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

CHARACTERIZATION OF RETENTION PROCESSES AND THEIR EFFECT ON THE ANALYSIS OF TRACER TESTS IN FRACTURED RESERVOIRS

By

Gardner William Walkup Jr.

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

June 1984

ACKNOWLEDGEMENTS

The United States Department of Energy provided funds for this study under contract number DE-AT03-80SF-11459. The data used in this work was made available by the Institute of Nuclear Sciences, Department of Industrial and Scientific Research New Zealand. I would like to thank my research advisor Dr. Roland N. Horne for his support and guidance in all phases of this study. And finally I would like to thank Patti Juell Walkup for her support, encouragement, and countless proof readings of this text.

ABSTRACT

Retention processes such as adsorption and diffusion into an immobile region can effect tracer movement through a fractured reservoir. This study has conducted experimental work and has developed a two-dimensional model to characterize retention processes. A method to directly determine some important flow parameters, such as the fracture aperture, from the analysis of tracer tests has been developed as a result of the new two-dimensional model.

The experimental work consisted of batch experiments designed to both reproduce earlier work and to determine the magnitude of the retention effects. Negligible retention was observed from which it was concluded that the batch experiments were not sensitive enough and that more sensitive flowing tests were needed.

A two-dimensional model that represents a fractured medium by a mobile region, in which convection, diffusion, and adsorption are allowed, and an immobile region in which only diffusion and adsorption are allowed has been developed. It was possible to demonstrate how each of the mass-transfer processes included in the model affect tracer return curves by producing return curves for any set of the defining variables.

Field data from the New Zealand was numerically fit with the model. The optimum values of the parameters determined from curve fitting provided a direct estimate of the fracture width and could be used to estimate other important flow parameters if experimentally determinable values were known.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section 1: INTRODUCTION

Tracers have long been used by petroleum reservoir engineers to gain information on reservoir heterogeneities, but have recently gained importance to geothermal engineers because of the problem of waste water reinjection. In most geothermal utilizations, only steam is used to drive the turbines and any produced water as well as a smaller amount of steam condensate must be disposed of. This waste water is at high temperature and has environmentally hazardous levels of dissolved materials and is usually reinjected since surface disposal of these waters is no longer an acceptable procedure in most places.

The reinjection of waste water can serve a second purpose other than disposal by maintaining reservoir pressure and mass of fluid in place. However these possible benefits must be related to the potentially damaging effects that the cooler (than reservoir fluids) injected water will have on the reservoir. If the injected water travels to the production well so quickly that it does not heat up to the original reservoir temperature, it will reduce the enthalpy of the produced water. This results in a smaller steam fraction in the produced fluid, and a smaller flow rate for a given wellhead pressure since the flow of the wells is strongly governed by the hydrostatic pressure of the fluid column. Thus less energy can be produced. Such "short-circuiting" has been observed in several geothermal fields.[1]

It is the task of the reservoir engineer to determine how the waste water should be reinjected so that the harmful effects of the cooler water is minimized. Tracers have proven useful for this task. By injecting tracers and observing their returns at production wells, one can get an idea how the injected water travels through the reservoir. Such tests have shown some unexpected results. In Japan, tracer tests have recorded mean displacement of tracers at a rate as high as 78 m/hr. and as low as 0.5 m/hr. [1] Similar flow rates were observed in New Zealand. [2] It has been demonstrated that there is a correspondence between fast tracer return rates and wells that show enthalpy declines upon reinjection. [1]

While tracer testing has proven useful, the analysis of these tests has been mostly qualitative. In order to predict thermal breakthroughs and enthalpy declines, quantitative data on reservoir flow parameters are needed. Currently there are no methods to directly determine these parameters from tracer tests in geothermal reservoirs. There are two main problems that make the analysis of tracer tests in geothermal reservoirs difficult.

The first problem is that most geothermal reservoirs are highly fractured. Thus the quantitative analysis of tracer testing in porous media, developed for the oil and gas industry, does not apply to geothermal reservoirs.

The second problem is modeling all the processes that can occur to a tracer as it moves through the reservoir. Besides the macroscopic processes of convection and dispersion, such microscopic processes as diffusion, chemical reaction, ion exchange, adsorption and decay can occur which effect the analysis of tracer tests. Quantitative analysis of tracer tests depends on the ability to describe accurately all processes that occur to the tracer as it travels through the reservoir.

In this study, experimental work was conducted to examine transport properties and a two-dimensional model was developed to describe those processes which can effect the analysis of tracer return curves. A method to directly determine some important flow parameters from the analysis of tracer test has been developed as a result of the new two-dimensional model.

Section 2: LITERATURE REVIEW

Strom and Johnson (1950) demonstrated the importance of tracer tests to reservoir engineers by verifying the existence of directional permeability with the use of brine and fluorescein dyes.[3] Many other uses for tracer test were soon found. A fairly complete list of information obtainable from tracer tests has been given by Wagner (1974).[4]

Early analysis of tracer tests tended to ignore the microscopic processes such as diffusion, ion exchange, and adsorption. These early studies only considered convection and dispersion.[5] The corresponding dispersion-convection governing differential equations has been solved for several boundary conditions by Carslaw and Jagger (1959).[6] A summary of the use of such equations and the empirical correlations used to determine the parameters in those equations is given by Perkins and Johnson (1963).[7]

In order to increase correspondence between theoretical and experimental results, other flow processes were considered. Coats and Smith (1964) included diffusion into a stagnant pore volume.[8] A correction to the boundary conditions used in this study was given by Brigham (1974).[9]

The above references do not necessarily assume a porous media but rather develop general flow models. Most further developments in the petroleum literature are limited to porous media and as such are of limited value to understanding tracer flow in highly fractured geothermal reservoirs.

Many additional refinements to the basic dispersion-convection model are found in the ground-water hydrology and soil chemistry literature. The inclusion of adsorption into the model with stagnant pore volume was shown by van Genuchten and Wierenga (1976).[10] Cleary and van Genuchten (1979) showed how also to include decay and chemical reaction in the model.[11]

Recent field experience as described by Horne (1982) and Tester, Bivins, and Potter (1982) demonstrate the need to apply a detailed model to the analysis of tracer tests.[1,12] An experimental study by Breitenbach (1982) showed that considerable retention of chemical tracer possibly occurs with geothermal material (unconsolidated).[13]

Horne and Rodriguez (1983) presented a one-dimensional model for flow in a fracture.[14] This model included convection and diffusion (Taylor Dispersion) within the fracture. Fossum and Horne (1982) applied this model to field data from Wairakei with some success.[15]

Jensen (1983) extended this model by allowing the fracture to communicate by diffusion with a porous matrix.[16] Adsorption was also allowed in both the fracture and the matrix. Jensen applied this model to the same Wairakei data with greater success. While Jensen's model fitted well with the data it revealed only partial information about flow characteristics or reservoir parameters because of the lack of direct measurements of some of the process parameters.

Section 3: EXPERIMENTAL WORK

The goals of the experimental phase of this study were:

(1) To locate the mechanisms of the retention seen in Breitenbach's study.[13]

(2) To determine the magnitude of the retention processes under batch conditions.

A 60° axonometrix view of the apparatus used for this experiment is shown in Figure (1). A schematic of the flow paths is shown in Figure (2). This equipment was designed by A. Sageev[17] and was later modified by Breitenbach.” Detailed discussions of the apparatus and of the subsequent modifications to the apparatus used in this study are given in these earlier studies.

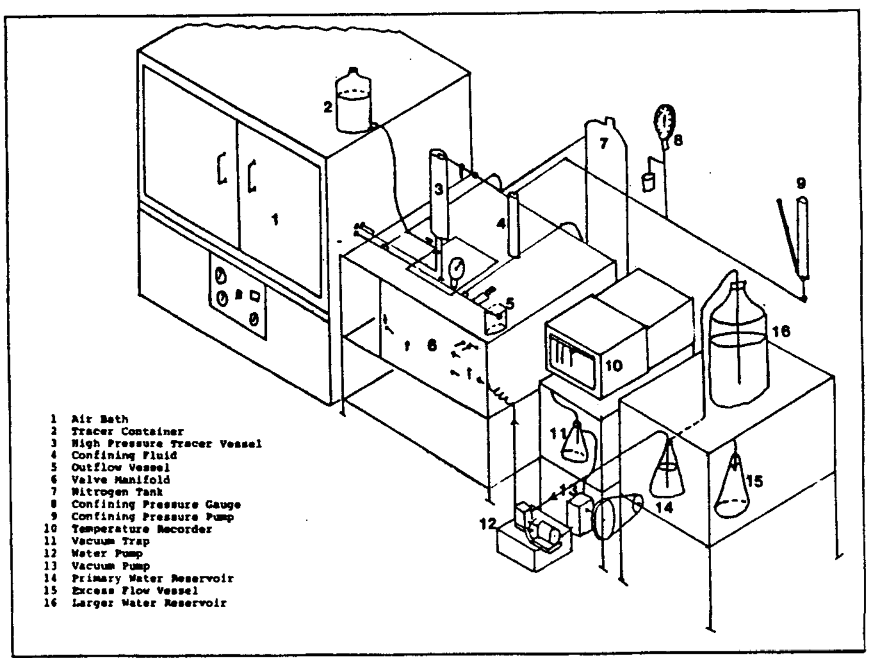

Figure 1 - VIEW OF APPARATUS”

An additional modification for this study was the alteration of the core sleeve. Previous experiments used a viton sleeve to support the unconsolidated core but because of possible interaction between the viton and the chemical

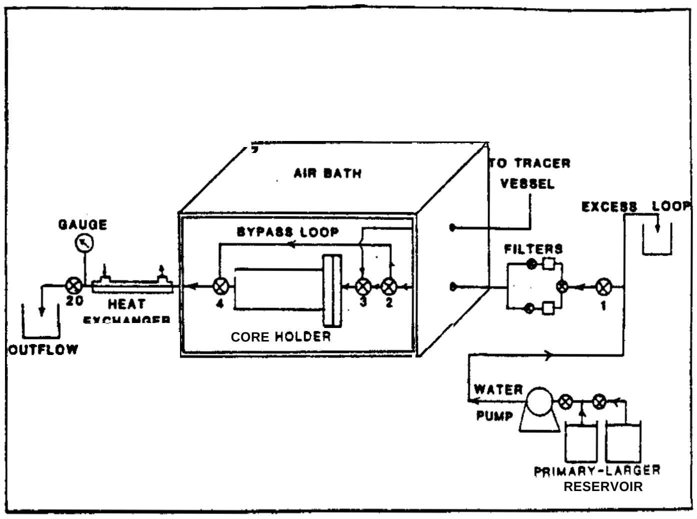

Figure 2 - FLOW PATH OF FLUIDS[17]

tracer, the viton sleeve was replaced by a stainless steel sleeve. The stainless steel sleeve also allowed the apparatus to operate at higher temperatures. It was also necessary to modify the endplugs to hold the new sleeve. The new sleeve and the modified end plugs are shown in Figure **(3)** through (5).

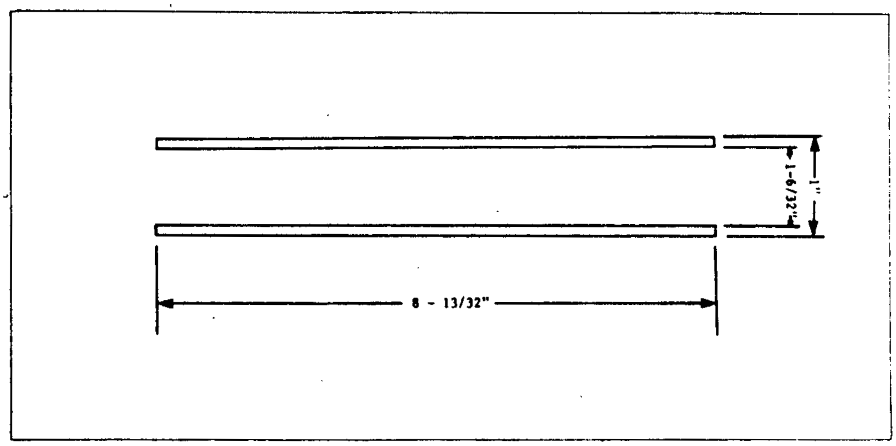

Figure 3 - NEW CORE SLEEVE[17]

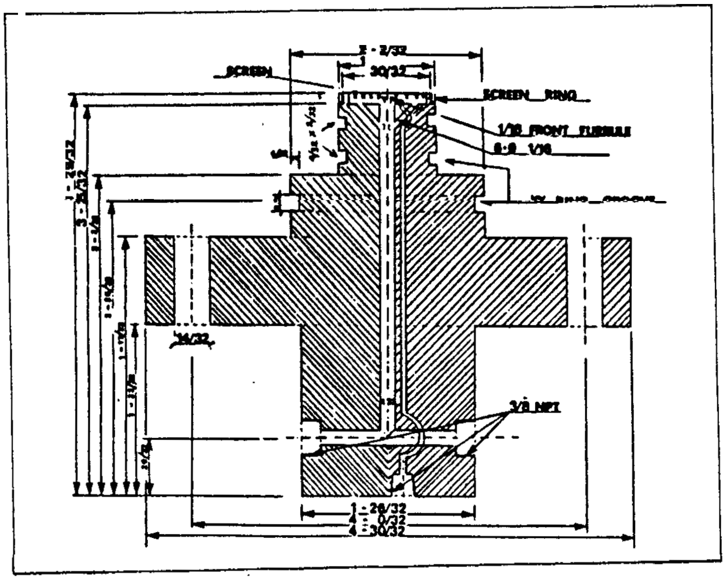

Figure 4 - NEW UPSTREAM CORE END-PLUG"

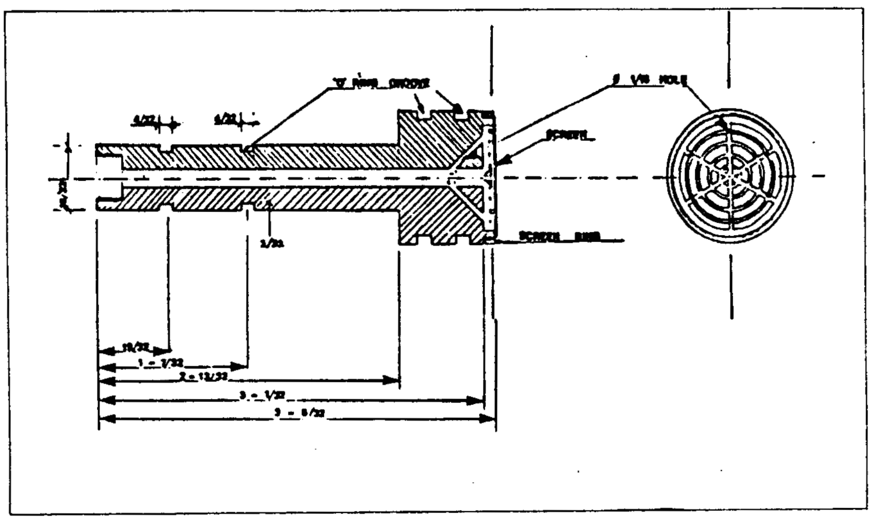

Figure 5 - NEW DOWNSTREAM CORE END-PLUG[19]

The core material used was unconsolidated reservoir rock. The first material used was reservoir rock from Klamath Falls, Oregon and the second was from Los Azufres, Mexico.

The core material from Klamath Falls was comprised of drill cuttings collected from the producing zone (600-660 ft) of a well near the location of a tracer test conducted in May 1983 by the Stanford Geothermal Program''. This material was described by the driller as "black lava". A geological report of the cuttings was done and described the cuttings as fine-grained andesite or basalt with a minimum of alteration. Before the cuttings were loaded into the core holder, they were cleaned, dried, and sieved. A review of the sieve analysis is shown in Table (1).

Table 1- SIEVE ANALYSIS BY PWCENT OF TOTAL MASS

MATERIAL	MESH SIZE					
	<100	100-120	120-140	140-170	170-200	>200
KLAMATH FALLS	77.3	6.1	3.9	3.5	2.6	6.6
LOSAZUFRES	93.6	...	1.1*	0.3	0.6	4.4

*(100-140) mesh

The core material from the Los Azufres field was collected from an outcrop in the field and is described as a typical andesite of the reservoir. In this case the material was crushed, cleaned, dried, and sieved before loading into the core holder. Table (1) also summarizes the sieve analysis for the Los Azufres material.

A detailed step-by-step procedure for this experiment with this equipment is given by Breitenbach.[13]

The general procedure was to first load the core holder with reservoir material. The core holder was then put into the pressure cell and connected with all the flow lines. A vacuum was then applied to the downstream end of the core to remove any air. The core was then brought up to the desired pressure and temperature. After completely flushing the core with distilled water, the core was flooded with approximately three pore volumes of tracer. The tracer used was sodium iodide where the iodide ion was the chemical species traced. The effluent was collected. The cell was then isolated and allowed to sit for the desired residence time. The core was then flushed with six pore volumes of distilled water, and the effluent was again collected.

Determination of the amount of tracer retained in the core was achieved by mass balance calculation. The concentrations of the input and effluents were measured by specific ion electrode analysis, using a Fisher "Accument", Model 750 Selective Ion Analyzer. Description of this analyzer and its use is given by Jackson.[20]

Section 4: RESULTS OF EXPERIMENTAL STUDY

Seven different runs were made. Five runs with the Klamath Falls (KF) core material and two runs with the Los Azufres (LA) core material.

Tracer concentration for all runs was approximately 20ppm. Temperature was varied from room temperature to 300 F. Confining pressure was 1500 psi. Residence times were varied from two hours to 72 hours. Table (2) summarizes all the runs and gives the calculated percent mass of tracer retained.

Table 2 - EXPERIMENTAL RESULTS

RUN	MATERIAL	CONC. (PPM)	TEMP. (F)	RESIDENCE TIME(HR)	PERCENT RETAINED
1	KF	18.0	194	2	-2.23
2	KF	10.1	194	24	6.71
3	KF	15.0	210	60	1.76
4	KF	21.0	300	72	9.09
5	KF	17.2	300	24	-4.65
6	LA	23.2	300	72	-4.76
7	LA	22.6	300	44	4.81

Section 5: DISCUSSION OF EXPERIMENTAL STUDY

Table (2) shows that the calculated percent tracer retained ranged from 9.1 to -4.8 percent. The negative retention values mean that more tracer was calculated coming out than was injected.

