Stanford Geothermal Program
Interdisciplinary Research in
Engineering and Earth Sciences
STANFORD UNIVERSITY

MATRIX DIFFUSION AND ITS EFFECT ON THE MODELING OF TRACER RETURNS FROM THE FRACTURED GEOTHERMAL RESERVOIR AT WAIRAKEI, NEW ZEALAND

By

Clair Lynn Jensen

A Report Submitted to the Department of Petroleum
Engineering of Stanford University in Partial
Fulfillment of the Requirements for the
Degree of Master of Science

December 1983

ABSTRACT

Tracer tests performed at. the geothermal reservoir at Wairakei, New Zealand have been analyzed. A mathematical and physical description which models tracer flow through individual fractures with diffusion into the surrounding porous matrix has been used. Observed tracer return profiles matched significantly well with the model calculations. From the model, first tracer arrival times and the number of individual fractures (the principal conduits of fluid flow in the reservoir) joining the injector-producer wells can be determined. If the porosity, adsorption distribution coefficient, bulk density and effective diffusion coefficent are known, fracture widths may be calculated. Hydrodynamic dispersion down the length of the fracture is a physical component not taken into account in this model. Future studies may be warranted in order to determine the necessity of including this factor. In addition to the tracer profile matching by the matrix diffusion model, comparisons with a simpler fracture flow model by Fossum and Horne (1982) were made. The inclusion of the matrix diffusion effects was seen to significantly improve the fit to the observed data.

Approved for the Department:

Roland N. Horne (advisor)

ACKNOWLEDGEMENT

I would like to thank my parents for their investment of time, money and energy in the development of my life and environment, so that I could be where I am today. Heartfelt thanks is given to Camille Bouza for her patience, understanding and encouragements. Thanks and appreciation are also due my advisor, Dr. Roland N. Horne, for his wisdom and help in the pursuit and development of this research report. The United States Department of Energy provided funds for my schooling at Stanford University and computing associated with this report through contract number DE-AT03-80SF11459 to the Stanford Geothermal Program. The data used in this work was made available by the Institute of Nuclear Sciences, Department of Industrial and Scientific Research, Gracefield, New Zealand. Assistance in the interpretation and conceptual understanding of the tracer model analysis was also provided by Gardner Walkup and John Marcou.

TABLE OF CONTENTS

		Page
Section 1	Introduction	1
Section 2	Literature Review	3
Section 3	Formulation	8
	A Geology	8
	B Tracer Test Data	12
	C Mathematical and Physical Model	13
	D Computer Program	25
Section 4	Results	29
Section 5	Conclusions and Recommendations	34
	Nomenclature	36
	References	38

Appendix A Tracer Return Profiles

Appendix B Computer Program and Sample Output

Appendix C Fitted Tracer Return Profiles

Section 1. INTRODUCTION

In many geothermal development schemes, produced geothermal waters are reinjected for the purpose of disposal and pressure maintenance. The known effects of reinjecting water are: improved or degraded thermal recovery (depending on underground flow paths and velocities); permeability changes in the reservoir; pressure maintenance of reservoir fluid; and possible re-routing of natural underground water pathways. Horne (1982) presents a summary of such experience on a worldwide basis.

Since both detrimental and beneficial effects have been observed, reservoir tests to determine the effects of a proposed reinjection system are desireable. Also, various reservoir parameters and the mechanics of fluid flow in the reservoir need be investigated. Interwell tracer tests have made significant contributions to the understanding of fluid flow in natural underground reservoirs. Radioactive and chemical tracers have been used for many years in groundwater hydrology to study the movement of water through porous media, but until recently little has been reported on their use in geothermal systems.

In addition to the test itself, there needs to be some method to analyze the data obtained. To date, tracer returns from geothermal reservoirs have been analyzed in only a semi-quantitative sense to determine transit times, flow velocities and pathways.

In 1982, Fossum and Horne presented an analysis of tracer data from field results at Wairakei, New Zealand, including a model describing linear flow through a fracture with hydrodynamic dispersion. This physical and mathematical model unfortunately proved to be only partially adequate in its modeling of fluid flow, and does not fit well to many of the test results from the fractured Wairakei geothermal reservoir.

In searching for and testing of a physical model that would better mathematically fit the tracer return data, it has been found here that a 'double-porosity' model is more satisfactory. The 'double-porosity' model formulated in this work includes diffusion of tracer into the porous matrix in addition to flow through the fractures in the reservoir.

Section 2. LITERATURE REVIEW

Interwell tracer tests have been an important tool in the analysis of fluid flow in various rock matrix systems. Wagner (1977) listed information obtainable from tracer tests in the oil industry, for example: identification of poor injection wells; delineation of flow barriers; directional flow trends and volumetric sweep patterns; determination of relative velocities of injected fluids; and evaluation of sweep improvement treatments. The geothermal industry has used interwell tracer tests to help investigate possible damage from cooling due to reinjection of produced waters and to ascertain flow patterns in the reservoir. For example, in New Zealand, as described by McCabe, Barry and Manning (1983), radioactive tracers have been used to determine flow velocities and general flow directions. Nakamura (1981) describes tracer tests performed in Japan which clearly showed short-circuiting and a decline in production due to cold water reinjection. The nuclear industry and governmental agencies make use of tracer tests to help search for possible waste disposal sites and to study the characteristics of such sites. Two important papers that address the implications of tracer tests to underground storage and disposal of nuclear waste are those by Webster, Proctor snd Marine (1970), and Lester, Jansen, and Burkholder (1975). As noted by Grisak and Pickens (1980), tracer tests in fractured low matrix porosity rock have been performed in several hydrogeologic contexts, such as groundwater age dating, contaminant transport, and groundwater flow velocity or dispersion characteristics of a geologic medium.

Several mathematical models for describing fluid and tracer flow through porous media have been presented. Intensive studies in this area began with the study of chromatographic and ion exchange separation processes. Lapidus and Amundson (1952) presented the mathematics of equilibrium and nonequilibrium adsorption, and longitudinal diffusion under various boundary conditions in solid material packed columns. Gershon and Nir (1969) presented the effects of initial and boundary conditions on the distribution of the tracer in time and distance for several one-dimensional systems (infinite, semi-infinite and finite) of tagged liquid flowing through a solid matrix. The effects of hydrodynamic dispersion, diffusion, radioactive decay, and simple chemical interactions of the tracer were included. Field experiments using fluorescent dye and radioactive tracers were employed by Tester, Bivens, and Potter (1982) to characterize a hot, low-matrix permeability, hydraulically-fractured granitic reservoir. Tracer profiles and residence time distributions were used to delineate changes in the fracture system, diagnosing flow patterns, and in identifying new injection and production zones. One- and two-dimensional theoretical dispersion models utilizing single and multiple porous, fractured zones with velocity and formation dependent effects are presented and discussed with repect to field data.

Until recently, most mathematical models were based upon a porous media physical model. These porous media type models are useful, but since most geothermal reservoirs are highly fractured they are not entirely applicable, for they assume some type of uniform sweep through the reservoir. Horne and Rodriguez (1983) presented a mathematical model based on the physics of dispersion during fluid flow through fractures, thus forming a basis for the derivation of a transfer function to be used in the interpretation of field observations. Fossum and Horne (1982) utilized this model to analyze tracer return profiles for the Wairakei geothermal field. A double flowpath model was found to give a more accurate data match than a single component model, though interwell flow over long distances was interpreted to occur in only a very few open fissures. However, other tracer test data more recently obtained from Wairakei has proven to be poorly fitted by this fracture flow model.

A possible explanation for this poor fit was indicated by laboratory studies performed by Breitenbach (1982). Significant retention of the tracer in reservoir rocks was observed. The processes producing tracer retention could include adsorption, diffusion, dissolution and ion exchange.

Many current madels describing tracer migration in the ground are based on the assumption that the tracer is retarded by some sorbtion mechanism. Since the sorbtion mechanisms are not well understood, assumptions such as reversibility and instantaneous equilibration are normally made. To calculate tracer migration in bedrock, either of two tranport mechanisms may be used. For porous bedrock, where the water is assumed to flow evenly through all the pores, the bulk of the rock is equilibrated with the tracer-containing water. This is called bulk reaction. For sparsely fissured bedrock the assumption is that the flow is in the macrofissures and the tracer only reacts with the fissure surface. The fluid does not penetrate into the rock matrix to any appreciable depth. This is called surface reaction. Using these two models, Neretnieks (1980) attempted to reproduce experimental data, without success. From his experimentation, Neretnieks (1980) determined that diffusion into the rock matrix can enhance the retardation by many orders of magnitude compared to retardation by surface reaction in fissures only, and that the magnitude of the retardation depends very much on the fissure widths and spacings.

Grisak and Pickens (1980) presented a study concerning the effect of matrix diffusion on solute transport through fractured media. Transport is considered in a manner conceptually similar to 'double-porosity' or 'intra-aggregate' transport models. A finite element model was developed to simulate nonreactive and reactive solute transport by advection, mechanical dispersion, and diffusion in a unidirectional flow field. The numerical model and the laboratory tracer test data provided insight into the processes controlling solute transport in fractured media.

From studies of the migration of radionuclides in the bedrock surrounding nuclear waste repositories, Neretnieks, Eriksen, and Tahtinen (1982) developed a mathematical and physical model describing tracer movement in a single fissure of granitic rock. This model takes into account instantaneous sorption on the surface of the fissure, and loss of tracer from the fluid flowing in the fissure due to diffusion into the porous matrix. It is this model that is used to help gain insight and a physical understanding of the fluid flow implied by the tracer tests performed at the geothermal reservoir field at Wairakei, New Zealand.

Section 3. FORMULATION

In this section, the tracer test data obtained from Wairakei, New Zealand is discussed. A mathematical and physical model is presented along with the computer program which uses the model to analyze the data. The geology of the Wairakei geothermal field is also briefly presented to give a physical understanding of the hydrothermal aquifer surroundings and to help in the analysis of the results computed by the model.

A. Geology

Wairakei, the site of New Zealand's first geothermal power station, is one of the larger hydrothermal areas in the active volcanic belt extending from the National Park Volcanoes south of Lake Taupo, in a north-easterly direction to White Island in the the Bay of Plenty. DiPippo (1980) cites the Wairakei field as the longest operated liquid dominated geothermal reservoir in the world. Figure 3.1 from DiPippo shows the location, and Figure 3.2 from Grindley (1965) shows the generalized geology and tectonits of Wairakei and the Central

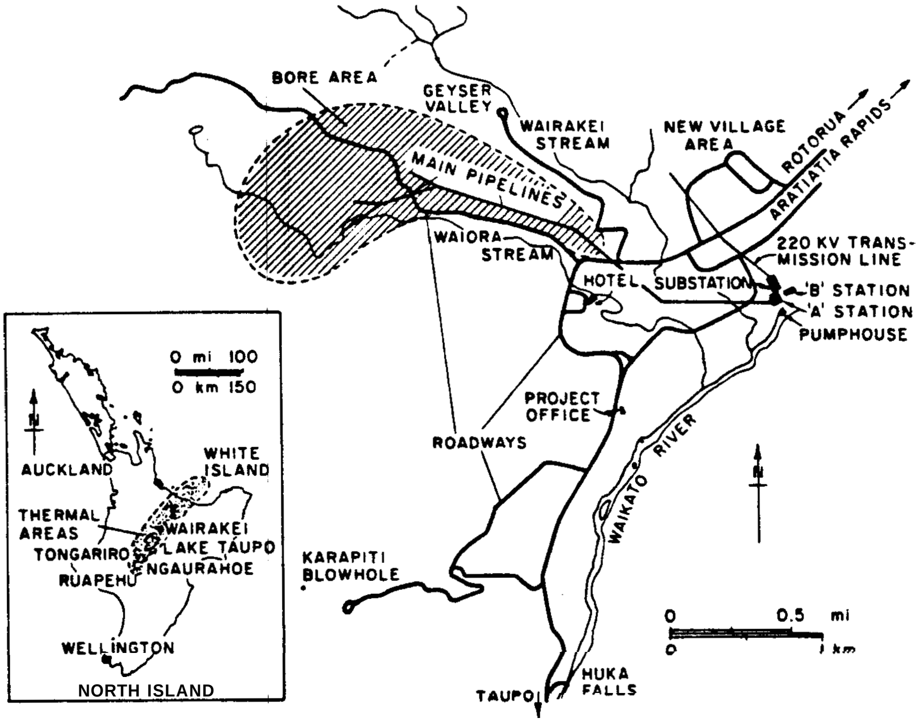


FIGURE 3.1 Location of Wairakei geothermal field, North Island, New Zealand.

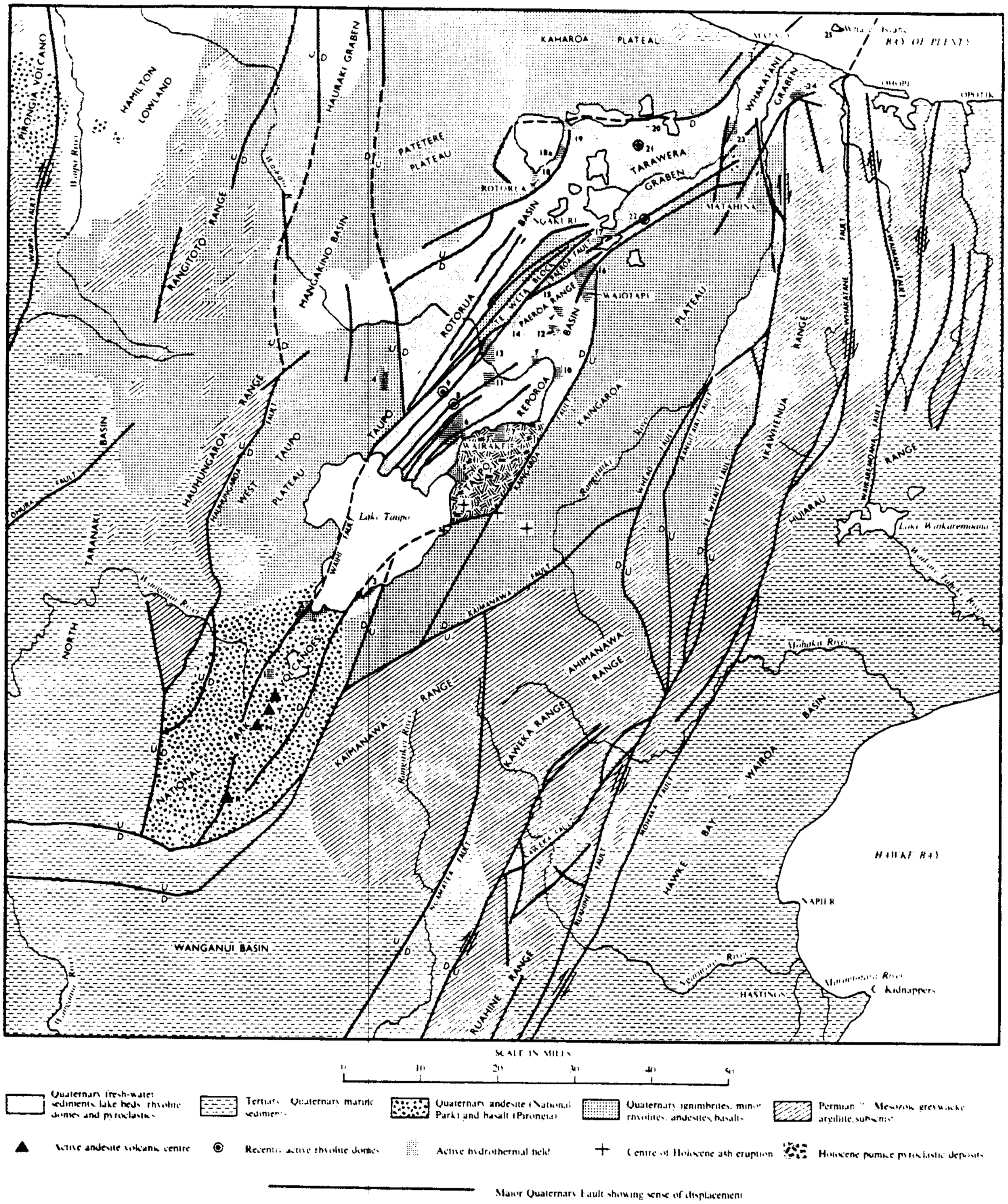


FIGURE 3.2 Generalized geological and tectonic map of Central Volcanic Region, New Zealand, showing Holocene volcanism, hydrothermal fields, and important faults.

Volcanic Region.

The Wairakei hydrothermal area occupies a surface area of about ten square miles. The area is located on the left bank of the Waikato River extending west for about three miles from the river. The Wairakei geothermal field includes several separated centers of thermal activity, notably Karapiti, halfway between Taupo and Wairakei, the Waiora Valley to the west of Wairakei, and Geyser Valley along Wairakei Stream to the northwest of Wairakei. These separated centers merge at depth into one connected area.

The Wairakei Block is an elliptical structure trending north-northeast. Gravity and magnetic suveys have shown evidence of basement uplift in an area less than two square miles. The uplift appears to be in the form of two domes separated by a narrow downfaulted zone. The maximum concentration of deep-seated hydrothermal activity is closely connected with these structures and especially with the faults.

Faults are common throughout the Wairakei area and many show as small surface scarplets or lineaments. The faults are important in the drilling of high-pressure production steam wells where the intervening country rock is relatively impermeable due to cementation by hydrothermal minerals. Drilling of successful wells in this type of country commonly depends on intersection of a fault at depth by the drillhole giving the necessary increased permeability. All faults encountered in the field whether subsurface or surface, appear to be dominantly normal faults and are downthrown in the direction of dip of the fault planes. See Figure 3.3 for an areal view showing the Wairakei production area in relation to the major faults. The dip of the Waiora Fault, assuming that the subsurface fissures mark the intersection of the fault plane by drillholes varies from 85° to 88°. A prominent fault trace, the Upper Waiora Fault, extends across the upper Waiora Valley in a northeast direction. Temperatures recorded (214°F) and the high rank hydrothermal alteration are sufficiently encouraging at a depth of 650 feet to suggest that this fault may well be another important feed for hydrothermal water into the Waiora aquifer. The Kaiapo Fault trace can be seen by surface expressions to the southwest and appears to link with the Kaiapo Fault scarp, and may be its northeast continuation. The Kaiapo Fault changes downthrow to the northwest, this change taking place across a prominent northwest cross fault. To the southwest, the fault: is again downthrown to the northwest as far as the Kaiapo Scarp. The Wairakei Fault is probably the subsurface extension of a fault trace two hundred feet to the northwest of hole WK24. The Wairakei Fault can be traced, by surface exposures, southwestwards

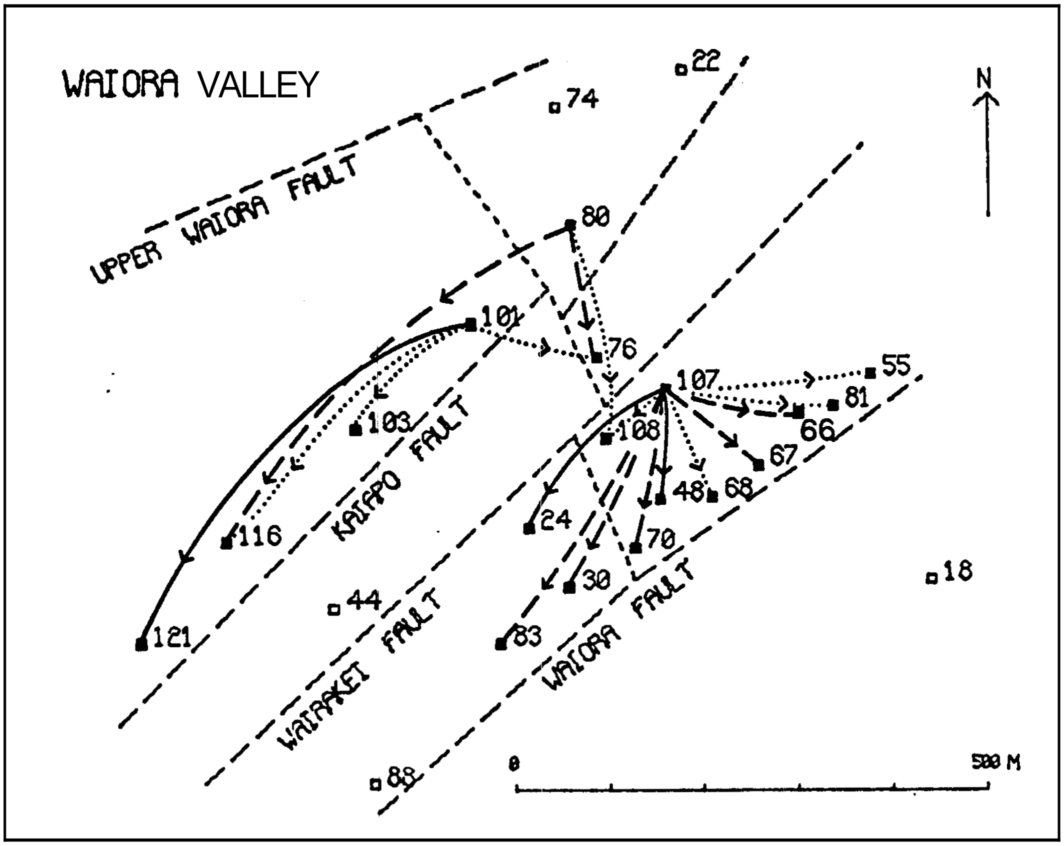


FIGURE **3.3** Map showing Wairakei Production area, and relation to major faults.

as far as the Kaiapo Scarp. Surface displacements are down to the southeast, which is in accord with the direction of dip (86") inferred from hole WK24. The Wairakei and Waiora Faults intercept each other at about 3600 feet below the surface.

