TFMFIT: A PROGRAM TO FIT A GENERALIZED CONVECTION-DISPERSION MODEL TO EXPERIMENTAL DATA

D. A. Barry, Garrison Sposito and J. Coves

Department of Soil and Environmental Sciences
University of California, Riverside

July, 1988

TFMFIT: A PROGRAM TO FIT A GENERALIZED CONVECTION-DISPERSION MODEL TO EXPERIMENTAL DATA

D. A. Barry

Garrison Sposito

J. Coves

Department of Soil and Environmental Sciences
University of California
Riverside, CA 92521-0424

ACKNOWLEDGEMENTS

The development of TFMFIT was supported in part by National Science Foundation Grant ECE-8513726 and in part by grant W-703 from the Water Resources Center, University of California. Gratitude is expressed to the Academic Computing Center, University of California, Riverside, for time on the VAX 8700 and SDSC CRAY X-MP/48 computers, and to the San Diego Supercomputer Center (SDSC) for the use of its facilities. Dr. A. L. Wood, Robert S. Kerr Environmental Research Laboratory, Ada, Oklahoma, provided a copy of the experimental database on solute movement at the Borden site. Gratitude is expressed to Drs. W. A. Jury and J. S. Dyson for their reviews, and to Ms. L. DeHayes for her excellent typing of the present manuscript.

CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

INTRODUCTION

The task of obtaining estimates of model parameters (the "inverse problem") is encountered frequently in practice. The transfer function model is a general formulation describing solute transport from an "entrance surface" to an "exit surface" of a porous domain. The probability that solute will arrive at an exit surface is given by its travel time probability density function (pdf). A very general pdf based on the convection-dispersion equation is presented. The pdf incorporates many solute transport mechanisms, including sorption, volatilization and biodegradation. This document describes the pdf as well as the structure and usage of a versatile computer code. The code uses nonlinear least-squares fitting to find optimal parameter values by matching transfer function model predictions with measured experimental data. The program makes use of some standard computational algorithms in the widely-available IMSL package. Use of the program is demonstrated with synthetic data as well as data from a comprehensive field experiment. Breakthrough curves can be plotted using an additional code.

ANNOUNCEMENT

Persons wishing to obtain machine-readable copies of the codes listed here should contact D. A. Barry at the given departmental address or via electronic mail at BARRYD@UCRVMS.BITNET.

BIBLIOGRAPHY

Key: CDE - Convection-Dispersion Equation
CF - Curve Fitting
TFM - Transfer Function Model

Beck, J. V., and K. J. Arnold. 1977. Parameter estimation in engineering and science. John Wiley & Sons, New York. (CF)

Carnahan, C. L., and J. S. Remer. 1984. Nonequilibrium and equilibrium sorption with a linear sorption isotherm during mass transport through an infinite porous medium: some analytical solutions. *J. Hydrol.* 73:227-258. (CDE)

Jury, W. A., G. Sposito, and R. E. White. 1986. A transfer function model of solute transport through soil. 1. Fundamental concepts. *Water Resour. Res.* 22:243-247. (TFM)

Marquadt, D. W. 1963. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* 11:431-441. (CF)

Nauman, E. B., and B. A. Buffham. 1983. Mixing in continuous flow systems. John Wiley & Sons, Inc., New York. (CDE, TFM)

Parker, J. C., and M. Th. van Genuchten. 1984. Determining transport parameters from laboratory and field tracer experiments. Bulletin 84-3, Virginia Agricultural Experiment Station, Blacksburg. (CF, CDE)

Rinaldo, A., and A. Marani. 1987. Basin scale model of solute transport. *Water Resour. Res.* 23(11):2107-2118. (TFM)

Sposito, G., R. E. White, P. R. Darrah, and W. A. Jury. 1986. A transfer function model of solute transport through soil. 3. The convection-dispersion equation. *Water Resour. Res.* 22:255-262. (CDE, TFM)

White, R. E., J. S. Dyson, R. A. Haigh, W. A. Jury, and G. Sposito. 1986. A transfer function model of solute transport through soil. 2. Illustrative applications. *Water Resour. Res.* 22(2):248-254. (TFM)

THE TRANSFER FUNCTION MODEL

Fundamental Concepts

Central to the theory of subsurface solute transport is the prediction of concentrations and mass fluxes as functions of space and time, given prior information on the manner in which solute has entered the subsurface zone. Typically, the prior information required specifies fluxes and concentrations during the time period in which solute passed through an interface separating the zone of transport from the solute source. Transport into a vadose zone, for example, can involve the movement of deposited solute downward across the land surface or through the interface between a disposal site (landfill) and the underlying vadose zone. Transport into a groundwater zone can involve solute movement across a phreatic surface, or through an aquitard, or through the perforated casing of an injection well. More subtle modes of entry of solute into the subsurface zone are possible. Solutes can enter subsurface waters by traversing solid/liquid interfaces (desorption and dissolution processes) or by crossing air/liquid interfaces (absorption processes). Solutes can simply appear in subsurface water as a result of biological or chemical transformations of other solutes. In these cases, the "entrance surface" through which solute passes to begin a subsurface transport process takes on a highly convoluted, evanescent shape that defies smooth mathematical description at the local scale. This same difficulty may also attend the specification of an "exit surface", through which solute moves out of a monitored subsurface zone.

The very complicated nature of the initial conditions of solute movement in natural porous media apply equally to the subsurface transport process itself. Heterogeneity in the solid matrix and void spaces of the vadose and groundwater zones contributes to pronounced variability in the mechanisms by which solutes are moved and transformed. Identification of the broad classes of these mechanisms is reasonably complete, but quantitative details of their modes of operation at field and regional scales are mostly lacking.

These uncertainties and the goal of developing a predictive theory of subsurface solute transport that is not overly specialized lead one to a consideration of the problem based on probability and only fundamental physical notions. Under the broadest possible circumstances, let t' be the time at which a solute molecule first appears in the transporting portion of the void space of a vadose or groundwater zone, and let τ denote the time interval during which it exists in a monitored subsurface zone. The variable t' is the solute input time and τ will be termed the solute lifetime. Now define $g(\tau|t')\ d\tau$ as the conditional probability that the solute lifetime lies in the interval τ to $\tau + d\tau$, given that the input time was t'. The function $g(\tau|t')$ is the solute lifetime probability density function. Thus $g(\tau|t')\ d\tau$ is the probability that a monitored solute molecule which entered the transporting portion of a subsurface zone at time t' has disappeared from the subsurface zone at a later time $t' + \tau$. The geometric complexity associated with the modes of solute entrance or disappearance, noted above, and the mechanisms by which transport and transformation are realized in the subsurface zone are contained implicitly in $g(\tau|t')$.

Physical Significance of $g(\tau|t')$

The lifetime probability density function (pdf) contains all of the information necessary to construct a solute transport theory with predictive capability. If $Q_{in}(t')$ is the rate of solute mass input divided by the total mass input, and $Q_{out}(t)$ is the rate of solute mass output similarly normalized, then the identity

$$Q_{out}(t) = \int_0^t g(t - t'|t')\, Q_{in}(t')\, dt' \qquad (1)$$

relates the input rate over the time interval (0,t) to the output rate observed at time t. Equation 1 is a result derived from probability theory applied to $g(\tau|t')$ (Jury et al., 1986). Under conditions wherein it is meaningful to relate Q_{out} and Q_{in} directly to solute concentrations (e.g., steady water flow), eq. 1 can be reformulated in terms of those concentrations instead of mass transfer rates.

If solute input occurs after "time zero" over an interval that is very small compared to the time period over which output monitoring occurs, then Q_{out} is simply proportional to the conditional lifetime pdf $g(t|0)$. This lifetime pdf can be measured by observing the response of a subsurface zone to a narrow-pulse input of solute, and the measured $g(t|0)$ will represent the net effect of convection, dispersion, sorption, interphase transfer, biochemical transformation, etc., on the lifetime of a solute in the monitored region. Since eq. 1 is essentially a probabilistic statement of mass balance, no particular mechanistic cause of solute transport is implied by it. Any subsurface process or combination of processes consistent with mass balance is representable by a lifetime pdf, so long as the strong law of large numbers can be applied to interpret eq. 1 physically (cf. Barry and Sposito, 1988).

Since $g(\tau|t')$ is a pdf, it can be normalized and used to calculate (conditional) moments in the usual way (cf. Valocchi, 1986):

$$\int_0^\infty g(\tau|t')\, d\tau \equiv 1 \tag{2}$$

$$\int_0^\infty \tau^n\, g(\tau|t')\, d\tau \equiv E(\tau^n|t') \tag{3}$$

where E( ) is the mathematical expectation operator. Of interest also is the median lifetime, t_m, which is the solution of the integral equation:

$$\int_0^{t_m} g(\tau|t')\, d\tau = \frac{1}{2} \tag{4}$$

under the condition that eq. 2 holds. Note that t_m is conditional on the input time t'. For a conservative solute whose loss from a subsurface zone is monitored at a well-defined exit surface (e.g., a perforated well casing), the parameter t_m can serve to establish the time-of-arrival of the solute at this surface. If the volumetric water flux density is known and steady throughout the zone of transport, it can be multiplied by t_m and divided by a nominal distance of transport to estimate the void space fraction that is effective as the transporting portion.

If enough physical information about a subsurface transport process is available to permit a parametric model of the lifetime pdf (e.g., a lognormal function or an inverse Gaussian function), this model can be introduced into eqs. 1 to 4 to provide explicit mathematical representations of Q_{out} and the time-moments for use in model validation studies of field-scale solute transport.

Travel Time pdfs and the CDE

Analytical models of the travel time pdf for reactive solutes can be constructed from solutions of the convection-dispersion equation (CDE). The mathematical basis for this possibility is the Duhamel theorem (Carslaw and Jaeger, 1959) which states, in the present context, that solutions of linear CDEs always can be written in the form of eq. 1, with effluent and input flux concentrations replacing Q_{out} and Q_{in}, respectively, and with the travel time pdf explicitly independent of the input time. For solute transport under steady water flow conditions, the Duhamel theorem leads to the linear convolution expression:

$$c_{ex}(t) = \int_0^t g(t - t')\, c_{ent}(t')\, dt' \qquad (5)$$

where c_{ex} and c_{ent} are flux concentrations at the exit and entrance surfaces, respectively. The pdf g(t) is identified with the time-derivative of a flux concentration solution of a linear CDE subject to a constant flux at the entrance surface. It follows from eq. 5

that travel time pdfs can be calculated as time derivatives of solutions of the CDE subject to a constant flux concentration boundary condition (Sposito et al., 1986). These model pdfs are independent of the input time t' and contain physical parameters — like dispersion coefficients, convection velocities, and sorption distribution coefficients — that model the solute transport and transformation mechanisms assumed in the underlying CDE. The model pdfs thus are useful for comparative studies of transport phenomena in different vadose or groundwater zone situations and for studies of the influences of different transport mechanisms on the lineshapes of travel time spectra.

The general relationship between the travel time pdf in eq. 5 and a steady-flow, linear CDE can be epitomized in the following algorithm:

(1) A steady-water-flow CDE is established on the basis of hypotheses about the mechanisms of solute transport in a vadose or groundwater zone (convection, dispersion, sorption, biodegradation, etc.).

(2) The linear partial differential equation developed in step (1) is solved subject to a step-input boundary condition on the entrance surface.

(3) The travel time pdf is calculated as the time-derivative of the solution developed in step (2) on the exit surface. This pdf can be introduced into eq. 5 to predict exit-surface concentrations for arbitrary entrance-surface concentrations.

A GENERAL CONVECTION-DISPERSION EQUATION

A CDE can be developed to describe mathematically the following mechanistic picture of solute transport in groundwater:

(1) Solute moves in three spatial dimensions through an aquifer of uniform porosity in which the groundwater flow velocity v is steady and uniform, with the x_3-axis chosen to point along the direction of flow.

(2) Solute dispersion is anisotropic and is represented by the three constant dispersion coefficients, D_1, D_2, and D_3.

(3) The solute can be partitioned among the gas phase, the mobile and immobile portions of the liquid phase, and the solid phase. Biodegradation is possible in all phases.

The convection-dispersion equation that results from the mechanistic picture above can be expressed in the dimensionless form:

$$BR \frac{\partial c_m}{\partial T} + (1 - B) R \frac{\partial c_{im}}{\partial T} = \sum_{i=1}^{3} \frac{1}{P_i} \frac{\partial^2 c_m}{\partial X_i^2} - \frac{\partial c_m}{\partial X_3}$$

$$- BC c_m - (1 - B) D c_{im} \qquad (6)$$

$$(1 - B) R \frac{\partial c_{im}}{\partial T} + (1 - B) D c_{im} = W(c_m - c_{im}) \qquad (7)$$

where: c_m = solute concentration $[ML^{-3}]$ in the mobile portion of the liquid phase

c_{im} = solute concentration $[ML^{-3}]$ in the immobile portion of the liquid phase

$T = \theta_m vt/(\theta L)$ = dimensionless time coordinate

$X_i = x_i/L$ = dimensionless space coordinate (i=1,2,3)

Dimensionless Model Parameters

$$B = (\theta_m + f\ \rho_b\ K_d + \theta_{gm}\ K_H^{-1})/(\theta + \rho_b\ K_d + \theta_g\ K_H^{-1}) \qquad (8)$$

where: $\theta_{(i)m}$ = (im)mobile liquid content $[L^3L^{-3}]$

$\theta_{g(i)m}$ = air content in contact with (im)mobile liquid $[L^3L^{-3}]$

θ = liquid content $[L^3L^{-3}] = \theta_{im} + \theta_m$

θ_g = air content $[L^3L^{-3}] = \theta_{gim} + \theta_{gm}$

f = fraction of adsorbing solid phase in contact with mobile liquid

ρ_b = dry bulk density $[ML^{-3}]$

K_d = distribution coefficient for adsorption $[L^3M^{-1}]$

K_H = Henry's Law constant for solute partitioning between liquid and air ($c_{liquid} = K_H\ c_{gas}$)

$$R = 1 + (\rho_b K_d + \theta_g K_H^{-1})/\theta \qquad (9)$$

$$P_i = vL/D_i \qquad (i=1,2,3) \qquad (10)$$

where: v = velocity of mobile liquid phase $[LT^{-1}]$, taken along x_3

L = distance from solute input point to monitoring point [L], taken along x_3

D_i = solute dispersion coefficient $[L^2T^{-1}]$ in the X_i direction (i=1,2,3) $= D_i^{\,l} + (\theta_{gm}\ K_H^{-1}\ D_i^{\,g}/\theta_m)$

$$C = (\theta_m\ \beta_m^{\,l} + fK_d\ \beta_m^{\,s} + \theta_{gm}\ K_H^{-1}\ \beta_{im}^{\,g})\ L/[\theta_m\ v(1 - B)] \qquad (11)$$

where: $\beta_j{}^i$ = first-order solute decay constant $[T^{-1}]$ for i^{th} component (gas, liquid, or solid) in or contacting the j^{th} portion of the liquid phase (mobile or immobile)

$D_i{}^j$ = solute dispersion coefficient $[L^2T^{-1}]$ in the X_i direction in the j^{th} (gas, liquid) phase.

$$D = [\theta_{im}\,\beta_{im}{}^{l} + (1 - f)\,K_d\,\beta_{im}{}^{s} + \theta_{im} + \theta_{gim}\,K_H{}^{-1}\,\beta_{im}{}^{g}]\,L \Big/ [\theta_m\,v(1 - B)] \qquad (12)$$

$$W = (\alpha^{l} + \alpha^{g}\,K_H{}^{-1})\,L/(\theta_m v) \qquad (13)$$

where: α_i^j = linear solute transfer coefficient $[T^{-1}]$ for i^{th} component into the immobile portion of the liquid phase, j = l (liquid) or g (gas).

The initial and boundary conditions imposed on eqs. 6 and 7 are:

$$c_m(\vec{X},0) = c_{im}(\vec{X},0) = 0 \qquad (14a)$$

$$c_m(\vec{X}_{ent},T) = 1 \qquad \lim_{X_i \uparrow \infty} c_m(\vec{X},T) = 0 \qquad (14b)$$

where $\vec{X}_{ent}$ refers to any point on the entrance surface. Specific, uniform initial values of c_m and c_{im} in the aquifer and of c_m on the entrance surface have been assumed to be removed explicitly by subtraction and scaling operations.

An area-averaged, dimensionless travel time pdf corresponding to the boundary-value problem expressed in eqs. 6 and 14 is calculated according to the equation:

$$g(T) = \int_{X_3}^{Y_0} \frac{\partial c_m}{\partial T}\,Y\,dY \qquad (15)$$

where X_3 and Y_0 are the chosen spatial limits on the exit surface and Y is a dimensionless polar coordinate:

$$Y^2 = (P_1/P_3)\ X_1{}^2 + (P_2/P_3)\ X_2{}^2 + X_3{}^2 \qquad (16)$$

Equation 15 represents an effective travel time pdf averaged over the directions transverse to the direction of groundwater flow. The averaging process in eq. 15 has the effect of creating a one-dimensional travel time pdf.

SOLUTIONS FOR THE TRAVEL TIME PDF

Laplace Domain Solution

Equations 6, 7, and 14 can be solved by Laplace transformation. The resulting Laplace transform of g(T) defined by eq. 15 is then:

$$\overline{g}(s) = \frac{P_3 \exp(P_3 X_3/2)\{\exp[-X_3\ F(s)/2] - \exp[-Y_0\ F(s)/2]\}}{F(s)\{1 - \exp[P_3\ (X_3 - Y_0)/2)\}} \qquad (17)$$

where:

s is the Laplace transform parameter

$\overline{g}(s)$ is the Laplace transform of g(T), and

$$F(s) = \{P_3{}^2 + 4BRP_3 s\ [1 + \frac{(1-B)\ W/B}{(1-B)\ Rs + W + D(1-B)}]$$

$$+ 4BP_3 C + \frac{4(1-B)\ P_3\ WD}{(1-B)\ Rs + W + D(1-B)}\ \}^{1/2} \qquad (18)$$

The value of X_3 is either 0 (for volatilization at the surface) or 1 (for transport within the porous domain), whereas X_1 and X_2 can have any non-negative value. The pdf is area-averaged, so, from eq. 15, X_2 and X_3 define the region averaged over. If x_3 is the distance along the vertical axis, $X_3 = 0$ is the soil surface and $X_3 = 1$ is the monitoring depth L.

The model pdf describes the travel times of a solute which can move by convection and dispersion in the mobile liquid and air phases, can be volatilized, can adsorb, and can decay by first-order kinetics. Adsorption, decay, and volatilization are permitted in both the mobile and immobile phases. The volatilization pdf is found by setting $X_3 = 0$.

Inversion of the Laplace-Transformed pdf

Two choices are available to compute numerical values of g(t). The Laplace transform of g(t), i.e., $\overline{g}(s)$, can be inverted numerically using standard inversion routines (e.g. Talbot, 1979) or g(t) itself can be derived and computed directly. TFMFIT uses both approaches and, therefore, the analytical inversion of $\overline{g}(s)$, as given by eq. 17, is now presented.

Equation 17 is rewritten as:

$$\overline{g}(s) = A[\overline{g}_1(s) - \overline{g}_2(s)] \qquad (19)$$

where

$$A = P_3 \exp(P_3X_3/2)/(1 - a)$$

$$\overline{g}_1(s) = \exp[-X_3F(s)/2]/[\sqrt{4BP_3R}\ F_1(s)]$$

$$\overline{g}_2(s) = \exp[-Y_0F(s)/2]/[\sqrt{4BP_3R}\ F_1(s)]$$

with

$$a = \exp[P_3(X_3 - Y_0)/2]$$

and

$$F_1(s) = [s + \frac{C}{R} + \frac{(s + \frac{D}{R})(\frac{P_3}{4BR} + \frac{W}{BR}) + \frac{P_3W}{4R^2B(1-B)}}{s + \frac{D}{R} + \frac{W}{R(1 - B)}}]^{1/2}$$

Observe that

$$\overline{g}_1(s) = \int_{-\infty}^{-X_3\sqrt{BP_3R}} \frac{\exp[F_1(s)X]}{2\sqrt{BP_3R}}\, dX \qquad (20)$$

the function $\overline{g}_2(s)$ can be similarly defined by replacing the upper limit of integration in eq. 20 by $-Y_0\sqrt{BP_3R}$. Thus, the problem of inverting eq. 17 has been reduced to finding the inverse of $\exp[F_1(s)X]$.

A convenient method of achieving this goal is the iterated Laplace Transform inversion procedure (Sneddon, 1972) as illustrated by De Smedt and Wierenga (1979). Briefly, the basic steps involved in this procedure are: (i) to denote the Laplace transform parameter s as two (or more) new parameters s_1 and s_2, with s_1 and s_2 chosen such that inversion with respect to these variables is straightforward, (ii) to invert with respect to s_1 and s_2 so that the final result is given as a convolution over the expression obtained. Carrying out (i) first, one writes $F_1(s)$ as $F_1(s_1,s_2)$ where

$$F_1(s_1,s_2) = [s_1 + \frac{C}{R} + \frac{(s_2 + \frac{D}{R})(\frac{P_3}{4BR} + \frac{W}{BR}) + \frac{P_3W}{4R^2B(1-B)}}{s_2 + \frac{D}{R} + \frac{W}{R(1-B)}}]^{1/2} \qquad (21)$$

The operation of inverting eq. 21 with respect to s_1, treating s_2 as a constant (the Laplace Transform operator is denoted by $\pounds$) gives

$$\pounds_1^{-1}\{\exp[XF_1(s_1,s_2)]\} =$$

$$-X\exp\{\frac{-X^2}{4T_1} - \frac{CT_1}{R} - \frac{[(s_2 + \frac{D}{R})(\frac{P_3}{4BR} + \frac{W}{BR}) + \frac{P_3W}{4R^2B(1-B)}]\,T_1}{s_2 + \frac{D}{R} + \frac{W}{R(1-B)}}\}/\sqrt{4\pi T_1{}^3} \qquad (22)$$

Next, the right-hand side of eq. 22 is inverted with respect to s_2 using the result of Appendix 2,

$$\mathcal{L}_2^{-1}\ \{\exp[XF_1(s_1,s_2)]\} = -X\ \exp[\ \frac{-X^2}{4T_1} - \frac{CT_1}{R} - \frac{DT_2}{R} - T_1\ (\ \frac{P_3}{4BR} + \frac{W}{BR}\ )]$$

$$\{\delta(T_2) + \frac{W}{R}\sqrt{\frac{T_1}{T_2B(1-B)}}\ \exp[\ \frac{-WT_2}{R(1-B)}\ ]\ I_1[\ \frac{2W}{R}\quad \frac{T_1T_2}{B(1-B)}\ ]\}/\sqrt{4\pi T_1^{\ 3}} \qquad (23)$$

where $I_n(\ )$ is the modified Bessel function of order n and $\delta(\ )$ represents the Dirac delta function. Finally, after performing the integrations defined by eq. 20 and carrying out some algebra, step (ii) is performed and g(T) is given as

$$g(T) = \sqrt{\frac{P_3}{4\pi RB}}\ \frac{h_1(-T)}{(1-a)}\ \{\ [h_2(X_3,T) - ah_2(Y_0,T)]/\sqrt{T}\ +$$

$$\frac{W}{R\sqrt{B(1-B)}}\ \int_0^T I_1[\ \frac{2W}{R}\ \sqrt{\frac{\tau(T-\tau)}{B(1-B)}}\ ]\ \frac{h_1(\tau)}{\sqrt{\tau}}\ \exp[-\tau(D + \frac{W}{1-B})/R]$$

$$[h_2(X_3,T-\tau) - ah_2(Y_0,T-\tau)]\ d\tau\} \qquad (24)$$

where

$$h_1(T) = \exp[T(C + W/B)/R]$$

$$h_2(X,T) = \exp[(-BRP_3(X - \frac{T}{BR}\ )^2/(4T)]$$

The Case B = 1

This case arises typically when the porous medium is completely saturated, there is no immobile water within the system, and the solute under consideration acts as an ideal tracer (C=0), or is subject to first-order production or decay (C≠0). Letting B=1 in eq. 24 presents obvious computational difficulties and so an explicit g(t) for B=1 is necessary. Finding g(t) under this restriction is straightforward. The result is:

$$g(T) = \sqrt{\frac{P_3}{4\pi RT}}\ \frac{\exp[-(T/R)(C + P_3/4) + P_3X_3/2]}{1-a}$$

$$\{\exp[-X_3{}^2P_3R/(4T)] - \exp[-Y_0{}^2P_3R/(4T)]\} \qquad (25)$$

DESCRIPTION OF THE CODE

A FORTRAN program to accomplish the nonlinear least-squares fitting of the model to experimental data has been implemented on a CRAY X-MP/48 using single precision CRAY FORTRAN (CFT) and operating under the CRAY Time Sharing System (CTSS). An analogous version operates on a VAX 8000 series system using double precision VAX FORTRAN. CRAY single precision storage is as precise as VAX double precision storage, so both codes yield virtually identical results. The code, which is named TFMFIT, is based on the maximum neighborhood method of Marquardt (1963) as implemented by van Genuchten (1980, 1981) and Parker and van Genuchten (1984) (cf. Meeter, 1964; Daniel and Wood, 1973). The CRAY and VAX versions of the source code are given in Appendix 1a and 1b, respectively. The algorithm attempts to find the combination of parameter values that will minimize the sum of the residuals squared between the experimental data and model predictions.

There are many existing parameter estimation procedures (Beck and Arnold, 1977), of which the Marquardt (1963) algorithm is but one. Hendrickson et al. (1988) have shown that, in the particular case of rainfall-runoff models, there is no single "best" algorithm that is optimally efficient, or even convergent to prior known parameter values. This conclusion likely applies to the estimation of transport parameters as well. It is suggested, therefore, that all parameter estimates provided by TFMFIT be subjected to critical scrutiny in the light of the model used and the experiment being modeled.

The code consists of the main program (named TFMFIT), the subroutines MATINV, MODEL, ZRES, FLINV, DCADRE, and REALTIME, and the functions ROOT, F, F1, MMBSI1, and FGRAND. The main program handles the input and output of data, checks that parameters are within allowable ranges, and implements the fitting algorithm. MATINV performs matrix inversion as required by the fitting algorithm. MODEL organizes data for evaluating the TFM. ZRES converts parameters from real to complex values, then calls FLINV, the IMSL (International Mathematical Subroutine Library, 1982) subroutine for the numerical inversion of Laplace transforms, which inverts eq. 17 to give solute concentrations in "real" time. FLINV requires an estimate of the maximum of the real part of the singularities of the function to be inverted. This estimate is calculated by ZRES using the function ROOT. The actual evaluation of eq. 17 occurs in the functions F and F1. REALTIME computes the analytical solution for the TFM as given by eqs. 24 or 25. The integration in eq. 24 is calculated numerically with the IMSL subroutine DCADRE. FGRAND calculates the integrand in eq. 24 for use in DCADRE, while MMBSI1 (IMSL, 1982) evaluates the Bessel function as required by FGRAND. The fact that the code calls IMSL routines does not limit its usefulness, since the IMSL package is very widely available.