An error analysis shows an experimental error to be about 5.0% . The values of percent retained all (but one) fall within 5.0% of no retention at all.

The present results are considerably different from those of Breitenbach [13] study. Results are summarized in Table(3), showing values of percent retained ranging from a low of 17 percent to a high of about 70 percent.

Table 3 - RESULTS FROM BREITENBACH[13]

RUN	CONC. (PPM)	TEMP. (F)	RESIDENCE TIME (HR)	PERCENT RETAINED
4	10	300	72	30.6
5	20	300	72	68.6
6	50	300	72	67.5
7	100	300	72	69.4
8	500	300	72	61.6
10	10	300	24	25.9
12	10	300	2	16.9

It is possible to explain the fact that the present study sees little if any retention, and it is also possible to postulate some explanations for the different results between this and Breitenbach's study.

Since Breitenbach also used outcroppings from the Los Azufres field the difference between the two studies cannot be explained on the basis of different core material. However, Breitenbach used a viton sleeve to hold the core material while the present study used a stainless steel sleeve. The sleeve was changed because it was supposed that the viton might possibly adsorb the iodide tracer. This is the most likely explanation for the differences between the two studies.

Another possible reason for the difference is that the previous study did not use high temperature valves while the present study did. It is not possible to determine if the earlier valves did leak but if they did the differences in results could be explained.

A close examination of the procedure used in these batch experiments suggests some reasons why negligible retention was seen. Using unconsolidated material as designed, the number of mass transfer processes that could result in retention of the chemical tracer are limited. In particular the loss of tracer from a mobile region to an immobile region due to diffusion is not allowed. This is because the entire core must be considered a mobile region.

Other processes which are allowed are those that can be classified as surface retention processes. An example of a surface retention process is adsorption. By isolating this one type of retention process these batch experiments demonstrate that surface retention processes are negligible.

This can be explained by examining the important parameters for surface processes. The first important parameter is surface area. Obviously the more surface available the more surface processes will occur. The unconsolidated material used in this experiment gives more surface area per weight than would be expected under reservoir conditions if flow were occurring in a fracture. Surface retention processes would therefore be be magnified under these experimental conditions. However another effect needs to be considered and that is the relative volume of tracer injected. Three pore volumes of tracer were injected in this experiment while in a field tracer test orders of magnitude less than three pore volumes of tracer are injected. The result is that in the experimental case the number of surface sites available for surface retention processes are overloaded in comparison to that in a realistic case. So even if all surface sites were active in retention the retention seen, using the adopted procedures, would be small.

An obvious solution to this problem would be to inject less tracer. Unfortunately, experimental constraints such as the sensitivity of the analysis technique and the error in the mass balance calculation will not allow for less tracer to be injected.

Another method for experimentally analyzing those mass transfer processes whose net effect is the retention of a tracer is to run flowing experiments. Previous studies have shown that flowing experiments are more sensitive than batch experiments.[21] These earlier studies have been for porous media and thus the analysis of the tracer return curves from these studies is not directly applicable to a fractured media.

While there are models that have attempted to fit field data for a fractured reservoir, these models are not useful in examining the retention processes.[12,15,16] Before an experiment could be designed to run flowing studies, a model was needed in order to examine the magnitudes of various retention processes, be they surface processes or bulk processes (diffusion from mobile to an immobile phase).

This study has developed such a model, the derivation of which is now discussed.

Section 6: THEORETICAL DEVELOPMENT OF TWO-DIMENSIONAL MODEL

The two dimensional, two control volume model used for this development is shown in Figure *(6).* The first control volume represents the mobile region where convection, diffusion, and adsorption are allowed. The second control volume represents the immobile region where only diffusion and adsorption are allowed.

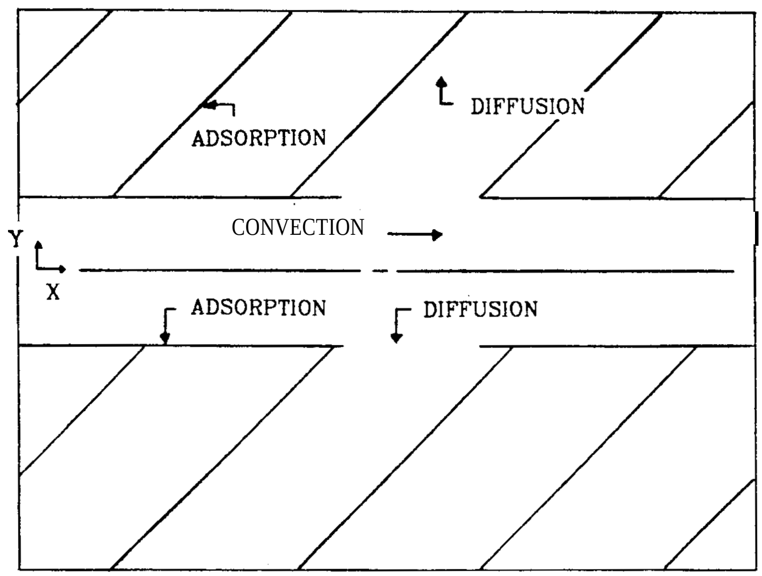

Figure *6* - SCHEMATIC OF TWO-DIMENSIONAL MODEL

A general mass balance on control volume (1) is

$$\begin{aligned}&(\textit{rate change of mass of species in control volume}) = \\ &\quad(\textit{net mass rate of species into control volume}) + \\ &\quad\quad(\textit{production of species in control volume}) \end{aligned} \tag{6.1}$$

Assuming:

(1) Production of species is negligible

(2) Density of species is constant

allows Equation (6.1) to be simplified. Thus Equation (6.1) becomes

$$\frac{\partial A_k}{\partial t} = -div(\, j_k^T\,) \tag{6.2}$$

where

A_k = Mass of species per total volume

j_k^T = Total mass flux of species

The mass per volume (A_k) can be expressed as

$$A_k = q_{a,m} + \varphi_m C_m \qquad (6.3)$$

where

$q_{a,m}$ = Total mass adsorbed per total volume

C_m = Concentration of species in mobile phase

φ_m = Portion of porosity due to mobile region

The adsorption term in Equation **(6.3)** can be expressed as

$$q_{a,m} = \rho_b \, P \, q_m \qquad (6.4)$$

where

ρ_b = Bulk density

P = Fraction of total adsorption sites in the mobile region

q_m = Adsorbed concentration in the mobile region per bulk volume

Substitution of equations **(6.3)** and **(6.4)** into Equation (6.2) gives

$$\frac{\partial A_k}{at} = \frac{\partial[\rho_b \, P \, q_m + \varphi_m \, C_m]}{at} \qquad (6.5)$$

Substitution of Equation (6.5) into Equation (6.1) gives

$$\frac{\partial[\rho_b \, P \, q_m + \varphi_m \, C_m]}{\partial t} = -div\,(j_k^T) \qquad (6.6)$$

The right hand side of Equation **(6.6)** can be expanded to

$$j_k^T = J_c \, C_m + j_k^d \qquad (6.7)$$

where

J_c = Convective flux density

j_k^d = Diffusive flux

Substituting Equation (6.7) into Equation (6.6) and differentiating gives

$$-div\,(j_k^T) = J_c^x \frac{\partial C_m}{\partial x} + C_m \frac{\partial J_c^x}{\partial x} + J_c^y \frac{\partial C_m}{\partial y} + C_m \frac{\partial J_c^y}{\partial y} + \frac{\partial j_k^{d,x}}{\partial x} + \frac{\partial j_k^{d,y}}{\partial y} \quad (6.8)$$

Assuming

(3) $J_c^y = 0$, convective flow in the x-direction only

(4) steady flow, $\frac{\partial J_c^x}{\partial x} = 0$

gives

$$-div(\,j_k^T\,) = [J_c^x \frac{\partial C_m}{\partial x} + \frac{\partial j_k^{d,x}}{\partial x} + \frac{\partial j_k^{d,y}}{\partial y}] \quad (6.9)$$

Assuming

(5) $\frac{\partial j_k^{x,d}}{\partial x}$ is negligible

gives

$$-div(\,j_k^T\,) = J_c^x \frac{\partial C_m}{\partial x} + \frac{\partial j_{k_{y,D}}}{\partial y} \quad (6.10)$$

Substitution of Equation (6.10) into Equation (6.6) gives

$$\frac{\partial\,[\,\rho_b\ P\ q_m\ +\ \varphi_m\ C_m\,]}{\partial t} = -[\,J_c^x \frac{\partial C_m}{\partial x} - \frac{\partial j_k^{y,d}}{\partial y}] \quad (6.11)$$

Because steady flow has been assumed, the convective term can be expressed as

$$J_c^x = V_m\ \varphi_m \quad (6.12)$$

The diffusion term can be expanded using Fick's Law of diffusion, as

$$j_k^{y,d} = -\varphi_m\ D_m^y \frac{\partial C_m}{\partial y} \quad (6.13)$$

where

D_m^y = Diffusion coefficient in the mobile phase in the y-direction

Substituting Equation (6.13) and Equation (6.12) into Equation (6.11) gives

$$\frac{\partial[\rho_b\ P\ q_m\ +\ \varphi_m\ C_m]}{\partial t} = \varphi_m D_m^y \frac{\partial^2 C_m}{\partial y^2} - \varphi_m\ V_m\ \frac{\partial C_m}{\partial x} \quad (6.14)$$

Differentiating gives

$$\rho_b \, P \, \frac{\partial q_m}{\partial t} + \varphi_m \frac{\partial C_m}{\partial t} = \varphi_m \, D^y_m \frac{\partial^2 C_m}{\partial y_2} - \varphi_m \, V_m \, \frac{\partial C_m}{\partial x} \qquad (6.15)$$

Equation (6.15) has two time dependent variables, the flowing concentration (C_m) and the adsorbed concentration (q_m). To reduce the number of variables to one, the adsorbed concentration is expressed as a function of the flowing concentration. There are many choices for such a relationship, but this study has used the simple Freundlich linear isotherm. This isotherm assumes equilibrium and instantaneous adsorption. The adsorbed concentration is related to the flowing concentration by

$$q_m = k \, C_m \qquad (6.16)$$

where

k = adsorption constant which is a function of temperature only

Applying this relationship to Equation (6.15) gives

$$[\, \varphi_m + \rho_b \, P \, k_m \,] \, \frac{\partial C_m}{\partial t} = \varphi_m \, D^y_m \, \frac{\partial^2 C_m}{\partial y^2} - \varphi_m \, V_m \, \frac{\partial C_m}{\partial x} \qquad (6.17)$$

Applying the same mass balance on control volume (2) as was applied on control volume (1) gives

$$\frac{\partial A_k}{\partial t} = -div(\, j^T_k \,) \qquad (6.18)$$

For control volume (2) only diffusion and adsorption are allowed so (A_k) becomes

$$A_k = \rho_b \, (1-P) \, q_{im} + \varphi_{im} \, C_{im} \qquad (6.19)$$

The flux term becomes

$$j^T_k = J_c \, C_{im} + j^d_k \qquad (6.20)$$

The first term on the right hand side of Equation (6.20) is equal to zero because no convection is allowed in this control volume. Differentiating Equation (6.20) gives

$$dw(j_k^T) = \frac{\quad}{\text{ax}} + \frac{\partial j_k^{y,d}}{\partial y} \tag{6.21}$$

As in the mobile region the diffusion in the x-direction is assumed to be negligible. The diffusion term is expressed using Fick's **Law** to give

$$j_k^{y,D} = -\varphi_{im} \, D_{im}^{y} \frac{\partial C_{im}}{\partial y} \tag{6.22}$$

Substituting equations (6.22),(6.21),and (6.19) into Equation (6.18) gives

$$\rho_b \, (1-P) \frac{\partial q_{im}}{\partial t} + \varphi_{im} \frac{\partial C_{im}}{\partial t} = \varphi_{im} \, D_{im}^{y} \frac{\partial^2 C_{im}}{\partial y^2} \tag{6.23}$$

Again using the Freundlich linear isotherm to relate the adsorbed concentration to the fluid concentration gives

$$[\, \varphi_{im} + \rho_b \, (1-P) \, k \,] \frac{\partial C_{im}}{\partial t} = D_{im}^{y} \, \varphi_{im} \frac{\partial^2 C_{im}}{\partial y^2} \tag{6.24}$$

Equations (6.17) and (6.24) are the governing partial differential equations **for** the two-dimensional model. The initial condition used for this model is one of uniform concentration. This is given by

$$C_m(x,y,0) = C_{im}(x,y,0) = C_i \tag{6.25}$$

Symmetry is invoked at the centerline of the mobile region, so

$$\left. \frac{\partial C_m}{\partial y} \right]_{y=0} = 0 \tag{6.26}$$

At the interface of the two control volumes, concentration is forced to be continuous, thus

$$C_m \Big]_{y=w} = C_{im} \Big]_{y=w} \tag{6.27}$$

where

w = half width of the mobile region

The flux across the interface is also continuous , giving

$$\varphi_m \, D_m \left. \frac{\partial C_m}{\partial y} \right]_{y=w} = \varphi_{im} \, D_{im} \left. \frac{\partial C_{im}}{\partial y} \right|_{y=w} \tag{6.28}$$

The outer boundary condition in the y-direction is

$$\left.\frac{\partial C_{im}}{\partial y}\right]_{y=\infty} = 0 \tag{6.29}$$

The inlet condition in the x-direction is

$$C_m \Big]_{x=0} = C_o \tag{6.30}$$

In order to simplify the governing differential equations and the associated boundary conditions a set of "dimensionless" variables are introduced.

$$C_1 = \frac{C_m - C_i}{C_o - C_i} \tag{6.31}$$

$$C_2 = \frac{C_{im} - C_i}{C_o - C_i} \tag{6.32}$$

$$y_D = \frac{y}{w} \tag{6.33}$$

$$x_D = \frac{x}{w} \tag{6.34}$$

$$Pe = \frac{V_m w}{D_m} \tag{6.35}$$

$$\beta = \frac{\varphi_m + \rho\, P\, k}{\varphi_T + \rho\, k} \tag{6.36}$$

$$R = \left[\frac{\varphi_T + \rho\, k}{\varphi_m\, V_m}\right] w \tag{6.37}$$

$$\alpha = \left(\frac{\varphi_{im}}{\varphi_m}\right)\left(\frac{D_{im}}{D_m}\right) \tag{6.38}$$

It should be noted that all the variables are dimensionless except (R) which has units of reciprocal time. Using these dimensionless variables the partial differential equations become

$$\beta\, R\, \frac{\partial C_1}{\partial t} = \left(\frac{1}{Pe}\right) \frac{\partial^2 C_1}{\partial y_D^2} - \frac{\partial C_1}{\partial x_D} \tag{6.39}$$

and

$$(1 - \beta)\, R\, \frac{\partial C_2}{\partial t} = \left(\frac{\alpha}{Pe}\right) \frac{\partial^2 C_2}{\partial y_D^2} \tag{6.40}$$

where

1 = mobile region

2 = immobile region

The initial and boundary conditions become

$$C_1(x_D, y_D, 0) = C_2(x_D, y_D, 0) = 0 \tag{6.41}$$

$$\left.\frac{\partial C_1}{\partial y_D}\right]_{y_D=1} = 0 \tag{6.42}$$

$$C_1\Big]_{y_D=1} = C_2\Big]_{y_D=1} \tag{6.43}$$

$$\left.\frac{\partial C_1}{\partial y_D}\right]_{y_D=1} = \alpha \left.\frac{\partial C_2}{\partial y_D}\right]_{y_D=1} \tag{6.44}$$

$$\left.\frac{\partial C_2}{\partial y_D}\right]_{y_D=\infty} = 0 \tag{6.45}$$

$$C_1(0, y_D, t_D) = 1 \tag{6.46}$$

The complete solution of the simultaneous partial differential Equations **(6.39)** and **(6.40)** with boundary conditions **(6.41)-(6.46)** is given in Appendix (A).

The general method of solution was to transform the equations with the Laplace transform with respect to time (t) and then again with respect to (x). With the equations in the transformed space (p-space), the solution could be solved for directly. Unfortunately, the resulting analytic solution cannot be analytically inverted. Thus to express the solution in real space required use of the Stehfest numerical inversion algorithm.[22] The details of this evaluation process will be discussed later.

The analytical solution for the concentration in the mobile phase in p-space is

$$C_1^p = \frac{C_o}{s\,(p + s\beta R)} - \left[\frac{z\,\alpha\,C_o}{s\,(p + s\beta R)}\right] \frac{e^{m y_D} + e^{-m y_D}}{(1-\alpha)M(e^{m} - e^{-m}) + z\,\alpha(e^{m} + e^{-m})} \tag{6.47}$$

where

s = Laplace operator for transforming t

p = Laplace operator for transforming x

$$z = \left[\frac{Pe\,(1-\beta)Rs}{a}\right]^{1}$$

$$m = \left[\frac{p + s\beta R}{Pe}\right]^{\frac{1}{2}}$$

No mention of a fractured or a matrix is made in the above development, rather the only distinction is that between a mobile and an immobile region. Thus the model is general.

"he general nature of this model is best seen by considering the variables (Om), (φ_{im}), and (φ_m) where

(φ_m) = Portion of total porosity due to the mobile region

(φ_{im}) = Portion of total porosity due to the immobile region

(φ_T) = Total porosity

When considering the case of a completely saturated (single phase) porous medium, essentially the entire volume should be considered as mobile, thus

$$\varphi_m >> \varphi_{im}$$

This is true no matter what fraction of the entire reservoir is considered.

With regard to the fraction of mobile to immobile region, the completely saturated porous media is at one end of the spectrum with essentially everything being mobile while a fractured media is at the other end. In a fractured medium all but a small portion of the entire reservoir is immobile, thus over the entire reservoir

$$\varphi_{im} >> \varphi_m$$

The model is able to consider both porous and fractured media as well as intermediate cases. Such intermediate cases would include only partially saturated single phase reservoirs and could possibly include multi-phase systems if the loss of the traced material from the sampled fluid depended upon the concentration difference of the traced material between the sampled fluid and the other fluids. While this model has other applications, this study has concentrated on applying the model to fractured systems.