All of the faults so far described strike northeast parallel to the trend of the Taupo Volcanic Zone. These northeast faults are locally crossed almost at right angles by north-northwest faults, which appear to have some bearing on the location of the thermal activity. A most important north-northwest fault crosses the Waiora, Wairakei, Kaiapo, and Upper Waiora Faults in the western part of the Wairakei production area. This fault has had a long history of movement and may be indirectly controlling the present heat flow in the western part of the production area.

The reservoir itself is contained within the Waiora Formation (containing pyroclastic rocks, ignimbrites and interbedded sediments), and the Wairakei Ignimbrite formation (containing pulverulitic ignimbrite). The reservoir is overlain by the Huka Falls Formation (containing impermeable lake-deposited grey mudstone interbedded with various pyroclastic rocks) and the Wairakei Breccia (Lapilli Tuff) formation. Figures 3.4 and 3.5 show these formations pictorially.

For more information as to Wairakei stratigraphy,

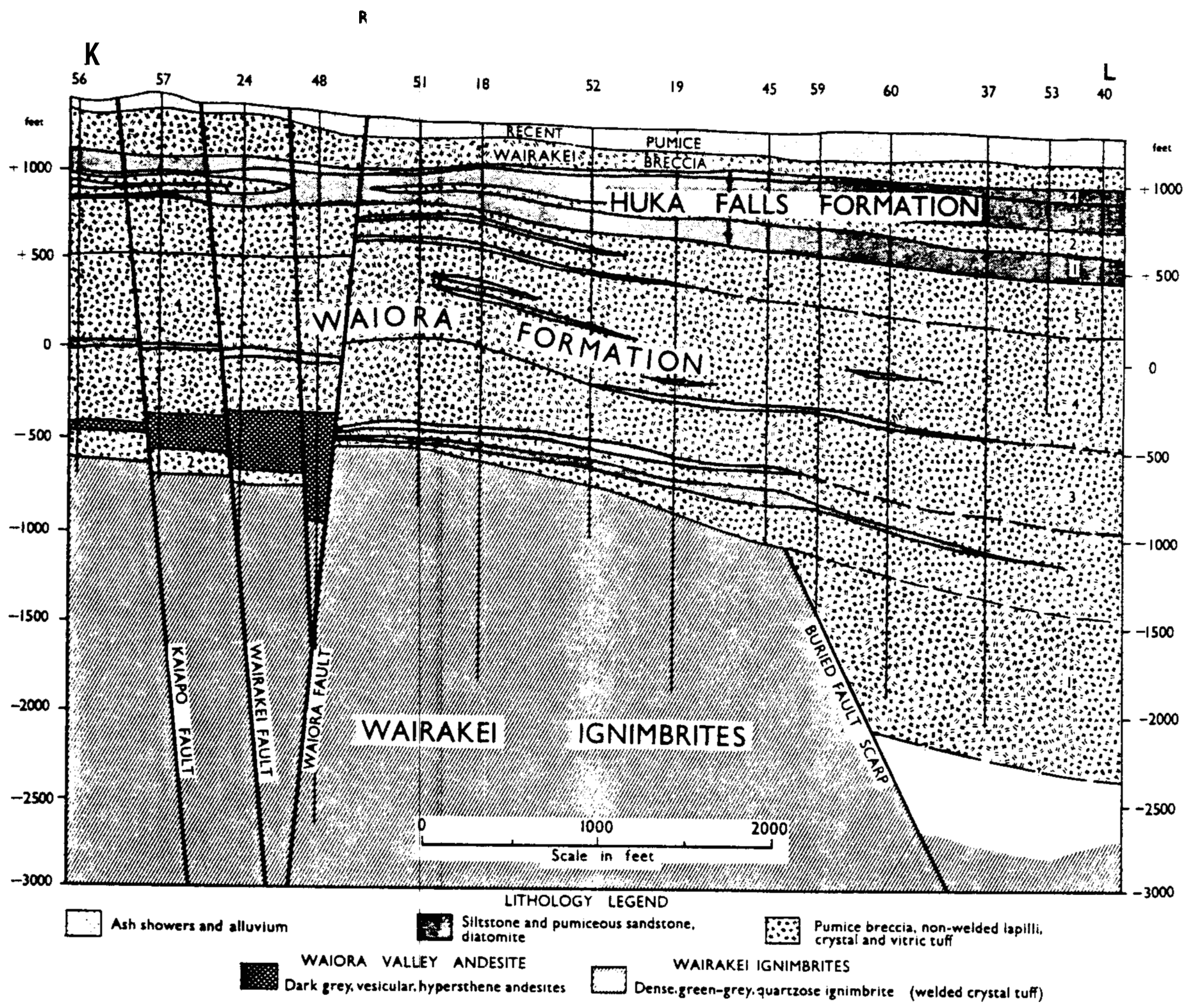


FIGURE **3.4** Wairakei Hydrothermal Field. Cross section from hole WK56 to WK40 showing geological structure.

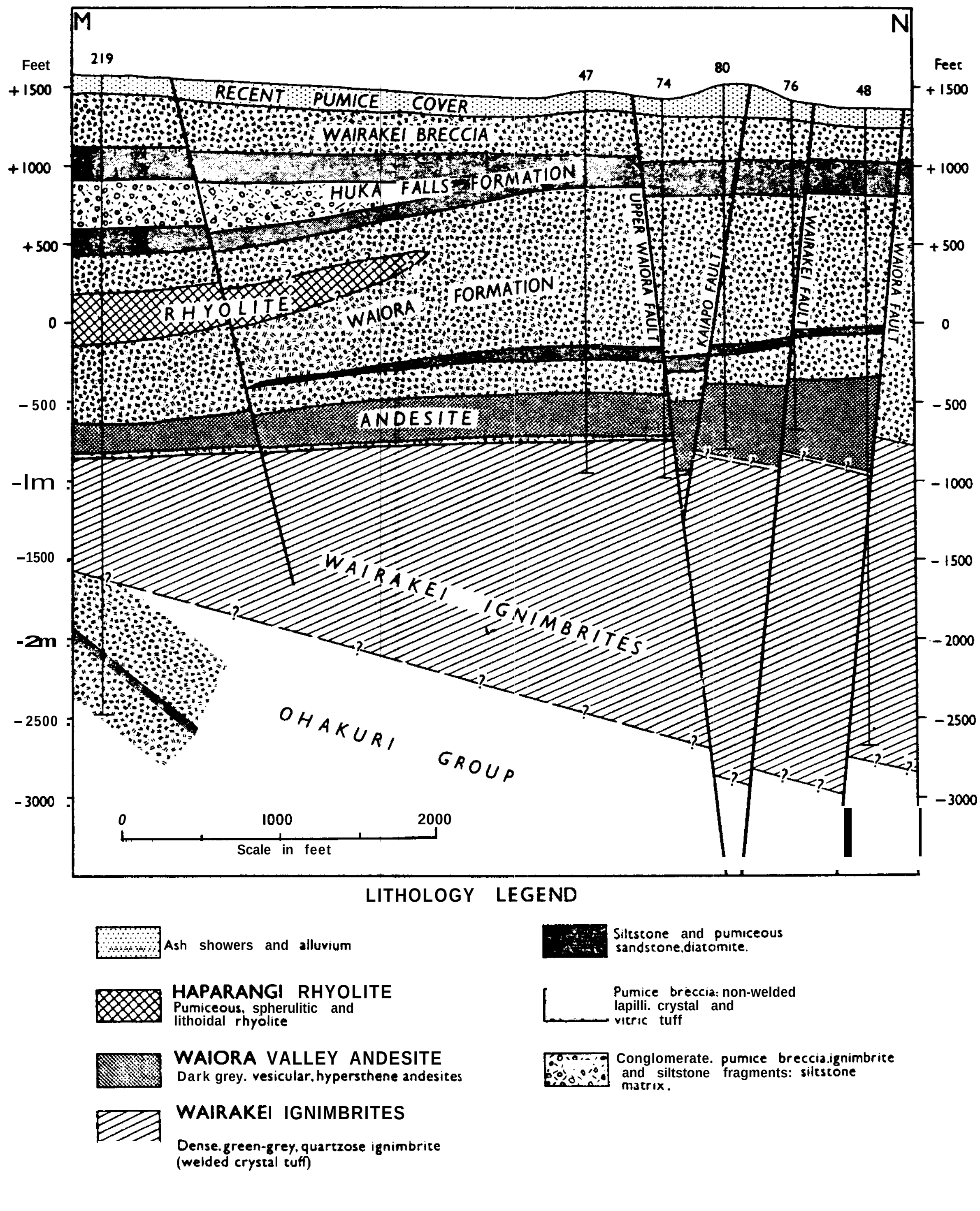


FIGURE 3.5 Wairakei Hydrothermal Field. Cross section from hole WK219 to WK48 showing geological structure.

structure and exploitation, see Grindley (1965).

B. Tracer Test Data

The tracer tests which produced the data used in this study were performed by the Institute of Nuclear Sciences, Department of Scientific and Industrial Research, New Zealand. Iodine-131 was used as the tracer. Its half-life is eight days. This eight-day half-life limited the field tests to four to five weeks, by which time a combination of decay corrections and variation of backgraund signals produced unacceptably large errors. This error becomes quite noticeable at late time for some of the tracer return data. For a detailed description of tracer injection methods, well monitoring and counting equipment used at Wairakei see McCabe, Barry and Manning (1983).

Two tracer injection tests were made that are analyzed in this report. One iodine-131 injection was made into well WK107 in March of 1979, and another into well WK101 in June of 1979. The quantities and depths of injection were 155 GBq at 334 meters in well WK107, and 165 GBq at 400 meters in well WK101. Responses from wells WK24, 30, 48, 55, 67, 68, 70, 81, 83, and 108 were monitored after the injection into WK107. Also, responses from wells WK18, 22, 24, 44, 48, 55, 74, 76, 88, 103, 116, and 121 were monitored after the injection into WK101. Not all of the monitored wells gave sufficient tracer returns and therefore are not analyzed in this report.

Plots of the data showing concentration versus time are shown in Appendix A. The data has been corrected for decay and background responses. Also, all negative values have been deleted, and straight lines drawn between points and through any missing data points. Missing data is due to instrument or field problems. Concentrations are scaled to units of injected amount divided by 10^{12} liters.

C. Mathematical and Physical Model

Most studies of groundwater flow within fractured media emphasize the dominating influence of fractures on the effective permeability of the rock mass. One-dimensional flow within a single fracture can be generally described by solution of the Navier-Stokes equation for nonturbulent flow of a viscous incompressible fluid between two parallel plates, neglecting inertial forces. Derivation of such a solution for one-dimensional laminar flow in a single fracture has been presented by Horne and Rodriguez (1983).

This single fracture model derived by Horne and Rodriguez (1983) was modified by Neretniek, Eriksen, and Tahtinen (1982) to include instantaneous linear equilibrium reaction with the surfaces of the fracture. The partial differential equation modeling this modification is very similar to that used by Horne and Rodriguez (1983). The solution is the same, except that the nonlinear parameter defining the mean residence time of the tracer is altered by a constant factor representing the adsorption of tracer onto the fracture walls. Since the solution is the same, calculated tracer return curves using this refined model are of the same shape as those using the Horne and Rodriguez model. As is indicated in this report, these models have been found to be unsatisfactory in modeling single peak Wairakei tracer test data.

Although fractures are the principal paths of groundwater flow and solute transport, the matrix adjacent to the fractures plays an important part in the overall solute transport process. The process of solute diffusion from a fracture into the adjacent matrix has been studied and modeled by Grisak and Pickens (1980) and by Neretnieks (1980 and 1982). This process is illustrated in Figure 3.6, which schematically depicts a

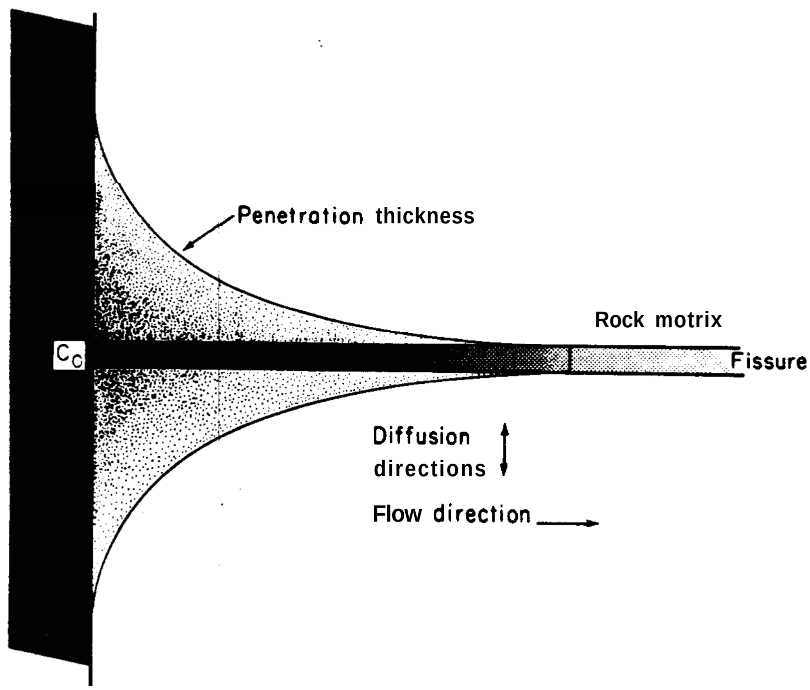


FIGURE 3.6 Fissure flow and sorption by diffusion into the rock matrix.

constant solute source of concentration C_o transported through a fracture. The effect of matrix diffusion is to provide solute storage, with the rate of change of storage within the matrix related to Fick's second law of diffusion. A one-dimensional form of the diffusion equation into the porous matrix is given by,

$$\frac{\partial}{\partial y}\left(\phi D_a \frac{\partial C_p}{\partial y}\right) = \phi \frac{\partial C_p}{\partial t} \qquad (3.1)$$

where the porosity ϕ and apparent diffusion coefficient D_a are assumed to be constant throughout the matrix contacted by the fluid, so that Eqn. 3.1 can be rewritten as,

$$D_a \frac{\partial^2 C_p}{\partial y^2} = \frac{\partial C_p}{\partial t} \qquad (3.2)$$

The net effect of matrix diffusion is to retard the arrival of the solute at any point along the fracture. If the source of the solute is discontinued, the effect will be to flush the fracture and reverse the concentration gradient, causing solute to move from the matrix into the fracture.

A general equation describing solute transport in a saturated medium can be written in two dimensions as:

$$\rho_b \frac{\partial S}{\partial t} + \frac{\partial(\phi C)}{\partial t} - \frac{\partial}{\partial x}\left(\phi D_{xx} \frac{\partial C}{\partial x} + \phi D_{xy} \frac{\partial C}{\partial y} - q_x C\right) -$$

$$\frac{\partial}{\partial y}\left(\phi D_{yx} \frac{\partial C}{\partial x} + \phi D_{yy} \frac{\partial C}{\partial y} - q_y C\right) = 0 \qquad (3.3)$$

where,

ρ_b = bulk density of the medium, M/L^3

S = amount of solute in the sorbed phase, M/M

x,y = Cartesian directions, L

D = hydrodynamic dispersion coefficients in the corresponding x,y-directions, L^2/T

q_x, q_y = Darcy velocities, L/T

This form of the equation includes the effects of adsorption by the medium, hydrodynamic dispersion, and advection. A linear equilibrium relationship between the dissolved and sorbed phases of the solute has been assumed and is written S=kC, where k is referred to as the adsorption distribution coefficient. Linear adsorption assumes that once the tracer and rock are brought sufficiently close together, adsorption will be an instantaneous process.

Simplifying Eqn 3.3 to mode1 a unidirectional flow field in a fractured porous medium gives,

$$(\phi + \rho_b k)\frac{\partial C}{\partial t} - \frac{\partial}{\partial x}\left(\phi D_{xx} \frac{\partial C}{\partial x}\right) + q_x \frac{\partial C}{\partial x} - \frac{\partial}{\partial y}\left(\phi D_{yy} \frac{\partial C}{\partial y}\right) = 0 \qquad (3.4)$$

where x is the direction of flow and y is normal to this direction. If it is assumed that the porosity ϕ, adsorption distribution coefficient k, bulk density ρ_b, hydrodynamic dispersion coefficients **D**, and the Darcy velocity q_x are constant in the region of interest, then Eqn. 3.4 becomes,

$$\left(1 + \frac{\rho_b k}{\phi}\right)\frac{\partial C}{\partial t} - D_{xx} \frac{\partial^2 C}{\partial x^2} + U_f \frac{\partial C}{\partial x} - D_{yy} \frac{\partial^2 C}{\partial y^2} = 0 \qquad (3.5)$$

This equation can be simplified further by neglecting hydrodynamic dispersion in the fracture so that the second term drops out. In its place, however, a term describing the **loss** of tracer from the fluid flowing in the fracture due to diffusion into the porous matrix of the wall is included. This new term is represented by,

$$\frac{2D_e}{\delta} \left. \frac{\partial C_p}{\partial y} \right|_{y=0}$$

Two different diffusion coefficients have been presented up to this point, D_a and D_e. The apparent and effective diffusion coefficients are related as follows:

$$D_a = \frac{D_e}{I \quad C} \quad \sim \quad \sim \quad \sim \tag{3.6}$$

The effective diffusion coefficient D_e is dependent on temperature, porosity, molecular diffusivity, and the geometry of the rock. $K_d \rho_b$ is a volumetric sorption equilibrium constant and is related to porosity ϕ, the solid rock density ρ_s and the adsorption distribution coefficient k by the equation,

$$K_d \rho_b = \phi + (1 - \phi) k \rho_s \tag{3.7}$$

Notice that if the solids are inert, i.e., k=0, the porous rock matrix still has a volumetric sorption equilibrium constant equal to its porosity ϕ. Rearrangement of Eqn. 3.7 gives,

$$\frac{K_d \rho_b}{\phi} = 1 + \frac{(1-\phi)}{\phi} k \rho_s \tag{3.8}$$

And since $\rho_s(1-\phi)=\rho_b$, Eqn 3.8 becomes

$$R = \frac{K_d \rho_b}{\phi} = 1 + \frac{k \rho_b}{\phi} \tag{3.9}$$

where R is refered to as the retardation factor. Using this above relation further simplifies Eqn 3.5.

