

MCMFIT: A Program to Fit a Generalized Non-linear Advection-Dispersion Model to Experimental Data

K. BAJRACHARYA AND D. A. BARRY



*Department of Environmental Engineering
Centre for Water Research
University of Western Australia
Nedlands, Western Australia 6009
Australia*

CONTENTS

	<i>Page</i>
TITLE PAGE	i
CONTENTS	ii
LIST OF TABLES	iii
LIST OF FIGURES	iii
NOTATION	iv
1. INTRODUCTION	1
2. MIXING CELL MODELS	1
2.1 Reaction processes	2
2.2 Mixing cell solution	3
3. PROGRAM DESCRIPTION	5
4. APPLICATIONS	9
REFERENCES	25
APPENDIX: LISTING OF MCMFIT	28

LIST OF TABLES

	<i>Page</i>
Table 1. Input data file for example 1	6
Table 2. Format of the input file for MCMFIT	7
Table 3. Parameter values used to generate the data in example 1	9
Table 4. MCMFIT results for example 1	10
Table 5. Parameter values used to create the data in Table 6	11
Table 6. Input data file for example 2	11
Table 7. MCMFIT results for data file given in Table 6	12
Table 8. MCMFIT result for constant VEL and D_s for data in Table 6	13
Table 9. Parameter values used to create the data in Table 10	14
Table 10. Input data file for example 4	14
Table 11. MCMFIT results for data in Table 10	15
Table 12. MCMFIT results for constant VEL and D_s for data in Table 10	16
Table 13. Parameter ranges used for Carbontetrachloride transport	18
Table 14. MCMFIT results for Carbontetrachloride data	18
Table 15. Parameter ranges for Tetrachloroethylene transport	19
Table 16 MCMFIT of two site adsorption model for TCE data	20
Table 17. MCMFIT results for TCE data (assuming fully kinetic model)	21
Table 18. MCMFIT results for laboraory data on Na-Ca exchange	23

LIST OF FIGURES

Figure 1. Fitting of hypothetical data for two site model	11
Figure 2. Fitting of hypothetical data for equilibrium model	14
Figure 3. Fitting of a concentration profile	17
Figure 4. Fitting field data on Carbontetrachloride	19
Figure 5. Fitting field data on Tetrachloroethylene	21
Figure 6. Fitting Na-Ca exchange data	24

NOTATION

C_{a1}	Penalty given by (18).
C_{a2}	Penalty given by (19).
C_r	Courant number, $V\Delta t/\Delta z$.
C	concentration of liquid phase, ML^{-3} .
C_0	influent liquid concentration, ML^{-3} .
$c(i)$	parameter values, $i = 1, NVAR$.
c_{max}	maximum parameter values.
c_{min}	minimum parameter values.
d_{max}	maximum observed concentration, ML^{-3} .
d_{min}	minimum observed concentration, ML^{-3} .
D_s	dispersion coefficient, L^2T^{-1} .
$f(C)$	function of C (adsorption isotherm).
$NVAR$	number of parameters.
P	column Peclet number, VL/D_s .
S	total solid phase concentration (equivalent liquid phase), ML^{-3} .
S_1	solid phase concentration on equilibrium sites (equivalent liquid phase), ML^{-3} .
S_2	solid phase concentration on kinetic nonequilibrium sites (equivalent liquid phase), ML^{-3} .
S_{2e}	equilibrium solid phase concentration on kinetic nonequilibrium sites (equivalent liquid phase), ML^{-3} .
t	time, T .
V	mean pore water velocity, LT^{-1} .
z	distance below soil surface, L .
α_i	isotherm fitting parameter used in (7b) ($i = 1,2,3,4$).
α_5	kinetic adsorption rate constant, T^{-1} .
α_6	fraction of the sites available for instantaneous adsorption.
β	difference between maximum and minimum concentrations.
Δt	time step, T .
Δz	spatial step, L .

1. INTRODUCTION

The problem of estimating model parameters is encountered frequently in practice. There are several packages available for estimating the parameters for linear advection-dispersion equations (ADE) for which there are exact solutions (e.g., CXTFIT, TFMFIT, etc.). For non-linear transport equations, the use of standard numerical solutions (e.g., Crank-Nicolson) to estimate parameters is very time consuming and hence inefficient. On the other hand, mixing-cell solutions are very efficient by comparison. In particular, the solution obtained from the improved mixing cell model has been found to agree very well with the results of a numerical Crank-Nicolson solution while being much more efficient (*Bajracharya and Barry, 1993a*). Thus, an improved mixing cell model has been used here to estimate model parameters for a variety of transport models. The code, MCMFIT, makes use of nonlinear least-squares fitting to find optimal parameter values by matching improved mixing cell model predictions with measured experimental data. The experimental data can be either in the form of a breakthrough curve or concentrations within a soil profile. The program can handle linear, Freundlich, Langmuir, and S-curve adsorption isotherms in conjunction with the transport equation. Both equilibrium and non-equilibrium (fully kinetic and two site adsorption) cases can be dealt with along with first and third type surface boundary conditions. Use of the program is demonstrated with a number of examples. Both synthetic data as well as data from field and laboratory experiments have been used in the illustrative examples.

2. MIXING CELL MODELS

Explicit finite difference schemes (mixing cell models) have been used frequently in the modelling of reactive and non-reactive solute transport problems in soil and groundwater (*Schweich and Sardin, 1981; Dance and Reardon, 1983; Schulz and Reardon, 1983; Schweich et al., 1983; Van Ommen, 1985; Appelo and Willemssen, 1987; Rao and Hathaway, 1989; Dudley et al., 1991*). These schemes can be easily programmed and are very efficient to compute. Besides, they can be linked with chemical speciation models to simulate complex reactions coupled with transport. The characteristic feature of the so-called mixing cell models is that physical dispersion in the ADE is simulated wholly by controlling the numerical dispersion introduced by the finite difference representation of the purely advective transport equation.

There are two types of predictive model used in analysing transport of reacting solutes (*Rubin, 1983*). In one, the reaction is sufficiently fast to be described by an equilibrium model. In the other, the reaction is relatively slow and is quantified by a nonequilibrium model. The governing one-dimensional solute transport equation with adsorption is given by (symbols are defined in the notation list):

$$\frac{\partial(S + C)}{\partial t} = D_s \frac{\partial^2 C}{\partial z^2} - V \frac{\partial C}{\partial z}, \quad (1)$$

where D_s and V are each assumed to be constant. Note that S is expressed here as an equivalent liquid phase concentration. If S is expressed as a solid phase concentration, it has to be multiplied by a factor RHO which is the ratio of bulk density to porosity of the medium. This equation describes solute transport subject to adsorption under steady flow conditions. Initially, the solute concentration in both the solid and liquid phases is taken to be:

$$C(z,0) = 0 \quad (2)$$

and

$$S(z,0) = 0. \quad (3)$$

It is immaterial whether the right-hand sides of (2) and (3) are zero or non-zero, so we take zero for convenience. The non-zero initial concentration can be easily changed in the program. The first boundary condition is a constant concentration of solute at $z = 0$ (entrance b. c. type 1) for a pulse of duration PULSET

$$C(0,t) = C_0, \quad 0 < t \leq \text{PULSET}. \quad (4a)$$

Equation (4a) is valid if C is considered to be a flux concentration (*Kreft and Zuber, 1978; van Genuchten and Parker, 1984; Barry and Sposito, 1988*), although it is often used for resident concentrations as well.

The other commonly used surface condition is the mixed, or flux boundary condition (entrance b. c. type 3), given by (*Brenner, 1962*):

$$VC(0,t) - D_s \frac{\partial C(0,t)}{\partial z} = VC_0, \quad 0 < t \leq \text{PULSET}. \quad (4b)$$

Equation (4b) results in a macroscopic discontinuity in concentration at $z = 0$. We require another boundary condition to specify the solution to (1). For a semi-infinite domain the appropriate condition is

$$\left. \frac{\partial C}{\partial z} \right|_{z \rightarrow \infty} = 0, \quad (5)$$

which simply states that C is bounded as $z \rightarrow \infty$. This is the condition satisfied by the mixing cell model because the solution domain used by mixing cell model is semi-infinite.

2.1 Reaction processes

The term $\partial S/\partial t$ in the left hand side of (1) describes the reaction process. When the reaction, e.g., an adsorption process or ion exchange reaction, is sufficiently fast, the adsorption isotherm or an exchange reaction is written as an algebraic function of C , and so is used to eliminate the adsorbed phase concentration, S . Such an operation gives rise to the equilibrium model. It should be noted that when an ion exchange reaction is considered, two chemical species are involved. However, the problem reduces to that of single species when the normality of the solution is held constant in the feed and in the native solution. When the reaction process is insufficiently fast, an additional differential rate equation must be used to solve (1). This results in the nonequilibrium model. A thorough discussion concerning classes of reactions can be found in the paper by *Rubin (1983)*. This equation models the reaction rate but does not explain the microscopic processes taking place. Adsorption kinetics can be expressed mathematically as:

$$\frac{\partial S}{\partial t} = \alpha_5 [f(C) - S], \quad (6)$$

where

$$S_e = f(C) \quad (7a)$$

is the adsorption isotherm at equilibrium. Equations (1) and (7) with appropriate entrance and exit conditions constitute the equilibrium model whereas equations (1), (6) and (7) with additional entrance conditions and exit conditions like equations (4a) or (4b) and (5) constitute the fully nonequilibrium model.

A model which considers both equilibrium and nonequilibrium adsorption simultaneously is the two site adsorption model as discussed by *Selim et al.* (1976), *Cameron and Klute* (1977) and *Rao et al.* (1979). The sorption sites consist of equilibrium and nonequilibrium fractions. At equilibrium, adsorption on both the equilibrium and the nonequilibrium sites is described by a nonlinear isotherm given by

$$S_1 = \alpha_6 f(C) \quad (8)$$

and

$$S_{2e} = (1 - \alpha_6) f(C), \quad (9)$$

where α_6 is the fraction of the total number of sites where equilibrium sorption occurs. It must be noted that, at equilibrium, the total adsorption i.e., $(S_1 + S_2)$ is that given by equation (7a). The two site model is then given by

$$\frac{\partial C}{\partial t} + \frac{\partial S_1}{\partial t} + \frac{\partial S_2}{\partial t} = D_s \frac{\partial^2 C}{\partial z^2} - V \frac{\partial C}{\partial z}, \quad (10)$$

with the nonequilibrium adsorption process defined by

$$\frac{\partial S_2}{\partial t} = \alpha_5 [(1 - \alpha_6) f(C) - S_2]. \quad (11)$$

Equations (8) - (11) with entrance boundary conditions (4a) or (4b) and exit condition (5) constitute the two site adsorption model. This model is mathematically identical to the problem of solute transport under physical nonequilibrium conditions, attributed to the large heterogeneities in microscopic pore water velocities (*van Genuchten*, 1981).

The two site adsorption models have been efficiently solved using mixing cell models and so the latter solution method is used in the fitting program. The improved mixing cell model is described below.

2.2 Mixing Cell Solution

Upon dropping second derivative term in the advective-dispersive equation, (1), one gets a pure advective transport equation given by:

$$\frac{\partial(C + S)}{\partial t} = -V \frac{\partial C}{\partial z}. \quad (12)$$

Finite-difference solutions of (12) produce numerical dispersion which is controlled to simulate solutions to (1).

The general solution for two site model using improved mixing cell model is presented below. Writing (10) in finite difference form at two time levels, we get (*Bajracharya and Barry*, 1993b)

$$(2 + C_r) C(i, j+1) + 2 [S_1(i, j+1) + S_2(i, j+1)] = 2 [C(i, j) + S_1(i, j) + S_2(i, j)] + C_r [C(i-1, j) - C(i, j) + C(i-1, j+1)], \quad (13)$$

where $C(i, j) = C(i\Delta z, j\Delta t)$ and i and j denote spatial and temporal coordinates in the finite difference grid system. The finite difference forms of (8) and (11) are,

$$S_1(i, j+1) = \alpha_6 f[C(i, j+1)] \quad (14)$$

and

$$S_2(i,j+1) = \frac{\frac{\alpha_5 \Delta t}{2} (1 - \alpha_6) \{f[C(i,j+1)] + f[C(i,j)]\} + \left(1 - \frac{\alpha_5 \Delta t}{2}\right) S_2(i,j)}{1 + \frac{\alpha_5 \Delta t}{2}}, \quad (15)$$

respectively.

In this scheme the advective term is approximated by the average of the backward difference approximations at the (i,j) and $(i,j+1)$ grid points as in the conventional Crank-Nicolson scheme (e.g., *Noye, 1982*). The combination of (13), (14), and (15) results in an equation with $C(i,j+1)$ as the only unknown and constitutes the improved two site mixing cell model solution for the problem (8) - (11), and (5). The entrance condition (4a) is straight-forward to apply. The entrance condition (4b) is incorporated at $z = 0$ by the central difference approximation (*Noye, 1982*). This reduces to a system of equations in two unknowns at the first two grid points. Newton's method is used to solve the two nonlinear equations. The two equations to be solved are:

$$\begin{aligned} \left(1 + \frac{C_r}{2} + \frac{C_r V \Delta z}{D_s}\right) C(0,j+1) + [S_1(0,j+1) + S_2(0,j+1)] - \frac{C_r}{2} C(1,j+1) = \\ \left(1 - \frac{C_r}{2} - \frac{C_r V \Delta z}{D_s}\right) C(0,j) + \frac{C_r}{2} C(1,j) + \frac{2 C_r V \Delta z C_0}{D_s} + [S_1(0,j) + S_2(0,j)], \end{aligned} \quad (16)$$

and

$$\begin{aligned} \left(1 + \frac{C_r}{2}\right) C(1,j+1) + [S_1(1,j+1) + S_2(1,j+1)] - \frac{C_r}{2} C(0,j+1) = \\ \frac{C_r}{2} C(0,j) + \left(1 - \frac{C_r}{2}\right) C(1,j) + [S_1(1,j) + S_2(1,j)]. \end{aligned} \quad (17)$$

The solid phase concentrations in (16) and (17) are eliminated using (14) and (15). Note that when α_6 equals 0 or 1 the two site adsorption model reduces to the fully kinetic or fully equilibrium models, respectively.

At the rest of the grid points, the scheme reduces to a single equation in one unknown (*Barry and Bajracharya, 1993*). The bisection method is used to solve these nonlinear equations. Due to rounding error, the bisection technique encountered some mild numerical problems when the liquid phase solute concentrations were close to zero. In such cases, the bisection method is terminated and the liquid phase concentration equated to zero.

The mixing cell method makes use of the inevitable numerical dispersion induced by the finite difference scheme adapted for the pure advective nonlinear transport equation to simulate physical dispersion in the full ADE. The numerical dispersion in the improved scheme is $V \Delta z / 2$ (*Bajracharya and Barry, 1993a*). This scheme is second order accurate. The spatial step in the model is fixed from the ratio $2D_s/V$ which is obtained by equating D_s to the numerical dispersion. When D_s is very large or V is very small, the spatial step becomes very large which is a limitation of mixing cell model. The program stops with the message "PECLET NUMBER TOO LOW" when the number of cells is less than 3. Similarly, the program halts with the message "VARIABLE ARRAY SIZE EXCEEDED" when the number of cells exceed 2000 while fitting a breakthrough curve and 200 when fitting a profile. This, however, can be adjusted by simply increasing the array size of the variables. Unlike the standard mixing cell model, the temporal step is independent of the spatial step in this case, permitting larger time steps than those allowed by standard mixing cell without affecting accuracy. However, the time steps chosen should be such that the results are sufficiently accurate.

The experimental data points are obtained either at regular or irregular intervals in time or space. The model generated concentrations may not exactly coincide the observed data spatially or temporally. Thus, interpolations are necessary to evaluate the experimental concentrations at the given point in space and time. Linear interpolation has been used for this purpose.

A general S-curve isotherm has been incorporated in the transport model. A function which gives an S-shaped isotherm is (Barry, 1992):

$$S = \alpha_1 \{ 1 - [1 + (\alpha_2 C)^{\alpha_3}]^{\alpha_4} \}. \quad (7b)$$

This equation can represent linear, Freundlich and Langmuir isotherms. For example, when $\alpha_1 = -1$ and $\alpha_4 = 1$, it reduces to Freundlich equation. Likewise, when $\alpha_1 = -1$, $\alpha_3 = \alpha_4 = 1$, it reduces to linear form. It reduces to the Langmuir type when $\alpha_4 = -1$ and $\alpha_3 = 1$. It is because of this versatility that the above equation was used in the development of the present fitting program. However, when the isotherm is linear, the mixing cell model gives a linear equation at each of the grid point and can be directly solved. Thus, the improved mixing cell model is, in fact, usually faster than the existing linear fitting packages which make use of complicated exact solutions involving the evaluation of complicated integrals.

3. PROGRAM DESCRIPTION

A FORTRAN program to carry out the nonlinear least-squares fitting of the general nonlinear transport model to experimental data has been developed. The program MCMFIT is based on the maximum neighbourhood method of *Marquardt* (1963) as implemented by *van Genuchten* (1980, 1981), *Parker and van Genuchten* (1984) and *Barry et al.* (1988). The listing of the program is given in the Appendix.

Hendrickson et al. (1988) have shown that, in the case of rainfall-runoff models, there is no single best algorithm that is optimally efficient, or even convergent to already known parameter values. This statement likely applies to the estimation of solute transport parameters as well. MCMFIT can fit nine parameters, viz., the pore water velocity (VEL), dispersion coefficient (D_s), pulse duration (PULSET), the four adsorption isotherm coefficients ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$), the rate coefficient (α_5) and the fraction of adsorption site (α_6). An attempt has been made to obtain optimal solutions in this study. A problem with the *Marquardt* (1963) and similar least-squares fitting method is that they are prone to find parameters corresponding to local minima rather than the global minimum. As is well known, the starting values of the fitted parameters must be chosen judiciously to avoid local minima. MCMFIT includes an option to carry out a random search of the parameter space in order to find a suitable set of initial parameter values. The following approach is used. Sets of parameter values are first generated randomly within user-specified parameter ranges. For each parameter set, the mixing cell solution is calculated and compared with the observed experimental data. Using the residual sum of squares as the error criterion, the best set of parameter values, out of all the randomly generated sets is stored. The greater the number of sets, the more likely the "best" is to be a good estimate of the global minimum. This parameter set is passed to the *Marquardt* (1963) algorithm to determine precisely the best-fit parameters.

MCMFIT is a constrained fitting program. In other words, ranges for the parameter values to be fitted have to be specified. MCMFIT maintains constraints on the parameter values by fitting an additional fictitious data point of zero concentration. To force the predicted concentration zero (to fit the additional fictitious data point), a method similar to penalty functions employed in classical optimization (*Beveridge and Schechter*, 1970) has been incorporated into the code. During the fitting, if the parameters all lie in the required ranges, then the fitting proceeds as usual. Any parameters outside the specified range are

penalised. Let ψ_u represent the set of parameters exceeding their upper bounds. Then, the penalty function is:

$$C_{a1} = \sum_{i \in \psi_u} \beta \exp \left[\frac{\mu(i)}{1 - \frac{\alpha(i)}{c_{\max}(i)}} \right]. \quad (18)$$

For parameters that are less than the specified minimum, the penalty is given by:

$$C_{a2} = \sum_{i \in \psi_l} \beta \exp \left[\frac{\mu(i)}{1 - \frac{\alpha(i)}{c_{\min}(i)}} \right], \quad (19)$$

where ψ_l is the set of parameters less than their lower bound,

$$\mu(i) = \frac{c_{\max}(i) - c_{\min}(i)}{\gamma}, \quad (20)$$

and

$$\beta = d_{\max} - d_{\min}. \quad (21)$$

Equations (18) and (19) are continuous for finite γ . Note that increasing γ (strictly positive) increases the penalty for an out-of-range parameter. If γ is increased, the penalty will eventually be large enough to force the parameter into the desired range. In MCMFIT, $\gamma = 10$ is taken initially. If the parameter values are still out of range then γ is increased by multiples of 10 until all parameters lie in the specified ranges.

The program consists of the main program, MCMFIT, and the subroutines MATINV, MODEL, FUNC1, FUNC2, FNC2, FUNC3, FUNC4, NLSYST and RAN3. The main program handles the input and output of data, generates the parameter data set for which the sum of residual squares is minimum and implements the fitting program. MATINV performs matrix inversion as required by the fitting program for parameter estimation. MODEL organises data predicted by various models FUNC1, FUNC2, FUNC3 and FUNC4. FNC2 is the function as required by the system of nonlinear equation solver, NLSYST. The subroutine, NLSYST has been adapted from the original published by *Gerald* (1978). The subroutine RAN3 generates random numbers between 0.0 and 1.0. This subroutine has been adapted from the book, Numerical Recipes, Press et al. (1992). The system time in seconds has been used as the seed for the random generator in MCMFIT.

