSIGN, DSQRT, SAVE, R(N), U(L), XNORM
      INTEGER FIRSTC, FIRSTR, INC(12, 8)
      LOGICAL NOWATE, PHILP1
      EXTERNAL ADA
C
      IF (ISEL .NE. 1) GO TO 3
         LP1 = L + 1
         LNL2 = L + 2 + NL
         LP2 = L + 2
         LPP1 = LPP2 - 1
         FIRSTC = 1
         LASTC = LPP1
         FIRSTR = LP1
         CALL INIT(L, NL, N, NMAX, LPP2, IV, T, W, ALF, ADA, ISEL,
     X   IPRINT, A, INC, NCON, NCONP1, PHILP1, NOWATE)
         IF (ISEL .NE. 1) GO TO 99
         GO TO 30
C
    3 CALL ADA (LP1,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,MIN0(ISEL,3))
      IF (ISEL .EQ. 2) GO TO 6
C                                                    ISEL = 3 OR 4
      FIRSTC = LP2
      LASTC = LPP1
      FIRSTR = (4 - ISEL)*L + 1
      GO TO 50
C                                                    ISEL = 2
    6 FIRSTC = NCONP1
      LASTC = LP1
      IF (NCON .EQ. 0) GO TO 30
      IF (A(1, NCON) .EQ. SAVE) GO TO 30
         ISEL = -7
```

```
      CALL VARERR (IPRINT, ISEL, NCON)
      GO TO 99
C                                                        ISEL = 1 OR 2
   30 IF (PHILP1) GO TO 40
      DO 35 I = 1, N
   35    R(I) = Y(I)
      GO TO 50
   40 DO 45 I = 1, N
   45    R(I) = Y(I) - R(I)
C                                                  WEIGHT APPROPRIATE COLUMN
   50 IF (NOWATE) GO TO 58
      DO 55 I = 1, N
      ACUM = W(I)
      DO 55 J = FIRSTC, LASTC
   55    A(I, J) = A(I, J) * ACUM
C
C        COMPUTE ORTHOGONAL FACTORIZATIONS BY HOUSEHOLDER
C        REFLECTIONS.  IF ISEL = 1 OR 2, REDUCE PHI (STORED IN THE
C        FIRST L COLUMNS OF THE MATRIX A) TO UPPER TRIANGULAR FORM,
C        (Q*PHI = S), AND TRANSFORM Y (STORED IN COLUMN L+1), GETTIN
C        Q*Y = R.  IF ISEL = 1, ALSO TRANSFORM J = D PHI (STORED IN
C        COLUMNS L+2 THROUGH L+P+1 OF THE MATRIX A), GETTING Q*J = F
C        IF ISEL = 3 OR 4, PHI HAS ALREADY BEEN REDUCED, TRANSFORM
C        ONLY J.  S, R, AND F OVERWRITE PHI, Y, AND J, RESPECTIVELY.
C        AND A FACTORED FORM OF Q IS SAVED IN U AND THE LOWER
C        TRIANGLE OF PHI.
C
   58 IF (L .EQ. 0) GO TO 75
      DO 70 K = 1, L
      KP1 = K + 1
      IF (ISEL .GE. 3 .OR. (ISEL .EQ. 2 .AND. K .LT.NCONP1)) GO TO
      ALPHA = DSIGN(XNORM(N+1-K, A(K, K)), A(K, K))
      U(K) = A(K, K) + ALPHA
      A(K, K) = -ALPHA
      FIRSTC = KP1
      IF (ALPHA .NE. 0.0) GO TO C6
      ISEL = -8
      CALL VARERR (IPRINT, ISEL, K)
      GO TO 99
C                                          APPLY REFLECTIONS TO COLUMNS
C                                          FIRSTC TO LASTC.
   66 BETA = -A(K, K) * U(K)
      DO 70 J = FIRSTC, LASTC
      ACUM = U(K)*A(K, J)
      DO 68 I = KP1, N
   68    ACUM = ACUM + A(I, K)*A(I, J)
      ACUM = ACUM / BETA
      A(K,J) = A(K,J) - U(K)*ACUM
      DO 70 I = KP1, N
   70    A(I, J) = A(I, J) - A(I, K)*ACUM
C
   75 IF (ISEL .GE. 3) GO TO 85
      RNORM = XNORM(N-L, R(LP1))
      IF (ISEL .EQ. 2) GO TO 99
      IF (NCON .GT. 0) SAVE = A(1, NCON)
C
C        F2 IS NOW CONTAINED IN ROWS L+1 TO N AND COLUMNS L+2 TO
C        L+P+1 OF THE MATRIX A.  NOW SOLVE THE L X L UPPER TRIANGUL
C        SYSTEM S*BETA = R1 FOR THE LINEAR PARAMETERS BETA.  BETA
C        OVERWRITES R1.
```

```
C
   85 IF (L .GT. 0) CALL BACSUB (NMAX, L, A, R)
C
C           MAJOR PART OF KAUFMAN'S SIMPLIFICATION OCCURS HERE.  COMPUT
C           THE DERIVATIVE OF ETA WITH RESPECT TO THE NONLINEAR
C           PARAMETERS
C
C    T    D ETA          T     L            D PHI(J)     D PHI(L+1)
C  Q * --------   =   Q * (SUM BETA(J) --------   + ----------)   =   F2*BET
C      D ALF(K)           J=1           D ALF(K)      D ALF(K)
C
C           AND STORE THE RESULT IN COLUMNS L+2 TO L+NL+1.  IF ISEL NOT
C           = 4, THE FIRST L ROWS ARE OMITTED.  THIS IS -D(Q2)*Y.  IF
C           ISEL NOT = 4 THE RESIDUAL R? = Q2*Y (IN COL. L+1) IS COPIED
C           TO COLUMN L+NL+2.  OTHERWISE ALL OF COLUPIN L+l IS COPIED.
C
      DO 95 I = FIRSTR, N
         IF (L .EQ. NCON) GO TO 95
         M = LP1
         DO 90 K = 1, NL
            ACUM = 0.
            DO 88 J = NCONPl, L
               IF (INC(K, J) .EQ. 0) GO TO 88
               M = M + 1
               ACUM = ACUM + A(I, M) * R(J)
   88          CONTINUE
            KSUB = LP1 + K
            IF (INC(K, LP1) .EQ. 0) GO TO 90
            M = M + 1
            ACUM = RCUM + A(I, M)
   90       A(I, KSU8) = ACUM
   95    A(I, LNL2) = R(I)
C
   99 RETURN
      END
C
      SUBROUTINE INIT(L, NL, N, NMAX, LPP2, IV, T, W, ALF, ADA, ISEL,
     X IPRINT, A, INC, NCON, NCONP1, PHILP1, NOWATE)
C
C        CHECK VALIDITY OF INPUT PARAMETERS, AND DETERMINE NUMBER OF
C        CONSTANT FUNCTIONS.
C
C     .....................................................................
C
      DOUBLE PRECISION A(NMAX, LPP2), ALF(NL), T(NMAX, IV), W(N),
     X DSQRT
      INTEGER OUTPUT, P, INC(12, 8)
      LOGICAL NOWATE, PHILP1
      DATA OUTPUT /6/
C
      LP1 = L + 1
      LNL2 = L + 2 + NL
C                                                 CHECK FOR VALID INPUT
      IF (L .GE. 0 .AND. NL .GE. 0 .AND. L+NL .LT. N .AND. LNL? .LE.
     X LPP2 .AND. 2*NL + 3 .LE. NMAX .AND. N .LE. NMAX .AND.
     X IV .GT. 0 .AND. .NOT. (NL .EQ. 0 .AND. L .EQ. 0)) GO TO 1
         ISEL = -4
         CALL VARERR (IPRINT, ISEL, 1)
         GO TO 99
C
```

```
    1 IF (L .EQ. 0 .OR. NL .EQ. 0) GO TO 3
         DO 2 J = 1, LP1
            DO 2 K = 1, NL
    2          INC(K, J) = 0
C
    3 CALL ADA (LP1, NL, N, NMAX, LPP2, IV, A, INC, T, ALF, ISEL)
C
      NOWATE = .TRUE.
      DO 9 I = 1, N
         NOWATE = NOWATE .AND. (W(I) .EQ. 1.0)
         IF (W(I) .GE. 0.) GO TO 9
C                                                  ERROR IN WEIGHTS
         ISEL = -6
         CALL VARERR (IPRINT, ISEL, I)
         GO TO 95,
    9    W(I) = DSQRT(W(I))
C
      NCON = L
      NCONPl = LP1
      PHILPl = L .EQ. 0
      IF (PHILP1 .OR. NL .EQ. 0) GO TO 99
C                                  CHECK INC MATRIX FOR VALID INPUT AN
C                                  DETERMINE NUMBER OF CONSTANT FCNS.
      P = 0
      DO 11 J = 1, LP1
         IF (P .EQ. 0) NCONPl = J
         DO 11 K = 1, NL
            INCEJ = INC(K, J)
            IF (INCKJ .NE. 0 .AND. INCKJ .NE. 1) GO TO 15
            IF (INCKJ .EQ. 1) P = P + 1
   11       CONTINUE
C
      NCON = NCONPl - 1
      IF (IPRINT .GE. 0) WRITE (OUTPUT, 210) NCON
      IF (L+P+2 .EQ. LPP2) GO TO 20
C                                                  INFUT ERROR IN INC MATRI
   15 ISEL = -5
      CALL VARERR (IPRINT, ISEL, 1)
      GO TO 99
C                                  DETERMINE IF PHI(L+1) IS IN THE MODEL
   20 DO 25 K = 1, NL
   25    IF (INC(K, LP1) .EQ. 1) PHILPl = .TRUE.
C
   99 RETURN
  210 FORMAT (33H0  NUMBER OF CONSTANT FUNCTIONS =, I4 /)
      END
      SUBROUTINE BACSUB (NMAX, N, A, X)
C
C        BACKSOLVE THE N X N UPPER TRIANGULAR SYSTEM A*X = B.
C        THE SOLUTION X OVERWRITES THE RIGHT SIDE B.
C
      DOUBLE PRECISION A(NMAX, N), X(N), ACUM
C
      X(N) = X(N) / A(N, N)
      IF (N .EQ. 1) GO TO 30
      NP1 = N + 1
      DO 20 IBACK = 2, N
         I = NP1 - IBACK
C           I = N-1, N-2, ..., 2, 1
         IP1 = I + 1
```

```
      ACUM = X(I)
      DO 10 J = IP1, N
10       ACUM = ACUM - A(I,J)*X(J)
20    X(I) = ACUM / A(I,I)
C
30 RETURN
   END
   SUBROUTINE POSTPR(L, NL, N, NMAX, LNL2, EPS, RNORM, IPRINT, ALF,
  X W, A, R, U, IERR)
C
C       CALCULATE RESIDUALS, SAMPLE VARIANCE, AND COVARIANCE MATRIX.
C       ON INPUT, U CONTAINS INFORMATION ABOUT HOUSEHOLDER REFLECTIONS
C       FROM DPA.  ON OUTPUT, IT CONTAINS THE LINEAR PARAMETERS.
C
   DOUBLE PRECISION A(NMAX, LNL2), ALF(NL), R(N), U(L), W(N), ACUM,
  X EPS, PRJRES, RNORM, SAVE, DABS
   INTEGER OUTPUT
   DATA OUTPUT /6/
C
   LP1 = L + 1
   LFNL = LNL2 - 2
   LNL1 = LPNL + 1
   DO 10 I = 1, N
10    W(I) = W(I)**2
C
C            UNWIND HOUSEHOLDER TRANSFORMATIONS TO GET RESIDUALS,
C            AND MOVE THE LINEAR PARAMETERS FROM R TO U.
C
   IF (L .EQ. 0) GO TO 30
   DO 25 KBACK = 1, L
      K = LP1 - KBACK
      KP1 = K + 1
      ACUM = 0.
      DO 20 I = KP1, N
20       ACUM = ACUM + A(I, K) * R(I)
      SAVE = R(K)
      R(K) =  ACUM / A(K, K)
      ACUM = -ACUM / (U(K) * A(K, K))
      U(K) = SAVE
      DO 25 I = KP1, N
25       R(I) = R(I) - A(I, K)*ACUM
C                                                  COMPUTE MEAN ERROR
30 ACUM = 0.
   DO 35 I = 1, N
35    ACUM = ACUM + R(I)
   SAVE = ACUM / N
C
C            THE FIRST L COLUMNS OF THE MATRIX HAVE BEEN REDUCED TO
C            UPPER TRIANGULAR FORM IN DPA.  FINISH BY REDUCING ROWS
C            L+1 TO N AND COLUMNS L+2 THROUGH L+NL+1 TO TRIANGULAR
C            FORM.  THEN SHIFT COLUMNS OF DERIVATIVE MATRIX OVER ONE
C            TO THE LEFT TO BE ADJACENT TO THE FIRST L COLUMNS.
C
   IF (NL .EQ. 0) GO TO 45
   CALL ORFAC1(NL+1, NMAX, N, L, IPRINT, A(1, L+2), PRJRES, 4)
   DO 40 I = 1, N
      A(I, LNL2) = R(I)
      DO 40 K = LP1, LNL1
40       A(I, K) = A(I, K+1)
C                                                  COMPUTE COVARIANCE MATRI
```