The retardation factor defines the mean velocity of the moving liquid relative to the mean velocity at which the tracer itself moves through the rock. This factor accounts for the slowing down of a tracer moving with the fluid due to the interaction with the solid. If there is no interaction between the tracer and the solid phase, k becomes zero and R reduces to one.

The last term in Eqn. 3.5 describes a diffusive flux into or out of the matrix adjacent to the fracture. This term is also represented by Eqn. 3.2 which can be decoupled to form two equations describing the physical situation of one-dimensional advective flow through a fracture with simultaneous tracer adsorption and diffusion into the surrounding porous matrix. The two equations describing this condition are as follows:

$$R\frac{\partial C_f}{\partial t} - \frac{2D_e}{\delta}\frac{\partial C_p}{\partial y}\bigg|_{y=0} + U_f\frac{\partial C_f}{\partial x} = 0 \qquad (3.10)$$

$$D_a\frac{\partial^2 C_p}{\partial y^2} = \frac{\partial C_p}{\partial t} \qquad \textbf{(3.11)}$$

where,

C_f = concentration of tracer in the liquid in the fracture

C_p = concentration of tracer in the liquid in the porous matrix

D_a = apparent diffusion coefficient, L^2/T

D_e = effective diffusion coefficient, equal to $D_a \rho_b K_d$, L^2/T

δ = fracture width, L

U_f = fluid velocity in the fracture, equal to x_o/t_w, L/T

t_w = residence time of water, T

x_o = pathlength of the fracture from injection well to production well, L

The initial and boundary conditions are a finite rectangular pulse of tracer with duration Δt introduced at the inlet of the fracture at time $t=0$, and the fracture and rock are originally free of tracer. These conditions can be expressed as follows:

Initial conditions,

$$C_p = C_f = 0 \qquad t<0 \quad \text{for all } x \text{ and } y$$

Boundary conditions,

$$C_p = 0 \qquad t>0 \quad \text{as} \quad y \rightarrow \infty$$

$C_f = C_o$ = initial tracer concentration in the fluid at $x=0$ during finite input of tracer of duration Δt.

The solution to Eqns. 3.10 and 3.11 subject to the given initial and boundary conditions is, according to Carslaw and Jaeger (1959, p.396),

$$C_f = 0 \quad \text{for} \quad t<t_w R$$

and,

$$C_f/C_o = f(t + \Delta t) - f(t) \qquad \text{for} \quad t > t_w R$$

where,

$$f(t) = \text{erfc} \frac{D_e t_w}{\delta \left(D_a (t - t_w R) \right)^{0.5}}$$

t_w = water residence time

$t_w R$ = first tracer arrival time

Since C_o equals the total mass input over time At divided by the total volume flow rate time At, $M/(Q\Delta t)$, and the input pulse duration is very small, the solution can be rewritten as follows:

$$C_f = \frac{M}{Q\Delta t}\left(f(t+\Delta t) - f(t) \right)$$

$$= \frac{M}{Q} \lim_{\Delta t \to 0} \left(\frac{f(t+\Delta t) - f(t)}{\Delta t} \right)$$

$$= \frac{M}{Q} \frac{\partial f}{\partial t}$$

Because $\frac{d\left(\text{erfc}(x)\right)}{dt} = -\frac{2}{\sqrt{\pi}} e^{-x^2} \frac{dx}{dt}$ **, we have that**

$$C_f = \frac{M}{Q} \left\{ \frac{D_e t_w}{D_a^{0.5}(t_w R)^{1.5} \delta \sqrt{\pi} \left(\frac{t}{t_w R} - 1 \right)^{1.5}} \exp\left(\frac{-\left(\frac{D_e t_w}{D_a^{0.5}(t_w R)^{0.5} \delta} \right)^2}{\left(\frac{t}{t_w R} - 1 \right)} \right) \right\}$$

(3.12)

If $a = \frac{D_e}{D_a^{0.5}} \frac{t_w}{(t_w R)^{0.5}} \frac{1}{\delta} = (D_e \phi t_w)^{0.5}/\delta$ and $\beta = \frac{1}{t_w R}$

are substituted into Eqn 3.12, the following simplified solution is obtained:

$$C_f = C(t;\alpha,\beta) = \frac{M}{Q} \left\{ \frac{\alpha\beta}{\sqrt{\pi}(\beta t - 1)^{1.5}} \exp\left(- \frac{\alpha^2}{(\beta t - 1)} \right) \right\} \quad (3.13)$$

Rewriting the nonlinear parameters in terms of a_j and E (a linear scaling parameter) yields,

$$C = C_f \frac{Q}{M} = \frac{E \alpha_1 \alpha_2}{\sqrt{\pi}(\alpha_2 t - 1)^{1.5}} \exp\left(- \frac{a_1^2}{\alpha_2 t - 1} \right) \quad (3.14)$$

where, $\alpha_1 = \alpha$ and $\alpha_2 = \beta$.

The linear parameter E normalizes the flow fraction to one. This normalization is needed because precise information on the initial concentration injected into the fracture system connected with the producing well is not available. This does not affect the shape of the calculated tracer profile, but merely the size.

Tester, Bivens, and Potter (1982) proposed the use of an objective function F over N measured data points in order to analyze for optimum values α_1 and α_2 in the transfer function $C(t;\alpha_1,\alpha_2)$ for a given tracer return profile. When F, given by

$$F = \sum_{i=1}^{N} \left(C(t;\alpha_1,\alpha_2) - C_i\right)^2 \qquad (3.15)$$

is minimized, optimum values of α_1 and α_2 result. A multifracture model assuming one-dimensional flow in separate fractures and which gives the predicted tracer concentration response is given by

$$C = \sum_{j=1}^{M} \varepsilon_j C_j(t;\alpha_{1j},\alpha_{2j}) \qquad (3.16)$$

where ε_j is the fraction of flow in fracture path j. The relative flow fractions in the fracture system communicating with the production well and the injection well is given by

$$\sum_{j=1}^{M} \frac{E_j}{E} = 1 \qquad (3.17)$$

This multifractvre model is used to determine whether the tracer returns to a producing well is a result of flow through one or more fractures. Once the above objective function is minimized, the resulting optimized parameters are used to give information about the fracture system and flow mechanisms in the geothermal reservoir.

D. Computer Program

Optimization of the parameters in the transfer function $C(t;\alpha_1,\alpha_2)$ **is** accomplished using a nonlinear least-squares method of curve fitting. The main program calls for the input **of** the tracer return data, the number of parameters being used, and estimates of the nonlinear parameters. Subroutine VARPRO (written by Stanford University Department of Computer Science) and its accompaning subroutines are called to optimize the objective function. The main program then calls for the plotting of the tracer return data along with the computed best fit tracer return profile, and the optimal values of both the nonlinear and linear parameters of the given transfer function are printed.

VARPRO is based on a paper by Golub and Pereya (1973). Least-squares fit of nonlinear models of the form

$$C(t;\varepsilon,\alpha) = \sum_{j=1}^{M} \varepsilon_j C_j(t;\alpha_{ij}) \qquad i=1,2 \tag{3.18}$$

where,

M = number of proposed paths

t = independent variable

C_j = observed dependent variable

ε_j = linear parameter

α_{ij} = nonlinear parameters

can be performed by separately optimizing the linear parameters ε_j, and the nonlinear parameters α_{ij}.

The objective function,

$$F(\varepsilon_j,\alpha_j) = \sum_{i=1}^{N} \left(C_i - C(t;\varepsilon_j,\alpha_j)\right)^2 \tag{3.19}$$

is substituted with the first estimates of the nonlinear parameters a_j. The program iterates to determine the nonlinear parameters a_j, afterwhich the linear parameters ε_j are calculated.

The numerical nonlinear least-squares routine utilizes a Taylor expansion of the transfer function C by expanding with respect to the nonlinear parameters α_j. Linear least-squares is then used to determine the optimum values for the parameter increments, $\delta\alpha_j$. Mathematically this is shown as follows:

$$C(t_i;\alpha_j,\varepsilon_j) - C_o = \sum_{j=1}^{M}\left(\frac{\partial C_{oj}}{\partial \alpha_j}\,\partial\alpha_j\right) \qquad i=1,2,\ldots N \qquad (3.20)$$

The derivatives are evaluated at the starting point C_o. The residual **F** can then be expressed as,

$$F(\varepsilon_i,\alpha_i) = \Sigma\left((C_i - C_o) - \sum_{i=1}^{N}\frac{\partial C_o}{\partial \alpha_i}\,\delta\alpha_i\right)^2 \qquad (3.21)$$

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 the objective function $F(\varepsilon_j,\alpha_j)$. All parameters are incremented simultaneously so that the maximum variation of F is attained. The gradient of F determines the magnitude of the largest change, and giving it the opposite direction indicates the path of steepest descent. The objective is to change $\delta\alpha_j$ so that $F(\varepsilon_j,\alpha+\delta\alpha_j)<F(\varepsilon_j,\alpha_j)$. This is documented fully in the computer program.

To more fully explore the workings of this nonlinear least-squares method, see Fossum (1982) and the technical report by Golub and Pereya (1973). The computer program is listed in Appendix B along with a sample program output.

Section 4. RESULTS

The tracer return data for the various wells were fitted to the mathematical model using the computer program discussed previously. The figures shown in Appendix C show the fitted data profiles. The squares represent the data and the solid line is the calculated curve fit. For comparison purposes, accompanying some of the figures is a corresponding curve fit using the model presented by Fossum and Horne (1982). Remember that their model includes only advection and dispersion along one or more non-connecting or channeled fractures. The model presented in this report includes adsorption, advection, and diffusion into the surrounding porous matrix. This inclusion in the model of diffusion of tracer into the matrix gives considerable improvement in the curve fit of the tracer return profiles. Furthermore, in many of the wells only single fracture modeling is required to smoothly fit the data, whereas multi-fracture modeling was required in the cases presented by Fossum and Horne (1982). This is more pleasing since most curves can be fitted if several linear combinations of the single path equation are used, irrespective of the physical applicability.

Values for the flow fractions and nonlinear parameters α and β for the different calculated tracer return curve fits are given in Table 4.1.

For a few of the tracer return data, double fracture modeling was possible but did not substantially improve the single fracture curve fits. Where improvement was possible, however, these fits are included in place of the single curve fits.

It is noted that not all tracer returns are well fitted. These are wells WK68, 67, 116, and 121. Reasons for poor fits may be that (1) hydrodynamic dispersion down the length of the fracture needs to be included to better model the fluid and tracer flow, (2) the instantaneous linear adsorption assumption is not valid, (3) temperature effects on k, and D_e are of importance, or (4) the data itself for some reason is suspect.

Well WK121 is an interesting case in that a good fit was obtained when modeled as a double fracture case. However, a negative flow fraction is calculated. This anomaly could have a physical or mathematical significance, but most likely is an artifact of the curve fitting technique itself, in that more than one approach to convergence may be possible.

To gain a bettgr understanding of the parameters α and β, hypothetical tracer return profiles calculated by varying one of the parameters while keeping the other

TABLE 4.1

Production well	Injector-Producer Distance (meters)	Flow Fraction E_j/E	Nonlinear Parameters α	$1/\beta$ (days)	Minimum Flow Velocity (m/hr)	Fracture Width (mm) $\phi=1\%$	$\phi=5\%$
WK24	210	1.000	1.250	0.231	37.9	0.08	0.18
WK30	240	0.811	1.370	4.367	2.3	0.32	0.71
		0.189	1.270	3.212	3.1	0.29	0.66
WK48	120	0.450	1.393	0.293	17.1	0.08	0.18
		0.550	1.669	1.040	4.8	0.13	0.28
WK55	220	1.000	2.578	2.671	3.4	0.13	0.29
WK67	120	1.000	2.736	1.651	3.0	0.10	0.22
WK68	120	1.000	2.049	2.919	1.7	0.17	0.39
WK70	170	1.000	2.483	2.033	35	0.12	0.27
WK81	175	1.000	1.535	3.659	2.0	0.26	0.58
WK83	330	1.000	2.167	2.550	5.4	0.15	0.34
WK108	80	1.000	1.685	6.782	0.5	0.32	0.72
WK103	165	1.000	3.437	0.619	11.1	0.05	0.11
WK116	350	0.259	0.920	4.696	3.1	0.49	1.09
		0.741	3.844	0.626	23.4	0.04	0.10
WK121	490	1.000	0.916	1.451	14.1	0.27	0.61
		0.530	2.555	0.719	28.4	0.07	0.15
		-0.4'70	2.100	1.265	16.1	0.11	0.25

All production wells produce tracer injected at well WK107, except wells WK103, 116, and 121 which produce tracer injected at WK101.

constant were plotted. These plots are shown in Figures 4.1 and 4.2.

Note in Figure 4.1, that as the nonlinear parameter β increases, that is, the tracer arrival time decreases, the plotting trace begins at an earlier and earlier time. The peak also increases in height with increasing β.

In Figure 4.2, it can be seen that as α decreases the peak increases dramatically, and the tailing of the peak is reduced. These effects can be related to the physical parameters contained in the dimensionless parameter α. Remember that $\alpha=(D_e \phi t_w)^{0.5}/\delta$. If the fracture width 6 were to increase, causing α to decrease, it would be expected that the peak would be sharper and less spread out. This is because increased flow would occur through the enlarged fracture thus causing less matrix diffusion and less spreading of the tracer return profile. If the effective diffusion coefficient were increased in value (indicating increased diffusion in the porous matrix) it would be expected that the passage of tracer through the fracture/porous media system would be hindered, again causing the tracer profile to spread out. This effect is as observed for increased α. If the water retention time were to increase due to increased pathlength or decreased fluid velocity, an increase in profile spread would also be expected. Increased

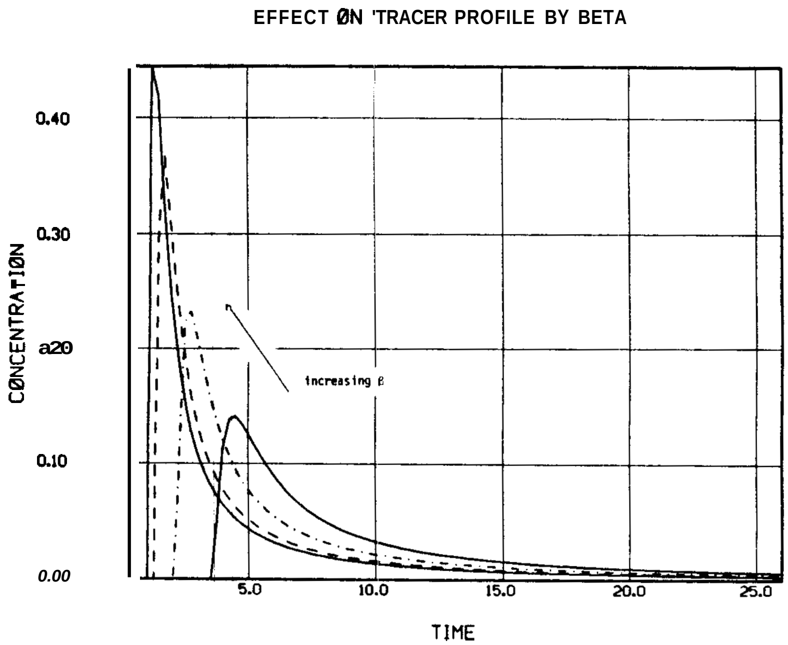


FIGURE **4.1** Plots of transfer function $C(t;\alpha,\beta)$ where β is varied and α is kept constant.

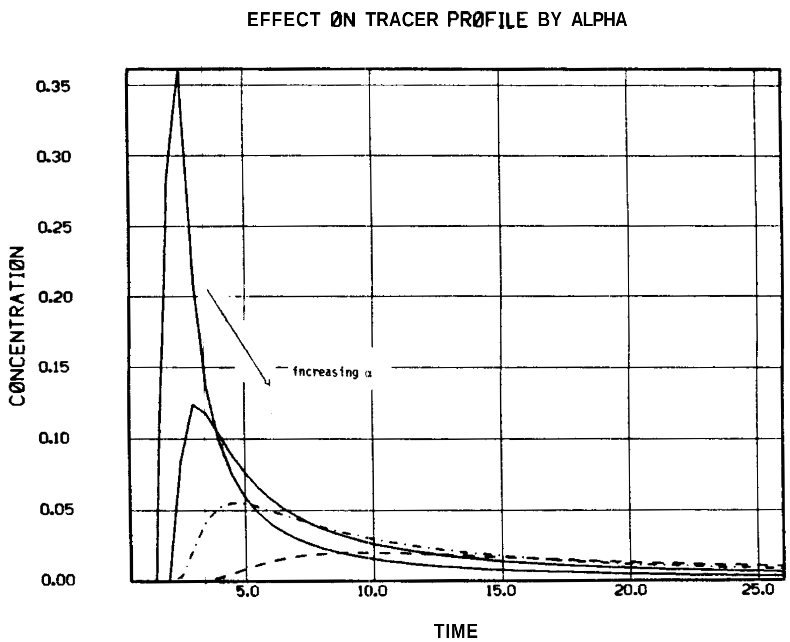


FIGURE 4.2 Plots of ,transfer function $C(t;\alpha,\beta)$ where α is varied and **B** is kept constant.

porosity would increase the volume of fluid in the matrix rock, allow more volume into which the tracer could diffuse, and thus increase the profile spread.

The goal of tracer return analysis is to infer information concerning the flow velocities, fracture widths, flow pathways, and reservoir rock and fluid properties such as diffusion and adsorption coefficients. To do this with the Wairakei data at hand requires some knowledge of the parameters ρ_b, D_e, k, and ϕ.

If a nonsorbing tracer is used then k=0, R=1, and $K_d \rho_b = \phi$. In this nonsorbing case some knowledge of the porosity ϕ and effective diffusion coefficient D_e is required to calculate fracture width values for the corresponding curve fit. In Table 4.1, fracture width values are given based on the nonsorbed tracer assumption and the effective diffusion coefficient value of 4.32 x 10^{-6} m^2/day (5 x 10^{-11} m^2/s). The value for D_e is a medium value obtained from a range of values given by Neretnieks (1980) for nonsorbing tracers in granites. This value is not necessarily the proper value to be used in this case, but it does allow approximate fracture width values to be calculated. Also, since the matrix porosity of the Waitakei reservoir is not definitively known, porosity values of 1% and 5% were used in the the calculations.

In Table 4.1, flow velocities have been calculated based on the Injector-producer distances and calculated first tracer arrival times. An assumption of the tracer not being sorbed to the reservoir rock (R=1) is also made in these calculations. As the injector-producer distances are not necessarily representative of pathlengths in the reservoir, these calculated velocities are minimum values.