Upon execution, MCMFIT looks for an input file (MCMFIT.DAT is the default) containing the necessary information on the model to be fitted and the experimental data. Once this file name is entered, the program then prompts for the results filename. A plot filename which consists of the observed and the fitted results can also be created. Table 1 gives example parameter values as required by the MCMFIT. The line-by-line format of this file is presented in Table 2.

Table 1. Input data file for example 1.

NVAR	NOB	MIT	MAXTRY	IPLLOT	MODE	NRAN
9	51	30	20	1	2	100
CLEN	C01	DT	RHO	TIMET		
100.0000	1.0000	1.0000	1.0000			
Example 1						
Fitting hypothetical experimental data (NRAN = 100)						
vel	ds	pulset	alpha1	alpha2	alpha3	alpha4
0	0	0	0	1	1	0
5.0000	12.5000	10.0000	-1.0000	0.5000	0.1000	1.0000
5.0000	12.5000	10.0000	-1.0000	2.5000	1.8000	1.0000

alpha5	alpha6
1	1
0.1000	0.1000
4.0000	1.0000
0.0000	4.0000
0.0000	8.0000
0.0000	12.0000
0.0000	16.0000
0.0000	20.0000
0.0000	24.0000
0.0002	28.0000
0.0049	32.0000
0.0319	36.0000
0.0968	40.0000
0.1830	44.0000
0.2546	48.0000
0.2895	52.0000
0.2875	56.0000
0.2606	60.0000
0.2223	64.0000
0.1822	68.0000
0.1455	72.0000
0.1145	76.0000
0.0893	80.0000
0.0694	84.0000
0.0539	88.0000
0.0420	92.0000
0.0328	96.0000
0.0258	100.0000
0.0204	104.0000
0.0162	108.0000
0.0130	112.0000
0.0104	116.0000
0.0085	120.0000
0.0069	124.0000
0.0057	128.0000
0.0047	132.0000
0.0039	136.0000
0.0032	140.0000
0.0027	144.0000
0.0023	148.0000
0.0019	152.0000
0.0017	156.0000
0.0014	160.0000
0.0012	164.0000
0.0010	168.0000
0.0009	172.0000
0.0008	176.0000
0.0007	180.0000
0.0006	184.0000
0.0005	188.0000
0.0005	192.0000
0.0004	196.0000
0.0004	200.0000

Table 2. Format of input file as required by MCMFIT

Line	Columns	Format	Description
2	1-10	I10	NVAR: Number of variables in the model.
	11-20	I10	NOB: Number of observations.
	21-30	I10	MIT: Maximum number of iterations allowed in the least-squares fitting. If set to 0, the program simply calculates breakthrough curve using initial parameter values.
	31-40	I10	MXTRY: Number of trial solutions allowed at each iteration.
	41-50	I10	IPLOT: 1 (0) - do (not) create plot file.

	51-60	I10	MODE: codes 1 to 4 selects the model to be fitted. 1. Two site model with b. c. type 1 (breakthrough curve); 2. Two site model with b. c. type 3 (breakthrough curve); 3. Two site model with b. c. type 1 (profile); 4. Two site model with b. c. type 3 (profile);
	61-70	I10	NRAN: Number of times the random set of parameters is to be generated.
4	1-10	F10.4	CLEN: Column length.
	11-20	F10.4	C01: Influent concentration.
	21-30	F10.4	DT: Time step.
	31-40	F10.4	RHO: Ratio of bulk density to porosity of the medium.
	41-50	F10.4	TIMET: Time at which concentration profile is to be generated.
5	1-80	A80	Descriptive label.
6	1-80	A80	Descriptive label.
7	1-10, 11-20, etc.	2A4	Parameter names. VEL = pore velocity. D_s = dispersion coefficient. PULSET = pulse period. $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are s-curve isotherm parameters. α_5 = kinetic mass transfer coefficient. α_6 = fraction of equilibrium site.
8	"	I10	Fitting code: (0) 1- (do not) fit this parameter.
9	"	F10.4	Lower limit allowed for parameter.
10	"	F10.4	Upper limit allowed for parameter.
11	1-10	F10.4	Value of the observed concentration (blank for MIT = 0).

The output file consists of:

1. Descriptive title;
2. Initial values of coefficients (obtained from the random search);
3. Iteration number and sum-of-squares for the given parameter values;
4. Correlation matrix of the fitted parameters;
5. The R^2 value for the regression;
6. Optimized parameter values and their 95% confidence limits;
7. Fitted and observed data as ordered by the input file;
8. Fitted and observed data as ordered by the residual magnitude.

The plot file consists of the following:

1. List of observed data (Concentration and time/space);
2. List of fitted data (Concentration and time/space).

4. APPLICATIONS

Example 1.

The first example considered is the fitting of breakthrough data resulting from a two site model with boundary condition type 3. This example shows that by increasing the number of random searches, a more suitable initial parameter set can be determined, resulting in rapid convergence to the correct parameter values. There is, however, the computational expense of the time taken in the generation of the near optimal initial parameter set. A synthetic breakthrough data set has been used to demonstrate this example. This synthetic data set was then fitted using MCMFIT in which 10 and 100 random sets of parameter values were generated. The parameter values used to generate the synthetic data are shown in Table 3. This particular example considers a Freundlich isotherm.

Table 3. Parameter values used to generate the data in example 1.

C01	VEL	D_s	CLEN	PULSET	α_1	α_2	α_3	α_4	α_5	α_6
1.0	5.0	12.5	100.0	10.0	-1.0	1.5	0.8	1.0	1.0	0.5

This represents a column of length 100 cm, through which a reactive solute with a unit relative concentration is flowing at a pore velocity of 5 cm d⁻¹. The medium is characterised by a hydrodynamic dispersion coefficient of 12.5 cm² d⁻¹. Of the medium's exchange sites, half are equilibrium sites and half are nonequilibrium. The solute is introduced as a 10-day pulse. The breakthrough data obtained from this hypothetical experiment (shown in Table 1) were fitted using MCMFIT. The data were rounded to four decimal places and a total of 50 data pairs were used. In order to reduce the number of parameters, velocity, dispersion coefficient, and pulse time were held constant. These parameters are assumed known. The four remaining parameters, viz., the Freundlich isotherm parameters α_2 and α_3 , the kinetic rate coefficient, α_5 , and fraction of the adsorption site, α_6 were fitted. The range of these fitting parameters are shown in the data file in Table 1. Also 100 random sets were generated. Observe that S-curve isotherm reduces to Freundlich isotherm when $\alpha_1 = -1$ and $\alpha_4 = 1$. The results as obtained from MCMFIT are shown in Table 4.

Table 4. MCMFIT result for example 1.

NON-LINEAR LEAST-SQUARES ANALYSIS
 Example 1
 Fitting hypothetical experimental data (NRAN = 100)

INITIAL VALUES OF COEFFICIENTS

```
=====
NAME          INITIAL VALUE
vel.....    5.0000E+00
ds.....     1.2500E+01
pulset.....  1.0000E+01
alpha1..... -1.0000E+00
alpha2.....  1.5430E+00
alpha3.....  8.5778E-01
alpha4.....  1.0000E+00
alpha5.....  1.0821E+00
alpha6.....  3.4864E-01
```

ITERATION	SSQ	alpha2	alpha3	alpha5	alpha6
0	8.8110E-04	1.5430E+00	8.5778E-01	1.0821E+00	3.4864E-01
1	1.2745E-05	1.4961E+00	7.9619E-01	1.2924E+00	3.1728E-01
2	6.6993E-07	1.4989E+00	7.9938E-01	1.0280E+00	4.9256E-01
3	3.3301E-08	1.5000E+00	7.9994E-01	9.9541E-01	5.0198E-01
4	3.2525E-08	1.5000E+00	7.9994E-01	9.9813E-01	5.0080E-01
5	3.2524E-08	1.5000E+00	7.9994E-01	9.9812E-01	5.0079E-01

CORRELATION MATRIX

```
=====
      1      2      3      4
1  1.0000
2  0.9413  1.0000
3 -0.9489 -0.8566  1.0000
4  0.9471  0.8537 -0.9991  1.0000
```

RSQUARE FOR REGRESSION =0.99999991
 NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS

```
=====
```

VARIABLE	NAME	VALUE	S.E. COEFF.	95% CONFIDENCE LIMITS		
				T-VALUE	LOWER	UPPER
1	alpha2	1.5000E+00	1.8714E-04	8.0154E+03	1.4996E+00	1.5004E+00
2	alpha3	7.9994E-01	1.0216E-04	7.8303E+03	7.9973E-01	8.0014E-01
3	alpha5	9.9812E-01	1.1081E-02	9.0072E+01	9.7583E-01	1.0204E+00
4	alpha6	5.0079E-01	5.5396E-03	9.0403E+01	4.8965E-01	5.1194E-01

-----ORDERED BY COMPUTER INPUT-----

NO	T1	OBS	Y FITTED	RESI- DUAL
1	4.000E+00	0.000E+00	3.725E-09	-3.725E-09
2	8.000E+00	0.000E+00	3.725E-09	-3.725E-09
..
..
49	1.960E+02	4.000E-04	4.068E-04	-6.802E-06
50	2.000E+02	4.000E-04	3.599E-04	4.008E-05

-----ORDERED BY RESIDUAL-----

NO	T1	OBS	Y FITTED	RESI- DUAL
29	1.160E+02	1.040E-02	1.045E-02	-5.398E-05
14	5.600E+01	2.875E-01	2.875E-01	4.830E-05
..
..
4	1.600E+01	0.000E+00	3.725E-09	-3.725E-09
5	2.000E+01	0.000E+00	3.725E-09	-3.725E-09

When 10 random parameter sets were used, the fit obtained was not satisfactory (not included). However, by increasing the number of random parameter sets to 100, the initial guesses for the fitting parameters were observed to be sufficiently close to optimal values as is evident from Table 4. The data and the fitted curves are shown in Figure 1.

Note that the fitted parameters in Table 4 are highly correlated. That means there exists different sets of parameter values for which the fit is almost as good. *Beven et al.* (1993) have also reported the very strong dependence between the fitted parameters while using CXTFIT (*Parker and van Genuchten, 1984*). In this example, the fitted parameter values were essentially exact. In practice, data are affected by random measurement and analysis errors. Thus, good fits of the data do not necessarily imply that the solute transport and reaction mechanisms have been correctly identified.

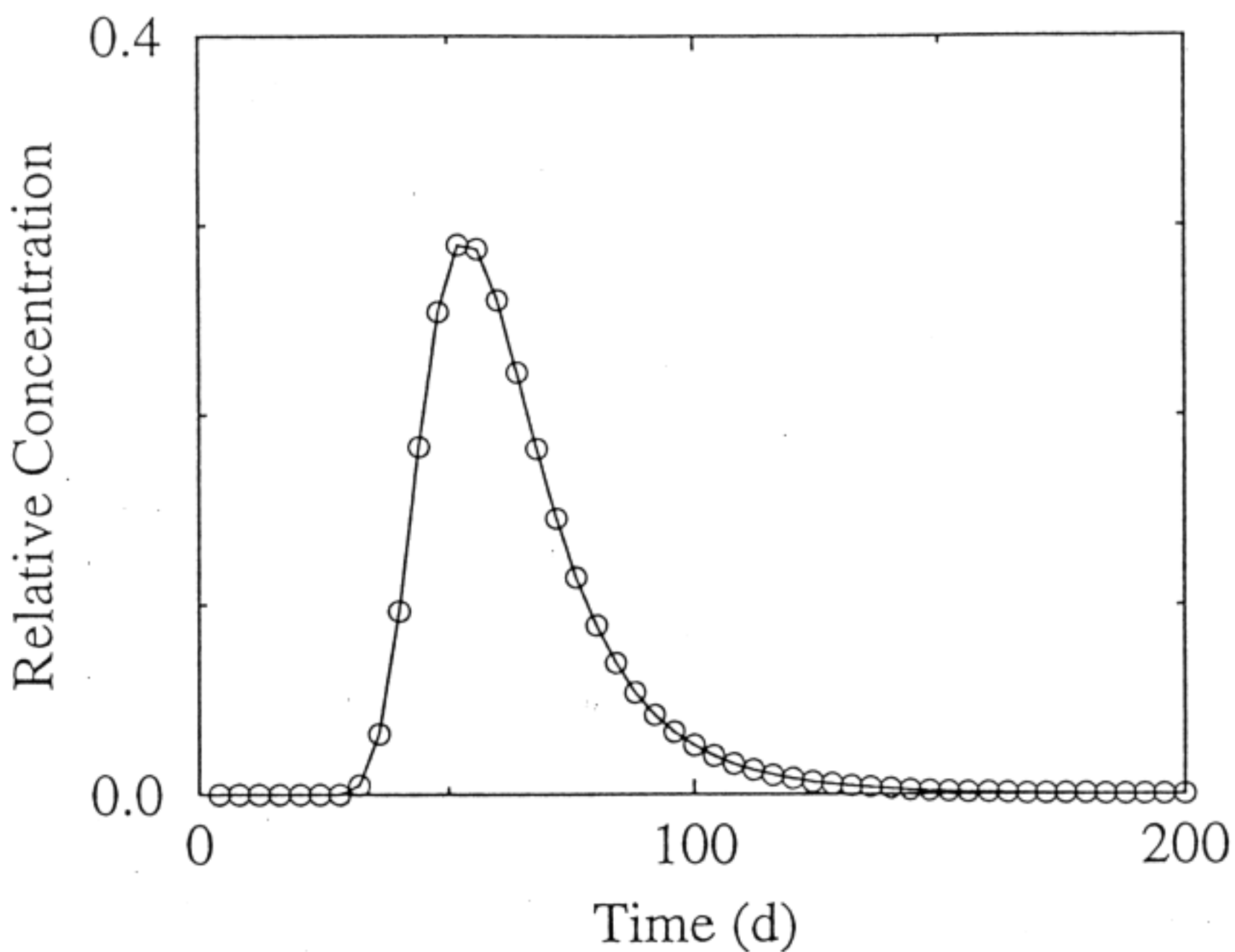


Figure 1. Fitting of hypothetical data from the two site model with boundary condition type 3. Breakthrough data (circles) are fitted by MCMFIT (line).

Example 2.

As a second example, an equilibrium Freundlich adsorption isotherm is considered. This example shows clearly the problem of converging to incorrect parameter values although maintaining an excellent fit. In the example shown below, a data set was fitted using MCMFIT in which NRAN chosen was 300. The correct parameter values used to generate the data were:

Table 5. Parameter values used to create the data in Table 6.

C01	VEL	D _s	CLEN	PULSET	α_1	α_2	α_3	α_4	α_5	α_6
1.0	5.0	12.5	100.0	9999.0	-1.0	0.872	0.456	1.0	0.0	1.0

Solute with a relative concentration of unity is passed through the column and the effluent concentration at the exit is measured until the effluent and influent concentrations are equal. The pulse time was set to a very large number so that a complete breakthrough was obtained. The data points were rounded to four decimal places and 23 data pairs were used in the fitting process. The data file is given in Table 6. In this case, the velocity, dispersion coefficient and Freundlich isotherm parameters were fitted. The kinetic coefficient was set at zero and fraction of sites at 1 and these are held constant so that the model then reduces to equilibrium model. The boundary condition chosen is type one so that MODE = 1.

Table 6. Input data file for example 2.

NVAR	NOB	MIT	MAXTRY	IPLLOT	MODE	NRAN
9	23	30	20	1	1	300
CLEN	C01	DT	RHO	TIMET		
100.0000	1.0000	1.0000	1.4500			
Example 2.						
Fitting hypothetical data with four parameters.						
vel	ds	pulset	alpha1	alpha2	alpha3	alpha4
1	1	0	0	1	1	0
1.0000	10.5000	9999.0000	-1.0000	0.1720	0.1560	1.0000

8.0000	14.5000	9999.0000	-1.0000	1.9720	1.5560	1.0000
alpha5	alpha6					
0	0					
0.0000	1.0000					
0.0000	1.0000					
0.0000	38.0000					
0.0000	39.0000					
0.0000	40.0000					
0.0006	41.0000					
0.0076	42.0000					
0.0403	43.0000					
0.1194	44.0000					
0.2411	45.0000					
0.3815	46.0000					
0.5174	47.0000					
0.6354	48.0000					
0.7308	49.0000					
0.8045	50.0000					
0.8598	51.0000					
0.9003	52.0000					
0.9296	53.0000					
0.9505	54.0000					
0.9654	55.0000					
0.9759	56.0000					
0.9832	57.0000					
0.9884	58.0000					
0.9920	59.0000					
0.9944	60.0000					

Table 7 gives the results of the data fitting.

Table 7. MCMFIT results for Data file given in Table 6.

NON-LINEAR LEAST-SQUARES ANALYSIS
 Example 2.
 Fitting hypothetical data with four parameters.

INITIAL VALUES OF COEFFICIENTS

```
=====
NAME          INITIAL VALUE
vel.....    6.3003E+00
ds.....     1.4423E+01
pulset.....  9.9990E+03
alpha1..... -1.0000E+00
alpha2.....  1.6432E+00
alpha3.....  5.9423E-01
alpha4.....  1.0000E+00
alpha5.....  0.0000E+00
alpha6.....  1.0000E+00
```

ITERATION	SSQ	vel	ds	alpha2	alpha3
0	3.6297E-02	6.3003E+00	1.4423E+01	1.6432E+00	5.9423E-01
1	1.2282E-03	6.1975E+00	1.4066E+01	1.6655E+00	5.5783E-01
2	9.2950E-04	6.1621E+00	1.4036E+01	1.6510E+00	5.5075E-01
3	5.6912E-04	6.1617E+00	1.3988E+01	1.6506E+00	5.5029E-01

NO FURTHER DECREASE IN SSQ OBTAINED AFTER 20 TRIALS

CORRELATION MATRIX

```
=====
      1      2      3      4
1  1.0000
2 -0.9572  1.0000
3  0.9872 -0.9080  1.0000
4  0.2657 -0.4880  0.1132  1.0000
```

RSQUARE FOR REGRESSION =0.99986218
 NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS

```
=====
VARIABLE  NAME      VALUE      S.E. COEFF.  95% CONFIDENCE LIMITS
          T1      OBS      T-VALUE     LOWER      UPPER
1        vel     6.1617E+00  6.2310E-02   9.8887E+01  6.0317E+00  6.2917E+00
2        ds     1.3988E+01  1.6329E-01   8.5660E+01  1.3647E+01  1.4328E+01
3      alpha2   1.6506E+00  4.3919E-02   3.7582E+01  1.5590E+00  1.7422E+00
4      alpha3   5.5029E-01  5.5533E-03   9.9093E+01  5.3871E-01  5.6188E-01
```

-----ORDERED BY COMPUTER INPUT-----

NO	T1	OBS	Y FITTED	RESI- DUAL
1	3.800E+01	0.000E+00	5.338E-08	-5.338E-08
2	3.900E+01	0.000E+00	5.449E-06	-5.449E-06
..
..
22	5.900E+01	9.920E-01	9.924E-01	-3.576E-04
23	6.000E+01	9.944E-01	9.948E-01	-3.920E-04

-----ORDERED BY RESIDUAL-----				
NO	T1	OBS	Y FITTED	RESI- DUAL
6	4.300E+01	4.030E-02	5.310E-02	-1.280E-02
7	4.400E+01	1.194E-01	1.317E-01	-1.225E-02
..
..
2	3.900E+01	0.000E+00	5.449E-06	-5.449E-06
1	3.800E+01	0.000E+00	5.338E-08	-5.338E-08

It is observed that the final values did not converge to the correct parameter values. α_2 is highly correlated with dispersion coefficient, D_s and this dispersion coefficient in turn is highly correlated with velocity.

Example 3.

The above example was fitted again by running MCMFIT but with the correct velocity and dispersion coefficient fixed on input. This time, α_2 and α_3 converged to the correct values as shown in Table 8. The number of parameter sets generated in this case was only 10 because there were only two parameters fitted.

Table 8. MCMFIT result for fixed velocity and dispersion coefficient for data in Table 6.

NON-LINEAR LEAST-SQUARES ANALYSIS
Example 3.
Fitting hypothetical data with two parameters.