```
   45 A(1, LNL2) = RNORM
      ACUM = RNORM*RNORM/(N - L - NL)
      A(2, LNL2) = ACUM
      CALL COV(NMAX, LPNL, ACUM, A)
C
      IF (IPRINT .LT. 0) GO TO 99
      IJRITE (OUTPUT, 209)
      IF (L .GT. 0) WRITE (OUTPUT, 210) (U(J), J = 1, L)
      IF (NL .GT. 0) WRITE (OUTPUT, 211) (ALF(K), K = 1, NL)
      WRITE (OUTPUT, 214) RNORM, SAVE, ACUM
      IF (DABS(SAVE) .GT. EPS) WRITE (OUTPUT, 215)
      IF (DABS(SAVE) .GT.EPS) WRITE (3,215)
      WRITE (OUTPUT, 209)
       WRITE (3,209)
   99 RETURN
C
  209 FORMAT (1H0, 50(1H'))
  210 FORMAT (20H0  LINEAR PARAMETERS // (7E15.7))
  211 FORMAT (23H0  NONLINEAR PARAMETERS // (7E15.7))
  211, FORMAT (21H0  NORM OF RESIDUAL =, E15.7, 33H EXPECTED ERROR OF OE
     XERVATIONS =, E15.7, / 39H   ESTIMATED VARIANCE OF OBSERVATIONS =,
     X E15.7 )
  215 FORMAT (95H  WARNING -- EXPECTED ERROR OF OBSERVATIONS IS NOT ZEI
     X.  COVARIANCE MATRIX MAY BE MEANINGLESS. /)
      END
      SUBROUTINE COV(NMAX, N, SIGMA2, A)
C
C          COMPUTE THE SCALED COVARIANCE MATRIX OF THE L + NL
C       PARAMETERS.  THIS INVOLVES COMPUTING
C
C                                  2     -1    -T
C                            SIGMA  *  T   * T
C
C       WHERE THE (L+NL) X (L+NL) UPPER TRIANGULAR MATRIX T IS
C       DESCRIBED IN SUBROUTINE POSTPR.  THE RESULT OVERWRITES THE
C       FIRST L+NL ROWS AND COLUMNS OF THE MATRIX A.  THE RESULTING
C       MATRIX IS SYMMETRIC.  SEE REF. 7, PP. 67-70, 281.
C
C     ................................................................
C
      DOUBLE PRECISION A(NMAX, N), SUM, SIGMA2
C
      DO 10 J = 1, N
   10    A(J, J) = 1./A(J, J)
C
C                  INVERT T UPON ITSELF
C
      IF (N .EQ. 1) GO TO 70
      NM1 = N - 1
      DO 60 I = 1, NM1
         IP1 = I + 1
         DO 60 J = IP1, N
            JM1 = J - 1
            SUM = 0.
            DO 50 M = I, JM1
   50          SUM = SUM + A(I, M) * A(M, J)
   60       A(I, J) = - SUM * A(J, J)
C
C                NOW FORM THE MATRIX PRODUCT
C
```

```
   70 DO 90 I = 1, N
         DO 90 J = I, N
            SUM = 0.
            DO 80 M = J, N
   80          SUM = SUM + A(I, M) * A(J, M)
            SUM = SUM * SIGMA2
            A(I, J) = SUM
   90       A(J, I) = SUM
C
      RETURN
      END
      SUBROUTINE VARERR (IPRINT, IERR, K)
C
C        PRINT ERROR MESSAGES
C
      INTEGER ERRNO, OUTPUT
      DATA OUTPUT /6/
C
      IF (IPRINT .LT. 0) GO TO 99
      ERRNO = IABS(IERR)
      GO TO (1, 2, 99, 4, 5, 6, 7, 8), ERRNO
C
    1 WRITE (OUTPUT, 101)
      GO TO 99
    2 WRITE (OUTPUT, 102)
      GO TO 99
    4 WRITE (OUTPUT, 104)
      GO TO 99
    5 WRITE (OUTPUT, 105)
      GO TO 99
    6 WRITE (OUTPUT, 106) K
      GO TO 99
    7 WRITE (OUTPUT, 107) K
      GO TO 99
    8 WRITE (OUTPUT, 108) K
C
   99 RETURN
  101 FORMAT (46H0  PROBLEM TERMINATED FOR EXCESSIVE ITERATIONS //)
  102 FORMAT (49H0  PROBLEM TERMINATED BECAUSE OF ILL-CONDITIONING //)
  104 FORMAT (/ 50H INPUT ERROR IN PARAMETER L, NL, N, LPP2, OR NMAX. /
  105 FORMAT (68H0  ERROR -- INC MATRIX IMPROPERLY SPECIFIED, OR DISAGR
     XES WITH LPP2. /)
  106 FORMAT (19H0  ERROR -- WEIGHT(, I4, 14H) IS NEGATIVE. /)
  107 FORMAT (28H0  ERROR -- CONSTANT COLUMN , I3, 37H MUST BE COMPUTED
     XONLY WHEN ISEL = 1.  /)
  108 FORMAT (33H0  CATASTROPHIC FAILURE -- COLUMN , I4, 28H IS ZERO, S
     XE DOCUMENTATION. /)
      END
      DOUBLE PRECISION FUNCTION XNORM(N, X)
C
C        COMPUTE THE L2 (EUCLIDEAN) NORM OF A VECTOR, MAKING SURE TO
C        AVOID UNNECESSARY UNDERFLOWS.  NO ATTEMPT IS MADE TO SUPPRESS
C        OVERFLOWS.
C
      DOUBLE PRECISION X(N), RMAX, SUM, TERM, DABS, DSQRT
C
C            FIND LARGEST (IN ABSOLUTE VALUE) ELEMENT
      RMAX = 0.
      DO 10 I = 1, N
         IF (DABS(X(I)) .GT. RMAX) RMAX = DABS(X(I))
```

```
   10     CONTINUE
C
       SUM = 0.
       IF (RMAX .EQ. 0.) GO TO 30
       DO 20 I = 1, N
          TERM = 0.
          IF (RMAX + DABS(X(I)) .NE. RMAX) TERM = X(I)/RMAX
   20     SUM = SUM + TERM*TERM
C
   30  XNORM = RMAX*DSQRT(SUM)
   99  RETURN
       END
```

```
0           NUMBER OF NONLINEAR PARAMETERS

  2
0           INITIAL EST. OF NONLIN. PARAMETERS

0.01
0.02
0           NUMBER OF LINEAR PARAMETERS

  3
0           NUMBER OF OBSERVATIONS

 33
0            # OF INDEPENDENT VARIABLES T

   1
OOBSERVATIONS

  1      0.844
  2      0.908
  3      0.932
  4      0.936
  5      0.925
  6      0.908
  7      0.331
  8      0.850
  9      0.818
 10      0.784
 11      0.751
 12      0.718
 13      0.GS5
 14      0.658
 15      0.628
 16      0.603
 17      0.580
 13      0.553
 19      0.533
 20      0.522
 21      0.506
 22      0.490
 23      0.478
 24      0.467
 25      0.457
 26      0.448
 27      0.438
 28      0.431
 29      0.424
 30      0.420
 31      0.414
 32      0.411
 33      0.406
OINDEPENDENT VARIABLES

  1        0.0
  2       10.0
  3       20.0
  4       30.0
  5       40.0
  6       50.0
  7       60.0
```

```
 8         70.0
 9         80.0
10         90.0
11        100.0
12        110.0
13        120.0
14        130.0
15        140.0
16        150.0
17        160.0
18        170.0
19        180.0
20        190.0
21        200.0
22        210.0
23        220.0
24        230.0
25        240.0
26        250.0
27        260.0
28        270.0
20        280.0
30        290.0
31        300.0
32        310.0
33        320.0
0 INCIDENCE MATRIX    INC(I,J)=

  0
  1
  0
  0
  0
  1
0COLUMN #1 OF A(I,J) MATRIX

 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
 1.
```

Here the given data values are fitted by the model

$$F(t, x) = x_1 + x_2 \exp(-x_4 t) + x_3 \exp(-x_5 t).$$

The data values are given in Table 1 They were supplied by Dr A. M. Sargeson of the Research School of Chemistry in the Australian National University. The progress of the algorithm is summarized in Table 2.

TABLE 1
Data for exponential fitting problem

i	t_i	y_i	i	t_i	y_i
1	0	0·844	18	170	0·558
2	10	0·908	19	280	0·538
3	20	0.932	20	190	0522
4	30	0·936	21	200	0·506
5	40	0·925	22	210	0·490
6	50	0·908	23	220	0·478
7	60	0.881	24	230	0·467
8	70	0·850	25	240	0.457
9	80	0·818	26	250	0·448
10	90	0·784	27	260	0·438
11	100	0·751	28	270	0431
12	110	0.718	29	280	0424
13	120	0·685	30	290	0.420
14	130	0·658	31	300	0.414
15	140	0·628	32	310	0·411
16	150	0·603	33	320	0·406
17	160	0·580			

TABLE 2
Summary of numerical results in exponential fitting problem

i	v	$\|\|f\|\|^2$	x_1	x_2	x_3	x_4	x_5
0	18·586	0·879 E0	0·5	1·5	-1·0	0·01	0.02
1	18·586	0·161 E0	0·4984	1.4993	–1·0007	0.01443	0.02329
2	9·293	0.103 E0	0.4888	1·4967	–1·0034	0·01693	0.02879
3	4·646	0·523 E-1	04586	1.4896	-1.0106	0.01462	0.02606
4	2.323	0·941 E-2	0.4140	1·4796	-1.0209	0·01309	0·02634
5	1·116	0771 E-3	03831	1.4799	-1.0211	0·01207	0.02573
6	0.5808	0987 E-4	0·3715	1.4866	-1.0150	0·01174	0·02511
7	0·2904	0·770 E-4	0.3691	1·4893	–1·0134	0·01167	0.02489
8	0.1452	0.766 E-4	0·3690	1·4913	-1.0152	0.01167	0.02485
9	0.0726	0754 E-4	0·3691	1·4987	–1·0227	0·01169	0.02479
10	0·0363	0.720 E-4	0·3697	1·5246	–1·0491	0·01179	0·02455
11	0·0182	0782 E-4					
	0.0272	0·684 E-4	0·3705	1·5616	-1.0867	0·01191	0·02423
12	0.0272	0·650 E-4	0.3711	1·5914	-1.1169	0·01201	0.02401
13	0.0136	0·740 E-4					
	0.0204	0629 E-4	0·3719	1·6322	-1.1158	0.01213	0.02371
14	0·0204	0·602 E-4	0.3723	1·6641	-1·1906	0·01222	0.02350
15	0·0102	0.679 E-4					
	0·0153	0.591 E 4	0.3729	1.7061	-1.2331	0.01234	0.02324
16	0.0153	0·573 E-4	0·3733	1·7380	-1.2652	0·01242	0.02306
17	0.0077	0.615 E-4					
	0·0115	0567 E-4	0.3738	1·7779	-1.3056	0·01252	0.02284
18	0.0115	0·557 E-4	0.3742	1·8071	-1.3350	0.01259	0.02270
19	0·0057	0·570 E 4					
	0·0086	0·554 E-4	0.3745	1.8413	–1·3694	0.01267	002253
20	0·0086	0·549 E-4	0.3748	1·8648	-1.3932	0·01272	0·02242
21	0·0043	0·551 E-4					
	0·0065	0·548 E-4	0.3750	1.8898	-1.4183	0.01277	0.02231
22	0·0065	0·547 E-4	0·3751	1·9054	-1.4340	0.01281	0·02225
23	0·0032	0·547 E-4	0·3753	1·9250	-1.4538	0·01285	0.02217
24	0·0016	0·547 E-4	0.3754	1·9343	-1.4631	0.01286	0·02213
25	0·0008	0·546 E-4	03754	1·9358	–1·4646	0.01287	0.02212
26	0·0004	0·546 E-4	0·3754	1·9358	–1·4647	0.01287	0.02212
27	0·0002	0·546 E-4	03754	1·9358	-1.4647	0.01287	0.02212

Match Values

```
0     0  NORM OF RESIDUAL =  0.7012746D-01
      NU =  0.1000000D 01
0  ITERATION   1    NONLTNEAR PARAMETERS
0  0.1127396D-01  0.2304597D-01
0     1  NORM OF RESIDUAL =  0.2347806D-01
      NU =  0.5000000D 00
      NORM(DELTA-ALF) / NORM(ALF) =   0.129D 00
0  ITERATION   2    NONLINEAR PARAMETERS
0  0.1185148D-01  0.2411646D-01
0     1  NORM OF RESIDUAL =  0.8501084D-02
      NU =  0.2500000D 00
      NORM(DELTA-ALF) / NORM(ALF) =   0.453D-01
0  ITERATION   3    NONLINEAR PARAMETERS
0  0.1213711D-01  0.2368091D-01
0     1  NORM OF RESIDUAL =  0.7833385D-02
      NU =  0.1250000D 00
      NORM(DELTA-ALF) / NORM(ALF) =   0.196D-01
0  ITERATION   4    NONLINEAR PARAMETERS
0  0.1250873D-01  0.2283212D-01
0     1  NORM OF RESIDUAL =  0.7489091D-02
      NU =  0.6250000D-01
      NORM(DELTA-ALF) / NORM(ALF) =   0.356D-01
0  ITERATION   5    NONLINEAR PARAMETERS
0  0.1278089D-01  0.2228172D-01
0     1  NORM OF RESIDUAL =  0.7398365D-02
      NU =  0.3125000D-01
      NORM(DELTA-ALF) / NORM(ALF) =   0.239D-01
0  ITERATION   6    NONLINEAR PARAMETERS
0  0.1285971D-01  0.2213731D-01
0     1  NORM OF RESIDUAL =  0.7392535D-02
      NU =  0.1562500D-01
      NORM(DELTA-ALF) / NORM(ALF) =   0.643D-02
0  ITERATION   7    NONLINEAR PARAMETERS
0  0.1286723D-01  0.2212332D-01
0     1  NORM OF RESIDUAL =  0.7392493D-02
      NU =  0.7812500D-02
      NORM(DELTA-ALF) / NORM(ALF) =   0.621D-03
0  ITERATION   8    NONLINEAR PARAMETERS
0  0.1286753D-01  0.2212272D-01
0     1  NORM OF RESIDUAL =  0.7392493D-02
      NU =  0.3906250D-02
      NORM(DELTA-ALF) / NORM(ALF) =   0.263D-04
0''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
0  LINEAR PARAMETERS

  0.3754100D 00  0.1935842D 01 -0.1464682D 01
0  NONLINEAR PARANETERS

  0.1286753D-01  0.2212272D-01
0''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

Chapter VI

MAIN PROGRAMS

A. SINGLE FLOW PATH

The transfer function is:

$$\tilde{C} = \frac{s}{2\sqrt{[\Pi(u/L)t\ (\eta/uL)]}}\ e^{-\frac{(1-(u/L)t)^2}{4(u/L)t(\eta/uL)}}$$

Rewriting the non-linear parameters in terms of a_j and E (a linear scaling parameter) yields:

$$\frac{\tilde{C}}{s} = \frac{E}{2\sqrt{(\Pi\alpha_1\alpha_2)t}}\ e^{-\frac{(1-\alpha_2 t)^2}{4\alpha_1\alpha_2 t}}$$

The linear parameter normalizes the flow fraction to one. This had to be done because precise information on the initial concentration was not available. This does not affect the shape of the tracer profile, only the size.