Section 5. CONCLUSIONS AND RECOMMENDATIONS

1. Tracer diffusion into the matrix of the Wairakei geothermal reservoir **is** an important factor in the mechanism of fluid flow. Estimated reservoir parameters such as fracture widths, fluid velocities and dispersion characteristics are difficult to accurately interpret in a fractured reservoir without accounting for matrix diffusion. The diffusion of tracers into the rock matrix and their sorption onto the surfaces of the rock are the main mechanisms retarding migration through fractures.

2. In using the fracture model presented by Fossum and Horne to analyze the Wairakei data, a double flowpath model gave a ,more accurate data match than a single component model. However, **in** using the matrix diffusion model presented in this report, single fracture flowpath modeling was sufficient in many of the cases.

3. Without further investigation of representative values for the effective diffusion coefficient D_e, bulk rock density ρ_b, porosity ϕ, and the adsorption distribution coefficient k, quantitative values for the various reservoir and fluid flow properties cannot be

accurately calculated for the Wairakei reservoir.

4. Further study into the modeling of tracer flow through fractured media which takes into account hydrodynamic dispersion down the length of the fracture in addition to diffusion into the porous matrix may be warranted.

Nomenclature

α, β	nonlinear parameters
C_f	concentration of tracer in fracture
C_p	concentration of tracer in porous matrix
D	hydrodynamic dispersion coefficient
D_a	apparent diffusion coefficient
D_e	effective diffusion coefficient
δ	fracture width
E	linear scaling factor
E	fraction of flow
F	objective function
k	adsorption distribution coefficient
$K_d \rho_b$	volumetric sorption equilibrium constant
M	number of proposed fracture paths
N	number of data points
ρ_b	bulk density of the medium
ρ_s	solid rock density
q	Darcy velocity
R	retardation factor
S	amount of salute in the sorbed phase
t_w	water residence time
ϕ	porosity
U_f	fluid velocity in the fracture

x_o pathlength of fracture from injection well to production well.

x, y Cartesian directions

References

1. Breitenbach, K.A., Chemical Tracer Retention in Porous Media, Stanford Geothermal Program, SGP-TR-53, Stanford CA, May 1982.

2. Carslaw, H.S. and Jaeger, J.E., Conduction of Heat in Solids, 2nd ed., Oxford University Press, New **York**, 1959.

3. DiPippo, R., Geothermal Energy as a Source of Electricity, **U.S.** DOE, Washington, D.C., 1980.

4. Fossum, M.P., Tracer Analysis in a Fractured Geothermal Reservoir: Field Results from Wairakei. New Zealand, Stanford Geothermal Program, SGP-TR-56, Stanford CA, June 1982.

5. Fossum, M.P. and Horne, R.N., "Interpretation of Tracer Return Profiles at Wairakei Geothermal Field Using Fracture Analysis," Geothermal Resources Council, Transactions, Vol. 6, October 1982.

6. Gershon, N.D. and Nir, A., "Effects of Boundary Conditions of Models on Tracer Distribution in Flow through Porous Mediums," Water Resources Research, Vol. **5**, No. 4, 830-839, 1969.

7. Golub, G.H. and Pereya, V., "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate," SIAM J. Numerical Analysis, Vol. **10,** No. 2, 413-431, 1973.

8. Grindley, G.W., The Geology, Structure, and Exploitation of the Wairakei Geothermal Field, Taupo, New Zealand, N.Z. Geological Survey Bull. 75, 1965.

9. Grisak, G.E. and Pickens, J.F., "Solute Transport Through Fractured Media," Water Resources Research, **Vol.** 16, No. 4, 719-739, 1980.

10. Horne, R.N., Effects of Water Injection Into Fractured Geothermal Reservoirs: A Summary of Experience Worldwide, Stanford Geothermal Program, SGP-TR-57, Stanford CA, June 1982.

11. Horne, R.N. and Rodriguez, F., "Dispersion in Tracer Flow in Fractured Geothermal Systems," Geophysical Research Letters, Vol. 10, No. 4, 289-292, 1983.

12. Laoidus. L. and Amundson, N.R.. "Mathematics of Adsorption in Beds," J. Physical Chem., Vol. 56, 984-988, 1952.

13. Lester, D.H., Jansen, G. and Burkholder, H.C., "Migration of Radionuclide Chains Through an Adsorbing Medium," AIChE Symp. Series, Vol. 71, No. 152, 202-213, 1975.

14. McCabe, W.J., Barry, B.J. and Manning, M.R., "Radioactive Tracers in Geothermal Underground Water Flow Studies," Geothermics, Vol. 12, 83-110, 1983.

15. Nakamura, H., "Development and Utilization of Geothermal Energy in Japan," Geothermal Resources Council, Transactions, Vol. 5, 33, 1981.

16. Neretnieks, I., "Diffusion in the Rock Matrix: An Important Factor in Radionuclide Retardation?," J. of Geophysical Research, Vol. 85, No. B8, 4379-4397, 1980.

17. Neretnieks, I., Eriksen, T. and Tahtinen, P., "Tracer Movement in a Single Fissure in Granitic Rock: Some Experimental Results and Their Interpretation," Water Resources Research, Vol. 18, No. 4, 849-858, 1982.

18. Tester, J.N., Bivens, R.L. and Potter, R.M., "Interwell Tracer Analysis of a Hydraulically Fractured Granitic Geothermal Reservoir," Society of Petroleum Engineers Journal, Vol. 22, 537-545, 1982.

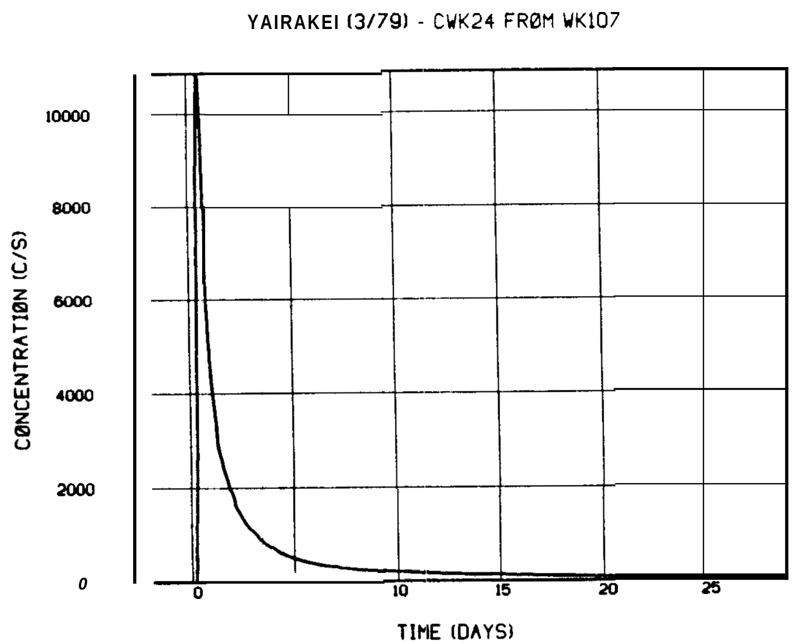
19. Wagner, O.R., "The Use of Tracers in Diagnosing Interwell Reservoir Heterogeneities - Field Results," J. of Petroleum Technology, November, 1977.

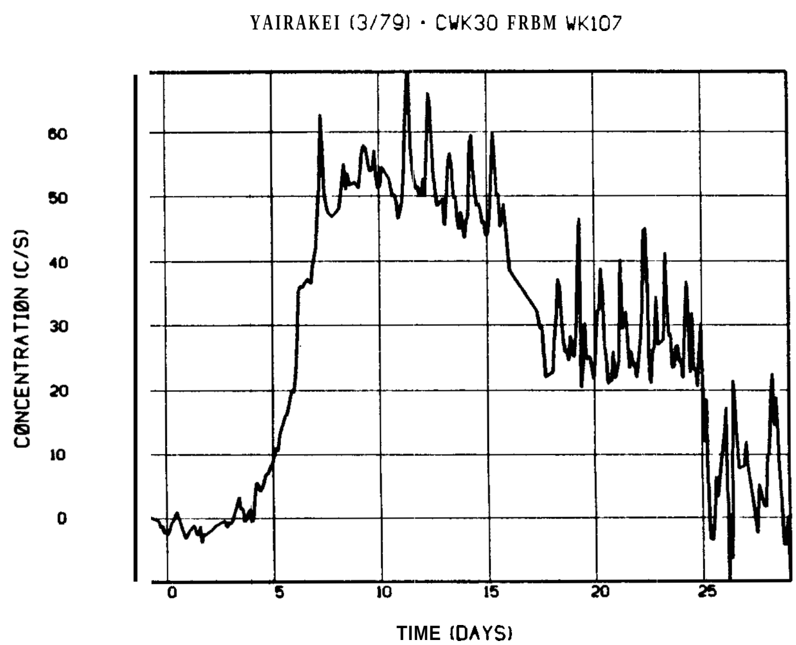
20. Webster, D.S., Proctor, J.F. and Marine, I.W., "Two-Well Tracer Test in Fractured Crystalline Rock," Geological Survey Water-Supply Paper, 1544-1, 1970.

Appendix A

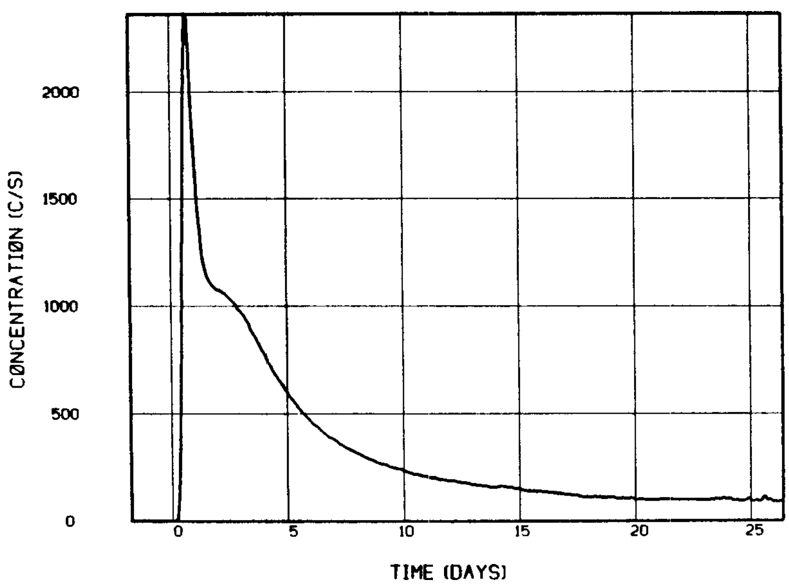
Tracer Return Profiles

This data was collected at the Wairakei geothermal field by the Institute of Nuclear Sciences, Department of Scientific Research, Gracefield, New Zealand, and was made available to the Stanford Geothermal Program by Dr. W. J. McCabe.

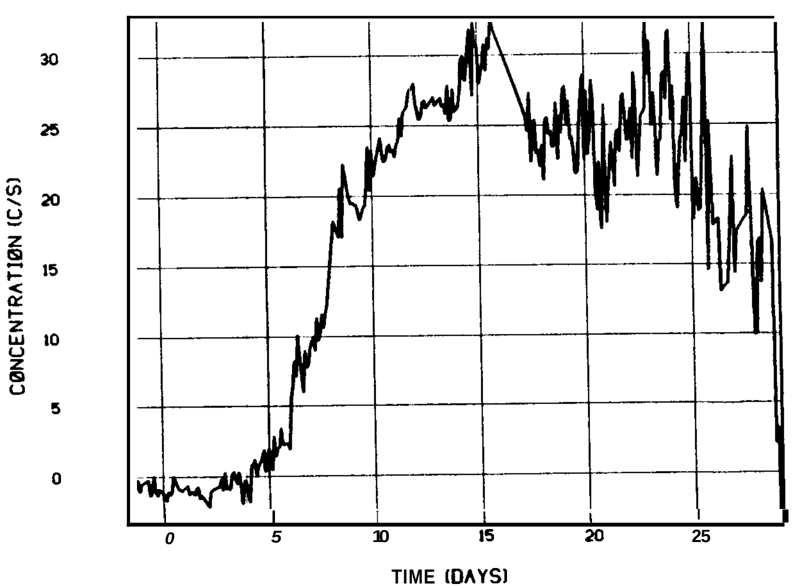

YAIRAKEI (3/79) - CWK24 FRØM WK107
CØNCENTRATIØN (C/S)
10000
8000
6000
4000
2000
0
0
10
15
20
25
TIME (DAYS)


YAIRAKEI (3/79) · CWK30 FRBM WK107
CØNCENTRATIØN (C/S)
60
50
40
30
20
10
0
0
5
10
15
20
25
TIME (DAYS)