When considering fractured systems the nature the testing procedure is important in the understanding of the different variables. As discussed above, the portion of the total porosity due to the mobile phase is given by (φ_m). When tracer testing in a fractured medium, only a finite amount of tracer is injected, thus not the entire reservoir is examined. In this case $(a,)$ is more accurately the portion of the encountered porosity due to the mobile phase rather than the portion of the total porosity due to the mobile phase. This is not necessarily a handicap to tracer analysis as will be discussed later.

Section 7: EVALUATION TECHNIQUE

The solution to the two dimensional model is for a step input and is analytic only in p-space, where p-space is two Laplace transformations away from real space. Any investigation of how the different parameters that were included in the physical model effect tracer movement in a reservoir requires expressing the solution in real space for any value of the dimensionless variables. Furthermore, since most tracer tests are not step inputs, the solution for a step input of finite duration (a finite-step) and the solution for a spike input (infinitesimally short duration) are needed.

A computer program GENERATE.STEP was developed to evaluate the solution for a step input in real space. A listing of GENERATE-STEP is given in Appendix (B).

GENERATE.STEP is made up of five parts; the main program (MAIN), the function INVERSE1, the subroutine SFUNCTION, the function INVERSE2, and the subroutine PFUNCTION.

The main program (MAIN) reads and writes the values of the dimensionless variables and the time steps at which the solution is to be evaluated. MAIN then evaluates the solution by calling the function INVERSE1.

INVERSE1 is the Stehfest numerical inversion algorithm used to invert from (x,y,s)-space to (x,y,t)-space (real space). This algorithm requires an evaluation of the solution in (x,y,s)-space and gets this by calling the subroutine SFUNCTION.

Since the solution is not analytic in (x,y,s)-space, SFUNCTION gives an evaluation of the solution in (x,y,s)-space by calling the function INVERSE2.

INVERSE2 is again the Stehfest algorithm which is used here to invert from (p,y,s)-space to (x,y,s)-space. The expression of the solution in (p,y,s)-space which is needed by INVERSE2 is evaluated by calling PFUNCTION.

PFUNCTION evaluates the value of the solution in (p,y,s)-space from the analytic expression of the solution.

The program then returns control to MAIN which writes the value of the solution for all the time steps. Examples of input and output files are also given in Appendix (B).

The use of the Stehfest algorithm to invert the solution results in "noise" or error caused by the numerical technique. This error is greatest where the function to be inverted is not smooth. A consequence of this error in the calculation procedure is that negative concentrations are sometimes calculated in region where noise dominates. Since negative concentrations are clearly not allowed, the program sets all negative values to zero.

The evaluation of the solution for a step input is given by GENERATE.STEP, but the evaluation for a finite-step and a spike-step input required modifications to the above procedure.

To evaluate the solution for a finite-step the program GENERATE.FINSTEP was developed. This program is very similar to GENERATE.STEP and has used the concept of superpositon in time to generate the results for a finite-step. Using superpositon, the concentration after the step input has ended is given by

$$C_{fs}(t,\alpha_i) = C_s(t+\Delta t,\alpha_i) - C_S(t,\alpha_i) \qquad (7.1)$$

where

C_{fs} = Concentration for finite step t

C_s = Concentration for step input

Δt = Duration of finite step

t = Time since the end of the step input

The only part of GENERATE.STEP that needed to be modified was the main program MAIN. The modified MAIN that was used in GENERATE.FINSTEP is given in Appendix (C). All other programs in GENERATE.FINSTEP are exactly those already given in GENERATE.STEP. Since MAIN is changed in GENERATE.FINSTEP, the input file is different than that used in GENERATE.STEP. An example input file is also given in Appendix (C).

The evaluation of the solution for a spike-input is greatly simplified by the solution technique used. It can be shown that the response of a spike-input is merely the time derivative of a step input. Using the Laplace property

$$\frac{\partial F}{\partial t} = L^{-1}[s\ f(s)] \tag{7.2}$$

it is easy to see that all that is needed to do to get the spike-input from the step input is to multiply the expression for the step input in (x,y,s)-space by (s) before it is inverted to (x,y,t)-space. This is easily done by modifying SFUNCTION . Thus the program to calculate the results for a spike-input, GENERATE.SPIKE, is exactly like GENERATE.STEP except for a slight modification in SFUNCTION. A listing of the SFUNCTION used in GENERATE.STEP is given in Appendix (D). Since the main program is not changed in GENERATE.SPIKE it requires the same input as GENERATE.STEP.

Section 8: RESULTS OF TWO-DIMENSIONAL MODEL

It was possible to examine how each of the variables in the two-dimensional model affects tracer movement through a reservoir by examining the return curves that were generated by the procedure described above. While one could look at all three type of tests (step, finite-step, spike), only the finite-step and spike are practically applicable. Since most field tests are more like a spike-input, this study examines how each of the dimensionless variables effect tracer return curves for a spike-input of tracer.

The step-input program (GENERATE.STEP) was used to check the analysis procedure. One would expect that if a system were subjected to a step-input of unit concentration then the response of this system would have an initial delay followed by an asymptotic approach to unity. Entering base values of the variables and evaluating the solution gave a curve similar to what was expected. Figure *(7)* shows a response for a typical step-input.

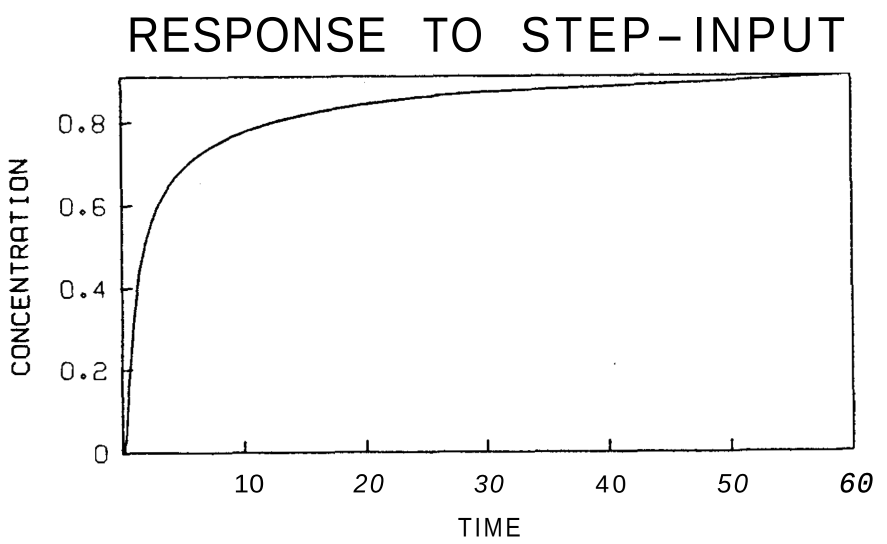

Figure 7 - RESPONSE TO STEP INPUT

In the generation of any return curve, the value of all six dimensionless variables need to be entered. A simplifying assumption was made when a value of (y_D) was entered. The concentration profile at a given distance (x_D) should be the area average of the calculated values for all (y_D) values, where (y_D) ranges from zero to one. This averaging procedure was not done, instead the value of the concentration calculated at $(y_D) = 0.5$ was used as the average value. This assumption was made to simplify an already complicated procedure and to prevent an already long running (cpu time) computer program from becoming prohibitively long. The basis for this assumption was a series of calculations of the concentration profile across a fracture. These profiles showed an essentially flat profile with little variation in concentration across the fracture. This flat profile is to be expected because in the solution of the two-dimensional model it was assumed that the velocity profile was not a function of (y).

The sensitivity of the model to the five dimensionless variables; (x_D), (Pe), (β), (R), and (a) was studied by examining how a typical return curve was affected by varying each of the variables independently. The base values of the dimensionless variables used were determined by "eye" fitting the model to real data. The base values used in the sensitivity studies are shown in Table (4). The sensitivity study showed large differences in sensitivity among the five variables.

Table 4 - BASE VALUES USED IN SENSITIVITY STUDY

Pe	β	R	a	x_D
0.02	0.50	.002	.10	10000

The dimensionless variable (x_D) is the dimensionless distance between wells. The two-dimensional model was more sensitive to (x_D) than any of the other variables. In order to be able to graph the results of varying (xg) on a single plot the value of $\langle x_D \rangle$ could only be increased and decreased by a factor of two. The base case $(x_D = 10000)$ and the higher and lower cases are shown in Figure *(8)*.

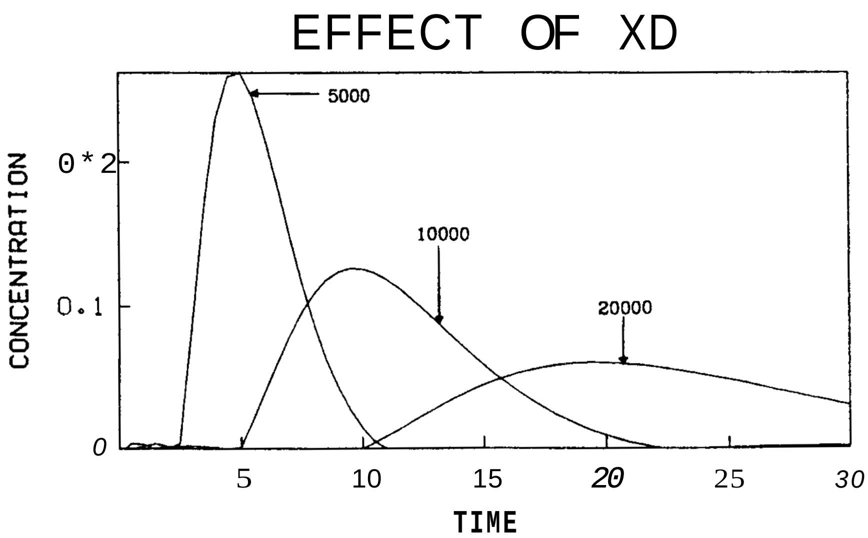

Figure *8* - EFFECT OF X_D

As would be expected when the dimensionless distance is decreased the tracer both breaks through earlier and has a higher peak concentration than the base case. When $\langle x_D \rangle$ **is** increased, equivalent to a greater distance between wells, the breakthrough occurs later and the peak concentration is less. Not only breakthrough times and peak concentration are changed, but the shape of the curve is changed as well. In particular the backside of the return curves are quite different depending on the value of $\langle x_D \rangle$. The backside of the return curves **is** where retention effects are visible and as would be expected the greater the distance between wells the more retention occurs and thus the longer the tailing effects.

The dimensionless variable (Pe) is given by

$$Pe = \frac{V_m w}{D_m}$$

and is a modified form of the Peclet number, an important variable in many mass transport systems. Usually the characteristic length that would have been used to make this group dimensionless would have been the actual distance between the wells, but this study has used the fracture half-width **(w).** The result is that the values of the Peclet number here are orders of magnitude less than commonly seen.

Despite its frequent use as a group to define many systems, the two-dimensional model showed small sensitivity to the Peclet number. Other recent studies have seen similar **effects.**[21] Figure (9) shows the effect of decreasing the Peclet number by a factor of ten and increasing the Peclet number by a factor of five. The most apparent effect of changing the Peclet number is to change the amount of retention or equivalently the amount of tailing of the return curve. Figure (9) shows that the larger the Peclet number the more the tailing effect. Figure (9) also shows that relatively large changes in the Peclet number cause small changes in the breakthrough times. Another important observation is that the changes in the Peclet number do not create symmetric changes in the return curve.

The sensitivity of the model to (a), where

$$\alpha = \left(\frac{\varphi_{im}}{\varphi_m}\right)\left(\frac{D_{im}}{D_m}\right)$$

is shown in Figure (10). Like the Peclet number the relative sensitivity of the model to (a) is small. **A** decrease in (a) by a factor of ten results in a higher peak concentration, less tailing effects and a similar shape when compared to the base case. An increase by a factor of eight results in much more tailing and a lower peak concentration. Breakthrough times on all the curves are similar.

The effect of changing the variable (R) where

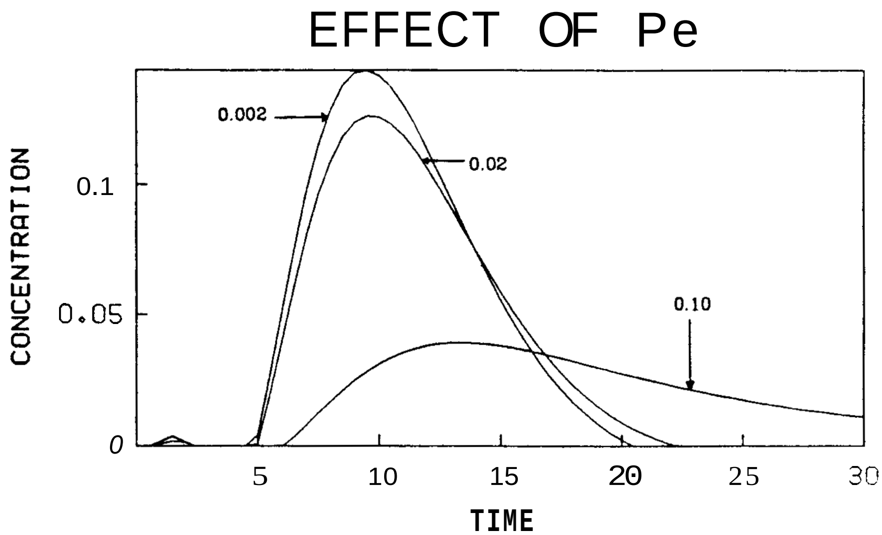

Figure 9 - EFFECT OF Pe

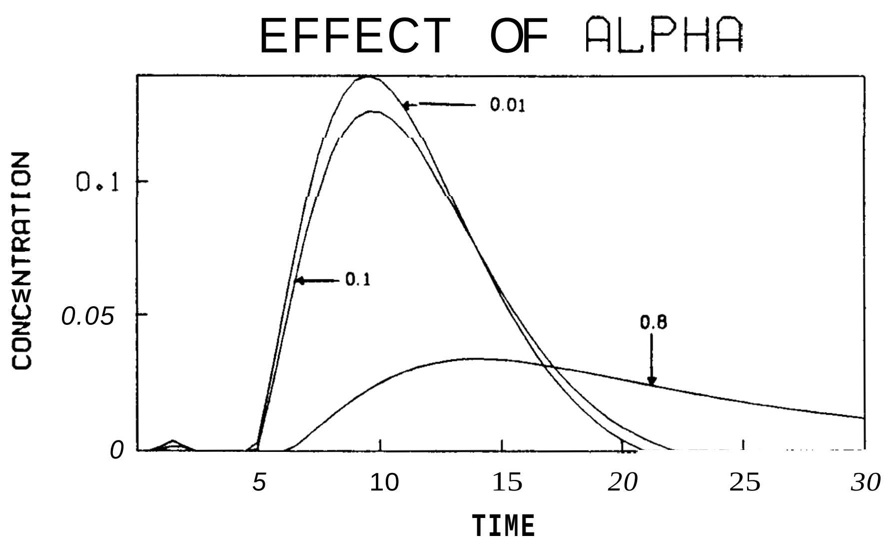

Figure 10 - EFFECT OF a

$$R = \left[\frac{\varphi_T + \rho k}{\varphi_m V_m}\right] w$$

is similar to the effect of changing (x_D). Doubling the base value of (R) results in more tailing and a slower breakthrough. Decreasing the base value **by** a factor of two gives a profile with a higher peak concentration and a earlier breakthrough. These results can be seen in Figure (11).

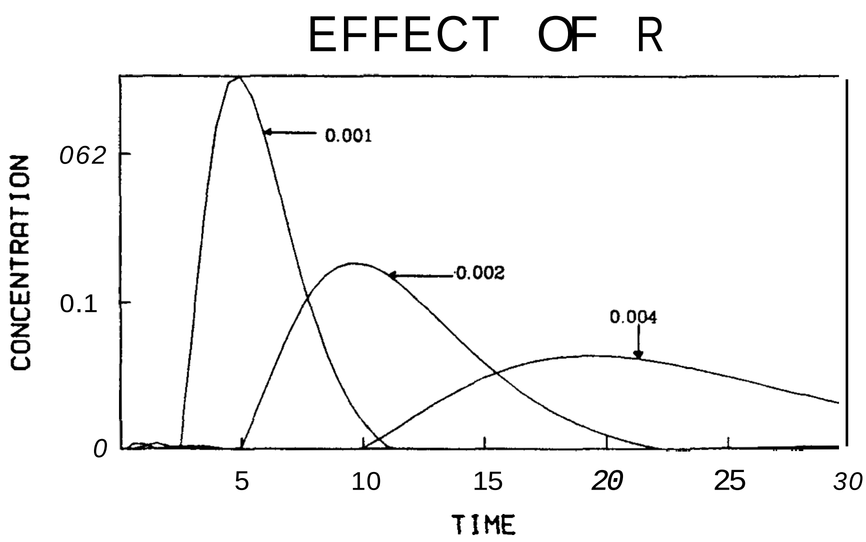

Figure 11 - EFFECT OF R

The sensitivity of the two-dimensional model to (β) is intermediate between the high sensitivity of the model to (x_D) and (R) and the low sensitivity of the model to (Pe) and *(a)*. The variable (β) is defined as

$$\beta = \frac{\varphi_m + \rho P k}{\varphi_T + \rho k}$$

and gives the fraction of the total retardation due to the fractured region. By definition (β) is constrained to lie between zero and one. As can be seen in Figure (11) decreasing (@) from (0.5) to (0.1) resulted in an increase in the peak concentration, a decrease in the breakthrough time and a decrease in the amount of tailing. An increase in (β) gave opposite results.