As was noted above, model predictions can be obtained either through numerical inversion of eq. 17, or via numerical integration of eq. 24. Both methods require roughly the same amount of CPU time except for the case B=1 (eq. 25), when the analytical solution is much more efficient. The numerical inversion is more sensitive to numerical instabilities, particularly for large values of P_3 (e.g., $P_3 > 90$). However, the properties of the Laplace transform facilitate the incorporation of arbitrary boundary conditions into the solution. This version of TFMFIT includes the Laplace transform inversion primarily to allow users a straightforward way of incorporating complex time variations in the boundary conditions into their analyses with very few modifications of the code.

The code has been optimized to take advantage of the vector processing available on the CRAY computer. Compared with a scalar machine (e.g., the VAX computer), the relative speed of the vectorized code increases with the number of experimental data points. However, it should be pointed out that the main computational chore is the many numerical transform inversions of eq. 17, or numerical evaluation of eq. 24, required to achieve an optimal fit of the model with the experimental data. The IMSL library has not been vectorized to date, although a CRAY-vectorized IMSL is to appear sometime in the future. In the following, a number of fitting examples are given. It was found that TFMFIT ran roughly 10 times faster on the San Diego Supercomputer Center CRAY X-MP/48, as compared with the VAX 8700 at the University of California, Riverside. This increase in speed will almost certainly jump by at least another factor of 10 when the vectorized IMSL object code becomes available.

Table 1. Parameter file required by TFMFIT. This particular file is used in Example 1c.

```
    40        30         1         1
Example 1c.  Synthetic data with some random noise.
Run on the San Diego Supercomputer Center's CRAY X-MP/48.
P3....    X3....    Yo....    B.....    R.....    W.....    D.....
    5.        1.       1.5        .8       1.1        .5        .4
     1         0         1         1         1         0         0
    .1        .0    1.0001        .1        .1        .1        .1
 1000.        1.     1000.        1.     1000.     1000.     1000.
C.....
    .3
     0
    .1
 1000.
```

Operation

On execution, TFMFIT accepts an input file, a file containing the experimental data, and prompts for a "results" file name. A plot file can also be created if desired. Table 1 contains an example input parameter file. The line-by-line organization of this file is documented in Table 2. Table 3 contains an example experimental data file.

Table 2. Detailed information regarding the input parameter file for TFMFIT (example given in Table 1).

Line	Columns	Format	Description
1	1-10	I10	Maximum number of interactions allowed in the least-squares fitting. If set to 0, the program simply calculates concentrations at specified times using the initial parameter values.
	11-20	I10	Number of trial solutions allowed at each iteration.
	21-30	I10	Plot code 1(0) - do (not) create plot file.
	31-40	I10	0 Use numerical Laplace transform always. 1 Use numerical Laplace transform until numerical problems occur, then use analytical solution. 2 Use analytical solution always.
2-3	1-80	A80	Descriptive labels.
4,9			Parameter names (P_3, X_3, etc.)
5,10	1-10,11-20,etc.	F10.5	Initial guesses for parameters
6,11	"	I10	Fitting code: (0)1 - (don't) fit this parameter.
7,12	"	F10.5	Lower limit allowed for parameter.
8,13	"	"	Upper limit allowed for parameter.

Each line in this file contains, in order, observed concentration and normalized time, formatted as 2F10.5. Table 4c contains the output file for the example contained in Table 1. The information contained in this file is self-explanatory for the most part. In order, the file gives:

Table 3. File of observed experimental concentration data and corresponding normalized time (used in Example 1a and b).

0.006	.025
0.289	.05
0.787	.075
1.134	.1
1.300	.125
1.346	.15
1.326	.175
1.271	.2
1.202	.225
1.128	.25
1.055	.275
0.985	.3
0.919	.325
0.858	.35
0.802	.375
0.751	.4
0.703	.425
0.660	.45
0.620	.475
0.584	.5
0.550	.525
0.519	.55
0.491	.575
0.464	.6
0.417	.65
0.375	.7
0.339	.75
0.308	.8
0.280	.85
0.256	.9
0.234	.95
0.214	1.
0.197	1.05
0.181	1.1
0.167	1.15
0.154	1.2
0.142	1.25
0.132	1.3
0.122	1.35
0.113	1.4
0.105	1.45
0.098	1.5
0.091	1.55
0.085	1.6
0.080	1.65
0.075	1.7
0.070	1.75
0.065	1.8
0.061	1.85
0.057	1.9
0.054	1.95
0.051	2.
0.048	2.05
0.045	2.1
0.042	2.15
0.040	2.2
0.038	2.25
0.035	2.3
0.033	2.35
0.032	2.4
0.030	2.45
0.028	2.5

(i) descriptive title,

(ii) names and initial values of the TFM parameters,

(iii) iteration number and sum-of-squares for the given parameter values,

(iv) correlation matrix of the varied parameters,

(v) the R^2 value for the regression,

(vi) optimized parameter values and their 95% confidence limit bounds,

(vii) fitted and observed data as ordered by the input file,

(viii) fitted and observed data as ordered by the residual magnitude.

Finally, Table 7 contains the plot file for Example 1a. The file contains the experimental data, followed by the fitting results. Each line in the file contains

(i) concentration (observed or fitted)

(ii) measurement time

(iii) number of observation (corresponds to experimental-data file).

Data in the plot file can be accessed using the format: 2(1PE15.4),I3.

EXAMPLE APPLICATIONS

Three examples of the use of TFMFIT are given in the following. The first example is based on a synthetic data set, while the second and third fit actual field breakthrough data.

The first example is designed to demonstrate (i) the importance of choosing an accurate set of initial estimates of the model parameters; and (ii) the effect of random noise in the data on the parameter estimates given by TFMFIT.

Table 4a. TFMFIT results for Example 1a.

```
Nonlinear Least-Squares Analysis.

   Example 1a.  Synthetic data.
   Run on the University of California, Riverside VAX 8700.

Initial values of coefficients
==============================
  Name      Initial value
  P3........  1.0000D+01
  X3........  1.0000D+00
  Yo........  2.0000D+00
  B.........  7.0000D-01
  R.........  1.5000D+00
  W.........  5.0000D-01
  D.........  4.0000D-01
  C.........  3.0000D-01

Iteration      SSQ          P3....       Yo....       B.....       R.....
    0       1.8529D+01   1.0000D+01   2.0000D+00   7.0000D-01   1.5000D+00
    1       6.8219D+00   1.0000D-01   4.3867D+00   6.4063D-01   1.4218D+00
    2       4.7969D+00   1.0737D-01   1.9842D+00   1.0000D+00   2.2021D+00
    3       4.3949D+00   1.9970D-01   2.3464D+00   1.0000D+00   1.0635D+00
    4       3.5267D+00   3.8783D-01   2.5605D+00   1.0000D+00   5.8744D-01
    5       5.2240D-01   6.1306D-01   2.5398D+00   1.0000D+00   5.7978D-01
    6       1.2371D-01   8.0911D-01   1.7804D+00   1.0000D+00   6.6648D-01
    7       2.1128D-02   8.4716D-01   1.6386D+00   1.0000D+00   7.3615D-01
    8       1.3683D-02   8.3317D-01   1.5404D+00   1.0000D+00   7.8885D-01
    9       9.6114D-03   8.1665D-01   1.4746D+00   1.0000D+00   8.3337D-01
   10       7.0667D-03   8.0340D-01   1.4239D+00   1.0000D+00   8.7105D-01
   11       5.3947D-03   7.9339D-01   1.3830D+00   1.0000D+00   9.0297D-01
   12       4.2540D-03   7.8592D-01   1.3492D+00   1.0000D+00   9.3020D-01
   13       3.4542D-03   7.8031D-01   1.3208D+00   1.0000D+00   9.5357D-01
   14       3.1180D-03   7.5984D-01   1.2022D+00   1.0000D+00   1.0501D+00
   15       1.8622D-03   7.6014D-01   1.1009D+00   1.0000D+00   1.1398D+00
   16       9.7106D-04   7.6661D-01   1.0936D+00   1.0000D+00   1.1468D+00
   17       9.7076D-04   7.6625D-01   1.0969D+00   1.0000D+00   1.1441D+00
   18       9.7073D-04   7.6634D-01   1.0959D+00   1.0000D+00   1.1450D+00
   19       9.7073D-04   7.6632D-01   1.0962D+00   1.0000D+00   1.1447D+00
   20       9.7073D-04   7.6632D-01   1.0961D+00   1.0000D+00   1.1448D+00

  Correlation Matrix
  ==================
        1          2          3          4
  1   1.0000
  2  -0.4948     1.0000
  3   0.0000     0.0000     0.0000
  4   0.2605    -0.9667     0.0000     1.0000

Rsquare for regression =     0.9999064965D+00.


Nonlinear Least-Squares Analysis, Final Results.
================================================
                                                            95% Confidence Limits
Variable    Name        Value        S.E.Coeff.  T-Value     Lower         Upper
    1       P3....   7.6632D-01    5.2570D-03 1.458D+02  7.5580D-01    7.7685D-01
    2       Yo....   1.0961D+00    2.4983D-02 4.388D+01  1.0461D+00    1.1461D+00
    3       B.....   1.000CD+00    4.0910D-33 2.444D+32  1.0000D+00    1.0000D+00
    4       R.....   1.1448D+00    2.1199D-02 5.400D+01  1.1024D+00    1.1873D+00
```

```
                        Ordered by computer input
No.      T            Conc Obs       Conc Fit       Residual
  1   2.500D-02      6.000D-03      7.522D-03     -1.522D-03
  2   5.000D-02      2.890D-01      2.992D-01     -1.024D-02
  3   7.500D-02      7.870D-01      7.915D-01     -4.483D-03
  4   1.000D-01      1.134D+00      1.130D+00      3.856D-03
  5   1.250D-01      1.300D+00      1.293D+00      7.196D-03
  6   1.500D-01      1.346D+00      1.340D+00      5.639D-03
  7   1.750D-01      1.326D+00      1.323D+00      2.974D-03
  8   2.000D-01      1.271D+00      1.272D+00     -1.156D-03
  9   2.250D-01      1.202D+00      1.206D+00     -3.635D-03
 10   2.500D-01      1.128D+00      1.133D+00     -5.419D-03
 11   2.750D-01      1.055D+00      1.061D+00     -5.858D-03
 12   3.000D-01      9.850D-01      9.908D-01     -5.850D-03
 13   3.250D-01      9.190D-01      9.248D-01     -5.756D-03
 14   3.500D-01      8.580D-01      8.631D-01     -5.136D-03
 15   3.750D-01      8.020D-01      8.061D-01     -4.125D-03
 16   4.000D-01      7.510D-01      7.536D-01     -2.591D-03
 17   4.250D-01      7.030D-01      7.053D-01     -2.283D-03
 18   4.500D-01      6.600D-01      6.610D-01     -9.660D-04
 19   4.750D-01      6.200D-01      6.202D-01     -2.116D-04
 20   5.000D-01      5.840D-01      5.828D-01      1.242D-03
 21   5.250D-01      5.500D-01      5.483D-01      1.705D-03
 22   5.500D-01      5.190D-01      5.166D-01      2.448D-03
 23   5.750D-01      4.910D-01      4.873D-01      3.725D-03
 24   6.000D-01      4.640D-01      4.602D-01      3.774D-03
 25   6.500D-01      4.170D-01      4.120D-01      4.968D-03
 26   7.000D-01      3.750D-01      3.705D-01      4.455D-03
 27   7.500D-01      3.390D-01      3.346D-01      4.383D-03
 28   8.000D-01      3.080D-01      3.033D-01      4.674D-03
 29   8.500D-01      2.800D-01      2.759D-01      4.072D-03
 30   9.000D-01      2.560D-01      2.518D-01      4.184D-03
 31   9.500D-01      2.340D-01      2.305D-01      3.501D-03
 32   1.000D+00      2.140D-01      2.116D-01      2.437D-03
 33   1.050D+00      1.970D-01      1.947D-01      2.320D-03
 34   1.100D+00      1.810D-01      1.796D-01      1.432D-03
 35   1.150D+00      1.670D-01      1.660D-01      1.008D-03
 36   1.200D+00      1.540D-01      1.538D-01      2.432D-04
 37   1.250D+00      1.420D-01      1.427D-01     -6.952D-04
 38   1.300D+00      1.320D-01      1.327D-01     -6.654D-04
 39   1.350D+00      1.220D-01      1.235D-01     -1.546D-03
 40   1.400D+00      1.130D-01      1.152D-01     -2.238D-03
 41   1.450D+00      1.050D-01      1.076D-01     -2.647D-03
 42   1.500D+00      9.800D-02      1.007D-01     -2.693D-03
 43   1.550D+00      9.100D-02      9.431D-02     -3.312D-03
 44   1.600D+00      8.500D-02      8.844D-02     -3.445D-03
 45   1.650D+00      8.000D-02      8.304D-02     -3.040D-03
 46   1.700D+00      7.500D-02      7.805D-02     -3.052D-03
 47   1.750D+00      7.000D-02      7.344D-02     -3.440D-03
 48   1.800D+00      6.500D-02      6.917D-02     -4.171D-03
 49   1.850D+00      6.100D-02      6.521D-02     -4.211D-03
 50   1.900D+00      5.700D-02      6.153D-02     -4.534D-03
 51   1.950D+00      5.400D-02      5.811D-02     -4.115D-03
 52   2.000D+00      5.100D-02      5.493D-02     -3.930D-03
 53   2.050D+00      4.800D-02      5.196D-02     -3.961D-03
 54   2.100D+00      4.500D-02      4.919D-02     -4.190D-03
 55   2.150D+00      4.200D-02      4.660D-02     -4.600D-03
 56   2.200D+00      4.000D-02      4.418D-02     -4.176D-03
 57   2.250D+00      3.800D-02      4.191D-02     -3.907D-03
```

```
58     2.300D+00      3.500D-02      3.978D-02     -4.779D-03
59     2.350D+00      3.300D-02      3.778D-02     -4.783D-03
60     2.400D+00      3.200D-02      3.591D-02     -3.908D-03
61     2.450D+00      3.000D-02      3.415D-02     -4.145D-03
62     2.500D+00      2.800D-02      3.249D-02     -4.487D-03

                              Ordered by residual
No.        T             Conc Obs       Conc Fit        Residual
 2     5.000D-02      2.890D-01      2.992D-01     -1.024D-02
 5     1.250D-01      1.300D+00      1.293D+00      7.196D-03
11     2.750D-01      1.055D+00      1.061D+00     -5.858D-03
12     3.000D-01      9.850D-01      9.908D-01     -5.850D-03
13     3.250D-01      9.190D-01      9.248D-01     -5.756D-03
 6     1.500D-01      1.346D+00      1.340D+00      5.639D-03
10     2.500D-01      1.128D+00      1.133D+00     -5.419D-03
14     3.500D-01      8.580D-01      8.631D-01     -5.136D-03
25     6.500D-01      4.170D-01      4.120D-01      4.968D-03
59     2.350D+00      3.300D-02      3.778D-02     -4.783D-03
58     2.300D+00      3.500D-02      3.978D-02     -4.779D-03
28     8.000D-01      3.080D-01      3.033D-01      4.674D-03
55     2.150D+00      4.200D-02      4.660D-02     -4.600D-03
50     1.900D+00      5.700D-02      6.153D-02     -4.534D-03
62     2.500D+00      2.800D-02      3.249D-02     -4.487D-03
 3     7.500D-02      7.870D-01      7.915D-01     -4.483D-03
26     7.000D-01      3.750D-01      3.705D-01      4.455D-03
27     7.500D-01      3.390D-01      3.346D-01      4.383D-03
49     1.850D+00      6.100D-02      6.521D-02     -4.211D-03
54     2.100D+00      4.500D-02      4.919D-02     -4.190D-03
30     9.000D-01      2.560D-01      2.518D-01      4.184D-03
56     2.200D+00      4.000D-02      4.418D-02     -4.176D-03
48     1.800D+00      6.500D-02      6.917D-02     -4.171D-03
61     2.450D+00      3.000D-02      3.415D-02     -4.145D-03
15     3.750D-01      8.020D-01      8.061D-01     -4.125D-03
51     1.950D+00      5.400D-02      5.811D-02     -4.115D-03
29     8.500D-01      2.800D-01      2.759D-01      4.072D-03
53     2.050D+00      4.800D-02      5.196D-02     -3.961D-03
52     2.000D+00      5.100D-02      5.493D-02     -3.930D-03
60     2.400D+00      3.200D-02      3.591D-02     -3.908D-03
57     2.250D+00      3.800D-02      4.191D-02     -3.907D-03
 4     1.000D-01      1.134D+00      1.130D+00      3.856D-03
24     6.000D-01      4.640D-01      4.602D-01      3.774D-03
23     5.750D-01      4.910D-01      4.873D-01      3.725D-03
 9     2.250D-01      1.202D+00      1.206D+00     -3.635D-03
31     9.500D-01      2.340D-01      2.305D-01      3.501D-03
44     1.600D+00      8.500D-02      8.844D-02     -3.445D-03
47     1.750D+00      7.000D-02      7.344D-02     -3.440D-03
43     1.550D+00      9.100D-02      9.431D-02     -3.312D-03
46     1.700D+00      7.500D-02      7.805D-02     -3.052D-03
45     1.650D+00      8.000D-02      8.304D-02     -3.040D-03
 7     1.750D-01      1.326D+00      1.323D+00      2.974D-03
42     1.500D+00      9.800D-02      1.007D-01     -2.693D-03
41     1.450D+00      1.050D-01      1.076D-01     -2.647D-03
16     4.000D-01      7.510D-01      7.536D-01     -2.591D-03
22     5.500D-01      5.190D-01      5.166D-01      2.448D-03
32     1.000D+00      2.140D-01      2.116D-01      2.437D-03
33     1.050D+00      1.970D-01      1.947D-01      2.320D-03
17     4.250D-01      7.030D-01      7.053D-01     -2.283D-03
40     1.400D+00      1.130D-01      1.152D-01     -2.238D-03
21     5.250D-01      5.500D-01      5.483D-01      1.705D-03
39     1.350D+00      1.220D-01      1.235D-01     -1.546D-03
```

```
 1     2.500D-02     6.000D-03     7.522D-03    -1.522D-03
34     1.100D+00     1.810D-01     1.796D-01     1.432D-03
20     5.000D-01     5.840D-01     5.828D-01     1.242D-03
 8     2.000D-01     1.271D+00     1.272D+00    -1.156D-03
35     1.150D+00     1.670D-01     1.660D-01     1.008D-03
18     4.500D-01     6.600D-01     6.610D-01    -9.660D-04
37     1.250D+00     1.420D-01     1.427D-01    -6.952D-04
38     1.300D+00     1.320D-01     1.327D-01    -6.654D-04
36     1.200D+00     1.540D-01     1.538D-01     2.432D-04
19     4.750D-01     6.200D-01     6.202D-01    -2.116D-04
```

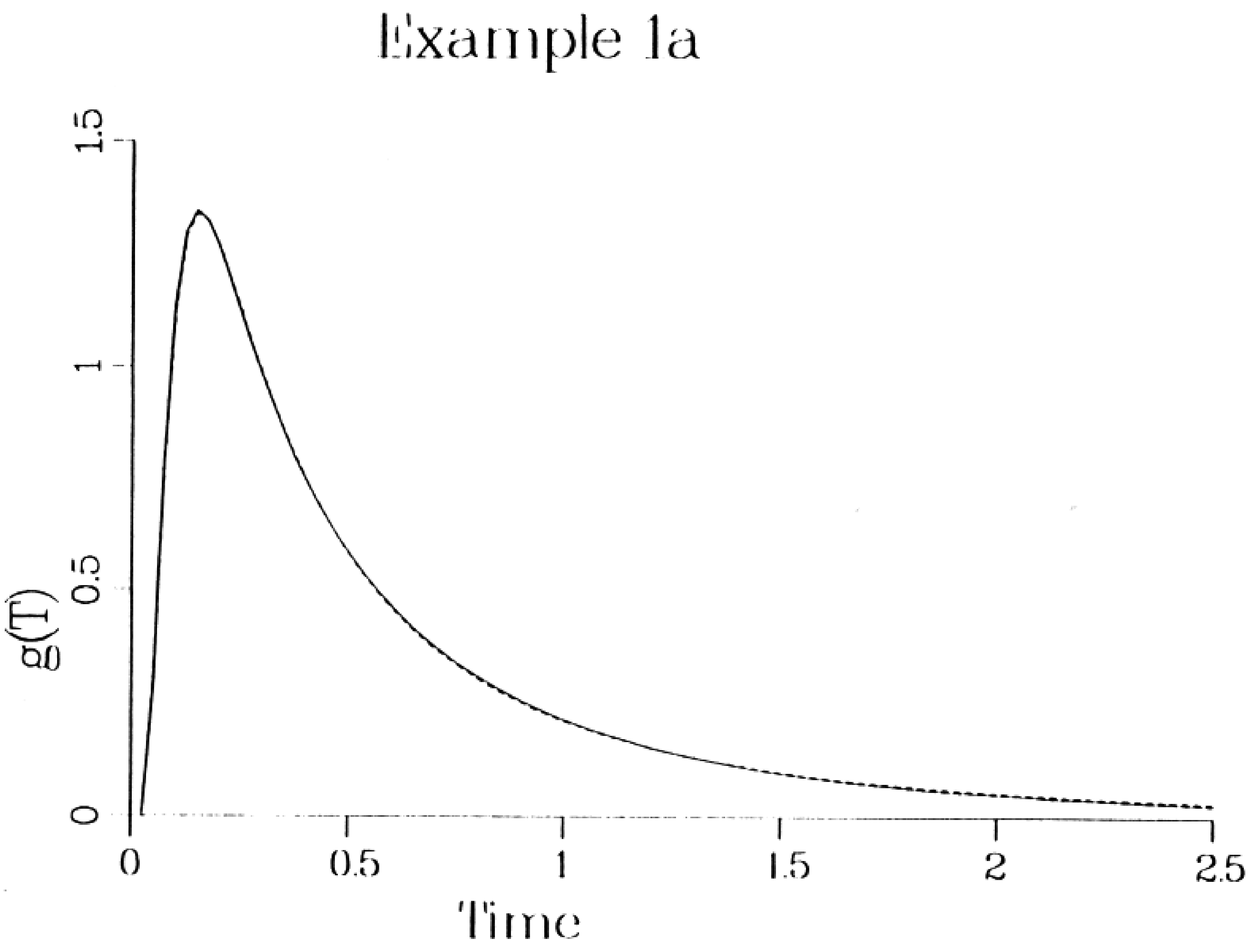


Fig. 1a. Model fit for Example 1a.

A data set was created using TFMFIT (with MIT = 0 in the input data file) with the following parameters:

$$P_3 = 1 \qquad R = 1$$
$$X_3 = 1 \qquad W = 0.5$$
$$Y_0 = 1.1 \qquad D = 0.4$$
$$B = 0.9 \qquad C = 0.3$$

The results produced by the model were stored for the range $0.025 \leq T \leq 2.5$, with an increment of 0.025 or 0.05, giving a total of 62 (T, C) data pairs. These data were rounded to three decimal places (Table 3) and then used as the experimental data set. In Example 1a (cf. Table 1), the initial parameter estimates were assumed to be

$$P_3 = 10 \qquad B = 0.7$$
$$Y_0 = 2 \qquad R = 1.5$$

with the other parameters assumed known and set to their exact values. The results of this exercise are given in Table 4a, where the fitted values are found to be

$$P_3 = 0.766 \qquad B = 1$$
$$Y_0 = 1.096 \qquad R = 1.145$$

The model prediction using these parameters is compared with the input data in Fig. 1a. Although the fit is excellent, the fitting algorithm has converged to incorrect parameter values.

In the next case, Example 1b, the initial parameter guesses were altered to

$P_3 = 5$ $\quad$ $B = 0.8$

$Y_0 = 1.5$ $\quad$ $R = 1.1$

with results in Table 4b. With more accurate initial guesses, the results indicate a rapid convergence to the known parameter values. It is clear that the most sound procedure is to estimate as precisely as possible the allowable parameter ranges prior to, and independently of, fitting the data.

The effect of random noise in the data is illustrated in Example 1c. A pseudo-random error term was added to each of the synthetic concentration data used previously. The error terms were drawn from a Gaussian distribution with a mean and standard deviation of 0 and 0.05, respectively (IMSL routine G05DDF). Noisy data less than zero were reset to zero. Utilizing the same input parameters as Example 1b, the fitting algorithm gives the output shown in Table 4c. Figure 1b displays the best-fit curve compared with the noisy data. The disparity between the best-fit curve and the source data is apparent in the 95% confidence limits of the parameters given in Table 4c. Although it is true that the range of each parameter encompasses the known value, it is also clear that the random noise in the initial data set decreases the reliability of parameter estimates. The true values lie close to the center of the given ranges because of the simple structure of the random errors. We observe that random noise with a non-zero drift component, or noise that is multiplicative, will produce correspondingly different behavior in the parameter estimates and ranges. The major point is that the nature of random noise has a direct effect on the best-fit parameters obtained.