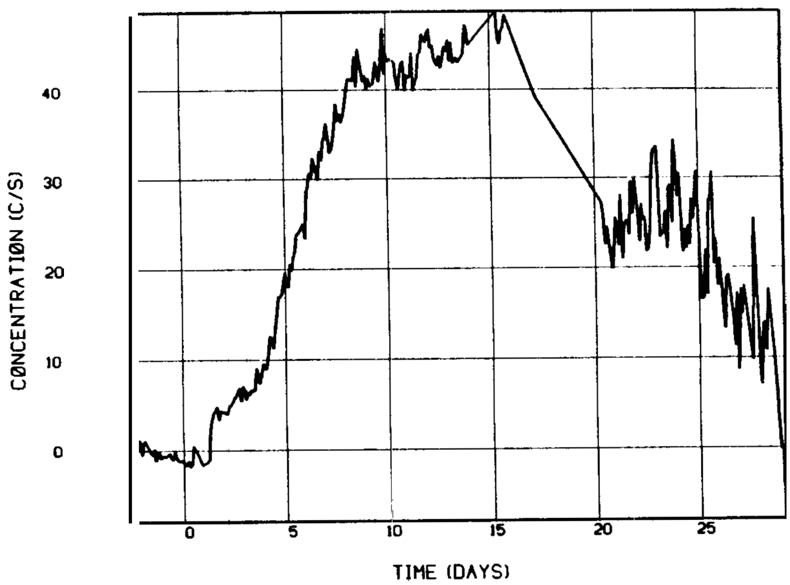
WAIRAKEI (3/79) - CWK48 FRBM WK107



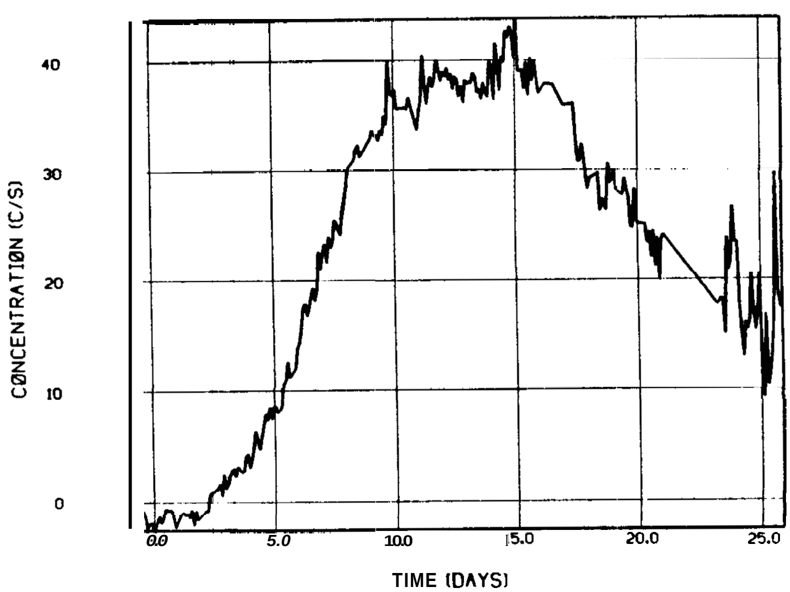
YAIRAKEI (3/79) - CWK55 FRBM WK107



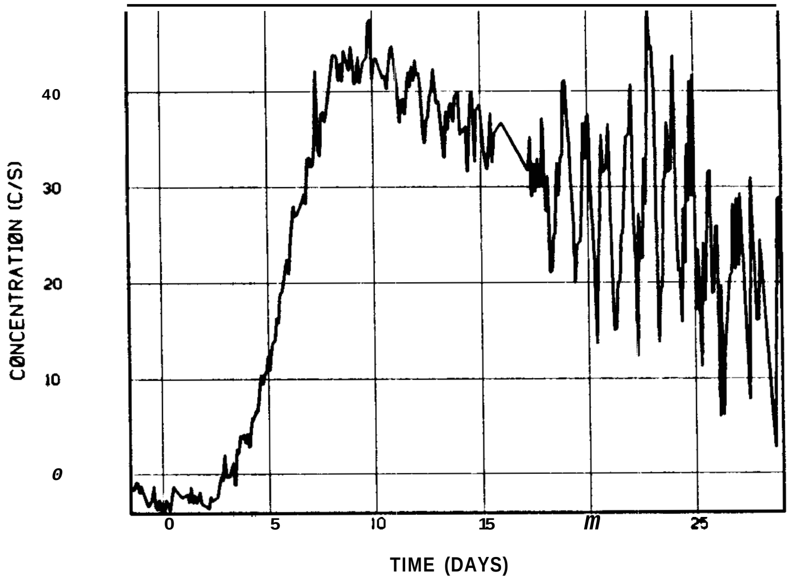
YAIRAKEI (3/79) - CWK67 FRØM WK107



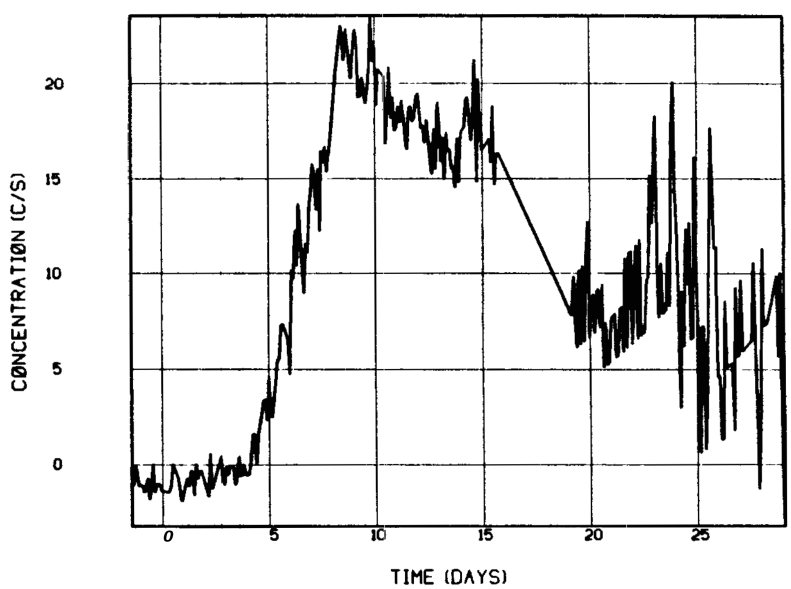
WAIRAKEI (3/79) - CWK68 FRBM WK107



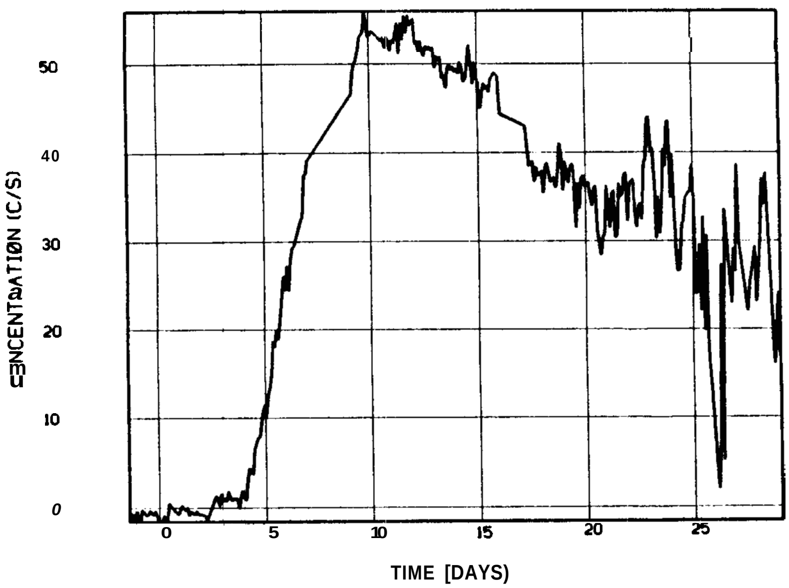
YAIRAKEI (3/79) - CWK70 FRBM WK107



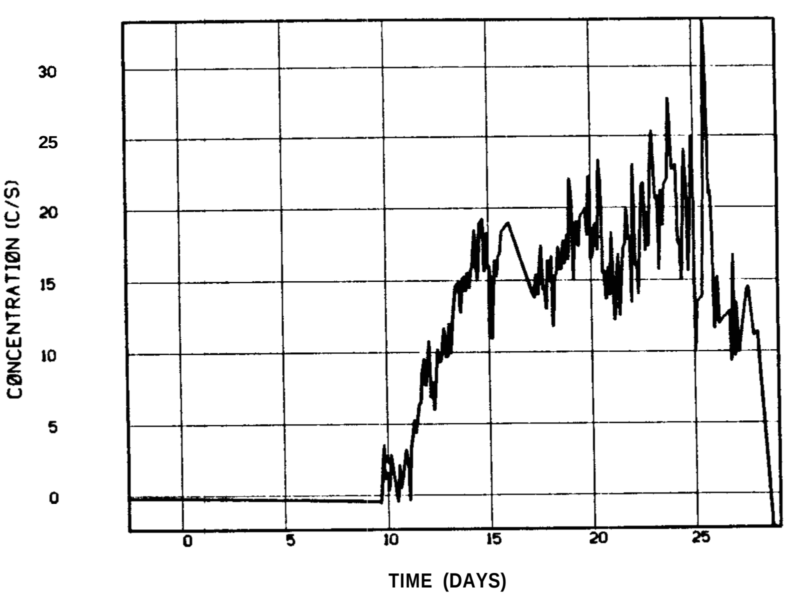
UAIRAKEI 13/79) - CWK81 FRBM WK107



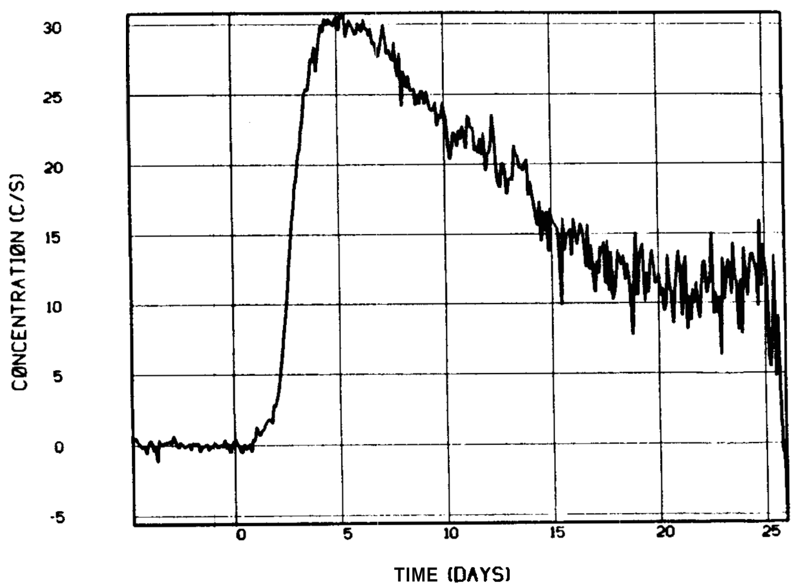
UAIRAKEI (3/79) - CWK83 FRØM WK107



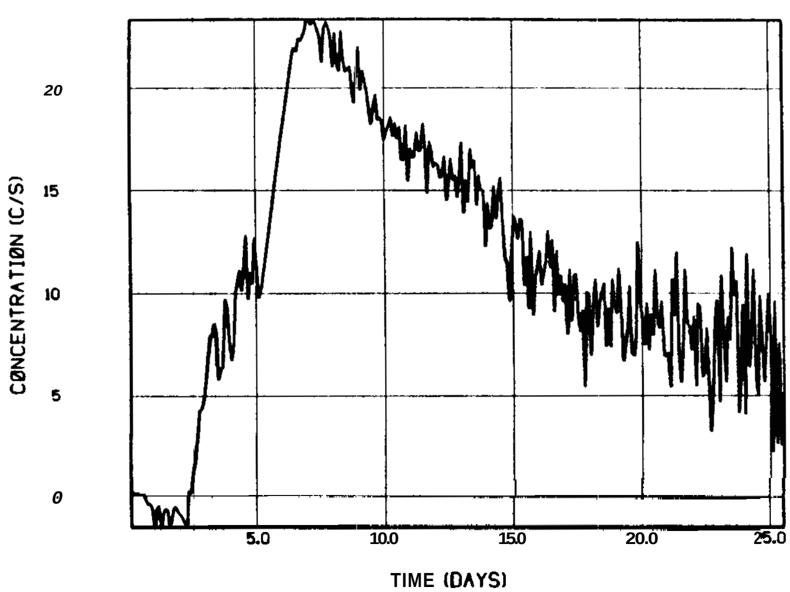
UAIRAKEI (3/79) - CWK108 FRØM WK107



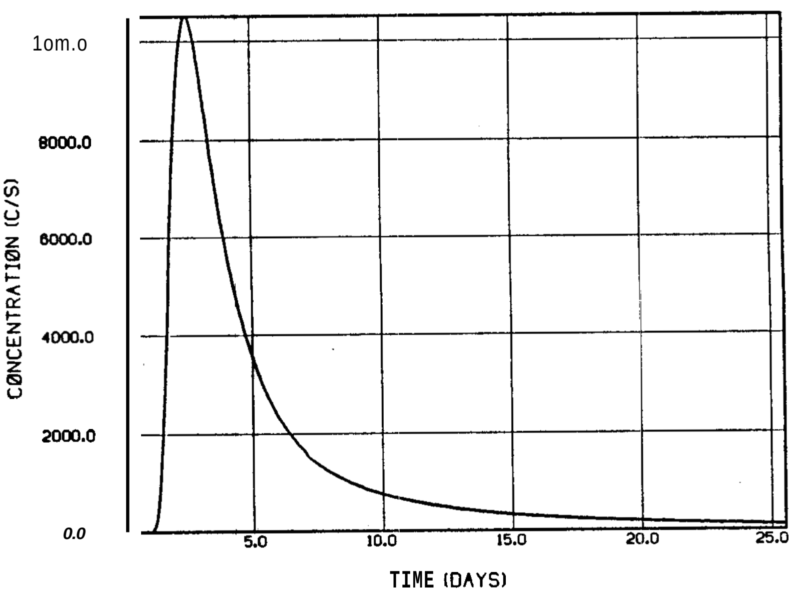
UAIRAKEI 17/79) - CWK103 FRBM WK101



UAIRAKEI (7/79) - CWK116 FRBM WK101



WAIRAKEI 17/79) - CWK121 FRBM WK101



Appendix B

Computer Program and Sample Output

```
//TRACER JOB
// EXEC FORTCL
//FORT.SYSIN DD *
C ********************************************************************
C *********************************************************************
C
C                  PROGRAM BEGINS
C
C *********************************************************************
C *********************************************************************
C
C
      IMPLICIT REAL*8(A-B,D-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(400),T(400),ALF(14),BETA(7),W(400),A(400,13),
     *INC(14,8),C(400,8),CTITLE(20),CT(400),CY(400),DIM(7),OUT(7)
C
C
C          SET PARAMETERS FOR VARPRO.
C
C
      EXTERNAL ADA
      IPLOT=1
      IF (IPLOT.EQ.1) CALL STARTG('GENIL*',0.0)
      NMAX = 400
      IPRINT=1
C
C
C          READ DATA SEQUENTIAL ORDERING AND
C          PROPER FORMATTING ARE IMPORTANT.
C
      READ (5,70) CTITLE
 70   FORMAT (20A4)
      WRITE (6,71) CTITLE
 71   FORMAT (1H0,10X,20A4)
C
C
C        NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (5,*) NL
      WRITE(6,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C        L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      L=NL/2
C
C
C        ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (5,*) (DIM(I),OUT(I),I=1,L)
      DO 80 I=1,L
```

```
      II=2*I-1
      ALF(II)=DIM(I)
 80   ALF(II+1)=1./OUT(I)
      WRITE(6,21)(ALF(I),I=1,NL)
 21   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F7.3))
      WRITE (6,20) (DIM(I),OUT(I),I=1,L)
 20   FORMAT (/,'0 DIMENSIONLESS NUMBER    TRACER ARRIVAL TIME',/,
     # (5X,F9.5,22X,F7.3))
C
C
      LPP2=L+NL+2
C
C
C
C       N IS THE NUMBER OF OBSERVATIONS
C
C
      READ(5,*) N
      WRITE(6,35) N
 35   FORMAT(/1H0,10X,'NUMBER OF OBSERVATIONS'//(I4))
C
C
C       IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      IV=1
C
C
C     T IS THE INDEPENDENT VARIABLE
C     Y IS THE N-VECTOR OF OBSERVATIONS
C
C
      READ(5,*)(T(I),Y(I),I=1,N)
     WRITE(6,60)(T(I),Y(I),I=1,N)
 60  FORMAT(1H0,'INDEPENDENT VARIABLES       DEPENDENT VARIABLES'//
     * ,(5X,F8.3,21X,F9.3))
C
C
C     W(I) ARE THE WEIGHTING PARAMETERS
C
C
        DO 1 I=1,N
 1       W(I)=1.0
C
C
      CALL VARPRO(L,NL ,N ,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
      WRITE (6,13)
      LP1=L+1
      CALL ADA (LP1,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,2)
      DO 8 I=1,N
      C(I,LP1)=0.
      DO 9 J=1,L
      C(I,J)=BETA(J)*A(I,J)
 9    C(I,LP1)=C(I,LP1)+C(I,J)
      WRITE (6,14) Y(I),C(I,LP1),(C(I,J),J=1,L)
      CY(I)=Y(I)
      CT(I)=T(I)
 8    CONTINUE
```

```
 13     FORMAT(1H0,'  ACTUAL      CALC     COMP#1     COMP#2',//)
 14     FORMAT (1X,8F10.4)
C
        DO 22 I=1,L
        II=2*I-1
        DIM(I)=ALF(II)
 22     OUT(I)=1./ALF(II+1)
        SUM=0.
        DO 25 J=1,L
 25     SUM=SUM+BETA(J)
       DO 93 I=1,L
93     BETA(I)=BETA(I)/SUM
        WRITE (6,38) (BETA(I),DIM(I),OUT(I),I=1,L)
 38     FORMAT (/,'0 FRACTION    DIMENSIONLESS NUMBER    ARRIVAL TIME',
      A /,(5X,F7.3,5X,F7.3,22X,F7.3))
        IF (IPLOT.NE.1) STOP
        CALL GRAPHG ('*',0,N,CT,CY,4,'TIME (DAYS)*',
      * *CONCENTRATION (C/S)*',CTITLE)
        CALL LINESG ('SOLD,VBRT*',N,CT,C(1,LP1))
        CALL EXITG
         STOP
         END
C
C
C ******************************************************************
C ******************************************************************
C
C                    SUBROUTINES
C
C ******************************************************************
C ******************************************************************
C
C
C
C
        SUBROUTINE ADA (LP,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION ALF(NL),A(NMAX,LPP2),T(NMAX),INC(14,8),D(400,7)
C
C
C
        L=LP-1
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
        IF(ISEL.EQ.2)  GO TO 90
        IF(ISEL.EQ.3)  GO TO 165
        DO 1 J=1,L
        DO 1 K=1,NL
        INC(K,J)=0.0
        IF ((K+1)/2.EQ.J) INC(K,J)=1.0
 1      CONTINUE
C
C
C
```

```
C
C       THE VECTOR-SAMPLED FUNCTIOIHS 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
   90        DO 81 I=1,N
      DO 81 J=1,L
      K1=2*J-1
      K2=2*J
      IF (ALF(K2)*T(I).GT.1.0) GO TO 82
      A(I,J)=0.
      D(I,J)=0.
      GO TO 81
82    A(I,J)=ALF(K1)*ALF(K2)/(1.772453851*
     #  (ALF(K2)*T(I)-1.0)**(1.5))*
     #  DEXP(-1.*ALF(K1)**2/(ALF(K2)*T(I)-1.0))
       D(I,J)=A(I,J)
C
   81       CONTINUE
C
C
C
        IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
  165       DO 170 I=1,N
C
C
      DO 170 J=1,NL
      K1=(J+1)/2
      K2=2*K1
      K3=K2-1
      JJ=L+J+1
      IF (ALF(K2)*T(I).GT.1.0) GO TO 171
      A(I,JJ)=0.
      GO TO 170
 171  IF ((J/2)*2.EQ.J) GO TO 300
      A(I,JJ)=D(I,K1)*(1.0/ALF(K3) -
     #  2.0*ALF(K3)/(ALF(K2)*T(I)-1.0))
      GO TO 170
300   A(I,JJ)=D(I,K1)*(1.0/ALF(K2)-1.5/(ALF(K2)*T(I)-1.0) *
     #  T(I)+(ALF(K3)**2)*T(I)/((ALF(K2)*T(I)-1.)**2))
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
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 FIX ONE OR MORE OF THE BETA'S  (RATHER THAN LET THEM BE
C       DETERMINED).  VRRPRO 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 THE
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 US'ED 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 EXCEED 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 = 0.
C                  CAUTION:  THE DECLARED ROW DIMENSION OF INC (IN ADA)
C                  MUST CURRENTLY BE SET TO 12.  SEE 'RESTRICTIONS' BELOW.
C          NMAX    THE DECLRRED 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 THE
C                  NONLINEAR ONES), COLUMN L+NL+1 WILL CONTAIN THE
C                  WEIGHTED RESIDUALS (Y - ETA), A(1, L+NL+2) WILL CONTAIN
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 IS
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 ERRORS.
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                      PARAVETER TOO LARGE.)  ALSO SEE IERR = -8 BELOW.
C                 -4   INPUT ERROR IN PARAMETER N, L, NL, LPP2, OR NMAX.
C                 -5   INC MATRIX 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 HAS
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, ORFAC2, BACSUB, 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 THEIR
C        PARTIAL DERIVATIVES D PHI(J)/D ALF(K), AT THE SAMPLE POINTS
C        T(I).  THIS ROUTINE MUST BE DECLARED 'EXTERNAL' IN THE CALLING
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 FUNICTIONS 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 APPEAR
C        FIRST, STARTING IN COLUMN 1.  THE COLUMN N-VECTORS OF NONZERO
C        PARTIAL DERIVATI'VES 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 COMPUTE 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, AND
C        DERIVATIVES ARE NOT NEEDED.)
C
C     RESTRICTIONS
C
C           THE SUBROUTINES DPA, INIT (AND ADA) CONTAIN THE LOCALLY
C        DIMENSIONED MATRIX INC, WHOSE DIMENSIONS ARE CURRENTLY SET FOR
C        MAXIMA OF L+1 = 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 PROGRAM
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 MATRIX
C        A MUST BE LINEARLY INDEPENDENT.
C
C           OPTIONAL NOTE:  AS WILL BE NOTED FROM THE SAMPLE SUBPROGRAM
C        ADA, THE 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 EXPENSE 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, AND
C        AFTER THE COMPUTATION OF THE PHI'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 THE
C        DATA STATEMENTS IN VARPRO, DPA, POSTPR, AND VARERR.  THE
C        PROGRAM HAS BEEN CHECKED FOR PORTABILITY BY THE BELL LABS PFORT
C        VERIFIER.  FOR MACHINES WITHOUT DOUBLE PRECISION HARDWARE, 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 IN
C        VARPRO, (C) DSIGN, DSQRT AND DABS TO SIGN, SQRT AND ABS,
C        RESPECTIVELY, AND (D) DEXP TO EXP IN THE SAMPLE PROGRAMS 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 THE 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 INCLWDE 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 OBSERVATIONS, OTHERWISE THE VARIANCES WILL NOT BE THE
C        SAME.  IF THE OBSERVATIONS ARE NOT THE SAME SIZE, THIS CAN BE
C        CURED BY WEIGHTIMG.
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 AXES 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 PARTIAL DERIVATIVES.  IF THESE
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        HAY 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 LEVEMBERG-MARQUARDT ALGORITHH.  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 CAN
C        BE FOUND IN REF. (3), ANI) 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.  DRAPER, 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, EPS1, 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 CONVERGENCE
C            TEST:  EPSl I$ AN ABSOLUTE AND RELATIVE TOLERANCE FOR THE
C            NORM OF THE PROJECTION OF THE RESIDUAL ONTO THE RANGE OF THE
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
      NLP1 = 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
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 DFA (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
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 PARAMETE
   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, NLPl
               JSUB = LP1 + J
               ISUB = NLPl + 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
   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, I5, 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 RESIDUAL
C          PRODUCED BY DPA, R2 IS 'THE TRANSFORMED RESIDUAL FROM DPA, AND
C          DR' IS IN UPPER TRIANGULAR FORM (AS IN REF. (2), P. 18).
C          DR IS STORED IN ROWS L+1 TO N AND COLUMNS L+2 TO L + NL + 1 OF
C          THE MATRIX A (I.E., COL'UMNS 1 TO NL OF THE MATRIX 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, XNORM
C
      NL = NLPl - 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, NLP1
            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 USE
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, NLPl
            JSUB = NLPl + J
            B(K, J) = B(LPK, J)
   40       B(JSUB, K) = B(LPK, J)
   50    B(NL23, K) = XNORM(K, B(LP1, K))
C
   99 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 DERIVATIVE
C         ACCORDING TO REF. (5), IS
C                                                  -1
C                     D(Q2 * Y) = -42 * 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(14, 8)
      LOGICAL NOWATE, PHILPl
      EXTERNAL ADA
C
      IF (ISEL .NE. 1) GO TO 3
         LP1 = L + 1
         LNL2 = L + 2 + NL
         LP2 = L + 2
         LPPl = LPP2 - 1
         FIRSTC = 1
         LASTC = LPPl
         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 = LPPl
      FIRSTR = (4 - ISEL)*L + 1
      GO TO 50
C                                                     ISEL = 2
    6 FIRSTC = NCONPl
      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 (PHILPI) 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), GETTING
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 66
         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 66
         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 TRIANGULAR
C           SYSTEM S*BETA = R1 FOR THE LINEAR PARAMETERS BETA.  BETA
C           OVERWRITES Rl.
C
   85 IF (L .GT. 0) CALL BACSUB (NMAX, L, A, R)
C
C           MAJOR PART OF KAUFMAN'S SIMPLIFICATION OCCURS HERE.  COMPUTE
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*BETA
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 R2 = Q2*Y (IN COL. L+1) IS COPIED
C           TO COLUMN L+NL+2.  OTHERWISE ALL OF COLUMN L+1 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 = NCONP1, 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 = ACUM + A(I, M)
   90       A(I, KSUB) = 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(14, 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. LNL2 .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 99
    9    W(I) = DSQRT(W(I))
```

```
C
      NCON = L
      NCONP1 = LP1
      PHILPl = L .EQ. 0
      IF (PHILPl .OR. NL .EQ. 0) GO TO 99
C                                        CHECK INC MATRIX FOR VALID INPUT AND
C                                        DETERMINE NUMBER OF CONSTANT FCNS.
      P = 0
      DO 11 J = 1, LP1
         IF (P .EQ. 0) NCONP1 = J
         DO 11 K = 1, NL
            INCKJ = 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                                                 INPUT ERROR IN INC MATRIX
   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
      LPNL = LNL2 - 2
      LNL1 = LPNL + 1
      DO 10 1 = 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 = KP1r 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 MATRIX
   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
      WRITE (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)
      WRITE (OUTPUT, 209)
   99 RETURN
C
  209 FORMAT (1H0, 50(1H'))
  210 FORMAT (20H0  LINEAR PARAMETERS // (7E15.7))
```

```
  211 FORMAT (23H0  NONLINEAR PARAMETERS // (7E15.7))
  214 FORMAT (21H0  NORM OF RESIDUAL =, E15.7, 33H EXPECTED ERROR OF OBS
     XERVATIONS =, E15.7, / 39H   ESTIMATED VARIANCE OF OBSERVATIONS =,
     X E15.7 )
  215 FORMAT (95H  WARNING -- EXPECTED ERROR OF OBSERVATIONS IS NOT ZERO
     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 DISAGRE
     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, SE
     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
//LKED.SYSLMOD DD DSN=WYL.JE.CLJ.SETH(MELISSA),DISP=OLD
```