The above sensitivity study shows that the two-dimensional model is affected differently by the five dimensionless variables that define the model. Within the five dimensionless variables there are at least eight unknown physical parameters (w, φ_m, φ_T, P, k, V_m, D_m, D_{im}), thus there is no unique combination of physical parameters that can be determined from or can determine the five

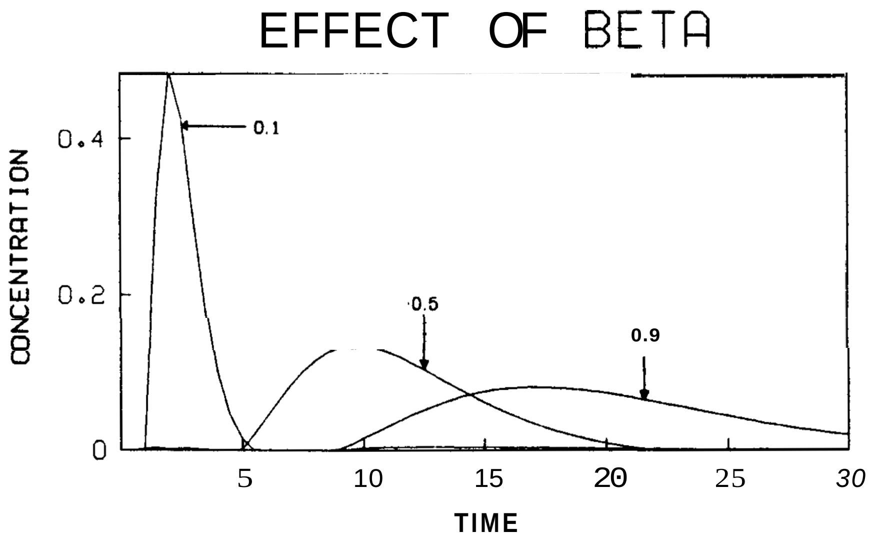

Figure 12 - EFFECT OF β

dimensionless variables. This two-dimensional model makes possible the production of tracer return curves for any given set of physical parameters and the associated dimensionless variables.

Section 9: TRACER TEST ANALYSIS TECHNIQUE; NUMERICAL CURVE FITTING

The initial goal of this study was to model and quantify those processes that affect the movement of a tracer as it moves through a reservoir. The two-dimensional model described above allows this by producing tracer return curves for any given set of dimensional variables that define the system. This forward type of problem, a problem where the input and system are known and the output from the system is desired, may be used to study the effects of different processes on the model but it is not helpful for the inverse problem. The inverse problem, frequently encountered in reservoir engineering, is where both the input and the output are known while the system is the unknown. This is the type of problem that must be dealt with in the interpretation of an actual field tracer test. In a field case the details of how the tracer was injected are known (the input), and the tracer return curves are known (the output), what is desired is an interpretation of the reservoir (the system).

The general procedure to solve the inverse problem is to statistically fit a model to the real data. From this fit the optimum values of the variables that define the model may be determined. Hopefully from the values of the defining variables it may be possible to say something about the reservoir. Since the two-dimensional model developed in this study is very general and the variables that define this model give information about the reservoir, it was hoped that this model could be applied in the interpretation of tracer tests. It should be noted that previous studies, in particular Jensen's [16] study, have attempted to fit models to real data with considerable success. Unfortunately the variables that were determined from the fitting process did not reveal much about the reservoir. The advantage of the present model is that the variables involved are more directly associated with reservoir properties.

Before the two-dimensional model can be applied to a fitting process the solution must be put in a form that is open to statistical analysis. The solution for a step input in terms of the dimensionless variables can be written as

$$C_s\,(t;\alpha_i) \;=\; C_o\; F(t,\alpha_i) \tag{9.1}$$

where

$C_s(t;\alpha_i)$ = Concentration at time t for a step input.

C_o = Concentration at inlet

$F(t;\alpha_i)$ = Solution for unit-step input at time t

α_i = Dimensionless variables (i=1,5)

α_1 = Pe

$\alpha_2 = \beta$

α_3 = R

$\alpha_4 = a$

$\alpha_5 = x_D$

Using superpositon, the solution for a finite-step (C_{fs}) can be written as

$$C_{fs} \;=\; C_o[F(t \;\mathbf{t}\; At) - F(t)] \qquad \text{for } t > \Delta t \tag{9.2}$$

The inlet condition (C_0) can **be** expressed as

$$C_o \;=\; \frac{A4}{Q\Delta t} \tag{9.3}$$

where

M = Total mass input

Q = Total volume fiowrate

At = Duration of input

Substituting the expression for the inlet concentration into the Equation (9.2) **gives**

$$C_{fs} \;=\; \frac{M}{Q}\left[\frac{F(t+\Delta t) - F(t)}{At}\right] \tag{9.4}$$

Allowing *(At)* to approach zero is equivalent to having a spike-input. Thus a spike input is given by

$$C_{sp} = \frac{M}{Q} \lim_{\Delta t \to 0} \left[\frac{F(t+\Delta t) - F(t)}{\Delta t} \right]$$
$$= \frac{M}{Q} \frac{\partial F}{\partial t} \tag{9.5}$$

This solution can be written in a generalized form as

$$C_{sp} = E\, f(t;\alpha_i) \tag{9.6}$$

The term (E) is a normalization factor that normalizes the function to one. Since there is no detailed knowledge of the inlet conditions at the entrance to the fracture, a normalization of the solution is needed. This normalization of the solution has no effect on the shape of return curves it only changes the size of the curve.

The parameters in Equation (9.6) were optimized by using a non-linear least squares method of curve fitting. This curve fitting was done by using VARPRO[23], a computer program developed by the Computer Science Department of Stanford University. VARPRO optimizes both the non-linear and the linear parameters of a given function.

The method of curve fitting used in VARPRO is based on a paper by Golub and Pereya.[24] It is shown that a non-linear model of the form

$$\eta(\alpha,\beta;T) = \sum_{j=1}^{L} \beta_j\, \varphi_j(\alpha;T) + \varphi_{L+1}(\alpha;T) \tag{9.7}$$

Where

η = Model to be fit

a = Non-linear parameters

β = Linear parameters

T = Independent variable

L = Number of linear parameters

φ = Nonlinear function

can be fitted by a non-linear least squares method by separately optimizing the linear parameters and the non-linear parameters.

In the present case there is only one linear parameter (E) and five non-linear parameters (α_i). The objective function (0) which is minimized by the least squares fit is given by

$$O(E,\alpha_i) = [C_i - C(t;E,\alpha_i)]^2 \tag{9.8}$$

where

C_i = Observed concentrations

C = Calculated concentrations

This function is minimized by using initial estimates of the non-linear parameters and then iterating to determine the optimum values of the non-linear parameters. The optimum linear parameter is then determined.

The details of how VARPRO works are discussed elsewhere.[15,23] It is important to note that since a Taylor expansion of the objective function (0) with respect to the non-linear parameters (α_i) is used, an expression of the derivative of the two-dimensional solution with respect to the non-linear parameters was needed.

A summary of the input requirements of VARPRO is

(1) N observed concentrations (C_i)

(2) Value of the independent variable (T) at each data point

(3) Estimate of the non-linear parameters

(4) Evaluation of the solution at any given (T) and for any set of dimensionless variables (α_i)

(5) Evaluation of the derivative of the solution with respect to the non-linear parameters at any given (T) for any set of (α_i).

The subroutine that calculated the solution and the derivative of the solution with respect to the non-linear parameters was called ADA. ADA needed to include the double Stehfest numerical inversion techniques used in the different GENERATE programs {see Section 7). Since the analytic solution is available only in (p,y,s)-space, the calculation of the derivatives was of necessity in (p,y,s)-space also. Thus ADA needed to doubly invert both the solution and the derivatives. The calculation of the derivatives is discussed in Appendix (E).

A main program **(MAIN)** was also needed to; read in the data and the initial estimates of the non-dimensional variables, to call VARPRO, and to print the final results. A listing of CURVE.FIT, which is the program that incorporates VARPRO and all required subroutines, is given in Appendix (F).

The goal of CURVE.FIT is to determine the optimum values of the five dimensionless variables for a given set of real data. The goal of the entire tracer analysis is to determine something of the nature of the reservoir. This is done by relating the dimensionless variables to the reservoir parameters. There can be no unique determination of all of the different reservoir parameters because there are more unknown reservoir parameters than dimensionless variables. However it is possible to uniquely determine some of the physical parameters from the dimensionless variables.

The most important parameter that can be determined is the fracture half-width (w), which can be obtained directly. Using the definition of $\langle x_D \rangle$, the fracture half-width is given by

$$w = \frac{x}{x_D} \tag{9.7}$$

where

x = distance between wells

The fracture aperture is important not only to the flow model, as was shown in the sensitivity study, but also to any subsequent heat-transfer model that would be used to forecast thermal breakthrough. This ability to solve directly for the fracture width is a major advantage over preceding methods of curve fitting.

The other reservoir parameters cannot be directly determined but could be approximated if some additional information were available. From the definition of the Peclet number

$$Pe = \frac{V_m w}{D_m} \tag{9.9}$$

it can be seen that if the value for the velocity in the mobile phase (V_m) were known then the diffusion coefficient in the mobile phase could be calculated. The velocity term can be approximated by using the breakthrough time (t_{bt}) and the distance between the wells as

$$V_m = \frac{x}{t_{bt}} \tag{9.10}$$

This approximation ignores retardation effects. Using Equation (9.10). (D_m) can be approximated by

$$D_m = \frac{x\ w}{t_{bt}\ Pe} \tag{9.11}$$

Combining the definitions for (β) and (R), equations (6.36) and **(6.37)** respectively, gives (φ_m) as

$$\varphi_m = \frac{\rho P k}{[1 - \frac{V_m R k}{w}]} \tag{9.12}$$

The values of (R), and (β) are determined from the curve fitting procedure, (w) can be calculated and (V_m) can be approximated as discussed. It may be possible to determine (k) experimentally and (ρ) can be estimated. Values for (P) cannot be determined, but since (P) by definition ranges from zero to one only, equation (9.12) can give a range for (φ_m).

A range for (φ_{im}) can be determined if a value for (D_{im}) can be experimentally determined since

$$\varphi_{im} = \varphi_m \left(\frac{D_m}{D_{im}}\right) \tag{9.13}$$

As shown above, the two-dimensional model developed in this study can be posed in the form necessary to apply a numerical curve fitting procedure. From this curve fitting technique it is possible to determine the optimum values of the dimensionless variables , and from the values of these variables it is possible to directly calculated the fracture width and to indirectly determine some of the other physical parameters used to develop the model. The application of this technique to real data is now discussed.

Section 10: APPLICATION OF TRACER TEST ANALYSIS TECHNIQUE

The analysis technique discussed in the previous section was tested by applying it to data from tracer tests in the Wairakei geothermal field in New Zealand. This data was collected by the Institute of Nuclear Sciences of the Department of Scientific and Industrial Research, New Zealand, and made available to the Stanford Geothermal Program for this study. No attempt was made to interpret to results on a field wide basis, rather the purpose was only to test the curve fitting procedure.

The first important result found from attempting to fit the model to real data was that the initial values of the non-linear variables enter into the curve fitting process had to be "good guesses". If the initial values were not good choices the matching process would fail altogether. Good choices were determined by first generating return curves with the GENERATE.SPIKE program given in Section **6** that were similar to the real data.

The second important result was that the curve fitting procedure had very slow convergence with the five parameter model. The consequence of this was that more than one combination of initial guesses of the non-linear parameters and subsequent numerical curve fitting was necessary to produce an acceptable fit. It was found that it usually took at least three series of guesses and numerical curve fittings to create a final fit. **A** flow chart of the procedure is shown in Figure **(13)**. The most important step in the overall procedure was the intermediate step where the best fit values determined after numerically curve fitting were changed before re-entering the numerical curve fitting procedure. This step required a knowledge of how the variables affect the shape of the return curves. The sensitivity study discussed in Section ***(7)*** provided this information.

The third general result found when using this curve fitting procedure with real data is that the procedure requires a large amount of computing time. An example case of twenty iterations on a data set of forty points took about 200 minutes of c.p.u. time on a DEC VAX #750. Since any final match required many such fits the computing time became a constraint.

The result of the curve fitting procedure for wells #24, #103, and #121 are shown in Figures (14), (15), and (16) respectively. In these figures the data are shown as crosses and the generated curves using the optimum values for the variables are shown as solid lines.

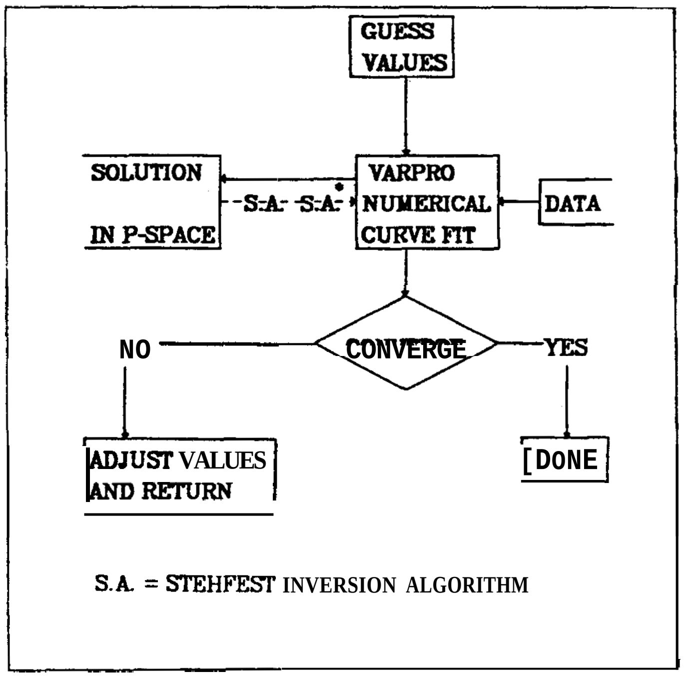

Figure 13 - CURVE FITTING PROCEDURE

In order to compare with other models, the results from the Fossum[15] and Jensen[16] models for the same wells using a single fracture fit are shown in Figures (17)-(22). The fits from the present study are better than those from Fossum's model and are comparable to those from Jensen's model.

The values of the dimensionless variables used to generate the curves shown in Figures (14)-(16) are given in Table (5). The only reservoir parameter that can be directly determined from the dimensionless variables is the fracture

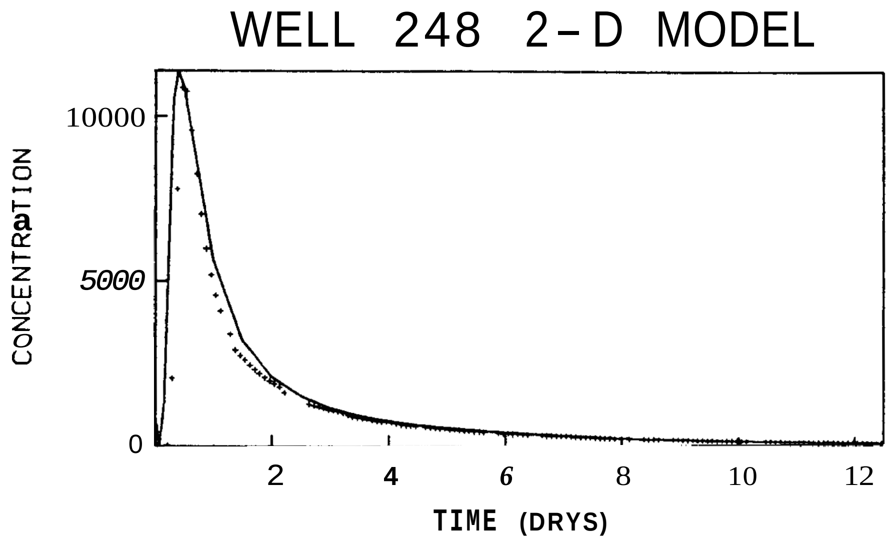

Figure 14 - WELL #24 FIT **WITH TWO-DIMENSIONAL MODEL**

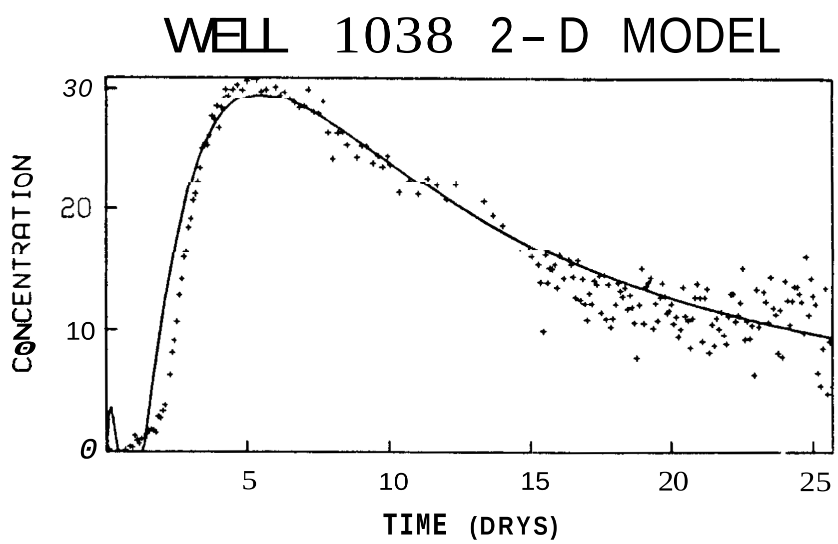

Figure 15 - WELL #103 FIT **WITH TWO-DIMENSIONAL MODEL**

width. The value of the fracture width for each case is **given in Table (6). The fracture widths shown in Table (6) range** from a **low of** 2.7mm to a **high** of **10.**1mm.