Table 4b. TFMFIT results for Example 1b.

```
Nonlinear Least-Squares Analysis.


    Example 1b.  Synthetic data.
    Run on the San Diego Supercomputer Center's CRAY X-MP/48.

 Initial values of coefficients
 ==============================
   Name     Initial value
   P3........  5.0000e+00
   X3........  1.0000e+00
   Yo........  1.5000e+00
   B.........  8.0000e-01
   R.........  1.1000e+00
   W.........  5.0000e-01
   D.........  4.0000e-01
   C.........  3.0000e-01

 Iteration       SSQ           P3....       Yo....       B.....        R.....
     0       1.3592e+01    5.0000e+00   1.5000e+00   8.0000e-01   1.1000e+00
     1       6.8960e+00    2.7011e+00   1.4678e+00   7.3696e-01   1.0383e+00
     2       1.7297e+00    1.9764e+00   1.3634e+00   6.7476e-01   9.5592e-01
     3       3.8833e-02    1.4246e+00   1.3579e+00   6.4810e-01   9.0108e-01
     4       2.1538e-02    1.2844e+00   1.3209e+00   7.1827e-01   8.8033e-01
     5       9.4346e-03    1.2124e+00   1.2884e+00   7.8115e-01   8.6344e-01
     6       8.9642e-03    1.0609e+00   1.2018e+00   8.7836e-01   8.8634e-01
     7       1.9869e-03    9.9409e-01   1.1206e+00   8.9902e-01   9.7878e-01
     8       9.1449e-06    9.9886e-01   1.1012e+00   9.0015e-01   9.9935e-01
     9       6.0226e-06    1.0000e+00   1.1005e+00   8.9993e-01   9.9968e-01
    10       6.0218e-06    9.9998e-01   1.1005e+00   8.9994e-01   9.9972e-01

   Correlation Matrix
   ==================
         1          2          3          4
   1   1.0000
   2   0.0158     1.0000
   3  -0.8892    -0.2385     1.0000
   4  -0.2852    -0.9524     0.4149     1.0000

 Rsquare for regression =     0.9999994200e+00.


Nonlinear Least Squares Analysis, Final Results.
================================================
                                                          95% Confidence Limits
 Variable    Name       Value        S.E.Coeff.  T-Value    Lower        Upper
     1       P3....   9.9998e-01    1.0000e-03 9.999e+02  9.9798e-01   1.0020e+00
     2       Yo....   1.1005e+00    1.6761e-03 6.565e+02  1.0971e+00   1.1038e+00
     3       B.....   8.9994e-01    4.2503e-04 2.117e+03  8.9909e-01   9.0079e-01
     4       R.....   9.9972e-01    1.3399e-03 7.461e+02  9.9704e-01   1.0024e+00

                          Ordered by computer input
  No.       T            Conc Obs       Conc Fit       Residual
   1     2.500e-02     6.000e-03     6.478e-03     -4.785e-04
   2     5.000e-02     2.890e-01     2.888e-01      2.148e-04
   3     7.500e-02     7.870e-01     7.871e-01     -1.036e-04
   4     1.000e-01     1.134e+00     1.134e+00     -2.866e-04
   5     1.250e-01     1.300e+00     1.300e+00      2.217e-04
   6     1.500e-01     1.346e+00     1.346e+00      1.338e-04
  7     1.750e-01     1.326e+00     1.325e+00     5.411e-04
  8     2.000e-01     1.271e+00     1.271e+00    -3.806e-04
  9     2.250e-01     1.202e+00     1.202e+00    -2.930e-04
 10     2.500e-01     1.128e+00     1.128e+00    -3.698e-04
 11     2.750e-01     1.055e+00     1.055e+00     6.589e-05
 12     3.000e-01     9.850e-01     9.847e-01     2.636e-04
 13     3.250e-01     9.190e-01     9.190e-01     2.151e-05
 14     3.500e-01     8.580e-01     8.581e-01    -5.452e-05
 15     3.750e-01     8.020e-01     8.020e-01     2.354e-05
 16     4.000e-01     7.510e-01     7.505e-01     4.987e-04
 17     4.250e-01     7.030e-01     7.033e-01    -2.978e-04
 18     4.500e-01     6.600e-01     6.601e-01    -6.015e-05
 19     4.750e-01     6.200e-01     6.203e-01    -3.344e-04
 20     5.000e-01     5.840e-01     5.838e-01     1.766e-04
 21     5.250e-01     5.500e-01     5.502e-01    -2.023e-04
 22     5.500e-01     5.190e-01     5.192e-01    -1.892e-04
 23     5.750e-01     4.910e-01     4.905e-01     4.721e-04
 24     6.000e-01     4.640e-01     4.640e-01     1.024e-05
 25     6.500e-01     4.170e-01     4.165e-01     5.131e-04
 26     7.000e-01     3.750e-01     3.753e-01    -3.276e-04
 27     7.500e-01     3.390e-01     3.394e-01    -4.287e-04
 28     8.000e-01     3.080e-01     3.079e-01     6.158e-05
 29     8.500e-01     2.800e-01     2.802e-01    -1.649e-04
 30     9.000e-01     2.560e-01     2.556e-01     4.374e-04
 31     9.500e-01     2.340e-01     2.337e-01     3.232e-04
 32     1.000e+00     2.140e-01     2.141e-01    -1.377e-04
 33     1.050e+00     1.970e-01     1.966e-01     3.726e-04
 34     1.100e+00     1.810e-01     1.809e-01     1.061e-04
 35     1.150e+00     1.670e-01     1.667e-01     2.850e-04
 36     1.200e+00     1.540e-01     1.539e-01     9.635e-05
 37     1.250e+00     1.420e-01     1.423e-01    -3.000e-04
 38     1.300e+00     1.320e-01     1.318e-01     2.338e-04
 39     1.350e+00     1.220e-01     1.222e-01    -1.824e-04
 40     1.400e+00     1.130e-01     1.135e-01    -4.511e-04
 41     1.450e+00     1.050e-01     1.055e-01    -4.765e-04
 42     1.500e+00     9.800e-02     9.818e-02    -1.783e-04
 43     1.550e+00     9.100e-02     9.149e-02    -4.903e-04
 44     1.600e+00     8.500e-02     8.535e-02    -3.515e-04
 45     1.650e+00     8.000e-02     7.971e-02     2.922e-04
 46     1.700e+00     7.500e-02     7.451e-02     4.884e-04
 47     1.750e+00     7.000e-02     6.972e-02     2.793e-04
 48     1.800e+00     6.500e-02     6.530e-02    -2.974e-04
 49     1.850e+00     6.100e-02     6.121e-02    -2.082e-04
 50     1.900e+00     5.700e-02     5.742e-02    -4.232e-04
 51     1.950e+00     5.400e-02     5.392e-02     8.447e-05
 52     2.000e+00     5.100e-02     5.066e-02     3.389e-04
 53     2.050e+00     4.800e-02     4.764e-02     3.616e-04
 54     2.100e+00     4.500e-02     4.483e-02     1.722e-04
 55     2.150e+00     4.200e-02     4.221e-02    -2.118e-04
 56     2.200e+00     4.000e-02     3.977e-02     2.255e-04
 57     2.250e+00     3.800e-02     3.750e-02     4.980e-04
 58     2.300e+00     3.500e-02     3.538e-02    -3.805e-04
 59     2.350e+00     3.300e-02     3.340e-02    -3.986e-04
 60     2.400e+00     3.200e-02     3.155e-02     4.546e-04
 61     2.450e+00     3.000e-02     2.981e-02     1.887e-04
 62     2.500e+00     2.800e-02     2.819e-02    -1.873e-04

                          Ordered by residual
No.         T           Conc Obs      Conc Fit      Residual
  7     1.750e-01     1.326e+00     1.325e+00     5.411e-04
```

```
25   6.500e-01   4.170e-01   4.165e-01    5.131e-04
16   4.000e-01   7.510e-01   7.505e-01    4.987e-04
57   2.250e+00   3.800e-02   3.750e-02    4.980e-04
43   1.550e+00   9.100e-02   9.149e-02   -4.903e-04
46   1.700e+00   7.500e-02   7.451e-02    4.884e-04
 1   2.500e-02   6.000e-03   6.478e-03   -4.785e-04
41   1.450e+00   1.050e-01   1.055e-01   -4.765e-04
23   5.750e-01   4.910e-01   4.905e-01    4.721e-04
60   2.400e+00   3.200e-02   3.155e-02    4.546e-04
40   1.400e+00   1.130e-01   1.135e-01   -4.511e-04
30   9.000e-01   2.560e-01   2.556e-01    4.374e-04
27   7.500e-01   3.390e-01   3.394e-01   -4.287e-04
50   1.900e+00   5.700e-02   5.742e-02   -4.232e-04
59   2.350e+00   3.300e-02   3.340e-02   -3.986e-04
 8   2.000e-01   1.271e+00   1.271e+00   -3.806e-04
58   2.300e+00   3.500e-02   3.538e-02   -3.805e-04
33   1.050e+00   1.970e-01   1.966e-01    3.726e-04
10   2.500e-01   1.128e+00   1.128e+00   -3.698e-04
53   2.050e+00   4.800e-02   4.764e-02    3.616e-04
44   1.600e+00   8.500e-02   8.535e-02   -3.515e-04
52   2.000e+00   5.100e-02   5.066e-02    3.389e-04
19   4.750e-01   6.200e-01   6.203e-01   -3.344e-04
26   7.000e-01   3.750e-01   3.753e-01   -3.276e-04
31   9.500e-01   2.340e-01   2.337e-01    3.232e-04
37   1.250e+00   1.420e-01   1.423e-01   -3.000e-04
17   4.250e-01   7.030e-01   7.033e-01   -2.978e-04
48   1.800e+00   6.500e-02   6.530e-02   -2.974e-04
 9   2.250e-01   1.202e+00   1.202e+00   -2.930e-04
45   1.650e+00   8.000e-02   7.971e-02    2.922e-04
 4   1.000e-01   1.134e+00   1.134e+00   -2.866e-04
35   1.150e+00   1.670e-01   1.667e-01    2.850e-04
47   1.750e+00   7.000e-02   6.972e-02    2.793e-04
12   3.000e-01   9.850e-01   9.847e-01    2.636e-04
38   1.300e+00   1.320e-01   1.318e-01    2.338e-04
56   2.200e+00   4.000e-02   3.977e-02    2.255e-04
 5   1.250e-01   1.300e+00   1.300e+00    2.217e-04
 2   5.000e-02   2.890e-01   2.888e-01    2.148e-04
55   2.150e+00   4.200e-02   4.221e-02   -2.118e-04
49   1.850e+00   6.100e-02   6.121e-02   -2.082e-04
21   5.250e-01   5.500e-01   5.502e-01   -2.023e-04
22   5.500e-01   5.190e-01   5.192e-01   -1.892e-04
61   2.450e+00   3.000e-02   2.981e-02    1.887e-04
62   2.500e+00   2.800e-02   2.819e-02   -1.873e-04
39   1.350e+00   1.220e-01   1.222e-01   -1.824e-04
42   1.500e+00   9.800e-02   9.818e-02   -1.783e-04
20   5.000e-01   5.840e-01   5.838e-01    1.766e-04
54   2.100e+00   4.500e-02   4.483e-02    1.722e-04
29   8.500e-01   2.800e-01   2.802e-01   -1.649e-04
32   1.000e+00   2.140e-01   2.141e-01   -1.377e-04
 6   1.500e-01   1.346e+00   1.346e+00    1.338e-04
34   1.100e+00   1.810e-01   1.809e-01    1.061e-04
 3   7.500e-02   7.870e-01   7.871e-01   -1.036e-04
36   1.200e+00   1.540e-01   1.539e-01    9.635e-05
51   1.950e+00   5.400e-02   5.392e-02    8.447e-05
11   2.750e-01   1.055e+00   1.055e+00    6.589e-05
28   8.000e-01   3.080e-01   3.079e-01    6.158e-05
18   4.500e-01   6.600e-01   6.601e-C1   -6.015e-05
14   3.500e-01   8.580e-01   8.581e-01   -5.452e-05
15   3.750e-01   8.020e-01   8.020e-01    2.354e-05
13   3.250e-01   9.190e-01   9.190e-01    2.151e-05
24   6.000e-01   4.640e-01   4.640e-01    1.024e-05
```

Table 4c. TFMFIT results for Example 1c.

```
Nonlinear Least-Squares Analysis.


    Example 1c.  Synthetic data with some random noise.
    Run on the San Diego Supercomputer Center's CRAY X-MP/48.

 Initial values of coefficients
 ==============================
   Name     Initial value
   P3........  5.0000e+00
   X3........  1.0000e+00
   Yo........  1.5000e+00
   B.........  8.0000e-01
   R.........  1.1000e+00
   W.........  5.0000e-01
   D.........  4.0000e-01
   C.........  3.0000e-01

 Iteration       SSQ          P3....       Yo....       B.....       R.....
    0       1.3505e+01    5.0000e+00   1.5000e+00   8.0000e-01   1.1000e+00
    1       7.1649e+00    2.7478e+00   1.4752e+00   7.3760e-01   1.0408e+00
    2       1.9765e+00    1.9864e+00   1.3790e+00   6.7564e-01   9.6002e-01
    3       1.2335e-01    1.3667e+00   1.3907e+00   6.4608e-01   9.0428e-01
    4       1.1124e-01    1.2739e+00   1.3505e+00   7.0409e-01   8.9271e-01
    5       1.0429e-01    1.2210e+00   1.3178e+00   7.5252e-01   8.8019e-01
    6       1.0252e-01    1.1100e+00   1.2432e+00   8.3231e-01   8.8774e-01
    7       9.9338e-02    1.0942e+00   1.2200e+00   8.3417e-01   9.2096e-01
    8       9.9315e-02    1.0902e+00   1.2229e+00   8.3617e-01   9.2018e-01
    9       9.9314e-02    1.0904e+00   1.2216e+00   8.3594e-01   9.2122e-01
   10       9.9314e-02    1.0904e+00   1.2216e+00   8.3594e-01   9.2122e-01

   Correlation Matrix
   ==================
          1          2           3          4
   1   1.0000
   2   0.4535      1.0000
   3  -0.8364     -0.3648     1.0000
   4  -0.5434     -0.9652     0.3184     1.0000

 Rsquare for regression =     0.9904731890e+00.


Nonlinear Least Squares Analysis, Final Results.
================================================
                                                          95% Confidence Limits
 Variable    Name       Value         S.E.Coeff.  T-Value    Lower         Upper
     1       P3....   1.0904e+00     1.1048e-01 9.870e+00  8.6927e-01   1.3116e+00
     2       Yo....   1.2216e+00     1.8887e-01 6.468e+00  8.4349e-01   1.5997e+00
     3       B.....   8.3594e-01     5.3324e-02 1.568e+01  7.2920e-01   9.4269e-01
     4       R.....   9.2122e-01     1.4144e-01 6.513e+00  6.3809e-01   1.2044e+00

                              Ordered by computer input
  No.        T            Conc Obs        Conc Fit        Residual
   1     2.500e-02       1.228e-02       6.909e-03       5.371e-03
   2     5.000e-02       2.189e-01       2.795e-01      -6.060e-02
   3     7.500e-02       7.625e-01       7.675e-01      -4.987e-03
   4     1.000e-01       1.154e+00       1.118e+00       3.529e-02
   5     1.250e-01       1.255e+00       1.291e+00      -3.664e-02
   6     1.500e-01       1.378e+00       1.343e+00       3.537e-02
  7     1.750e-01     1.364e+00     1.325e+00     3.841e-02
  8     2.000e-01     1.293e+00     1.272e+00     2.156e-02
  9     2.250e-01     1.159e+00     1.202e+00    -4.264e-02
 10     2.500e-01     1.065e+00     1.126e+00    -6.053e-02
 11     2.750e-01     1.062e+00     1.050e+00     1.193e-02
 12     3.000e-01     9.336e-01     9.782e-01    -4.465e-02
 13     3.250e-01     8.852e-01     9.106e-01    -2.537e-02
 14     3.500e-01     8.772e-01     8.481e-01     2.915e-02
 15     3.750e-01     7.857e-01     7.907e-01    -4.991e-03
 16     4.000e-01     7.176e-01     7.383e-01    -2.066e-02
 17     4.250e-01     7.046e-01     6.904e-01     1.422e-02
 18     4.500e-01     6.173e-01     6.467e-01    -2.940e-02
 19     4.750e-01     5.717e-01     6.068e-01    -3.508e-02
 20     5.000e-01     5.909e-01     5.702e-01     2.063e-02
 21     5.250e-01     5.042e-01     5.368e-01    -3.252e-02
 22     5.500e-01     5.086e-01     5.060e-01     2.590e-03
 23     5.750e-01     5.194e-01     4.777e-01     4.165e-02
 24     6.000e-01     4.316e-01     4.517e-01    -2.002e-02
 25     6.500e-01     4.315e-01     4.052e-01     2.627e-02
 26     7.000e-01     3.683e-01     3.653e-01     3.011e-03
 27     7.500e-01     3.614e-01     3.306e-01     3.081e-02
 28     8.000e-01     3.595e-01     3.002e-01     5.925e-02
 29     8.500e-01     2.861e-01     2.735e-01     1.261e-02
 30     9.000e-01     3.346e-01     2.498e-01     8.477e-02
 31     9.500e-01     2.149e-01     2.288e-01    -1.381e-02
 32     1.000e+00     2.123e-01     2.099e-01     2.359e-03
 33     1.050e+00     1.828e-01     1.930e-01    -1.023e-02
 34     1.100e+00     1.656e-01     1.778e-01    -1.220e-02
 35     1.150e+00     1.548e-01     1.640e-01    -9.241e-03
 36     1.200e+00     2.331e-02     1.515e-01    -1.282e-01
 37     1.250e+00     1.172e-01     1.402e-01    -2.303e-02
 38     1.300e+00     8.462e-02     1.299e-01    -4.524e-02
 39     1.350e+00     1.126e-01     1.204e-01    -7.883e-03
 40     1.400e+00     5.178e-02     1.118e-01    -6.003e-02
 41     1.450e+00     4.501e-02     1.039e-01    -5.891e-02
 42     1.500e+00     7.510e-03     9.667e-02    -8.916e-02
 43     1.550e+00     9.140e-02     9.001e-02     1.390e-03
 44     1.600e+00     1.050e-01     8.388e-02     2.113e-02
 45     1.650e+00     1.344e-01     7.824e-02     5.613e-02
 46     1.700e+00     1.297e-01     7.304e-02     5.664e-02
 47     1.750e+00     1.064e-01     6.823e-02     3.814e-02
 48     1.800e+00     3.694e-02     6.379e-02    -2.685e-02
 49     1.850e+00     0.            5.968e-02    -5.968e-02
 50     1.900e+00     5.724e-02     5.587e-02     1.372e-03
 51     1.950e+00     1.027e-02     5.234e-02    -4.207e-02
 52     2.000e+00     1.064e-01     4.906e-02     5.737e-02
 53     2.050e+00     5.560e-02     4.602e-02     9.579e-03
 54     2.100e+00     2.734e-02     4.319e-02    -1.585e-02
 55     2.150e+00     1.128e-01     4.056e-02     7.227e-02
 56     2.200e+00     4.982e-02     3.812e-02     1.170e-02
 57     2.250e+00     0.            3.583e-02    -3.583e-02
 58     2.300e+00     4.562e-02     3.371e-02     1.191e-02
 59     2.350e+00     0.            3.172e-02    -3.172e-02
 60     2.400e+00     4.180e-02     2.987e-02     1.193e-02
 61     2.450e+00     0.            2.814e-02    -2.814e-02
 62     2.500e+00     4.895e-02     2.652e-02     2.243e-02

                                Ordered by residual
No.        T          Conc Obs      Conc Fit      Residual
 36     1.200e+00     2.331e-02     1.515e-01    -1.282e-01
```

```
42   1.500e+00   7.510e-03   9.667e-02   -8.916e-02
30   9.000e-01   3.346e-01   2.498e-01    8.477e-02
55   2.150e+00   1.128e-01   4.056e-02    7.227e-02
 2   5.000e-02   2.189e-01   2.795e-01   -6.060e-02
10   2.500e-01   1.065e+00   1.126e+00   -6.053e-02
40   1.400e+00   5.178e-02   1.118e-01   -6.003e-02
49   1.850e+00   0.          5.968e-02   -5.968e-02
28   8.000e-01   3.595e-01   3.002e-01    5.925e-02
41   1.450e+00   4.501e-02   1.039e-01   -5.891e-02
52   2.000e+00   1.064e-01   4.906e-02    5.737e-02
46   1.700e+00   1.297e-01   7.304e-02    5.664e-02
45   1.650e+00   1.344e-01   7.824e-02    5.613e-02
38   1.300e+00   8.462e-02   1.299e-01   -4.524e-02
12   3.000e-01   9.336e-01   9.782e-01   -4.465e-02
 9   2.250e-01   1.159e+00   1.202e+00   -4.264e-02
51   1.950e+00   1.027e-02   5.234e-02   -4.207e-02
23   5.750e-01   5.194e-01   4.777e-01    4.165e-02
 7   1.750e-01   1.364e+00   1.325e+00    3.841e-02
47   1.750e+00   1.064e-01   6.823e-02    3.814e-02
 5   1.250e-01   1.255e+00   1.291e+00   -3.664e-02
57   2.250e+00   0.          3.583e-02   -3.583e-02
 6   1.500e-01   1.378e+00   1.343e+00    3.537e-02
 4   1.000e-01   1.154e+00   1.118e+00    3.529e-02
19   4.750e-01   5.717e-01   6.068e-01   -3.508e-02
21   5.250e-01   5.042e-01   5.368e-01   -3.252e-02
59   2.350e+00   0.          3.172e-02   -3.172e-02
27   7.500e-01   3.614e-01   3.306e-01    3.081e-02
18   4.500e-01   6.173e-01   6.467e-01   -2.940e-02
14   3.500e-01   8.772e-01   8.481e-01    2.915e-02
61   2.450e+00   0.          2.814e-02   -2.814e-02
48   1.800e+00   3.694e-02   6.379e-02   -2.685e-02
25   6.500e-01   4.315e-01   4.052e-01    2.627e-02
13   3.250e-01   8.852e-01   9.106e-01   -2.537e-02
37   1.250e+00   1.172e-01   1.402e-01   -2.303e-02
62   2.500e+00   4.895e-02   2.652e-02    2.243e-02
 8   2.000e-01   1.293e+00   1.272e+00    2.156e-02
44   1.600e+00   1.050e-01   8.388e-02    2.113e-02
16   4.000e-01   7.176e-01   7.383e-01   -2.066e-02
20   5.000e-01   5.909e-01   5.702e-01    2.063e-02
24   6.000e-01   4.316e-01   4.517e-01   -2.002e-02
54   2.100e+00   2.734e-02   4.319e-02   -1.585e-02
17   4.250e-01   7.046e-01   6.904e-01    1.422e-02
31   9.500e-01   2.149e-01   2.288e-01   -1.381e-02
29   8.500e-01   2.861e-01   2.735e-01    1.261e-02
34   1.100e+00   1.656e-01   1.778e-01   -1.220e-02
60   2.400e+00   4.180e-02   2.987e-02    1.193e-02
11   2.750e-01   1.062e+00   1.050e+00    1.193e-02
58   2.300e+00   4.562e-02   3.371e-02    1.191e-02
56   2.200e+00   4.982e-02   3.812e-02    1.170e-02
33   1.050e+00   1.828e-01   1.930e-01   -1.023e-02
53   2.050e+00   5.560e-02   4.602e-02    9.579e-03
35   1.150e+00   1.548e-01   1.640e-01   -9.241e-03
39   1.350e+00   1.126e-01   1.204e-01   -7.883e-03
 1   2.500e-02   1.228e-02   6.909e-03    5.371e-03
15   3.750e-01   7.857e-01   7.907e-01   -4.991e-03
 3   7.500e-02   7.625e-01   7.675e-01   -4.987e-03
26   7.000e-01   3.683e-01   3.653e-01    3.011e-03
22   5.500e-01   5.086e-01   5.060e-01    2.590e-03
32   1.000e+00   2.123e-01   2.099e-01    2.359e-03
43   1.550e+00   9.140e-02   9.001e-02    1.390e-03
50   1.900e+00   5.724e-02   5.587e-02    1.372e-03
```

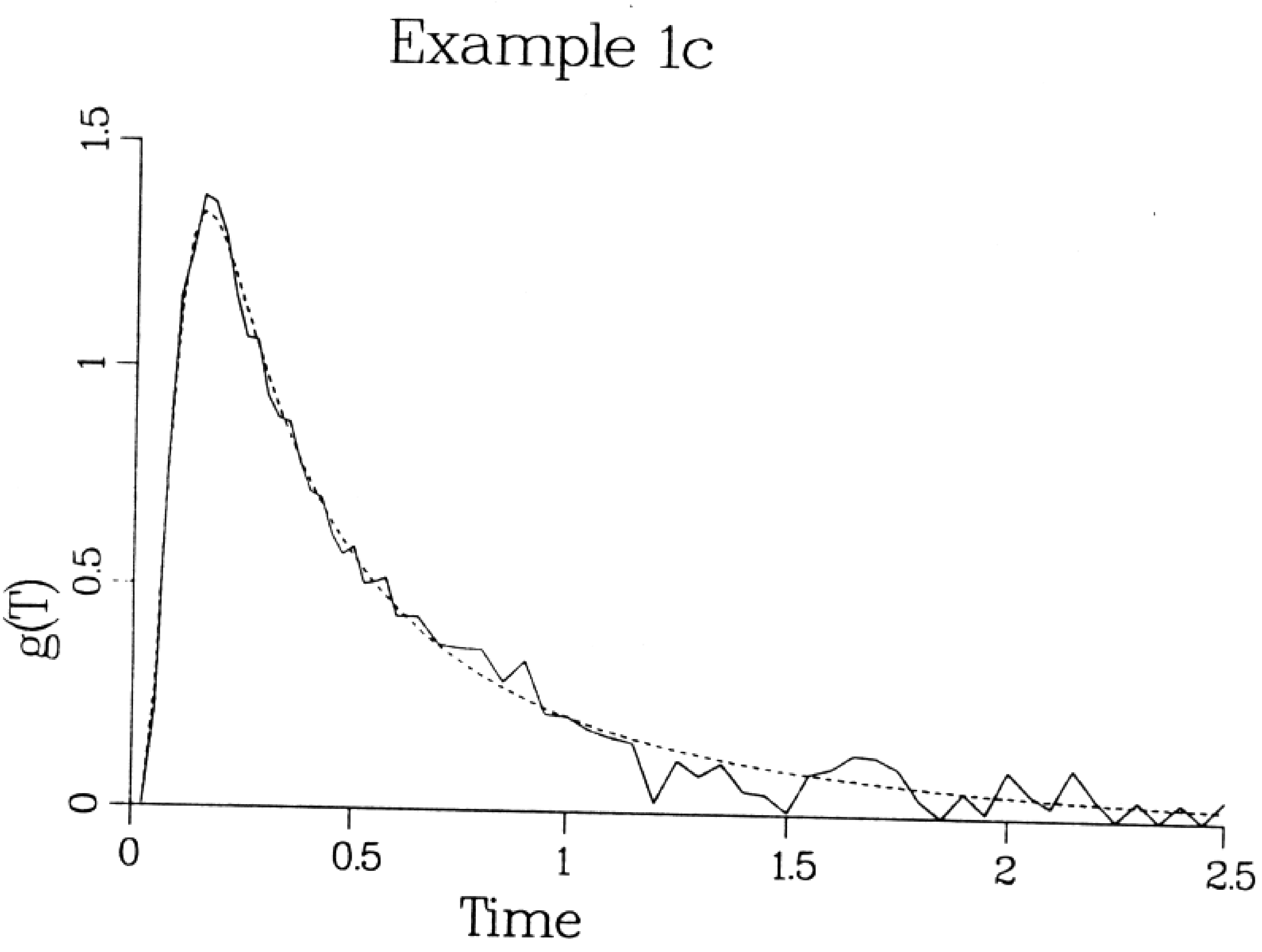


Fig. 1b. TFMFIT results for the noisy synthetic data.

The second and third examples are based on data obtained from a large-scale solute transport experiment at the Borden site, Ontario, Canada (Roberts and Mackay, 1986). The details of the experiment are readily available (e.g., Freyberg, 1986; Goltz and Roberts, 1986; Mackay et al., 1986; Sudicky, 1986), and will not be repeated here except in the context of using TFMFIT. In the following the measured breakthrough curves of two solutes--carbon tetrachloride and tetrachloroethylene--are considered.

As shown by eq. 15, the pdf g(T) is averaged over Y from $Y = X_3$ to $Y = Y_0$, where $X_3 \leq Y_0$. From eq. 16 it can be seen that $Y = X_3$ implies $X_1 = X_2 = 0$. Thus, the pdf is averaged from a specified position on the X_3 axis (X_3 = 0 or 1) to Y_0. In principle, the limits of eq. 15 can be specified according to the location of the data, given that data are often collected away from the X_3 axis. This would require, however, a priori knowledge of P_i, i=1,2,3, in eq. 16. Because this information is unavailable, it is recommended that data be fitted using X_3 = 0 or 1. TFMFIT has been restricted to these possibilities.