B. DUAL FLOW PATH

The function is as follows:

$$\frac{\tilde{C}}{s} = \frac{\varepsilon_1}{2\sqrt{\Pi\alpha_1\alpha_2 t}}\ e^{-\frac{(1-\alpha_2 t)^2}{4\alpha_1\alpha_2 t}} + \frac{\varepsilon_2}{2\sqrt{\Pi\alpha_3\alpha_4 t}}\ e^{-\frac{(1-\alpha_4 t)^2}{4a_3a_4 t}}$$

The linear parameters ε_1 and ε_2 divided by E are the relative flow fractions.

$$\frac{\varepsilon_1}{E} + \frac{\varepsilon_2}{E} = 1$$

The actual programs are listed in the remainder of this section. These printouts include the driving programs for Varpro for both one and two flow path cases, all input data, and the required transfer function derivatives.

A. SINGLE FLOW PATH

```
C ***************************************************************
C ***************************************************************
C
C                  PROGRAM BEGINS
C
C ***************************************************************
C ***************************************************************
C
C
C$ASSIGN 1 TO FILE WK116SC
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
C$ASSIGN 3 TO FILE TESTCASE INOUT FORMAT=F LENGTH=8000
C$ASSIGN 9 TO FILE OPTPARAM INOUT FORMAT=F LENGTH=800
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMENSIONS FOR THE INCEDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(90),T(90),ALF(2),BETA(10),W(100),A(50,5),
     *INC(12,2),C(50)
C
C
C         SET PARAMETERS FOR VARPRO.
C
C
      EXTERNAL ADA
       NMAX=50
      LPP2=5
      IPRINT=1
      IV=1
           N=50
     NL=2
      L=1
         DO 1 I=1,N
 1        W(I)=1.0
C
C
C         READ DATA FROM WK116S. SEQUENTIAL ORDERING AND
C         PROPER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (1,10) NL
 10   FORMAT(I3)
      WRITE(2,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C       ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (1,15)(ALF(I),I=1,2)
 15   FORMAT(F7.3)
      WRITE(2,20)(ALF(I),I=1,2)
 20   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F7.3))
C
```

```
C
C       L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      READ(1,22) L
 22   FORMAT(I3)
      WRITE(2,25)L
 25   FORMAT(1H0,10X,'NUMBER OF LINEAR PARAMETERS'//(I3))
C
C
C       N IS THE NUMBER OF OBSERVATIONS
C
C
      READ(1,30) N
 30   FORMAT(I4)
      WRITE(2,35) N
 35   FORMAT(1H0,10X,'NUMBER OF OBSERVATIONS'//(I3))
C
C
C       IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      READ(1,38) IV
 38   FORMAT(I4)
      WRITE(2,401 IV
 40   FORMAT(1H0,10X,' # OF INDEPENDENT VARIABLES T'//(I4))
C
C
C       Y IS THE N-VECTOR OF OCSERVATIONS
C
C
      READ(1,45)(Y(I),I=1,N)
 45   FORMAT(F7.2)
      WRITE(2,50)(Y(I),I=1,N)
 50   FORMAT(1H0,'OBSERVATIONS'//(F7.2)'
C
C
C       T IS THE INDEPENDENT VARIABLE
C
C
      READ(1,55)(T(I),I=1,N)
 55   FORMAT(F7.3)
     WRITE(2,60)(T(I),I=1,N)
 60   FORMAT(1H0,'INDEPENDENT VARIABLES'//(F6.1))
C
C
      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
C
C
C
C
       DO 23 I=1,50
 23       C(I)= 300.73*(1.0/(2.0*DSQRT(3.14159*.160*.10*T(I))))*
     * DEXP(-1.0*(((1.0-.10*T(I))**2)/(4.0*.160*.10*T(I))))
      WRITE(3,26)(I,C(I),I=1,N)
 26      FORMAT(1H0,'CALCULATED CONCENTRATION IS  '//(I4,5X,F6.2))
C
C
```

```
C
      STOP
      END
C
C
C
C
C
C ***********************************************************************
C ***********************************************************************
C
C                    SUBROUTINES
C
C ***********************************************************************
C ***********************************************************************
C
C
C
C
      SUBROUTINE ADA (L,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALF(2),A(50,5),T(50),INC(12,8),B(60,60),Y(50
     *)
C
C
C
C
C
C
C       THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C       INC(K,J)=1 IF THE NONLINEAP, PARAMETER ALF(K) APPEARS
C       IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C       INC(K,J) TO ZERO.)
C
C
      INC(1,1)=1.0
      INC(2,1)=1.0
C
C
      WRITE(2,70)(INC(I,1),I=1,2)
 70   FORMAT(1H0,' INCIDENCE MATRIX   INC(I,J)= '//(I3))
C
C
C       THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
C       THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C       B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C       J=1,L. THE CONSTANT FUNCTIONS ?HI WHICH DO NOT
C       DSPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C       APPEAR FIRST.
C
C
C
C
C
C
      IF(ISEL.EQ.2)  GO TO 90
      IF(ISEL.EQ.3)  GO TO 165
       . DO 81 I=1,N
         A(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
```

```
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
 81            CONTINUE
C
        WRITE(2,103) (A(I,1),I=1,N)
C
  103         FORMAT(1H0,'COLUMN #1 OF A(I,J) MATRIX'//(F9.5))
C
C
C
  90         DO 95 I=1,N
 95       A(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
C
        IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
 165       DO 170 I=1,N
       B(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
C
C
        A(I,3)= (B(I,1)/ALF(1))*((((1.0-ALF(2)*T(I))**2)/(4.0*
     *              ALF(1)*ALF(2)*T(I)))- 0.5)
170       A(I,4)=(B(I,1)/ALF(2))*((((1.0-(ALF(2)*T(I))**2))/(4.0*
     *          ALF(1)*ALF(2)*T(I)))-0.5)
C
       WRITE (2,180)(A(I,3),I=1,N)
 180   FORMAT(1H0,'COLUMN # 3 OF A(I,J) MATRIX'//(F9.5))
C
C
  200         CONTINUE
C
        RETURN
        END
C
C
C
C
C
C
C
C
      SUBROUTINE VARPRO (L, NL, N, NMAX, LPP2, IV, T, Y, W, ADA, A,
     X IPRINT, ALF, BETA, IERR)
```

```
0           NUMBER OF NONLINEAR PARAMETERS

  2
0           INITIAL EST. OF NONLIN. PARAMETERS

  0.290
  0.060
0           NUMBER OF LINEAR PARAMETERS

  1
0           NUMBER OF OBSERVATIONS

 50
0            # OF INDEPENDENT VARIABLES T

   1
OOBSERVATIONS

   5.00
   3.00
   3.00
   2.00
   2.00
  10.00
  41.00
  70.00
 100.00
 130.00
 165.00
 189.00
 196.00
 227.00
 228.00
 230.00
 229.00
 230.00
 229.00
 220.00
 220.00
 218.00
 210.00
 205.00
 192.00
 183.00
 170.00
 160.00
 140.00
 130.00
 140.00
 135.00
 136.00
 135.00
 137.00
 135.00
 133.00
 128.00
 113.00
 105.00
 102.00
 100.00
```

98.00
101.00
105.00
104.00
99.00
95.00
90.00
85.00

OINDEPENDENT VARIABLES

0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.3
7.6
5.0
8.3
5.6
9.0
0.3
9.6
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.3
13.4
13.6
14.0
14.3
14.6
15.0
15.5
16.0
16.5
17.0
17.3
17.6
17.8
17.9
18.0
18.3
18.6
19.0
19.5
20.0

```
0 INCIDENCE MATRIX    INC(I,J)=

   1
   1
```

```
C ***************************************************************
C ***************************************************************
C
C                  PROGRAM BEGINS
C
C ***************************************************************
C ***************************************************************
C
C
CSASSIGN 1 TO FILE WK116SC
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
C$ASSIGN 3 TO FILE TESTCRSE INOUT FORMAT=F LENGTH=8000
CSASSIGN 9 TO FILE OPTPARAM INOUT FORMAT=F LENGTH=800
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMENSIONS FOR THE INCEDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(90),T(90),ALF(2),BETA(10),W(100),A(50,5),
     *INC(12,2),C(50)
C
C
C         SET PARAMETERS FOR VARPRO.
C
C
      EXTERNAL ADA
       NMAX=50
      LPP2=5
      IPRINT=1
       MV=1
           N=50
     NL=2
      L=1
        DO 1 I=1,N
 1       W(I)=1.0
C
C
C         READ DATA FROM WK116S. SEQUENTIAL ORDERING AND
C         PROPER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (1,10) NL
 10   FORMAT(I3)
      WRITE(2,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C       ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (1,15)(ALF(I),I=1,2)
 15   FORMAT(F7.3)
      WRITE(2,20)(ALF(I),I=1,2)
 20   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F7.3))
C
```

```
C
C       L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      READ(1,22) L
 22   FORMAT(I3)
      WRITE(2,25)L
 25   FORMAT(1H0,10X,'NUMBER OF LINEAR PARAMETERS'//(I3))
C
C
C       N IS THE NUMBER OF OBSERVATIONS
C
C
      READ(1,30) N
 30   FORMAT(I4)
      WRITE(2,35) N
 35   FORMAT(1H0,10X,'NUMBER OF OBSERVATIONS'//(I3))
C
C
C       IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      READ(1,38) IV
 3s   FORMAT(I4)
      WRITE(2,40) IV
 40   FORMAT(1H0,10X,' # OF INDEPENDENT VARIABLES T'//(I4))
C
C
C       Y IS THE N-VECTOR OF OBSERVATIONS
C
C
      READ(1,45)(Y(I),I=1,N)
 45   FORMAT(F7.2)
      WRITE(2,50)(Y(I),I=1,N)
 50   FORMAT(1H0,'OBSERVATIONS'//(F7.2))
C
C
C       T IS THE INDEPENDENT VARIASLE
C
C
      READ(1,55)(T(I),I=1,N)
 55   FORMAT(F7.3)
      WRITE(2,60)(T(I),I=1,N)
 60   FORMAT( 1H0,'INDEPENDENT VARIABLES'//(F6.1))
C
C
      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
C
C
C
C
       DO 23 I=1,50
 23       C(I)= 137.69*(1.0/(2.0*DSQRT(3.14159*.253*.0705*T(I))))*
     * DEXP(-1.0*(((1.0-.0705*T(I))**2)/(4.0*.253*.070*T(I))))
      WRITE(3,26)(I,C(I),I=1,N)
 26      FORMAT(1H0,'CALCULATED CONCENTRATION IS  '//(I4,5X,F6.2))
C
C
```

```
C
      STOP
      END
C
C
C
C
C
```

```
0           NUMBER OF NONLINEAR PARAMETERS

  2
0           INITIAL EST. OF NONLIN. PARAMETERS

  0.100
  0.050
0           NUMBER OF LINEAR PARAMETERS

  1
0           NUMBER OF OBSERVATIONS

 50
0            # OF INDEPENDENT VARIABLES T

   1
OOBSERVATIONS

   0.01
   0.01
   1.00
   4.00
   a.00
  10.00
  18.00
  23.00
  33.00
  37.00
  55.00
  62.00
  63.00
  75.00
  83.00
  85.00
  8G.00
  87.00
  33.00
  88.00
  87.50
  83.00
  82.00
  79.00
  75.00
  77.00
  75.60
  70.50
  69.80
  65.00
  60.00
  65.00
  76.00
  73.00
  72.00
  70.00
  69.80
  70.00
  69.00
  65.00
  67.00
  68.00
```