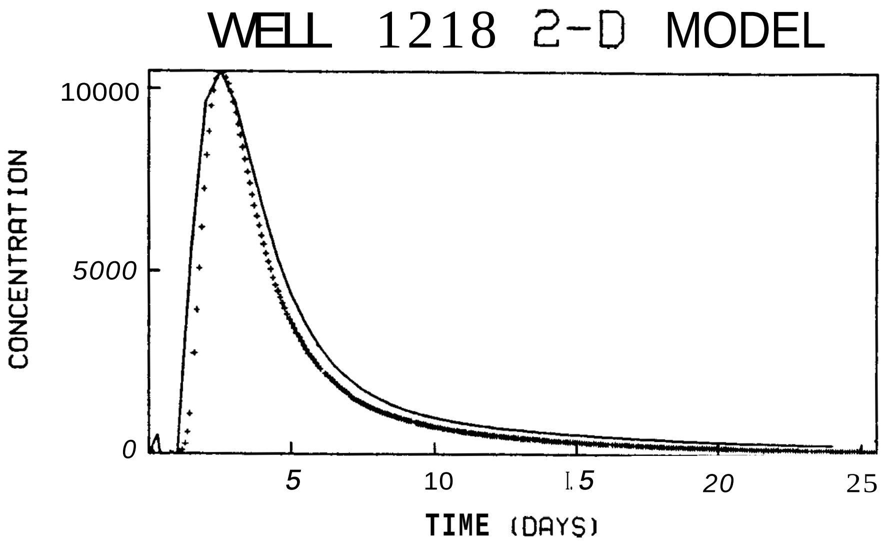

Figure **16** - WELL #121 FIT WITH TWO-DIMENSIONAL MODEL

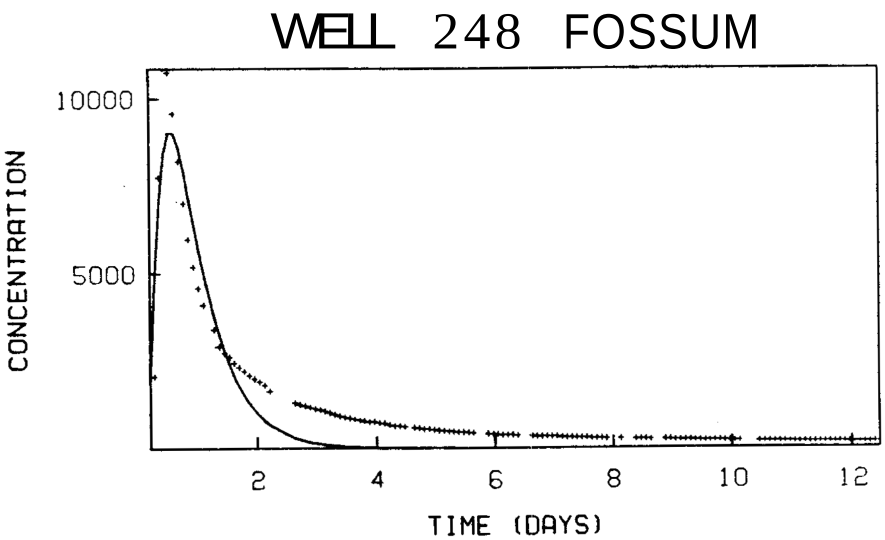

Figure **17** - WELL **#24** FIT WITH FOSSUM'S MODEL

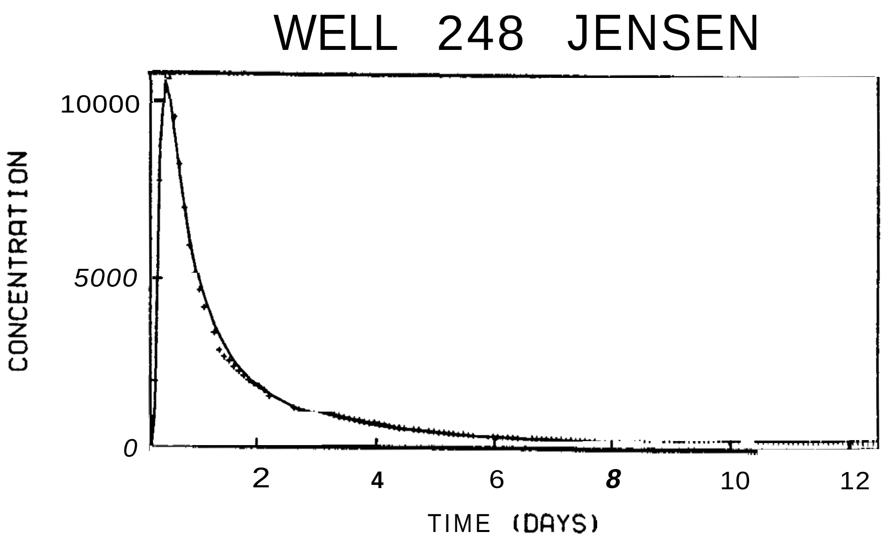

Figure 18 - WELL #24 FIT WITH JENSEN'S MODEL

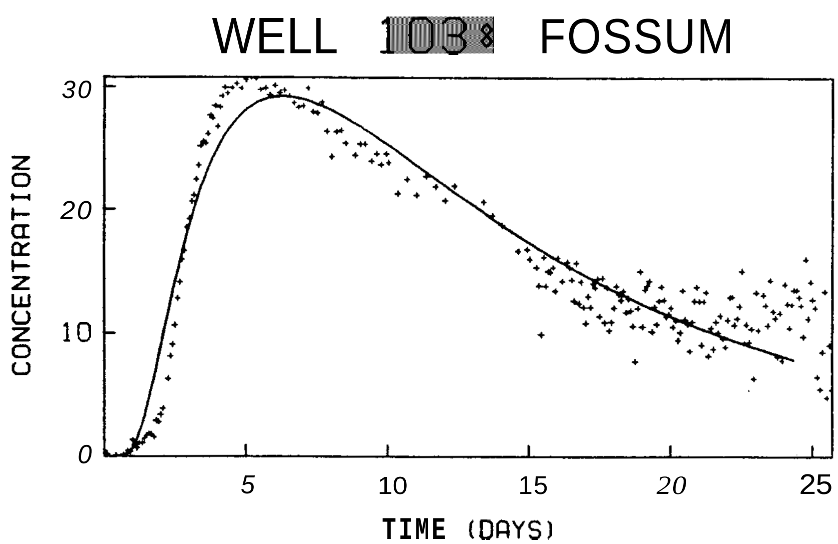

Figure 19 - WELL #103 FIT WITH FOSSUM'S MODEL

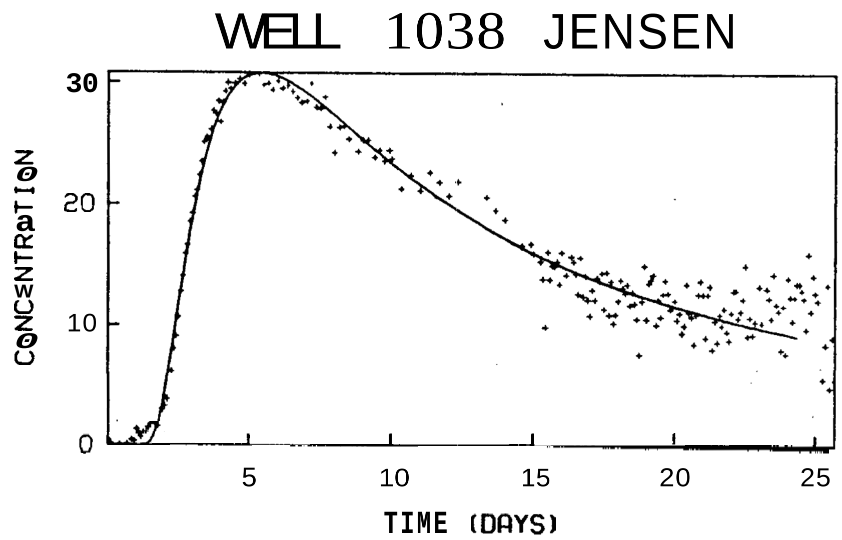

Figure 20 - WELL #103 FIT WITH JENSEN'S MODEL

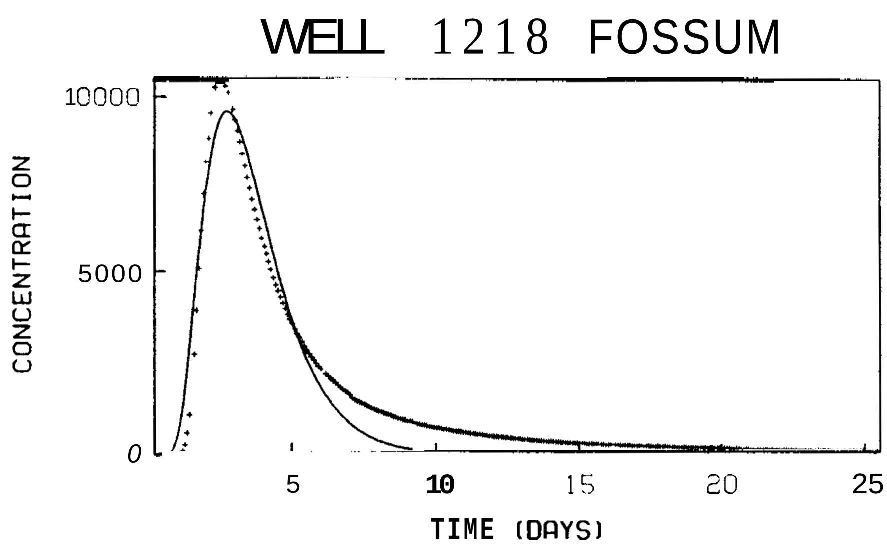

Figure 21 - WELL #121 FIT WITH FOSSUM'S MODEL

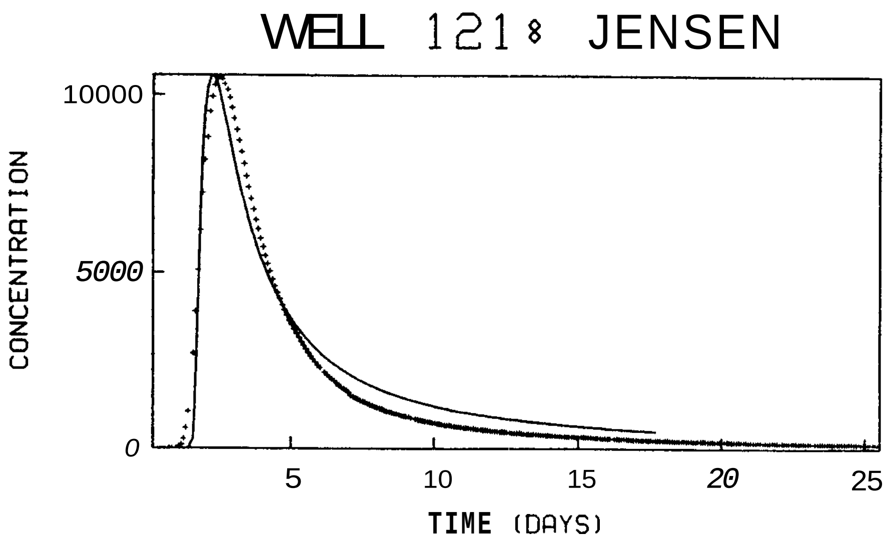

Figure 22 - WELL #121 FIT WITH JENSEN'S MODEL

Table 5 - BEST FIT VALUES					
WELL#	x_D	Pe	β	R	a
24	1.55305	0.201	0.502	2.01E-06	0.0021
103	5.01E04	0.200	0.450	2.00E-05	0.110
121	9.71304	0.170	0.500	3.44E-05	0.004

TABLE 6 - CALCULATED FRACTURE WIDTHS	
WELL#	FRACTURE WIDTH (mm)
24	2.7
103	6.6
121	10.1

Section 11: CONCLUSIONS

(1) Laboratory batch experiments run with Los Azufres, Mexico and Klamath Falls, Oregon reservoir rock are not sensitive enough to study the mass transfer processes active in tracer movement through a reservoir.

(2) A two-dimensional model that represents a fractured medium by a mobile region, in which convection, diffusion, and adsorption are allowed, and an immobile region, in which only diffusion and adsorption are allowed, can be used to represent tracer movement through a geothermal reservoir.

(3) The two-dimensional model that was derived in this study has demonstrated how each of the various mass-transfer processes included in the two-dimensional model affect tracer return curves.

(4) It is possible to numerically fit real data to the two-dimensional model.

(5) The optimum values of the parameters determined from the curve fitting procedure provide a direct estimate of the fracture width and can be used to estimate other reservoir flow parameters if experimentally determinable values are known.

Section 12: SUGGESTIONS FOR FURTHER WORK

An experimental study has been designed that would allow for experimental verification of the two-dimensional model developed in this study. This proposed study would use the same equipment as the experimental phase of the present study with only a few modifications.

The basic idea would be to separate the stainless steel core holder into a mobile and a immobile region by packing the center portion of core holder with larger grain material than the outer region. A large permeability difference between the two regions would effectively cause the center region to be mobile and the outer region to be immobile. Previous studies [25] have shown that the difference in grain size required to achieve a 40:1 permeability ratio between center and outer region is not very large. Sand with a (8-12) mesh range, packed to approximately 35 percent porosity will give about a 1700 Darcy permeability while a (40-60) mesh range will only give about a 45 Darcy permeability.

Separation of the two sands would be maintained by a wire mesh tubular holder placed inside of the stainless steel core holder. Other necessary modifications to the present equipment would include:

> (1) Using endplugs with a single port entrance to restrict flow to the center region.
>
> (2) Rearranging the valves so that a instantaneous switch from water to tracer can be made.

By conducting flow tests in the apparatus described above it would be possible to verify that the two-dimensional model is correctly determining the "fracture" size by varying the diameter of the center region. Multiple tests (varying flow rates) could be used to determine other flow parameters such as the diffusion coefficients.

Section 13: REFERENCES

1. Horne, R.N., "Geothermal Reinjection Experience in Japan," Journal of Petroleum Technology, March 1982.

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

3. Strom, P.W., Johnson, W.E., "Field Experiments with Chemical Tracers in Flood Waters," Producers Monthly, Vol. 15, No. 2, 1951, 11-18.

4. Wagner, O.R., "The use of Tracers in Diagnosing Interwell Reservoir Heterogeneities - Field Results," Journal of Petroleum Technology, November, 1977.

5. Deans, H.A., "A Mathematical Model for Dispersion in the Direction of Flow in Porous Media," Trans., AIME (1963), Vol. 228, 49.

6. Carslaw, H.S., Jaeger, J.C.. Conduction of Heat in Solids, Clarenden Press, Oxford, 1959.

7. Perkins, R.K., Johnson, O.C., "A Preview of Diffusion and Dispersion in Porous Media," Trans. AIME (1963), Vol. 228, 70.

8. Coats, K.H., Smith, B.D., "Dead End Pore Volume and Dispersion in Porous Medial," SPE of AIME Trans., March 1964, 73-84.

9. Brigham, W.E., "Mixing Equations in Short Laboratory Cores," SPE of AIME Trans., Feburary, 1974, 91-99

10. van Genuchten, M. Th., Wierenga, P.J.. "Mass Transfer Studies in Sorbing Porous Media; I, Analytical Solutions," Soil Science of America Journal, Vol.40, No.4, August, 1976, 473-480.

11. van Genuchten, M. Th., Cleary, R.W., "Movement of Solutes in Soil; Computer-Simulated and Laboratory Results," Soil Chemistry B. Physico-Chemical Models, Elsevier Scientific Publishing Company, New York, 1982.

12. 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.

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

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

15. 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.

16. Jensen, C.L., Matrix Diffusion and its Effect on the Modeling of Tracer Returns from the Fractured Geothermal Reservoir at Wairakei, New Zealand , Stanford Geothermal Program, SGP-TR-XX, Stanford CA, December 1983.

17. Sageev, A., Design and Construction of an Absolute Permeameter to Measure the Effect of Elevated Temperature on the Absolute Permeability to Distilled Water of Unconsolidated Sand Cores , Stanford Geothermal Program, SGP-TR-43, Stanford CA, December 1980.

18. Johnson, S.E., Doublet Tracer Testing at Klamath Falls Geothermal Resource: A Model Comparison , Stanford Geothermal Program, SGP-TR-XX, Stanford CA, June 1984.

19. Sufi, A.H., Ramey H.J., Brigham, W.E., Temperature Effects on Oil-Water Relative Permeabilities for Unconsolidated Sands , Stanford University Petroleum Research Institute, Stanford CA, October 1982.

20. Jackson, P.B., Method for the Collection and Analvsis of Sample Fluids During a Tracer Test , Stanford Geothermal Program, SGP-TR-XX, Stanford CA, June 1984.

21. van Genuchten, M.Th., "Non-Equilibrium Transport Parameters from Miscible Displacement Experiments," United States Department of Agriculture Science and Education Adminstration, Report No. 119. U.S. Salinity Laboratory, Riverside CA, February 1981.

22. Stehfest, H., "Algorithm 368 Numerical Inversion of Laplace Transforms," Communications of the ACM , January 1970.

23. VARPRO, Computer Science Department, Stanford University, Stanford CA.

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

25. van Poollen, H.K., Tinsley, J.M., and Saunders, C.D., "Hydraulic Fracturing - Fracture Flow Capacity **vs.** Well Productivity," Trans. AIME (1958), Vol. 213.91.