INITIAL VALUES OF COEFFICIENTS

NAME	INITIAL VALUE
vel.....	5.0000E+00
ds.....	1.2500E+01
pulset.....	9.9990E+03
alpha1.....	-1.0000E+00
alpha2.....	9.5759E-01
alpha3.....	5.7734E-01
alpha4.....	1.0000E+00
alpha5.....	0.0000E+00
alpha6.....	1.0000E+00

ITERATION	SSQ	alpha2	alpha3
0	9.5704E-02	9.5759E-01	5.7734E-01
1	2.0859E-03	8.8543E-01	4.8095E-01
2	9.0476E-06	8.7298E-01	4.5741E-01
3	2.9954E-08	8.7202E-01	4.5605E-01
4	1.7618E-08	8.7199E-01	4.5599E-01
5	1.7597E-08	8.7199E-01	4.5599E-01

CORRELATION MATRIX

	1	2
1	1.0000	
2	0.7148	1.0000

RSQUARE FOR REGRESSION =1.00000000
NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS

VARIABLE	NAME	VALUE	S.E. COEFF.	95% CONFIDENCE LIMITS		
				T-VALUE	LOWER	UPPER
1	alpha2	8.7199E-01	9.2274E-06	9.4500E+04	8.7197E-01	8.7201E-01
2	alpha3	4.5599E-01	1.6550E-05	2.7552E+04	4.5595E-01	4.5602E-01

-----ORDERED BY COMPUTER INPUT-----				
NO	T1	OBS	Y FITTED	RESI- DUAL
1	3.800E+01	0.000E+00	1.078E-10	-1.078E-10
2	3.900E+01	0.000E+00	1.097E-07	-1.097E-07
..
..
22	5.900E+01	9.920E-01	9.920E-01	4.631E-05
23	6.000E+01	9.944E-01	9.944E-01	-4.824E-05

-----ORDERED BY RESIDUAL-----				
NO	T1	OBS	Y FITTED	RESI- DUAL
5	4.200E+01	7.600E-03	7.551E-03	4.852E-05
23	6.000E+01	9.944E-01	9.944E-01	-4.824E-05
..
..
2	3.900E+01	0.000E+00	1.097E-07	-1.097E-07
1	3.800E+01	0.000E+00	1.078E-10	-1.078E-10

The data and the fitted curve are shown in Figure 2.

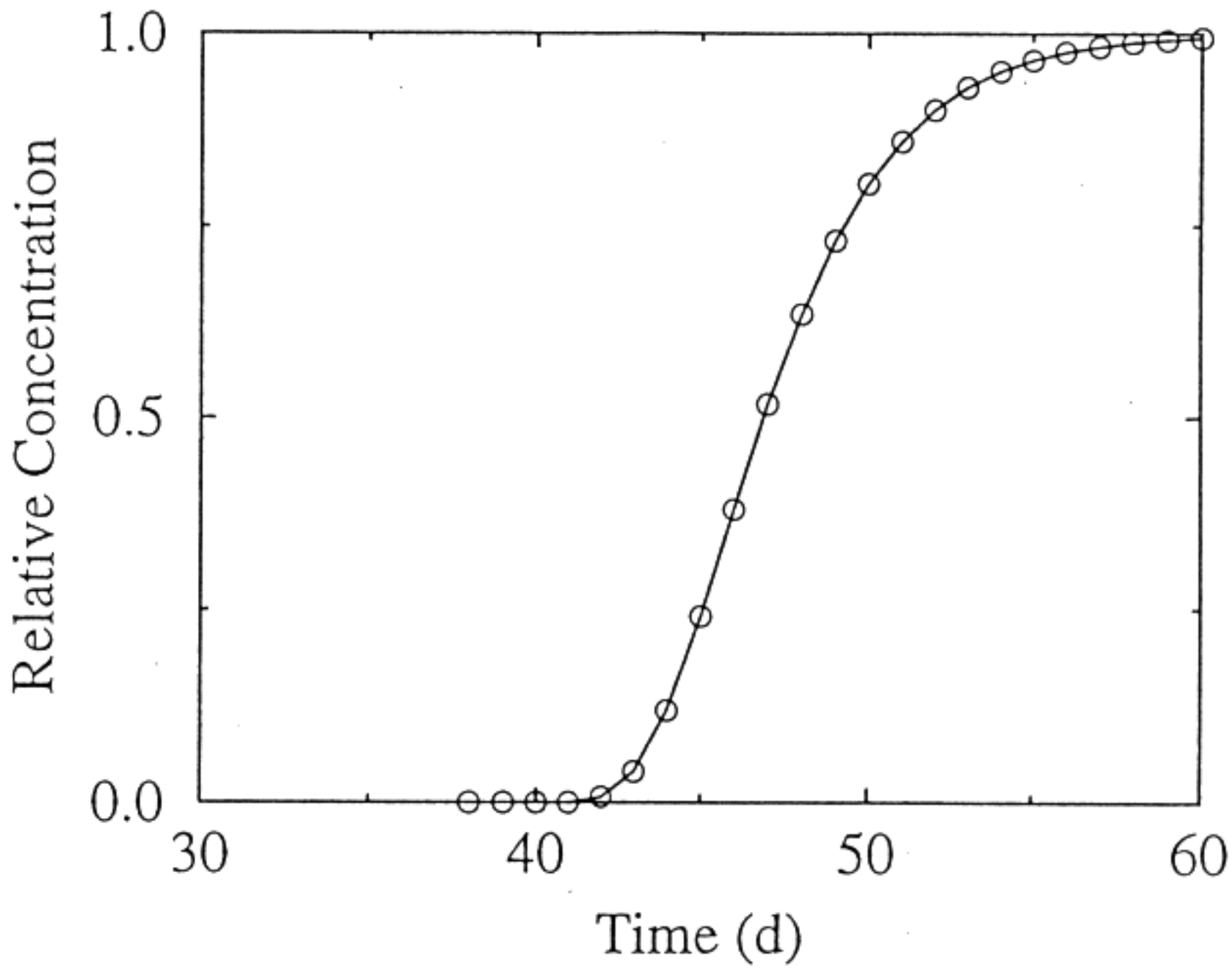


Figure 2. Fitting of hypothetical data from the equilibrium model with boundary condition type 1. Breakthrough data (circles) are fitted by MCMFIT (line).

Example 4.

We show next the use of MCMFIT in fitting a concentration profile. Concentration profile along a hypothetical soil column was generated using the parameters shown in Table 9 at a time of 40 d. The entrance b. c. type 1 was considered.

Table 9. Parameter values used to create the data in Table 10.

C01	VEL	D_s	CLEN	PULSET	α_1	α_2	α_3	α_4	α_5	α_6
1.0	1.0	1.0	30.0	9999.0	-1.0	0.872	0.456	1.0	0.0	1.0

In this case, a column of 30 cm is assumed to be flooded with relative solute concentration of unity at a pore water velocity of 1 cm d^{-1} and dispersion coefficient, $1 \text{ cm}^2 \text{ d}^{-1}$ for a period of 40 d and the relative concentration determined along the column at that time. The input data file is shown in Table 10. 15 data pairs have been fitted in this example.

Table 10. Input data file for example 4.

```

NVAR      NOB      MIT      MAXTRY    IPLOT     MODE      NRAN
  9        15      30       20        1         3        100
CLEN      C01      DT       RHO       TIMET
30.0000  1.0000  1.0000  1.4500  40.0000
Example 4
Fitting concentration profile data
  vel      ds      pulset   alpha1   alpha2   alpha3   alpha4
    1       1       0        0        1        1        0
0.5000    0.5000  923.0000 -1.0000  0.1720  0.1560  1.0000
2.0000    2.0000  923.0000 -1.0000  1.8720  10.3560  1.0000
alpha5    alpha6
  0        0
0.0000    1.0000
0.0000    1.0000

1.0000    2.0000
0.9999    4.0000

```


0.9992	6.0000
0.9958	8.0000
0.9840	10.0000
0.9499	12.0000
0.8655	14.0000
0.6866	16.0000
0.3855	18.0000
0.0818	20.0000
0.0010	22.0000
0.0000	24.0000
0.0000	26.0000
0.0000	28.0000
0.0000	30.0000

Table 11 gives the results of fitting the given concentration profile.

Table 11. MCMFIT results for data in Table 10.

NON-LINEAR LEAST-SQUARES ANALYSIS
 Example 4
 Fitting concentration profile data

INITIAL VALUES OF COEFFICIENTS

```

=====
NAME          INITIAL VALUE
vel.....    1.1306E+00
ds.....     1.7719E+00
pulset.....  9.2300E+02
alpha1..... -1.0000E+00
alpha2.....  1.3483E+00
alpha3.....  4.7033E-01
alpha4.....  1.0000E+00
alpha5.....  0.0000E+00
alpha6.....  1.0000E+00
  
```

ITERATION	SSQ	vel	ds	alpha2	alpha3
0	3.9037E-02	1.1306E+00	1.7719E+00	1.3483E+00	4.7033E-01
1	3.2906E-02	1.1711E+00	1.8810E+00	1.7082E+00	2.4518E-01
2	1.1449E-02	1.1688E+00	1.8892E+00	1.8194E+00	2.6198E-01
3	6.7657E-03	1.1668E+00	1.8952E+00	1.8470E+00	2.6623E-01
4	4.3468E-03	1.1652E+00	1.8999E+00	1.8681E+00	2.6840E-01
5	1.6527E-03	1.1562E+00	1.9067E+00	1.8681E+00	2.6593E-01
6	9.7178E-04	1.1460E+00	1.8858E+00	1.8681E+00	2.4505E-01
7	9.1663E-04	1.1438E+00	1.8811E+00	1.8681E+00	2.4143E-01
8	9.1263E-04	1.1435E+00	1.8807E+00	1.8681E+00	2.4125E-01
9	9.1255E-04	1.1435E+00	1.8807E+00	1.8681E+00	2.4127E-01

CORRELATION MATRIX

```

=====
          1          2          3          4
1    1.0000
2    0.7868    1.0000
3    0.0033    0.0016    1.0000
4    0.8143    0.5981    0.0000    1.0000
  
```

RSQUARE FOR REGRESSION =0.99972236
 NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS

```

=====
VARIABLE  NAME      VALUE      S.E. COEFF.  95% CONFIDENCE LIMITS
          1         vel    1.1435E+00  4.4114E-03  T-VALUE      LOWER      UPPER
          2         ds     1.8807E+00  1.3996E-02  1.3438E+02  1.8502E+00  1.9112E+00
          3      alpha2    1.8681E+00  1.6291E-04  1.1467E+04  1.8677E+00  1.8685E+00
          4      alpha3    2.4127E-01  1.0147E-02  2.3777E+01  2.1916E-01  2.6338E-01
  
```

-----ORDERED BY COMPUTER INPUT-----

NO	X1	OBS	Y FITTED	RESI- DUAL
1	2.000E+00	1.000E+00	9.999E-01	9.537E-05
2	4.000E+00	9.999E-01	9.994E-01	4.715E-04
3	6.000E+00	9.992E-01	9.983E-01	9.387E-04
4	8.000E+00	9.958E-01	9.924E-01	3.428E-03
5	1.000E+01	9.840E-01	9.825E-01	1.501E-03
6	1.200E+01	9.499E-01	9.434E-01	6.490E-03
7	1.400E+01	8.655E-01	8.540E-01	1.145E-02
8	1.600E+01	6.866E-01	6.956E-01	-8.997E-03
9	1.800E+01	3.855E-01	3.867E-01	-1.230E-03
10	2.000E+01	8.180E-02	7.451E-02	7.293E-03
11	2.200E+01	1.000E-03	2.526E-02	-2.426E-02
12	2.400E+01	0.000E+00	7.801E-08	-7.801E-08
13	2.600E+01	0.000E+00	1.062E-08	-1.062E-08
14	2.800E+01	0.000E+00	4.547E-13	-4.547E-13
15	3.000E+01	0.000E+00	4.547E-13	-4.547E-13

-----ORDERED BY RESIDUAL-----

NO	X1	OBS	Y FITTED	RESI- DUAL
11	2.200E+01	1.000E-03	2.526E-02	-2.426E-02
7	1.400E+01	8.655E-01	8.540E-01	1.145E-02

8	1.600E+01	6.866E-01	6.956E-01	-8.997E-03
10	2.000E+01	8.180E-02	7.451E-02	7.293E-03
6	1.200E+01	9.499E-01	9.434E-01	6.490E-03
4	8.000E+00	9.958E-01	9.924E-01	3.428E-03
5	1.000E+01	9.840E-01	9.825E-01	1.501E-03
9	1.800E+01	3.855E-01	3.867E-01	-1.230E-03
3	6.000E+00	9.992E-01	9.983E-01	9.387E-04
2	4.000E+00	9.999E-01	9.994E-01	4.715E-04
1	2.000E+00	1.000E+00	9.999E-01	9.537E-05
12	2.400E+01	0.000E+00	7.801E-08	-7.801E-08
13	2.600E+01	0.000E+00	1.062E-08	-1.062E-08
14	2.800E+01	0.000E+00	4.547E-13	-4.547E-13
15	3.000E+01	0.000E+00	4.547E-13	-4.547E-13

It is clear that the parameter set have converged to wrong values as in the previous example.

Example 5.

In this example run, the pore water velocity and the dispersion coefficient were fixed at the correct values and all remaining parameters were the same as in the previous example except for the number of random searches (NRAN) which was reduced to 10. The results of MCMFIT is shown in Table 12.

Table 12. MCMFIT results with fixed velocity and dispersion coeff. for data in Table 10.

NON-LINEAR LEAST-SQUARES ANALYSIS
Example 5
Fitting concentration profile data using two parameters.

INITIAL VALUES OF COEFFICIENTS

```
=====
NAME          INITIAL VALUE
vel.....    1.0000E+00
ds.....    1.0000E+00
pulset..... 9.2300E+02
alpha1..... -1.0000E+00
alpha2.....  4.4029E-01
alpha3.....  3.6540E-01
alpha4.....  1.0000E+00
alpha5.....  0.0000E+00
alpha6.....  1.0000E+00
```

ITERATION	SSQ	alpha2	alpha3
0	3.1420E-01	4.4029E-01	3.6540E-01
1	3.4022E-02	1.0063E+00	6.5779E-01
2	5.9749E-04	8.9647E-01	4.7043E-01
3	5.3422E-07	8.7268E-01	4.5691E-01
4	6.9929E-09	8.7206E-01	4.5611E-01
5	6.7414E-09	8.7205E-01	4.5609E-01

CORRELATION MATRIX

```
=====
          1          2
1    1.0000
2    0.6421    1.0000
```

RSQUARE FOR REGRESSION =1.00000000
NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS

```
=====
```

VARIABLE	NAME	VALUE	S.E. COEFF.	95% CONFIDENCE LIMITS		
				T-VALUE	LOWER	UPPER
1	alpha2	8.7205E-01	2.2465E-05	3.8817E+04	8.7200E-01	8.7210E-01
2	alpha3	4.5609E-01	2.5538E-05	1.7859E+04	4.5604E-01	4.5615E-01

-----ORDERED BY COMPUTER INPUT-----

NO	X1	OBS	Y FITTED	RESI- DUAL
1	2.000E+00	1.000E+00	1.000E+00	8.136E-06
2	4.000E+00	9.999E-01	9.999E-01	1.647E-05
3	6.000E+00	9.992E-01	9.992E-01	4.735E-05
4	8.000E+00	9.958E-01	9.958E-01	-5.889E-06
5	1.000E+01	9.840E-01	9.840E-01	-3.095E-05
6	1.200E+01	9.499E-01	9.499E-01	-2.377E-05
7	1.400E+01	8.655E-01	8.655E-01	1.774E-05
8	1.600E+01	6.866E-01	6.866E-01	-8.158E-06
9	1.800E+01	3.855E-01	3.855E-01	1.083E-05
10	2.000E+01	8.180E-02	8.181E-02	-1.250E-05
11	2.200E+01	1.000E-03	9.558E-04	4.415E-05
12	2.400E+01	0.000E+00	1.293E-08	-1.293E-08
13	2.600E+01	0.000E+00	4.547E-13	-4.547E-13
14	2.800E+01	0.000E+00	4.547E-13	-4.547E-13
15	3.000E+01	0.000E+00	4.547E-13	-4.547E-13

NO	X1	OBS	Y FITTED	RESI- DUAL
3	6.000E+00	9.992E-01	9.992E-01	4.735E-05
11	2.200E+01	1.000E-03	9.558E-04	4.415E-05
5	1.000E+01	9.840E-01	9.840E-01	-3.095E-05
6	1.200E+01	9.499E-01	9.499E-01	-2.377E-05
7	1.400E+01	8.655E-01	8.655E-01	1.774E-05
2	4.000E+00	9.999E-01	9.999E-01	1.647E-05
10	2.000E+01	8.180E-02	8.181E-02	-1.250E-05
9	1.800E+01	3.855E-01	3.855E-01	1.083E-05
8	1.600E+01	6.866E-01	6.866E-01	-8.158E-06
1	2.000E+00	1.000E+00	1.000E+00	8.136E-06
4	8.000E+00	9.958E-01	9.958E-01	-5.889E-06
12	2.400E+01	0.000E+00	1.293E-08	-1.293E-08
13	2.600E+01	0.000E+00	4.547E-13	-4.547E-13
14	2.800E+01	0.000E+00	4.547E-13	-4.547E-13
15	3.000E+01	0.000E+00	4.547E-13	-4.547E-13

Here, it is observed that the parameters converged to the correct values. The data and the fitted concentration profiles are shown in Figure 3.

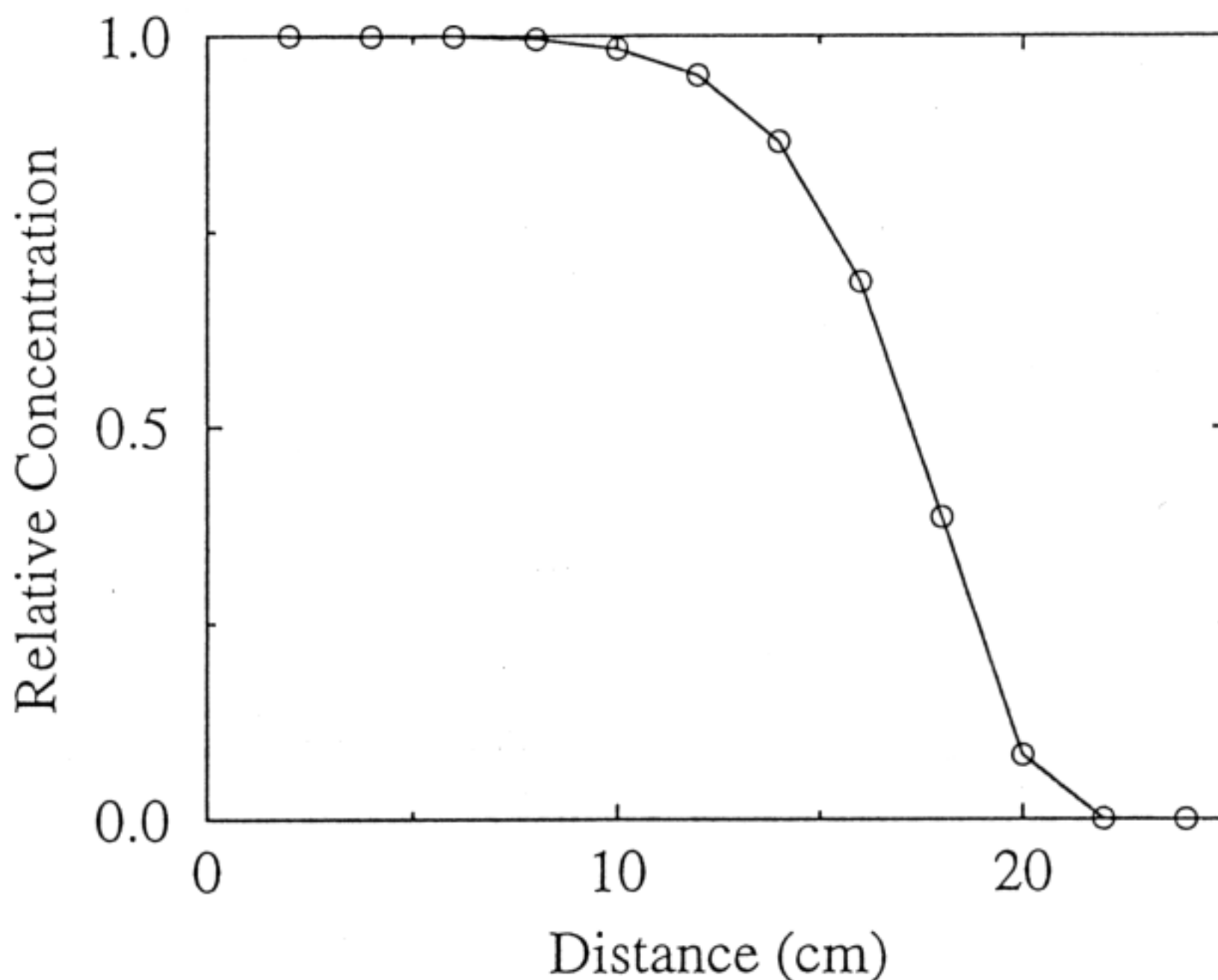


Figure 3. Fitting of a concentration profile. Data (circles) are fitted by MCMFIT (line).