In Example 2, we consider first the measured breakthrough of carbon tetrachloride at a sampling location in the Borden aquifer (see Roberts et al., 1986, for details). The physical characteristics of this chemical, as well as tetrachloroethylene (considered in Example 3), with respect to the Borden site have been documented by Curtis et al. (1986),

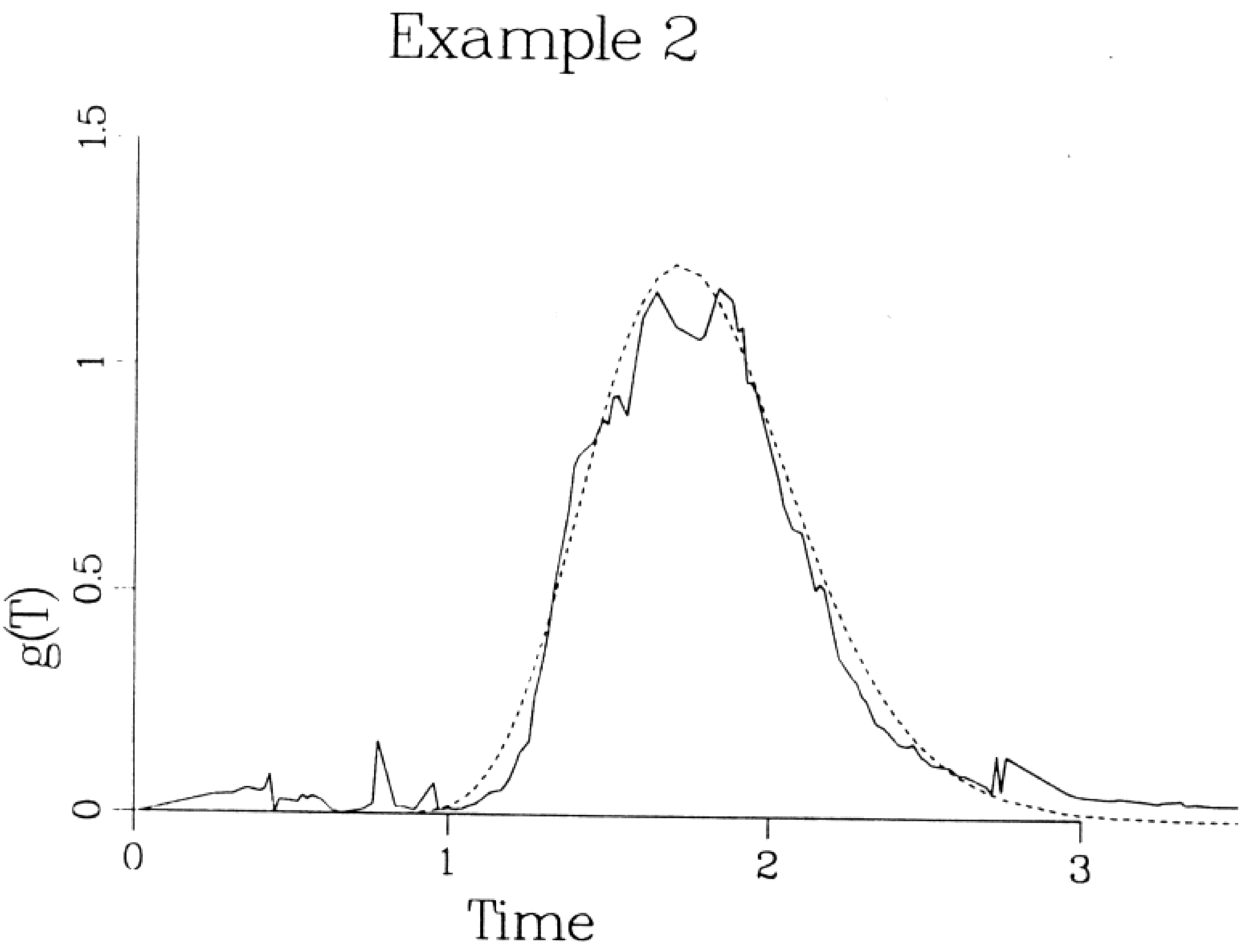


Fig. 2. TFMFIT results for the carbon tetrachloride data from the Borden transport experiment.

Goltz and Roberts (1986), Mackay et al. (1986) and Roberts et al. (1986). As a first approximation, we assume that carbon tetrachloride moves in a fully saturated medium in which the mobile liquid content is the entire water content (B = 1) and that it does not undergo first-order decay (C = 0). The required solution is then given by eq. 25 with C = 0.

The model is given in dimensionless form. The data, therefore, must be given in dimensionless form also, so that the physical meaning of the parameters may be determined. For the time variable, the following operation is performed: $T = vt/L$. This definition of dimensionless time is a specialization of that given below eq. 7 for the case $\theta_m = \theta$. This condition seems to be a reasonable assumption for the Borden aquifer. For porous media not satisfying this condition, θ_m must be estimated *a priori*, but it can be seen from eqs. 6 and 7 that T and R are inversely proportional, so that any error in defining T will adjust R proportionally (cf. Barry, 1988). Secondly, the model assumes a Dirac pulse of solute as the initial condition. Although the exact form of the initial condition becomes less important with increasing time as the solute pulse loses its "memory" of its initial shape, normalization of the concentration data is still required. The concentration data were normalized to a dimensionless variable according to

$$C_{norm} = \frac{C_{meas}L}{vA_r} \qquad (26)$$

where C_{norm} = normalized concentration, C_{meas} = measured concentration, and A_r is given by

$$A_r = \int_0^\infty C_{meas}(L,t)dt \qquad (27)$$

The results of the fitting for this case are given in Fig. 2 and Table 5. Figure 2 indicates that a good fit of the data has been obtained. This is confirmed by the data presented in Table 5. An interesting result in this table is contained in the correlation matrix, where it is shown that the correlation of Y_0 with itself is 0. This result implies that varying Y_0 has no effect on the quality of the model fit to the data. Of course, this result is theoretically invalid. However, it holds in that there is no difference numerically. The reason for insensitivity becomes clear on examination of eq. 25. When P_3 is large then the terms containing Y_0 become relatively small, and their effect on the solution becomes negligible. In such cases, the fitted value of the parameter has little significance.

The final example (Example 3) uses breakthrough curve data for tetrachloroethylene, also from the Borden experiment. The assumptions

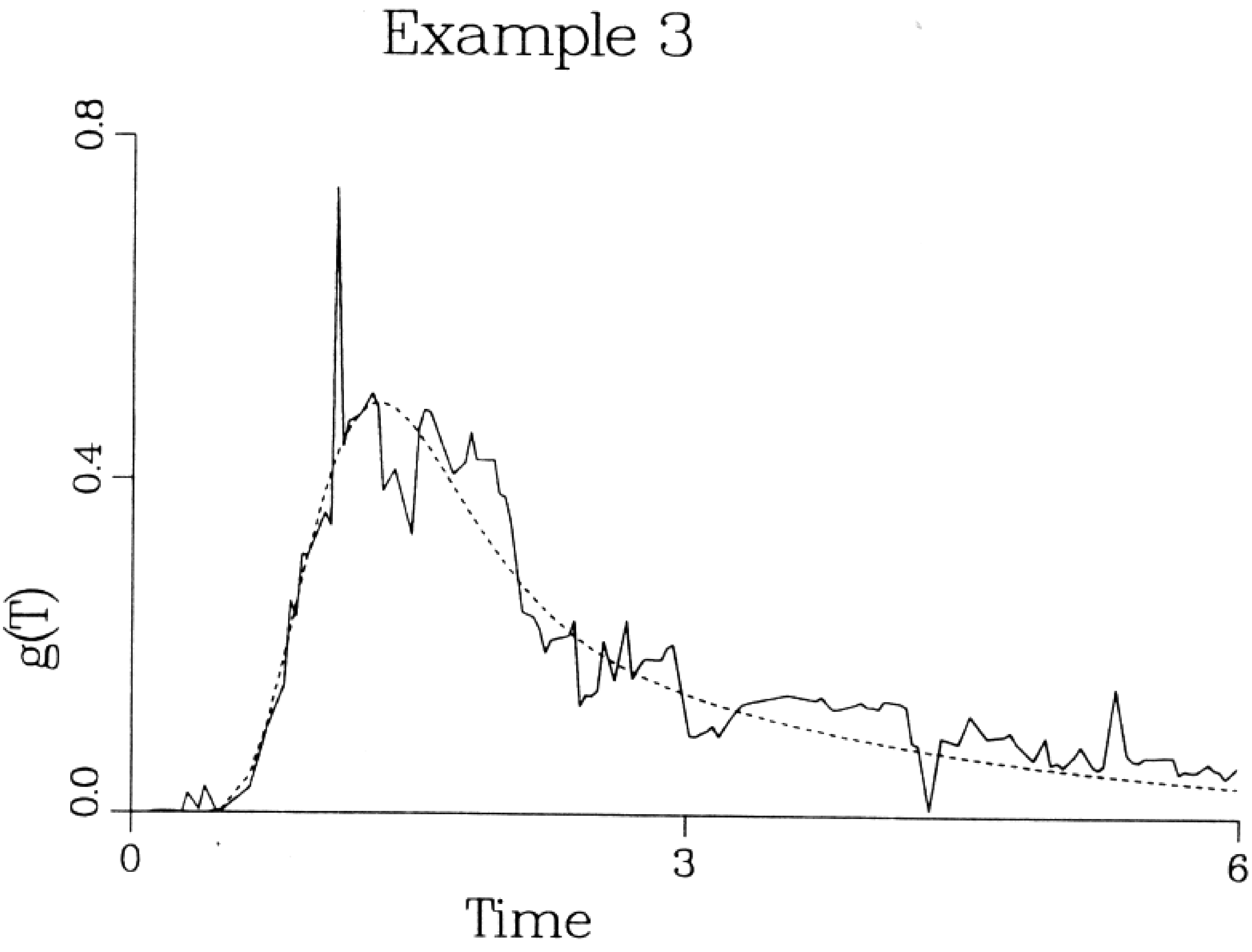


Fig. 3. TFMFIT results for the tetrachloroethylene data from the Borden experiment.

Table 5. TFMFIT results for Example 2.

```
Nonlinear Least-Squares Analysis.


    Example 2.  Borden Experiment Data, Carbon Tetrachloride
    Run on the San Diego Supercomputer Center's CRAY X-MP/48.

 Initial values of coefficients
 ==============================
   Name     Initial value
   P3........  4.0000e+01
   X3........  1.0000e+00
   Yo........  1.4000e+00
   B.........  1.0000e+00
   R.........  2.0000e+00
   W.........  0.
   D.........  0.
   C.........  0.

 Iteration      SSQ          P3....      Yo....       R.....
     0       6.7241e+00   4.0000e+01  1.4000e+00   2.0000e+00
     1       4.6922e+00   4.0967e+01  4.6245e+01   1.5904e+00
     2       6.9345e-01   4.5320e+01  4.6245e+01   1.7561e+00
     3       4.3173e-01   5.5005e+01  4.6245e+01   1.7429e+00
     4       4.2567e-01   5.6833e+01  4.6245e+01   1.7426e+00
     5       4.2564e-01   5.6937e+01  4.6245e+01   1.7424e+00
     6       4.2564e-01   5.6942e+01  4.6245e+01   1.7423e+00

   Correlation Matrix
   =================
         1          2          3
   1  1.0000
   2  0.0000     0.0000
   3  0.0008     0.0000     1.0000

 Rsquare for regression =    0.9742779735e+00.


Nonlinear Least Squares Analysis, Final Results.
================================================
                                                         95% Confidence Limits
 Variable    Name       Value        S.E.Coeff.  T-Value   Lower        Upper
     1       P3....   5.6942e+01     1.4994e+00 3.798e+01  5.3972e+01   5.9912e+01
     2       Yo....   4.6245e+01     6.0575e-32 7.634e+32  4.6245e+01   4.6245e+01
     3       R.....   1.7423e+00     4.5816e-03 3.803e+02  1.7333e+00   1.7514e+00

                                  Ordered by computer input
  No.        T          Conc Obs        Conc Fit      Residual
   1     1.492e-02     0.              2.855-709     -2.855-709
   2     2.536e-01     4.090e-02       3.134e-31      4.090e-02
   3     3.133e-01     4.165e-02       2.131e-23      4.165e-02
   4     3.580e-01     5.636e-02       2.744e-19      5.636e-02
   5     4.028e-01     4.706e-02       3.957e-16      4.706e-02
   6     4.177e-01     5.513e-02       3.099e-15      5.513e-02
   7     4.326e-01     8.881e-02       2.089e-14      8.881e-02
   8     4.475e-01     0.              1.230e-13     -1.230e-13
   9     4.625e-01     2.941e-02       6.401e-13      2.941e-02
  10     5.072e-01     2.695e-02       4.814e-11      2.695e-02
  11     5.221e-01     2.410e-02       1.699e-10      2.410e-02
  12     5.371e-01     3.947e-02       5.549e-10      3.947e-02
```

13	5.520e-01	3.103e-02	1.688e-09	3.103e-02
14	5.669e-01	3.862e-02	4.812e-09	3.862e-02
15	5.967e-01	2.941e-02	3.275e-08	2.941e-02
16	6.266e-01	8.160e-03	1.813e-07	8.160e-03
17	6.564e-01	9.500e-04	8.392e-07	9.492e-04
18	7.310e-01	9.490e-03	2.044e-05	9.470e-03
19	7.459e-01	1.679e-02	3.531e-05	1.675e-02
20	7.608e-01	2.239e-02	5.940e-05	2.233e-02
21	7.757e-01	1.661e-01	9.749e-05	1.660e-01
22	8.354e-01	1.528e-02	5.662e-04	1.471e-02
23	8.652e-01	1.499e-02	1.214e-03	1.378e-02
24	8.802e-01	1.072e-02	1.731e-03	8.989e-03
25	8.951e-01	9.490e-03	2.431e-03	7.059e-03
26	9.548e-01	7.097e-02	8.170e-03	6.280e-02
27	9.697e-01	1.101e-02	1.070e-02	3.069e-04
28	9.846e-01	1.006e-02	1.385e-02	-3.795e-03
29	9.995e-01	1.623e-02	1.773e-02	-1.500e-03
30	1.014e+00	1.101e-02	2.244e-02	-1.143e-02
31	1.044e+00	1.224e-02	3.485e-02	-2.261e-02
32	1.059e+00	1.803e-02	4.281e-02	-2.478e-02
33	1.074e+00	2.201e-02	5.210e-02	-3.009e-02
34	1.089e+00	2.571e-02	6.285e-02	-3.714e-02
35	1.134e+00	4.915e-02	1.050e-01	-5.584e-02
36	1.149e+00	5.295e-02	1.227e-01	-6.973e-02
37	1.164e+00	5.437e-02	1.423e-01	-8.795e-02
38	1.179e+00	6.585e-02	1.640e-01	-9.811e-02
39	1.193e+00	8.274e-02	1.876e-01	-1.049e-01
40	1.208e+00	1.091e-01	2.134e-01	-1.042e-01
41	1.223e+00	1.404e-01	2.411e-01	-1.007e-01
42	1.253e+00	1.670e-01	3.025e-01	-1.355e-01
43	1.268e+00	2.638e-01	3.361e-01	-7.229e-02
44	1.283e+00	3.131e-01	3.713e-01	-5.819e-02
45	1.298e+00	3.653e-01	4.081e-01	-4.283e-02
46	1.313e+00	4.270e-01	4.464e-01	-1.940e-02
47	1.328e+00	5.076e-01	4.858e-01	2.179e-02
48	1.372e+00	6.898e-01	6.096e-01	8.018e-02
49	1.387e+00	7.838e-01	6.519e-01	1.318e-01
50	1.402e+00	8.084e-01	6.943e-01	1.141e-01
51	1.447e+00	8.369e-01	8.194e-01	1.753e-02
52	1.462e+00	8.635e-01	8.596e-01	3.890e-03
53	1.477e+00	8.872e-01	8.986e-01	-1.143e-02
54	1.492e+00	8.748e-01	9.362e-01	-6.138e-02
55	1.507e+00	9.356e-01	9.722e-01	-3.668e-02
56	1.522e+00	9.394e-01	1.006e+00	-6.710e-02
57	1.551e+00	8.948e-01	1.069e+00	-1.739e-01
58	1.581e+00	1.044e+00	1.122e+00	-7.788e-02
59	1.596e+00	1.110e+00	1.144e+00	-3.414e-02
60	1.641e+00	1.167e+00	1.196e+00	-2.903e-02
61	1.701e+00	1.091e+00	1.226e+00	-1.352e-01
62	1.775e+00	1.063e+00	1.204e+00	-1.416e-01
63	1.790e+00	1.072e+00	1.193e+00	-1.206e-01
64	1.835e+00	1.177e+00	1.146e+00	3.062e-02
65	1.880e+00	1.148e+00	1.084e+00	6.442e-02
66	1.895e+00	1.082e+00	1.060e+00	2.151e-02
67	1.910e+00	1.091e+00	1.036e+00	5.566e-02
68	1.924e+00	9.678e-01	1.010e+00	-4.204e-02
69	1.939e+00	9.678e-01	9.834e-01	-1.554e-02
70	1.954e+00	9.489e-01	9.561e-01	-7.280e-03
71	2.014e+00	7.913e-01	8.425e-01	-5.116e-02
72	2.029e+00	7.543e-01	8.135e-01	-5.917e-02

```
 73    2.044e+00    7.012e-01    7.845e-01    -8.329e-02
 74    2.074e+00    6.500e-01    7.268e-01    -7.683e-02
 75    2.103e+00    6.405e-01    6.701e-01    -2.961e-02
 76    2.148e+00    5.114e-01    5.881e-01    -7.665e-02
 77    2.163e+00    5.266e-01    5.618e-01    -3.520e-02
 78    2.178e+00    5.143e-01    5.361e-01    -2.185e-02
 79    2.223e+00    3.625e-01    4.630e-01    -1.006e-01
 80    2.268e+00    3.169e-01    3.964e-01    -7.949e-02
 81    2.282e+00    3.046e-01    3.757e-01    -7.111e-02
 82    2.297e+00    2.771e-01    3.557e-01    -7.868e-02
 83    2.312e+00    2.657e-01    3.365e-01    -7.086e-02
 84    2.342e+00    2.144e-01    3.004e-01    -8.599e-02
 85    2.372e+00    2.040e-01    2.673e-01    -6.328e-02
 86    2.387e+00    1.917e-01    2.518e-01    -6.013e-02
 87    2.417e+00    1.670e-01    2.230e-01    -5.595e-02
 88    2.447e+00    1.623e-01    1.968e-01    -3.455e-02
 89    2.461e+00    1.698e-01    1.847e-01    -1.486e-02
 90    2.491e+00    1.395e-01    1.623e-01    -2.285e-02
 91    2.506e+00    1.338e-01    1.520e-01    -1.823e-02
 92    2.521e+00    1.214e-01    1.423e-01    -2.082e-02
 93    2.581e+00    1.158e-01    1.084e-01     7.347e-03
 94    2.596e+00    1.053e-01    1.011e-01     4.191e-03
 95    2.611e+00    9.868e-02    9.428e-02     4.402e-03
 96    2.626e+00    9.868e-02    8.784e-02     1.084e-02
 97    2.700e+00    7.496e-02    6.113e-02     1.383e-02
 98    2.715e+00    5.883e-02    5.676e-02     2.067e-03
 99    2.730e+00    1.452e-01    5.268e-02     9.249e-02
100    2.745e+00    6.357e-02    4.886e-02     1.471e-02
101    2.760e+00    1.414e-01    4.529e-02     9.609e-02
102    2.954e+00    6.262e-02    1.620e-02     4.642e-02
103    2.984e+00    5.541e-02    1.374e-02     4.167e-02
104    3.013e+00    5.152e-02    1.163e-02     3.989e-02
105    3.043e+00    4.934e-02    9.831e-03     3.951e-02
106    3.073e+00    4.754e-02    8.298e-03     3.924e-02
107    3.103e+00    4.744e-02    6.994e-03     4.045e-02
108    3.118e+00    4.991e-02    6.417e-03     4.349e-02
109    3.207e+00    4.289e-02    3.802e-03     3.909e-02
110    3.237e+00    3.862e-02    3.185e-03     3.544e-02
111    3.252e+00    4.298e-02    2.914e-03     4.007e-02
112    3.312e+00    4.640e-02    2.034e-03     4.437e-02
113    3.327e+00    3.682e-02    1.858e-03     3.496e-02
114    3.357e+00    3.871e-02    1.549e-03     3.716e-02
115    3.386e+00    3.729e-02    1.290e-03     3.600e-02
116    3.416e+00    3.568e-02    1.073e-03     3.461e-02
117    3.431e+00    3.558e-02    9.781e-04     3.460e-02
118    3.461e+00    3.463e-02    8.123e-04     3.382e-02
119    3.491e+00    3.568e-02    6.739e-04     3.501e-02

                          Ordered by residual
No.        T          Conc Obs     Conc Fit      Residual
 57    1.551e+00    8.948e-01    1.069e+00    -1.739e-01
 21    7.757e-01    1.661e-01    9.749e-05     1.660e-01
 62    1.775e+00    1.063e+00    1.204e+00    -1.416e-01
 42    1.253e+00    1.670e-01    3.025e-01    -1.355e-01
 61    1.701e+00    1.091e+00    1.226e+00    -1.352e-01
 49    1.387e+00    7.838e-01    6.519e-01     1.318e-01
 63    1.790e+00    1.072e+00    1.193e+00    -1.206e-01
 50    1.402e+00    8.084e-01    6.943e-01     1.141e-01
 39    1.193e+00    8.274e-02    1.876e-01    -1.049e-01
 40    1.208e+00    1.091e-01    2.134e-01    -1.042e-01
```

41	1.223e+00	1.404e-01	2.411e-01	-1.007e-01
79	2.223e+00	3.625e-01	4.630e-01	-1.006e-01
38	1.179e+00	6.585e-02	1.640e-01	-9.811e-02
101	2.760e+00	1.414e-01	4.529e-02	9.609e-02
99	2.730e+00	1.452e-01	5.268e-02	9.249e-02
7	4.326e-01	8.881e-02	2.089e-14	8.881e-02
37	1.164e+00	5.437e-02	1.423e-01	-8.795e-02
84	2.342e+00	2.144e-01	3.004e-01	-8.599e-02
73	2.044e+00	7.012e-01	7.845e-01	-8.329e-02
48	1.372e+00	6.898e-01	6.096e-01	8.018e-02
80	2.268e+00	3.169e-01	3.964e-01	-7.949e-02
82	2.297e+00	2.771e-01	3.557e-01	-7.868e-02
58	1.581e+00	1.044e+00	1.122e+00	-7.788e-02
74	2.074e+00	6.500e-01	7.268e-01	-7.683e-02
76	2.148e+00	5.114e-01	5.881e-01	-7.665e-02
43	1.268e+00	2.638e-01	3.361e-01	-7.229e-02
81	2.282e+00	3.046e-01	3.757e-01	-7.111e-02
83	2.312e+00	2.657e-01	3.365e-01	-7.086e-02
36	1.149e+00	5.295e-02	1.227e-01	-6.973e-02
56	1.522e+00	9.394e-01	1.006e+00	-6.710e-02
65	1.880e+00	1.148e+00	1.084e+00	6.442e-02
85	2.372e+00	2.040e-01	2.673e-01	-6.328e-02
26	9.548e-01	7.097e-02	8.170e-03	6.280e-02
54	1.492e+00	8.748e-01	9.362e-01	-6.138e-02
86	2.387e+00	1.917e-01	2.518e-01	-6.013e-02
72	2.029e+00	7.543e-01	8.135e-01	-5.917e-02
44	1.283e+00	3.131e-01	3.713e-01	-5.819e-02
4	3.580e-01	5.636e-02	2.744e-19	5.636e-02
87	2.417e+00	1.670e-01	2.230e-01	-5.595e-02
35	1.134e+00	4.915e-02	1.050e-01	-5.584e-02
67	1.910e+00	1.091e+00	1.036e+00	5.566e-02
6	4.177e-01	5.513e-02	3.099e-15	5.513e-02
71	2.014e+00	7.913e-01	8.425e-01	-5.116e-02
5	4.028e-01	4.706e-02	3.957e-16	4.706e-02
102	2.954e+00	6.262e-02	1.620e-02	4.642e-02
112	3.312e+00	4.640e-02	2.034e-03	4.437e-02
108	3.118e+00	4.991e-02	6.417e-03	4.349e-02
45	1.298e+00	3.653e-01	4.081e-01	-4.283e-02
68	1.924e+00	9.678e-01	1.010e+00	-4.204e-02
103	2.984e+00	5.541e-02	1.374e-02	4.167e-02
3	3.133e-01	4.165e-02	2.131e-23	4.165e-02
2	2.536e-01	4.090e-02	3.134e-31	4.090e-02
107	3.103e+00	4.744e-02	6.994e-03	4.045e-02
111	3.252e+00	4.298e-02	2.914e-03	4.007e-02
104	3.013e+00	5.152e-02	1.163e-02	3.989e-02
105	3.043e+00	4.934e-02	9.831e-03	3.951e-02
12	5.371e-01	3.947e-02	5.549e-10	3.947e-02
106	3.073e+00	4.754e-02	8.298e-03	3.924e-02
109	3.207e+00	4.289e-02	3.802e-03	3.909e-02
14	5.669e-01	3.862e-02	4.812e-09	3.862e-02
114	3.357e+00	3.871e-02	1.549e-03	3.716e-02
34	1.089e+00	2.571e-02	6.285e-02	-3.714e-02
55	1.507e+00	9.356e-01	9.722e-01	-3.668e-02
115	3.386e+00	3.729e-02	1.290e-03	3.600e-02
110	3.237e+00	3.862e-02	3.185e-03	3.544e-02
77	2.163e+00	5.266e-01	5.618e-01	-3.520e-02
119	3.491e+00	3.568e-02	6.739e-04	3.501e-02
113	3.327e+00	3.682e-02	1.858e-03	3.496e-02
116	3.416e+00	3.568e-02	1.073e-03	3.461e-02
117	3.431e+00	3.558e-02	9.781e-04	3.460e-02

```
 88    2.447e+00    1.623e-01    1.968e-01    -3.455e-02
 59    1.596e+00    1.110e+00    1.144e+00    -3.414e-02
118    3.461e+00    3.463e-02    8.123e-04     3.382e-02
 13    5.520e-01    3.103e-02    1.688e-09     3.103e-02
 64    1.835e+00    1.177e+00    1.146e+00     3.062e-02
 33    1.074e+00    2.201e-02    5.210e-02    -3.009e-02
 75    2.103e+00    6.405e-01    6.701e-01    -2.961e-02
  9    4.625e-01    2.941e-02    6.401e-13     2.941e-02
 15    5.967e-01    2.941e-02    3.275e-08     2.941e-02
 60    1.641e+00    1.167e+00    1.196e+00    -2.903e-02
 10    5.072e-01    2.695e-02    4.814e-11     2.695e-02
 32    1.059e+00    1.803e-02    4.281e-02    -2.478e-02
 11    5.221e-01    2.410e-02    1.699e-10     2.410e-02
 90    2.491e+00    1.395e-01    1.623e-01    -2.285e-02
 31    1.044e+00    1.224e-02    3.485e-02    -2.261e-02
 20    7.608e-01    2.239e-02    5.940e-05     2.233e-02
 78    2.178e+00    5.143e-01    5.361e-01    -2.185e-02
 47    1.328e+00    5.076e-01    4.858e-01     2.179e-02
 66    1.895e+00    1.082e+00    1.060e+00     2.151e-02
 92    2.521e+00    1.214e-01    1.423e-01    -2.082e-02
 46    1.313e+00    4.270e-01    4.464e-01    -1.940e-02
 91    2.506e+00    1.338e-01    1.520e-01    -1.823e-02
 51    1.447e+00    8.369e-01    8.194e-01     1.753e-02
 19    7.459e-01    1.679e-02    3.531e-05     1.675e-02
 69    1.939e+00    9.678e-01    9.834e-01    -1.554e-02
 89    2.461e+00    1.698e-01    1.847e-01    -1.486e-02
 22    8.354e-01    1.528e-02    5.662e-04     1.471e-02
100    2.745e+00    6.357e-02    4.886e-02     1.471e-02
 97    2.700e+00    7.496e-02    6.113e-02     1.383e-02
 23    8.652e-01    1.499e-02    1.214e-03     1.378e-02
 30    1.014e+00    1.101e-02    2.244e-02    -1.143e-02
 53    1.477e+00    8.872e-01    8.986e-01    -1.143e-02
 96    2.626e+00    9.868e-02    8.784e-02     1.084e-02
 18    7.310e-01    9.490e-03    2.044e-05     9.470e-03
 24    8.802e-01    1.072e-02    1.731e-03     8.989e-03
 16    6.266e-01    8.160e-03    1.813e-07     8.160e-03
 93    2.581e+00    1.158e-01    1.084e-01     7.347e-03
 70    1.954e+00    9.489e-01    9.561e-01    -7.280e-03
 25    8.951e-01    9.490e-03    2.431e-03     7.059e-03
 95    2.611e+00    9.868e-02    9.428e-02     4.402e-03
 94    2.596e+00    1.053e-01    1.011e-01     4.191e-03
 52    1.462e+00    8.635e-01    8.596e-01     3.890e-03
 28    9.846e-01    1.006e-02    1.385e-02    -3.795e-03
 98    2.715e+00    5.883e-02    5.676e-02     2.067e-03
 29    9.995e-01    1.623e-02    1.773e-02    -1.500e-03
 17    6.564e-01    9.500e-04    8.392e-07     9.492e-04
 27    9.697e-01    1.101e-02    1.070e-02     3.069e-04
  8    4.475e-01    0.           1.230e-13    -1.230e-13
  1    1.492e-02    0.           2.855-709    -2.855-709
```

made regarding the model parameters for carbon tetrachloride apply to tetachloroethylene except that, in this case, some kinetically-controlled partitioning of the solute between the liquid and solid phases is expected (f < 1 in eq. 8; i.e., B < 1). The parameters D and C (eqs. 11 and 12) were assigned values of zero based on the physical behavior of the chemical. The other parameters were fitted with the results given in Table 6 and Fig. 3. Again, the optimal model fit is quite good and the model is insensitive to Y_0, as shown by the correlation matrix in Table 6.