Appendix A: DERIVATION OF SOLUTION TO TWO DIMENSIONAL MODEL

The defining partial differential equations are

$$\beta\, R\, \frac{\partial C_1}{\partial t} = \left(\frac{1}{Pe}\right) \frac{\partial^2 C_1}{\partial y_D^2} - \frac{\partial C_1}{\partial x_D} \tag{A.1}$$

and

$$(1-\beta)\, R\, \frac{\partial C_2}{\partial t} = \left(\frac{\alpha}{Pe}\right) \frac{\partial^2 C_2}{\partial y_D^2} \tag{A.2}$$

where

1 = mobile region

2 = immobile region

The initial and boundary conditions are

$$C_1(x_D,y_D,0) = C_2(x_D,y_D,0) = 0 \tag{A.3}$$

$$\left.\frac{\partial C_1}{\partial y_D}\right]_{y_D=1} = 0 \tag{A.4}$$

$$C_1\Big]_{y_D=1} = C_2\Big]_{y_D=1} \tag{A.5}$$

$$\left.\frac{\partial C_1}{\partial y_D}\right]_{y_D=1} = \alpha \left.\frac{\partial C_2}{\partial y_D}\right]_{y_D=1} \tag{A.6}$$

$$\left.\frac{\partial C_2}{\partial y_D}\right]_{y_D=\infty} = 0 \tag{A.7}$$

$$C_1(0,y_D,t_D) = 1 \tag{A.8}$$

Transforming equation (A.l) with respect to t (ie. t goes to s)

$$\beta R[s\bar{C}_1 - C_1\Big]_{t=0}] = \left(\frac{1}{Pe}\right)\frac{\partial^2 \bar{C}_1}{\partial y_D^2} - \frac{\partial C_1}{\partial x_D}$$

$$s\beta R\bar{C}_1 = \left(\frac{1}{Pe}\right)\frac{\partial^2 \bar{C}_1}{\partial y_D^2} - \frac{\partial C_1}{\partial x_D} \tag{A.9}$$

Transforming equation (A.9) with respect to (x_D) (ie. x_D goes to **p)**

$$s\beta RC_1^p = \left(\frac{1}{Pe}\right)\frac{\partial^2 \bar{C}_1}{\partial y_D^2} - [pC_1^p - \bar{C}_1\Big]_{x=0}]$$

Transforming boundary condition **(A.5)** and rearranging gives

$$\left(\frac{1}{Pe}\right)\frac{\partial^2 \bar{C}_1}{\partial y_D^2} - (p + s\beta R)C_1^p = \frac{-C_o}{s} \tag{A.10}$$

Equation (A.10) can be treated as an ordinary differential equation in (y_D), and can be solved by method of undetermined coefficients. The solution to the corresponding homogeneous equation is

$$(\overline{C}_1^p)_H = Ae^{my_D} + Be^{-my_D}$$

where

$$m = \left(\frac{1}{Pe}\right)^{\frac{1}{2}} (p + s\beta R)^{\frac{1}{2}}$$

The solution of the corresponding particular problem is

$$(\overline{C}_1^p)_P = \frac{C_o}{s(p + s\beta R)}$$

The general solution is given by the sum of the homogeneous and particular solutions, thus the general solution is

$$\overline{C}_1^p = Ae^{my_D} + Be^{-my_D} + \frac{C_o}{s(p + s\beta R)} \tag{A.11}$$

Transforming equation (A.2) with respect to (t) (ie. t goes to s)

$$(1-\beta)R[s\overline{C}_2 - C_2\big]_{t=0}] = \left(\frac{\alpha}{Pe}\right)\frac{\partial^2 \overline{C}_2}{\partial y_D^2}$$

or

$$\left(\frac{\alpha}{Pe}\right)\frac{\partial^2 \overline{C}_2}{\partial y_D^2} - s(1-\beta)R\overline{C}_2 = 0$$

This is an ordinary differential equation whose solution is given by

$$\overline{C}_2 = M\,e^{zy_D} + N\,e^{-zy_D}$$

where

$$z = \left[\frac{Pe\,(1-\beta)Rs}{a}\right]^{\frac{1}{2}}$$

Boundary equation (A.4) determines that (M) is equal to zero and boundary equation (A.2) gives

$$N = (\overline{C}_1\big]_{y_D=0})e^{z(1-y_D)}$$

Thus the solution for (C_2) in (x,y,s)-space is

$$\bar{C}_2 \;=\; (\,\bar{C}_1\big]_{y_D=1})e \qquad ^{-\,y_D)} \tag{A.12}$$

Since in equation **(A.12)** only (C,) is a function of $(\mathbf{x}_D)$ equation **(A.12)** can be transformed with respect to (x_D) (ie. $\mathbf{x}_D$ goes to p) as

$$C_2^p \;=\; (\,C_1^p\big]_{y_D=1})]\; e^{z(1\,-\,y_D)} \tag{A.13}$$

To determine the unknown parameters **A** and **B** in equation **(A.ll)** both equations **(A.ll)** and **(A.13)** must be solved simultaneously. Applying boundary equation **(A.4)** gives

$$A = B$$

Applying boundary equation **(A.6)** gives

$$A \;=\; -\Big[\frac{z\,\alpha\,C_o}{s\,(p\;+\;s\,\beta R)}\Big]\,\Big[\frac{1}{(1-\alpha)M(e^{m}-e^{-m})}+z\,\alpha(e^{m}\;+\;e^{-m})\Big] \tag{A.14}$$

Using equation (A.14) in equation **(A.11)** gives the solution of the partial differential equation **(A.l)** and **(A.2)** for (C,) as

$$C_1^p \;=\; \frac{C_o}{s\;(\,p\;+\,s\,\beta R\;)}\;-\;\Big[\frac{z\,\alpha\,C_o}{s(\,p\;+\;s\,\beta R)}\Big]\;\frac{e^{\,m y_D}\;+\;e^{\,-m y_D}}{(1-\alpha)M(e^{m}-e^{-m})\;+\;z\,\alpha(e^{m}\;+\;e^{-m})} \tag{A.15}$$

Appendix 3: LISTING OF GENERATE STEP AND SAMPLE INPUT AND OUTPUT

```
C       ****************************************************************
C                              GENERATE STEP
C       ****************************************************************
C                              PROGRAM BEGINS
C
C       ****************************************************************
C
        IMPLICIT REAL*8(A-B,D-H,O-Z)
        DIMENSION CTITLE(20),T(400),ALF(14),A(400,13)
C
C
C       ******************************************************************
C
C
C       READ AND WRITE INITIAL DATA
C
        READ(5,10)CTITLE
10      FORMAT(20A4)
        WRITE (6,20) CTITLE
20      FORMAT (10X,20A4)
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C       SET NL=5 FOR SINGLE FRACTURE
C
        NL=5
        WRITE(6,30)NL
30      FORMAT('NUMBER OF NONLINEAR PARAMETERS = ',I2)
C
C       L IS THE NUMBER OF LINEAR PARAMETERS
C
        L=NL/5
        WRITE(6,35)L
35      FORMAT('NUMBER OF LINEAR PARAMETERS = ',I2)
C
C       N IS THE NUMBER OF OBSERVATIONS
C
        READ (5,*) N
        WRITE (6,40) N
40      FORMAT('NUMBER OF OBSERVATIONS = ',I4)
C
C
C
C
C       READ AND WRITE ESTIMATES OF NONLINEAR PARAMETERS
C       ALF(1)=PECLET NUMBER = PE
C       ALF(2)=BETA
```

```
C     ALF(3)=R
C     ALF(4)=ALPHA
C     ALF(5)=XD
C
      WRITE(6,42)
42    FORMAT(//,'INITIAL ESTIMATES OF NON-LINEAR PARAMTERS')
      DO 50 J=1,NL
      READ (5,*) ALF(J)
      WRITE(6,45)J,ALF(J)
50    CONTINUE
45    FORMAT(/,'ALF',I2,'=',E15.5)
C
C     READ AND WRITE DATA
C     T IS THE INDEPENDENT VARIBLE TIME
      WRITE(6,51)
51    FORMAT(//,'TIME STEPS',/)
C
      DO 52 KK=1,N
      READ(5,*) T(KK)
      WRITE(6,60)KK,T(KK)
52    CONTINUE
60    FORMAT(9X,I3,13X,F8.3)
C
C
C     SET PARAMETERS FOR STEHFEST INVERSION
C
      NN=10
      M=2424
C
C     CALCULATE MATRIX A
C
C     CALCULATE THE CONCENTRATION ONLY
C
      K=1
90    DO 100 J=1,N
      TD=T(J)
      XD=ALF(5)
      A(J,K)=XINVR1(TD,NN,M,XD,ALF)
      IF(A(J,K).LT.0.0) A(J,K)=0.0
100   CONTINUE
C
C     PRINT OUT RESULTS
      WRITE(6,153)
153   FORMAT(//,T20,'RESULTS',/,T5,'TIME',T25,'CONC.')
C
      WRITE(6,155)(T(I),A(I,1), I=1,N)
155   FORMAT(//,(E12.5,T20,E12.5))
C
      END
C
C     ************************************************************************
C     ************************************************************************
C                       FUNCTION INVERSE1
C     ************************************************************************
```

```
C                  THE STEHFEST ALGORITHM
C               ******************************
C
C
C           THIS IS THE FIRST INVERSION WHICH GOES FROM (X,Y,S)-SPACE
C           BACK TO REAL TIME. IT NEEDS THE EVALUATION OF THE
C           FUNCTION IN S-SPACE AND TO DO THIS IT CALLS SFUNCTION
C           AS A FUNCTION
C
            FUNCTION XINVR1(TD,N,M,XD,ALF)
C           THIS FUNTION COMPUTES NUMERICALLY THE LAPLACE TRNSFORM
C           INVERSE OF F(S).
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION G(50),V(50),H(25)
C
C           NOW IF THE ARRAY V(I) WAS COMPUTED BEFORE THE PROGRAM
C           GOES DIRECTLY TO THE END OF THE SUBRUTINE TO CALCULATE
C           F(S).
      IF (N.EQ.M) GO TO 17
      M=N
      DLOGTW=0.6931471805599
      NH=N/2
C
C           THE FACTORIALS OF 1 TO N ARE CALCULATED INTO ARRAY G
      G(1)=1
      DO 1 I=2,N
      G(I)=G(I-1)*I
1     CONTINUE
C
C           TERMS WITH K ONLY ARE CALCULATED INTO ARRAY H
      H(1)=2./G(NH-1)
      DO 6 I=2,NH
      FI=I
      IF(I-NH) 4,5,6
4     H(I)=FI**NH*G(2*I)/(G(NH-I)*G(I)*G(I-1))
      GO TO 6
5     H(I)=FI**NH*G(2*I)/(G(I)*G(I-1))
6     CONTINUE
C
C           THE TERMS (-1)**NH+1 ARE CALCULATED.
C           FIRST THF TERM FOR I=1
      SN=2*(NH-NH/2*2)-1
C
C           THE REST OF THE SN'S ARECALCULATED IN THE MAIN RUTINE
C
C
C           THE ARRAY V(I) IS CALCULATED.
      DO 7 I=1,N
C
C           FIRST SET V(I)=0
      V(I)=0
C
C           THE LIMITS FOR K ARE ESTABLISHED.
C           THE LOWER LIMIT IS K1=INTEG((I+1/2))
```

```
      K1=(I+1)/2
C
C          THE UPPER LIMIT IS K2=MIN(I,N/2)
      K2=I
      IF (K2-NH) 8,8,9
9     K2=NH
C
C          THE SUMMATION TERM IN V(I) IS CALCULATED.
8     DO 10 K=K1,K2
      IF (2*K-I) 12,13,12
12    IF (I-K) 11,14,11
11    V(I)=V(I)+H(K)/(G(I-K)*G(2*K-I))
      GO TO 10
13    V(I)=V(I)+H(K)/G(I-K)
      GO TO 10
14    V(I)=V(I)+H(K)/G(2*K-I)
10    CONTINUE
C
C          THE V(I) ARRAY IS FINALLY CALCULATED BY WEIGHTING
C          ACCORDING TO SN.
      V(I)=SN*V(I)
C
C          THE TERM SN CHANGES ITS SIGN EACH ITERATION
      SN=-SN
7     CONTINUE
C
C          THE NUMERICAL APPROXIMATION IS CALCULATED
17    XINVR1=0.
      A=DLOGTN/TD
      DO 15 I=1,N
      ARG-A*I
      XINVR1=XINVR1+V(I)*SFUNC(ARG,I,XD,ALF)
15    CONTINUE
      XINVR1=XINVR1*A
18    RETURN
      END


C
C     ***************************************************
C     ***************************************************
C                   FUNCTION SFUNCTION
C     ***************************************************
C
      FUNCTION SFUNC(S,I,XD,ALF)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     THIS FUNCTION SIMPLY EVALUATES THE DESIRED FUNCTION IN
C     S-SPACE BY CALLING FOR A SECOND INVERSION USING THE
C     STEHFAST ALGORITHM. IT SHOULD BE NOTED THAT THE VARIBLE
C     THAT IS TRANSFORMED FROM GOING FROM S-SPACE TO P-SPACE
C     IS XD THUS S IS A CONSTANT FROM THIS POINT ON.
C
```

```
C
      XD=XD
      S=S
      N=10
      M=2424
      SFUNC =XINVR2(XD,N,M,S,ALF)
      RETURN
      END
C     ***************************************************************
C     ***************************************************************
C                FUNCTION INVERSE2
C     ***************************************************************
C                THE STEHFEST ALGORITHM
C             ***************************
C
C
C          THIS FUNCTION WILL INVERT FROM (P,Y,S)-SPACE TO (X,Y,S)-
C          SPACE WITH XD BEING THE VARIBLE OF INVERSION. THE FUNCTION
C          NEEDS AN EXPRESSION FOR THE FUNCTION IN P-SPACE AND
C          THIS IS DONE BY CALLING PFUNC(XD,S)
C
           FUNCTION XINVR2(XD,N,M,S,ALF)
C          THIS FUNTION COMPUTES NUMERICALLY THE LAPLACE TRNSFORM
           INVERSE OF F(S)
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION G(50),V(50),H(25)
C
C          NOW IF THE ARRAY V(I) WAS COMPUTED BEFORE THE PROGRAM
C          GOES DIRECTLY TO THE END OF THE SUBRUTINE TO CALCULATE
C          F(S).
      IF (N.EQ.M) GO TO 17
      M=N
      DLOGTW=0.6931471805599
      NH=N/2
C
C          THE FACTORIALS OF 1 TO N ARE CALCULATED INTO ARRAY G.
      G(1)=1
      DO 1 I=2,N
      G(I)=G(I-1)*I
1     CONTINUE
C
C          TERMS WITH K ONLY ARE CALCULATED INTO ARRAY H.
      H(1)=2./G(NH-1)
      DO 6 I=2,NH
      FI=I
      IF(I-NH) 4,5,6
4     H(I)=FI**NH*G(2*I)/(G(NH-I)*G(I)*G(I-1))
      GO TO 6
5     H(I)=FI**NH*G(2*I)/(G(I)*G(I-1))
6     CONTINUE
C
C          THE TERMS (-1)**NH+1 ARE CALCULATED.
C          FIRST THE TERM FOR I=1
      SN=2*(NH-NH/2*2)-1
```

```
C
C         THE REST OF THE SN'S ARECALCULATED IN THE MAIN RUTINE
C
C
C         THE ARRAY V(I) IS CALCULATED.
      DO 7 I=1,N
C
C         FIRST SET V(I)=0
      V(I)=0
C
C         THE LIMITS FOR K ARE ESTABLISHED.
C         THE LOWER LIMIT IS K1=INTEG((I+1/2))
      K1=(I+1)/2
C
C         THE UPPER LIMIT IS K2=MIN(I,N/2)
      K2=I
      IF (K2-NH) 8,8,9
9     K2=NH
C
C         THE SUMMATION TERM IN V(I) IS CALCULATED.
8     DO 10 K=K1,K2
      IF (2*K-I) 12,13,12
12    IF (I-K) 11,14,11
11    V(I)=V(I)+H(K)/(G(I-K)*G(2*K-I))
      GO TO 10
13    V(I)=V(I)+H(K)/G(I-K)
      GO TO 10
14    V(I)=V(I)+H(K)/G(2*K-I)
10    CONTINUE
C
C         THE V(I) ARRAY IS FINALLY CALCULATED BY WEIGHTING
C         ACCORDING TO SN.
      V(I)=SN*V(I)
C
C         THE TERM SN CHANGES ITS SIGN EACH ITERATION.
      SN=-SN
7     CONTINUE
C
C         THE NUMERICAL APPROXIMATION IS CALCULATED.
17    XINVR2=0.0
      A=DLOGTW/XD
      DO 15 I=1,N
      ARG=A*I
      XINVR2=XINVR2+V(I)*PFUNC(ARG,I,S,ALF)
15    CONTINUE
      XINVR2 = XINVR2*A
18    RETURN
      END


C     ****************************************************************
C     ****************************************************************
C
```

```
C                FUNCTION   PFUNC
C      ******************************************************************
C
       FUNCTION PFUNC(P,I,S,ALF)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION F(4),ALF(20)
       YD=0.5
C
C      INITIALLY SET ALL VALUES TO ZERO
       DO 1 I=1,4
       F(I)=0.0
1      CONTINUE
       FP=0.0
C
C      CALCULATE THE COMPONENTS OF THE CONCENTRATION FUNCTION
C
       Z=((ALF(1)*(1.-ALF(2))*ALF(3)*S)/ALF(4))**0.5
       XM=((P+S*ALF(2)*ALF(3))/ALF(1))**0.5
       ZM=Z/XM
       F(1)=1/(S*(P+(S*ALF(2)*ALF(3))))
       F(2)=ZM*ALF(4)/(S*(P+(S*ALF(2)*ALF(3))))
       F(3)=(1.0+(ZM*ALF(4))-ALF(4))*EXP(XM) +
      *(ALF(4)+(ZM*ALF(4))-1.0)*EXP(-XM)
       F(4)=EXP(XM*YD)+EXP(-XM*YD)
       FP=F(1)-(F(2)*F(4)/F(3))
       PFUNC=FP
       RETURN
       END
THE SAMPLE INPUT FOLLOWS ****************************************
***************************************************************
TEST: GENERATE A STEP
20
0.19972E-05
0.0070049
0.16531E06
1
2.
3.0
4.0
5.
7.
9.
11
13.
15.
17.
19
21
23.
25.
27.
29.
THE SAMPLE OUTPUT FOLLOWS**************************************************
*****************************************************************
```

TEST: GENERATE A STEP
NUMBER OF NONLINEAR PARAMETERS = 5
NUMBER OF LINEAR PARAMETERS = 1
NUMBER OF OBSERVATIONS = 20

INITIAL ESTIMATES OF NON-LINEAR PARAMTERS

ALF 1= 0.20140e+00

ALF 2- 0.49942e+00

ALF 3= 0.19972e-05

ALF 4= 0.70049e-02

ALF 5= 0.16531e+06

TIME STEPS

1	0.010
2	0.160
3	0.500
4	1.000
5	2.000
6	3.000
7	4.000
8	5.000
9	7.000
10	9.000
11	11.000
12	13.000
13	15.000
14	17.000
15	19.000
16	21.000
17	23.000
18	25.000
19	27.000
20	29.000

RESULTS

TIME	CONC
0.10000e-01	0. e+00
0.16000e+00	0. e+00
0.50000e+00	0.12592e+00
0.10000e+01	0.32652e+00
0.20000e+01	0.51122e+00
0.30000e+01	0.59747e+00
0.40000e+01	0.64972e+00

0.50000e+01	0.68576e+00
0.70000e+01	0.73350e+00
0.90000e+01	0.76453e+00
0.11000e+02	0.78677e+00
0.13000e+02	0.80371e+00
0.15000e+02	0.81716e+00
0.17000e+02	0.82818e+00
0.19000e+02	0.83743e+00
0.21000e+02	0.84533e+00
0.23000e+02	0.85217e+00
0.25000e+02	0.85818e+00
0.27000e+02	0.86352e+00
0.29000e+02	0.86829e+00