Example 6.

We next apply the fitting program to the carbontetrachloride (CT) breakthrough curve at a sampling location in the Borden aquifer (*Roberts and Mackay, 1986*). The experiment consisted of a pulse injection of 12 m^3 of a uniform solution containing two inorganic tracers and five halogenated organic chemicals into a shallow unconfined aquifer. The details of the experiment are documented in several papers (e.g., *Freyberg, 1986; Goltz and Roberts, 1986; Mackay et al, 1986; Sudicky, 1986*). It is assumed that the disappearance of CT is attributed to adsorption onto the aquifer material and this process follows a Freundlich isotherm. The first type boundary condition is assumed in this case. *Goltz and Roberts (1986)* assumed the result of the solute injection as a rectangular prism in which a uniform slug of solute having width 3.2 m and concentration C_0 . Clearly, the initial plume configuration is subject to uncertainty. In the mixing cell solution, the solute pulse was assumed to be transported at the mean ground water velocity of 0.091 m d^{-1} . The pulse time was then estimated as $3.2/0.091 = 35 \text{ d}$. The breakthrough curve of CT obtained at a point 5 m downstream of the injection point was considered. The remaining four parameters, viz., the dispersion coefficient (D_s), the pulse time, and the Freundlich parameters (α_2 and α_3) were fitted to the data using the nonlinear equilibrium model. In

Table 13 below, the parameter ranges used are given. The number of parameter sets generated randomly was 100 (NRAN).

Table 13. Parameter ranges used in estimating parameters for Carbontetrachloride transport.

D_s	PULSET	α_2	α_3
0.007-0.03	20.0-36.0	0.1-8.0	0.1-1.0

The result of the fitting exercise is presented in Table 14.

Table 14. MCMFIT results for Borden Experiment data on Carbontetrachloride.

NON-LINEAR LEAST-SQUARES ANALYSIS
 Example 5.
 Fitting Carbon Tetrachloride data using equilibrium model.

INITIAL VALUES OF COEFFICIENTS

```

=====
NAME          INITIAL VALUE
vel.....    9.1000E-02
ds.....     1.8589E-02
pulset.....  2.4185E+01
alpha1..... -1.0000E+00
alpha2.....  8.2485E-01
alpha3.....  6.4385E-01
alpha4.....  1.0000E+00
alpha5.....  0.0000E+00
alpha6.....  1.0000E+00
  
```

ITERATION	SSQ	ds	pulset	alpha2	alpha3
0	4.2620E-01	1.8589E-02	2.4185E+01	8.2485E-01	6.4385E-01
1	4.2241E-01	2.2723E-02	2.4185E+01	7.7831E-01	6.8267E-01
2	1.7053E-01	1.4537E-02	2.4185E+01	8.4675E-01	7.7292E-01
3	7.0564E-02	9.0278E-03	2.4185E+01	7.8699E-01	7.7899E-01
4	5.6070E-02	9.7910E-03	2.4185E+01	8.6386E-01	8.4573E-01
5	5.3814E-02	8.8565E-03	2.4185E+01	8.6383E-01	8.4520E-01
6	5.3119E-02	9.0311E-03	2.4185E+01	8.6646E-01	8.5170E-01
7	5.2950E-02	9.1016E-03	2.4185E+01	8.7344E-01	8.6119E-01
8	5.2908E-02	9.1281E-03	2.4185E+01	8.7632E-01	8.6511E-01
9	5.2908E-02	9.1256E-03	2.4185E+01	8.7631E-01	8.6519E-01
10	5.2908E-02	9.1256E-03	2.4185E+01	8.7631E-01	8.6519E-01

CORRELATION MATRIX

```

=====
      1      2      3      4
1  1.0000
2  0.0000  1.0000
3 -0.1608  0.0000  1.0000
4 -0.2414  0.0000  0.8952  1.0000
  
```

RSQUARE FOR REGRESSION =0.97086011
 NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS

```

=====
VARIABLE  NAME      VALUE      S.E. COEFF.  95% CONFIDENCE LIMITS
          T-VALUE  LOWER      UPPER
1         ds      9.1256E-03  1.6707E-04  5.4620E+01  8.7941E-03  9.4570E-03
2         pulset  2.4185E+01  2.3002E-32  1.0514E+33  2.4185E+01  2.4185E+01
3         alpha2  8.7631E-01  1.4360E-02  6.1027E+01  8.4782E-01  9.0480E-01
4         alpha3  8.6519E-01  1.8329E-02  4.7202E+01  8.2882E-01  9.0156E-01
  
```

-----ORDERED BY COMPUTER INPUT-----

NO	T1	OBS	Y FITTED	RESI- DUAL
1	1.000E+00	5.400E-03	4.657E-10	5.400E-03
2	1.700E+01	1.400E-02	1.863E-09	1.400E-02
..
102	1.980E+02	2.150E-02	1.588E-02	5.620E-03
103	2.000E+02	1.900E-02	1.442E-02	4.577E-03

-----ORDERED BY RESIDUAL-----

NO	T1	OBS	Y FITTED	RESI- DUAL
57	1.040E+02	3.072E-01	3.781E-01	-7.093E-02
21	5.200E+01	5.700E-02	1.188E-06	5.700E-02
..
17	4.400E+01	3.000E-04	2.384E-09	3.000E-04
8	3.000E+01	0.000E+00	1.863E-09	-1.863E-09

It is interesting to note that the correlation coefficient of the parameter "pulset" to itself is zero and also this parameter did not change with succeeding iterations. This is because the time step (DT) adopted (1 d) in the mixing cell solution was too coarse. The Freundlich isotherm parameters K and N in the equation $S = KC^N$ as determined by the fitting results were 0.16 and 0.86, respectively. The corresponding parameters reported as obtained from independent batch experiments were 0.25 and 0.89. The results are thus satisfactory. Figure 4 shows the observed and the fitted breakthrough curves of CT.

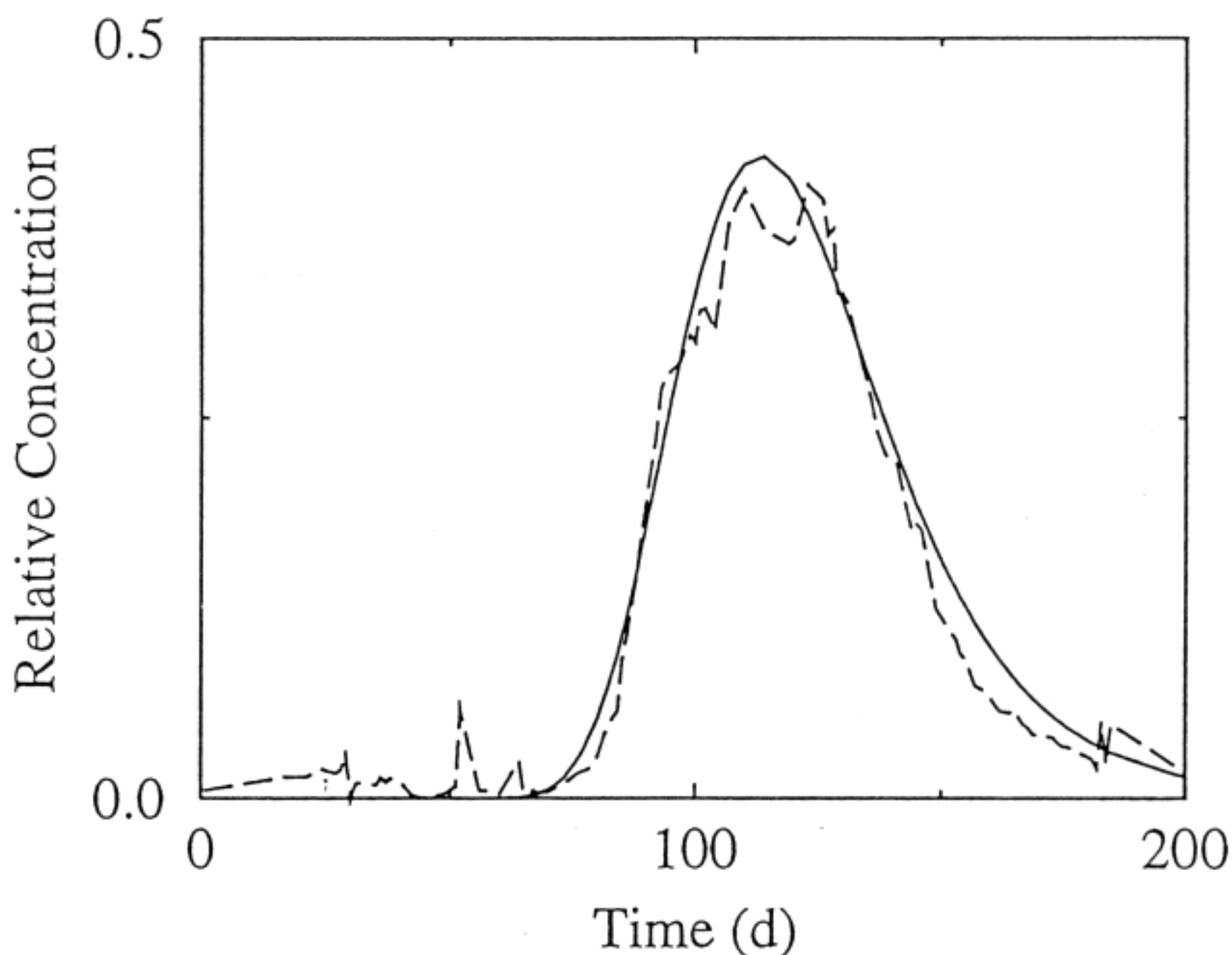


Figure 4. Field data on Carbontetrachloride (dashes) and fitted by MCMFIT (line).

Example 7.

The seventh example is that of tetrachloroethylene (TCE), also from the Borden experiment. In this case a kinetic model was fitted because there is some evidence of kinetically controlled partitioning of the solute between the liquid and solid phases (*Barry et al.*, 1988). Thus, the two site adsorption model was fitted. Here the velocity was held constant and the other six parameters were fitted. The parameter range is given below in Table 15.

Table 15. Parameter ranges used in estimating parameters for Tetrachloroethylene transport.

D_s	PULSET	α_2	α_3	α_5	α_6
0.006-0.03	23.0-36.0	0.1-12.0	0.5-1.0	0.1-10.0	0.0-1.0

The fitting results are shown in Table 16.

Table 16. MCMFIT of two site adsorption model for data on Tetrachloroethylene.

NON-LINEAR LEAST-SQUARES ANALYSIS
 Example 7.
 Fitting Tetrachloroethylene data using two site model with b. c. type 1.

INITIAL VALUES OF COEFFICIENTS

```
=====
NAME          INITIAL VALUE
vel.....    9.1000E-02
ds.....     1.0582E-02
pulset.....  2.6644E+01
alpha1..... -1.0000E+00
alpha2.....  2.1890E+00
alpha3.....  6.9375E-01
alpha4.....  1.0000E+00
alpha5.....  7.8713E+00
alpha6.....  8.0579E-01
```

ITERATION	SSQ	ds	pulset	alpha2	alpha3	alpha5	alpha6
0	2.5727E-02	1.0582E-02	2.6644E+01	2.1890E+00	6.9375E-01	7.8713E+00	8.0579E-01
1	2.4656E-02	1.0449E-02	2.6644E+01	2.1934E+00	6.9153E-01	3.0462E+00	6.8058E-01
2	1.6435E-02	1.0507E-02	2.6644E+01	2.2758E+00	6.6807E-01	5.7418E+00	7.9750E-01
3	1.6099E-02	1.0252E-02	2.6644E+01	2.2815E+00	6.6442E-01	8.9931E+00	9.2950E-01
4	1.6096E-02	1.0253E-02	2.6644E+01	2.2815E+00	6.6439E-01	9.9477E+00	9.3650E-01
5	1.6087E-02	1.0257E-02	2.6644E+01	2.2818E+00	6.6427E-01	9.9477E+00	9.7620E-01
6	1.6083E-02	1.0259E-02	2.6644E+01	2.2819E+00	6.6422E-01	9.9477E+00	9.9582E-01
7	1.6018E-02	1.0388E-02	2.6644E+01	2.2857E+00	6.6423E-01	9.9477E+00	9.9582E-01
8	1.5981E-02	1.0272E-02	2.6644E+01	2.2929E+00	6.6274E-01	9.9477E+00	9.9582E-01
9	1.5976E-02	1.0300E-02	2.6644E+01	2.2930E+00	6.6279E-01	9.9477E+00	9.9582E-01
10	1.5976E-02	1.0300E-02	2.6644E+01	2.2930E+00	6.6279E-01	9.9477E+00	9.9582E-01

CORRELATION MATRIX

```
=====
      1      2      3      4      5      6
1  1.0000
2  0.0000  0.0000
3 -0.1582  0.0000  1.0000
4 -0.0533  0.0000  0.6318  1.0000
5 -0.8936  0.0000  0.0659  0.1422  1.0000
6  0.8936  0.0000 -0.0659 -0.1422 -1.0000  1.0000
```

RSQUARE FOR REGRESSION =0.94505187
 NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS

```
=====
```

VARIABLE	NAME	VALUE	S.E. COEFF.	95% CONFIDENCE LIMITS	T-VALUE	LOWER	UPPER
1	ds	1.0300E-02	9.9975E-04	1.0303E+01	8.2967E-03	1.2304E-02	
2	pulset	2.6644E+01	1.7043E-32	1.5633E+33	2.6644E+01	2.6644E+01	
3	alpha2	2.2930E+00	3.2974E-02	6.9540E+01	2.2269E+00	2.3591E+00	
4	alpha3	6.6279E-01	2.2964E-02	2.8862E+01	6.1676E-01	7.0881E-01	
5	alpha5	9.9477E+00	8.8377E+01	1.1256E-01	-1.6717E+02	1.8707E+02	
6	alpha6	9.9582E-01	8.8470E+00	1.1256E-01	-1.6735E+01	1.8727E+01	

-----ORDERED BY COMPUTER INPUT-----

NO	T1	OBS	Y FITTED	RESI- DUAL
1	2.100E+01	5.200E-03	1.819E-12	5.200E-03
2	2.800E+01	1.870E-02	1.819E-12	1.870E-02
..
..
59	4.610E+02	1.820E-02	9.169E-03	9.031E-03
60	6.300E+02	2.000E-03	3.028E-03	-1.028E-03

-----ORDERED BY RESIDUAL-----

NO	T1	OBS	Y FITTED	RESI- DUAL
44	2.610E+02	1.640E-02	7.853E-02	-6.213E-02
41	2.390E+02	1.483E-01	1.089E-01	3.937E-02
..
..
27	1.490E+02	8.510E-02	8.526E-02	-1.650E-04
28	1.540E+02	1.262E-01	1.262E-01	-1.862E-06

It is evident from the parameter values that the model is almost an equilibrium model with the fraction of equilibrium sites very close to 1. Here, also, the model fit is quite good and the model is insensitive to pulse time because the time step chosen was too coarse (DT = 1 d). The breakthrough curve and the fitted curve are shown in Figure 5.

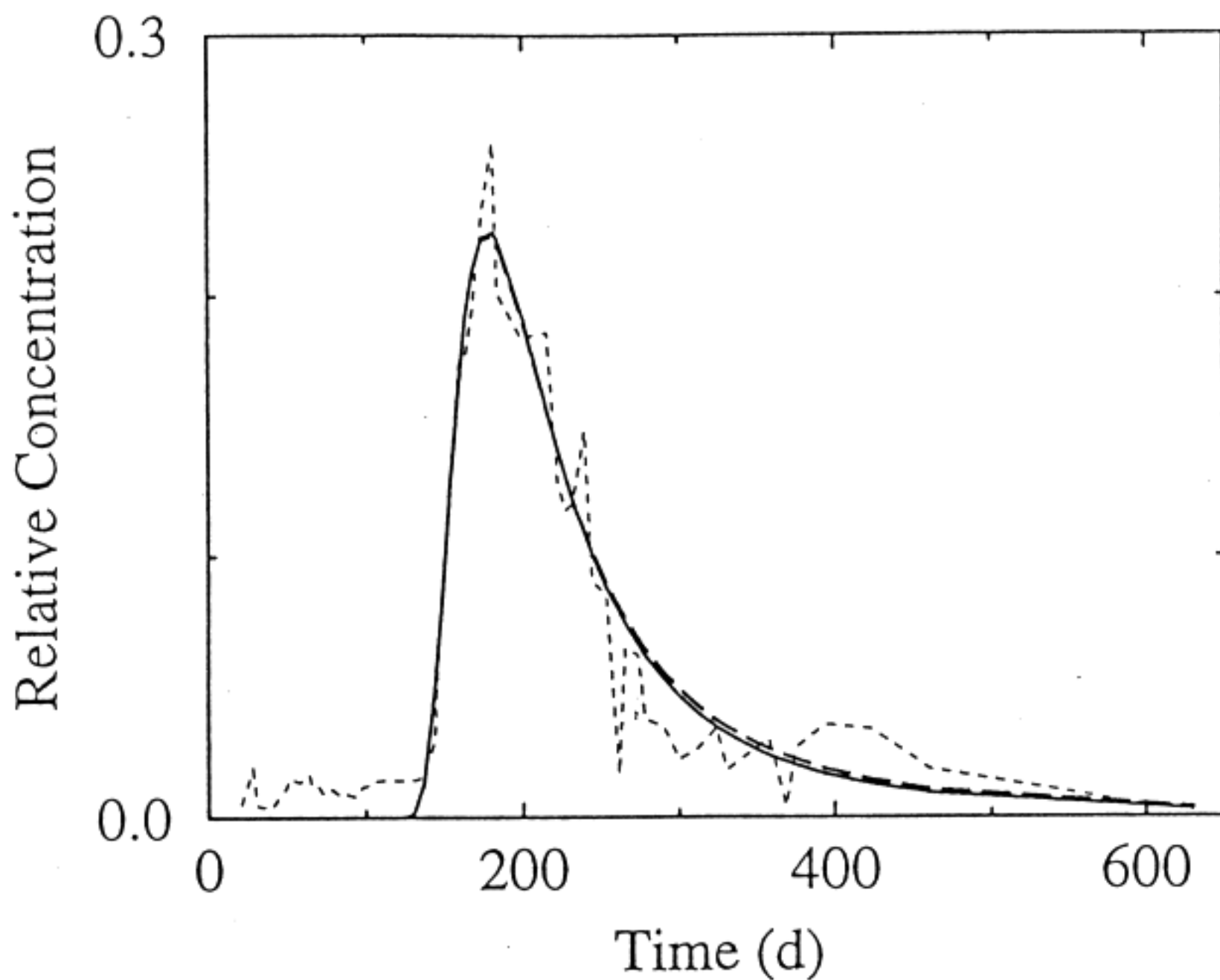


Figure 5. Field data on Tetrachloroethylene (short dashes) and fitted by MCMFIT using two site model (line) and fully kinetic model (long dashes).

Example 8.

To determine how well the fully kinetic model fits the data, another run was conducted with results as shown in Table 17. In this case α_6 was set at 0 and the range of α_5 used was 0.01 to 200.0 d^{-1} .

Table 17. Fully kinetic MCMFIT results for TCE data.

NON-LINEAR LEAST-SQUARES ANALYSIS
 Example 8.
 Fiting Tetrachloroethylene data using fully kinetic model with b. c. type 1.