69.00
68.00
72.00
71.00
72.00
'73.00
72.00
70.00

OINDEPENDENT VARIABLES

1.0
2.0
3.0
3.5
3.7
4.0
4.2
4.5
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.3
8.5
8.7
9.0
9.5
10.0
10.3
10.6
11.0
11.3
11.6
12.0
12.3
12.6
13.0
13.3
14.0
14.3
14.6
15.0
15.3
15.6
16.0
16.3
1G.6
17.0
17.3
17.6
18.0
18.3
18.6
19.0
19.5
20.0

```
0 INCIDENCE MATRIX    INC(I,J)=

  1
  1
```

```
C ***************************************************************
C ***************************************************************
C
C                  PROGRAM BEGINS
C
C ***************************************************************
C ***************************************************************
C
C
C$ASSIGN 1 TO FILE WKllGSC
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
C$ASSIGN 3 TO FILE TESTCASE INOUT FORMAT=F LENGTH=8000
C$ASSIGN 9 TO FILE OPTPARAM INOUT FORMAT=F LENGTH=800
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMENSIONS FOR THE INCEDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(90),T(90),ALF(2),BETA(10),W(100),A(50,5),
     *INC(12,2),C(50)
C
C
C         SET PARAMETERS FOR VARPRO.
C
C
      EXTERNAL ADA
       NMAX=50
      LPP2=5
      IPRINT=1
      IV=1
           N=50
     NL=2
      L=1
        DO 1 I=1,N
 1       W(I)=1.0
C
C
C         READ DATA FROM WK116S. SEQUENTIAL ORDERING AN9
C         PROPER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (1,10) NL
 10   FORMAT(I3)
      WRITE(2,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C       ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (1,15)(ALF(I),I=1,2)
 15   FORMAT(F7.3)
      WRITE(2,20)(ALF(I),I=1,2)
 20   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F7.3))
C
```

```
C
C       L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      READ(1,22) L
 22   FORMAT(I3)
      WRITE(2,25)L
 25   FORMAT(1H0,10X,'NUMBER OF LINEAR PARAMETERS'//(I3))
C
C
C       N IS THE NUMBER OF OBSERVATIONS
C
C
      READ(1,30) N
 30   FORMAT(I4)
      WRITE(2,35) N
 35   FORMAT(1H0,10X,'NUMBER OF OBSERVATIONS'//(I3))
C
C
C       IV IS THZ NUMBER OF INDEPENDENT VARIAGLES T
C
C
      READ(1,38) IV
 38   FORMAT(I4)
      WRITE(2,40) IV
 40   FORMAT(1H0,10X,' # OF INDEPENDENT VARIABLES T'//(I4))
C
C
C       Y IS THE N-VECTOR OF OESERVATIOGS
C
C
      READ(1,45)(Y(I),I=1,N)
 45   FORMAT(F7.2)
      WRITE(2,50)(Y(I),I=1,N)
 50   FORMAT(1H0,'OBSERVATIONS'//(F7.2))
C
C
C       T IS THE INDEPENDENT VARIABLE
C
C
      READ(1,55)(T(I),I=1,N)
 55   FORMAT(F7.3)
      WRITE(2,60)(T(I),I=1,N)
 60   FORMAT(1H0,'INDEPENDENT VARIABLES'//(F6.1))
C
C
      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
C
C
C
C
       DO 23 I=1,50
 23       C(I)= 18.8107*(1.0/(2.0*DSQRT(3.14159*.119*.078*T(I))))*
     * DEXP(-1.0*(((1.0-.078*T(I))**2)/(4.0*.119*.0785*T(I))))
      WRITE(3,26)(I,C(I),I=1,N)
 26      FORMAT(1H0,'CALCULATED CONCENTRATION IS  '//(I4,5X,F6.2))
C
C
```

```
C
      STOP
      END
C
C
C
C
C
C ***********************************************************************
C ***********************************************************************
C
C                    SUBROUTINES
C
C ***********************************************************************
C ***********************************************************************
C
C
C
C
      SUBROUTINE ADA (L,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALF(2),A(50,5),T(50),INC(12,8),B(60,60),Y(50
     *)
C
C
C
C
C
C
C       THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C       INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C       IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C       INC(K,J) TO ZERO.)
C
C
       INC(1,1)=1.0
       INC(2,1)=1.0
C
C
      WRITE(2,70)(INC(I,1),I=1,2)
 70   FORMAT(1H0,' INCIDENCE MATRIX   INC(I,J)= '//(I3))
C
C
C       THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
C       THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C       B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C       J=1,L. THE CONSTANT FUNCTIONS PHI WHICH DO NOT
C       DEPEND UPON ANY NOHLINEAR PARAMETERS ALF MUST
C       APPEAR FIRST.
C
C
C
C
C
       IF(ISEL.EQ.2)  GO TO 90
       IF(ISEL.EQ.3)  GO TO 165
          DO 81 I=1,N
          A(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
```

```
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
 81            CONTINUE
C
        WRITE(2,103) (A(I,1),I=1,N)
C
  103         FORMAT(1H0,'COLUMN #1 OF A(I,J) MATRIX'//(F9.5))
C
C
C
  90         DO 95 I=1,N
 95       A(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
C
        IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
 165        DO 170 I=1,N
      B(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
C
C
        A(I,3)= (B(I,1)/ALF(1))*((((1.0-ALF(2)*T(I))**2)/(4.0*
     *              ALF(1)*ALF(2)*T(I)))- 0.5)
170       A(I,4)=(B(I,1)/ALF(2))*((((1.0-(ALF(2)*T(I))**2))/(4.0*
     *          ALF(1)*ALF(2)*T(I)))-0.5)
C
       WRITE (2,180)(A(I,3),I=1,N)
 180   FORMAT(1H0,'COLUMN # 3 OF A(I,J) MATRIX'//(F9.5))
C
C
  200        CONTINUE
C
        RETURN
        END
C
C
C
C
C
C
C
C
      SUBROUTINE VARPRO (L, NL, N, NMAX, LPP2, IV, T, Y, W, ADA, A,
     X IPRINT, ALF, BETA, IERR)
```

```
0           NUMBER OF NONLINEAR PARAMETERS

  2
0           INITIAL EST. OF NONLIN. PARAMETERS

  0.101
  0.068
0           NUMBER OF LINEAR PARAMETERS

  1
0           NUMBER OF OBSERVATIONS

 50
0            # OF INDEPENDENT VARIABLES T

   1
COBSERVATIONS

   0.01
   0.01
   0.01
   0.01
   1.30
   1.50
   3.20
   5.00
   6.50
   7.20
  10.00
  10.20
  11.10
  13.80
  12.50
  13.60
  14. 10
  14.30
  15.80
  15.00
  15.80
  14.90
  15.00
  15.00
  15.30
  15.60
  15.70
  15.30
  15.20
  14.50
  14.40
  14.50
  14.70
  13. 10
  12.30
  13.70
  15.00
  17.00
  14.40
  13.00
  10.80
  11.80
```

10.00
10.10
10.50
12.80
10.10
10.20
9.80
8.50
0INDEPENDENT VARIABLES

1.0
2.0
3.0
4.0
5.0
5.3
5.6
6.0
G.3
6.6
7.0
7.2
7.4
7.6
7.3
8.0
8.2
8.4
8.6
9.0
3.2
9.4
9.6
9.8
10.0
10.3
10.6
11.0
11.3
11.6
12.0
12.3
12.6
13.0
13.3
13.5
14.0
14.3
14.G
15.0
15.5
16.0
16.5
17.0
17.5
18.0
19.0
19.5
19.7
20.0

```
0 INCIDENCE MATRIX    INC(I,J)=

   1
   1
```

B. DUAL FLOW PATH

```
C ***************************************************************
C ***************************************************************
C
C                  PROGRAM BEGINS
C
C ***************************************************************
C ***************************************************************
C
C
C$ASSIGN 1 TO FILE WK116SC
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
C$ASSIGN 3 TO FILE TESTCASE INOUT FORMAT=F LENGTH=8000
CSASSIGN 9 TO FILE OPTPARAM INOUT FORMAT=F LENGTH=800
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMENSIONS FOR THE INCIDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(50),T(50),ALF(4),BETA(10),W(100),A(50,8),
     *INC(12,2),C(50,30),D(60,60)
C
C
C         SET PARAMETERS FOR VARPRO.
C
C
      EXTERNAL ADA
      NMAS = 50
       LPP2=8
      IPRINT=1
      IV=1
      N=50
      NL=4
      L=2
        30 1 I=1,N
 1       W(I)=1.0
C
C
C         READ DATA FROM WK116S. SEQUENTIAL ORDERING AND
C         PROPER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (1,10) NL
 10   FORMAT(I3)
      WRITE(2,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C       ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (1,15)(ALF(I),I=1,4)
 15   FORMAT(F7.3)
      WRITE(2,20)(ALF(I),I=1,4)
 20   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F7.3))
C
```

```
C
C       L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      READ(1,22) L
 22   FORMAT(I5)
      WRITE(2,25)L
 25   FORMAT(1H0,10X,'NUMBER OF LINEAR PARAMETERS'//(I3))
C
C
C       N IS THE NUMBER OF OBSERVATIONS
C
C
      READ(1,30) N
 30   FORMAT(I4)
      WRITE(2,35) N
 35   FORMAT(1H0,10X,'NUMBER OF OBSERVATIONS'//(I3))
C
C
C       IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      READ(1,38) IV
 38   FORMAT(I4)
      WRITE(2,40) IV
 40   FORMAT(1H0,10X,' # OF INDEPENDENT VARIABLES T'//(I4))
C
C
C       Y IS THE N-VECTOR OF OBSERVATIONS
C
C
      READ(1,45)(Y(I),I=1,50)
 45   FORMAT(F7.2)
      WRITE(2,50)(Y(J),J=1,N)
 50   FORMAT(1H0,'OBSERVATIONS'//(F7.2))
C
C
C       T IS THE INDEPENDENT VARIABLE
C
C
      READ(1,55)(T(I),I=1,N)
 55   FORMAT(F7.3)
      WRITE(2,60)(T(I),I=1,N)
 60   FORMAT(1H0,'INDEPENDENT VARIABLES'//(F6.1))
C
C
      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
          ALF(1)=.024
          ALF(2)=.052
          ALF(3)=.107
          ALF(4)=.111
C
         DO 8 I=1,N
         C(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
         C(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
```

```
C
      C(I,3)=(  29.63*C(I,1)+   261.62*C(I,2))
 8      CONTINUE
C
C
C
C
C
C
C
       WRITE(3,13)(C(I,3),I=1,N)
 13    FORMAT(1H0,'THE CALCULATED CONCENTRATION IS ',(F9.4))
C
        STOP
        END
C
C
C ***********************************************************************
C ***********************************************************************
C
C                     SUBROUTINES
C
C ***********************************************************************
C ***********************************************************************
C
C
C
C
       SUBROUTINE ADA (L,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION ALF(4),A(50,8),T(50),INC(12,8),B(60,60),Y(50
     *),D(60,60)
C
C
C
C
C
C
C       THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C       INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C       IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C       INC(K,J) TO ZERO.)
C
C
         INC(1,1)=1.0
         INC(1,2)=0.0
         INC(2,1)=1.0
         INC(2,2)=0.0
         INC(3,1)=0.0
         INC(3,2)=1.0
         INC(4,1)=0.0
         INC(4,2)=1.0
C
C
       WRITE(2,70)((INC(I,J),J=1,2),I=1,4)
 70    FORMAT(1H0,' INCIDENCE MATRIX   INC(I,J)= '//(I3))
C
C
C       THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
```

```
C       THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C       B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C       J=1,L. THE CONSTANT FIJNCTIONS PHI WHICH DO NOT
C       DEPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C       APPEAR FIRST.
C
C
C
C
C
C
      IF(ISEL.EQ.2) GO TO 90
      IF(ISEL.EQ.3) GO TO 165
 90         DO 81 I=1,N
        A(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
          A(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
C
C
 81       CONTINUE
C
C
       WRITE(2,103) (A(I,1),I=1,N)
C
 103         FORMAT(1H0,'COLUMN #1 OF A(I,J) MATRIX'//(F9.5))
       WRITE(2,104) (A(I,2),I=1,N)
 104       FORMAT(1H0,'COLUMN #2 OF A(I,J) MATRIX'//(F9.5))
       IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
 165      DO 170 I=1,N
      D(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
       D(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
C
        A(I,4)= (D(I,1)/ALF(1))*((((1.0-ALF(2)*T(I))**2)/(4.0*
     *              ALF(1)*ALF(2)*T(I)))- 0.5)
       A(I,5)=(D(I,1)/ALF(2))*((((1.0-(ALF(2)*T(I))**2))/(4.0*
     *          ALF(1)*ALF(2)*T(I)))-0.5)
C
        A(I,6)= (D(I,2)/ALF(3))*((((1.0-ALF(4)*T(I))**2)/(4.0*
     *              ALF(3)*ALF(4)*T(I)))- 0.5)
       A(I,7)=(D(I,2)/ALF(4))*((((1.0-(ALF(4)*T(I))**2))/
     *          (4.0*ALF(3)*ALF(4)*T(I)))-0.5)
C
 170          CONTINUE
C
  200      CONTINUE
C
       RETURN
       END
```

```
C
C
C
C
C
C
C
C
C
      SUBROUTINE VARPRO (L, NL, N, NMAX, LPP2, IV, T, Y, W, ADA, A,
     X IPRINT, ALF, BETA, IERR)
```

```
0           NUMBER OF NONLINEAF! PARAMETERS

  4
0           INITIAL EST. OF NONLIN. PARAMETERS

  0.290
  0.060
  0.210
  0.120
0           NUMBER OF LINEAR PARAMETERS

  2
0           NUMBER OF OBSERVATIONS

 50
0            # OF INDEPENDENT VARIABLES T

   1
0OBSERVATIONS

   5.00
   3.00
   3.00
   2.00
   2.00
  10.00
  41.00
  73.00
 100.30
 130.00
 105.00
 189.00
 196.00
 227.00
 228.00
 230.00
 229.00
 230.00
 229.00
 220.00
 220.00
 215.00
 210.00
 205.00
 102.00
 183.00
 170.00
 160.00
 140.00
 130.00
 140.00
 135.30
 136.00
 138.00
 137.00
 135.00
 133.00
 128.00
 113.00
 105.00
```