Appendix C: LISTING OF MAIN OF GENERATE.FINSTEP AND SAMPLE INPUT

```
C      ************************************************************
C                         GENERATE.FINSTEP
C      ************************************************************
C                          PROGRAM BEGINS
C
C      ************************************************************
C
       IMPLICIT REAL*8(A-B,D-H,O-Z)
       DIMENSION CTITLE(20),T(400),ALF(14),A(400,13),B(400,13)
C
C
C      ****************************************************************
C
C
C      READ AND WRITE INITIAL DATA
C
       READ(5,10)CTITLE
10     FORMAT(20A4)
       WRITE (6,20) CTITLE
20     FORMAT (10X,20A4)
C
C      NL IS THE NUMBER OF NONLINEAR PARAMETERS
C      SET NL=5 FOR SINGLE FRACTURE
C
       NL=5
       WRITE(6,30)NL
30     FORMAT('NUMBER OF NONLINEAR PARAMETERS = ',I2)
C
C      L IS THE NUMBER OF LINEAR PARAMETERS
C
       L=NL/5
       WRITE(6,35)L
35     FORMAT('NUMBER OF LINEAR PARAMETERS = ',I2)
C
C      N IS THE NUMBER OF OBSERVATIONS
C
       READ (5,*) N
       WRITE (6,40) N
C
C      DELT IS THE DURATION OF THE STEP INPUT
C
40     FORMAT('NUMBER OF OBSERVATIONS = ',I4)
       READ (5,*)DELT
       WRITE (6,41) DELT
41     FORMAT('DURATION OF STEP INPUT = ',F6.2)
C
```

```
C
C
C
C      READ AND WRITE ESTIMATES OF NONLINEAR PARAMETERS
C      ALF(1)=PECLET NUMBER = PE
C      ALF(2)=BETA
C      ALF(3)=R
C      ALF(4)=ALPHA
C      ALF(5)=XD
C
       WRITE(6,42)
42     FORMAT(//,'INITIAL ESTIMATES OF NON-LINEAR PARAMTERS')
       DO 50 J=1,NL
       READ (5,*) ALF(J)
       WRITE(6,45)J,ALF(J)
50     CONTINUE
45     FORMAT(/,'ALF',I2,'=',E15.5)
C
C      READ AND WRITE DATA
C      T IS THE INDEPENDENT VARIBLE TIME
       WRITE(6,51)
51     FORMAT(//,'TIME STEPS',/)
C
       DO 52 KK=1,N
       READ(5,*) T(KK)
       WRITE(6,60)KK,T(KK)
52     CONTINUE
60     FORMAT(9X,I3,13X,F8.3)
C
C
C      SET PARAMETERS FOR STEHFEST INVERSION
C
       NN=10
       M=2424
C
C      CALCULATE MATRIX A
C
C      CALCULATE THE CONCENTRATION ONLY
C      USE SUPERPOSITION
C
       K=1
90     DO 100 J=1,N
       XD=ALF(5)
       TD=T(J)
       A(J,K)=XINVR1(TD,NN,M,XD,ALF)
       IF(A(J,K).LT.0.0) A(J,K)=0.0
       IF(TD.LE.DELT) GO TO 100
       TD=T(J)-DELT
       B(J,K)=XINVR1(TD,NN,M,XD,ALF)
       A(J,K)=A(J,K)-B(J,K)
       IF(A(J,K).LT.0.0) A(J,K)=0.0
100    CONTINUE
C
C      PRINT OUT RESULTS
```

```
      WRITE(6,153)
153   FORMAT(//,T20,'RESULTS',/,T5,'TIME',T25,'CONC.')
C
      WRITE(6,155)(T(I),A(I,1), I=1,N)
155   FORMAT(//,(E12.5,T20,E12.5))
C
      END
A SAMPLE INPUT FOLLOWS ****************************************
**************************************************************
TEST: GENERATE A FINITE-STEP
27
10
0.19972E-05
0.0070049
0.16531E06
1.
2.
3
4.0
5
6.
7
8.
9
10
11.
12.
13
14.
15
16
17.
18
19
20.
22
24
26.
28.
```

Appendix D: LISTING OF SFUNCTION FOR GENERATE. **SPIKE**

```
C
C      ***************************************************************
C      ***************************************************************
C                    FUNCTION SFUNCTION
C      ***************************************************************
C
       FUNCTION SFUNC(S, I, XD, ALF)
       IMPLICIT REAL*8(A-H, O-Z)
C
C      THIS FUNCTION SIMPLY EVALUATES THE DESIRED FUNCTION IN
C      S-SPACE BY CALLING FOR A SECOND INVERSION USING THE
C      STEHFAST ALGORITHM. IT SHOULD BE NOTED THAT THE VARIBLE
C      THAT IS TRANSFORMED FROM GOING FROM S-SPACE TO P-SPACE
C      IS XD THUS S IS A CONSTANT FROM THIS POINT ON.
C
C      NOTE THE TIME DERIVATIVE NECESSARY FOR THE SPIKE INPUT
C      IS ACHIEVED BY MULTIPLING THE FUNCTION BY S
C
       XD=XD
       S=S
       N=10
       M=2424
       SFUNC =S*XINVR2(XD, N, M, S, ALF)
       RETURN
       END
```

Appendix E: CALCULATION OF THE DERIVATIVES OF THE TWO-DIMENSIONAL MODEL

The calculation of the derivative of the solution equation **(6.39)** with respect to the five dimensionless variables was done in (p,y,s)-space. These values were then doubly inverted and entered to the curve fitting program as needed. The analytic expression for the solution is

$$C_1^p = \frac{C_o}{s\,(p + s\beta R\,)} - \left[\frac{z\alpha C_o}{s(p + s\beta R)}\right]\frac{e^{my_D} + e^{-my_D}}{(1-\alpha)M(e^{m} - e^{-m}) + z\alpha(e^{m} + e^{-m})} \tag{E.1}$$

To simplify the calculation of the derivatives, the solution has been separated into the following functions

$$f_1 = \frac{1}{s(p + s\beta R)} \tag{E.2}$$

$$f_2 = \frac{(\frac{z}{m})\alpha}{s(p + s\beta R)} \tag{E.3}$$

$$f_3 = \left\{[1 + (\frac{z}{m})\alpha - \alpha]e^{m} + [\alpha + (\frac{z}{m})\alpha - 1]e^{-m}\right\} \tag{E.4}$$

$$f_4 = (e^{my_D} + e^{-my_D}) \tag{E.5}$$

The derivative of the solution with respect to any of the dimensionless variables can be calculated from the derivative of the above simplified functions with respect to the variables. The following notation greatly simplifies the generalization of the calculations. Let

$Pe = \alpha_1$

$\beta = \alpha_2$

$R = \alpha_3$

$\alpha = \alpha_4$

$x_D = \alpha_5$

In general the derivative of the solution with respect to any dimensionless variable (α_j) is given by

$$\frac{\partial F}{\partial \alpha_j} = f_2 f_3 f_4 \frac{\partial f_1}{\partial \alpha_j} + f_1 f_3 f_4 \frac{\partial f_2}{\partial \alpha_j} + f_1 f_2 f_4 \frac{\partial f_3}{\partial \alpha_j} + f_1 f_2 f_3 \frac{\partial f_4}{\partial \alpha_j} \qquad \text{(E.6)}$$

All that is needed to complete equation (E.6) is an evaluation of $\left(\frac{\partial f_i}{\partial \alpha_j}\right)$. This partial derivative term is denoted by PD(i,j), where

i = Function number

j = Variable number

For (α_1) the partial derivatives have been calculated as

$$PD(1,1) = 0$$

$$PD(2,1) = \frac{\alpha_4 z}{m^3 \alpha_1 s}$$

$$PD(3,1) = \frac{m\alpha_4}{2\alpha_1}\left[\left(\frac{2z}{m^2} + 1 - \frac{z}{m} - \frac{1}{\alpha_4}\right)e^{m}\left(\frac{2z}{m^2} + 1 + \frac{z}{m} - \frac{1}{\alpha_4}\right)e^{-m}\right]$$

$$PD(4,1) = \frac{m y_D}{2\alpha_1}\left[e^{-m y_D} - e^{m y_D}\right]$$

For (α_2) the partial derivatives are

$$PD(1,2) = -\frac{\alpha_3}{m^4 \alpha_1^2}$$

$$PD(2,2) = -\frac{\alpha_3}{2m^3}\left[\frac{1}{z} + \frac{z\alpha_4}{\alpha_1^2 m^2}\right]$$

$$PD(3,2) = \frac{s\alpha_3}{2m\alpha_1}\left[\left[1 + \frac{z\alpha_4}{m} - \frac{\alpha_1^2}{z} - \frac{\alpha_4 z}{m^2} - \alpha_4\right]e^{m} - \left[\alpha_4 + \frac{z\alpha_4}{m} + \frac{\alpha_1^2}{z} + \frac{\alpha_4 z}{m^2} - 1\right]e^{-m}\right]$$

$$PD(4,2) = \frac{\alpha_3 s y_D}{2\alpha_1 m}\left(e^{m y_D} - e^{-m y_D}\right)$$

For (α_3) the partial derivatives are

$$PD(1,3) = -\frac{\alpha_2}{m^4 \alpha_1^2}$$

$$PD(2,3) = \frac{1}{2m^3}\left[\frac{(1-\alpha_2)}{z} - \frac{3z\alpha_4\alpha_2}{\alpha_1^2 m^2}\right]$$

$$PD(3,3) = \frac{s\alpha_2}{2m\alpha_1}\left\{\left[1 + \frac{z\alpha_4}{m} - \frac{(1-\alpha_2)\alpha_1^2}{\alpha_2 z} - \frac{\alpha_4 z}{m^2} - \alpha_4\right]e^{m} - \left[\alpha_4 + \frac{z\alpha_4}{m} + \frac{(1-\alpha_2)\alpha_1^2}{\alpha_2 z} + \frac{\alpha_4 z}{m^2} - 1\right]e^{-m}\right\}$$

$$PD(4,3) = \frac{\alpha_2 s y_D}{2\alpha_1 m}\left(e^{m y_D} - e^{-m y_D}\right)$$

For (α_4) the partial derivatives are

$$PD(1,4) = 0$$

$$PD(2,4) = \frac{z}{\alpha_1 m^3}$$

$$PD(3,4) = [\frac{1}{2}(\frac{z}{m}) - 1]e^{m} + [1 - \frac{1}{2}(\frac{z}{m})]e^{-m}$$

$$PD(4,4) = 0$$

The above shows how the derivative of the solution can be calculated for all the dimensionless variables except (x_D). This derivative was calculated using the following property of the Laplace transform

$$\frac{\partial F}{\partial x_D} = p(L^{-1} f^{p})$$

Thus the derivative with respect to (x_D) was calculated by multilpying the solution by **p** before it was numerically inverted from (p,y,s)-space to (x,y,s)-space.