INITIAL VALUES OF COEFFICIENTS

```

=====
NAME          INITIAL VALUE
vel.....    9.1000E-02
ds.....     1.2213E-02
pulset.....  2.7155E+01
alpha1..... -1.0000E+00
alpha2.....  1.9976E+00
alpha3.....  4.7769E-01
alpha4.....  1.0000E+00
alpha5.....  4.8712E+01
alpha6.....  0.0000E+00
  
```

ITERATION	SSQ	ds	pulset	alpha2	alpha3	alpha5
0	3.9573E-02	1.2213E-02	2.7155E+01	1.9976E+00	4.7769E-01	4.8712E+01
1	3.8500E-02	1.2134E-02	2.7155E+01	1.9994E+00	4.7916E-01	1.3959E+02
2	3.8451E-02	1.2132E-02	2.7155E+01	1.9996E+00	4.7925E-01	1.7254E+02
3	3.8427E-02	1.2130E-02	2.7155E+01	1.9996E+00	4.7929E-01	1.9635E+02
4	3.8425E-02	1.2130E-02	2.7155E+01	1.9996E+00	4.7929E-01	1.9869E+02
5	2.4581E-02	1.1404E-02	2.7155E+01	2.1443E+00	5.3632E-01	1.9869E+02
6	1.8327E-02	1.1269E-02	2.7155E+01	2.3177E+00	6.1930E-01	1.9869E+02
7	1.7923E-02	1.1498E-02	2.7155E+01	2.3047E+00	6.2842E-01	1.9869E+02
8	1.6878E-02	1.0912E-02	2.7155E+01	2.2933E+00	6.3234E-01	1.9869E+02
9	1.6813E-02	1.0700E-02	2.7155E+01	2.3017E+00	6.3384E-01	1.9869E+02
10	1.6803E-02	1.0797E-02	2.7155E+01	2.3043E+00	6.3603E-01	1.9869E+02

NO FURTHER DECREASE IN SSQ OBTAINED AFTER 20 TRIALS

CORRELATION MATRIX

	1	2	3	4	5
1	1.0000				
2	0.0000	0.0000			
3	-0.2201	0.0000	1.0000		
4	0.1617	0.0000	0.6239	1.0000	
5	0.0000	0.0000	0.0000	0.0000	1.0000

RSQUARE FOR REGRESSION =0.94220605

NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS

VARIABLE	NAME	VALUE	S.E. COEFF.	95% CONFIDENCE LIMITS		
				T-VALUE	LOWER	UPPER
1	ds	1.0797E-02	3.8843E-04	2.7796E+01	1.0019E-02	1.1575E-02
2	pulset	2.7155E+01	1.7322E-32	1.5676E+33	2.7155E+01	2.7155E+01
3	alpha2	2.3043E+00	3.3978E-02	6.7820E+01	2.2363E+00	2.3724E+00
4	alpha3	6.3603E-01	2.2144E-02	2.8722E+01	5.9167E-01	6.8040E-01
5	alpha5	1.9869E+02	1.3294E-01	1.4946E+03	1.9842E+02	1.9896E+02

-----ORDERED BY COMPUTER INPUT-----

NO	T1	OBS	Y FITTED	RESI- DUAL
1	2.100E+01	5.200E-03	4.547E-13	5.200E-03
2	2.800E+01	1.870E-02	4.547E-13	1.870E-02
..
..
59	4.610E+02	1.820E-02	1.079E-02	7.412E-03
60	6.300E+02	2.000E-03	3.847E-03	-1.847E-03

-----ORDERED BY RESIDUAL-----

NO	T1	OBS	Y FITTED	RESI- DUAL
44	2.610E+02	1.640E-02	8.072E-02	-6.432E-02
41	2.390E+02	1.483E-01	1.101E-01	3.815E-02
..
..
28	1.540E+02	1.262E-01	1.274E-01	-1.156E-03
27	1.490E+02	8.510E-02	8.579E-02	-6.904E-04

Observe that the kinetic rate coefficient is very high (198 d^{-1}). This implies that equilibrium model is possibly as appropriate. The Freundlich parameters for TCE as obtained from the fitting results in Tables 16 and 17 are both of almost the same magnitude. The values for K and N in the Freundlich equation $S = K C^N$, are 0.3 and 0.64 respectively. The values determined from independent batch experiments were, however, 0.93 and 0.79 respectively. The values of dispersion coefficient determined by two site and fully kinetic model is also almost the same as seen from Tables 16 and 17. The fitted curve is shown in Figure 5 (long dashes).

It is clear from the present analysis that the tailing effects are due to the nonlinear isotherm, even though an equilibrium model was used. The tailing effect increases with increasing sorption as is evident from Figures 4 and 5. The isotherm for CT is almost linear and hence exhibits little tailing whereas the isotherm of TCE is highly nonlinear and exhibits substantial tailing in the breakthrough curve. *Goltz and Roberts (1986)*, *Sposito and Jury (1988)* and *Barry et al. (1988)* used a linear isotherm to model the TCE breakthrough curve and concluded that the tailing effect was well described by the nonequilibrium model. It is not clear whether the tailing is caused by the nonlinear adsorption or nonequilibrium exchange. To answer this the key step is to identify the correct isotherm which describes the adsorption process. Both linear and Freundlich isotherms have been studied and both agree well with the observed change in liquid phase concentration in batch studies for the same aquifer material and same solute (*Curtis et al., 1986*) although the fitted TCE Freundlich isotherm parameters in Table 17 show more nonlinearity than the batch results. Hence, based on the fitted curves, it is not clear which is the dominant process. We point out that the batch determined adsorption coefficients may not be appropriate for the flow through systems. The issue of the applicability of batch data to sorption in flowing systems has been reported in a number of studies (e.g., *Akratnakul et al., 1983*; *Theis et al., 1988*; *Miller et al., 1989*).

Example 9.

The last example consists of data from a laboratory experiment on Na-Ca exchange during soil column miscible displacement studies (*Pandey et al., 1982*). The adsorption

isotherm assumed for this soil system is a Langmuir isotherm. The S-curve isotherm reduces to this form when $\alpha_3 = 1$ and $\alpha_4 = -1$. The pore water velocity and dispersion coefficients were fixed from the reported values of *Pandey et al.* (1982) and the two isotherm parameters were fitted using equilibrium model with entrance boundary condition type 1. The parameter 'RHO' was entered as that reported in the paper, a value of 2.68. The parameter ranges used for α_1 and α_2 were respectively -0.42 to -0.02 and -0.92 to -0.12. The results of fitting are shown in Table 18.

Table 18. MCMFIT of laboratory data on Na-Ca exchange.

```

NON-LINEAR LEAST-SQUARES ANALYSIS
Example 9.
Fitting Pandey's data using equilibrium model with b.c. type 1.

INITIAL VALUES OF COEFFICIENTS
=====
NAME          INITIAL VALUE
vel.....    3.5950E+01
ds.....     3.3270E+01
pulset.....  9.9230E+03
alpha1.....  -2.5334E-01
alpha2.....  -7.6779E-01
alpha3.....  1.0000E+00
alpha4.....  -1.0000E+00
alpha5.....  0.0000E+00
alpha6.....  1.0000E+00

ITERATION      SSQ          alpha1      alpha2
0             1.0005E-02    -2.5334E-01 -7.6779E-01
1             4.5657E-03    -1.4021E-01 -8.5435E-01
2             1.6736E-03    -1.4900E-01 -8.6402E-01
3             1.6723E-03    -1.4872E-01 -8.6489E-01
4             1.6723E-03    -1.4872E-01 -8.6489E-01

CORRELATION MATRIX
=====
          1          2
1     1.0000
2    -0.9623     1.0000

RSQUARE FOR REGRESSION =0.99763814
NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS
=====
VARIABLE  NAME      VALUE      S.E. COEFF.  95% CONFIDENCE LIMITS
          alpha1  -1.4872E-01  1.5853E-02  T-VALUE      LOWER      UPPER
          alpha2  -8.6489E-01  2.2375E-02  -9.3812E+00  -1.8528E-01 -1.1217E-01
          alpha2  -8.6489E-01  2.2375E-02  -3.8655E+01  -9.1648E-01 -8.1329E-01

-----ORDERED BY COMPUTER INPUT-----
NO      X1          OBS          Y          RESI-
          X1          OBS          FITTED      DUAL
1      1.241E+00  9.793E-01  9.846E-01  -5.297E-03
2      3.793E+00  9.379E-01  9.407E-01  -2.832E-03
3      6.414E+00  8.690E-01  8.867E-01  -1.766E-02
4      8.759E+00  8.429E-01  8.376E-01  5.328E-03
5      1.138E+01  7.857E-01  7.826E-01  3.081E-03
6      1.387E+01  7.429E-01  7.301E-01  1.279E-02
7      1.648E+01  7.000E-01  6.740E-01  2.601E-02
8      1.869E+01  6.143E-01  6.256E-01  -1.128E-02
9      2.057E+01  5.655E-01  5.834E-01  -1.787E-02

-----ORDERED BY RESIDUAL-----
NO      X1          OBS          Y          RESI-
          X1          OBS          FITTED      DUAL
7      1.648E+01  7.000E-01  6.740E-01  2.601E-02
9      2.057E+01  5.655E-01  5.834E-01  -1.787E-02
3      6.414E+00  8.690E-01  8.867E-01  -1.766E-02
6      1.387E+01  7.429E-01  7.301E-01  1.279E-02
8      1.869E+01  6.143E-01  6.256E-01  -1.128E-02
4      8.759E+00  8.429E-01  8.376E-01  5.328E-03
1      1.241E+00  9.793E-01  9.846E-01  -5.297E-03
5      1.138E+01  7.857E-01  7.826E-01  3.081E-03
2      3.793E+00  9.379E-01  9.407E-01  -2.832E-03

```

The fitted isotherm parameters are close to the reported batch values ($\alpha_1 = -0.22$ and $\alpha_2 = -0.82$) and the fit is better than that reported by *Pandey et al.*, 1982. The data and the fitted results are shown in Figure 6.

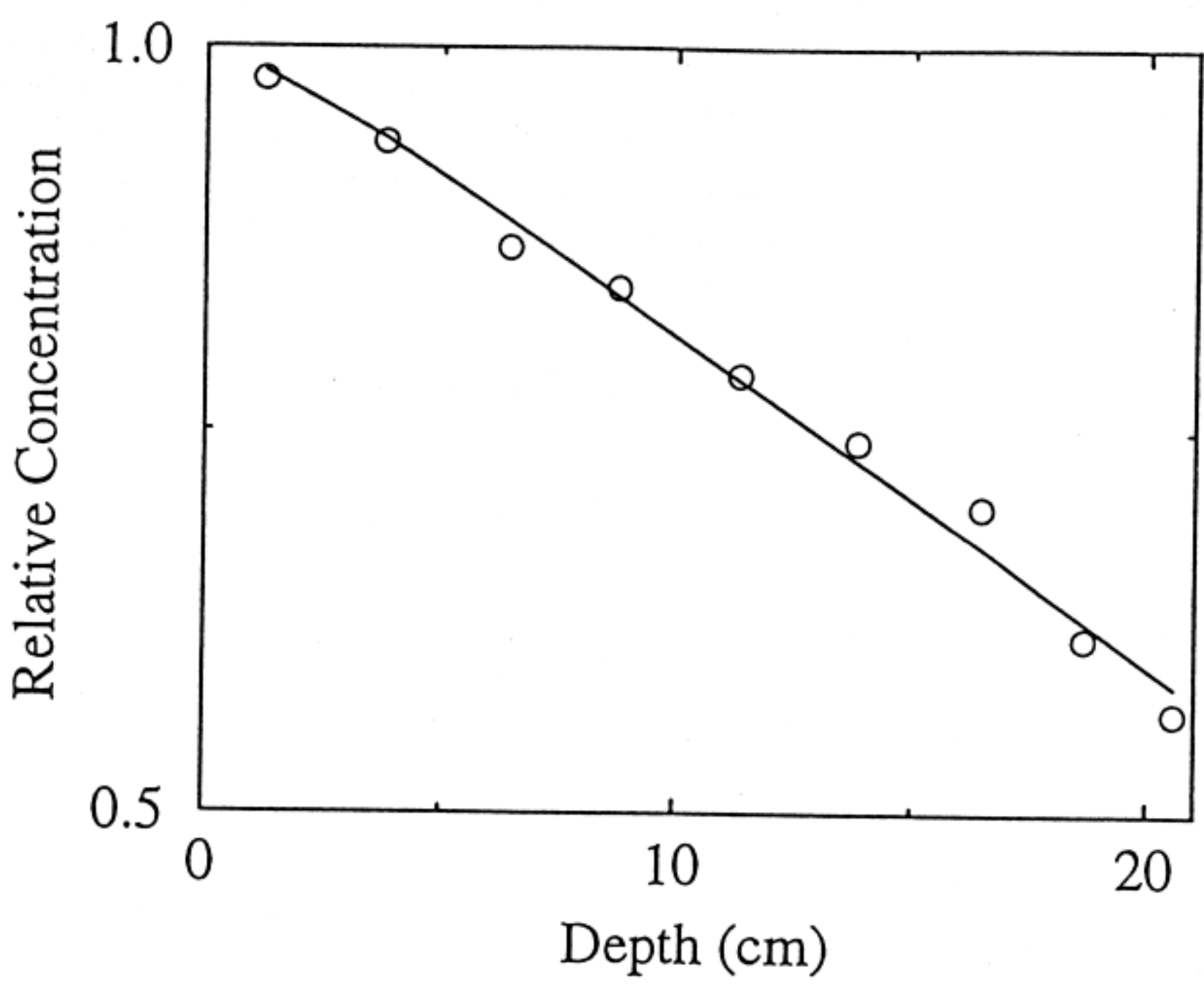


Figure 6. Laboratory data (circles) and MCMFIT fit (line) for Na-Ca exchange.

REFERENCES

- Akratanakul, S., L. Boersma, and G. O. Klock, 1983. Sorption processes in soils as influenced by pore water velocity: 2. Experimental results. *Soil Sci.*, **135**, 331-341, 1983.
- Appelo, C. A. J., and A. Willemssen, Geochemical calculations and observations on salt water intrusions, I. A combined geochemical/mixing cell model, *J. Hydrol.*, **94**, 313-330, 1987.
- Bajracharya, K., and D. A. Barry, Mixing cell models for nonlinear equilibrium single species adsorption and transport, *J. Contam. Hydrol.*, **12**, 227-243, 1993a.
- Bajracharya, K., and D. A. Barry, Mixing cell models for nonlinear nonequilibrium single species adsorption and transport, *Water Resour. Res.*, **29**, 1405-1414, 1993b.
- Barry, D. A., Modelling contaminant transport in the subsurface: Theory and computer programs, in *Modelling Chemical Transport in Soil: Natural and Applied Contaminants*, H. Ghadiri and C. W. Rose (eds.), Lewis Publishers, Chelsea, Michigan, pp. 105-144, 1992.
- Barry, D. A., and K. Bajracharya, Rapid computation of solute transport with nonlinear, nonequilibrium reactions. Preprints of papers of the Hydrology and Water Resources Conference, Newcastle, N.S.W., The Institution of Engineers, Australia, 30 June - 2 July, 1993.
- Barry, D. A., and G. Sposito, Application of the convection-dispersion model to solute transport in finite soil columns, *Soil Sci. Soc. Am. J.*, **52**, 3-9, 1988.
- Barry, D. A., G. Sposito, and J. Coves, *TFMFIT: A program to fit a generalized convection-dispersion model to experimental data*, Department of Soil and Environmental Sciences, University of California, Riverside, 1988.
- Beven, K. J., D. E. Henderson, and A. D. Reeves, Dispersion parameters for undisturbed partially saturated soil, *J. Hydrol.*, **143**, 19-43, 1993.
- Beveridge, G. S. G., and R. S. Schechter, *Optimization: Theory and practice*, 773 pp., McGraw-Hill Book Company, 1970.
- Brenner, H., The diffusion model of longitudinal mixing in beds of finite length. Numerical values, *Chem. Eng. Sci.*, **17**, 229-243, 1962.
- Cameron, D. A., and A. Klute, Convective-dispersive solute transport with a combined equilibrium and kinetic adsorption model, *Water Resour. Res.*, **13**, 183-188, 1977.
- Curtis, G. P., P. V. Roberts, and M. Reinhard, 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**, 2059-2067, 1986.
- Dance, J. T., and E. J. Reardon, Migration of contaminants in groundwater at a landfill: A case study, 5. Cation migration in the dispersion test, *J. Hydrol.*, **63**, 109-130, 1983.
- Dudley, L. M., J. E. Mclean, T. H. Furst, and J. J. Jurinak, Sorption of cadmium and copper from an acid mine waste extract by two calcareous soils: Column studies, *Soil Sci.*, **151**, 121-135, 1991.

- Freyberg, D. L., 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, 1986.
- Gerald, C. F., *Applied Numerical Analysis*, 518 pp., Addison-Wesley Publishing Company, 1978.
- Goltz, M. N., and P. V. Roberts, Interpreting organic solute transport data from a field experiment using physical nonequilibrium models, *J. Contam. Hydrol.*, **1**, 77-93, 1986.
- Hendrickson, J. D., S. Sorooshian, and L. E. Brazil, Comparison of Newton-type and direct search algorithms for calibration of conceptual rainfall-runoff models, *Water Resour. Res.*, **24**, 691-700, 1988.
- Kreft, A., and A. Zuber, On the physical meaning of the dispersion equation and its solutions for different initial and boundary conditions, *Chem. Eng. Sci.*, **33**, 1471-1480, 1978.
- Mackay, D. M., D. L. Freyberg, P. V. Roberts, and J. A. Cherry, A natural gradient experiment on solute transport in a sand aquifer, 1. Approach and overview of plume movement, *Water Resour. Res.*, **22**, 2017-2029, 1986.
- Marquardt, D. W., An algorithm for least-squares estimation of nonlinear parameters, *J. Soc. Ind. Appl. Math.*, **11**, 431-441, 1963.
- Miller, C. T., M. E. Summer, and W. P. Miller, A comparison of batch and flow generated anion adsorption isotherms, *Soil Sci. Soc. Am. J.*, **53**, 373-380, 1989.
- Noye, J., Finite difference methods for partial differential equations, In J. Noye (ed.), *Numerical Solutions of Partial Differential Equations*, 647 pp., North-Holland Publishing Company, Amsterdam, Netherlands, 1982.
- Parker, J. C., and M. Th. van Genuchten, *Determining Transport Parameters from Laboratory and Field Tracer Experiments*, 96 pp., Bull. 84-3, Virginia Polytechnic Institute and State Univ., Blacksburg, Virginia, 1984.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran, The Art of Scientific Computing*, 967 pp., Cambridge University Press, 1992.
- Rao, B. K., and D. L. Hathaway, Three-dimensional mixing cell solute transport model and its application, *Ground Water*, **27**, 509-516, 1989.
- Rao, P. S. C., J. M. Davidson, R. E. Jessup, and H. M. Selim, Evaluation of conceptual models for describing nonequilibrium adsorption-desorption of pesticides during steady flow in soils, *Soil Sci. Soc. Am. J.*, **43**, 22-28, 1979.
- Roberts, P. V., and D. M. Mackay, editors, A natural gradient experiment on solute transport in a sand aquifer. Technical Report No. 292, Department of Civil Engineering, Stanford University, Stanford, CA 94305, 1986.
- Rubin, J., Transport of reacting solutes in porous media: Relation between mathematical nature of problem formulation and chemical nature of reactions, *Water Resour. Res.*, **19**, 1231-1252, 1983.
- Sardin, M., D. Schweich, F. J. Leij, and M. Th. van Genuchten, Modelling the nonequilibrium transport of linearly interacting solutes in porous media: A review,

- Schulz, H. D., and E. J. Reardon, A combined mixing cell/analytical model to describe two-dimensional reactive solute transport for unidirectional groundwater flow, *Water Resour. Res.*, **19**, 493-502, 1983.
- Schweich, D., and M. Sardin, Adsorption, partition, ion exchange and chemical reaction in batch reactors or in columns, *J. Hydrol.*, **50**, 1-33, 1981.
- Schweich, D., M. Sardin, and J. P. Gaudet, Measurement of a cation exchange isotherm from elution curves obtained in a soil column: Preliminary results, *Soil Sci. Soc. Am. J.*, **47**, 32-37, 1983.
- Selim, H. M., J. M. Davidson, and R. S. Mansell, Evaluation of a two site adsorption-desorption model for describing solute transport in soils, in *Proceedings of the 1976 Summer Computer Simulation Conference*, July 1976, La Jolla, Calif., Simulation Councils, pp. 444-448, 1976.
- Sposito, G., *The Chemistry of Soils*, 277 pp., Oxford Univ. Press, New York, 1989.
- Sposito, G., and W. A. Jury, The lifetime probability density function for solute movement in the subsurface zone. *J. Hydrol.*, **102**, 503-518, 1988.
- Sudicky, E. A., 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, 1986.
- Theis, L. T., R. Iyer, and L. W. Kaul, 1988. Kinetic studies of Cadmium and Ferricyanide adsorption on goethite, *Environ. Sci. Technol.*, **22**, 1013-1017, 1988.
- van Genuchten, M. Th., and J. C. Parker, Boundary conditions for displacement experiments through short laboratory soil columns, *Soil Sci. Soc. Am. J.*, **48**, 703-708, 1984.
- van Genuchten, M. Th., *Determining Transport Parameters from Solute Displacement Experiments*, Research Report No. 118, U. S. Salinity Laboratory, Riverside, California, 37 pp., 1980.
- van Genuchten, M. Th., *Non-Equilibrium Transport Parameters from Miscible Displacement Experiments*, Research Report No. 119, U. S. Salinity Laboratory, Riverside, California, 88 pp., 1981.
- Van Ommen, H. C., The "mixing-cell" concept applied to transport of non-reactive and reactive components in soils and groundwater, *J. Hydrol.*, **78**, 201-213, 1985.