102.00
100.00
98.00
101.00
105.00
104.00
99.00
95.00
90.00
85.00
OINDEPENDENT VARIABLES

0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.3
7.6
8.0
5.3
8.6
9.0
9.3
9.6
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.3
13.4
13.6
14.0
14.3
14.6
15.0
15.5
16.0
16.5
17.0
17.3
17.6
17.8
17.9
15.0
18.3
18.6
19.0

```
   19.5
   20.0
0 IKCIDENCE MATRIX    INC(I,J)=

   1
   0
   1
   0
   0
   1
   0
   1
```

```
C ****************************************************************
C ****************************************************************
C
C                 PROGRAM BEGINS
C
C ****************************************************************
C ****************************************************************
C
C
CSASSIGN 1 TO FILE WKllGSC
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
C$ASSIGN 3 TO FILE TESTCASE INOUT FORMAT=F LENGTH=8000
C$ASSIGN 9 TO FILE OPTPARAM INOUT FORMAT=F LENGTH=800
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMEKSIONS FOR THE INCIDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(50),T(50),ALF(4),BETA(10),W(100),A(50,8),
     *INC(12,2),C(50,30),D(60,60)
C
C
C         SET PARAMETERS FOR VARPRO.
C
C
      EXTERNAL ADA
      NMAX = 50
       LPP2=8
      IPRINT=1
      IV=1
      N=50
      NL=4
      L=2
        DO 1 I=1,N
 1       W(I)=1.0
C
C
C         READ DATA FROM WK116S. SEQUENTIAL ORDERING AND
C         PROPER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (1,10) NL
 10   FORMAT(I3)
      WRITE(2,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C       ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (1,15)(ALF(I),I=1,4)
 15   FORMAT(F7.3)
      WRITE(2,20)(ALF(I),I=1,4)
 20   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F7.3))
C
```

```
C
C       L IS THE NUMBER OF LINEAR PARAMETERS
C
C
        READ(1,22) L
 22     FORMAT(I3)
        WRITE(2,25)L
 25     FORMAT(1H0,10X,'NUMBER OF' LINEAR PARAMETERS'//(I3))
C
C
C       N IS THE NUMBER OF OBSERVATIONS
C
C
        READ(1,30) N
 30     FORMAT(I4)
        WRITE(2,35) N
 35     FORMAT(1H0,10X,'NUMBER OF OBSERVATIONS'//(I3))
C
C
C       IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
        READ(1,38) IV
 38     FORMAT(I4)
        WRITE(2,40) IV
 40     FORMAT(1H0,10X,' # OF INDEPENDENT VARIABLES T'//(I4))
C
C
C       Y IS THE N-VECTOR OF OBSERVATIONS
C
C
        READ(1,45)(Y(I),I=1,50)
 45     FORMAT(F7.2)
        WRITE(2,50)(Y(J),J=1,N)
 50    FORMAT(1H0,'OBSERVATIONS'//(F7.2))
C
C
C       T IS THE INCEPENDENT VARIABLE
C
C
        READ(1,55)(T(I),I=1,N)
 55     FORMAT(F7.3)
       WRITE(2,60)(T(I),I=1,N)
 60    FORMAT(1H0,'INDEPENDENT VARIABLES'//(F6.1))
C
C
        CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
      *IPRINT,ALF,BETA,IERR)
C
            ALF(1)=.063
            ALF(2)=.045
            ALF(3)=.076
            ALF(4)=.112
C
           DO 8 I=1,N
           C(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
           C(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
```

```
C
      C(I,3)=(  57.2*C(I,1)+   81.98*C(I,2))
 8      CONTINUE
C
C
C
C
C
C
C
       WRITE(3,13)(C(I,3),I=1,N)
 13    FORMAT(1H0,'THE CALCULATED CONCENTRATION IS ',(F9.4))
C
        STOP
        END
C
C
C ***********************************************************************
C ***********************************************************************
C
C                    SUBROUTINES
C
C ***********************************************************************
C ***********************************************************************
C
C
C
C
       SUSROUTINE ADA (L,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
CSASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION ALF(4),A(50,8),T(50),INC(12,8),B(60,60),Y(50
     *),D(60,60)
C
C
C
C
C
C
C       THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C       INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C       IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C       INC(K,J) TO ZERO.)
C
C
         INC(1,1)=1.0
         INC(1,2)=0.0
         INC(2,1)=1.0
         INC(2,2)=0.0
         INC(3,1)=0.0
         INC(3,2)=1.0
         INC(4,1)=0.0
         INC(4,2)=1.0
C
C
       WRITE(2,70)((INC(I,J),J=1,2),I=1,4)
 70    FORMAT(1H0,' INCIDENCE MATRIX   INC(I,J)= '//(I3))
C
C
C       THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
```

```
C       THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C       B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C       J=1,L. THE CONSTANT FUNCTIONS PHI WHICH DO NOT
C       DEPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C       APPEAR FIRST.
C
C
C
C
C
C
      IF(ISEL.EQ.2) GO TO 90
      IF(ISEL.EQ.3) GO TO 165
   90       DO 81 I=1,N
         A(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
         A(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     *  DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
C
C
   81       CONTINUE
C
C
      WRITE(2,103) (A(I,1),I=1,N)
C
  103       FORMAT(1H0,'COLUMN #1 OF A(I,J) MATRIX'//(F9.5))
      WRITE(2,104) (A(I,2),I=1,N)
  104       FORMAT(1H0,'COLUMN #2 OF A(I,J) MATRIX'//(F9.5))
      IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
  165       DO 170 I=1,N
       D(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
       D(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
C
       A(I,4)= (D(I,1)/ALF(1))*((((1.0-ALF(2)*T(I))**2)/(4.0*
     *              ALF(1)*ALF(2)*T(I)))- 0.5)
       A(I,5)=(D(I,1)/ALF(2))*((((1.0-(ALF(2)*T(I))**2))/(4.0*
     *          ALF(1)*ALF(2)*T(I)))-0.5)
C
       A(I,6)= (D(I,2)/ALF(3))*((((1.0-ALF(4)*T(I))**2)/(4.0*
     *              ALF(3)*ALF(4)*T(I)))- 0.5)
       A(I,7)=(D(I,2)/ALF(4))*((((1.0-(ALF(4)*T(I))**2))/
     *          (4.0*ALF(3)*ALF(4)*T(I)))-0.5)
C
  170            CONTINUE
C
  200       CONTINUE
C
       RETURN
       END
```

```
C
C
C
C
C
C
C
C
      SUBROUTINE VARPRO (L, NL, N, NMAX, LPP2, IV, T, Y, W, ADA, A,
     X IPRINT, ALF, BETA, IERR.)
C
```

```
0            NUMBER OF NONLINEAR PARAMETERS

  4
0            INITIAL EST. OF NONLIN. PARAMETERS

  0.100
  0.050
  0.070
  0.270
0            NUMBER OF LINEAR PARAMETERS

  2
0            NUMBER OF OBSERVATIONS

 50
0             # OF INDEPENDENT VARIABLES T

   1
OOBSERVATIONS

   0.01
   0.01
   1.00
   4.00
   8.00
  10.00
  18.00
  23.00
  33.00
  37.00
  55.00
  62.00
  C8.00
  75.00
  33.00
  85.00
  86.00
  87.00
  83.00
  83.00
  87.50
  83.00
  82.00
  73.00
  78.00
  77.00
  75.60
  70.50
  69.80
  65.00
  60.00
  65.00
  76.00
  73.00
  72.00
  70.00
  69.80
  70.00
  69.00
  65.00
```

67.00
68.00
69.00
68.00
72.00
71.00
72.00
73.00
72.00
70.00

OINDEPENDENT VARIABLES

1.0
2.0
3.0
3.5
3.7
11.0
4.2
4.5
4.8
5.0
5.5
6.0
6.5
7.0
7.5
8.0
3.3
8.5
8.7
9.0
9.5
10.0
10.3
10.6
11.0
11.3
11.6
12.0
12.3
12.6
13.0
13.3
14.0
14.3
14.6
15.0
15.3
15.6
16.0
16.3
16.6
17.0
17.3
17.6
18.0
18.3
18.6
19.0

```
   19.5
   20.0
0 INCIDENCE MATRIX    INC(I,J)=

   1
   0
   1
   0
   0
   1
   0
   1
```

```
C ***************************************************************
C ***************************************************************
C
C                  PROGRAM BEGINS
C
C ***************************************************************
C ***************************************************************
C
C
C$ASSIGN 1 TO FILE WK116SC
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
C$ASSIGN 3 TO FILE TESTCASE INOUT FORMAT=F LENGTH=8000
C$ASSIGN 9 TO FILE OPTPARAM INOUT FORMAT=F LENGTH=800
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C       SET DIMENSIONS FOR VARPRO. BE CAREFUL WHEN SETTING THE
C       DIMENSIONS FOR THE INCIDENCE MATRIX INC. SEE NOTE.
C
      DIMENSION Y(50),T(50),ALF(4),BETA(10),W(100),A(50,8),
     *INC(12,2),C(50,30),D(60,60)
C
C
C         SET PARAMETERS FOR VARPRO.
C
C
      EXTERNAL ADA
      NMAX = 50
       LPP2=8
      IPRINT=1
      IV=1
      N=50
      NL=4
      L=2
        DO 1 I=1,N
 1       W(I)=1.0
C
C
C         READ DATA PROM WK116S. SEQUENTIAL ORDERING AND
C         PROPER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR PRRAMGTERS
C
C
      READ (1,10) NL
 10   FORMAT(I3)
      WRITE(2,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C       ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (1,15)(ALF(I),I=1,4)
 15   FORMAT(F7.3)
      WRITE(2,20)(ALF(I),I=1,4)
 20   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F7.3))
C
```

```
C
C        L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      READ(1,22) L
 22   FORMAT(I3)
      WRITE(2,25)L
 25   FORMAT(1H0,10X,'NUMBER OF LINEAR PARAMETERS'//(I3))
C
C
C        N IS THE NUMBER OF OBSERVATIONS
C
C
      READ(1,30) N
 30   FORMAT(I4)
      WRITE(2,35) N
 35   FORMAT(1H0,10X,'NUMBER OF OBSERVATIONS'//(I3))
C
C
C        IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      READ(1,38) IV
 38   FORMAT(I4)
      WRITE(2,40) IV
 40   FORMAT(1H0,10X,' # OF INDEPENDENT VARIABLES T'//(I4))
C
C
C       Y IS THE N-VECTOR OF OBSERVATIONS
C
C
      READ(1,45)(Y(I),I=1,50)
 45   FORMAT(F7.2)
      WRITE(2,50)(Y(J),J=1,N)
 50  FORMAT(1H0,'OBSERVATIONS'//(F7.2))
C
C
C       T IS THE INDEPENDENT VARIABLE
C
C
      READ(1,55)(T(I),I=1,N)
 55   FORMAT(F7.3)
     WRITE(2,60)(T(I),I=1,N)
 60   FORMAT(1H0,'INDEPENDENT VARIABLES'//(F6.1))
C
C
      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
          ALF(1)=.101
          ALF(2)=.068
          ALF(3)=.023
          ALF(4)=.119
C
         DO 8 I=1,N
         C(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
         C(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
```

```
C
      C(I,3)=(  14.8*C(I,1)+   3.64*C(I,2))
           C(I,4)=14.8*C(I,1)
           C(I,5)= 3.64*C(I,2)
 8       CONTINUE
C
C
C
C
C
C
C
       WRITE(3,13)(C(I,3),I=1,N)
 13    FORMAT(1H0,'THE CALCULATED CONCENTRATION IS ',(F9.4))
       WRITE(3,14)(C(I,4),I=1,N)
 14    FORMAT(1H0,'COMPONENT #1 CONCENTRATION IS',(F9.4))
       WRITE(3,17)(C(I,5),I=1,N)
 17    FORMAT(1H0,'COMPONENT #2 CONCENTRATION IS',(F9.4))
C
        STOP
        END
C
C
C ***************************************************************
C ***************************************************************
C
C                    SUBROUTINES
C
C ***************************************************************
C ***************************************************************
C
C
C
C
       SUBROUTINE ADA (L,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
C$ASSIGN 2 TO FILE RESULT INOUT FORMAT=F LENGTH=80
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION ALF(4),A(50,8),T(50),INC(12,8),B(60,60),Y(50
     *),D(60,60)
C
C
C
C
C
C
C       THE INCICENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C       INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C       IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C       INC(K,J) TO ZERO.)
C
C
         INC(1,1)=1.0
         INC(1,2)=0.0
         INC(2,1)=1.0
         INC(2,2)=0.0
         INC(3,1)=0.0
         INC(3,2)=1.0
         INC(4,1)=0.0
         INC(4,2)=1.0
C
```

```
C
      WRITE(2,70)((INC(I,J),J=1,2),I=1,4)
 70   FORMAT(1H0,' INCIDENCE MATRIX   INC(I,J)= '//(I3))
C
C
C       THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
C       THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C       B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C       J=1,L. THE CONSTANT FUNCTIONS PHI WHICH DO NOT
C       DEPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C       APPEAR FIRST.
C
C
C
C
C
C
      IF(ISEL.EQ.2) GO TO 90
      IF(ISEL.EQ.3) GO TO 165
 90       DO 81 I=1,N
        A(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
        A(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
C
C
 81       CONTINUE
C
C
      WRITE(2,103) (A(I,1),I=1,N)
C
 103      FORMAT(1H0,'COLUMN #1 OF A(I,J) MATRIX'//(F9.5))
      WRITE(2,104) (A(I,2),I=1,N)
 104      FORMAT(1H0,'COLUMN #2 OF A(I,J) MATRIX'//(F9.5))
      IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
 165      DO 170 I=1,N
      D(I,1)= (1.0/(2.0*DSQRT(3.14159*ALF(1)*ALF(2)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(2)*T(I))**2)/(4.0*ALF(1)*ALF(2)*T(I))))
      D(I,2)= (1.0/(2.0*DSQRT(3.14159*ALF(3)*ALF(4)*T(I))))*
     * DEXP(-1.0*(((1.0-ALF(4)*T(I))**2)/(4.0*ALF(3)*ALF(4)*T(I))))
C
C
      A(I,4)= (D(I,1)/ALF(1))*((((1.0-ALF(2)*T(I))**2)/(4.0*
     *             ALF(1)*ALF(2)*T(I)))- 0.5)
      A(I,5)=(D(I,1)/ALF(2))*((((1.0-(ALF(2)*T(I))**2))/(4.0*
     *          ALF(1)*ALF(2)*T(I)))-0.5)
C
      A(I,6)= (D(I,2)/ALF(3))*((((1.0-ALF(4)*T(I))**2)/(4.0*
     *             ALF(3)*ALF(4)*T(I)))- 0.5)
      A(I,7)=(D(I,2)/ALF(4))*((((1.0-(ALF(4)*T(I))**2))/
     *          (4.0*ALF(3)*ALF(4)*T(I)))-0.5)
C
```

```
  170             CONTINUE
C
  200       CONTINUE
C
          RETURN
          END
C
C
C
C
C
C
C
C
      SUBROUTINE VARPRO (L, NL, N, NMAX, LPP2, IV, T, Y, W, ADA, A,
     X IPRINT, ALF, BETA, IERR)
```

```
0           NUMBER OF NONLINEAR PARAMETERS

  4
0           INITIAL EST. OF NONLIN. PARAMETERS

  0.101
  0.068
  0.060
  0.119
0           NUMBER OF LINEAR PARAMETERS

  2
0           NUMBER OF OBSERVATIONS

 50
0            # OF INDEPENDENT VARIABLES T

   1
OOBSERVATIONS

    0.01
    0.01
    0.01
    0.01
    1.30
    1.53
    3.20
    5.00
    6.50
    7.20
   10.00
   10.20
   11.10
   13.80
   12.50
   13.60
   14.10
   14.30
   15.80
   15.00
   15.80
   14.90
   15.00
   15.00
   15.30
   15.60
   15.70
   15.30
   15.20
   14.50
   14.40
   14.80
   14.70
   13. 10
   12.30
   13.70
   15.00
   17.00
   14.40
   13.00
```

10.80
11.80
10.00
10.10
10.50
12.80
10.10
10.20
9.80
8.50
OINDEPENDENT VARIABLES

1.0
2.0
3.0
4.0
5.0
5.3
5.6
6.0
6.3
6.6
7.0
7.2
7.Q
7.6
7.8
8.0
8.2
3.4
5.6
3.0
9.2
3.4
9.6
9.5
10.0
10.3
10.6
11.0
11.3
11.6
12.0
12.3
12.6
13.0
13.3
13.6
14.0
14.3
14.6
15.0
15.5
16.0
16.5
17.0
17.5
1s.0
19.0
19.5

```
   19.7
   20.0
0 INCIDENCE MATRIX    INC(I,J)=

   1
   0
   1
   0
   0
   1
   0
   1
```

Chapter VII

DATA ANALYSIS

A. DATA MATCH

The tracer profile data were fitted to both the one- and two-component transfer functions (Eqs. 56 and 57). Table 7.1 compares the norm of the residual (Eq. 1) after using each model. The results show that incorporating two flow paths improves the goodness-of-fit significantly.