WAIRAKEI (3/79) - CWK24 FROM WK107*

NUMBER OF NONLINEAR PARAMETERS

2

INITIAL EST. OF NONLIN. PARAMETERS

2.000
5.000

DIMENSIONLESS NUMBER	TRACER ARRIVAL TIME
2.00000	0.200

NUMBER OF OBSERVATIONS

93

INDEPENDENT VARIABLES	DEPENDENT VARIABLES
0.214	28.510
0.297	2043.906
0.380	7757.337
0.464	10865.406
0.547	10752.924
0.630	9576.211
0.714	8226.813
0.797	7012.052
0.880	5984.576
0.964	5198.999
1.047	4588.288
1.130	4092.422
1.297	3386.888
1.380	2895.055
1.464	2727.387
1.547	2606.242
1.630	2446.038
1.714	2321.840
1.797	2194.641
1.880	2078.807
1.964	1973.313
2.047	1890.512
2.130	1792.964
2.214	1615.511
2.630	1280.402
2.714	1232.835
2.797	1187.192
2.880	1145.455
2.964	1111.014
3.047	1079.655
3.130	1044.407
3.214	1002.250
3.297	944.850
3.380	909.742
3.464	879.628
3.547	848.368
3.630	817.099
3.714	792.777
3.797	769.525
3.880	748.345
3.964	751.802
4.047	714.117

4.130	695.975
4.214	658.868
4.297	638.992
4.380	618.845
4.464	607.310
4.630	572.912
4.714	559.063
4.797	543.320
4.880	532.194
4.964	517.672
5.047	502.855
5.130	491.666
5.214	483.097
5.297	469.851
5.380	462.071
5.464	450.761
5.547	446.069
5.630	432.668
5.880	407.597
5.964	397.611
6.047	392.204
6.130	380.086
6.214	375.645
6.297	369.281
6.380	359.749
6.630	345.746
6.714	340.128
6.797	338.155
6.880	338.703
6.964	333.483
7.047	331.370
7.130	317.503
7.214	305.410
7.297	300.365
7.380	295.185
7.464	292.692
7.547	287.393
7.630	284.710
7.714	279.273
7.797	275.762
7.880	271.511
8.130	263.995
8.380	256.012
8.464	249.844
8.547	247.240
8.630	244.606
8.880	236.365
8.964	234.301
9.047	229.099
9.130	225.334
9.214	223.842

NUMBER OF CONSTANT FUNCTIONS = 0

0 NORM OF RESIDUAL = 0.1280073D+05
NU = 0.1000000D+01
ITERATION 1 NONLINEAR PARAMETERS
0.1951013D+01 0.7306868D+01
1 NORM OF RESIDUAL = 0.5307679D+04
NU = 0.5000000D+00
NORM(DELTA-ALF) / NORMCALF) = 0.305D+00

```
ITERATION   2    NONLINEAR PARAMETERS
0.1790861D+01  0.6391822D+01
  1  NORM OF RESIDUAL =  0.4667214D+04
  NU =  0.2500000D+00
  NORM(DELTA-ALF) / NORM(ALF) =   0.140D+00
ITERATION   3    NONLIMEAR PARAMETERS
0.1397078D+01  0.4619397D+01
  1  NORM OF RESIDUAL =  0.2752487D+04
  NU =  0.1250000D+00
  NORM(DELTA-ALF) / NORM(ALF) =   0.376D+00
ITERATION   4    NONLINEAR PARAMETERS
0.1242511D+01  0.4320153D+01
  1  NORM OF RESIDUAL =  0.1722342D+04
  NU =  0.6250000D-01
  NORM(DELTA-ALF) / NORM(ALF) =   0.749D-01
ITERATION   5    NONLINEAR PARAMETERS
0.1249233D+01  0.4327062D+01
  1  NORM OF RESIDUAL =  0.1716735D+04
  NU =  0.3125000D-01
  NORM(DELTA-ALF) / NORM(ALF) =   0.214D-02
ITERATION   6    NONLINEAR PARAMETERS
0.1248118D+01  0.4323440D+01
  1  NORM OF RESIDUAL =  0.1716674D+04
  NU =  0.1562500D-01
  NORM(DELTA-ALF) / NORM(ALF) =   0.842D-03
ITERATION   7    NONLINEAR PARAMETERS
0.1248064D+01  0.4323031D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.7812500D-02
  NORM(DELTA-ALF) / NORM(ALF) =   0.916D-04
ITERATION   8    NONLIHEAR PARAMETERS
0.1248037D+01  0.4322911D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.3906250D-02
  NORM(DELTA-ALF) / NORM(ALF) =   0.272D-04
ITERATION   9    NONLIMEAR PARAMETERS
0.1248032D+01  0.4322888D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.1953125D-02
  NORM(DELTA-ALF) / NORM(ALF) =   0.530D-05
ITERATION  10    NONLIHEAR PARAMETERS
0.1248031D+01  0.4322883D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.9765625D-03
  NORM(DELTA-ALF) / NORM(ALF) =   0.121D-05
ITERATION  11    NONLINEAR PARAMETERS
0.1248031D+01  0.4322882D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.4882812D-03
  NORM(DELTA-ALF) / NORM(ALF) □   0.258D-06
ITERATION  12    NONLINEAR PARAMETERS
0.1248031D+01  0.4322881D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.2441406D-03
  NORM(DELTA-ALF) / NORM(ALF) =   0.564D-07
''''''''''''''''''''''''''''''''''''''''''''''''''''
 LINEAR PARAMETERS

0.1655775D+05
 NONLINEAR PARAMETERS
```

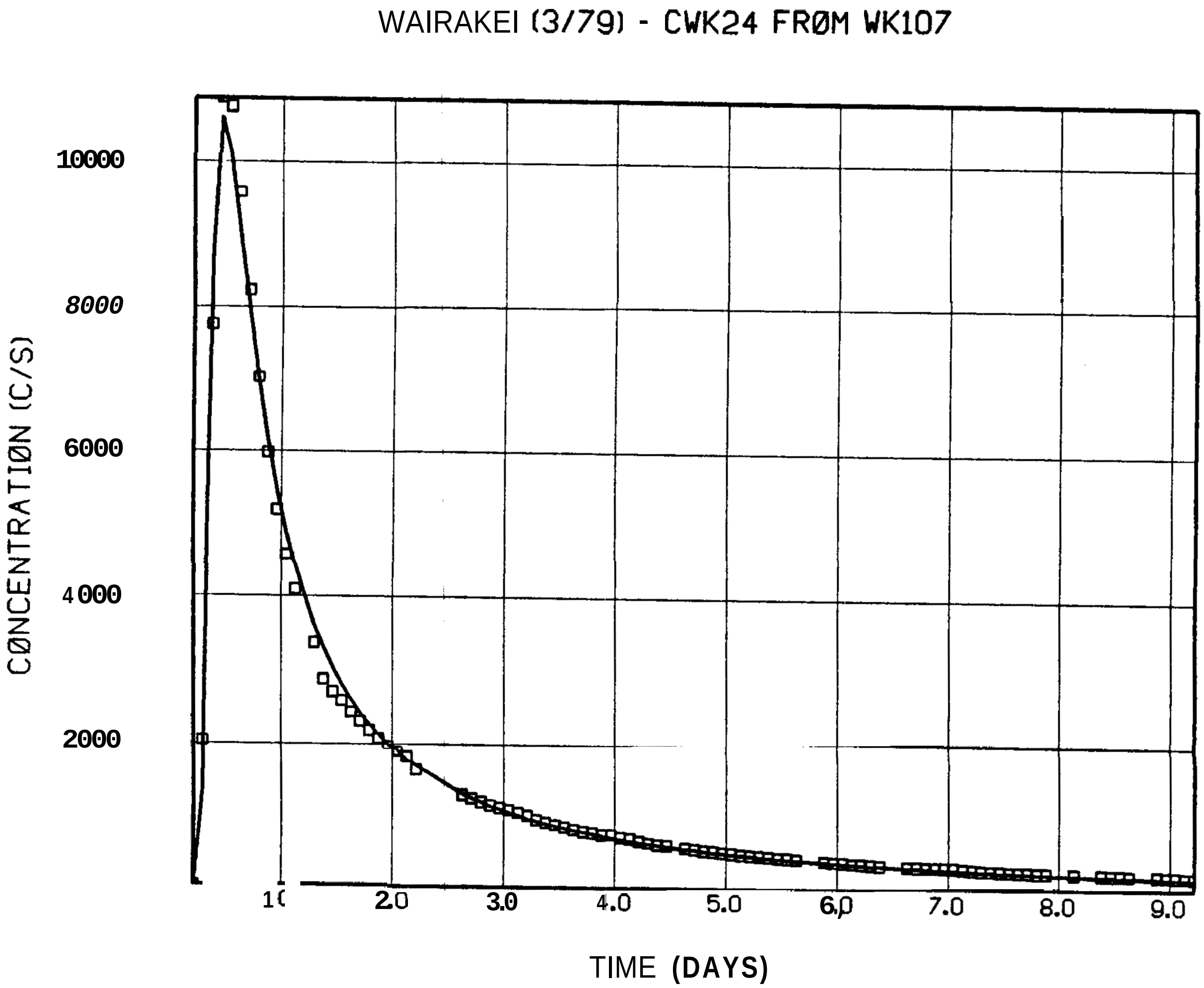
```
 0.1248031D+01  0.4322881D+01
  NORM OF RESIDUAL =   0.1716672D+04 EXPECTED ERROR OF OBSERVATIONS =  0.
  ESTIMATED VARIANCE OF OBSERVATIONS =  0.3274403D+05
 WARNING -- EXPECTED ERROR OF OBSERVATIONS IS NOT ZERO.  COVARIANCE MATR

*************************************************
```

ACTUAL	CALC	COMP#1	COMP#2
28.5100	0.0	0.0	
2043.9060	1380.2258	1380.2258	
7757.3370	8667.5898	8667.5898	
10865.4060	10619.6406	10619.6406	
10752.9240	10097.2422	10097.2422	
9576.2110	9022.6719	9022.6719	
8226.8130	7926.6328	7926.6328	
7012.0520	6970.8984	6970.8984	
5984.5760	6158.7656	6158.7656	
5198.9990	5467.9453	5467.9453	
4588.2880	4893.8203	4893.8203	
4092.4220	4407.9609	4407.9609	
3386.8880	3634.8909	363111.8909	
2895.0550	3328.4558	33281.4558	
2727.3870	3058.7834	30581.7834	
2606.2420	2825.5759	2825.5759	
2446.0380	2620.0679	26201.0679	
2321.8400	2435.8870	2435.8870	
2194.6410	2273.8853	2273.8853	
2078.8070	2128.8738	2128.8738	
1973.3130	1997.0005	1999.0005	
1890.5120	1879.4253	1874.4253	
1792.9640	1772.8572	1772.8572	
1615.5110	1674.8044	1674.8044	
1280.4020	1298.8887	12981.8887	
1232.8350	1239.8137	1239.8137	
1187.1920	1185.6943	1185.6943	
1145.4550	1135.3857	1135.3857	
1111.0140	1087.9790	1087.9790	
1079.6550	1044.2793	1044.2793	
1044.4070	1003.4192	1003.4192	
1002.2500	964.7019	964.7019	
944.8500	928.8235	926.8235	
909.7420	895.1094	895.1094	
879.6280	863.0107	863.0107	
848.3680	833.1306	833.1306	
817.0990	804.9316	804.9316	
792.7770	777.9731	773.9731	
769.5250	752.7786	752.7786	
748.3450	728.9126	726.9126	
751.8020	706.0134	704.0134	
714.1170	684.5388	684.5388	
695.9750	664.1294	664.1294	
658.8680	644.4846	649.4846	
638.9920	626.0051	626.0051	
618.8450	608.3909	608.3909	
607.3100	591.3887	591.3887	
572.9120	560.0254	566.0254	
559.0630	545.1943	545.1943	
543.3200	531.1709	531.1709	

532.1940	517.7378	517.7378
517.6720	504.7092	504.7092
502.8550	492.3628	492.3628
491.6660	480.5112	480.5112
483.0970	468.9929	468.9929
469.8510	458.0562	458.0562
462.0710	447.5374	447.5374
450.7610	437.2954	437.2954
446.0690	427.5527	427.5527
432.6680	418.1665	418.1665
407.5970	391.8708	391.8708
397.6110	383.6477	383.6477
392.2040	375.8010	375.8010
380.0860	368.2180	368.2180
375.6450	360.7998	360.7998
369.2810	353.7109	353.7109
359.7490	346.8511	346.8511
345.7460	327.4724	327.4724
340.1280	321.3625	321.3625
338.1550	315.5098	315.5098
338.7030	309.8323	309.8323
333.4830	304.2576	304.2576
331.3700	298.9116	298.9116
317.5030	293.7202	293.7202
305.4100	288.6172	288.6172
300.3650	283.7185	283.7185
295.1850	278.9568	278.9568
292.6920	274.2720	274.2720
287.3930	269.7700	269.7700
284.7100	265.3896	265.3896
279.2730	261.0762	261.0762
275.7620	256.9275	256.9275
271.5110	252.8874	252.8874
263.9950	241.3363	241.3363
256.0120	230.6383	230.6383
249.8440	227.2199	227.2199
247.2400	223.9243	223.9243
244.6060	220.7076	220.7076
236.3650	211.4695	211.4695
234.3010	208.5092	2081.5092
229.0990	205.6513	205.6513
225.3340	202.8581	202.8581
223.8420	200.0950	200.0950

FRACTION	DIMENSIONLESS NUMBER	ARRIVAL TINE
1.000	1.248	0.231


WAIRAKEI (3/79) - CWK24 FRØM WK107
10000
8000
6000
4000
2000
CØNCENTRATIØN (C/S)
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
TIME (DAYS)