LISTING OF THE MCMFIT PROGRAM

```

PROGRAM MCMFIT
  DIMENSION Y(200),F(200),R(200),DELZ(200,14),TH(28),
1B(28),E(14),P(14),
1PHI(14),Q(14),LSORT(200),TB(28),A(14,14),C(14),D(14,14),
2C10(0:2000),S10(0:2000),CMIN(14),CMAX(14)
  COMMON/MODAT1/X(3,200),INDEX(14),NOB,NVAR
  COMMON/MODAT2/CLEN,C01,C10,S10,DELT,RHO,TIMET
  COMMON/MODAT3/CMIN,CMAX
  COMMON/MODAT4/DFT,MODE
  COMMON/MODAT5/DT
  COMMON/MODAT7/BETA
  CHARACTER TITLE*80, BI(28)*4
  CHARACTER*25 FNAME
  INTEGER*4 HR, HM, SEC
  DATA STOPCR/1.E-4/

C
C SET UP I/O FILES
C
C NOTE: PLOT OUTPUT FILE CONTAINS FOR I=1,NOB
C       Y(I),X(1,I),I IN (4E15.4,I3) FORMAT
C       FOLLOWED BY
C       F(I),X(1,I),I IN (4E15.4,I3) FORMAT
C       WHERE
C       Y ARE INPUT CONCENTRATION VALUES
C       F ARE PREDICTED CONCENTRATION VALUES
C       X'S ARE INDEPENDENT TIME OR DISTANCE VALUES
C
WRITE(*, 2000)
READ(*, 2001)FNAME
IF (FNAME.EQ.' ') FNAME = 'MCMFIT.DAT'
OPEN(1,FILE=FNAME,STATUS='OLD')
WRITE(*, 2002)
READ(*, 2001)FNAME
IF (FNAME.EQ.' ') FNAME = 'MCMFIT.OUT'
OPEN(2,FILE=FNAME,STATUS='UNKNOWN')

C
C INITIAL CONCENTRATIONS
C CHANGE HERE FOR NON-ZERO INITIAL CONCENTRATIONS
C
DO 132 I1=0,2000
  C10(I1)=0.
132 S10(I1)=0.

C
C READ INPUT PARAMETERS
C
READ(1,2006,END=999)NVAR,NOB,MIT,MAXTRY,I PLOT,MODE,NRAN
READ(1,1050,END=999)CLEN,C01,DT,RHO,TIMET
IF(I PLOT.NE.0) THEN
  WRITE(*, 2003)
  READ(*, 2001)FNAME
  IF (FNAME.EQ.' ') FNAME = 'MCMFIT.PLT'
  OPEN(7,FILE=FNAME,STATUS='UNKNOWN')
ENDIF

C
C NVAR = NUMBER OF COEFFICIENTS (14 MAX)
C NOB = NUMBER OF OBSERVATIONS
C MIT = MAX NUMBER OF ITERATIONS (MIT=0 SOLVES DIRECT PROBLEM ONLY)
C MAXTRY=MAX TRIAL SOLUTIONS AT ITERATION BEFORE QUITTING
C SUGGEST 10 TO 50, SMALLER VALUES REDUCE RUN TIME
C BUT MAY MISS CONVERGENT SOLUTION
C I PLOT= CODE FOR PLOT FILE OUTPUT
C 0=NO PLOT FILE; 1=PLOT FILE IS OUTPUT
C
IF (MAXTRY.EQ.0)MAXTRY=25
IF(MIT.NE.0) WRITE(2,1035)
DO 4 I=1,2
  READ(1,1001) TITLE
4 WRITE(2,1002) TITLE
  NU1=NVAR+1
  NU2=2*NVAR
DO 5 N=1,2
  IF(N.EQ.2.AND.NVAR.LE.7)GO TO 5

C
C READ COEFFICIENTS NAMES
C
  N1=14*N-13
  N2=MIN0(14*N,NU2)
  READ(1,1004) (BI(I),I=N1,N2)

C
C READ INDICES
C
  N1=7*N-6
  N2=MIN0(7*N,NVAR)
  READ(1,1006) (INDEX(I),I=N1,N2)

C
C READ CONSTRAINTS ON PARAMETER VALUES
C CMIN AND CMAX ARE MIN AND MAX CONSTRAINTS
C (CMIN(I)=CMAX(I)=0 WILL OMIT CONSTRAINT ON PARAMETER I)

```



```

      READ(1,1005) (CMIN(I),I=N1,N2)
      READ(1,1005) (CMAX(I),I=N1,N2)
5 CONTINUE
C
C   READ EXPERIMENTAL DATA
C
      DO 10 I=1,NOB
10  READ(1,1005) Y(I),X(1,I)
      Y(NOBI+1)=0.
      X(1,NOBI+1)=-1.
      NOBI=NOBI+1
      IF(MIT.EQ.0) GO TO 18
C
C   OBTAIN MAXIMUM AND MINIMUM DATA POINTS
C
      DMAX=Y(1)
      DMIN=Y(1)
      DO 15 I=1,NOB
          IF (DMAX.LT.Y(I+1)) DMAX=Y(I+1)
          IF (DMIN.GT.Y(I+1)) DMIN=Y(I+1)
15  CONTINUE
      BETA=DMAX-DMIN
C
C   RANDOM GENERATION OF PARAMETERS
C   ITIME IS USED FOR SEEDING RANDOM GENERATOR, RAN3 (ANOTHER SEED COULD BE USED)
C
      CALL ITIME(HR,HM,SEC)
      IDUM=HR*3600+HM*60+SEC
18  OLDSUM=1.E35
      DO 320 II=1,NRAN
          DO 320 I=1,NVAR
              S=RAN3(IDUM)
              C(I)=S
320  C(I)=(CMAX(I)-CMIN(I))*C(I)+CMIN(I)
C
C   SOLVE THE TRANSPORT EQUATION
C
      X(1,0)=0.
      FPULSE=0.
      NCONDI=0
      DFT=0.
      IF (MODE.LE.2) THEN
          DO 330 J=1,NOB-1
              DELT=X(1,J)-X(1,J-1)-DFT
              IF (MODE.EQ.1) THEN
                  CALL FUNC1(F(J),X(1,J),C,NCONDI,FPULSE)
              ELSEIF (MODE.EQ.2) THEN
                  CALL FUNC2(F(J),X(1,J),C,NCONDI,FPULSE)
              ENDIF
330  CONTINUE
          ELSEIF (MODE.GT.2) THEN
              IF (MODE.EQ.3) THEN
                  CALL FUNC3(F,X,C,NCONDI,FPULSE)
              ELSEIF (MODE.EQ.4) THEN
                  CALL FUNC4(F,X,C,NCONDI,FPULSE)
              ENDIF
          ENDIF
C
C   CALCULATING SSQ
C
332  SUMB=0.
      DO 340 I=1,NOB-1
          R(I)=Y(I)-F(I)
340  SUMB=SUMB+R(I)*R(I)
      IF (SUMB.LT.OLDSUM) THEN
          OLDSUM=SUMB
          DO 350 J=NU1,NU2
350  B(J)=C(J-NVAR)
          IF(OLDSUM.LT.1.E-3) GOTO 370
      ENDIF
360 CONTINUE
370 WRITE(2,1007)
      DO 7 I=1,NVAR
          J=2*I-1
          7 WRITE(2,1008) BI(J),BI(J+1),B(I+NVAR)
C
C   REARRANGE VARIABLE ARRAYS
C
      NP=0
      DO 20 I=NU1,NU2
          TB(I)=B(I)
          IF(INDEX(I-NVAR).EQ.0) GO TO 20
          NP=NP+1
          K=2*NP-1
          J=2*(I-NVAR)-1
          BI(K)=BI(J)
          BI(K+1)=BI(J+1)
          B(NP)=B(I)
          TB(NP)=B(I)
          TH(NP)=B(NP)
20  TH(I)=B(I)
C
C   COMPUTATION STARTS
C
      GA=.02
      NIT=0
      NP2=2*NP
      CALL MODEL(TH,F)
      IF (MIT.EQ.0) GO TO 140

```

```

SSQ=0.
DO 32 I=1,NOB
  R(I)=Y(I)-F(I)
32 SSQ=SSQ+R(I)*R(I)
  WRITE(2,1011) (BI(J),BI(J+1),J=1,NP2,2)
  WRITE(2,1012) NIT,SSQ,(B(I),I=1,NP)
C
C
C
  BEGIN ITERATION
34 NIT=NIT+1
  NTRIAL=0
  GA=.1*GA
  DO 38 J=1,NP
    TEMP=TH(J)
    TH(J)=1.01*TH(J)
    Q(J)=0.
    CALL MODEL(TH,DELZ(1,J))
    DO 36 I=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)=TEMP
  DO 44 I=1,NP
    DO 42 J=1,I
      SUM=0.
      DO 40 K=1,NOB
40        SUM=SUM+DELZ(K,I)*DELZ(K,J)
      D(I,J)=10000.*SUM/(TH(I)*TH(J))
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 DO 52 I=1,NP
  DO 52 J=1,NP
52 A(I,J)=D(I,J)/(E(I)*E(J))
C
C
C
  A IS THE SCALED MOMENT MATRIX
  DO 54 I=1,NP
    P(I)=Q(I)/E(I)
    PHI(I)=P(I)
54 A(I,I)=A(I,I)+GA
  CALL MATINV(A,NP,P)
C
C
C
  P/E IS THE CORRECTION VECTOR
  STEP=1.
56 DO 58 I=1,NP
  TB(I)=P(I)*STEP/E(I)+TH(I)
58 CONTINUE
60 DO 62 I=1,NP
  IF(TH(I)*TB(I))66,66,62
62 CONTINUE
  SUMB=0.
  CALL MODEL(TB,F)
  DO 64 I=1,NOB
    R(I)=Y(I)-F(I)
64 SUMB=SUMB+R(I)*R(I)
66 SUM1=0.
  SUM2=0.
  SUM3=0.
  DO 68 I=1,NP
    SUM1=SUM1+P(I)*PHI(I)
    SUM2=SUM2+P(I)*P(I)
68 SUM3=SUM3+PHI(I)*PHI(I)
  ARG=0.
  IF(SUM2*SUM3.NE.0.) ARG=SUM1/SQRT(SUM2*SUM3)
  ARG1=0.
  IF(NP.GT.1) ARG1=SQRT(1.-ARG*ARG)
  ANGLE=57.29578*ATAN2(ARG1,ARG)
  DO 72 I=1,NP
    IF(TH(I)*TB(I))74,74,72
72 CONTINUE
  NTRIAL=NTRIAL+1
  IF(NTRIAL.GT.MAXTRY) GO TO 95
  IF(SUMB/SSQ-1.) 80,80,74
74 IF(ANGLE-30.)76,76,78
76 STEP=.5*STEP
  GO TO 56
78 GA=10.*GA
  GO TO 50
C
C
C
  PRINT COEFFICIENTS AFTER EACH ITERATION
80 CONTINUE
  DO 82 I=1,NP
82 TH(I)=TB(I)
  WRITE(2,1012) NIT,SUMB,(TH(I),I=1,NP)
  DO 86 I=1,NP
    IF(ABS(P(I)*STEP/E(I))/(1.0D-20+ABS(TH(I))))-STOPCR) 86,86,94
86 CONTINUE
  GO TO 96
94 SSQ=SUMB
  IF(NIT.LT.MIT) GO TO 34
  IF(NIT.EQ.MIT) WRITE(2,1034) MIT
  GO TO 96
95 WRITE(2,1038) MAXTRY
C
C
C
  END OF ITERATION LOOP

```

```

96 CONTINUE
CALL MATINV(D,NP,P)
C
C
C WRITE CORRELATION MATRIX
DO 98 I=1,NP
E(I)=SQRT(D(I,I))
98 IF(E(I).EQ.0.) E(I)=1.E-30
IF(NP.EQ.1) GO TO 104
WRITE(2,1013) (I,I=1,NP)
DO 102 I=1,NP
DO 100 J=1,I
100 A(J,I)=D(J,I)/(E(I)*E(J))
102 WRITE(2,1014) I,(A(J,I),J=1,I)
C
C
C CALCULATE REGRESSION COEFFICIENT
104 SUMC=0.
SUMC2=0.
DO 106 I=1,NOB
SUMC=SUMC+Y(I)
106 SUMC2=SUMC2+Y(I)*Y(I)
RSQ=1.-SUMB/(SUMC2-SUMC*SUMC/NOB)
WRITE(2,1041) RSQ
C
C
C CALCULATE 95% CONFIDENCE INTERVAL
Z=1./FLOAT(NOB-NP)
SDEV=SQRT(Z*SUMB)
TVAR=1.96+Z*(2.3779+Z*(2.7135+Z*(3.187936+2.466666*Z**2)))
IF(NP.EQ.1)WRITE(2,1042)
IF(NP.GT.1)WRITE(2,1015)
DO 108 I=1,NP
SECOEF=E(I)*SDEV
TVALUE=TH(I)/SECOEF
TSEC=TVAR*SECOEF
TMCOE=TH(I)-TSEC
TPCOE=TH(I)+TSEC
J=2*I-1
IF(NP.EQ.1)
1 WRITE(2,1043) I,BI(J),BI(J+1),TH(I),SECOEF,TMCOE,TPCOE
108 IF(NP.GT.1)
1 WRITE(2,1016) I,BI(J),BI(J+1),TH(I),SECOEF,TVALUE,TMCOE,TPCOE
C
C
C PREPARE FINAL OUTPUT
110 LSORT(1)=1
DO 115 J=2,NOB
TEMP=ABS(R(J))
K=J-1
DO 111 L=1,K
LL=LSORT(L)
IF(TEMP-ABS(R(LL))) 112,112,111
111 CONTINUE
LSORT(J)=J
GO TO 115
112 KK=J
113 KK=KK-1
LSORT(KK+1)=LSORT(KK)
IF(KK-L) 114,114,113
114 LSORT(L)=J
115 CONTINUE
IF (MODE.GT.2) THEN
WRITE(2,1017)
ELSE
WRITE(2,1021)
ENDIF
IF(IPLOT.NE.0)WRITE(7,*) ' OBSERVED DATA '
DO 116 I=1,NOB-1
IF(IPLOT.NE.0)WRITE(7,1047) Y(I),X(1,I),I
116 WRITE(2,1018) I,X(1,I),Y(I),F(I),R(I)
IF(IPLOT.NE.0)WRITE(7,*) ' FITTED RESULT '
DO 117 I=1,NOB-1
IF(IPLOT.NE.0)WRITE(7,1047) F(I),X(1,I),I
117 CONTINUE
IF (MODE.GT.2) THEN
WRITE(2,1019)
ELSE
WRITE(2,1022)
ENDIF
DO 118 I=1,NOB-1
J=LSORT(NOB+1-I)
118 WRITE(2,1018) J,X(1,J),Y(J),F(J),R(J)
GO TO 150
140 WRITE(2,1030) (J,J=1,NX)
DO 145 I=1,NOB-1
145 WRITE(2,1033)X(1,I),F(I),I
150 CONTINUE
C
C
C END OF PROBLEM
999 STOP
1001 FORMAT(A80)
1002 FORMAT(11X,80A)
1004 FORMAT(7(2X,2A4))
1005 FORMAT(7F10.4)
1006 FORMAT(7I10)
1007 FORMAT(//11X,'INITIAL VALUES OF COEFFICIENTS'/11X,30(1H=)/12X,
1'NAME',11X,'INITIAL VALUE')

```



```

1008 FORMAT(11X,2A4,4(1H.),1PE12.4)
1009 FORMAT(/11X,'OBSERVED DATA',/11X,13(1H=)/11X,'OBS. NO.',5X,
1 ' Y ',9X,' X1 ',9X,' X2 ',14X,'X3')
1010 FORMAT(11X,I5,6X,1PE12.4,3(4X,1PE12.4))
1011 FORMAT(/11X,'ITERATION',6X,'SSQ',4X,5(5X,2A4))
1012 FORMAT(11X,I5,3X,1PE13.4,2X,5(1PE13.4))
1013 FORMAT(///,11X,'CORRELATION MATRIX'/11X,18(1H=)/14X,10(4X,I2,5X))
1014 FORMAT(11X,I3,10(2X,F7.4,2X))
1015 FORMAT(11X,'NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS',
1/11X,48(1H=)//56X,'95% CONFIDENCE LIMITS'/11X,'VARIABLE',3X,'NAME'
2,5X,'VALUE',8X,'S.E. COEFF.',4X,'T-VALUE',6X,'LOWER',8X,'UPPER')
1016 FORMAT(14X,I2,4X,2A4,1X,1PE11.4,2X,1PE11.4,2X,1PE11.4,2X,1PE11.4,
12X,1PE11.4)
1017 FORMAT(/11X,23(1H-),'ORDERED BY COMPUTER INPUT',24(1H-)/
137X,' Y ',10X,'RESI-'/11X,'NO',4X,' X1 ',
210X,'OBS',9X,'FITTED',9X,'DUAL')
1018 FORMAT(10X,I3,4(2X,1PE12.3))
1019 FORMAT(/11X,27(1H-),'ORDERED BY RESIDUAL',28(1H-)/
137X,' Y ',10X,'RESI-'/11X,'NO',4X,' X1 ',
210X,'OBS',9X,'FITTED',9X,'DUAL')
1020 FORMAT(///11X,'END OF PROBLEM'/11X,14(1H=))
1021 FORMAT(/11X,23(1H-),'ORDERED BY COMPUTER INPUT',24(1H-)/
137X,' Y ',10X,'RESI-'/11X,'NO',4X,' T1 ',
210X,'OBS',9X,'FITTED',9X,'DUAL')
1022 FORMAT(/11X,27(1H-),'ORDERED BY RESIDUAL',28(1H-)/
137X,' Y ',10X,'RESI-'/11X,'NO',4X,' T1 ',
210X,'OBS',9X,'FITTED',9X,'DUAL')
1030 FORMAT(/11X,9(1H-),'RESULTS FOR INITIAL COEFFICIENT VALUES',9(1H-
1)/4X,' X1 ',2X,' Y ',8X,' NO.')
1033 FORMAT(2(3X,1PE12.4),5X,I3)
1034 FORMAT(/11X,'CONVERGENCE CRITERIA NOT MET IN',I3,' ITERATIONS')
1035 FORMAT(21X,'NON-LINEAR LEAST-SQUARES ANALYSIS')
1038 FORMAT(/11X,'NO FURTHER DECREASE IN SSQ OBTAINED AFTER ',I2,' TRIA
1LS')
1041 FORMAT(/11X,'RSQUARE FOR REGRESSION =',F10.8)
1042 FORMAT(11X,'NON-LINEAR LEAST SQUARES ANALYSIS, FINAL RESULTS',
1/11X,48(1H=)//56X,'95% CONFIDENCE LIMITS'/11X,'VARIABLE',5X,'NAME'
2,5X,'VALUE',6X,'S.E. COEFF.',6X,'LOWER',8X,'UPPER')
1043 FORMAT(14X,I2,6X,2A4,1X,1PE11.4,2X,1PE11.4,2X,1PE11.4,2X,
11PE11.4)
1047 FORMAT(2(1PE15.4),2X,I3)
1050 FORMAT(/5E10.4)
2000 FORMAT(' ENTER NAME OF INPUT DATA FILE.',
1 ' (DEFAULT = MCMFIT.DAT)')
2001 FORMAT(A25)
2002 FORMAT(' ENTER NAME OF PRIMARY OUTPUT FILE.',
1 ' (DEFAULT = MCMFIT.OUT)')
2003 FORMAT(' ENTER NAME OF PLOT OUTPUT FILE IF USED.',
$ ' (DEFAULT = MCMFIT.PLT)')
2006 FORMAT(/7I10)
END

```

C

```
SUBROUTINE MATINV(A,NP,B)
```

C

```
PURPOSE: PERFORM MATRIX INVERSION FOR PARAMETER ESTIMATION
```

C

```
DIMENSION A(14,14),B(28),INDEX(14,2)
```

```
DO 2 J=1,14
```

```
2 INDEX(J,1)=0
```

```
I=0
```

```
4 AMAX=-1.
```

```
DO 10 J=1,NP
```

```
IF(INDEX(J,1)) 10,6,10
```

```
6 DO 10 K=1,NP
```

```
IF(INDEX(K,1)) 10,8,10
```

```
8 P=ABS(A(J,K))
```

```
IF(P.LE.AMAX) GO TO 10
```

```
IR=J
```

```
IC=K
```

```
AMAX=P
```

```
10 CONTINUE
```

```
IF(AMAX) 30,30,14
```

```
14 INDEX(IC,1)=IR
```

```
IF(IR.EQ.IC) GO TO 18
```

```
DO 16 L=1,NP
```

```
P=A(IR,L)
```

```
A(IR,L)=A(IC,L)
```

```
16 A(IC,L)=P
```

```
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)=1.0
```

```
DO 20 L=1,NP
```

```
20 A(IC,L)=A(IC,L)*P
```

```
B(IC)=B(IC)*P
```

```
DO 24 K=1,NP
```

```
IF(K.EQ.IC) GO TO 24
```

```
P=A(K,IC)
```

```
A(K,IC)=0.
```

```
DO 22 L=1,NP
```

```
22 A(K,L)=A(K,L)-A(IC,L)*P
```

```
B(K)=B(K)-B(IC)*P
```

```
24 CONTINUE
```

```
GO TO 4
```

```
26 IC=INDEX(I,2)
```

```
IR=INDEX(IC,1)
```

```

DO 28 K=1,NP
  P=A(K,IR)
  A(K,IR)=A(K,IC)
28 A(K,IC)=P
  I=I-1
30 CONTINUE
  IF(I) 26,32,26
32 RETURN
END