TABLE 7.1

1-Component Model

Well	Norm
WK 116	85.0
WK 76	60.9
WK 108	9.5

2-Component Model

Well	Norm	% Reduction
WK 116	36.8	56.0
WK 76	19.8	67.0
WK 108	6.4	32.0

The maximum concentration of the tracer profile occurs at the average residence time. The mean value of $\bar{u}t/L$ and setting the derivative equal to zero. Thus,

$$\bar{t} = \frac{1}{\frac{\bar{u}}{L}\sqrt{\left(\frac{\eta}{uL}\right)^2 + \left(\frac{\eta}{uL}\right) + 1}}$$

A comparison of calculated and observed values is made in Tables 7.2 and 7.3.

TABLE 7.2

1-Component Model

Well	$\bar{t}$, Calculated/days	$\bar{t}$, Observed/days	% Error
WK 116	9.18	7.6	17
WK 76	12.35	8.7	29
WK 108	12.00	10.0	16

TABLE 7.3

2-Component Model

Well	Flow Fraction, $\bar{t}_1$	$\bar{t}_1$, Calculated	$\bar{t}_2$, Calculated	$\bar{t}$ Observed	% Error in $\bar{t}_1$
WK 116	.87	8.5	18.9	7.6	10
WK 76	.58	8.6	21.5	8.7	1
WK 108	.20	8.3	13.9	12.0	13

The calculated initial peaks are considerably closer to the observed values in the two-component model. Secondary peaks are not well matched even though the curve-fit is better. From Figs. 7.1 through 7.3 it can be seen that the data scatters more after the maximum concentration is attained.

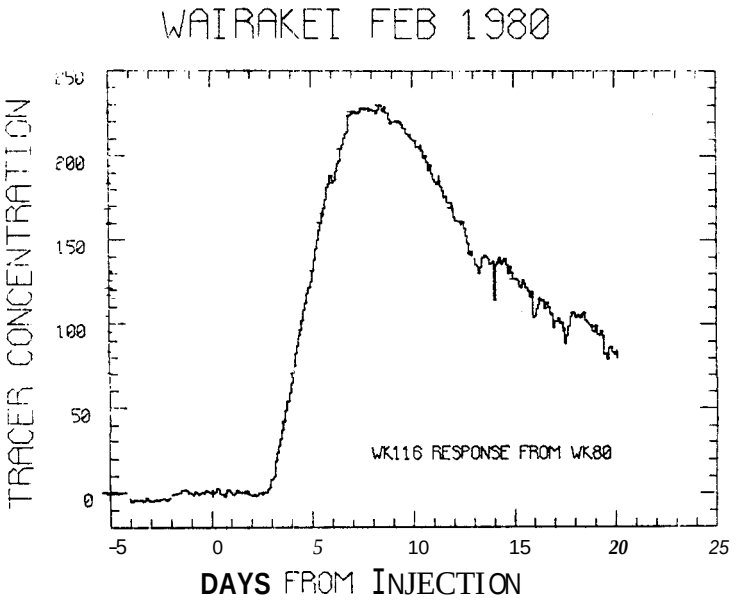


Fig. 7.1

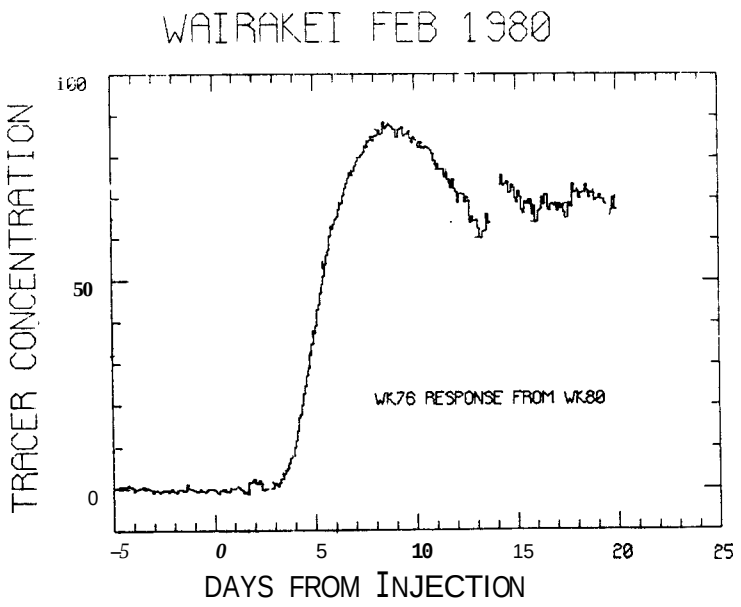


Fig. 7.2

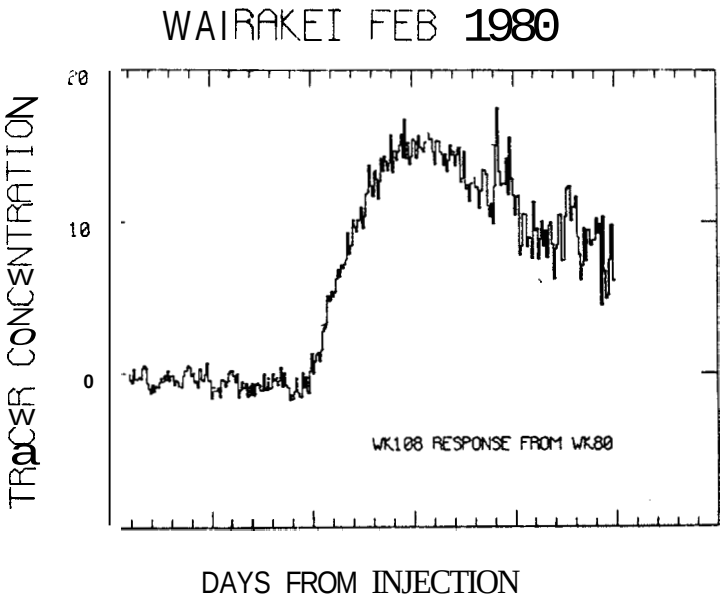


Fig. 7.3

It is difficult to determine the secondary peaks from the empirical data as well. This may account for the poor correlation. However, if the number of components were increased, it is likely that a better fit would result. It should also be noted that the larger flow fraction does not always correspond to the initial peak. In WK 108, the smaller volume flow path contributes more to the maximum concentration. Therefore, caution should be exercised when analyzing average residence times.

B. WAIRAKEI FIELD ANALYSIS

As indicated in Table 7.4, the total returns after tracer injection into WK 80 for the WK 76 and WK 116 observation wells were about ten times higher than those into WK 101. This suggests that flow along the Kaiapo fault from WK 80 to WK 116 is greater than flow within the fault block from WK 101 to WK 116.

TABLE 7.4[13]

Tracer Recovery From Injection in WK 101

Well	Days to 10% of Peak	Days to Peak Arrival	Peak Concentration	Total Recovery, %
WK 116	2.5	7.5	23	.05
WK 76	2.5	7--12	10	.05

Tracer Recovery From Injection in WK 80

WK 116	3.3	7.6	230	0.40
WK 76	4.0	8.7	88	0.24
WK 108	5.5	10.0	16	0.06

A similar effect is observed between the inflow wells and WK 76. Therefore, focussing the interpretation of the model's results on flow along or within faults seems reasonable. Figure 7.4 illustrates an approximate plan view of the fault structure.

If inter-well flow along faults is dominant, then one would expect longer average residence times and more longitudinal dispersion. Table 7.5 lists the results for one flow path model. Figures 7.5 through 7.7 illustrate the tracer profiles. In all three wells, the Peclet numbers are about the same, while there are significant differences in average residence times. The mean time differences accord with what is observed geologically: namely, the flow path between WK 116 and WK 80 is along or within the Kaiapo fault, whereas the flow paths to WK 76 and WK 108 are smaller transverse faults. Although the one component model cannot distinguish between intrafault flow or multi-channeling along a fault,

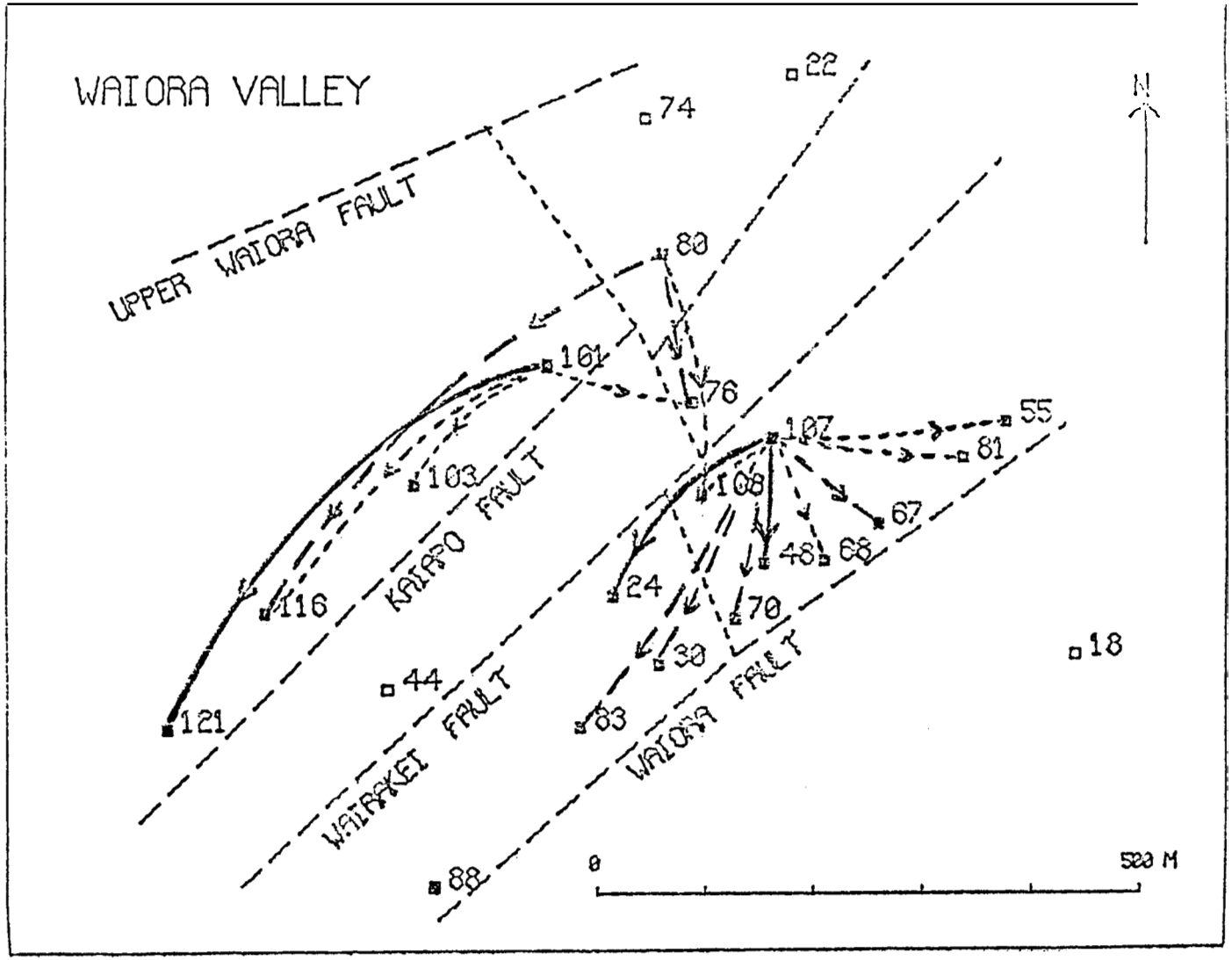

WAIORA VALLEY
UPPER WAIORA FAULT
KAIAPO FAULT
WAIRAKEI FAULT
WAIORA FAULT
N
0
500 M
22
74
80
101
76
107
55
81
103
108
67
48
68
116
24
70
30
18
44
121
83
88

Fig. 7.4

For All Figures:

Calculated Values: ——

Observed Values: ▢

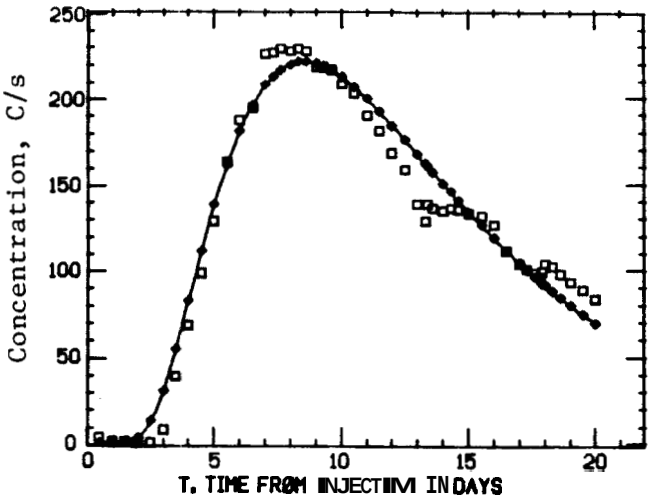


Fig. 7.5: WK 110

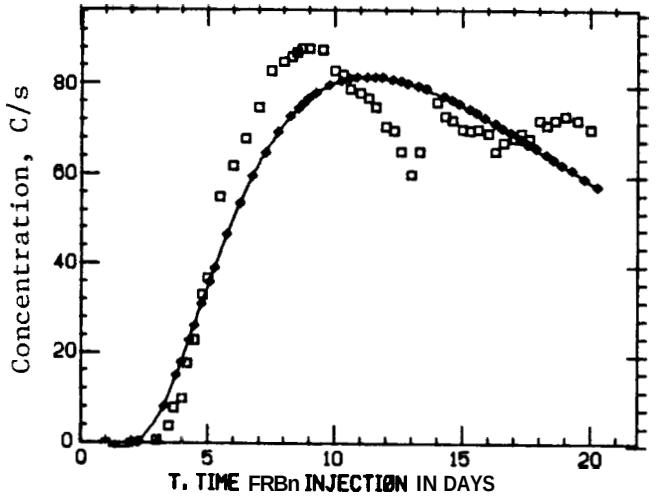


Fig. 7.6: WK 76

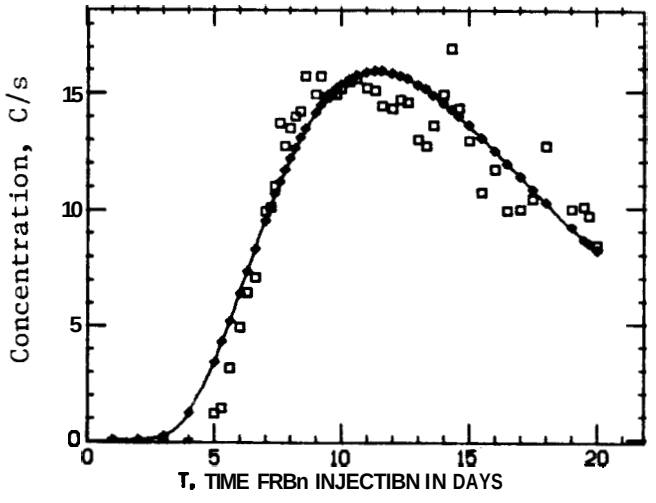


Fig. 7.7: WK 108

TAHLE 7.5

1-Component Model

Well	$\bar{t}$, Calculated	Peclet Number	Injector-Producer Length
WK 116	9.18	6.25	500 m
WK 76	12.35	3.95	145 m
WK 108	12.00	8.4	230 m

the low Peclet numbers indicate that more than one flow path is probable. If flow were primarily one-dimensional within a fault or fracture, the order of magnitude of the Peclet numbers would be 400 for a 1 cm fracture.

In the two-component case, if the observed tracer profile were dominated by mixing from different channels along the fault, then the expected Peclet number would still be quite small, even though effective dispersion within individual flow paths was substantially lower. Table 7.6 enumerates the two-component parameters and Figs. 7.8 through 7.16 show the respective tracer profiles.

In every instance, the larger flow fraction corresponds to a smaller Peclet number. This suggests that the primary flow paths have larger aperature and faster flow, since the Peclet number is proportional to $L/(b^2u)$. The smaller flow paths with larger Peclet numbers could be longer or narrower fractures. WK 116, for example, has a smaller volume path with an average residence time of 18 days. This would support the conclusion that it is either longer or has slower flow.

The two flow fractions in WK 76 are approximately the same, but the mean residence times are much different. The observed dispersion is

TABLE 7.6

2-Component Model

Well	Flow Fraction	$\bar{t}$, Calculated	Peclet Number	Injector-Producer Length
WK 116	.87	8.5	*9.3*	500 m
	.13	18.9	*41.6*	
WK 76	*.42*	21.5	15.80	145 m
	.58	8.6	*13.15*	
WK 108	.80	8.30	*9.9*	230 m
	.20	13.95	*43.4*	

similarly low, so both paths are probably highly channeled. The different mean residence times imply two different path lengths or flow speeds.

Once the Peclet numbers and average residence times have been computed, the physical characteristics of the fracture system can be described. Since Pe = uL/η, it can be rewritten as:

$$Pe = \frac{105}{2} \left(\frac{L}{\bar{u}} \right) \left(\frac{D}{b^2} \right)$$

Rearranging:

$$b = \sqrt{\left(\frac{105}{2} \right) D \left(\frac{L}{\bar{u}} \right) \frac{1}{Pe}}$$

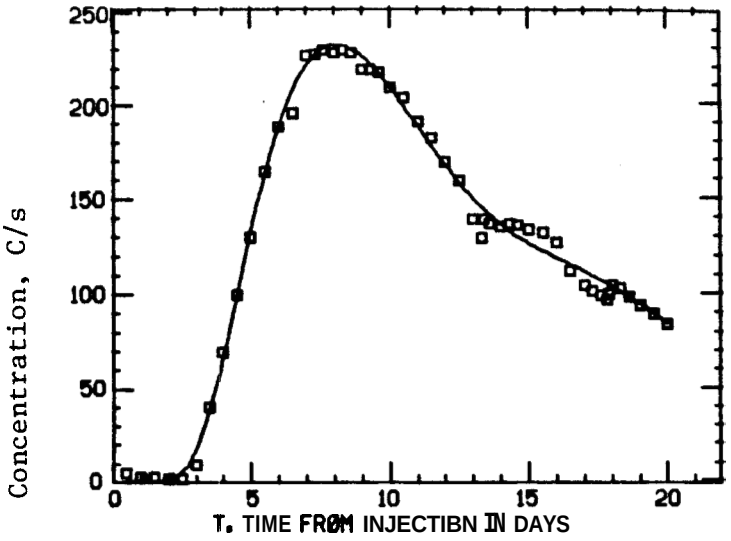
where:

$$D = 8.64 \times 10^{-6} \text{ m}^2/\text{day}$$

For All Figures:

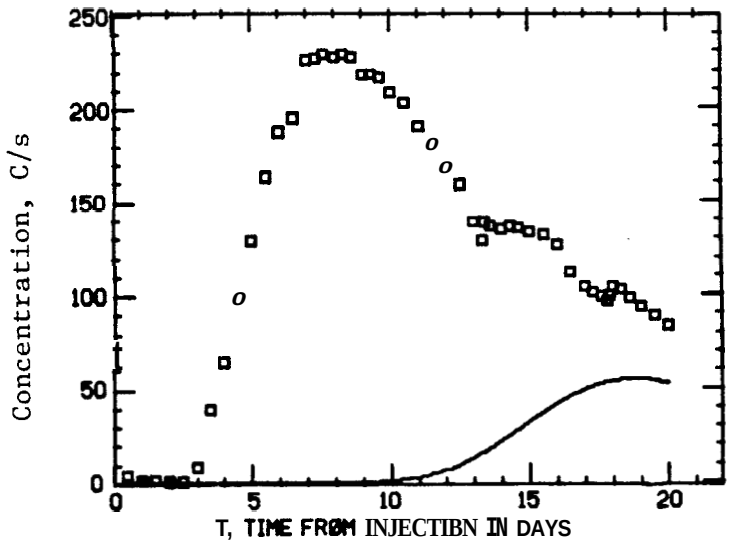
Calculated Values: ———

Observed Values: □



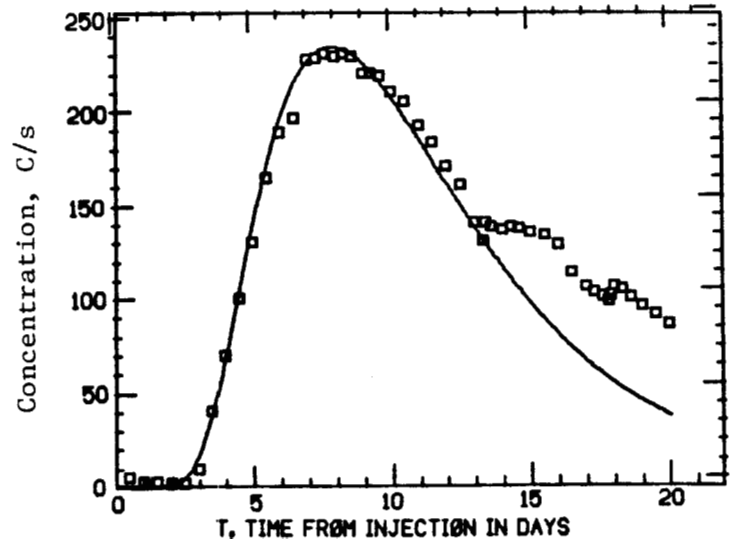
Fitted Profile

Fig. 7.8: WK 116



Component #1

Fig. 7.9: WK 116



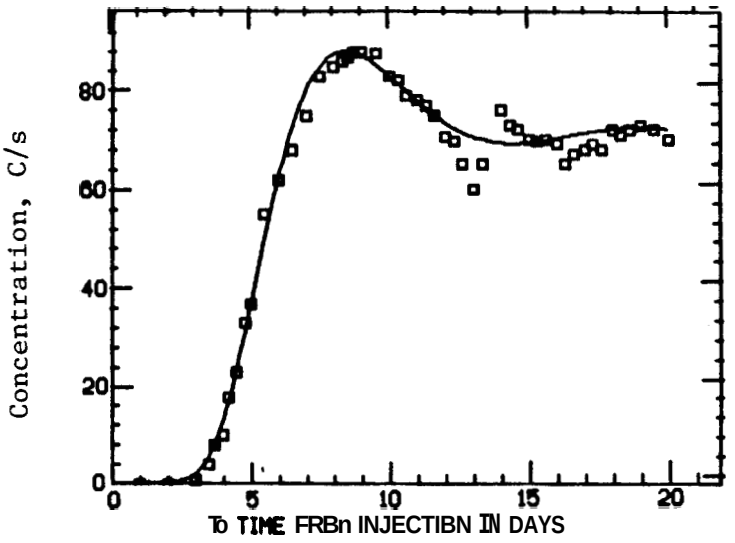
Component #2

Fig. 7.10: WK 116

For **All** Figures:

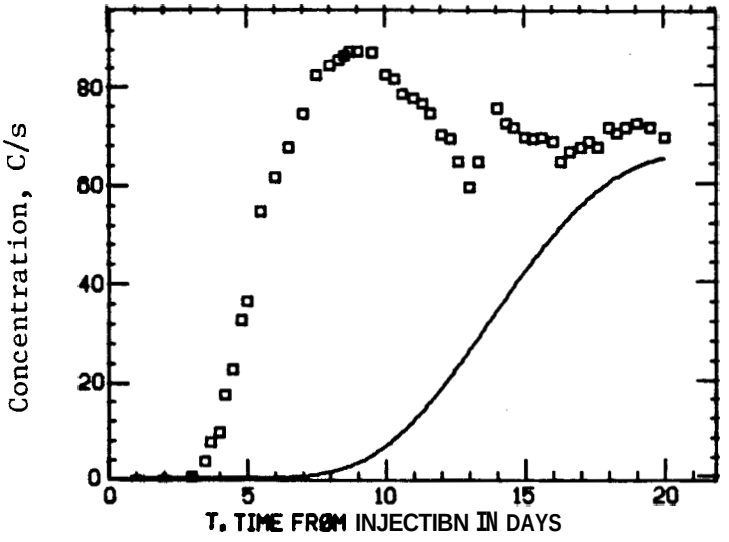
Calculated Values: ———

Observed Values: □



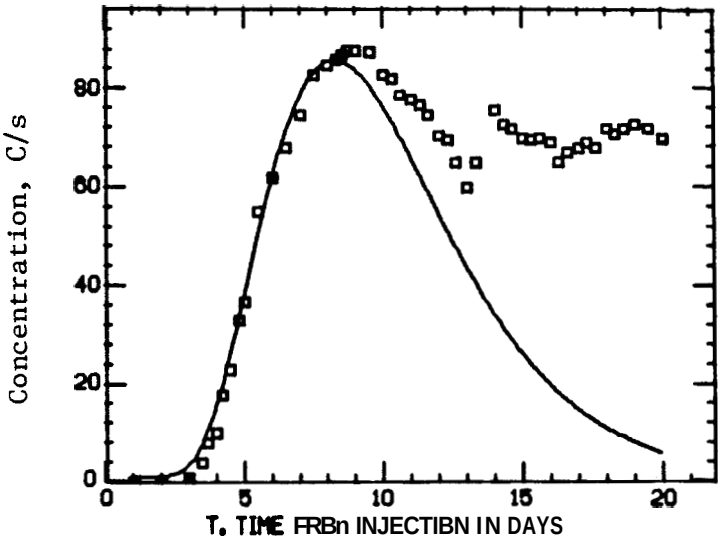
Fitted Profile

Fig. 7.11: WK 76



Component #1

Fig. 7.12: WK 76



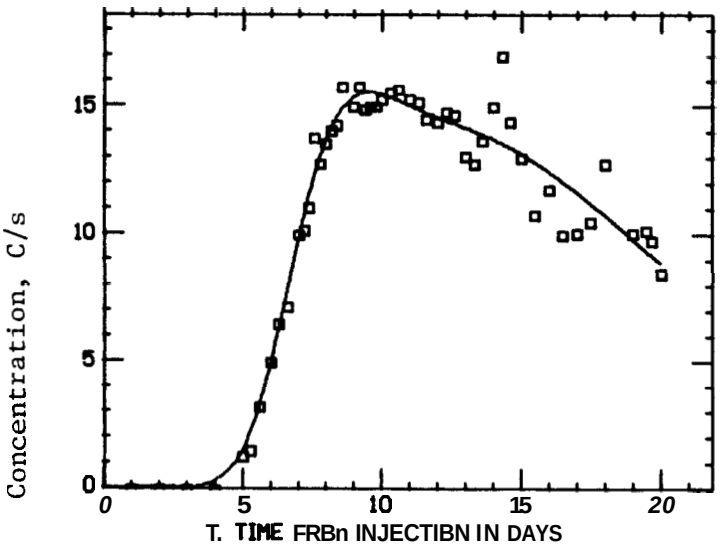
Component #2

Fig. 7.13: WK 76

For All Figures:

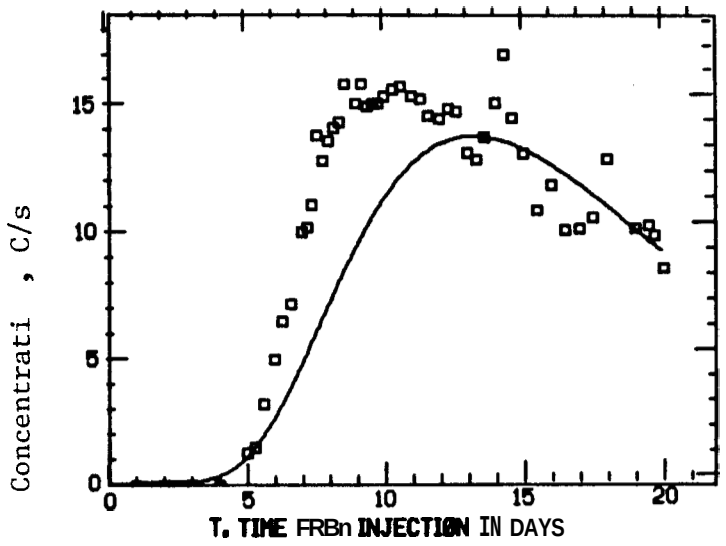
Calculated Values: ——

Observed Values: ▢



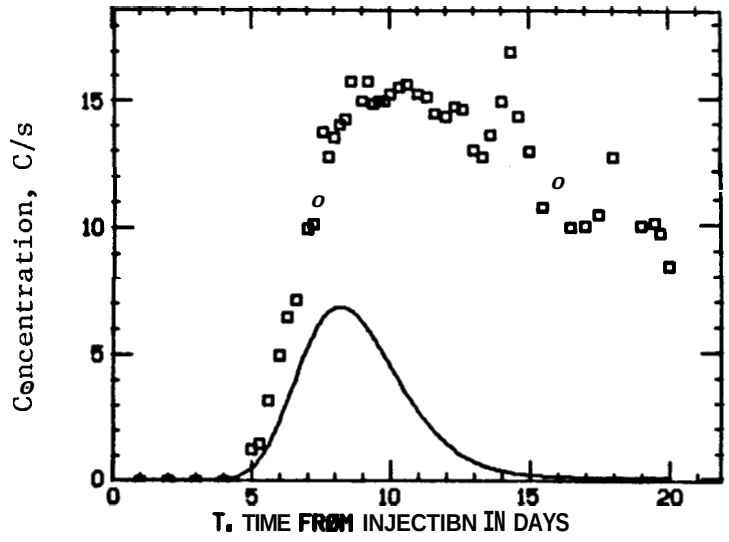
Fitted Profile

Fig. *7.14:* WK 108



Component #1

Fig. 7.15: WK 108



Component *ii2*

Fig. 7.16: WK 108

Table 7.7 lists the values €or the two-component case.

TABLE 7.7

Well	Flow Fraction	b (Fracture Aperature, mm)	$\bar{u}$ (Average Velocity, m/day)	L (Injector-Producer Length)
WK 116	.87	21	55.0	500
	.13	14	36.0	
WK 76	.42	25	6.5	145
	.58	18	16.3	
WK 108	.80	20	27.4	230
	.20	12	15.6	

Data from a field test in Onuma, Japan, were also analyzed for comparison. The tracer profile is shown in Fig. 7.17 and the well locations in Fig. 7.18.[7]

The poor match obtained from the Japanese data is a result of continuous reinjection of the tracer. Instead of the concentration approaching zero as it did at Wairakei, it approaches 0.3. This indicates that the tracer becomes uniformly mixed over time. A more accurate curve fit could be generated by adding an exponential background function.

From Table 7.8, it is evident that the values of b and $\bar{u}$ are not significantly different from those at Wairakei. More importantly, there has already been reinjection of waste-water at Onuma. The 0-3Ra well has had a greater enthalpy loss than other production wells in the field and the steam production rate has decreased. Therefore, one could infer

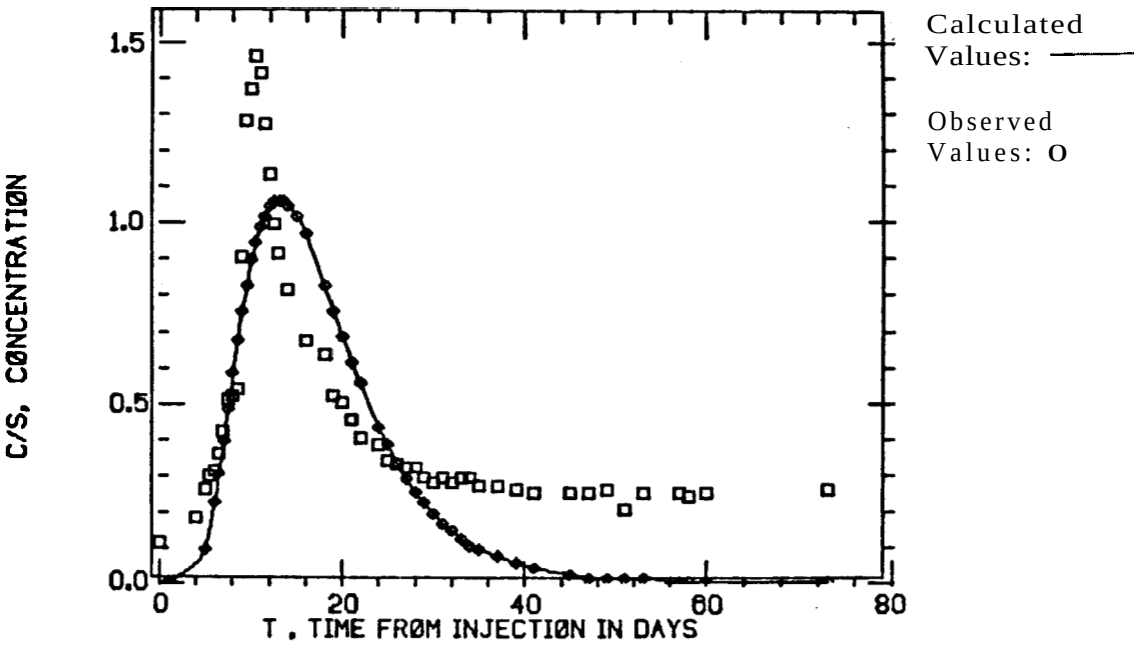


Fig. 7.17

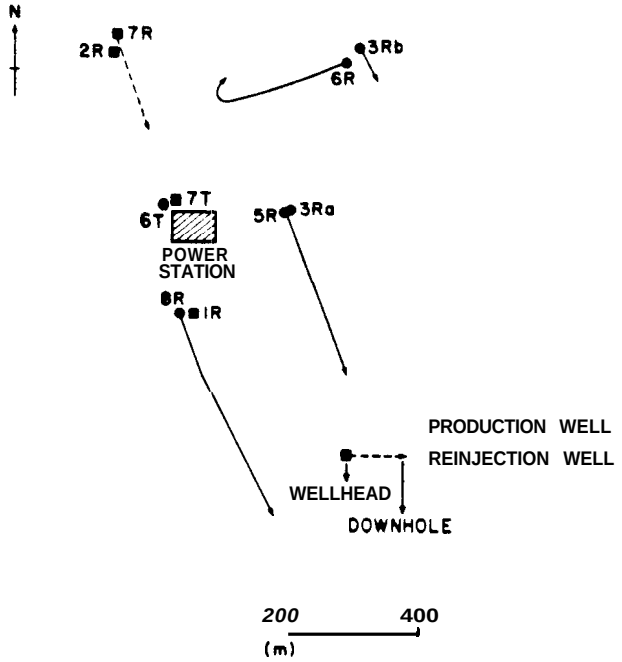


Fig. 7.18: SKETCH *MAP* OF WELL LOCATIONS AT ONUMA

TABLE 7.8

Well	Flow Fraction	b (cm)	$\bar{u}$ (m/day)	L
3Ra	1.0	8.2	24.3	350

from the similar tracer results and calculated fracture characteristics, that poorly placed reinjection wells at Wairakei might cause thermal drawdown.

Combining the above analysis with available information on Wairakei's geology, a preliminary description of the fracture system in the area around well KW 80 can be formulated.

1. Flow in the direction of WK 116 is primarily along the Kaiapo Fault and is dominated by extensive channeling. Simple single fracture connections without mixing are possible, but only if the fracture aperature is 1 cm or more. The flow does, however, appear to be within the fault.
2. Transverse faulting in the direction of WK 76 provides two major pathways. One is much longer than the other, and both have major flow along the fault.
3. There are two minor flow paths to WK 108. Channeled flow along the transverse fault and perhaps smaller fractures with little multi-channel mixing is indicated.

In light of the above results, there is evidence that most of the flow occurs in fractures along the faults, but is probably not one-dimensional. Therefore, a multi-dimensional flow configuration would probably result in an improvement of the inferred results.

Chapter VIII

CONCLUSIONS

1. A double flowpath model gives a more accurate data match than a single component model in most instances. However, larger numbers of flow paths do not appear to be necessary.

2. Effective dispersion in both one- and two-component models is larger than that predicted for one-dimensional flow in fractures. This is perhaps because flow is dominanted by multi-channel mixing within the faults.

3. Flow in the Wairakei field is not in single, simple fractures. If if were, average Peclet numbers would be larger and average residence times lower. It is 'clear, therefore, that some fracture branching is enoucntered during the flow.

4. Analysis of a tracer profile's components yields a preliminary idea of what type of inter-well connection exists.

5. This model indicates the type of flow paths that exist. It can say little about their configuration.

NOMENCLATURE

N = number of observations

C_i = measured exit tracer concentration

R = objective function

$\bar{C}$ = transfer function (represents produced tracer concentration)

u = velocity

$\bar{u}$ = average velocity

t = time

L = length between producer and injector

Pe = Peclet number

η = effective longitudinal dispersion coefficient

M = number of component fractures

E_j = fraction of flow in path j

E = linear scaling factor

y = variable width of the fracture

b = 1/2 of the fracture width

C_0 = initial concentration

D = molecular diffusivity constant

y_d = dimensionless width

b_d = dimensionless time

p = pressure

μ = viscosity

x_d = dimensionless distance

$\overline{C}_d$ = average concentration across the fracture thickness

h = height of the fracture

q = volumetric flowrate

q_d = dimensionless volumetric flowrate

s = activity of radioactive tracer

Chapter IX

REFERENCES

1. Brigham, W. E., and Smith, D. H.: "Prediction of Tracer Behavior in Five-Spot Flow," SPE 1130, presented at the SPE of AIME 40th Annual Fall Meeting, Denver, Colorado, Oct. 1965.

2. Baldwin, D. E., Jr.: "Prediction of Tracer Performance in a Five-Spot Pattern," J. Petroleum Technology, Vol 18 (April 1966), 513-517.

3. Yuen, D. C., Brigham, W. E., and Cinco-L., H.: "Analysis of Five Spot Tracer Tests to Determine Reservoir Layers," U.S. Dept. of Energy Report SAN-1265, Feb. 1979.

4. Abbaszadeh-Dehghani, M., and Brigham, W. E.: "Analysis of Well-to-Well Tracer Flow to Determine Reservoir Heterogeneity," SPE 10760, presented at the 1982 California Regional Meeting of the SPE, San Francisco, CA, March 24-26.

5. Wagner, O. R.: "The use of Tracers in Diagnosing Interwell Reservoir Heterogeneities--Field Results," SPE 6046, presented at the meeting of the SPE of AIME, New Orleans, Lousiana, 1976.

6. Tester, J. N., Bivins, R. L., and Potter, R. M.: "Interwell Tracer Analysis of a Hydraulically Fractured Granite Geothermal Reservoir," SPE 8270, presented at the SPE of AIME 54th Annual Fall Meeting, Las Vegas, NV, Sept. 1979.

7. Horne, R. N.: "Geothermal Reinjection Experience in Japan," J. Petroleum Technology, Vol. 34 (1982), 495-503.

8. Horne, R. N.: "Tracer Analysis of Fractured Geothermal Systems," Geothermal Resources Council, Transactions, Vol. 5 (1981), 291-294.

9. Horne, R. N., and Rodriguez, F.: "Dispersion in Tracer Flow in Fractured Geothermal Systems," Proceedings, 7th Annual Stanford University Geothermal Workshop (1981).

10. Golub, G., and Pereya, V.: "The Differentiation of Pseudo-Inverses and Nonlinear Least-Squares Problems Whose Variables Separate," Technical Report STAN-CS-72-261, Stanford University, Stanford, CA, 1972.

11. Osborne, M. R.: "Some Aspects of Non-Linear Least-Squares Calculations," in Lootsma, ed., Numerical Methods for Non-Linear Optimization, Academic Press, London (1972).

12. McCabe, W. J., Manning, M. R., and Barry, B. J.: "Tracer Tests--Wairakei," Institute of Nuclear Sciences Report INS-R-275, Dept. of Scientific and Industrial Research, Lower Hutt, New Zealand, July 1980. Geothermal Circular WJMcC 2.

13. Levenspiel, O., and Smith, W. K.: "Notes on the Diffusion-Type Model for the Longitudinal Mixing of Fluids in Flow," Chemical Engineering Science, Vol. 6 (1957), 227-233.

14. Vetter, O. J., and Zinnow, K. P.: "Evaluation of Well-to-Well Tracers for Geothermal Reservoirs," Lawrence Berkeley Laboratory Report #LBL-11500, August 1981.

15. Rodriguez, F.: "Dispersion of a Tracer Solution in a Single Fracture," draft paper, 1981.

16. Taylor, G. I.: "Dispersion of Soluble Matter in Solvent Flowing Slowly Through a Tube," Proceedings, Royal Society, Vol. 219 (1953), 186-203.

17. DiPippo, R.: Geothermal Energy as a Source of Electricity, U.S. DOE, Washington, D.C. (1980).

18. Ito, J., Kubota, Y., and Kurosawa, M.: "Tracer Tests of the Geothermal Hot Water at Onuma Geothermal Field," Japan Geothermal Association Journal, Vol. 15 (1978), 87-97 (in Japanese).