Appendix C

Fitted Tracer Return Profiles

Appendix B

Computer Program and Sample Output

```
//TRACER JOB
// EXEC FORTCL
//FORT.SYSIN DD *
C ******************************************************************
C *******************************************************************
C
C                  PROGRAM BEGINS
C
C *******************************************************************
C *******************************************************************
C
C
      IMPLICIT REAL*8(A-B,D-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(400),T(400),ALF(14),BETA(7),W(400),A(400,13),
     *INC(14,8),C(400,8),CTITLE(20),CT(400),CY(400),DIM(7),OUT(7)
C
C
C          SET PARAMETERS FOR VARPRO.
C
C
      EXTERNAL ADA
      IPLOT=1
      IF (IPLOT.EQ.1) CALL STARTG('GENIL*',0.0)
      NMAX = 400
      IPRINT=1
C
C
C          READ DATA SEQUENTIAL ORDERING AND
C          PROPER FORMATTING ARE IMPORTANT.
C
      READ (5,70) CTITLE
 70   FORMAT (20A4)
      WRITE (6,71) CTITLE,
 71   FORMAT (1H0,10X,20A4)
C
C
C        NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C
      READ (5,*) NL
      WRITE(6,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR PARAMETERS'//(I3))
C
C
C        L IS THE NUMBER OF LINEAR PARAMETERS
C
C
      L=NL/2
C
C
C        ESTIMATES OF THE NONLINEAR PARAMETERS
C
C
      READ (5,*) (DIM(I),OUT(I),I=1,L)
      DO 80 I=1,L
```

```
      II=2*I-1
      ALF(II)=DIM(I)
 80   ALF(II+1)=1./OUT(I)
      WRITE(6,21)(ALF(I),I=1,NL)
 21   FORMAT(1H0,10X,'INITIAL EST. OF NONLIN. PARAMETERS'//(F7.3))
      WRITE (6,20) (DIM(I),OUT(I),I=1,L)
 20   FORMAT (/,'0 DIMENSIONLESS NUMBER    TRACER ARRIVAL TIME',/,
     # (5X,F9.5,22X,F7.3))
C
C
      LPP2=L+NL+2
C
C
C
C      N IS THE NUMBER OF OBSERVATIONS
C
C
      READ(5,*) N
      WRITE(6,35) N
 35   FORMAT(/1H0,10X,'NUMBER OF OBSERVATIONS'//(I4))
C
C
C      IV IS THE NUMBER OF INDEPENDENT VARIABLES T
C
C
      IV=1
C
C
C     T IS THE INDEPENDENT VARIABLE
C     Y IS THE N-VECTOR OF OBSERVATIONS
C
C
      READ(5,*)(T(I),Y(I),I=1,N)
     WRITE(6,60)(T(I),Y(I),I=1,N)
 60  FORMAT(1H0,'INDEPENDENT VARIABLES       DEPENDENT VARIABLES'//
     * ,(5X,F8.3,21X,F9.3))
C
C
C     W(I) ARE THE WEIGHTING PARAMETERS
C
C
        DO 1 I=1,N
 1       W(I)=1.0
C
C
      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
      WRITE (6,13)
      LP1=L+1
      CALL ADA (LP1,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,2)
      DO 8 I=1,N
      C(I,LP1)=0.
      DO 9 J=1,L
      C(I,J)=BETA(J)*A(I,J)
 9    C(I,LP1)=C(I,LP1)+C(I,J)
      WRITE (6,14) Y(I),C(I,LP1),(C(I,J),J=1,L)
      CY(I)=Y(I)
      CT(I)=T(I)
 8    CONTINUE
```

```
 13     FORMAT(1H0,'  ACTUAL      CALC     COMP#1     COMP#2',//)
 14     FORMAT (1X,8F10.4)
C
        DO 22 I=1,L
        II=2*I-1
        DIM(I)=ALF(II)
 22     OUT(I)=1./ALF(II+1)
        SUM=0.
        DO 25 J=1,L
 25     SUM=SUM+BETA(J)
       DO 93 I=1,L
 93    BETA(I)=BETA(I)/SUM
        WRITE (6,38) (BETA(I),DIM(I),OUT(I),I=1,L)
 38     FORMAT (/,'0 FRACTION    DIMENSIONLESS NUMBER    ARRIVAL TIME',
     # /,(5X,F7.3,5X,F7.3,22X,F7.3))
        IF (IPLOT.NE.1) STOP
        CALL GRAPHG ('*',0,N,CT,CY,4,'TIME (DAYS)*',
     * 'CONCENTRATION (C/S)*',CTITLE)
        CALL LINESG ('SOLD,VBRT*',N,CT,C(1,LP1))
        CALL EXITG
         STOP
         END
C
C
C ****************************************************************
C ****************************************************************
C
C                    SUBROUTINES
C
C ****************************************************************
C ****************************************************************
C
C
C
C
        SUBROUTINE ADA (LP,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION ALF(NL),A(NMAX,LPP2),T(NMAX),INC(14,8),D(400,7)
C
C
C
        L=LP-1
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
        IF(ISEL.EQ.2) GO TO 90
        IF(ISEL.EQ.3)  GO TO 165
        DO 1 J=1,L
        DO 1 K=1,NL
        INC(K,J)=0.0
        IF ((K+1)/2.EQ.J) INC(K,J)=1.0
 1      CONTINUE
C
C
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
  90         DO 81 I=1,N
      DO 81 J=1,L
      K1=2*J-1
      K2=2*J
      IF (ALF(K2)*T(I).GT.1.0) GO TO 82
      A(I,J)=0.
      D(I,J)=0.
      GO TO 81
82    A(I,J)=ALF(K1)*ALF(K2)/(1.772453851*
     #  (ALF(K2)*T(I)-1.0)**(1.5))*
     #  DEXP(-1.*ALF(K1)**2/(ALF(K2)*T(I)-1.0))
       D(I,J)=A(I,J)
C
  81        CONTINUE
C
C
C
        IF (ISEL.EQ.2)  GO TO 200
C
C
C
C
C
C
C
 165      DO 170 I=1,N
C
C
      DO 170 J=1,NL
      K1=(J+1)/2
      K2=2*K1
      K3=K2-1
      JJ=L+J+1
      IF (ALF(K2)*T(I).GT.1.0) GO TO 171
      A(I,JJ)=0.
      GO TO 170
 171  IF ((J/2)*2.EQ.J) GO TO 300
      A(I,JJ)=D(I,K1)*(1.0/ALF(K3) -
     #  2.0*ALF(K3)/(ALF(K2)*T(I)-1.0))
      GO TO 170
300   A(I,JJ)=D(I,K1)*(1.0/ALF(K2)-1.5/(ALF(K2)*T(I)-1.0) *
     #  T(I)+(ALF(K3)**2)*T(I)/((ALF(K2)*T(I)-1.)**2))
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
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 FIX 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 PRDBLEM IS ALSO REFERRED TO AS 'MULTIPLE
C     NONLINEAR REGRESSION'.  FOR USE IN STATISTICAL ESTIMATION,
C     VARPRO RETURNS THE RESIDUALS, THE COVARIANCE MATRIX OF THE
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 EXCEED 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 = 0.
C                 CAUTION:  THE DECLARED ROW DIMENSION OF INC (IN ADA)
C                 MUST CURRENTLY BE SET TO 12.  SEE 'RESTRICTIONS' BELOW.
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 CONTAIN5 THE PHI(J)'S AND THEIR DERIVATIVES (SEE
C                 BELOW).  ON OUTPUT, THE FIRST L+NL ROWS AND COLUMNS OF
C                 A WILL C'ONTAIN 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 THE
C                 NONLINEAR ONES), COLUMN L+NL+1 WILL CONTAIN THE
C                 WEIGHTED RESIDUALS (Y - ETA), A(1, L+NL+2) WILL CONTAIN
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 If
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 ERRORS.
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 MATRIX IMPROPERLY SPECIFIED, OR P DISAGREES
C                     WITH LPP2.
C                 -6  A WEIGHT WAS NEGATIVE.
C                 -7  'CONISTANT' COLUMN WAS COMPUTED MORE THAN ONCE.
C                 -8  CATASTROPHIC FAILURE - A COLUMN OF THE A MATRIX HAS
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, ORFAC2, BACSUB, 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 THEIR
C         PARTIAL DERIVATIVES D PHI(J)/D ALF(K), AT THE SAMPLE POINTS
C         T(I).  THIS ROUTINE MUST BE DECLARED 'EXTERNAL' IN THE CALLING
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 APPEAR
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 COMPUTE 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, AND
C         DERIVATIVES ARE NOT NEEDED.)
C
C      RESTRICTIONS
C
C            THE SUBROUTINES DPA, INIT (AND ADA) CONTAIN THE LOCALLY
C         DIMENSIONED MATRIX INC, WHOSE DIMENSIONS ARE CURRENTLY SET FOR
C         MAXIMA OF L+1 = 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 PROGRAM
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 MATRIX
C         A MUST BE LINEARLY INDEPENDENT.
C
C            OPTIONAL NOTE:  AS WILL BE NOTED FROM THE SAMPLE SUBPROGRAM
C         ADA, THE 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 EXPENSE 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, AND
C         AFTER THE COMPUTATION OF THE PHI'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 THE
C         DATA STATEMENTS IN VARPRO, DPA, POSTPR, AND VARERR.  THE
C         PROGRAM HAS BEEN CHECKED FOR PORTABILITY BY THE BELL LABS PFORT
C         VERIFIER.  FOR MACHINES WITHOUT DOUBLE PRECISION HARDWARE, 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 IN
C         VARPRO, (C) DSIGN, DSQRT AND DABS TO SIGN, SQRT AND ABS,
C         RESPECTIVELY, AND (D) DEXP TO EXP IN THE SAMPLE PROGRAMS 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 THE 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 VARIANCES 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 AXES 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 PARTIAL DERIVATIVES.  IF THESE
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 CAN
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-INVERBES 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. DRAPER, 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, EPS1, GNSTEP, NU, PRJRES, R, RNEW, XNORM
      INTEGER B1, OUTPUT
      LOGICAL SKIP
      EXTERNAL ADA
      DATA EPS1 /1.D-6/, ITMAX /50/, OUTPUT /6/
C
C           THE FOLLOWING TWO PARAMETERS ARE USED IN THE CONVERGENCE
C           TEST:  EPS1 IS AN ABSOLUTE AND RELATIVE TOLERANCE FOR THE
C           NORM OF THE PROJECTION OF THE RESIDUAL ONTO THE RANGE OF THE
C           JACOBIAN OF THE VARIABLE PROJECTION FUNCTIONAL.
C           ITMAX IS THE MAXIMUM NUMBER OF FUNCTION AND DERIVATIVE
C           EVALUATIONS ALLOWED.  CAUTION:  EPS1 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
      NLP1 = 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
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 NEM
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
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 PARAMETEI
   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
   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, I5, 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 RESIDUAL
C         PRODUCED BY DPA, R2 IS THE TRANSFORMED RESIDUAL FROM DPA, AND
C         DR' IS IN UPPER TRIANGULAR FORM (AS IN REF. (2), P. 18).
C         DR IS STORED IN ROWS L+1 TO N AND COLUMNS L+2 TO L + NL + 1 OF
C         THE MATRIX A (I.E., COLUMNS 1 TO NL OF THE MATRIX 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, ALPHAS B(NMAX, NLP1), BETA, DSIGN, PRJRES,
     X U, XNORM
C
      NL = NLPl - 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 TRXANGULAR FORM AND TRANSFORMED RESIDUAL, FOR USE
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, NLPl
            JSUB = NLPl + J
            B(K, J) = B(LPK, J)
   40       B(JSUB, K) = B(LPK, J)
   50    B(NL23, K) = XNORM(K, B(LP1, K))
C
   99 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 DERIVATIVE
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(14, 8)
      LOGICAL NOWATE, PHILPl
      EXTERNAL ADA
C
      IF (ISEL .NE. 1) GO TO 3
         LP1 = L + 1
         LNL2 = L + 2 + NL
         LP2 = L + 2
         LPPl = LPP2 - 1
         FIRSTC = 1
         LASTC = LPPl
         FIRSTR = LP1
         CALL INIT(L, NL, N, NMAX, LPP2, IV, T, W, ALF, ADA, ISEL,
     X   IPRINT, A, INC, )ICON, NCONPl, 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 = LPPl
      FIRSTR = (4 - ISEL)*L + 1
      GO TO 50
C                                                    ISEL = 2
    6 FIRSTC = NCONPl
      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), GETTING
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, Rn 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 66
         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 66
         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 TRIANGULAR
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.  COMPUTE
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*BETA
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 R2 = Q2*Y (IN COL. L+1) IS COPIED
C           TO COLUMN L+NL+2.  OTHERWISE ALL OF COLUMN L+1 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 = NCONP1, 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 = ACUM + A(I, M)
   90       A(I, KSUB) = 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(14, 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. LNL2 .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 99
    9    W(I) = DSQRT(W(I))
```

```
C
      NCONP1 = L
      NCONP1 = LP1
      PHILP1 = L .EQ. 0
      IF (PHILPI .OR. NL .EQ. 0) GO TO 99
C                                       CHECK INC MATRIX FOR VALID INPUT AND
C                                       DETERMINE NUMBER OF CONSTANT FCNS.
      P = 0
      DO 11 J = 1, LP1
         IF (P .EQ. 0) NCONP1 = J
         DO 11 K = 1, NL
            INCKJ = 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 = NCONP1 - 1
      IF (IPRINT .GE. 0) WRITE (OUTPUT, 210) NCON
      IF (L+P+2 .EQ. LPP2) GO TO 20
C                                                  INPUT ERROR IN INC MATRIX
   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) PHILP1 = .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
      LPNL = 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 MATRIX
   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
      WRITE (OUTPUT, 209)
      IF (L .GT. 0) WRITE (OUTPUT, 210) (U(J), J = 1, L)
      IF (NL .GT. 0) WRIT'E (OUTPUT, 211) (ALF(K), K = 1, NL)
      WRITE (OUTPUT, 214) RNORM, SAVE, ACUM
      IF (DABS(SAVE) .GT. EPS) WRITE (OUTPUT, 215)
      WRITE (OUTPUT, 209)
   99 RETURN
C
  209 FORMAT (1H0, 50(1H'))
  210 FORMAT (20H0  LINEAR PARAMETERS // (7E15.7))
```

```
  211 FORMAT (23H0  NONLINEAR PARAMETERS // (7E15.7))
  214 FORMAT (21H0  NORM OF RESIDUAL =, E15.7, 33H EXPECTED ERROR OF OBS
     XERVATIONS =, E15.7, / 39H   ESTIMATED VARIANCE OF OBSERVATIONS =,
     X E15.7 )
  215 FORMAT (95H  WARNING -- EXPECTED ERROR OF OBSERVATIONS IS NOT ZERO
     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 (/ SOH INPUT ERROR IN PARAMETER L, NL, N, LPP2, OR NMAX. /)
  105 FORMAT (68H0  ERROR -- INC MATRIX IMPROPERLY SPECIFIED, OR DISAGRE
     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, SE
     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
//LKED.SYSLMOD DD DSN=WYL.JE.CLJ.SETH(MELISSA),DISP=OLD
```

WAIRAKEI (3/79) - CWK24 FROM WK107*

NUMBER OF NONLINEAR PARAMETERS

2

INITIAL EST. OF NONLIN. PARAMETERS

2.000
5.000

DIMENSIONLESS NUMBER	TRACER ARRIVAL TIME
2.00000	0.200

NUMBER OF OBSERVATIONS

93

INDEPENDENT VARIABLES	DEPENDENT VARIABLES
0.214	28.510
0.297	2043.906
0.380	7757.337
0.464	10865.406
0.547	10752.924
0.630	9576.211
0.714	8226.813
0.797	7012.052
0.880	5984.576
0.964	5198.999
1.047	4588.288
1.130	4092.422
1.297	3386.888
1.380	2895.055
1.464	2727.387
1.547	2606.242
1.630	2446.038
1.714	2321.840
1.797	2194.641
1.880	2078.807
1.964	1973.313
2.047	1890.512
2.130	1792.964
2.214	1615.511
2.630	1280.402
2.714	1232.835
2.797	1187.192
2.880	1145.455
2.964	1111.014
3.047	1079.655
3.130	1044.407
3.214	1002.250
3.297	944.850
3.380	909.742
3.464	879.628
3.547	848.368
3.630	817.099
3.714	792.777
3.797	769.525
3.880	748.345
3.964	751.802
4.047	714.117

4.130	695.975
4.214	658.868
4.297	638.992
4.380	618.845
4.464	607.310
4.630	572.912
4.714	559.063
4.797	543.320
4.880	532.194
4.964	517.672
5.047	502.855
5.130	491.666
5.214	483.097
5.297	469.851
5.380	462.071
5.464	450.761
5.547	446.069
5.630	432.668
5.880	407.597
5.964	397.611
6.047	392.204
6.130	380.086
6.214	375.645
6.297	369.281
6.380	359.749
6.630	345.746
6.714	340.128
6.797	338.155
6.880	338.703
6.964	333.483
7.047	331.370
7.130	317.503
7.214	305.410
7.297	300.365
7.380	295.185
7.464	292.692
7.547	287.393
7.630	284.710
7.714	279.273
7.797	275.762
7.880	271.511
8.130	263.995
8.380	256.012
8.464	249.844
8.547	247.240
8.630	244.606
8.880	236.365
8.964	234.301
9.047	229.099
9.130	225.334
9.214	223.842

NUMBER OF CONSTANT FUNCTIONS = 0

0 NORM OF RESIDUAL = 0.1280073D+05
NU = 0.1000000D+01
ITERATION 1 NONLINEAR PARAMETERS
0.1951013D+01 0.7306868D+01
1 NORM OF RESIDUAL = 0.5307679D+04
NU = 0.5000000D+00
NORM(DELTA-ALF) / NORM(ALF) = 0.305D+00

```
ITERATION   2    NONLINEAR PARAMETERS
0.1790861D+01  0.6391822D+01
  1  NORM OF RESIDUAL =  0.4667214D+04
  NU =  0.2500000D+00
  NORM(DELTA-ALF) / NORM(ALF) =   0.140D+00
ITERATION   3    NONLINEAR PARAMETERS
0.1397078D+01  0.4619397D+01
  1  NORM OF RESIDUAL =  0.2752487D+04
  NU =  0.1250000D+00
  NORM(DELTA-ALF) / NORM(ALF) =   0.376D+00
ITERATION   4    NONLINEAR PARAMETERS
0.1242511D+01  0.4320153D+01
  1  NORM OF RESIDUAL =  0.1722342D+04
  NU =  0.6250000D-01
  NORM(DELTA-ALF) / NORM(ALF) =   0.749D-01
ITERATION   5    NONLINEAR PARAMETERS
0.1249233D+01  0.4327062D+01
  1  NORM OF RESIDUAL =  0.1716735D+04
  NU =  0.3125000D-01
  NORM(DELTA-ALF) / NORM(ALF) =   0.214D-02
ITERATION   6    NONLINEAR PARAMETERS
0.1248118D+01  0.4323440D+01
  1  NORM OF RESIDUAL =  0.1716674D+04
  NU =  0.1562500D-01
  NORM(DELTA-ALF) / NORM(ALF) =   0.842D-03
ITERATION   7    NONLINEAR PARAMETERS
0.1248064D+01  0.4323031D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.7812500D-02
  NORM(DELTA-ALF) / NORM(ALF) =   0.916D-04
ITERATION   8    NONLINEAR PARAMETERS
0.1248037D+01  0.4322911D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.3906250D-02
  NORM(DELTA-ALF) / NORM(ALF) =   0.272D-04
ITERATION   9    NONLINEAR PARAMETERS
0.1248032D+01  0.4322888D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.1953125D-02
  NORM(DELTA-ALF) / NORM(ALF) =   0.530D-05
ITERATION  10    NONLINEAR PARAMETERS
0.1248031D+01  0.4322883D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.9765625D-03
  NORM(DELTA-ALF) / NORM(ALF) =   0.121D-05
ITERATION  11    NONLINEAR PARAMETERS
0.1248031D+01  0.4322882D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.4882812D-03
  NORM(DELTA-ALF) / NORM(ALF) =   0.258D-06
ITERATION  12    NONLINEAR PARAMETERS
0.1248031D+01  0.4322881D+01
  1  NORM OF RESIDUAL =  0.1716672D+04
  NU =  0.2441406D-03
  NORM(DELTA-ALF) / NORM(ALF) =   0.564D-07
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
LINEAR PARAMETERS

0.1655775D+05
NONLINEAR PARAMETERS
```

0.1248031D+01 0.4322881D+01

NORM OF RESIDUAL = 0.1716672D+04 EXPECTED ERROR OF OBSERVATIONS = 0.

ESTIMATED VARIANCE OF OBSERVATIONS = 0.3274403D+05

WARNING -- EXPECTED ERROR OF OBSERVATIONS IS NOT ZERO. COVARIANCE MATR

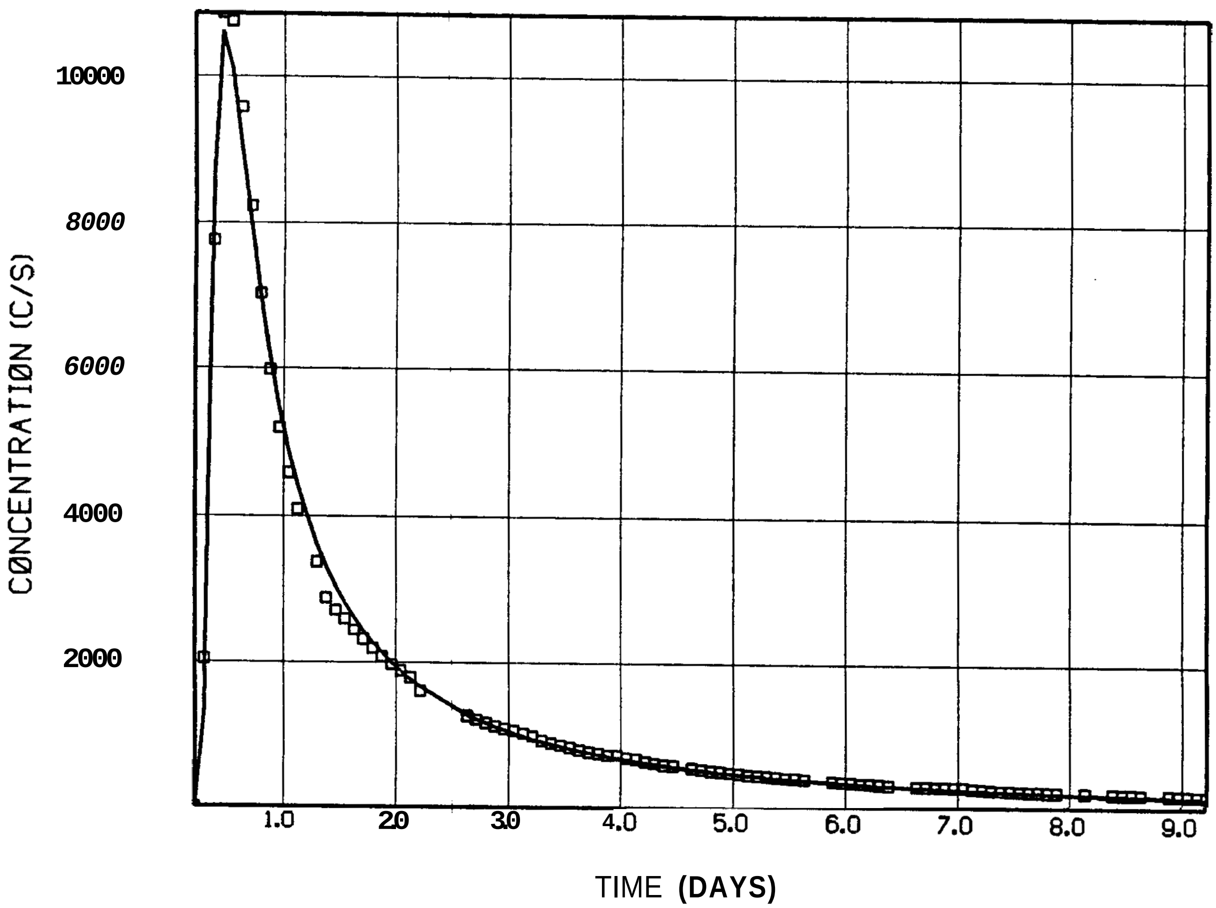
''