Plotting the TFMFIT Results

The data provided in the main TFMFIT output file (Tables 4-6) are essential for the purpose of model fitting. Often, however, a plot of the experimental data and model fit is required. The figures in this document were produced using the TFMPLOT code contained in Appendix 3. TFMFIT produces, if requested, a plot file. A particular case is given in Table 7, which is based on Example 1a (Fig. 1a). TFMPLOT reads the plot file data, as well as a parameter file, to produce a device-compatible file. In creating TFMPLOT, we have used the DISSPLA software package (Integrated Software Systems Corporation, 1985). Therefore, those wishing to use TFMPLOT will have to have access to the DISSPLA library on their system. The output devices supported by this software are IMAGEN and Tektronics terminals.

Table 6. TFMFIT results for Example 3.

```
Nonlinear Least-Squares Analysis.


    Example 3.  Borden Experiment Data, Tetrachlorlethylene
    Run on the San Diego Supercomputer Center's CRAY X-MP/48.

  Initial values of coefficients
  ==============================
    Name     Initial value
    P3........  2.0000e+01
    X3........  1.0000e+00
    Yo........  1.5000e+00
    B.........  5.0000e-01
    R.........  1.5000e+00
    W.........  1.0000e+00
    D.........  0.
    C.........  0.

  Iteration      SSQ            P3....       Yo....      B.....       R.....       W.....
      0       3.2681e+00     2.0000e+01   1.5000e+00   5.0000e-01   1.5000e+00   1.0000e+00
      1       1.9935e+00     1.9389e+01   1.5502e+01   4.1564e-01   1.6677e+00   1.5312e+00
      2       9.4697e-01     9.7706e+00   1.5502e+01   3.7841e-01   2.0211e+00   2.7727e+00
      3       6.8663e-01     9.9440e+00   1.5502e+01   4.7432e-01   1.9051e+00   3.5592e+00
      4       6.2469e-01     8.9514e+00   1.5502e+01   5.9549e-01   1.8349e+00   2.9327e+00
      5       5.8035e-01     8.2611e+00   1.5502e+01   7.0600e-01   1.8018e+00   1.8225e+00
      6       4.4876e-01     8.8292e+00   1.5502e+01   7.8083e-01   1.8481e+00   4.6663e-01
      7       4.0905e-01     9.5322e+00   1.5502e+01   7.6829e-01   2.0613e+00   2.1891e-01
      8       3.8109e-01     1.4751e+01   1.5502e+01   7.0294e-01   2.0309e+00   7.6583e-01
      9       3.2548e-01     1.3925e+01   1.5502e+01   6.8291e-01   2.1716e+00   5.2633e-01
     10       2.8981e-01     1.6211e+01   1.5502e+01   6.0643e-01   2.3788e+00   6.9709e-01
     11       2.8512e-01     1.8268e+01   1.5502e+01   5.8151e-01   2.4744e+00   7.6905e-01
     12       2.8502e-01     1.7614e+01   1.5502e+01   5.8772e-01   2.4660e+00   7.3865e-01
     13       2.8500e-01     1.7992e+01   1.5502e+01   5.8584e-01   2.4658e+00   7.5460e-01
     14       2.8498e-01     1.7931e+01   1.5502e+01   5.8585e-01   2.4685e+00   7.5003e-01
     15       2.8498e-01     1.7839e+01   1.5502e+01   5.8659e-01   2.4672e+00   7.4616e-01
     16       2.8498e-01     1.7878e+01   1.5502e+01   5.8646e-01   2.4663e+00   7.4905e-01
     17       2.8498e-01     1.7849e+01   1.5502e+01   5.8652e-01   2.4665e+00   7.4800e-01

  No further decrease in SSQ obtained after 30 trials.

    Correlation Matrix
    ==================
          1          2          3          4          5
    1   1.0000
    2   0.0000     0.0000
    3  -0.6188     0.0000     1.0000
    4   0.1426     0.0000    -0.7825     1.0000
    5   0.8588     0.0000    -0.5143    -0.0568     1.0000

  Rsquare for regression =    0.8841283406e+00.


Nonlinear Least Squares Analysis, Final Results.
================================================
                                                              95% Confidence Limits
  Variable    Name       Value         S.E.Coeff.  T-Value    Lower         Upper
      1       P3....   1.7849e+01      2.5896e+00 6.893e+00  1.2718e+01    2.2980e+01
      2       Yo....   1.5502e+01      5.0443e-32 3.073e+32  1.5502e+01    1.5502e+01
      3       B.....   5.8652e-01      2.5523e-02 2.298e+01  5.3595e-01    6.3709e-01
```

```
   4      R.....    2.4665e+00      9.9915e-02 2.469e+01  2.2686e+00    2.6645e+00
   5      W.....    7.4800e-01      1.1927e-01 6.272e+00  5.1168e-01    9.8432e-01

                              Ordered by computer input
No.          T            Conc Obs        Conc Fit        Residual
  1      9.349e-02       0.              8.808e-10      -8.808e-10
  2      1.247e-01       1.970e-03       4.252e-08       1.970e-03
  3      2.181e-01       2.090e-03       0.              2.090e-03
  4      2.805e-01       1.130e-03       6.292e-07       1.129e-03
  5      3.116e-01       2.358e-02       4.500e-06       2.358e-02
  6      3.740e-01       3.510e-03       1.013e-04       3.409e-03
  7      4.051e-01       3.215e-02       3.282e-04       3.182e-02
  8      4.675e-01       3.680e-03       2.046e-03       1.634e-03
  9      4.986e-01       2.750e-03       4.203e-03      -1.453e-03
 10      6.545e-01       3.013e-02       4.604e-02      -1.591e-02
 11      7.479e-01       1.014e-01       1.064e-01      -4.978e-03
 12      8.414e-01       1.524e-01       1.888e-01      -3.642e-02
 13      8.726e-01       2.538e-01       2.188e-01       3.493e-02
 14      9.038e-01       2.352e-01       2.492e-01      -1.395e-02
 15      9.349e-01       3.100e-01       2.792e-01       3.078e-02
 16      9.661e-01       3.071e-01       3.084e-01      -1.298e-03
 17      1.060e+00       3.592e-01       3.862e-01      -2.704e-02
 18      1.091e+00       3.447e-01       4.079e-01      -6.316e-02
 19      1.122e+00       7.387e-01       4.270e-01       3.117e-01
 20      1.153e+00       4.374e-01       4.434e-01      -6.024e-03
 21      1.184e+00       4.664e-01       4.572e-01       9.152e-03
 22      1.247e+00       4.751e-01       4.769e-01      -1.869e-03
 23      1.309e+00       4.983e-01       4.867e-01       1.151e-02
 24      1.340e+00       4.838e-01       4.883e-01      -4.501e-03
 25      1.371e+00       3.853e-01       4.878e-01      -1.025e-01
 26      1.434e+00       4.114e-01       4.816e-01      -7.028e-02
 27      1.527e+00       3.331e-01       4.621e-01      -1.289e-01
 28      1.558e+00       4.577e-01       4.535e-01       4.170e-03
 29      1.589e+00       4.809e-01       4.442e-01       3.664e-02
 30      1.621e+00       4.780e-01       4.344e-01       4.359e-02
 31      1.745e+00       4.055e-01       3.918e-01       1.380e-02
 32      1.808e+00       4.200e-01       3.699e-01       5.012e-02
 33      1.839e+00       4.548e-01       3.591e-01       9.566e-02
 34      1.870e+00       4.229e-01       3.485e-01       7.440e-02
 35      1.963e+00       4.229e-01       3.181e-01       1.048e-01
 36      1.995e+00       3.824e-01       3.086e-01       7.381e-02
 37      2.026e+00       3.795e-01       2.993e-01       8.015e-02
 38      2.057e+00       3.476e-01       2.904e-01       5.721e-02
 39      2.088e+00       2.955e-01       2.818e-01       1.364e-02
 40      2.119e+00       2.422e-01       2.736e-01      -3.141e-02
 41      2.182e+00       2.361e-01       2.581e-01      -2.201e-02
 42      2.213e+00       2.219e-01       2.509e-01      -2.895e-02
 43      2.244e+00       1.929e-01       2.439e-01      -5.099e-02
 44      2.275e+00       2.092e-01       2.373e-01      -2.816e-02
 45      2.368e+00       2.144e-01       2.192e-01      -4.859e-03
 46      2.400e+00       2.317e-01       2.137e-01       1.800e-02
 47      2.431e+00       1.295e-01       2.085e-01      -7.902e-02
 48      2.462e+00       1.425e-01       2.035e-01      -6.100e-02
 49      2.493e+00       1.417e-01       1.988e-01      -5.711e-02
 50      2.524e+00       1.477e-01       1.942e-01      -4.648e-02
 51      2.555e+00       2.077e-01       1.899e-01       1.782e-02
 52      2.618e+00       1.599e-01       1.818e-01      -2.188e-02
 53      2.680e+00       2.329e-01       1.744e-01       5.854e-02
 54      2.711e+00       1.628e-01       1.709e-01      -8.085e-03
 55      2.742e+00       1.741e-01       1.675e-01       6.556e-03
 56   2.774e+00    1.866e-01    1.643e-01    2.221e-02
 57   2.867e+00    1.857e-01    1.554e-01    3.025e-02
 58   2.898e+00    1.999e-01    1.527e-01    4.719e-02
 59   2.929e+00    2.048e-01    1.500e-01    5.477e-02
 60   3.023e+00    9.415e-02    1.426e-01   -4.843e-02
 61   3.054e+00    9.415e-02    1.402e-01   -4.610e-02
 62   3.085e+00    9.791e-02    1.380e-01   -4.007e-02
 63   3.116e+00    1.011e-01    1.358e-01   -3.469e-02
 64   3.148e+00    1.078e-01    1.336e-01   -2.589e-02
 65   3.179e+00    9.473e-02    1.316e-01   -3.684e-02
 66   3.241e+00    1.150e-01    1.276e-01   -1.257e-02
 67   3.303e+00    1.318e-01    1.238e-01    8.043e-03
 68   3.335e+00    1.347e-01    1.219e-01    1.278e-02
 69   3.553e+00    1.445e-01    1.101e-01    3.449e-02
 70   3.709e+00    1.382e-01    1.025e-01    3.568e-02
 71   3.740e+00    1.422e-01    1.011e-01    4.117e-02
 72   3.802e+00    1.275e-01    9.825e-02    2.921e-02
 73   3.833e+00    1.286e-01    9.688e-02    3.174e-02
 74   3.927e+00    1.335e-01    9.288e-02    4.066e-02
 75   3.958e+00    1.367e-01    9.158e-02    4.515e-02
 76   3.989e+00    1.309e-01    9.030e-02    4.064e-02
 77   4.020e+00    1.307e-01    8.905e-02    4.160e-02
 78   4.051e+00    1.280e-01    8.782e-02    4.022e-02
 79   4.083e+00    1.382e-01    8.660e-02    5.158e-02
 80   4.176e+00    1.347e-01    8.304e-02    5.166e-02
 81   4.207e+00    1.307e-01    8.188e-02    4.877e-02
 82   4.238e+00    8.777e-02    8.074e-02    7.028e-03
 83   4.270e+00    8.546e-02    7.962e-02    5.842e-03
 84   4.332e+00    7.130e-03    7.742e-02   -7.029e-02
 85   4.394e+00    9.444e-02    7.528e-02    1.916e-02
 86   4.488e+00    8.748e-02    7.217e-02    1.531e-02
 87   4.519e+00    1.014e-01    7.116e-02    3.023e-02
 88   4.550e+00    1.214e-01    7.017e-02    5.121e-02
 89   4.643e+00    9.473e-02    6.726e-02    2.747e-02
 90   4.737e+00    9.675e-02    6.447e-02    3.228e-02
 91   4.768e+00    1.011e-01    6.356e-02    3.754e-02
 92   4.799e+00    8.864e-02    6.267e-02    2.597e-02
 93   4.893e+00    6.894e-02    6.005e-02    8.886e-03
 94   4.955e+00    9.502e-02    5.837e-02    3.665e-02
 95   4.986e+00    6.315e-02    5.754e-02    5.605e-03
 96   5.017e+00    6.692e-02    5.673e-02    1.019e-02
 97   5.049e+00    5.996e-02    5.593e-02    4.033e-03
 98   5.111e+00    7.329e-02    5.435e-02    1.894e-02
 99   5.142e+00    8.517e-02    5.358e-02    3.159e-02
100   5.204e+00    6.257e-02    5.207e-02    1.050e-02
101   5.236e+00    5.794e-02    5.133e-02    6.614e-03
102   5.267e+00    6.286e-02    5.060e-02    1.226e-02
103   5.329e+00    1.541e-01    4.916e-02    1.049e-01
104   5.391e+00    7.590e-02    4.777e-02    2.813e-02
105   5.423e+00    6.894e-02    4.709e-02    2.185e-02
106   5.454e+00    6.721e-02    4.641e-02    2.080e-02
107   5.485e+00    7.213e-02    4.575e-02    2.638e-02
108   5.641e+00    7.271e-02    4.255e-02    3.016e-02
109   5.672e+00    5.272e-02    4.194e-02    1.078e-02
110   5.703e+00    5.736e-02    4.133e-02    1.603e-02
111   5.734e+00    5.504e-02    4.074e-02    1.430e-02
112   5.765e+00    5.591e-02    4.015e-02    1.576e-02
113   5.828e+00    6.634e-02    3.900e-02    2.734e-02
114   5.859e+00    5.938e-02    3.843e-02    2.095e-02
115   5.890e+00    5.707e-02    3.788e-02    1.919e-02
```

```
116     5.921e+00      4.809e-02      3.733e-02      1.076e-02
117     5.984e+00      6.112e-02      3.625e-02      2.487e-02

                           Ordered by residual
No.        T           Conc Obs       Conc Fit       Residual
 19     1.122e+00      7.387e-01      4.270e-01      3.117e-01
 27     1.527e+00      3.331e-01      4.621e-01     -1.289e-01
103     5.329e+00      1.541e-01      4.916e-02      1.049e-01
 35     1.963e+00      4.229e-01      3.181e-01      1.048e-01
 25     1.371e+00      3.853e-01      4.878e-01     -1.025e-01
 33     1.839e+00      4.548e-01      3.591e-01      9.566e-02
 37     2.026e+00      3.795e-01      2.993e-01      8.015e-02
 47     2.431e+00      1.295e-01      2.085e-01     -7.902e-02
 34     1.870e+00      4.229e-01      3.485e-01      7.440e-02
 36     1.995e+00      3.824e-01      3.086e-01      7.381e-02
 84     4.332e+00      7.130e-03      7.742e-02     -7.029e-02
 26     1.434e+00      4.114e-01      4.816e-01     -7.028e-02
 18     1.091e+00      3.447e-01      4.079e-01     -6.316e-02
 48     2.462e+00      1.425e-01      2.035e-01     -6.100e-02
 53     2.680e+00      2.329e-01      1.744e-01      5.854e-02
 38     2.057e+00      3.476e-01      2.904e-01      5.721e-02
 49     2.493e+00      1.417e-01      1.988e-01     -5.711e-02
 59     2.929e+00      2.048e-01      1.500e-01      5.477e-02
 80     4.176e+00      1.347e-01      8.304e-02      5.166e-02
 79     4.083e+00      1.382e-01      8.660e-02      5.158e-02
 88     4.550e+00      1.214e-01      7.017e-02      5.121e-02
 43     2.244e+00      1.929e-01      2.439e-01     -5.099e-02
 32     1.808e+00      4.200e-01      3.699e-01      5.012e-02
 81     4.207e+00      1.307e-01      8.188e-02      4.877e-02
 60     3.023e+00      9.415e-02      1.426e-01     -4.843e-02
 58     2.898e+00      1.999e-01      1.527e-01      4.719e-02
 50     2.524e+00      1.477e-01      1.942e-01     -4.648e-02
 61     3.054e+00      9.415e-02      1.402e-01     -4.610e-02
 75     3.958e+00      1.367e-01      9.158e-02      4.515e-02
 30     1.621e+00      4.780e-01      4.344e-01      4.359e-02
 77     4.020e+00      1.307e-01      8.905e-02      4.160e-02
 71     3.740e+00      1.422e-01      1.011e-01      4.117e-02
 74     3.927e+00      1.335e-01      9.288e-02      4.066e-02
 76     3.989e+00      1.309e-01      9.030e-02      4.064e-02
 78     4.051e+00      1.280e-01      8.782e-02      4.022e-02
 62     3.085e+00      9.791e-02      1.380e-01     -4.007e-02
 91     4.768e+00      1.011e-01      6.356e-02      3.754e-02
 65     3.179e+00      9.473e-02      1.316e-01     -3.684e-02
 94     4.955e+00      9.502e-02      5.837e-02      3.665e-02
 29     1.589e+00      4.809e-01      4.442e-01      3.664e-02
 12     8.414e-01      1.524e-01      1.888e-01     -3.642e-02
 70     3.709e+00      1.382e-01      1.025e-01      3.568e-02
 13     8.726e-01      2.538e-01      2.188e-01      3.493e-02
 63     3.116e+00      1.011e-01      1.358e-01     -3.469e-02
 69     3.553e+00      1.445e-01      1.101e-01      3.449e-02
 90     4.737e+00      9.675e-02      6.447e-02      3.228e-02
  7     4.051e-01      3.215e-02      3.282e-04      3.182e-02
 73     3.833e+00      1.286e-01      9.688e-02      3.174e-02
 99     5.142e+00      8.517e-02      5.358e-02      3.159e-02
 40     2.119e+00      2.422e-01      2.736e-01     -3.141e-02
 15     9.349e-01      3.100e-01      2.792e-01      3.078e-02
 57     2.867e+00      1.857e-01      1.554e-01      3.025e-02
 87     4.519e+00      1.014e-01      7.116e-02      3.023e-02
108     5.641e+00      7.271e-02      4.255e-02      3.016e-02
 72     3.802e+00      1.275e-01      9.825e-02      2.921e-02
```

42	2.213e+00	2.219e-01	2.509e-01	-2.895e-02
44	2.275e+00	2.092e-01	2.373e-01	-2.816e-02
104	5.391e+00	7.590e-02	4.777e-02	2.813e-02
89	4.643e+00	9.473e-02	6.726e-02	2.747e-02
113	5.828e+00	6.634e-02	3.900e-02	2.734e-02
17	1.060e+00	3.592e-01	3.862e-01	-2.704e-02
107	5.485e+00	7.213e-02	4.575e-02	2.638e-02
92	4.799e+00	8.864e-02	6.267e-02	2.597e-02
64	3.148e+00	1.078e-01	1.336e-01	-2.589e-02
117	5.984e+00	6.112e-02	3.625e-02	2.487e-02
5	3.116e-01	2.358e-02	4.500e-06	2.358e-02
56	2.774e+00	1.866e-01	1.643e-01	2.221e-02
41	2.182e+00	2.361e-01	2.581e-01	-2.201e-02
52	2.618e+00	1.599e-01	1.818e-01	-2.188e-02
105	5.423e+00	6.894e-02	4.709e-02	2.185e-02
114	5.859e+00	5.938e-02	3.843e-02	2.095e-02
106	5.454e+00	6.721e-02	4.641e-02	2.080e-02
115	5.890e+00	5.707e-02	3.788e-02	1.919e-02
85	4.394e+00	9.444e-02	7.528e-02	1.916e-02
98	5.111e+00	7.329e-02	5.435e-02	1.894e-02
46	2.400e+00	2.317e-01	2.137e-01	1.800e-02
51	2.555e+00	2.077e-01	1.899e-01	1.782e-02
110	5.703e+00	5.736e-02	4.133e-02	1.603e-02
10	6.545e-01	3.013e-02	4.604e-02	-1.591e-02
112	5.765e+00	5.591e-02	4.015e-02	1.576e-02
86	4.488e+00	8.748e-02	7.217e-02	1.531e-02
111	5.734e+00	5.504e-02	4.074e-02	1.430e-02
14	9.038e-01	2.352e-01	2.492e-01	-1.395e-02
31	1.745e+00	4.055e-01	3.918e-01	1.380e-02
39	2.088e+00	2.955e-01	2.818e-01	1.364e-02
68	3.335e+00	1.347e-01	1.219e-01	1.278e-02
66	3.241e+00	1.150e-01	1.276e-01	-1.257e-02
102	5.267e+00	6.286e-02	5.060e-02	1.226e-02
23	1.309e+00	4.983e-01	4.867e-01	1.151e-02
109	5.672e+00	5.272e-02	4.194e-02	1.078e-02
116	5.921e+00	4.809e-02	3.733e-02	1.076e-02
100	5.204e+00	6.257e-02	5.207e-02	1.050e-02
96	5.017e+00	6.692e-02	5.673e-02	1.019e-02
21	1.184e+00	4.664e-01	4.572e-01	9.152e-03
93	4.893e+00	6.894e-02	6.005e-02	8.886e-03
54	2.711e+00	1.628e-01	1.709e-01	-8.085e-03
67	3.303e+00	1.318e-01	1.238e-01	8.043e-03
82	4.238e+00	8.777e-02	8.074e-02	7.028e-03
101	5.236e+00	5.794e-02	5.133e-02	6.614e-03
55	2.742e+00	1.741e-01	1.675e-01	6.556e-03
20	1.153e+00	4.374e-01	4.434e-01	-6.024e-03
83	4.270e+00	8.546e-02	7.962e-02	5.842e-03
95	4.986e+00	6.315e-02	5.754e-02	5.605e-03
11	7.479e-01	1.014e-01	1.064e-01	-4.978e-03
45	2.368e+00	2.144e-01	2.192e-01	-4.859e-03
24	1.340e+00	4.838e-01	4.883e-01	-4.501e-03
28	1.558e+00	4.577e-01	4.535e-01	4.170e-03
97	5.049e+00	5.996e-02	5.593e-02	4.033e-03
6	3.740e-01	3.510e-03	1.013e-04	3.409e-03
3	2.181e-01	2.090e-03	0.	2.090e-03
2	1.247e-01	1.970e-03	4.252e-08	1.970e-03
22	1.247e+00	4.751e-01	4.769e-01	-1.869e-03
8	4.675e-01	3.680e-03	2.046e-03	1.634e-03
9	4.986e-01	2.750e-03	4.203e-03	-1.453e-03
16	9.661e-01	3.071e-01	3.084e-01	-1.298e-03
4	2.805e-01	1.130e-03	6.292e-07	1.129e-03
1	9.349e-02	0.	8.808e-10	-8.808e-10

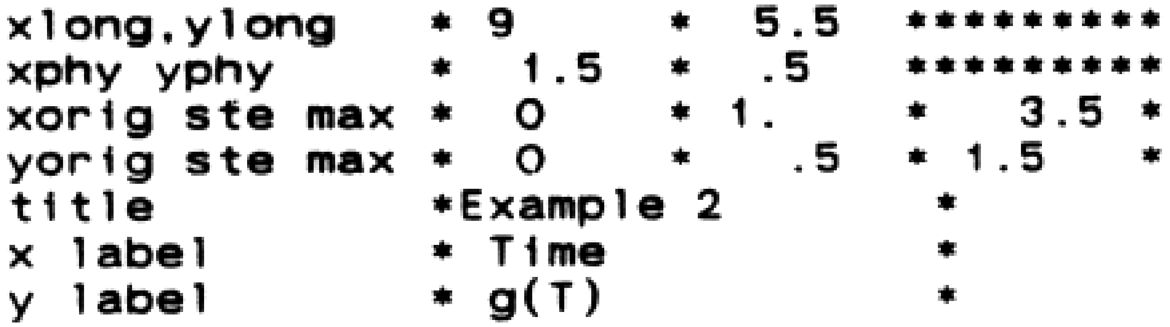
Table 7. Plot file produced by TFMFIT for Example 1a.