```

```

SUBROUTINE MODEL(BN,YCALC)

```

```

PURPOSE: ASSIGN COEFFICIENTS AND ROUTE EXECUTION FOR MODELS

```

```

DIMENSION YCALC(200),BN(28),C(14)
DIMENSION C10(0:2000),S10(0:2000)
COMMON/MODAT1/X(3,200),INDEX(14),NOB,NVAR
COMMON/MODAT2/CLEN,C01,C10,S10,DELT,RHO,TIMET
COMMON/MODAT4/DFT,MODE

```

```

UPDATE COEFFICIENT ARRAY

```

```

K=0
NU1=NVAR+1
NU2=NVAR*2
DO 2 I=NU1,NU2
  IF(INDEX(I-NVAR).EQ.0) GO TO 2
  K=K+1
  BN(I)=BN(K)
2 CONTINUE
DO 5 I=1,NVAR
5 C(I)=BN(NVAR+I)

```

```

CALL MODEL FUNCTION

```

```

X(1,0)=0.
FPULSE=0.
NCONDI=0
DFT=0.
IF (MODE.LE.2) THEN
  DO 10 J=1,NOB
    DELT=X(1,J)-X(1,J-1)-DFT
    IF(MODE.EQ.1) THEN
      CALL FUNC1(YCALC(J),X(1,J),C,NCONDI,FPULSE)
    ELSEIF(MODE.EQ.2) THEN
      CALL FUNC2(YCALC(J),X(1,J),C,NCONDI,FPULSE)
    ENDIF
10 CONTINUE
  ELSEIF (MODE.GT.2) THEN
    IF (MODE.EQ.3) THEN
      CALL FUNC3(YCALC,X,C,NCONDI,FPULSE)
    ELSEIF (MODE.EQ.4) THEN
      CALL FUNC4(YCALC,X,C,NCONDI,FPULSE)
    ENDIF
  ENDIF
RETURN
END

```

```

SUBROUTINE FUNC1(TOT,XX,C,NCONDI,FPULSE)

```

```

SOLUTION FOR S-CURVE ISOTHERM BY IMPROVED MIXING CELL
BY THE METHOD OF BISECTION (TWO SITE MODEL)

```

```

DIMENSION C(14),CMIN(14),CMAX(14)
DIMENSION C1(0:2000,2),S1(0:2000,2),SS1(0:2000,2),SC(2)
DIMENSION C10(0:2000),S10(0:2000),AK(14),JJ1(14),GAMMA(14)
COMMON/MODAT1/X(3,200),INDEX(14),NOB,NVAR
COMMON/MODAT2/CLEN,C01,C10,S10,DELT,RHO,TIMET
COMMON/MODAT3/CMIN,CMAX
COMMON/MODAT4/DFT,MODE
COMMON/MODAT5/DT
COMMON/MODAT7/BETA

```

```

FITTING ADDITIONAL DATA POINT TO SATISFY CONSTRAINTS

```

```

IF(XX.LT.0.) THEN
  TOT=0.
  DO 2 I=1,NVAR
    IF(INDEX(I).EQ.0) GOTO 2

```

```

  STARTING GAMMA VALUE

```

```

  IF(JJ1(I).EQ.0) GAMMA(I)=10.

```

```

  INCREASE BY MULTIPLES OF 10

```

```

  IF(C(I).GT.CMAX(I).OR.C(I).LT.CMIN(I)) THEN
    JJ1(I)=JJ1(I)+1
    GAMMA(I)=10.**(JJ1(I)+1)
    IF (GAMMA(I).GT.1.E10) GAMMA(I)=1.E10
  ENDIF

```

```

  CALCULATE AK'S AND THE PREDICTED ADDED DATA POINT

```

```

  AK(I)=(CMAX(I)-CMIN(I))/GAMMA(I)
  IF(C(I).GE.CMIN(I).AND.C(I).LE.CMAX(I)) THEN
    TOT=0.
  ELSEIF((C(I)-CMAX(I)).GT.1.E-12) THEN
    TOT=TOT+BETA*EXP(AK(I)/(1.-C(I)/CMAX(I)))

```

```

ELSEIF((CMIN(I)-C(I)).GT.1.E-12) THEN
  TOT=TOT+BETA*EXP(-AK(I)/(1.-C(I)/CMIN(I)))
ENDIF
2 CONTINUE
RETURN
ENDIF
DO 5 J=1,2
  C1(0,J)=C01
5 CONTINUE

C
C
C INPUT PARAMETERS

VEL=C(1)
DS=C(2)
PULSET=C(3)
ALPHA1=C(4)
ALPHA2=C(5)
ALPHA3=C(6)
ALPHA4=C(7)
ALPHA5=C(8)
ALPHA6=C(9)

C
C
C COMPUTE GRID SIZE

DX=2.*DS/VEL
RCLEN=CLEN-INT(CLEN/DX)*DX
NSTEP=2000
IF(RCLEN.LE.(DX*1.E-4)) THEN
  NCELL=CLEN/DX+DX*1.E-4
ELSE
  NCELL=INT(CLEN/DX+DX*1.E-4)+1
ENDIF

C
C
C NUMBER OF CELLS CONSTRAINT

IF (NCELL.GT.2000) THEN
  WRITE(*,*) ' VARIABLE ARRAY SIZE EXCEEDED '
  STOP
ELSEIF(NCELL.LT.3) THEN
  WRITE(*,*) ' PECTLET NUMBER TOO LOW: PROGRAM HALTED '
  STOP
ENDIF

C
C
C INITIAL CONCENTRATION

DO 20 I=1,NCELL
  IF (NCONDI .EQ. 0) THEN
    C1(I,1)=C10(I)
    S1(I,1)=S10(I)
  ENDIF
20 CONTINUE

C
C
C CALCULATION OF CONCENTRATION

CR=VEL*DT/DX
TOL=DMIN1((DELT*1.E-4),(DT*1.E-4))

C
C
C LINEAR AND NONLINEAR CASES ARE DEALT SEPARATELY

IF(ALPHA3.EQ.1. .AND. ALPHA4.EQ.1.) THEN

  C
  C
  C CASE OF LINEAR ISOTHERM

  ALIN1=1.+RHO*ALPHA2*ALPHA6
  ALIN2=RHO*ALPHA5*DT/2./(1.+ALPHA5*DT/2.)
  COEF1=ALIN1+CR/2.+ALIN2*(1.-ALPHA6)*ALPHA2
  COEF2=ALIN1-CR/2.-ALIN2*(1.-ALPHA6)*ALPHA2
  TIME=0.
  DO 195 J=1,NSTEP
    TIME=TIME+DT
    FPULSE=FPULSE+DT
    IF (FPULSE.EQ.DT) THEN
      C1(0,1)=.5*C01
    ELSE
      C1(0,1)=C01
    ENDIF
    IF(FPULSE.EQ.PULSET) THEN
      C1(0,1)=.5*C01
      C1(0,2)=0.
    ELSEIF(FPULSE.GT.PULSET) THEN
      C1(0,1)=0.
      C1(0,2)=0.
    ENDIF
    DO 95 I=1,NCELL
      C1(I,2)=(CR/2.*(C1(I-1,1)+C1(I-1,2))
      1 +COEF2*C1(I,1)+2.*ALIN2*S1(I,1))/COEF1
    95 S1(I,2)=ALIN2/RHO*((1.-ALPHA6)
    1 *ALPHA2*(C1(I,2)+C1(I,1))-S1(I,1))
    2 +S1(I,1)/(1.+ALPHA5*DT/2.)

    C
    C
    C STOP COMPUTING IF TIME IS EQUAL TO OR GREATER THAN DELT

    IF ((TIME-DELT).GE.TOL .OR. (DELT-TIME).LE.TOL) GOTO 210
    DO 97 I=1,NCELL
      C1(I,1)=C1(I,2)
    97 S1(I,1)=S1(I,2)
    195 CONTINUE
  ELSE

```


FOR ALL NONLINEAR CASES

TIME=0.

DO 200 J=1,NSTEP

TIME=TIME+DT

FPULSE=FPULSE+DT

IF (FPULSE.EQ.DT) THEN

C1(0,1)=.5*C01

ELSE

C1(0,1)=C01

ENDIF

IF (FPULSE.EQ.PULSET) THEN

C1(0,1)=.5*C01

C1(0,2)=0.

ELSEIF (FPULSE.GT.PULSET) THEN

C1(0,1)=0.

C1(0,2)=0.

ENDIF

DO 100 I=1,NCELL

C11=(C1(I,1)+CR/2.*(C1(I-1,2)

+C1(I-1,1)-C1(I,1)))/(1.+CR/2.)

1

COMPUTING EQUILIBRIUM CONCENTRATION BY METHOD OF BISECTION

A=0.

B=2.*C01

NNN=0

80

C1(I,2)=A

NNN=NNN+1

DO 85 K=1,2

SC(K)=(1.-ALPHA6)*ALPHA1

*(1.-(1.+(ALPHA2*C1(I,K)**ALPHA3)**ALPHA4)

1

SS1(I,K)=ALPHA6*ALPHA1

*(1.-(1.+(ALPHA2*C1(I,K)**ALPHA3)**ALPHA4)

85

S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))

+S1(I,1))/(1.+ALPHA5*DT/2.)

FCA=-C1(I,2)+C11+RHO

*(S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)

C1(I,2)=B

SC(2)=(1.-ALPHA6)*ALPHA1

*(1.-(1.+(ALPHA2*C1(I,2)**ALPHA3)**ALPHA4)

SS1(I,2)=ALPHA6*ALPHA1

*(1.-(1.+(ALPHA2*C1(I,2)**ALPHA3)**ALPHA4)

S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))

+S1(I,1))/(1.+ALPHA5*DT/2.)

FCC=-C1(I,2)+C11+RHO

*(S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)

IF (FCA*FCC.GT.0.) THEN

C1(I,2)=0.

S1(I,2)=0.

GOTO 100

ENDIF

C1(I,2)=(A+B)/2.

SC(2)=(1.-ALPHA6)*ALPHA1

*(1.-(1.+(ALPHA2*C1(I,2)**ALPHA3)**ALPHA4)

SS1(I,2)=ALPHA6*ALPHA1

*(1.-(1.+(ALPHA2*C1(I,2)**ALPHA3)**ALPHA4)

S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))

+S1(I,1))/(1.+ALPHA5*DT/2.)

FCC=-C1(I,2)+C11+RHO

*(S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)

IF (FCC.EQ.0.OR.(ABS(FCC).LT.1E-8)) THEN

GOTO 100

ELSEIF (ABS(A-B).LT.1.E-12) THEN

GOTO 100

ELSEIF (FCA*FCC.LT.0) THEN

A=A

B=C1(I,2)

ELSE

A=C1(I,2)

B=B

ENDIF

IF (NNN.GT.500) THEN

WRITE(*,*)

'BISECTION TERMINATED AFTER 500 ITERATIONS'

STOP

ENDIF

GOTO 80

100

CONTINUE

STOP COMPUTING IF TIME IS EQUAL TO OR GREATER THAN DELT

IF ((TIME-DELT).GE.TOL .OR. (DELT-TIME).LE.TOL) GOTO 210

DO 102 I=1,NCELL

C1(I,1)=C1(I,2)

102

S1(I,1)=S1(I,2)

200

CONTINUE

ENDIF

LINEAR INTERPOLATION IN SPACE AND TIME

DIFFERENCE IN TIME AND DELT TO BE PASSED TO CALCULATE NEXT DELT

210 DFT=TIME-DELT

IF (ABS(CLEN-NCELL*DX).LE.(DX*1.E-4)) THEN

TOT1=C1(NCELL,1)

TOT2=C1(NCELL,2)

LINEAR INTERPOLATION IN TIME

```

IF (ABS(DFT) .LE. TOL) THEN
  TOT=TOT2
ELSEIF (TOT2 .EQ. TOT1) THEN
  TOT=TOT2
ELSEIF (TOT2 .GT. TOT1) THEN
  TOT=TOT1+(TOT2-TOT1)/DT*(DT-(TIME-DELT))
ELSE
  TOT=TOT2+(TOT1-TOT2)/DT*(TIME-DELT)
ENDIF
ELSE
IF (C1(NCELL-1,2) .EQ. C1(NCELL,2)) THEN
  TOT11=C1(NCELL,2)
  TOT22=TOT11
ELSEIF (C1(NCELL-1,2) .GT. C1(NCELL,2)) THEN
  TOT11=C1(NCELL,1)+(C1(NCELL-1,1)-C1(NCELL,1))
  /DX*(NCELL*DX-CLEN)
  TOT22=C1(NCELL,2)+(C1(NCELL-1,2)-C1(NCELL,2))
  /DX*(NCELL*DX-CLEN)
ELSE
  TOT11=C1(NCELL-1,1)+(C1(NCELL,1)-C1(NCELL-1,1))
  /DX*(CLEN-(NCELL-1)*DX)
  TOT22=C1(NCELL-1,2)+(C1(NCELL,2)-C1(NCELL-1,2))
  /DX*(CLEN-(NCELL-1)*DX)
ENDIF
ENDIF

LINEAR INTERPOLATION IN TIME

IF (ABS(DFT) .LE. TOL) THEN
  TOT=TOT22
ELSEIF (TOT22 .EQ. TOT11) THEN
  TOT=TOT22
ELSEIF (TOT22 .GT. TOT11) THEN
  TOT=TOT11+(TOT22-TOT11)/DT*(DT-(TIME-DELT))
ELSE
  TOT=TOT22+(TOT11-TOT22)/DT*(TIME-DELT)
ENDIF
ENDIF

INITIAL CONCENTRATION FOR NEXT DATA

NCONDI=NCONDI+1
DO 215 I=1,NCELL+1
  C1(I,1)=C1(I,2)
215 S1(I,1)=S1(I,2)
RETURN
END

SUBROUTINE FUNC2(TOT,XX,C,NCONDI,FPULSE)

SOLUTION FOR S-CURVE ISOTHERM BY IMPROVED MIXING CELL
BY THE METHOD OF BISECTION (TWO SITE MODEL) WITH B. C. TYPE 3

DIMENSION X1(2),F(2)
DIMENSION C(14),CMIN(14),CMAX(14)
DIMENSION C1(0:2000,2),S1(0:2000,2),SS1(0:2000,2),SC(2000)
DIMENSION C10(0:2000),SKE(0:2,2),AK(14),GAMMA(14),JJ1(14)
DIMENSION S10(0:2000)
COMMON/MODAT1/X(3,200),INDEX(14),NOB,NVAR
COMMON/MODAT2/CLEN,C01,C10,S10,DELT,RHO,TIMET
COMMON/MODAT3/CMIN,CMAX
COMMON/MODAT4/DFT,MODE
COMMON/MODAT5/DT
COMMON/MODAT7/BETA
COMMON A1,A3,AC0,AC1,CR,ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,ALPHA6
EXTERNAL FCN2
DATA N,MAXIT,DELTA/2,20,1.E-12/
DATA XTOL,FTOL / 1.E-12, 1.E-12/

FITTING ADDITIONAL DATA POINT TO SATISFY CONSTRAINTS

IF(XX.LT.0.) THEN
  TOT=0.
  DO 2 I=1,NVAR
    IF(INDEX(I) .EQ.0) GOTO 2

STARTING GAMMA VALUE

IF(JJ1(I) .EQ.0) GAMMA(I)=10.

INCREASE BY MULTIPLES OF 10

IF(C(I) .GT. CMAX(I) .OR. C(I) .LT. CMIN(I)) THEN
  JJ1(I)=JJ1(I)+1
  GAMMA(I)=10.**(JJ1(I)+1)
ENDIF

CALCULATE AK'S AND THE PREDICTED ADDED DATA POINT

AK(I)=(CMAX(I)-CMIN(I))/GAMMA(I)
IF(C(I) .GT. CMIN(I) .AND. C(I) .LT. CMAX(I)) THEN
  TOTT=0.
ELSEIF((C(I)-CMAX(I)) .GT. 1.E-12) THEN
  TOT=TOT+BETA*EXP(AK(I)/(1.-C(I)/CMAX(I)))
ELSEIF((CMIN(I)-C(I)) .GT. 1.E-12) THEN
  TOT=TOT+BETA*EXP(-AK(I)/(1.-C(I)/CMIN(I)))
ENDIF
2 CONTINUE
RETURN

```

```

C      ENDIF
C
C      C00=C01
C      X1(1)=1.E-12
C      X1(2)=1.E-12
C
C      INPUT PARAMETERS
C
C      VEL=C(1)
C      DS=C(2)
C      PULSET=C(3)
C      ALPHA1=C(4)
C      ALPHA2=C(5)
C      ALPHA3=C(6)
C      ALPHA4=C(7)
C      ALPHA5=C(8)
C      ALPHA6=C(9)
C
C      COMPUTE GRID SIZE
C
C      DX=2.*DS/VEL
C      RCLEN=CLEN-INT(CLEN/DX)*DX
C      NSTEP=2000
C      IF(RCLEN.LE.(DX*1.E-4)) THEN
C          NCELL=CLEN/DX+DX*1.E-4
C      ELSE
C          NCELL=INT(CLEN/DX+DX*1.E-4)+1
C      ENDIF
C
C      NUMBER OF CELLS CONSTRAINT
C
C      IF (NCELL.GT.200) THEN
C          WRITE(*,*)' VARIABLE ARRAY SIZE EXCEEDED'
C          STOP
C      ELSEIF(NCELL.LT.3) THEN
C          WRITE(*,*)' PECTET NUMBER TOO LOW: PROGRAM HALTED'
C          STOP
C      ENDIF
C
C      INITIAL CONCENTRATION
C
C      IF (NCONDI .EQ. 0) THEN
C          DO 20 I=0,NCELL
C              C1(I,1)=C10(I)
20      S1(I,1)=S10(I)
C      ENDIF
C
C      CALCULATION OF CONCENTRATION
C
C      CR=VEL*DT/DX
C      TOL=MIN((DELT*1.E-4),(DT*1.E-4))
C
C      CASES OF LINEAR AND NONLINEAR PROBLEMS TREATED SEPARATELY
C
C      IF(ALPHA3.EQ.1. .AND. ALPHA4.EQ.1.) THEN
C
C          CASE OF LINEAR ISOTHERM
C
C          ALIN1=1.+RHO*ALPHA2*ALPHA6
C          ALIN2=RHO*ALPHA5*DT/2./(1.+ALPHA5*DT/2.)
C          COEF1=ALIN1+CR/2.+ALIN2*(1.-ALPHA6)*ALPHA2
C          COEF2=ALIN1-CR/2.-ALIN2*(1.-ALPHA6)*ALPHA2
C          ABC31=COEF1+CR*DX*VEL/DS-CR**2/4./COEF1
C          ABC32=COEF2-CR*DX*VEL/DS+CR**2/4./COEF1
C          ABC33=CR/2.*(1.+COEF2/COEF1)
C          TIME=0.
C          DO 195 J=1,NSTEP
C              TIME=TIME+DT
C              FPULSE=FPULSE+DT
C              IF(FPULSE.GT.PULSET) THEN
C                  C00=0.
C              ENDIF
C
C          COMPUTING EQUILIBRIUM CONCENTRATION AT POINTS 0 AND 1
C
C          C1(0,2)=(ABC32*C1(0,1)+ABC33*C1(1,1)
1      +2.*ALIN2*(CR/2./COEF1*S1(1,1)+S1(0,1))
2      +2.*CR*DX*VEL*C00/DS)/ABC31
C          C1(1,2)=(CR/2.*(C1(0,2)+C1(0,1))
1      +COEF2*C1(1,1)+2.*ALIN2*S1(1,1))/COEF1
C          DO 92 I=0,1
92      S1(I,2)=ALIN2/RHO*((1.-ALPHA6)
1      *ALPHA2*(C1(I,2)+C1(I,1))-S1(I,1))
C
C          SOLVE FOR REMAINING POINTS
C
C          DO 95 I=2,NCELL
C              C1(I,2)=(CR/2.*(C1(I-1,1)+C1(I-1,2))
1      +COEF2*C1(I,1)+2.*ALIN2*S1(I,1))/COEF1
95      S1(I,2)=ALIN2/RHO*((1.-ALPHA6)
1      *ALPHA2*(C1(I,2)+C1(I,1))-S1(I,1))
2      +S1(I,1)/(1.+ALPHA5*DT/2.)
C
C          STOP COMPUTING IF TIME IS EQUAL TO OR GREATER THAN DELT
C
C          IF ((TIME-DELT).GE.TOL .OR. (DELT-TIME).LE.TOL) GOTO 210
C          DO 97 I=0,NCELL
C              C1(I,1)=C1(I,2)
97      S1(I,1)=S1(I,2)