Appendix F: LISTING OF CURVEFIT

```
C       ************************************************************
C       ************************************************************
C                            PROGRAM BEGINS
C       ************************************************************
C                            MAIN PROGRAM
C
C       ************************************************************
C
        IMPLICIT REAL*8(A-B,D-H,O-Z)
        DIMENSION CTITLE(20),T(400),Y(400),ALF(14),BETA(7),W(400),
       *INC(14,8),A(400,13),C(400)
C
C
C       ***************************************************************
C
C       SET PARAMETERS FOR VARPRO
        EXTERNAL ADA
        NMAX=400
        IPRINT=1
C
C       READ AND WRITE INITIAL DATA
C
        READ(5,10)CTITLE
10      FORMAT(20A4)
        WRITE (6,20) CTITLE
20      FORMAT (10X,20A4)
C
C       NL IS THE NUMBER OF NONLINEAR PARAMETERS
C
C       SET NL=5 FOR A SINGLE FRACTURE
        NL=5
30      FORMAT (' NUMBER OF NONLINEAR PARAMETERS = ',I2)
C
C       L IS THE NUMBER OF LINEAR PARAMETERS
C
        L=NL/5
        WRITE(6,35)L
35      FORMAT('NUMBER OF LINEAR PARAMETERS = ',I2)
C
C       N IS THE NUMBER OF OBSERVATIONS
C
        READ (5,*) N
        WRITE (6,40) N
40      FORMAT (' NUMBER OF OBSERVATIONS = ',I4)
C
C       IV IS THE NUMBER OF INDEPENDENT VARIBLES
```

```
C
      IV = 1
C
C     SET CONSTANTS
C
      LPP2=L+NL+2
      LP=L+1
      WRITE(6,41)IV,LPP2,NMAX,LP
41    FORMAT(T20,'CONSTANTS'/(I3))
C
C     READ AND WRITE ESTIMATES OF NONLINEAR PARAMETERS
C     ALF(1)=PECLET NUMBER = PE
C     ALF(2)=BETA
C     ALF(3)=R
C     ALF(4)=ALPHA
C     ALF(5)=XD
C
      WRITE(6,42)
42    FORMAT(T20,'INITIAL ESTIMATES OF NONLINEAR PARAMETERS')
      DO 50 J=1,NL
      READ (5,*) ALF(J)
      WRITE(6,45)J,ALF(J)
50    CONTINUE
45    FORMAT('ALF',I2,'=',E15.5)
C
C     READ AND WRITE DATA
C     T IS THE INDEPENDENT VARIBLE TIME
C     Y IS THE DEPENDENT VARIBLE CONCENTRATION
C
      WRITE(6,51)
51    FORMAT(T6,'DATA NO ',T23,'TIME',T40,'CONCENTRATION')
      DO 52 KK=1,N
      READ(5,*) T(KK),Y(KK)
      WRITE(6,60)KK,T(KK),Y(KK)
52    CONTINUE
60    FORMAT(9X,I3,13X,F8.3,10X,F9.3)
C
C     SET W(I) THE WEIGHTING PARAMETERS
C
      DO 70 I=1,N
      W(I)=1.0
70    CONTINUE
C
C     CALL VARPRO TO DETERMINE THE BEST FIT VALUES
C
      CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR)
C
C     PRINT FINAL RESULTS AND CALCULATE BEST FIT RETURN CURVE USING
C     TIME POINTS IN ORIGINAL DATA. PLOT BOTH CALCULATED AND OBSERVED
C     RETURN CURVES
C
      WRITE (6,80)
80    FORMAT(T20,'BEST FIT VALUES')
```

```
      WRITE (6,85)(I,ALF(I), I=1,NL)
85    FORMAT('ALF',I2,'= ',E15.5)
      WRITE(6,90) (I,BETA(I), I=1,L)
90    FORMAT('BETA',I2,'= ',E15.5)
C
C     CALCULATE CONCENTRATION BY CALLING ADA
C
      ISEL=1
      LP=L+1
      WRITE(6,130)
130   FORMAT(1H0,'CALCULATED CONCENTRATIONS',/,T5,'TIME',
     *T15,'CONCENTRATION',//)
      CALL ADA(LP,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
      DO 140 I=1,N
      C(I)=BETA(1)*A(I,1)
      WRITE(6,155) T(I),C(I)
140   CONTINUE
155   FORMAT(1X,2F10.4)
100   STOP
      END
C
C
C     ************************************************************
C     ************************************************************
C
C                 SUBROUTINE ADA
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)
           L=LP-1
      NN=10
      M=2424
C
C     SET INCIDENCE MATRIX, INC (K,J) = 1 IF ALPHA(K) APPEARS IN PHI(J)
C
C
C     SKIP UNLESS ISEL IS EQUAL TO 1
C
      IF (ISEL.EQ.2) GO TO 90
      IF (ISEL.EQ.3) GO TO 110
      DO 10 J=1,L
      DO 10 K=1,NL
      INC(K,L)=0
      IF((K+4)/2.LT.5) INC(K,1)=1
      IF((K+4)/2.GE.5) INC(K,2)=1
10    CONTINUE
C
C     CALCULATE MATRIX A
C
C     CALCULATE COLUMN BY COLUMN
C     FIRST CALCULATE THE CONCENTRATION ONLY
```

```
C
90    DO 100 J=1,N
      DO 99 K=1,L
      TD=T(J)
      ISET=1
      XD=ALF(5*K)
      KK=K
      A(J,K)=XINVR1(TD,NN,M,XD,ALF,ISET,KK)
      IF(A(J,K).LT.0.0) A(J,K)=0.0
99    CONTINUE
100   CONTINUE
C     SKIP THE CALCULATION OF DERIVATIVES IF ISEL=2
C
      IF (ISEL.EQ.2) GO TO 360
C
C     CALCULATION OF THE DERIVATIVES
C     FIRST CALCULATE DERIVATIVE WITH RESPECT TO ALF(1) (PE)
C
110   DO 150 J=1,N
      DO 150 K=1,L
      TD=T(J)
      XD=ALF(5*K)
      ISET=2
      IB=3
      A(J,IB)=XINVR1(TD,NN,M,XD,ALF,ISET,K)
      IF(A(J,K).LE.0.0) A(J,IB)=0.0
150   CONTINUE
C
C     CALCULATE DERIVATE WITH RESPECT TO ALF(2),(IBETA)
C
      DO 200 J=1,N
      DO 200 K=1,L
      TD=T(J)
      XD=ALF(5*K)
      ISET = 3
      IB=4
      A(J,IB)=XINVR1(TD,NN,M,XD,ALF,ISET,K)
        IF(A(J,K).LE.0.0) A(J,IB)=0.0
200   CONTINUE
C
C     CALCULATE DERIVATIVE WITH RESPECT TO ALF(3),(R
C
      DO 250 J=1,N
      DO 250 K=1,L
      TD=T(J)
      XD=ALF(5*L)
      ISET=4
      IB=5
      A(J,IB)= XINVR1(TD,NN,M,XD,ALF,ISET,K)
        IF(A(J,K).LE.0.0) A(J,IB)=0.0
250   CONTINUE
C
C     CALCULATE DERIVATIVE WITH RESPECT TO ALF(4),(ALPHA)
C
```

```
      DO 300 J=1,N
      DO 300 K=1,L
      TD=T(J)
      XD=ALF(5*K)
      ISET=5
      IB= 6
      A(J,IB)= XINVR1(TD,NN,M,XD,ALF,ISET,K)
        IF(A(J,K).LE.0.0) A(J,IB)=0.0
300   CONTINUE
C
C     CALCULATE DERIVATIVE WITH RESPECT TO ALF(5);(XD)
C
      DO 350 J=1,N
      DO 350 K=1,L
      TD=T(J)
      XD=ALF(5*K)
      ISET=6
      IB=7
      A(J,IB)= XINVR1(TD,NN,M,XD,ALF,ISET,K)
        IF(A(J,K).LE.0.0) A(J,IB)=0.0
350   CONTINUE
360   CONTINUE
500   RETURN
      END
C
C     ********************************************************************
C     ********************************************************************
C                  FUNCTION INVERSE1
C     ********************************************************************
C                  THE STEHFEST ALGORITHM
C               ****************************
C
C
C          THIS IS THE FIRST INVERSION WHICH GOES FROM (X,Y,S)-SPACE
C          BACK TO REAL TIME. IT NEEDS THE EVALUATION OF THE
C          FUNCTION IN S-SPACE AND TO DO THIS IT CALLS SFUNCTION
C          AS A FUNCTION
C
           FUNCTION XINVR1(TD,N,M,XD)
C          THIS FUNTION COMPUTES NUMERICALLY THE LAPLACE TRNSFORM
C          INVERSE OF F(S).
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION G(50),V(50),H(25)
C
C          NOW IF THE ARRAY V(I) WAS COMPUTED BEFORE THE PROGRAM
C          GOES DIRECTLY TO THE END OF THE SUBRUTINE TO CALCULATE
C          F(S).
      IF (N.EQ.M) GO TO 17
      M=N
      DLOGTW=0.6931471805599
      NH=N/2
C
C          THE FACTORIALS OF 1 TO N ARE CALCULATED INTO ARRAY G
      G(1)=1
```

```
      DO 1 I=2,N
      G(I)=G(I-1)*I
1     CONTINUE
C
C          TERMS WITH K ONLY ARE CALCULATED INTO ARRAY H.
      H(1)=2./G(NH-1)
      DO 6 I=2,NH
      FI=I
      IF(I-NH) 4,5,6
4     H(I)=FI**NH*G(2*I)/(G(NH-I)*G(I)*G(I-1))
      GO TO 6
5     H(I)=FI**NH*G(2*I)/(G(I)*G(I-1))
6     CONTINUE
C
C          THE TERMS (-1)**NH+1 ARE CALCULATED.
C          FIRST THE TERM FOR I=1
      SN=2*(NH-NH/2*2)-1
C
C          THE REST OF THE SN'S ARECALCULATED IN THE MAIN RUTINE.
C
C
C          THE ARRAY V(I) IS CALCULATED.
      DO 7 I=1,N
C
C          FIRST SET V(I)=0
      V(I)=0.

C          THE LIMITS FOR K ARE ESTABLISHED.
C          THE LOWER LIMIT IS K1=INTEG((I+1/2))
      K1=(I+1)/2
C
C          THE UPPER LIMIT IS K2=MIN(I,N/2)
      K2=I
      IF (K2-NH) 8,8,9
9     K2=NH
C
C          THE SUMMATION TERM IN V(I) IS CALCULATED.
8     DO 10 K=K1,K2
      IF (2*K-I) 12,13,12
12    IF (I-K) 11,14,11
11    V(I)=V(I)+H(K)/(G(I-K)*G(2*K-I))
      GO TO 10
13    V(I)=V(I)+H(K)/G(I-K)
      GO TO 10
14    V(I)=V(I)+H(K)/G(2*K-I)
10    CONTINUE
C
C          THE V(I) ARRAY IS FINALLY CALCULATED BY WEIGHTING
C          ACCORDING TO SN.
      V(I)=SN*V(I)
C
C          THE TERM SN CHANGES ITS SIGN EACH ITERATION.
      SN=-SN
7     CONTINUE
```

```
C
C          THE NUMERICAL APPROXIMATION IS CALCULATED.
17     XINVR1=0.
       A=DLOGTW/TD
       DO 15 I=1,N
       ARG=A*I
       XINVR1=XINVR1+V(I)*SFUNC(ARG,I,XD)
15     CONTINUE
       XINVR1=XINVR1*A
18     RETURN
       END


C      ************************************************************
C      ************************************************************
C                      FUNCTION SFUNCTION
C      ************************************************************
C
       FUNCTION SFUNC(S,I,XD)
       IMPLICIT REAL*8(A-H,O-Z)
C
C      THIS FUNCTION SIMPLY EVALUATES THE DESIRED FUNCTION IN
C      S-SPACE BY CALLING FOR A SECOND INVERSION USING THE
C      STEHFAST ALGORITHM  IT SHOULD BE NOTED THAT THE VARIBLE
C      THAT IS TRANSFORMED FROM GOING FROM S-SPACE TO P-SPACE
C      IS XD THUS S IS A CONSTANT FROM THIS POINT ON.
C
C      THE S FUNCTION IS MULTIPLIED BY S IN ORDER TO CALCULATE THE
C      TIME DERIVATIVE THUS PRODUCING A SPIKE INPUT
C
       XD=XD
       S=S
       N=10
       M=2424
       SFUNC =S*XINVR2(XD,N,M,S)
       RETURN
       END

C      *************************************************************
C      *************************************************************
C                   FUNCTION INVERSE2
C      **************************************************************
C                    THE STEHFEST ALGORITHM
C                ****************************
C
C
C          THIS FUNCTION WILL INVERT FROM P-SPACE TO S-SPACE
C          WITH XD BEING THE VARIBLE OF INVERSION  THE FUNCTION
C          NEEDS AN EXPRESSION FOR THE FUNCTION IN P-SPACE AND
C          THIS IS DONE BY CALLING PFUNC(XD,S)
C
           FUNCTION XINVR2(XD,N,M,S)
C          THIS FUNTION COMPUTES NUMERICALLY THE LAPLACE TRNSFORM
```

```
C           INVERSE OF F(S).
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION G(50),V(50),H(25)
C
C           NOW IF THE ARRAY V(I) WAS COMPUTED BEFORE THE PROGRAM
C           GOES DIRECTLY TO THE END OF THE SUBRUTINE TO CALCULATE
C           F(S).
      IF (N.EQ.M) GO TO 17
      M=N
      DLOGTW=0.6931471805599
      NH=N/2
C
C           THE FACTORIALS OF 1 TO N ARE CALCULATED INTO ARRAY G.
      G(1)=1
      DO 1 I=2,N
      G(I)=G(I-1)*I
1     CONTINUE
C
C           TERMS WITH K ONLY ARE CALCULATED INTO ARRAY H.
      H(1)=2./G(NH-1)
      DO 6 I=2,NH
      FI=I
      IF(I-NH) 4,5,6
3     H(I)=FI**NH*G(2*I)/(G(NH-I)*G(I)*G(I-1))
      GO TO 6
      H(I)=FI**NH*G(2*I)/(G(I)*G(I-1))
6     CONTINUE
C
C           THE TERMS (-1)**NH+1 ARE CALCULATED
C           FIRST THE TERM FOR I=1
      SN=2*(NH-NH/2*2)-1
C
C           THE REST OF THE SN'S ARECALCULATED IN THE MAIN RUTINE
C
C
C           THE ARRAY V(I) IS CALCULATED.
      DO 7 I=1,N
C
C           FIRST SET V(I)=0
      V(I)=0.
C
C           THE LIMITS FOR K ARE ESTABLISHED.
C           THE LOWER LIMIT IS K1=INTEG((I+1/2))
      K1=(I+1)/2
C
C           THE UPPER LIMIT IS K2=MIN(I,N/2)
      K2=I
      IF (K2-NH) 8,8,9
9     K2=NH
C
C           THE SUMMATION TERM IN V(I) IS CALCULATED
8     DO 10 K=K1,K2
      IF (2*K-I) 12,13,12
12    IF (I-K) 11,14,11
```

```
11    V(I)=V(I)+H(K)/(G(I-K)*G(2*K-I))
      GO TO 10
13    V(I)=V(I)+H(K)/G(I-K)
      GO TO 10
14    V(I)=V(I)+H(K)/G(2*K-I)
10    CONTINUE
C
C          THE V(I) ARRAY IS FINALLY CALCULATED BY WEIGHTING
C          ACCORDING TO SN.
      V(I)=SN*V(I)
C
C          THE TERM SN CHANGES ITS SIGN EACH ITERATION
      SN=-SN
7     CONTINUE
C
C          THE NUMERICAL APPROXIMATION IS CALCULATED
17    XINVR2=0.0
      A=DLOGTW/XD
      DO 15 I=1,N
      ARG=A*I
      XINVR2=XINVR2+V(I)*PFUNC(ARG,I,S)
15    CONTINUE
      XINVR2 = XINVR2*A
16    RETURN
      END


C     ******************************************************************
C     ******************************************************************
C
C                FUNCTION   PFUNC
C     ******************************************************************
C
      FUNCTION PFUNC(P,I,S,ALF,ISET,K)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PDF(4,4),DF(5),F(4),ALF(20)
      YD=0.5
C
C     INITIALLY SET ALL VALUES TO ZERO
      DO 1 I=1,4
      DO 1 J=1,4 .
      PDF(I,J)=0.0
      DF(I)=0.0
      F(I)=0.0
      CONTINUE
      DF(5)=0.0
C
C     CALCULATE THE COMPONENTS OF THE CONCENTRATION FUNCTION
C     NOTE THAT IF ISET=1 THIS IS ALL THAT IS NEEDED
C
      Z=((ALF(1)*(1.-ALF(2))*ALF(3)*S)/ALF(4))**0.5
      XM=((P+S*ALF(2)*ALF(3))/ALF(1))**0.5
      ZM=Z/XM
```

```
      F(1)=1/(S*(P+(S*ALF(2)*ALF(3))))
      F(2)=ZM*ALF(4)/(S*(P+(S*ALF(2)*ALF(3))))
      F(3)=(1.0+(ZM*ALF(4))-ALF(4))*EXP(XM) +
     *(ALF(4)+(ZM*ALF(4))-1.0)*EXP(-XM)
      F(4)=EXP(XM*YD)+EXP(-XM*YD)
      FP=F(1)-(F(2)*F(4)/F(3))
      IF(ISET.EQ.1)GO TO 15
C
C     CALCULATE THE PARTIAL DERIVATIVES AND STORE IN PDF
C
C     FIRST CALCULATE WITH RESPECT TO ALF(1)
C
      PDF(1,1)=0.0
      PDF(2,1)=(ALF(4)*Z/((XM**3.)*(ALF(1)**2.)))*(1./S)
      P1=((2.*Z/(XM**2.))+1.-ZM-(1./ALF(4)))*EXP(XM)
      P2=((2.*Z/(XM**2.))+1.+ZM-(1./ALF(4)))*EXP(-XM)
      PDF(3,1)=(XM*ALF(4)/(2.*ALF(1)))*(P1+P2)
      PDF(4,1)=(XM*YD/(2.*ALF(1)))*(EXP(-XM*YD)-EXP(XM*YD))
C
C     CALCULATE WITH RESPECT TO ALF(2)
C
      PDF(1,2)=-ALF(3)/((XM**4.)*(ALF(1)**2.))
      P1=(3.*ALF(4)*Z)/((XM*ALF(1))**2.)
      PDF(2,2)=-ALF(3)/(2.*(XM**3.))*((1./Z)+P1)
      P2=(1.+ZM*ALF(4)-((ALF(1)**2.)/
     !(Z))-ZM*ALF(4)/XM-ALF(4))*EXP(XM)
      P3=(ALF(4)+ZM*ALF(4)+((ALF(1)**2.)/
     !(Z))+ZM*ALF(4)/XM-1.)*EXP(-XM)
      PDF(3,2)=((ALF(3)*S)/(2.*ALF(1)*XM))*(P2-P3)
      PDF(4,2)=(ALF(3)*S*YD/(2.*ALF(1)*XM))*(EXP(XM*YD)-EXP(-XM*YD))
C
C     CALCULATE WITH RESPECT TO ALF(3)

      PDF(1,3)=-ALF(2)/((XM**4.)*(ALF(1)**2.))
      P1=(3.*Z*ALF(2)*ALF(4)/((XM*ALF(1))**2.))
      PDF(2,3)=(1./(2.*(XM**3.)))*(((1.-ALF(2))/Z)-P1)
      P2=1.+ZM*ALF(4)+((ALF(1)**2.)*(1-ALF(2))/
     !(Z*ALF(2)))-ZM*ALF(4)/XM-ALF(4)
      P3=ALF(4)+ZM*ALF(4)-((ALF(1)**2.)*(1-ALF(2))/
     !(Z*ALF(2)))+ZM*ALF(4)/XM-1.
      PDF(3,3)=(S*ALF(2)/(2.*ALF(1)*XM))*(P2*EXP(XM)-P3*EXP(-XM))
      PDF(4,3)=(S*ALF(2)*YD/(2.*ALF(1)*XM))*(EXP(XM*YD)-EXP(-XM*YD))
C
C     CALCULATE WITH RESPECT TO ALF(4)
C
      PDF(1,4)=0.0
      PDF(2,4)=Z/(ALF(1)*(XM**3.))
      PDF(3,4)=((ZM/2.)-1.)*EXP(XM)+(1.-(ZM/2.))*EXP(-XM)
      PDF(4,4)=0.0
C
C     NOW CALCULATE THE DERIVATES
C
      X=F(1)*F(2)*F(3)*F(4)
      DO 10 I=1,4
```

```
      DO 9 J=1,4
      DF(I)=(X/F(J))*PDF(J,I)+DF(I).
9     CONTINUE
10    CONTINUE
C
C     CALCULATE THE PARTIAL DERIVATIVE WIHH RESPECT TO XD
C
      DF(5)=(P*FP)-(1./S)
C
C     CALCULATE THE NEEDED FUNCTIONS AS CALLED BY THE CALLING PROGRAM
C
15    IF(ISET.NE.1) GO TO 20
      PFUNC=FP
      GO TO 19
17    PFUNC=0.0
19    GO TO 200
20    IF(ISET.NE.2) GO TO 30
      PFUNC=DF(1)
      GO TO 29
25    PFUNC=0.0
29    GO TO 200
30    IF(ISET.NE.3)GO TO 40
      PFUNC=DF(2)
      GO TO 39
35    PFUNC=0.0
39    GO TO 200
40    IF(ISET.NE.4)GO TO 50
      PFUNC=DF(3)
      GO TO 49
45    PFUNC=0.0
49    GO TO 200
50    IF(ISET.NE.5)GO TO 60
      PFUNC=DF(4)
      GO TO 59
55    PFUNC=0.0
59    GO TO 200
60    IF(ISET.NE.6)GO TO 70
      PFUNC=DF(5)
      GO TO 69
65    PFUNC=0.0
69    GO TO 200
70    WRITE(6,75)ISET
75    FORMAT(//,'ERROR ISET OUT OF BOUNDS ISET =',3X,I4)
200   RETURN
      END
C
C
C     ******************************************************************
      ******************************************************************
C                         SUBROUTINE VARPRO
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                                I=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 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 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 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                    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  '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 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-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 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 NEW
```

```
C                NORM OF RESIDUAL.  NEW ALF IS STORED IN COLUMN B1 OF A.
C
   40     CALL DPA (L, NL, N, NMAX, LPP2, IV, T, Y, W, A(1, B1), ADA,
     X    IERR, IPRINT, A, BETA, A(1, LP1), RNEW)
          IF (IERR .NE. 2) GO TO 99
          ITER = ITER + 1
          ITERIN = ITERIN + 1
          SKIP = MOD(ITER, MODIT) .NE. 0
          IF (SKIP) GO TO 45
             WRITE (OUTPUT, 203) ITER
             WRITE (OUTPUT, 216) (A(K, B1), K = 1, NL)
             WRITE (OUTPUT, 207) ITERIN, RNEW
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 PARAMETER
   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             THXS 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, ALPHA, B(NMAX, NLP1), BETA, DSIGN, PRJRES,
     X U, XNORM
C
      NL = NLP1 - 1
      NL23 = 2*NL + 3
```

```
      LP1 = L + 1
C
      DO 30 K = 1, NL
         LPK = L + K
         ALPHA = DSIGN(XNORM(N+1-LPK, B(LPK, K)), B(LPK, K))
         U = B(LPK, K) + ALPHA
         B(LPK, K) = U
         BETA = ALPHA * U
         IF (ALPHA .NE. 0.0) GO TO 13
C                                                        COLUMN WAS ZERO
         IERR = -8
         CALL VARERR (IPRINT, IERR, LP1 + K)
         GO TO 99
C                                    APPLY REFLECTIONS TO REMAINING COLUMNS
C                                    OF B AND TO RESIDUAL VECTOR.
   13    KP1 = K + 1
         DO 25 J = KP1, 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

      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, NLP1
            JSUB = NLP1 + J
            B(K, J) = B(LPK, J)
   40       B(JSUB, K) = B(LPK, J)
   50    B(NL23, K) = XNORM(K, B(LP1, K))
C
   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, PHILP1
      EXTERNAL ADA
C
      IF (ISEL .NE. 1) GO TO 3
         LP1 = L + 1
         LNL2 = L + 2 + NL
         LP2 = L + 2
         LPP1 = LPP2 - 1
         FIRSTC = 1
         LASTC = LPP1
         FIRSTR = LP1
         CALL INIT(L, NL, N, NMAX, LPP2, IV, T, W, ALF, ADA, ISEL,
     X   IPRINT, A, INC, NCON, NCONP1, PHILP1, NOWATE)
         IF (ISEL .NE. 1) GO TO 99
         GO TO 30
C
    3 CALL ADA (LP1,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,MIN0(ISEL,3))
      IF (ISEL .EQ. 2) GO TO 6
C                                                   ISEL = 3 OR 4
      FIRSTC = LP2
      LASTC = LPP1
      FIRSTR = (4 - ISEL)*L + 1
      GO TO 50
C                                                   ISEL = 2
    6 FIRSTC = NCONP1
      LASTC = LP1
      IF (NCON .EQ. 0) GO TO 30
      IF (A(1, NCON) .EQ. SAVE) GO TO 30
         ISEL = -7
         CALL VARERR (IPRINT, ISEL, NCON)
         GO TO 99
C                                                    ISEL = 1 OR 2
   30 IF (PHILP1) GO TO 40
         DO 35 I = 1, N
   35       R(I) = Y(I)
         GO TO 50
   40    DO 45 I = 1, N
   45       R(I) = Y(I) - R(I)
C                                              WEIGHT APPROPRIATE COLUMNS
   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 HCUSEHOCQER
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              TPIANGLE 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
      NCON = L
      NCONP1 = LP1
      PHILP1 = L .EQ. 0
      IF (PHILP1 .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) 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 SCALEG 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, WAKING 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
```