These data are plotted in Fig. 1a.

```
6.0000e-03      2.5000e-02  1
2.8900e-01      5.0000e-02  2
7.8700e-01      7.5000e-02  3
1.1340e+00      1.0000e-01  4
1.3000e+00      1.2500e-01  5
1.3460e+00      1.5000e-01  6
1.3260e+00      1.7500e-01  7
1.2710e+00      2.0000e-01  8
1.2020e+00      2.2500e-01  9
1.1280e+00      2.5000e-01 10
1.0550e+00      2.7500e-01 11
9.8500e-01      3.0000e-01 12
9.1900e-01      3.2500e-01 13
8.5800e-01      3.5000e-01 14
8.0200e-01      3.7500e-01 15
7.5100e-01      4.0000e-01 16
7.0300e-01      4.2500e-01 17
6.6000e-01      4.5000e-01 18
6.2000e-01      4.7500e-01 19
5.8400e-01      5.0000e-01 20
5.5000e-01      5.2500e-01 21
5.1900e-01      5.5000e-01 22
4.9100e-01      5.7500e-01 23
4.6400e-01      6.0000e-01 24
4.1700e-01      6.5000e-01 25
3.7500e-01      7.0000e-01 26
3.3900e-01      7.5000e-01 27
3.0800e-01      8.0000e-01 28
2.8000e-01      8.5000e-01 29
2.5600e-01      9.0000e-01 30
2.3400e-01      9.5000e-01 31
2.1400e-01      1.0000e+00 32
1.9700e-01      1.0500e+00 33
1.8100e-01      1.1000e+00 34
1.6700e-01      1.1500e+00 35
1.5400e-01      1.2000e+00 36
1.4200e-01      1.2500e+00 37
1.3200e-01      1.3000e+00 38
1.2200e-01      1.3500e+00 39
1.1300e-01      1.4000e+00 40
1.0500e-01      1.4500e+00 41
9.8000e-02      1.5000e+00 42
9.1000e-02      1.5500e+00 43
8.5000e-02      1.6000e+00 44
8.0000e-02      1.6500e+00 45
7.5000e-02      1.7000e+00 46
7.0000e-02      1.7500e+00 47
6.5000e-02      1.8000e+00 48
6.1000e-02      1.8500e+00 49
5.7000e-02      1.9000e+00 50
5.4000e-02      1.9500e+00 51
5.1000e-02      2.0000e+00 52
4.8000e-02      2.0500e+00 53
4.5000e-02      2.1000e+00 54
4.2000e-02      2.1500e+00 55
4.0000e-02      2.2000e+00 56
3.8000e-02      2.2500e+00 57
3.5000e-02      2.3000e+00 58
3.3000e-02      2.3500e+00 59
3.2000e-02      2.4000e+00 60
3.0000e-02      2.4500e+00 61
2.8000e-02      2.5000e+00 62
6.4785e-03      2.5000e-02  1
2.8879e-01      5.0000e-02  2
7.8710e-01      7.5000e-02  3
1.1343e+00      1.0000e-01  4
1.2998e+00      1.2500e-01  5
1.3459e+00      1.5000e-01  6
1.3255e+00      1.7500e-01  7
1.2714e+00      2.0000e-01  8
1.2023e+00      2.2500e-01  9
1.1284e+00      2.5000e-01 10
1.0549e+00      2.7500e-01 11
9.8474e-01      3.0000e-01 12
9.1898e-01      3.2500e-01 13
8.5805e-01      3.5000e-01 14
8.0198e-01      3.7500e-01 15
7.5050e-01      4.0000e-01 16
7.0330e-01      4.2500e-01 17
6.6006e-01      4.5000e-01 18
6.2033e-01      4.7500e-01 19
5.8382e-01      5.0000e-01 20
5.5020e-01      5.2500e-01 21
5.1919e-01      5.5000e-01 22
4.9053e-01      5.7500e-01 23
4.6399e-01      6.0000e-01 24
4.1649e-01      6.5000e-01 25
3.7533e-01      7.0000e-01 26
3.3943e-01      7.5000e-01 27
3.0794e-01      8.0000e-01 28
2.8016e-01      8.5000e-01 29
2.5556e-01      9.0000e-01 30
2.3368e-01      9.5000e-01 31
2.1414e-01      1.0000e+00 32
1.9663e-01      1.0500e+00 33
1.8089e-01      1.1000e+00 34
1.6671e-01      1.1500e+00 35
1.5390e-01      1.2000e+00 36
1.4230e-01      1.2500e+00 37
1.3177e-01      1.3000e+00 38
1.2218e-01      1.3500e+00 39
1.1345e-01      1.4000e+00 40
1.0548e-01      1.4500e+00 41
9.8178e-02      1.5000e+00 42
9.1490e-02      1.5500e+00 43
8.5351e-02      1.6000e+00 44
7.9708e-02      1.6500e+00 45
7.4512e-02      1.7000e+00 46
6.9721e-02      1.7500e+00 47
6.5297e-02      1.8000e+00 48
6.1208e-02      1.8500e+00 49
5.7423e-02      1.9000e+00 50
5.3916e-02      1.9500e+00 51
5.0661e-02      2.0000e+00 52
4.7638e-02      2.0500e+00 53
4.4828e-02      2.1000e+00 54
4.2212e-02      2.1500e+00 55
3.9775e-02      2.2000e+00 56
3.7502e-02      2.2500e+00 57
3.5381e-02      2.3000e+00 58
3.3399e-02      2.3500e+00 59
3.1545e-02      2.4000e+00 60
2.9811e-02      2.4500e+00 61
2.8187e-02      2.5000e+00 62
```

An example parameter file required by TFMPLOT is given in Table 8. This file was used to create Fig. 2. The structure of the file is as follows: line 1 controls the total size of the plot; line 2 controls the position of the axes relative to the assumed physical edges of the plot (these first two lines normally will be unchanged); lines 3 and 4 give the limits and tick increments on the x and y axes, respectively; and lines 4-7 contain descriptive labels.

Table 8. Parameter file for TFMPLOT. An explanation of the file is given in the text.



```
xlong.ylong   * 9     *  5.5   *********
xphy yphy     *  1.5  *  .5    *********
xorig ste max *  O    * 1.     *   3.5  *
yorig ste max *  O    *   .5   * 1.5    *
title         *Example 2       *
x label       * Time           *
y label       * g(T)           *
```

REFERENCES

Barry, D. A. 1988. On "Elimination of time assignment bias in estimates of dispersion coefficient". Soil Sci. Soc. Am. J. 52:896-897.

Barry, D. A., and G. Sposito. 1988. Application of the convection-dispersion model to solute transport in finite soil columns. Soil Sci. Soc. Am. J. 52:3-9.

Beck, J. V., and K. J. Arnold. 1977. Parameter estimation in engineering and science. John Wiley & Sons, New York.

Carslaw, H. S., and J. C. Jaeger. 1959. Conduction of heat in solids. 2nd ed., Oxford, London.

Curtis, G. P., P. V. Roberts, and M. Reinhard. 1986. A natural gradient experiment on solute transport in a sand aquifer. 4. Sorption of organic solutes and its influence on mobility. Water Resour. Res. 22(13):2059-2067.

Daniel, C., and F. S. Wood. 1973. Fitting equations to data. Wiley-Interscience, New York.

De Smedt, F., and P. J. Wierenga. 1979. A generalized solution for solute flow in soils with mobile-immobile water. Water Resour. Res. 15:1137-1141.

Freyberg, D. L. 1986. A natural gradient experiment on solute transport in a sand aquifer. 2. Spatial moments and the advection and dispersion of nonreactive tracers. Water Resour. Res. 22:2031-2046.

Goldstein, S. 1953. On the mathematics of exchange processes in fixed columns. I. Mathematical solutions and asymptotic expansions. Proc. Roy. Soc. London. 219:151-185.

Goltz, M. N., and P. V. Roberts. 1986. Interpreting organic solute transport data from a field experiment using physical nonequilibrium models. J. Contaminant Hydrol. 1:77-93.

Hendrickson, J. D., S. Sorooshian, and L. E. Brazil. 1988. Comparison of Newton-type and direct search algorithms for calibration of conceptual rainfall-runoff models. Water Resour. Res. 24:691-700.

Integrated Software Systems Corporation. 1985. DISSPLA user's manual, verison 10.0, San Diego, California 92121.

International Mathematical Subroutine Library. 1982. Edition 9. Houston, Texas.

Jury, W. A., G. Sposito, and R. E. White. 1986. A transfer function model of solute transport through soil. 1. Fundamental concepts. Water Resour. Res. 22:243-247.

Mackay, D. M., D. L. Freyberg, P. V. Roberts, and J. A. Cherry. 1986. A natural gradient experiment on solute transport in a sand aquifer. 1. Approach and overview of plume movement. Water Resour. Res. 22:2017-2029.

Marquardt, D. W. 1963. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. 11:431-441.

Meeter, D. A. 1964. Non-linear least-squares (Gaushaus). Univ. of Wisconsin Computing Center. Program revised 1966.

Parker, J. C., and M. Th. van Genuchten. 1984. Determining transport parameters from laboratory and field tracer experiments. Bulletin 84-3, Virginia Agricultural Experiment Station, Blacksburg, Virginia.

Roberts, P. V., M. N. Goltz, and D. M. Mackay. 1986. A natural gradient experiment on solute transport in a sand aquifer. 3. Retardation estimates and mass balances for organic solutes. _Water Resour. Res._ 22(13):2047-2058.

Roberts, P. V., and D. M. Mackay. 1986. A natural gradient experiment on solute transport in a sand aquifer. Technical Report No. 292, Civil Engineering Department, Stanford University, California 94305.

Sneddon, I. N. 1972. The use of integral transforms. McGraw-Hill Book Company, New York.

Spiegel, M. R. 1965. Theory and problems of Laplace transforms. Schaum's Outline Series, McGraw-Hill Book Company, New York.

Sposito, G., R. E. White, P. R. Darrah, and W. A. Jury. 1986. A transfer function model of solute transport through soil. 3. The convection-dispersion equation. _Water Resour. Res._ 22:255-262.

Sudicky, E. A. 1986. A natural gradient experiment on solute transport in a sand aquifer: Spatial variability of hydraulic conductivity and its role in the dispersion process. _Water Resour. Res._ 22:2069-2082.

Valocchi, A. J. 1986. Effect of radial flow on deviations from local equilibrium during sorbing solute transport through homogeneous soils. _Water Resour. Res._ 22(12):1693-1701.

van Genuchten, M. Th. 1980. Determining transport parameters from solute displacement experiments. Res. Rep. No. 118, U.S. Salinity Laboratory, Riverside, California.

van Genuchten, M. Th. 1981. Non-equilibrium transport parameters from miscible displacement experiments. Res. Rep. No. 119, U.S. Salinity Laboratory, Riverside, California.

Walker, G. R. 1987. Solution to a class of coupled linear partial differential equations. IMA J. Appl. Math. 38:35-48.

APPENDIX 1a. FORTRAN SOURCE CODE FOR TFMFIT (SINGLE PRECISION CRAY VERSION)

```
c
c     Copyright, the Regents of the University of California.
c     Nonlinear least squares fitting for the TFM model.  The fitting portion
c     of the code is derived from the CXTFIT program in: Parker, J. C., and
c     M. Th. van Genuchten.  1984.  Determining transport parameters from
c     laboratory and field tracer experiments.  Bulletin 84-3, Virginia
c     Agricultural Experiment Station, Blacksburg.  The program is intended
c     to run on CRAY machines using CRAY FORTRAN.
c
      programtfmfit
      dimensiony(300),f(300),r(300),delz(300,8),th(16),b(16),e(8),p(8)
     z         ,phi(8),q(8),lsort(300),tb(16),a(8,8),d(8,8),x(300),
     z         bmax(8),bmin(8),bi(16),sumtot(300),ssqtot(300),
     z         sumbt(300),sum1t(8),sum2t(8),sum3t(8),secoef(8),
     z         tvalue(8),index(8),tpcoe(8),tmcoe(8),temp1(300),
     z         yyi(300),thh(8),tsec(8)
      common/modat1/x,index,nob,nvar,nu1,nu2,iopt
      charactertitle*80,bi*4,tit11*4,tit12*4,fname*8
      parameter(stopcr=1.e-4,p3low=1.e-6,blow=1.e-6)
      datanvar,nu1,nu2/8,9,16/
c
c     The following two subroutine calls are for CRAY machines running
c     CTSS (Cray Time Sharing System).  This call is to allow bidirectional
c     memory on the CRAY.
c
      callq8ebm
c
c     Send IMSL error and warning messages to the file named imslerr; the
c     terminal is unit 6.
c
      callugetio(3,nin,5)
      calllink('unit5=(imslerr,create,text),unit6=tty//')
      goto2
1     write(6,1000)
2     write(6,2000)
      read(6,2001)fname
      if(fname.eq.' ')fname='tfmdat'
      open(2,file=fname,status='old',err=1)
      goto4
3     write(6,1000)
4     write(6,2002)
      read(6,2001)fname
      if(fname.eq.' ')fname='tfmout'
      open(1,file=fname,status='new',err=3)
c
c     There are ncase fittings to do.
c
      do150ncase=1,20
c
c        Read input parameters.
c
         read(2,1006,end=999)mit,maxtry,iplot,iopt
c
c        nvar = number of coefficients (8, see parameter statement above).
c        nob  = number of observations (300 maximum, see experimental data
c               entry at label 10 below: can be increased if necessary).
c        mit  = maximum number of iterations (mit=0 solves the direct problem
c               only).
```

```
c      maxtry = maximum number of trial solutions at iteration before
c             quitting.  It is suggested that maxtry be in the range 10 to
c             50; smaller values reduce the run time but you may miss a
c             convergent solution.
c      iplot = code for plot file output:
c             0 = no plot file; 1 = plot file is output.
c      iopt = solution procedure option;
c             0 - use numerical Laplace transform inversion always;
c             1 - use numerical Laplace transform inversion until numerical
c                 problems occur then use analytical solution;
c             2 - use analytical solution always.
c
       if(mit.ne.0)then
c
c      For mit = 0, do not produce a plot file.
c
          if(iplot.eq.1.and.iplot1.ne.1)then
             goto6
5            write(6,1000)
6            write(6,2003)
             read(6,2001)fname
             if(fname.eq.' ')fname='tfmplt'
             open(3,file=fname,status='new',err=5)
             iplot1=1
          endif
       else
          iplot=0
       endif
       if(maxtry.eq.0)maxtry=5
       if(mit.ne.0)write(1,1035)
       write(1,1002)
       do7i=1,2
          read(2,1001)title
7      write(1,1001)title
c
c      Input the model parameters.
c
       do8n=1,2
          n14=n*14
          n7=n*7
          nvar7=nvar+n7
c
c         Read the coefficient names.
c
          n1=n14-13
          n2=min0(n14,nu2)
          read(2,1004)(bi(i),i=n1,n2)
c
c         Read the initial paraneter estimates.
c
          n1=nvar7-6
          n2=min0(nvar7,nu2)
          read(2,1005)(b(i),i=n1,n2)
c
c         Read the parameter indices.
c
          n1=n7-6
          n2=min0(n7,nvar)
          read(2,1006)(index(i),i=n1,n2)
c
```

```
c             Read in the constraints on parameter values.  bmin and bmax are
c             the arrays containing the minimum and maximum constraints.
c             bmin(i) = bmax(i) will omit constraint on parameter i.
c
              read(2,1005) (bmin(i),i=n1,n2)
8          read(2,1005) (bmax(i),i=n1,n2)
c
c          Check that the X3 index is set to 0 (i.e., X3 cannot be fitted).
c
           if(index(2).ne.0)then
              write(6,2015)
              index(2)=0
           endif
c
c          Check that the constraints on initial parameter values are allowable.
c
           if(b(9).le.0.)then
              b(9)=p3low
              write(6,2006)
           endif
           if(b(10).ne.0..and.b(10).ne.1.)then
              write(6,2007)
              stop
           endif
           if(b(10).eq.0.)then
              if(b(11).le.b(10))then
                 write(6,2008)
                 stop
              endif
           else
              if(b(11).le.b(10))then
                 write(6,2009)
                 stop
              endif
           endif
           if(b(12).le.0..or.b(12).gt.1.)then
              write(6,2010)
              stop
           endif
           if(b(13).lt.0.)then
              b(13)=0.
              write(6,2011)
           endif
           if(b(14).lt.0.)then
              b(14)=0.
              write(6,2012)
           endif
           if(b(15).lt.0.)then
              b(15)=0.
              write(6,2013)
           endif
           if(b(16).lt.0.)then
              b(16)=0.
              write(6,2014)
           endif
c
c          Check that the parameter ranges are correct.
c
           if(bmin(1).ne.bmax(1).and.index(1).eq.1)then
              if(bmin(1).le.0.)then
```

```
            write(6,2016)
            bmin(1)=p3low
         endif
      endif
      if(bmin(3).ne.bmax(3).and.index(3).eq.1)then
         if(b(10).eq.0.)then
            if(bmin(3).le.b(10))then
               write(6,2017)
               stop
            endif
         endif
         if(b(10).eq.1.)then
            if(bmin(3).le.b(10))then
               write(6,2018)
               stop
            endif
         endif
      endif
      if(bmin(4).ne.bmax(4).and.index(4).eq.1)then
         if(bmin(4).le.0.)then
            write(6,2019)
            bmin(4)=blow
         endif
         if(bmax(4).gt.1.)then
            write(6,2020)
            bmax(4)=1.
         endif
      endif
      if(bmin(5).ne.bmax(5).and.index(5).eq.1)then
         if(bmin(5).lt.0.)then
            write(6,2021)
            bmin(5)=0.
         endif
      endif
      if(bmin(6).ne.bmax(6).and.index(6).eq.1)then
         if(bmin(6).lt.0.)then
            write(6,2022)
            bmin(6)=0.
         endif
      endif
      if(bmin(7).ne.bmax(7).and.index(7).eq.1)then
         if(bmin(7).lt.0.)then
            write(6,2023)
            bmin(7)=0.
         endif
      endif
      if(bmin(8).ne.bmax(8).and.index(8).eq.1)then
         if(bmin(5).lt.0.)then
            write(6,2024)
            bmin(5)=0.
         endif
      endif
      write(1,1007)
      do9i=1,nvar
         j=2*i-1
9     write(1,1008)bi(j),bi(j+1),b(i+nvar)
c
c     Read in the experimental data.
c
      goto11
```

```
10          write(6,1000)
11          write(6,2004)
            read(6,2001)fname
            if(fname.eq.' ')fname='exptdata'
            open(4,file=fname,status='old',err=10)
            do12i=1,300
12          read(4,1005,end=13)y(i),x(i)
13          nob=i-1
            write(6,2005)nob
            nob1=i
            close(4)
c
c           Check that t > 0 for each datum.
c
            do14i=1,nob
               if(x(i).le.0.)then
                  write(6,2026)
                  stop
               endif
14          continue
c
c           Check that the data outnumber the number of variables to be fitted.
c
            ncnvar=0
            do15i=1,nvar
               if(index(i).eq.1)ncnvar=ncnvar+1
15          continue
            if(ncnvar.ge.nob)then
               write(6,2025)
               stop
            endif
c
c           Rearrange the variable arrays.
c
18          np=0
            do19i=nu1,nu2
19          tb(i)=b(i)
            do20i=nu1,nu2
               invar=i-nvar
               if(index(invar).eq.0)goto20
               np=np+1
               k=2*np-1
               j=2*invar-1
               tit11=bi(j)
               tit12=bi(j+1)
               bi(k)=tit11
               bi(k+1)=tit12
               b(np)=b(i)
               tb(np)=b(i)
               th(np)=b(np)
               bmin(np)=bmin(invar)
               bmax(np)=bmax(invar)
20          th(i)=b(i)
c
c           Evaluate the initial residual vector.
c
            ga=.02
            nit=0
            np2=2*np
            callmodel(th,f)
```

```
        if(mit.eq.0)goto140
        ssq=.0
        do31i=1,nob
           r(i)=y(i)-f(i)
31      ssqtot(i)=r(i)*r(i)
        do32i=1,nob
32      ssq=ssq+ssqtot(i)
        write(1,1011)(bi(j),bi(j+1),j=1,np2,2)
        write(1,1012)nit,ssq,(b(i),i=1,np)
c
c       Iterate to find the optimal model fit.
c
33      nit=nit+1
        ntrial=0
        ga=.1*ga
        do34j=1,np
34      temp1(j)=th(j)
        if(nit.eq.1)then
           do35j=1,np
35         e(j)=1.
        endif
        do38j=1,np
           th(j)=1.01*th(j)
           if(th(j).eq.0.)th(j)=.01
           q(j)=0.
           callmodel(th,delz(1,j))
           do36i=1,nob
              delz(i,j)=delz(i,j)-f(i)
36         q(j)=q(j)+delz(i,j)*r(i)
              q(j)=100.*q(j)/th(j)
38      th(j)=temp1(j)
c
c       q=xt*r  (steepest descent).
c
        do44i=1,np
           do42j=1,i
              sum=0.
              do40k=1,nob
40            sumtot(k)=delz(k,i)*delz(k,j)
              do41k=1,nob
41            sum=sum+sumtot(k)
              if(sum.ne.0.)then
                 d(i,j)=10000.*sum/th(i)/th(j)
              else
                 d(i,j)=0.
              endif
42         d(j,i)=d(i,j)
           e(i)=sqrt(d(i,i))
44      if(e(i).eq.0.)e(i)=1.e-30
50      do52i=1,np
           do52j=1,np
52      a(i,j)=d(i,j)/e(i)/e(j)
c
c       a is the scaled moment matrix.
c
        do54i=1,np
           p(i)=q(i)/e(i)
           phi(i)=p(i)
54      a(i,i)=a(i,i)+ga
        callmatinv(a,np,p)
```

```
c
c         p/e is the correction vector.
c
          step=1.
55        do56i=1,np
56         tb(i)=p(i)*step/e(i)+th(i)
          do58i=1,np
             if(bmin(i).eq.bmax(i))goto58
             if(tb(i).gt.bmax(i))tb(i)=bmax(i)
             if(tb(i).lt.bmin(i))tb(i)=bmin(i)
             p(i)=(tb(i)-th(i))*e(i)/step
58        continue
          do59i=1,np
59        thh(i)=th(i)*tb(i)
          do62i=1,np
             if(thh(i))66,66,62
62        continue
          sumb=0.
          callmodel(tb,f)
          do64i=1,nob
             r(i)=y(i)-f(i)
64        sumbt(i)=r(i)*r(i)
          do65i=1,nob
65        sumb=sumbt(i)+sumb
66        sum1=0.
          sum2=0.
          sum3=0.
          do68i=1,np
             sum1t(i)=p(i)*phi(i)
             sum2t(i)=p(i)*p(i)
68        sum3t(i)=phi(i)*phi(i)
          do69i=1,np
             sum1=sum1+sum1t(i)
             sum2=sum2+sum2t(i)
69        sum3=sum3+sum3t(i)
          arg=0.
          suu=sum2*sum3
          if(suu.gt.0.)arg=sum1/sqrt(suu)
          arg1=0.
          if(np.gt.1)arg1=sqrt(1.-arg*arg)
          if(arg.eq.0..and.arg1.eq.0.)arg=1.
          angle=57.295779*atan2(arg1,arg)
          do72i=1,np
             if(thh(i))74,74,72
72        continue
          ntrial=ntrial+1
          if(ntrial.gt.maxtry)goto95
          if(sumb/ssq-1.)80,80,74
74        if(angle-30.)76,76,78
76        step=.5*step
          goto55
78        ga=10.*ga
          goto50
c
c         Print the coefficients after each iteration.
c
80        continue
          do82i=1,np
82        th(i)=tb(i)
          write(1,1012)nit,sumb,(th(i),i=1,np)
```

```
           do86i=1,np
              if(abs(p(i)*step/e(i))/(1.e-20+abs(th(i)))-
     z           stopcr)86,86,94
86         continue
           goto96
94         ssq=sumb
           if(nit.lt.mit)goto33
           if(nit.eq.mit)write(1,1034)mit
           goto96
95         write(1,1038)maxtry
c
c          End of the iteration loop.
c
96         callmatinv(d,np,p)
c
c          Write the correlation matrix.
c
           do98i=1,np
98         e(i)=sqrt(abs(d(i,i)))
           do99i=1,np
99         if(e(i).eq.0.)e(i)=1.e-30
           if(np.eq.1)goto104
           write(1,1013)(i,i=1,np)
           do102i=1,np
              do100j=1,i
100           a(j,i)=d(j,i)/e(i)/e(j)
102        write(1,1014)i,(a(j,i),j=1,i)
c
c          Calculate the regression coefficient.
c
104        sumc=0.
           sumc2=0.
           do105i=1,nob
105        yyi(i)=y(i)*y(i)
           do106i=1,nob
              sumc=sumc+y(i)
106        sumc2=sumc2+yyi(i)
           vinny=nob
           rsq=1.-sumb/(sumc2-sumc*sumc/vinny)
           write(1,1041)rsq
c
c          Calculate the 95% confidence interval.
c
           qx=nob-np
           z=1./qx
           sdev=sqrt(z*sumb)
           tvar=1.96+z*(2.3779+z*(2.7135+z*(3.187936+2.466666*z*z)))
           if(np.eq.1)write(1,1042)
           if(np.gt.1)write(1,1015)
           do107i=1,np
              secoef(i)=e(i)*sdev
107        tvalue(i)=th(i)/secoef(i)
           do108i=1,np
108        tsec(i)=tvar*secoef(i)
           do109i=1,np
              tmcoe(i)=th(i)-tsec(i)
109        tpcoe(i)=th(i)+tsec(i)
           if(np.eq.1)then
              do110i=1,np
                 j=2*i-1
```

```
110                  write(1,1043)i,bi(j),bi(j+1),th(i),secoef(i),
     z                           tmcoe(i),tpcoe(i)
           else
              do111i=1,np
                 j=2*i-1
111              write(1,1016)i,bi(j),bi(j+1),th(i),secoef(i),
     z                           tvalue(i),tmcoe(i),tpcoe(i)
           endif
c
c          Prepare the final output.
c
           lsort(1)=1
           do112j=2,nob
112        temp1(j)=abs(r(j))
           do117j=2,nob
              k=j-1
              do113l=1,k
                 if(temp1(j)-abs(r(lsort(l))))114,114,113
113           continue
              lsort(j)=j
              goto117
114           kk=j
115           kk=kk-1
              lsort(kk+1)=lsort(kk)
              if(kk-l)116,116,115
116           lsort(l)=j
117        continue
           write(1,1017)
           do118i=1,nob
118        write(1,1018)i,x(i),y(i),f(i),r(i)
           if(iplot.ne.0)then
              do119i=1,nob
119           write(3,1047)y(i),x(i),i
              do120i=1,nob
120           write(3,1047)f(i),x(i),i
           endif
           write(1,1019)
           do121i=1,nob
              j=lsort(nob1-i)
121        write(1,1018)j,x(j),y(j),f(j),r(j)
           goto150
140        write(1,1030)(j,j=1,nx)
           do145i=1,nob
145        write(1,1033)i,x(i),f(i)
c
c       End of this job.
c
150     continue
999     stop
1000    format('Choose another file name.')
1001    format(a80)
1002    format(/)
1004    format(7(2x,2a4))
1005    format(7f10.5)
1006    format(7i10)
1007    format(/x,'Initial values of coefficients'/x,30(1h=)/3x,
     z             'Name',4x,'Initial value')
1008    format(x,2a4,4(1h.),1pe12.4)
1011    format(/x,'Iteration',6x,'SSQ',4x,8(5x,2a4))
1012    format(x,i5,3x,1pe13.4,2x,8(1pe13.4))
```

```
1013    format(/3x,'Correlation Matrix',/3x,18(1h=)/4x,10(4x,i2,5x))
1014    format(x,i3,10(2x,f7.4,2x))
1015    format(x,/,'Nonlinear Least Squares Analysis, Final Results.'
     z,/,48(1h=)/57x,'95% Confidence Limits'/x,'Variable',3x,'Name'
     z,7x,'Value',7x,'S.E.Coeff.',2x,'T-Value',4x,'Lower',9x,'Upper')
1016    format(4x,i2,3x,2a4,1pe13.4,2x,1pe13.4,e10.3,1pe12.4,
     z        x,1pe12.4)
1017    format(/,29x,'Ordered by computer input',/2x,'No.',4x,'  T  ',
     z 8x,'Conc Obs',6x,'Conc Fit',6x,'Residual')
1018    format(x,i3,x,1pe12.3,3(2x,1pe12.3))
1019    format(/,29x,'Ordered by residual',/2x,'No.',4x,'  T  '
     z        ,8x,'Conc Obs',6x,'Conc Fit',6x,'Residual')
1030    format(/x,'Results for initial coefficient values',//x,'No.',
     z   8x,'T  ',7x,'    Conc  ')
1033    format(x,i2,2(3x,1pe12.4))
1034    format(/x,'Convergence criteria not met in',i3,' iterations.')
1035    format(5x,/,'Nonlinear Least-Squares Analysis.')
1038    format(/x,'No further decrease in SSQ obtained after ',i2,
     z          ' trials.')
1041    format(/x,'Rsquare for regression =',e20.10,'.',/)
1042    format(x,'Nonlinear Least Squares Analysis, Final Results.'
     z/x,49(1h=)/57x,'95% Confidence Limits'/x,'Variable',3x,'Name'
     z,8x,'Value',8x,'S.E.Coeff.',6x,'Lower',10x,'Upper')
1043    format(x,i2,6x,2a4,x,1pe13.4,3x,1pe13.4,x,1pe13.4,2x,
     z          1pe13.4)
1047    format(2(1pe15.4),i3)
2000    format('Enter the name of the input data file',
     z          ' (default = tfmdat).')
2001    format(a8)
2002    format('Enter the name of the primary output file',
     z          ' (default = tfmout).')
2003    format('Enter the name of the plot output file',
     z          ' (default = tfmplt).')
2004    format('Enter the name of the file containing the ',
     z          'experimental data (default = exptdata).')
2005    format('There were ',i3,' experimental data read.')
2006    format('P3 must not be less than zero, it has been reset',
     z          ' to zero.')
2007    format('Set X3 as zero or one in the data file and rerun',
     z          ' the program.')
2008    format('Y must be greater than X3 (zero).  Reset Y in the',
     z          /,' data file and rerun the program.')
2009    format('Y must be greater than X3 (one).  Reset Y in the',
     z          /,' data file and rerun the program.')
2010    format('B must not be less than or equal to zero or greater',
     z' than one.'/,'Reset B in the data file and rerun the program.')
2011    format('R must not be less than zero, it has been reset ',
     z          'to zero.')
2012    format('W must not be less than zero, it has been reset ',
     z          'to zero.')
2013    format('D must not be less than zero, it has been reset ',
     z          'to zero.')
2014    format('C must not be less than zero, it has been reset ',
     z          'to zero.')
2015    format('X3 cannot be fitted.  INDEX(2) is reset to 0.')
2016    format('The minimum value of P3 cannot be less than or equal',
     z          ' to 0.  It ',/,'has been reset to P3low.')
2017    format('The minimum value of Yo must be greater than 0 (x3).'
     z          ,/,'Reset the lower limit on Yo and rerun.')
2018    format('The minimum value of Yo must be greater than 1 (x3).'
     z          ,/,'Reset the lower limit on Yo and rerun.')
2019      format('The minimum value of B cannot be less than or equal to',
     z          ' 0.',/,'It has been reset to Blow.')
2020      format('The maximum value of B cannot be greater than 1.',/,
     z          'It has been reset to one.')
2021      format('The minimum value of R cannot be less than 0.',/,
     z          'It has been reset to zero.')
2022      format('The minimum value of W cannot be less than 0.',/,
     z          'It has been reset to zero.')
2023      format('The minimum value of D cannot be less than 0.',/,
     z          'It has been reset to zero.')
2024      format('The minimum value of C cannot be less than 0.',/,
     z          'It has been reset to zero.')
2025      format('The experimental data must outnumber the number of'
     z          ,/,'parameters being fitted.')
2026      format('At least one time in the data file is less than or'
     z          ,/,'equal to zero.  Check and rerun.')
          end
          subroutinematinv(a,np,b)
c
c         Purpose: perform matrix inversion for parameter estimation.
c
          dimensionpp(8),a(8,8),b(16),index(8,2)
          do2j=1,8
2         index(j,1)=0
          i=0
4         amax=-1.
          do10j=1,np
             if(index(j,1))10,6,10
6            do9k=1,np
                if(index(k,1))9,8,9
8               p=abs(a(j,k))
                if(p.le.amax)goto9
                ir=j
                ic=k
                amax=p
9            continue
10        continue
          if(amax)30,30,14
14        index(ic,1)=ir
          if(ir.eq.ic)goto18
          do15l=1,np
15        pp(l)=a(ir,l)
          do16l=1,np
             a(ir,l)=a(ic,l)
16        a(ic,l)=pp(l)
          p=b(ir)
          b(ir)=b(ic)
          b(ic)=p
          i=i+1
          index(i,2)=ic
18        p=1./a(ic,ic)
          a(ic,ic)=1.
          do20l=1,np
20        a(ic,l)=a(ic,l)*p
          b(ic)=b(ic)*p
          do24k=1,np
             if(k.eq.ic)goto24
             p=a(k,ic)
             a(k,ic)=0.
```

```
            do22l=1,np
22          a(k,l)=a(k,l)-a(ic,l)*p
            b(k)=b(k)-b(ic)*p
24       continue
         goto4
26       ic=index(i,2)
         ir=index(ic,1)
         do27k=1,np
27       pp(k)=a(k,ir)
         do28k=1,np
            a(k,ir)=a(k,ic)
28       a(k,ic)=pp(k)
         i=i-1
30       continue
         if(i)26,32,26
32       return
         end
         subroutinemodel(bn,ycalc)
         dimensionycalc(300),bn(16),c(8),index(8),x(300)
         parameter(p3low=1.e-6,blow=1.e-6)
         common/modat1/x,index,nob,nvar,nu1,nu2,iopt
c
c        Update the coefficient array.
c
         k=0
         do2i=nu1,nu2
            if(index(i-nvar).eq.0)goto2
            k=k+1
            bn(i)=bn(k)
2        continue
         do5i=1,nvar
5        c(i)=bn(nvar+i)
c
c        Check the parameter ranges.
c
         if(index(1).eq.1)then
            if(c(1).le.0.)c(1)=p3low
         endif
         if(index(3).eq.1)then
            if(c(3).le.c(2))c(3)=c(2)+1.e-10
         endif
         if(index(4).eq.1)then
            if(c(4).gt.1.)c(4)=1.
            if(c(4).le.0.)c(4)=blow
         endif
         if(index(5).eq.1)then
            if(c(5).lt.0.)c(5)=0.
         endif
         if(index(6).eq.1)then
            if(c(6).lt.0.)c(6)=0.
         endif
         if(index(7).eq.1)then
            if(c(7).lt.0.)c(7)=0.
         endif
         if(index(8).eq.1)then
            if(c(8).lt.0.)c(8)=0.
         endif
c
c        Call the TFM model function.
c
```

```
      callzres(ycalc,x,c,nob)
      return
      end
      subroutinezres(zfinv,zt,zc,n)
      implicitcomplex(a-h,o-y)
      externalf
      dimensionzt(n),zfinv(n),zc(8),zx(300),index(8)
      common/modat1/zx,index,nob,nvar,nu1,nu2,iopt/rw/p3,x3,y,b,
     z     r,w,d,c
      datanotice/1/
c
c     The following parameters are for the IMSL Laplace inversion routine.
c     The inversion is accurate to nsig significant figures.  The maximum
c     number of iterations for each inversion is kmax.  See the IMSL manual
c     for details.
c
      parameter(nsig=4,kmax=499)
      if(iopt.eq.2)then
         callrealtime(zc,zt,zfinv,n)
         return
      endif
      icount=0
      p3=cmplx(zc(1),0.)
      x3=cmplx(zc(2),0.)
      y=cmplx(zc(3),0.)
      b=cmplx(zc(4),0.)
      r=cmplx(zc(5),0.)
      w=cmplx(zc(6),0.)
      d=cmplx(zc(7),0.)
      c=cmplx(zc(8),0.)
c
c     The function root calculates a number greater than or equal to the
c     real part of the maximum of the singularities of the function to be
c     inverted.  The argument is a dummy parameter.
c
      alpha=root(op)
      zzz=f1(alpha)
1     if(zzz.lt.0.)then
         zalpha=alpha
         alpha=alpha-cmplx(1.e-6*zalpha,0.)
         zzz=f1(alpha)
      endif
      icount=icount+1
      if(icount.le.10.and.zzz.lt.0.)goto1
      if(icount.gt.10)write(6,4)
      zalpha=alpha
c
c     The estimate has been found; proceed to the inversion using the IMSL
c     numerical Laplace transform inverter.
c
      callflinv(f,n,zt,zalpha,nsig,kmax,zfinv,ier)
      do2i=1,n
2     if(zfinv(i).lt.0.)zfinv(i)=0.
      if(ier.eq.0)return
c
c     Stop printing error messages for more than 10 errors.
c
      ierc=ierc+1
      if(ierc.eq.11)write(6,9)
      if(ierc.gt.10)goto3
```

```
        write(6,5)ier,(zc(i),i=1,8)
        if(ier.eq.129)then
           write(6,6)
           goto3
        endif
        if(ier.eq.130)then
           write(6,7)
           goto3
        endif
        write(6,8)
3       if(iopt.eq.1)then
           callrealtime(zc,zt,zfinv,n)
           if(notice.eq.1)then
              notice=0
              write(6,11)
           endif
        endif
        return
4       format('Problem in the required estimate for the maximum of the'
     z         ,/,'singularities in the Laplace transform of the TFM.',/,
     z         'Check the result subroutine (zres).')
5       format('Error in IMSL numerical Laplace transform inversion '
     z         ,'(ier = ',i3,').',/,'The parameter values used are: ',
     z         'p3 = ',e16.8,', x3 = ',e16.8,',',/,'y = ',e16.8,', b = ',
     z         e16.8,', r = ',e16.8,',',/,'w = ',e16.8,', d = ',e16.8,
     z         ' and c = ',e16.8,'.',/,'This error is due to the ',
     z         'following:')
6       format('The inversion algorithm was not able to achieve the'
     z         ,/,'accuracy requested within the kmax function '
     z         ,/,'evaluations for some times.')
7       format('An overflow would have occurred for a particular time.')
8       format('The inversion algorithm increased the integration ',
     z         'limit',/,'eight times without obtaining the requested ',
     z         'accuracy.')
9       format('Further error descriptions will not be printed.')
11      format('Analytical solution being utilized.')
        end
        complexfunctionroot(op)
c
c       The estimate of the maximum of the real part of the singularities of
c       the TFM is obtained here.
c
        implicitcomplex(a-h,o-y)
        parameter(yzero=(0.,0.),four=(4.,0.),two=(2.,0.),one=(1.,0.))
        common/rw/p3,x3,y,b,r,w,d,c
        if(b.eq.yzero)then
           w1=four*w
           root=-(p3*(w+d)+w1*d)/(p3+w1)/r
           return
        endif
        a=p3*p3+four*b*p3*c
        e=four*b*r*p3
        if(b.eq.one)then
           root=-a/e
           return
        endif
        if(w.eq.yzero)then
           bb=four*b
           root=-(p3+bb*c)/bb/r
           return
```

```
      endif
      f=four*r*p3*w
      g=four*p3*w*d
      h=d+w/(one-b)
      esr=e*r
      bi=(h*e+f+a*r)/esr
      bj=(a*h+g)/esr
      root=csqrt(bi*bi-four*bj)
      r2=-(root+bi)/two
      root=(root-bi)/two
      zr1=root
      zr2=r2
      if(zr2.gt.zr1)root=r2
      return
      end
      complexfunctionf(s)
c
c     The Laplace-transformed TFM is calculated here.
c
      implicitcomplex(a-h,o-y)
      common/rw/p3,x3,y,b,r,w,d,c
      parameter(yzero=(0.,0.),one=(1.,0.),two=(2.,0.))
      if(s.eq.yzero)then
         f=yzero
         return
      endif
      f11=f1(s)
      x32=x3/two
      f=p3*(cexp(x32*(p3-f11))-cexp(x32*p3-y/two*f11))/f11/(one
     z  -cexp(p3/two*(x3-y)))
      return
      end
      complexfunctionf1(s)
c
c     Function to calculate the f(s) portion of the TFM.
c
      implicitcomplex(a-h,o-y)
      common/rw/p3,x3,y,b,r,w,d,c
      parameter(yzero=(0.,0.),one=(1.,0.),four=(4.,0.))
      if(b.eq.one)then
         f1=csqrt(p3*(p3+four*(r*s+c)))
         return
      endif
      if(b.eq.yzero)then
         r1=r*s+d
         f1=csqrt(p3*p3+four*p3*w*r1/(r1+w))
         return
      endif
      b1=one-b
      b22=four*p3
      b2=b22*b
      r1=r*s
      r2=b1*r1+w+d*b1
      f1=p3*p3+b2*r1*(one+b1*w/b/r2)+b2*c+b22*b1*w*d/r2
      zff1=f1
      if(zff1.gt.0.)f1=csqrt(f1)
      return
      end
c
c     CRAY function to evaluate the TFM analytically.
```

```
c
      subroutinerealtime(cc,tt,res,n)
      dimensioncc(n),tt(n),res(n)
      externalfgrand
      common/param/d,p,b,r,w,c,t/tit1/os,p4p,rb,w1,tv,st,x,c1,b1,y
      parameter(zero=0.,pi=3.14159265358979323846264 3,rerr=1.e-5)
      x=cc(2)
      y=cc(3)
      p=cc(1)
      b=cc(4)
      r=cc(5)
      c=cc(8)
      d=cc(7)
      w=cc(6)
      pl=p/2.
      p4p=p/4.
      if(b.eq.1.)then
         pip=p4p/pi/r
         x2=-x*x
         y2=-y*y
         pl1=pl*x
         cz=c+p4p
         oz=1.-exp((x-y)*pl)
         do3i=1,n
            tr=tt(i)/r
            pq=p4p/tr
3        res(i)=sqrt(pip/tt(i))*exp(pl1-tr*cz)/oz*(exp(x2*pq)-
     z             exp(y2*pq))
      else
         os=pl*(x-y)
         c1=c+w/b
         rb=r*b
         pt1=sqrt(p4p/pi/rb)/(1.-exp(os))
         do4i=1,n
            pt=pt1*exp(-tt(i)/r*c1)
            ty=tt(i)/rb
            tr=-p4p/ty
            tv=exp(tr*(x-ty)**2)-exp(tr*(y-ty)**2+os)
            st=sqrt(tt(i))
            res(i)=tv/st
            b1=1.-b
            w1=w/r/sqrt(b*b1)
            if(w.eq.zero)then
               res(i)=res(i)*pt
            else
               t=tt(i)
               res(i)=pt*(res(i)+dcadre(fgrand,zero,tt(i),zero,rerr,
     z                 err,ier)*w1)
               if(icount.gt.10)goto4
               if(ier.eq.131.or.ier.eq.132.or.ier.eq.133)then
                  icount=icount+1
                  write(6,6)
               endif
            endif
4        continue
      endif
      return
6     format('Problem in the IMSL subroutine dcadre.  Treat results',
     z      ' with caution.')
      end
```

```
c
c     Calculating the integrand in the analytical real time solution.
c
      functionfgrand(tau)
      realmmbsi1
      common/param/d,p,b,r,w,c,t/tit1/os,p4p,rb,w1,tv,st,x,c1,b1,y
      parameter(zero=0.)
      if(t.eq.tau)then
         fgrand=zero
         return
      endif
      if(tau.eq.zero)then
         fgrand=w1*st*tv
         return
      endif
      o=t-tau
      ff=-p4p*rb/o
      dd=o/rb
      fgrand=mmbsi1(1,2.*w1*sqrt(tau*o),ier)*exp(tau/r*(c1-d-w/
     z          b1))*(exp(ff*(x-dd)**2)-exp(ff*(y-dd)**2+os))/sqrt(tau)
      if(ic.gt.10)goto1
      if(ier.eq.129.or.ier.eq.130)then
         ic=ic+1
         write(6,2)
      endif
1     return
2     format('Problem in the IMSL subroutine mmbsi1.  Check'
     z          ,/,'the function fgrand.')
      end
c
c     Copyright, the Regents of the University of California.
c
```