ACTUAL	CALC	COMP#1	COMP#2
28.5100	0.0	0.0	
2043.9060	1380.2258	1380.2258	
7757.3370	8667.5898	8667.5898	
10865.4060	10619.6406	10619.6406	
10752.9240	10097.2422	10097.2422	
9576.2110	9022.6719	9022,6719	
8226.8130	7926.6328	7926,6328	
7012.0520	6970.8984	6970.8984	
5984.5760	6158.7656	6158,7656	
5198.9990	5467.9453	5467.9453	
4588.2880	4893.8203	4893,8203	
4092.4220	4407.9609	4407,9609	
3386.8880	3634.8909	3634,8909	
2895.0550	3328.4558	3328,4558	
2727.3870	3058.7834	3058.7834	
2606.2420	2825.5759	2825,5759	
2446.0380	2620.0679	2620,0679	
2321.8400	2435.8870	2435.8870	
2194.6410	2273.8853	2273.8853	
2078.8070	2128.8738	2128,8738	
1973.3130	1997.0005	1997.0005	
1890.5120	1879.4253	1879,4253	
1792.9640	1772.8572	1772.8572	
1615.5110	1674.8044	1674.8044	
1280.4020	1298.8887	1298.8887	
1232.8350	1239.8137	1239.8137	
1187.1920	1185.6943	1185.6943	
1145.4550	1135.3857	1135.3857	
1111.0140	1087.9790	1087.9790	
1079.6550	1044.2793	1044.2793	
1044.4070	1003.4192	1003.4192	
1002.2500	964.7019	964.7019	
944.8500	928.8235	928.8235	
909.7420	895.1094	895.1094	
879.6280	863.0107	863.0107	
848.3680	833.1306	833.1306	
817.0990	804.9316	804.9316	
792.7770	777.9731	777.9731	
769.5250	752.7786	752.7786	
748.3450	728.9126	728.9126	
751.8020	706.0134	706.0134	
714.1170	684.5388	684.5388	
695.9750	664.1294	664.1294	
658.8680	644.4846	644.4846	
638.9920	626.0051	6261.0051	
618.8450	608.3909	608.3909	
607.3100	591.3887	591.3887	
572.9120	560.0254	560.0254	
559.0630	545.1943	545.1943	
543.3200	531.1709	531.1709	

532. 1940	517.7378	517.7378
517.6720	504.7092	504.7092
502.8550	492.3628	492.3628
491.6660	480.5112	480.5112
483.0970	468.9929	468.9929
469.8510	458.0562	458.0562
462.0710	447.5374	447.5374
450.7610	437.2954	437.2954
446.0690	427.5527	427.5527
432.6680	418.1665	418.1665
407.5970	391.8708	391.8708
397.6110	383.6477	383.6477
392.2040	375.8010	375.8010
380.0860	368.2180	368.2180
375.6450	360.7998	360.7998
369.2810	353.7109	353.7109
359.7490	346.8511	346.851 1
345.7460	327.4724	327.4724
340.1280	321.3625	321.3625
338.1550	315.5098	315.5098
338.7030	309.8323	309.8323
333.4830	304.2576	304.2576
331.3700	298.9116	298.9116
317.5030	293.7202	293.7202
305.4100	288.6172	288.6172
300.3650	283.7185	283.7185
295.1850	278.9568	278.9568
292.6920	274.2720	274.2720
287.3930	269.7700	269.7700
284.7100	265.3896	265.3896
279.2730	261.0762	261.0762
275.7620	256.9275	256.9275
271.5110	252.8874	252.8874
263.9950	241.3363	241.3363
256.0120	230.6383	230.6383
249.8440	227.2199	227.2199
247.2400	223.9243	223.9243
244.6060	220.7076	220.7076
236.3650	211.4695	211.4695
234.3010	208.5092	208.5092
229.0990	205.6513	205.6513
225.3340	202.8581	202.8581
223.8420	200.0950	200.0950

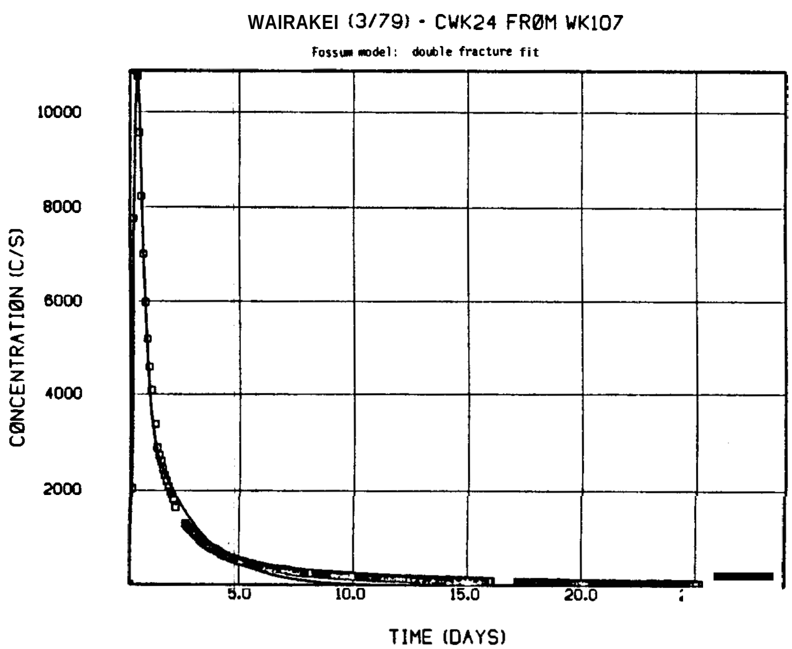
FRACTION	DIMENSIONLESS NUMBER	ARRIVAL TIME
1.000	1.248	0.231

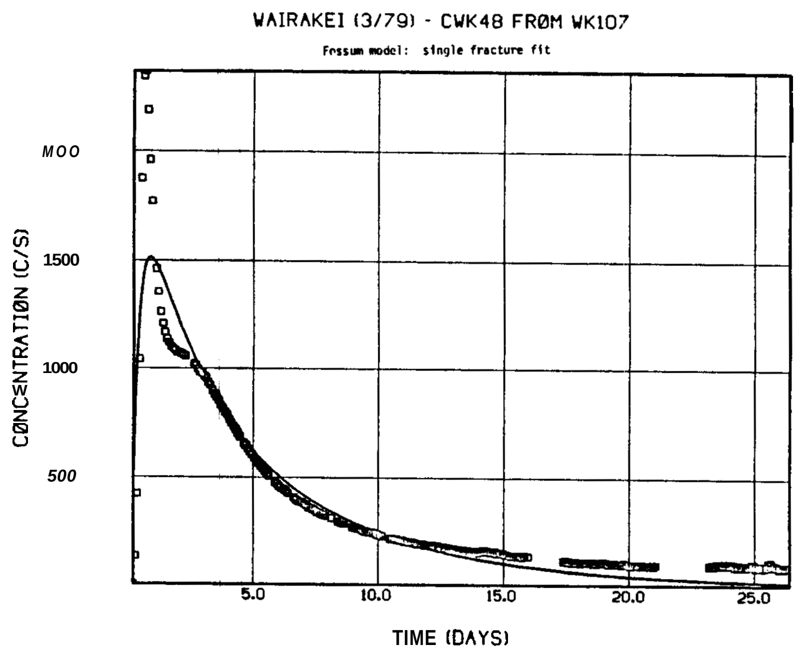
WAIRAKEI (3/79) - CWK24 FRBM WK107



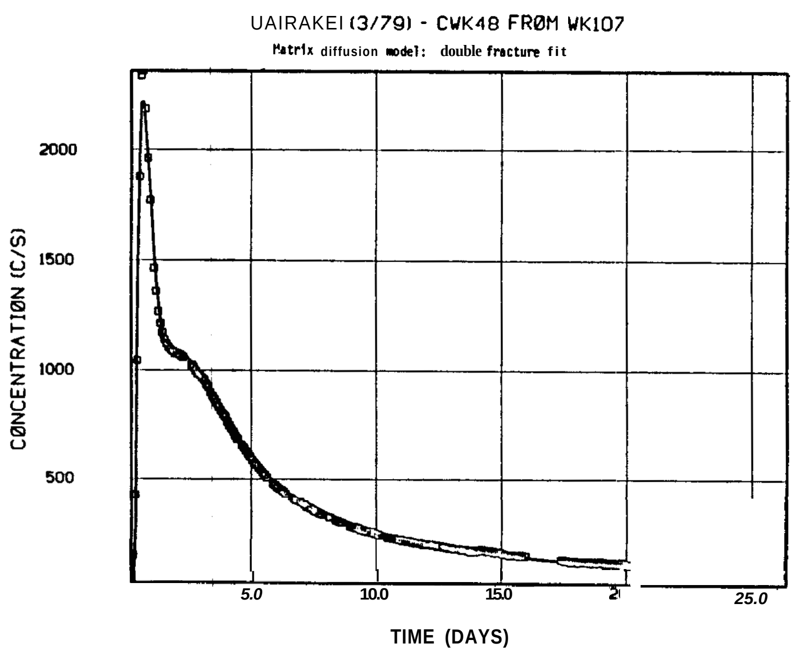
Appendix C

Fitted Tracer Return Profiles

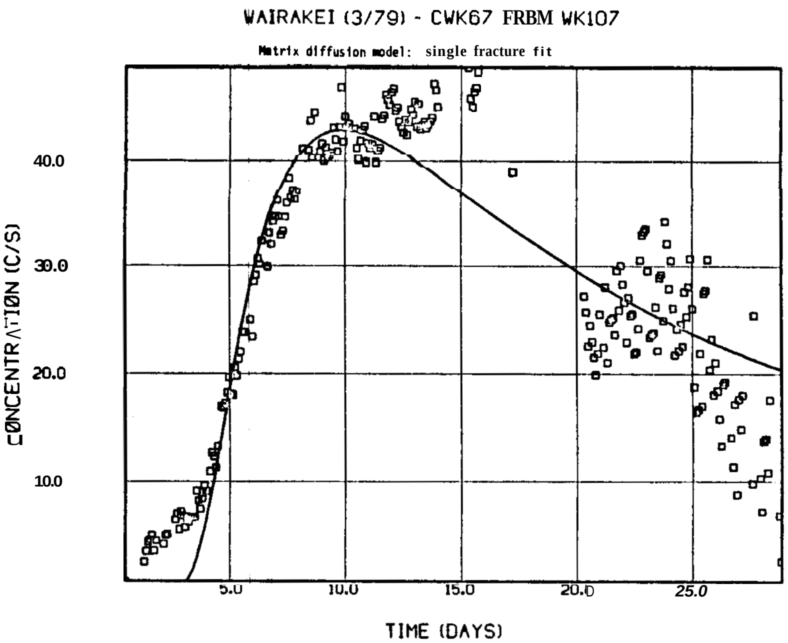

WAIRAKEI (3/79) - CWK24 FRØM WK107
Fossum model: double fracture fit
10000
8000
6000
4000
2000
CØNCENTRATIØN (C/S)
5.0
10.0
15.0
20.0
TIME (DAYS)



WAIRAKEI (3/79) - CWK48 FRØM WK107
Fossum model: single fracture fit
MOO
1500
1000
500
CØNC≡NTRATIØN (C/S)
5.0
10.0
15.0
20.0
25.0
TIME (DAYS)



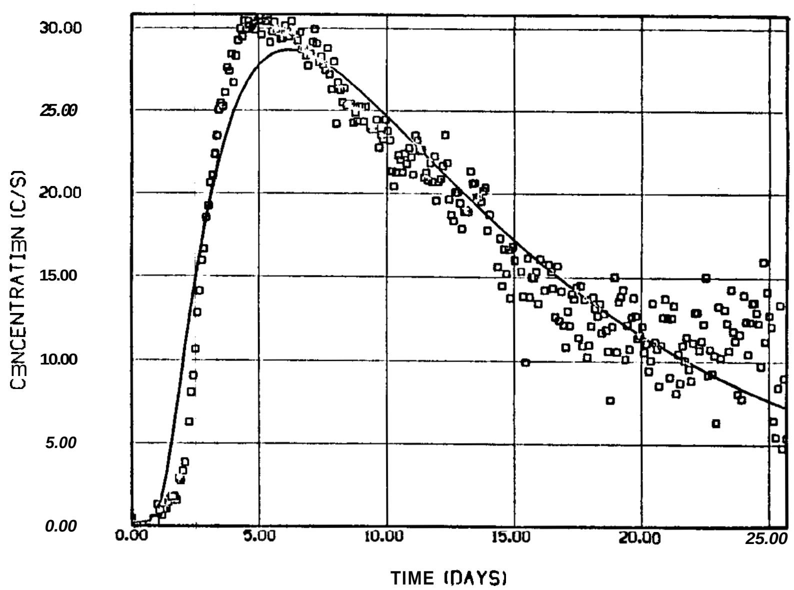
UAIRAKEI (3/79) - CWK48 FRØM WK107
Matrix diffusion model: double fracture fit
2000
1500
1000
500
CØNCENTRATIØN (C/S)
5.0
10.0
15.0
25.0
TIME (DAYS)



WAIRAKEI (3/79) - CWK67 FRBM WK107
Matrix diffusion model: single fracture fit
CØNCENTRATIØN (C/S)
40.0
30.0
20.0
10.0
5.0
10.0
15.0
20.0
25.0
TIME (DAYS)

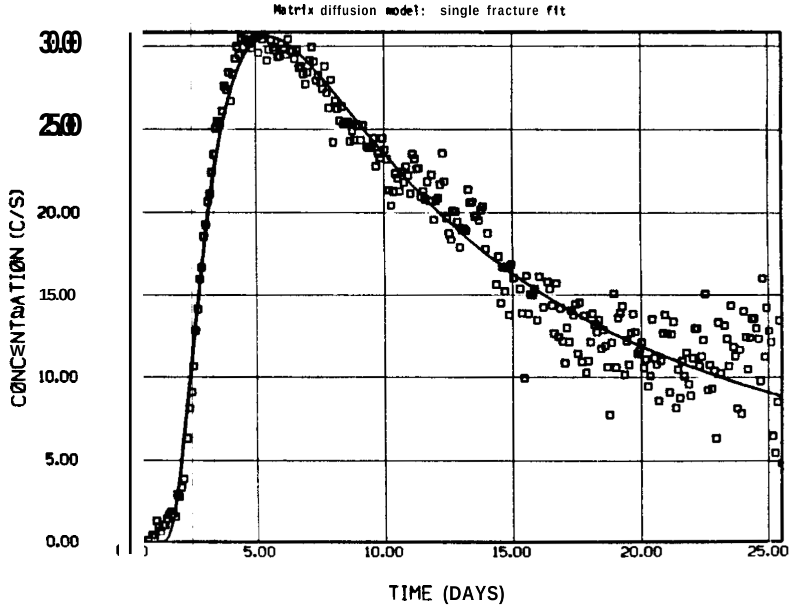
WAIRAKEI (7/79) - CWK103 FRØM WK101

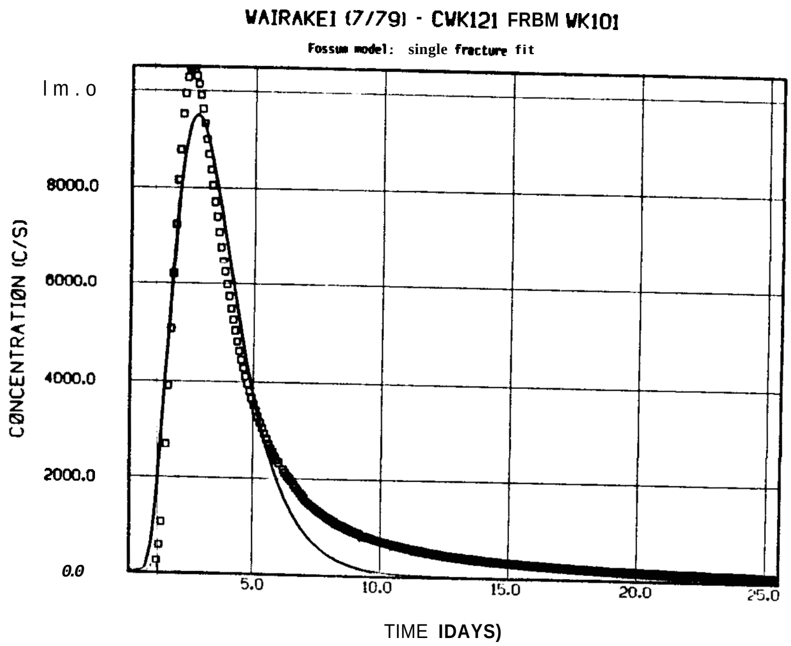
Fossum model: single fracture fit



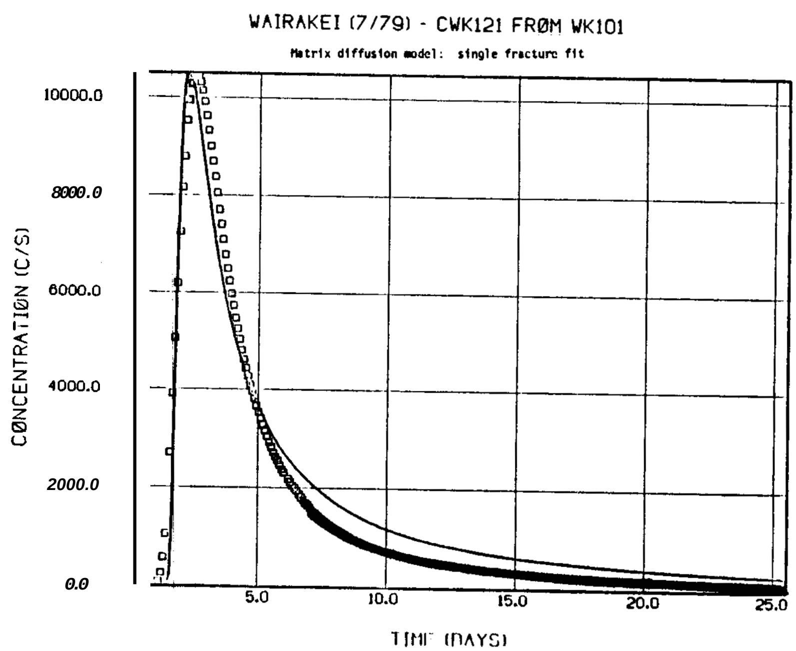
VAIRAKEI (7/79) - CWK103 FRBM WK101

Matrix diffusion model: single fracture fit





WAIRAKEI (7/79) - CWK121 FRBM WK101
Fossum model: single fracture fit
CONCENTRATION (C/S)
0.0
2000.0
4000.0
6000.0
8000.0
5.0
10.0
15.0
20.0
25.0
TIME (DAYS)



WAIRAKEI (7/79) - CWK121 FRØM WK101
Matrix diffusion model: single fracture fit
CØNCENTRATIØN (C/S)
0.0
2000.0
4000.0
6000.0
8000.0
10000.0
5.0
10.0
15.0
20.0
25.0