APPENDIX 1b. FORTRAN SOURCE CODE FOR TFMFIT (DOUBLE PRECISION VAX VERSION)

```
c
c     Copyright, the Regents of the University of California.
c     Nonlinear least squares fitting for the TFM model.  The fitting portion
c     of the code is derived from the CXTFIT program in: Parker, J. C., and M.
c     Th. van Genuchten.  1984.  Determining transport parameters from
c     laboratory and field tracer experiments.  Bulletin 84-3, Virginia
c     Agricultural Experiment Station, Blacksburg.  The program is intended
c     to run on VAX machines using VAX FORTRAN.
c
      programtfmfit
      implicitdoubleprecision(a-h,o-z)
      dimensiony(300),f(300),r(300),delz(300,8),th(16),b(16),e(8)
     z         ,p(8),phi(8),q(8),lsort(300),tb(16),a(8,8),d(8,8),
     z          x(300),bmax(8),bmin(8),bi(16),sumtot(300),
     z          ssqtot(300),sumbt(300),sum1t(8),sum2t(8),sum3t(8),
     z          secoef(8),tvalue(8),index(8),tpcoe(8),tmcoe(8),
     z          temp1(300),yyi(300),thh(8),tsec(8)
      common/modat1/x,index,nob,nvar,nu1,nu2,iopt
      charactertitle*80,bi*4,tit11*4,tit12*4,fname*16
      parameter(stopcr=1.d-4,p3low=1.d-6,blow=1.d-6)
      datanvar,nu1,nu2/8,9,16/
c
c     Send IMSL error and warning messages to imslerr.dat; the terminal is
c     unit 6.
c
      callugetio(3,nin,5)
      open(5,file='imslerr.dat',status='new')
      write(6,2000)
      read(6,2001)fname
      if(fname.eq.' ')fname='tfm.dat'
      open(2,file=fname,status='old')
      goto2
1     write(6,1000)
2     write(6,2002)
      read(6,2001)fname
      if(fname.eq.' ')fname='tfm.out'
      open(1,file=fname,status='new',err=1)
c
c     There are ncase fittings to do.
c
      do150ncase=1,20
c
c        Read input parameters.
c
         read(2,1006,end=999)mit,maxtry,iplot,iopt
c
c        nvar = number of coefficients (8, see parameter statement above).
c        nob  = number of observations (300 maximum, see experimental data
c               entry below).
c        mit  = maximum number of iterations (mit=0 solves the direct problem
c               only).
c        maxtry = maximum number of trial solutions at iteration before
c                 quitting.  It is suggested that this parameter be in the
c                 range 10 to 50; smaller values reduce the run
c                 time but you may miss a convergent solution.
c         iplot = code for plot file output:
c                0 - no plot file;
c                1 - plot file is output.
```

```
c          iopt = solution procedure option;
c                 0 - use numerical Laplace transform inversion always;
c                 1 - use numerical Laplace transform inversion until numerical
c                     problems occur then use analytical solution;
c                 2 - use analytical solution always.
c
           if(mit.ne.0)then
c
c          For mit = 0 do not produce a plot file.
c
              if(iplot.eq.1.and.iplot1.ne.1)then
                 write(6,2003)
                 read(6,2001)fname
                 if(fname.eq.' ')fname='tfm.plt'
                 open(3,file=fname,status='new')
                 iplot1=1
              endif
           else
              iplot=0
           endif
           if(maxtry.eq.0)maxtry=5
           if(mit.ne.0)write(1,1035)
           do4i=1,2
              read(2,1001)title
4          write(1,1001)title
c
c          Input the model parameters.
c
           do5n=1,2
              n14=n*14
              n7=n*7
              nvar7=nvar+n7
c
c             Read the coefficient names.
c
              n1=n14-13
              n2=jmin0(n14,nu2)
              read(2,1004) (bi(i),i=n1,n2)
c
c             Read the initial estimates.
c
              n1=nvar7-6
              n2=jmin0(nvar7,nu2)
              read(2,1005) (b(i),i=n1,n2)
c
c             Read in the indices.
c
              n1=n7-6
              n2=jmin0(n7,nvar)
              read(2,1006) (index(i),i=n1,n2)
c
c             Read in the constraints on the parameter values.  bmin and bmax
c             are the arrays containing the minimum and maximum constraints.
c             bmin(i) = bmax(i) will omit constraint on parameter i.
c
              read(2,1005) (bmin(i),i=n1,n2)
5          read(2,1005) (bmax(i),i=n1,n2)
c
c          Check that the x3 index is set to 0 (i.e., x3 cannot be fitted).
c
```

```
          if(index(2).ne.0) then
             write(6,2015)
             index(2)=0
          endif
c
c         Check the constraints on initial parameter values.
c
          if(b(9).le.0.d0) then
             b(9)=p3low
             write(6,2006)
          endif
          if(b(10).ne.0.d0.and.b(10).ne.1.d0) then
             write(6,2007)
             stop
          endif
          if(b(10).eq.0.d0) then
             if(b(11).le.b(10)) then
                write(6,2008)
                stop
             endif
          else
             if(b(11).le.b(10)) then
                write(6,2009)
                stop
             endif
          endif
          if(b(12).le.0.d0.or.b(12).gt.1.d0) then
             write(6,2010)
             stop
          endif
          if(b(13).lt.0.d0) then
             b(13)=0.d0
             write(6,2011)
          endif
          if(b(14).lt.0.d0) then
             b(14)=0.d0
             write(6,2012)
          endif
          if(b(15).lt.0.d0) then
             b(15)=0.d0
             write(6,2013)
          endif
          if(b(16).lt.0.d0) then
             b(16)=0.d0
             write(6,2014)
          endif
c
c         Check that the parameter ranges are correct.
c
          if(bmin(1).ne.bmax(1).and.index(1).eq.1) then
             if(bmin(1).le.0.d0) then
                write(6,2016)
                bmin(1)=p3low
             endif
          endif
          if(bmin(3).ne.bmax(3).and.index(3).eq.1) then
             if(b(10).eq.0.d0) then
                if(bmin(3).le.b(10)) then
                   write(6,2017)
                   stop
```

```
                 endif
              endif
              if(b(10).eq.1.d0) then
                 if(bmin(3).le.b(10)) then
                    write(6,2018)
                    stop
                 endif
              endif
           endif
           if(bmin(4).ne.bmax(4).and.index(4).eq.1) then
              if(bmin(4).le.0.d0) then
                 write(6,2019)
                 bmin(4)=blow
              endif
              if(bmax(4).gt.1.d0) then
                 write(6,2020)
                 bmax(4)=1.d0
              endif
           endif
           if(bmin(5).ne.bmax(5).and.index(5).eq.1) then
              if(bmin(5).lt.0.d0) then
                 write(6,2021)
                 bmin(5)=0.d0
              endif
           endif
           if(bmin(6).ne.bmax(6).and.index(6).eq.1) then
              if(bmin(6).lt.0.d0) then
                 write(6,2022)
                 bmin(6)=0.d0
              endif
           endif
           if(bmin(7).ne.bmax(7).and.index(7).eq.1) then
              if(bmin(7).lt.0.d0) then
                 write(6,2023)
                 bmin(7)=0.d0
              endif
           endif
           if(bmin(8).ne.bmax(8).and.index(8).eq.1) then
              if(bmin(5).lt.0.d0) then
                 write(6,2024)
                 bmin(5)=0.d0
              endif
           endif
           write(1,1007)
           do7i=1,nvar
              j=2*i-1
7          write(1,1008)bi(j),bi(j+1),b(i+nvar)
c
c          Read in the experimental data.
c
           write(6,2004)
           read(6,2001)fname
           if(fname.eq.' ')fname='expt.dat'
           open(4,file=fname,status='old')
           do10i=1,300
10         read(4,1005,end=11)y(i),x(i)
11         nob=i-1
           write(6,2005)nob
           nob1=i
           close(4)
```

```
c
c     Check that t > 0 for each datum.
c
      do12i=1,nob
         if(x(i).le.0.d0)then
            write(6,2026)
            stop
         endif
12    continue
c
c     Check that the data outnumber the number of variables to be fitted.
c
      ncnvar=0
      do15i=1,nvar
         if(index(i).eq.1)ncnvar=ncnvar+1
15    continue
      if(ncnvar.ge.nob)then
         write(6,2025)
         stop
      endif
c
c     Rearrange the variable arrays.
c
18    np=0
      do19i=nu1,nu2
19    tb(i)=b(i)
      do20i=nu1,nu2
         invar=i-nvar
         if(index(invar).eq.0)goto20
         np=np+1
         k=2*np-1
         j=2*invar-1
         tit11=bi(j)
         tit12=bi(j+1)
         bi(k)=tit11
         bi(k+1)=tit12
         b(np)=b(i)
         tb(np)=b(i)
         th(np)=b(np)
         bmin(np)=bmin(invar)
         bmax(np)=bmax(invar)
20    th(i)=b(i)
c
c     Evaluate the initial residual vector.
c
      ga=.02d0
      nit=0
      np2=2*np
      callmodel(th,f)
      if(mit.eq.0)goto140
      ssq=.0d0
      do31i=1,nob
         r(i)=y(i)-f(i)
31    ssqtot(i)=r(i)*r(i)
      do32i=1,nob
32    ssq=ssq+ssqtot(i)
      write(1,1011)(bi(j),bi(j+1),j=1,np2,2)
      write(1,1012)nit,ssq,(b(i),i=1,np)
c
c     Begin iteration.
```

```
c
33       nit=nit+1
         ntrial=0
         ga=.1d0*ga
         do34j=1,np
34       temp1(j)=th(j)
         if(nit.eq.1) then
            do35j=1,np
35          e(j)=1.d0
         endif
         do38j=1,np
            th(j)=1.01d0*th(j)
            if(th(j).eq.0.d0)th(j)=.01d0
            q(j)=0.d0
            callmodel(th,delz(1,j))
            do36i=1,nob
               delz(i,j)=delz(i,j)-f(i)
36          q(j)=q(j)+delz(i,j)*r(i)
               q(j)=100.d0*q(j)/th(j)
38       th(j)=temp1(j)
c
c        q=xt*r  (steepest descent).
c
         do44i=1,np
            do42j=1,i
               sum=0.d0
               do40k=1,nob
40             sumtot(k)=delz(k,i)*delz(k,j)
               do41k=1,nob
41             sum=sum+sumtot(k)
               if(sum.ne.0.d0) then
                  d(i,j)=10000.d0*sum/th(i)/th(j)
               else
                  d(i,j)=0.d0
               endif
42          d(j,i)=d(i,j)
            e(i)=dsqrt(d(i,i))
44       if(e(i).eq.0.d0)e(i)=1.d-30
50       do52i=1,np
            do52j=1,np
52       a(i,j)=d(i,j)/e(i)/e(j)
c
c        a is the scaled moment matrix.
c
         do54i=1,np
            p(i)=q(i)/e(i)
            phi(i)=p(i)
54       a(i,i)=a(i,i)+ga
         callmatinv(a,np,p)
c
c        p/e is the correction vector.
c
         step=1.d0
55       do56i=1,np
56       tb(i)=p(i)*step/e(i)+th(i)
         do58i=1,np
            if(bmin(i).eq.bmax(i))goto58
            if(tb(i).gt.bmax(i))tb(i)=bmax(i)
            if(tb(i).lt.bmin(i))tb(i)=bmin(i)
            p(i)=(tb(i)-th(i))*e(i)/step
```

```
58          continue
            do59i=1,np
59          thh(i)=th(i)*tb(i)
            do62i=1,np
               if(thh(i))66,66,62
62          continue
            sumb=0.d0
            callmodel(tb,f)
            do64i=1,nob
               r(i)=y(i)-f(i)
64          sumbt(i)=r(i)*r(i)
            do65i=1,nob
65          sumb=sumbt(i)+sumb
66          sum1=0.d0
            sum2=0.d0
            sum3=0.d0
            do68i=1,np
               sum1t(i)=p(i)*phi(i)
               sum2t(i)=p(i)*p(i)
68          sum3t(i)=phi(i)*phi(i)
            do69i=1,np
               sum1=sum1+sum1t(i)
               sum2=sum2+sum2t(i)
69          sum3=sum3+sum3t(i)
            arg=0.d0
            suu=sum2*sum3
            if(suu.gt.0.d0)arg=sum1/dsqrt(suu)
            arg1=0.d0
            if(np.gt.1)arg1=dsqrt(1.d0-arg*arg)
            if(arg.eq.0.d0.and.arg1.eq.0.d0)arg=1.d0
            angle=57.295779d0*datan2(arg1,arg)
            do72i=1,np
               if(thh(i))74,74,72
72          continue
            ntrial=ntrial+1
            if(ntrial.gt.maxtry)goto95
            if(sumb/ssq-1.d0)80,80,74
74          if(angle-30.d0)76,76,78
76          step=.5*step
            goto55
78          ga=10.d0*ga
            goto50
c
c           Print out the coefficients after each iteration.
c
80          do82i=1,np
82          th(i)=tb(i)
            write(1,1012)nit,sumb,(th(i),i=1,np)
            do86i=1,np
               if(dabs(p(i)*step/e(i))/(1.d-20+dabs(th(i)))-
      z           stopcr)86,86,94
86          continue
            goto96
94          ssq=sumb
            if(nit.lt.mit)goto33
            if(nit.eq.mit)write(1,1034)mit
            goto96
95          write(1,1038)maxtry
c
c           This is the end of the iteration loop.
```

```
c
96         callmatinv(d,np,p)
c
c          Write out the correlation matrix.
c
           do98i=1,np
98         e(i)=dsqrt(dabs(d(i,i)))
           do99i=1,np
99         if(e(i).eq.0.d0)e(i)=1.d-30
           if(np.eq.1)goto104
           write(1,1013)(i,i=1,np)
           do102i=1,np
              do100j=1,i
100           a(j,i)=d(j,i)/e(i)/e(j)
102        write(1,1014)i,(a(j,i),j=1,i)
c
c          Calculate the regression coefficient.
c
104        sumc=0.d0
           sumc2=0.d0
           do105i=1,nob
105        yyi(i)=y(i)*y(i)
           do106i=1,nob
              sumc=sumc+y(i)
106        sumc2=sumc2+yyi(i)
           vinny=nob
           rsq=1.d0-sumb/(sumc2-sumc*sumc/vinny)
           write(1,1041)rsq
c
c          Calculate the 95% confidence interval.
c
           tut=nob-np
           z=1.d0/tut
           sdev=dsqrt(z*sumb)
           tvar=1.96d0+z*(2.3779d0+z*(2.7135d0+
     z          z*(3.187936d0+2.466666d0*z*z)))
           if(np.eq.1)write(1,1042)
           if(np.gt.1)write(1,1015)
           do107i=1,np
              secoef(i)=e(i)*sdev
107        tvalue(i)=th(i)/secoef(i)
           do108i=1,np
108        tsec(i)=tvar*secoef(i)
           do109i=1,np
              tmcoe(i)=th(i)-tsec(i)
109        tpcoe(i)=th(i)+tsec(i)
           if(np.eq.1)then
              do110i=1,np
                 j=2*i-1
110              write(1,1043)i,bi(j),bi(j+1),th(i),secoef(i),
     z                        tmcoe(i),tpcoe(i)
           else
              do111i=1,np
                 j=2*i-1
111              write(1,1016)i,bi(j),bi(j+1),th(i),secoef(i),
     z                        tvalue(i),tmcoe(i),tpcoe(i)
           endif
c
c          Prepare the final output.
c
```

```
           lsort(1)=1
           do112j=2,nob
112        temp1(j)=dabs(r(j))
           do117j=2,nob
              k=j-1
              do113l=1,k
                 if(temp1(j)-dabs(r(lsort(l))))114,114,113
113           continue
              lsort(j)=j
              goto117
114           kk=j
115           kk=kk-1
              lsort(kk+1)=lsort(kk)
              if(kk-l)116,116,115
116          lsort(l)=j
117        continue
           write(1,1017)
           do118i=1,nob
118        write(1,1018)i,x(i),y(i),f(i),r(i)
           if(iplot.ne.0)then
              do119i=1,nob
119           write(3,1047)y(i),x(i),i
              do120i=1,nob
120           write(3,1047)f(i),x(i),i
           endif
           write(1,1019)
           do121i=1,nob
              j=lsort(nob1-i)
121        write(1,1018)j,x(j),y(j),f(j),r(j)
           goto150
140        write(1,1030)(j,j=1,nx)
           do145i=1,nob
145        write(1,1033)i,x(i),f(i)
c
c       This is the end of this job.
c
150     continue
999     stop
1000    format(' Choose another file name.')
1001    format(a80)
1004    format(7(2x,2a4))
1005    format(7f10.5)
1006    format(7i10)
1007    format(/x,'Initial values of coefficients'/x,30(1h=)/3x,
     z          'Name',6x,'Initial value')
1008    format(x,2a4,4(1h.),1pd12.4)
1011    format(/x,'Iteration',6x,'SSQ',4x,8(5x,2a4))
1012    format(x,i5,3x,1pd13.4,2x,8(1pd13.4))
1013    format(/3x,'Correlation Matrix',/3x,18(1h=)/4x,10(4x,i2,5x))
1014    format(x,i3,10(2x,f7.4,2x))
1015    format(/x,'Nonlinear Least-Squares Analysis, Final Results.'
     z/x,48(1h=)/57x,'95% Confidence Limits'/x,'Variable',3x,'Name'
     z,7x,'Value',7x,'S.E.Coeff.',2x,'T-Value',4x,'Lower',9x,'Upper')
1016    format(4x,i2,3x,2a4,1pd13.4,2x,1pd13.4,d10.3,1pd12.4,
     z        x,1pd12.4)
1017    format(/,29x,'Ordered by computer input',/2x,'No.',4x,'  T  ',
     z 8x,'Conc Obs',6x,'Conc Fit',6x,'Residual')
1018    format(x,i3,x,1pd12.3,3(2x,1pd12.3))
1019    format(/,29x,'Ordered by residual',/2x,'No.',4x,'  T  '
     z        ,8x,'Conc Obs',6x,'Conc Fit',6x,'Residual')
```

```
1030    format(/x,'Results for initial coefficient values',/x,'No.',
     z   8x,'T  ',7x,'    Conc  ')
1033    format(x,i2,2(3x,1pd12.4))
1034    format(/x,'Convergence criteria not met in',i3,' iterations.')
1035    format(5x,/,' Nonlinear Least-Squares Analysis.'/)
1038    format(/x,'No further decrease in SSQ obtained after ',i2,
     z          ' trials.')
1041    format(/x,'Rsquare for regression =',d20.10,'.',/)
1042    format(x,'Nonlinear Least-Squares Analysis, Final Results.'
     z/x,48(1h=)/57x,'95% Confidence Limits'/x,'Variable',3x,'Name'
     z,8x,'Value',8x,'S.E.Coeff.',6x,'Lower',10x,'Upper')
1043    format(x,i2,6x,2a4,x,1pd13.4,3x,1pd13.4,x,1pd13.4,2x,
     z          1pd13.4)
1047    format(2(1pd15.4),i3)
2000    format(' Enter the name of the input data file',
     z          ' (default = tfm.dat).')
2001    format(a16)
2002    format(' Enter the name of the primary output file',
     z          ' (default = tfm.out).')
2003    format(' Enter the name of the plot output file',
     z          ' (default = tfm.plt).')
2004    format(' Enter the name of the file containing the ',
     z          'experimental data (default =',/,' expt.dat).')
2005    format(' There were ',i3,' experimental data read.')
2006    format(' P3 must not be less than zero, it has been reset ',
     z          'to zero.')
2007    format(' Set X3 as zero or one in the data file and rerun ',
     z          'the program.')
2008    format(' Yo must be greater than X3 (zero).  Reset Yo in the',
     z          /,' data file and rerun the program.')
2009    format(' Yo must be greater than X3 (one).  Reset Yo in the',
     z          /,' data file and rerun the program.')
2010    format(' B must not be less than or equal to zero or greater ',
     z          'than one.',/,' Reset B in the data file and rerun the ',
     z          'program.')
2011    format(' R must not be less than zero, it has been reset ',
     z          'to zero.')
2012    format(' W must not be less than zero, it has been reset ',
     z          'to zero.')
2013    format(' D must not be less than zero, it has been reset ',
     z          'to zero.')
2014    format(' C must not be less than zero, it has been reset ',
     z          'to zero.')
2015    format(' X3 cannot be fitted.  INDEX(2) is reset to 0.')
2016    format(' The minimum value of P3 cannot be less than or equal ',
     z          'to 0.  It ',/,' has been reset to P3LOW.')
2017    format(' The minimum value of Yo must be greater than 0 (X3).'
     z          ,/,' Reset the lower limit on Yo and rerun.')
2018    format(' The minimum value of Yo must be greater than 1 (X3).'
     z          ,/,' Reset the lower limit on y and rerun.')
2019    format(' The minimum value of B cannot be less than or equal to'
     z          ,'0.',/,' It has been reset to Blow.')
2020    format(' The maximum value of B cannot be greater than 1.',/,
     z          ' It has been reset to one.')
2021    format(' The minimum value of R cannot be less than 0.',/,
     z          ' It has been reset to zero.')
2022    format(' The minimum value of W cannot be less than 0.',/,
     z          ' It has been reset to zero.')
2023    format(' The minimum value of D cannot be less than 0.',/,
     z          ' It has been reset to zero.')
```

```
 2024     format(' The minimum value of C cannot be less than 0.',/,
     z           ' It has been reset to zero.')
 2025     format(' The experimental data must outnumber the number of'
     z           ,/,' parameters being fitted.')
 2026     format(' At least one time in the data file is less than or'
     z           ,/,' equal to zero.  Check and rerun.')
          end
          subroutinematinv(a,np,b)
c
c         Purpose: perform matrix inversion for parameter estimation.
c
          implicitdoubleprecision(a-h,o-z)
          dimensionpp(8),a(8,8),b(16),index(8,2)
          do2j=1,8
 2        index(j,1)=0
          i=0
 4        amax=-1.d0
          do10j=1,np
             if(index(j,1))10,6,10
 6           do9k=1,np
                if(index(k,1))9,8,9
 8              p=dabs(a(j,k))
                if(p.le.amax)goto9
                ir=j
                ic=k
                amax=p
 9        continue
 10       continue
          if(amax)30,30,14
 14       index(ic,1)=ir
          if(ir.eq.ic)goto18
          do15l=1,np
 15       pp(l)=a(ir,l)
          do16l=1,np
             a(ir,l)=a(ic,l)
 16       a(ic,l)=pp(l)
          p=b(ir)
          b(ir)=b(ic)
          b(ic)=p
          i=i+1
          index(i,2)=ic
 18       p=1.d0/a(ic,ic)
          a(ic,ic)=1.d0
          do20l=1,np
 20       a(ic,l)=a(ic,l)*p
          b(ic)=b(ic)*p
          do24k=1,np
             if(k.eq.ic)goto24
             p=a(k,ic)
             a(k,ic)=0.d0
             do22l=1,np
 22          a(k,l)=a(k,l)-a(ic,l)*p
             b(k)=b(k)-b(ic)*p
 24       continue
          goto4
 26       ic=index(i,2)
          ir=index(ic,1)
          do27k=1,np
 27       pp(k)=a(k,ir)
          do28k=1,np
```

```
            a(k,ir)=a(k,ic)
28       a(k,ic)=pp(k)
         i=i-1
30       continue
         if(i)26,32,26
32       return
         end
         subroutinemodel(bn,ycalc)
         implicitdoubleprecision(a-h,o-z)
         dimensionycalc(300),bn(16),c(8),index(8),x(300)
         parameter(p3low=1.d-6,blow=1.d-6)
         common/modat1/x,index,nob,nvar,nu1,nu2,iopt
c
c        Update the coefficient array.
c
         k=0
         do2i=nu1,nu2
            if(index(i-nvar).eq.0)goto2
            k=k+1
            bn(i)=bn(k)
2        continue
         do5i=1,nvar
5        c(i)=bn(nvar+i)
c
c        Check the parameter ranges.
c
         if(index(1).eq.1)then
            if(c(1).le.0.d0)c(1)=p3low
         endif
         if(index(3).eq.1)then
            if(c(3).le.c(2))c(3)=c(2)+1.d-9
         endif
         if(index(4).eq.1)then
            if(c(4).gt.1.d0)c(4)=1.d0
            if(c(4).le.0.d0)c(4)=blow
         endif
         if(index(5).eq.1)then
            if(c(5).lt.0.d0)c(5)=0.d0
         endif
         if(index(6).eq.1)then
            if(c(6).lt.0.d0)c(6)=0.d0
         endif
         if(index(7).eq.1)then
            if(c(7).lt.0.d0)c(7)=0.d0
         endif
         if(index(8).eq.1)then
            if(c(8).lt.0.d0)c(8)=0.d0
         endif
c
c        Call the TFM model function.
c
         callzres(ycalc,x,c,nob)
         return
         end
         subroutinezres(zfinv,zt,zc,n)
         implicitcomplex*16(a-h,o-y),real*8(z)
         externalf
         dimensionzt(n),zfinv(n),zc(8),zx(300),index(8)
         common/modat1/zx,index,nob,nvar,nu1,nu2,iopt/tutt/p3,x3,y,b,
     z         r,w,d,c
```

```
      datanotice/1/
c
c     The following parameters are for the IMSL Laplace inversion routine.
c     The inversion is accurate to nsig significant figures.  The maximum
c     number of iterations for each inversion is kmax.  See the IMSL manual
c     for details.
c
      parameter(nsig=4,kmax=499)
      if(iopt.eq.2)then
         callrealtime(zc,zt,zfinv,n)
         return
      endif
      icount=0
      p3=dcmplx(zc(1),0.d0)
      x3=dcmplx(zc(2),0.d0)
      y=dcmplx(zc(3),0.d0)
      b=dcmplx(zc(4),0.d0)
      r=dcmplx(zc(5),0.d0)
      w=dcmplx(zc(6),0.d0)
      d=dcmplx(zc(7),0.d0)
      c=dcmplx(zc(8),0.d0)
c
c     The function root calculates a number greater than or equal to the
c     real part of the maximum of the singularities of the function to be
c     inverted.  The argument is a dummy parameter.
c
      alpha=root(op)
      zzz=f1(alpha)
1     if(zzz.lt.0.d0)then
         zalpha=alpha
         alpha=alpha-dcmplx(1.d-6*zalpha,0.d0)
         zzz=f1(alpha)
      endif
      icount=icount+1
      if(icount.le.10.and.zzz.lt.0.d0)goto1
      if(icount.gt.10)write(6,11)
      zalpha=alpha
c
c     The estimate has been found; proceed to the inversion using the IMSL
c     numerical Laplace transform inverter.
c
      callflinv(f,n,zt,zalpha,nsig,kmax,zfinv,ier)
      do7i=1,n
7     if(zfinv(i).lt.0.d0)zfinv(i)=0.d0
      if(ier.eq.0)return
c
c     Stop printing error messages for more than 10 errors.
c
      ierc=ierc+1
      if(ierc.eq.11)write(6,16)
      if(ierc.gt.10)goto10
      write(6,12)ier,(zc(i),i=1,8)
      if(ier.eq.129)then
         write(6,13)
         goto10
      endif
      if(ier.eq.130)then
         write(6,14)
         goto10
      endif
      write(6,15)
10    if(iopt.eq.1)then
         callrealtime(zc,zt,zfinv,n)
         if(notice.eq.1)then
            notice=0
            write(6,17)
         endif
      endif
      return
11    format(' Problem with the required estimate for the maximum of '
     z      ,'the',/,' singularities in the Laplace transform of the'
     z      ,' TFM.',/,' Check the result subroutine (zres).')
12    format(' Error in IMSL numerical Laplace transform inversion '
     z      ,'(ier = ',i3,').',/,' the parameter values used are: ',
     z      'P3 = ',e16.8,', X3 = ',e16.8,',',/' Yo = ',e16.8,
     z      ', B = ',e16.8,', R = ',e16.8,',',/,' W = ',e16.8,
     z      ', D = ',e16.8,' and C = ',e16.8,'.',/,
     z      ' This error is due to the following:')
13    format(' The inversion algorithm was not able to achieve the'
     z      ,/,' accuracy requested within the Kmax function '
     z      ,/,' evaluations for some times.')
14    format(' An overflow would have occurred for a particular
     ztime.')
15    format(' The inversion algorithm increased the integration ',
     z      ' limit',/,'eight times without obtaining the requested ',
     z      ' accuracy.')
16    format(' Further error descriptions will not be printed.')
17    format(' Analytical solution being utilized.')
      end
      complex*16functionroot(op)
      implicitcomplex*16(a-h,o-y),real*8(z)
c
c     The estimate of the maximum of the real part of the singularities of
c     the TFM is obtained here.
c
      parameter(yzero=(0.d0,0.d0),four=(4.d0,0.d0),two=(2.d0,0.d0),
     z          one=(1.d0,0.d0))
      common/tutt/p3,x3,y,b,r,w,d,c
      if(b.eq.yzero)then
         w1=four*w
         root=-(p3*(w+d)+w1*d)/(p3+w1)/r
         return
      endif
      a=p3*p3+four*b*p3*c
      e=four*b*r*p3
      if(b.eq.one)then
         root=-a/e
         return
      endif
      if(w.eq.yzero)then
         bb=four*b
         root=-(p3+bb*c)/bb/r
         return
      endif
      f=four*r*p3*w
      g=four*p3*w*d
      h=d+w/(one-b)
      esr=e*r
      bi=(h*e+f+a*r)/esr
      bj=(a*h+g)/esr
      root=cdsqrt(bi*bi-four*bj)
      r2=-(root+bi)/two
      root=(root-bi)/two
      zr1=root
      zr2=r2
      if(zr2.gt.zr1)root=r2
      return
      end
      complex*16functionf(s)
c
c     The Laplace-transformed TFM is calculated here.
c
      implicitcomplex*16(a-h,o-y),real*8(z)
      common/tutt/p3,x3,y,b,r,w,d,c
      parameter(yzero=(0.d0,0.d0),one=(1.d0,0.d0),two=(2.d0,0.d0))
      if(s.eq.yzero)then
         f=yzero
         return
      endif
      f11=f1(s)
      x32=x3/two
      f=p3*(cdexp(x32*(p3-f11))-cdexp(x32*p3-y/two*f11))/f11/(one
     z  -cdexp(p3/two*(x3-y)))
      return
      end
      complex*16functionf1(s)
c
c     Function to calculate the f(s) portion of the TFM.
c
      implicitcomplex*16(a-h,o-y),real*8(z)
      common/tutt/p3,x3,y,b,r,w,d,c
      parameter(yzero=(0.d0,0.d0),one=(1.d0,0.d0),four=(4.d0,0.d0))
      if(b.eq.one)then
         f1=cdsqrt(p3*(p3+four*(r*s+c)))
         return
      endif
      if(b.eq.yzero)then
         r1=r*s+d
         f1=cdsqrt(p3*p3+four*p3*w*r1/(r1+w))
         return
      endif
      b1=one-b
      b22=four*p3
      b2=b22*b
      r1=r*s
      r2=b1*r1+w+d*b1
      f1=p3*p3+b2*r1*(one+b1*w/b/r2)+b2*c+b22*b1*w*d/r2
      zff1=f1
      if(zff1.gt.0.d0)f1=cdsqrt(f1)
      return
      end
c
c     Subroutine to evaluate the TFM analytically.
c
      subroutinerealtime(cc,tt,res,n)
      implicitdoubleprecision(a-h,o-z)
      dimensioncc(n),tt(n),res(n)
      externalfgrand
      common/param/d,p,b,r,w,c,t/tit1/os,p4p,rb,w1,tv,st,x,c1,b1,y
      parameter(zero=0.d0,pi=3.141592653589793238462643d0,rerr=1.d-5)
      x=cc(2)
      y=cc(3)
      p=cc(1)
      b=cc(4)
      r=cc(5)
      c=cc(8)
      d=cc(7)
      w=cc(6)
      pl=p/2.d0
      p4p=p/4.d0
      if(b.eq.1.d0)then
         pip=p4p/pi/r
         x2=-x*x
         y2=-y*y
         pl1=pl*x
         cz=c+p4p
         oz=1.d0-dexp((x-y)*pl)
         do3i=1,n
            tr=tt(i)/r
            pq=p4p/tr
3        res(i)=dsqrt(pip/tt(i))*dexp(pl1-tr*cz)/oz*(dexp(x2*pq)-
     z              dexp(y2*pq))
      else
         os=pl*(x-y)
         c1=c+w/b
         rb=r*b
         pt1=dsqrt(p4p/pi/rb)/(1.d0-dexp(os))
         do4i=1,n
            pt=pt1*dexp(-tt(i)/r*c1)
            ty=tt(i)/rb
            tr=-p4p/ty
            tv=dexp(tr*(x-ty)**2)-dexp(tr*(y-ty)**2+os)
            st=dsqrt(tt(i))
            res(i)=tv/st
            b1=1.d0-b
            w1=w/r/dsqrt(b*b1)
            if(w.eq.zero)then
               res(i)=res(i)*pt
            else
               t=tt(i)
               res(i)=pt*(res(i)+dcadre(fgrand,zero,tt(i),zero,rerr,
     z                err,ier)*w1)
               if(icount.gt.10)goto4
               if(ier.eq.131.or.ier.eq.132.or.ier.eq.133)then
                  icount=icount+1
                  write(6,5)
               endif
            endif
4        continue
      endif
      return
5     format(' Problem in the IMSL subroutine DCADRE.  Treat results',
     z       ' with caution.')
      end
c
c     Calculating the integrand.
c
      functionfgrand(tau)
      implicitdoubleprecision(a-h,o-z)
      real*8mmbsi1
      common/param/d,p,b,r,w,c,t/tit1/os,p4p,rb,w1,tv,st,x,c1,b1,y
      parameter(zero=0.d0)
      if(t.eq.tau)then
        fgrand=zero
        return
      endif
      if(tau.eq.zero)then
        fgrand=w1*st*tv
        return
      endif
      o=t-tau
      ff=-p4p*rb/o
      dd=o/rb
      fgrand=mmbsi1(1,2.*w1*dsqrt(tau*o),ier)*dexp(tau/r*(c1-d-w/
     z     b1))*(dexp(ff*(x-dd)**2)-dexp(ff*(y-dd)**2+os))/
     z     dsqrt(tau)
      if(ic.gt.10)goto1
      if(ier.eq.129.or.ier.eq.130)then
        ic=ic+1
        write(6,2)
      endif
1     return
2     format(' Problem in the IMSL subroutine MMBSI1.  Check'
     z     ,/,' the function fgrand.')
      end
c
c     Copyright, the Regents of the University of California.
c
```

APPENDIX 2.

A USEFUL LAPLACE TRANSFORM

To invert $\overline{g}(s)$ as given by eq. 17, it is necessary to know the inverse transform of $\overline{f}(s) = \exp[-(sx+a)/(s+b)]$. The procedure used to derive this transform is similar to that used by Goldstein (1953). Again, let $\pounds$ denote the Laplace transform operator, then (Carslaw and Jaeger, 1959, pg. 495):

$$\pounds\{I_0(2\sqrt{\chi t})\} = \frac{\exp(\chi/s)}{s} \tag{A1}$$

Noting that the operators $\pounds$ and d/dt are commutative, and using identities 5 and 3 of Spiegel (1965, Appendix A), one obtains from eq. A1 (cf. Walker, 1987)

$$\pounds\{\exp(-bt)\sqrt{\frac{\chi}{t}}\, I_1\ (2\sqrt{\chi t}\ )\} = \exp[\chi/(s+b)] - 1 \tag{A2}$$

The next step is to replace χ in eq. A2 by bx-a. If each side of eq. A2 is multiplied by exp(-x) and then rearranged, we arrive at the required identity

$$\pounds\{\delta(t)\ \exp(-x) + \sqrt{(bx-a)/t}\ \exp[-(x+bt)]\ I_1[2\sqrt{(bx-a)t}]\} =$$

$$\exp[-(sx+a)/(s+b)] \tag{A3}$$

APPENDIX 3. TFMPLOT: A FORTRAN code utilizing the DISSPLA (Integrated Software Systems Corp., 1985) software package to generate plots using the output plot files from TFMFIT.

```
c
c     Copyright, the Regents of the University of California.
c     Program to create a DISSPLA plot file for the TFM results.
c
      programtfmplot
      characterxlabel*24,ylabel*24,title*24,tit*16,
     z         datfich*8,parfich*8,plotnam*4
      dimensionx(300),y(300)
      calllink("unit6=tty//")
      write(6,120)
      read(6,122)parfich
      plotnam=parfich(4:7)
      callkeep80(plotnam,3)
      callfr80id('TFM plot',1,1,0)
      callplts
      callpage(11,8.5)
      callucchar
      callcomplx
      callbasalf('standard')
      callheight(.3)
      callintaxs
      callblsym
      callsclpic(2,0)
      open(2,file=parfich,status='old')
      read(2,140)xlong,ylong
      read(2,140)xphy,yphy
      read(2,140)xorig,xstep,xmax
      read(2,140)yorig,ystep,ymax
      read(2,150)tit
      encode(24,160,title)tit
      read(2,150)tit
      encode(24,160,xlabel)tit
      read(2,150)tit
      encode(24,160,ylabel)tit
      callarea2d(xlong,ylong)
      cal!physor(xphy,yphy)
      callheadin(title,100,5,1)
      callxname(xlabel,100)
      callyname(ylabel,100)
      callgraf(xorig,xstep,xmax,yorig,ystep,ymax)
      write(6,126)
      read(6,122)datfich
      open(2,file=datfich,status='old')
      do1i=1,602
1     read(2,170,end=2)y(i),x(i)
2     num=(i-1)/2
      rewind2
      do4i=1,2
         do3j=1,num
3        read(2,170)y(j),x(j)
         if(i.eq.1)goto4
         calldash
4     callcurve(x,y,num,0)
      calldendpl(0)
      calldonepl
      stop
```

```
120     format('Enter the name of the parameter file.')
122     format(a8)
126     format('Enter the name of the data file.')
140     format(14x,3(x,f7.0))
150     format(15x,a16,x)
160     format(a16,'       $')
170     format(2(1pe15.4))
        end
c
c       Copyright, the Regents of the University of California.
c
```