```


NONLINEAR CASE

```
A1=1.+CR/2.+CR*VEL*DX/DS
A2=1.-CR/2.-CR*VEL*DX/DS
A3=RHO*ALPHA5*DT/2./(1.+ALPHA5*DT/2.)
TIME=0.
DO 200 J=1,NSTEP
  TIME=TIME+DT
  FPULSE=FPULSE+DT
  IF(FPULSE.GT.PULSET) THEN
    C00=0.
  ENDIF
```

COMPUTING EQUILIBRIUM CONCENTRATION AT POINTS 0 AND 1

```
DO 125 I=0,1
  SKE(I,1)=(1.-ALPHA6)*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,1))**ALPHA3)**ALPHA4)
125 1 SS1(I,1)=ALPHA6*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,1))**ALPHA3)**ALPHA4)
  AC0=A2*C1(0,1)+CR/2.*C1(1,1)+2.*CR*VEL*DX/DS*C00
  -A3*SKE(0,1)+2.*A3*S1(0,1)+RHO*SS1(0,1)
  AC1=(1.-CR/2.)*C1(1,1)+CR/2.*C1(0,1)
  -A3*SKE(1,1)+2.*A3*S1(1,1)+RHO*SS1(1,1)
```

SOLVE EQUATIONS

```
CALL NLSYST(FCN,N,MAXIT,X1,F,DELTA,XTOL,FTOL)
DO 127 I=0,1
  C1(I,2)=X1(I+1)
  SKE(I,2)=(1.-ALPHA6)*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
  SS1(I,2)=ALPHA6*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
127 1 S1(I,2)=(DT*ALPHA5/2.*(SKE(I,1)+SKE(I,2)-S1(I,1))
  +S1(I,1))/(1.+ALPHA5*DT/2.)
```

SOLVE FOR REMAINING POINTS

```
DO 100 I=2,NCELL
  C11=(C1(I,1)+CR/2.*(C1(I-1,2)
  +C1(I-1,1)-C1(I,1)))/(1.+CR/2.)
```

COMPUTING EQUILIBRIUM CONCENTRATION BY METHOD OF BISECTION

```
A=0.
B=2.*C01
NNN=0
80 C1(I,2)=A
  NNN=NNN+1
  DO 85 K=1,2
    SC(K)=(1.-ALPHA6)*ALPHA1
    * (1.-(1.+(ALPHA2*C1(I,K))**ALPHA3)**ALPHA4)
85 1 SS1(I,K)=ALPHA6*ALPHA1
    * (1.-(1.+(ALPHA2*C1(I,K))**ALPHA3)**ALPHA4)
    S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
    +S1(I,1))/(1.+ALPHA5*DT/2.)
    FCA=-C1(I,2)+C11+RHO
    * (S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
    C1(I,2)=B
    SC(2)=(1.-ALPHA6)*ALPHA1
    * (1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
    SS1(I,2)=ALPHA6*ALPHA1
    * (1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
    S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
    +S1(I,1))/(1.+ALPHA5*DT/2.)
    FCC=-C1(I,2)+C11+RHO
    * (S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
    IF(FCA*FCC.GT.0) THEN
      C1(I,2)=0.
      S1(I,2)=0.
      GOTO 100
    ENDIF
    C1(I,2)=(A+B)/2.
    SC(2)=(1.-ALPHA6)*ALPHA1
    * (1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
    SS1(I,2)=ALPHA6*ALPHA1
    * (1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
    S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
    +S1(I,1))/(1.+ALPHA5*DT/2.)
    FCC=-C1(I,2)+C11+RHO
    * (S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
    IF(FCC.EQ.0..OR.(ABS(FCC).LT.1.E-8)) THEN
      GOTO 100
    ELSEIF(ABS(A-B).LT.1.E-8) THEN
      GOTO 100
    ELSEIF(FCA*FCC.LT.0.) THEN
      A=A
      B=C1(I,2)
    ELSE
      A=C1(I,2)
      B=B
    ENDIF
    IF(NNN.GT.500) THEN
      WRITE(*,*)
      'BISECTION TERMINATED AFTER 500 ITERATIONS'
```

```

                STOP
                ENDIF
                GOTO 80
100    CONTINUE
C
C        STOP COMPUTING IF TIME IS EQUAL TO OR GREATER THAN DELT
C
        IF ((TIME-DELT).GE.TOL .OR. (DELT-TIME).LE.TOL) GOTO 210
        DO 102 I=0,NCELL
            C1(I,1)=C1(I,2)
102    S1(I,1)=S1(I,2)
200    CONTINUE
ENDIF
C
C        LINEAR INTERPOLATION IN SPACE AND TIME
C
210    DFT=TIME-DELT
        IF (ABS(CLEN-NCELL*DX).LE.(DX*1.E-4)) THEN
            TOT1=C1(NCELL,1)
            TOT2=C1(NCELL,2)
C
C        LINEAR INTERPOLATION IN TIME
C        DIFFERENCE IN TIME AND DELT TO BE PASSED TO CALCULATE NEXT DELT
C
            IF (ABS(DFT).LE.TOL) THEN
                TOT=TOT2
            ELSEIF (TOT2.EQ.TOT1) THEN
                TOT=TOT2
            ELSEIF (TOT2.GT.TOT1) THEN
                TOT=TOT1+(TOT2-TOT1)/DT*(DT-(TIME-DELT))
            ELSE
                TOT=TOT2+(TOT1-TOT2)/DT*(TIME-DELT)
            ENDIF
        ELSE
            IF (C1(NCELL-1,2).EQ.C1(NCELL,2)) THEN
                TOT11=C1(NCELL,2)
                TOT22=TOT11
            ELSEIF (C1(NCELL-1,2).GT.C1(NCELL,2)) THEN
                TOT11=C1(NCELL,1)+(C1(NCELL-1,1)-C1(NCELL,1))
1          /DX*(NCELL*DX-CLEN)
                TOT22=C1(NCELL,2)+(C1(NCELL-1,2)-C1(NCELL,2))
1          /DX*(NCELL*DX-CLEN)
            ELSE
                TOT11=C1(NCELL-1,1)+(C1(NCELL,1)-C1(NCELL-1,1))
1          /DX*(CLEN-(NCELL-1)*DX)
                TOT22=C1(NCELL-1,2)+(C1(NCELL,2)-C1(NCELL-1,2))
1          /DX*(CLEN-(NCELL-1)*DX)
            ENDIF
C
C        LINEAR INTERPOLATION IN TIME
C
            IF (ABS(DFT).LE.TOL) THEN
                TOT=TOT22
            ELSEIF (TOT22.EQ.TOT11) THEN
                TOT=TOT22
            ELSEIF (TOT22.GT.TOT11) THEN
                TOT=TOT11+(TOT22-TOT11)/DT*(DT-(TIME-DELT))
            ELSE
                TOT=TOT22+(TOT11-TOT22)/DT*(TIME-DELT)
            ENDIF
        ENDIF
C
C        INITIAL CONCENTRATION FOR NEXT DATA
C
        NCONDI=NCONDI+1
        DO 215 I=0,NCELL
            C1(I,1)=C1(I,2)
215    S1(I,1)=S1(I,2)
        RETURN
        END
C
C        SUBROUTINE FOR FUNCTIONS
C        FOR BC TYPE 3 (TWO SITE MODEL)
C
        SUBROUTINE FCN2(X,F)
        DIMENSION X(2),F(2)
        DIMENSION SC(0:2,2),SE(0:2,2),C10(0:2000),S10(0:2000)
        COMMON A1,A3,AC0,AC1,CR,ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,ALPHA6
        COMMON/MODAT2/CLEN,C01,C10,S10,DELT,RHO,TIMET
        SC(0,2) = (1.-ALPHA6)*ALPHA1*(1.
1-(1.+(ALPHA2*X(1))**ALPHA3)**ALPHA4)
        SC(1,2) = (1.-ALPHA6)*ALPHA1*(1.
1-(1.+(ALPHA2*X(2))**ALPHA3)**ALPHA4)
        SE(0,2) = ALPHA6*ALPHA1*(1.
1-(1.+(ALPHA2*X(1))**ALPHA3)**ALPHA4)
        SE(1,2) = ALPHA6*ALPHA1*(1.
1-(1.+(ALPHA2*X(2))**ALPHA3)**ALPHA4)
        F(1) = A1*X(1)+A3*SC(0,2)-CR/2.*X(2)+RHO*SE(0,2)-AC0
        F(2) = (1.+CR/2.)*X(2)+A3*SC(1,2)-CR/2.*X(1)+RHO*SE(1,2)-AC1
        RETURN
        END
C
        SUBROUTINE FUNC3(TOT,XX,C,NCONDI,FPULSE)
C
C        SOLUTION FOR S-CURVE ISOTHERM BY IMPROVED MIXING CELL
C        BY THE METHOD OF BISECTION (TWO SITE MODEL WITH B.C. TYPE 1)
C
        IMPLICIT REAL (A-H,O-Z)
        DIMENSION C(14),CMIN(14),CMAX(14)

```

```

DIMENSION C1(0:200,2),S1(0:200,2),SS1(0:200,2),SC(2)
DIMENSION C10(0:2000),S10(0:2000),
1TOT1(0:200),TOT2(0:200),TOT(200),XX(3,200)
DIMENSION AK(14),JJ1(14),GAMMA(14)
COMMON/MODAT1/X(3,200),INDEX(14),NOB,NVAR
COMMON/MODAT2/CLEN,C01,C10,S10,DELT,RHO,TIMET
COMMON/MODAT3/CMIN,CMAX
COMMON/MODAT4/DFT,MODE
COMMON/MODAT5/DT
COMMON/MODAT7/BETA
TOT1(0)=C01
TOT2(0)=C01
DO 1 J=1,2
1 C1(0,J)=C01

C
C
C
INPUT PARAMETERS

VEL=C(1)
DS=C(2)
PULSET=C(3)
ALPHA1=C(4)
ALPHA2=C(5)
ALPHA3=C(6)
ALPHA4=C(7)
ALPHA5=C(8)
ALPHA6=C(9)

C
C
C
COMPUTE GRID SIZE

DX=2.*DS/VEL
RCLEN=CLEN-INT(CLEN/DX)*DX
NSTEP=2000
IF(RCLEN.LE.(DX*1.E-4)) THEN
  NCELL=CLEN/DX+DX*1.E-4
ELSE
  NCELL=INT(CLEN/DX+DX*1.E-4)+1
ENDIF

C
C
C
NUMBER OF CELLS CONSTRAINT

IF (NCELL.GT.200) THEN
  WRITE(*,*) ' VARIABLE ARRAY SIZE EXCEEDED '
  STOP
ELSEIF(NCELL.LT.3) THEN
  WRITE(*,*) ' PECKET NUMBER TOO LOW: PROGRAM HALTED '
  STOP
ENDIF

C
C
C
INITIAL CONCENTRATION

DO 20 I=1,NCELL
  C1(I,1)=C10(I)
20 S1(I,1)=S10(I)

C
C
C
CALCULATION OF CONCENTRATION

CR=VEL*DT/DX
TOL=DT*1.E-4

C
C
C
LINEAR AND NONLINEAR CASES ARE DEALT SEPARATELY

IF(ALPHA3.EQ.1. .AND. ALPHA4.EQ.1.) THEN

  C
  C
  C
  CASE OF LINEAR ISOTHERM

  ALIN1=1.+RHO*ALPHA2*ALPHA6
  ALIN2=RHO*ALPHA5*DT/2./(1.+ALPHA5*DT/2.)
  COEF1=ALIN1+CR/2.+ALIN2*(1.-ALPHA6)*ALPHA2
  COEF2=ALIN1-CR/2.-ALIN2*(1.-ALPHA6)*ALPHA2
  TIME=0.
  DO 195 J=1,NSTEP
    TIME=TIME+DT
    FPULSE=FPULSE+DT
    IF (FPULSE.EQ.DT) THEN
      C1(0,1)=.5*C01
    ELSE
      C1(0,1)=C01
    ENDIF
    IF(FPULSE.EQ.PULSET) THEN
      C1(0,1)=.5*C01
      C1(0,2)=0.
    ELSEIF(FPULSE.GT.PULSET) THEN
      C1(0,1)=0.
      C1(0,2)=0.
    ENDIF
    DO 95 I=1,NCELL
      C1(I,2)=(CR/2.*(C1(I-1,1)+C1(I-1,2))
1      +COEF2*C1(I,1)+2.*ALIN2*S1(I,1))/COEF1
95 1 S1(I,2)=ALIN2/RHO*((1.-ALPHA6)
1 *ALPHA2*(C1(I,2)+C1(I,1))-S1(I,1))
2 +S1(I,1)/(1.+ALPHA5*DT/2.)

C
C
C
STOP COMPUTING IF TIME IS EQUAL TO OR GREATER THAN TIMET

IF ((TIME-TIMET).GE.TOL .OR. (TIMET-TIME).LE.TOL) GOTO 210
DO 97 I=1,NCELL
  C1(I,1)=C1(I,2)
97 S1(I,1)=S1(I,2)
195 CONTINUE

```



```

ELSE
C
C
C   FOR ALL NONLINEAR CASES
TIME=0.
DO 200 J=1,NSTEP
  TIME=TIME+DT
  FPULSE=FPULSE+DT
  IF (FPULSE.EQ.DT) THEN
    C1(0,1)=.5*C01
  ELSE
    C1(0,1)=C01
  ENDIF
  IF (FPULSE.EQ.PULSET) THEN
    C1(0,1)=.5*C01
    C1(0,2)=0.
  ELSEIF (FPULSE.GT.PULSET) THEN
    C1(0,1)=0.
    C1(0,2)=0.
  ENDIF
  DO 100 I=1,NCELL
    C11=(C1(I,1)+CR/2.*(C1(I-1,2)
1      +C1(I-1,1)-C1(I,1)))/(1.+CR/2.)
C
C
C   COMPUTING EQUILIBRIUM CONCENTRATION BY METHOD OF BISECTION
A=0.
B=2.*C01
NNN=0
80   C1(I,2)=A
    NNN=NNN+1
    DO 85 K=1,2
      SC(K)=(1.-ALPHA6)*ALPHA1
1      *(1.-(1.+(ALPHA2*C1(I,K))**ALPHA3)**ALPHA4)
85   SS1(I,K)=ALPHA6*ALPHA1
1      *(1.-(1.+(ALPHA2*C1(I,K))**ALPHA3)**ALPHA4)
1      S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
1      +S1(I,1))/(1.+ALPHA5*DT/2.)
1      FCA=-C1(I,2)+C11+RHO
1      *(S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
1      C1(I,2)=B
1      SC(2)=(1.-ALPHA6)*ALPHA1
1      *(1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
1      SS1(I,2)=ALPHA6*ALPHA1
1      *(1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
1      S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
1      +S1(I,1))/(1.+ALPHA5*DT/2.)
1      FCC=-C1(I,2)+C11+RHO
1      *(S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
1      IF (FCA*FCC.GT.0.) THEN
        C1(I,2)=0.
        S1(I,2)=0.
        GOTO 100
      ENDIF
1      C1(I,2)=(A+B)/2.
1      SC(2)=(1.-ALPHA6)*ALPHA1
1      *(1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
1      SS1(I,2)=ALPHA6*ALPHA1
1      *(1.-(1.+(ALPHA2*C1(I,2))**ALPHA3)**ALPHA4)
1      S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
1      +S1(I,1))/(1.+ALPHA5*DT/2.)
1      FCC=-C1(I,2)+C11+RHO
1      *(S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
1      IF (FCC.EQ.0.OR.(ABS(FCC).LT.1E-8)) THEN
        GOTO 100
      ELSEIF (ABS(A-B).LT.1.E-12) THEN
        GOTO 100
      ELSEIF (FCA*FCC.LT.0) THEN
        A=A
        B=C1(I,2)
      ELSE
        A=C1(I,2)
        B=B
      ENDIF
1      IF (NNN.GT.500) THEN
        WRITE(*,*)
1      'BISECTION TERMINATED AFTER 500 ITERATIONS'
        STOP
      ENDIF
1      GOTO 80
100  CONTINUE
C
C
C   STOP COMPUTING IF TIME IS EQUAL TO OR GREATER THAN TIMET
IF ((TIME-TIMET).GE.TOL .OR. (TIMET-TIME).LE.TOL) GOTO 210
DO 102 I=1,NCELL
  C1(I,1)=C1(I,2)
102  S1(I,1)=S1(I,2)
200  CONTINUE
ENDIF
C
C
C   LINEAR INTERPOLATION IN SPACE AND TIME
210 DFT=TIME-TIMET
IF (ABS(TIME-TIMET).LE.TOL) THEN
  DO 220 I=1,NCELL
220  TOT2(I)=C1(I,2)
C
C
C   LINEAR INTERPOLATION IN SPACE

```

```

C
DO 230 I=1,NOB-1
  JJ=(XX(1,I))/DX+1
  IF (XX(1,I).LE.DX) JJ=1
  IF (XX(1,I).GE.(NCELL-1)*DX) JJ=NCELL
  X0=(JJ-1)*DX
  IF (TOT2(JJ).GT.TOT2(JJ-1)) THEN
    CDIF=TOT2(JJ)-TOT2(JJ-1)
    TOT(I)=TOT2(JJ-1)+(XX(1,I)-X0)*CDIF/DX
  ELSE
    CDIF=TOT2(JJ-1)-TOT2(JJ)
    TOT(I)=TOT2(JJ)+(DX-XX(1,I)+X0)*CDIF/DX
  ENDIF
230 CONTINUE
ELSE
DO 260 I=1,NCELL+1
  TOT1(I)=C1(I,1)
260 TOT2(I)=C1(I,2)

C
C
C
LINEAR INTERPOLATION IN TIME

DO 300 I=1,NCELL+1
  IF (TOT2(I).GT.TOT1(I)) THEN
    TOT2(I)=TOT1(I)+(TOT2(I)-TOT1(I))/DT*DFT
  ELSEIF (TOT2(I).LE.TOT1(I)) THEN
    TOT2(I)=TOT2(I)+(TOT1(I)-TOT2(I))/DT*DFT
  ENDIF
300 CONTINUE

C
C
C
LINEAR INTERPOLATION IN SPACE

DO 310 I=1,NOB-1
  JJ=(XX(1,I))/DX+1
  IF (XX(1,I).LE.DX) JJ=1
  IF (XX(1,I).GE.(NCELL-1)*DX) JJ=NCELL
  X0=(JJ-1)*DX
  IF (TOT2(JJ).GT.TOT2(JJ-1)) THEN
    CDIF=TOT2(JJ)-TOT2(JJ-1)
    TOT(I)=TOT2(JJ-1)+(XX(1,I)-X0)*CDIF/DX
  ELSE
    CDIF=TOT2(JJ-1)-TOT2(JJ)
    TOT(I)=TOT2(JJ)+(DX-XX(1,I)+X0)*CDIF/DX
  ENDIF
310 CONTINUE
ENDIF

C
C
C
FITTING ADDITIONAL DATA POINT TO SATISFY CONSTRAINTS

TOT(NOB)=0.
DO 2 I=1,NVAR
  IF (INDEX(I).EQ.0) GOTO 2

C
C
C
STARTING GAMMA VALUE

IF (JJ1(I).EQ.0) GAMMA(I)=10.

C
C
C
INCREASE BY MULTIPLES OF 10

IF (C(I).GT.CMAX(I).OR.C(I).LT.CMIN(I)) THEN
  JJ1(I)=JJ1(I)+1
  GAMMA(I)=10.**(JJ1(I)+1)
ENDIF

C
C
C
CALCULATE AK'S AND THE PREDICTED ADDED DATA POINT

AK(I)=(CMAX(I)-CMIN(I))/GAMMA(I)
IF (C(I).GE.CMIN(I).AND.C(I).LE.CMAX(I)) THEN
  TOT=0
ELSEIF ((C(I)-CMAX(I)).GT.1.E-12) THEN
  TOT(NOB)=TOT(NOB)+BETA*EXP(AK(I)/(1.-C(I)/CMAX(I)))
ELSEIF ((CMIN(I)-C(I)).GT.1.E-12) THEN
  TOT(NOB)=TOT(NOB)+BETA*EXP(-AK(I)/(1.-C(I)/CMIN(I)))
ENDIF
2 CONTINUE
RETURN
END

C
SUBROUTINE FUNC4(TOT,XX,C,NCONDI,FPULSE)

C
C
C
SOLUTION FOR S-CURVE ISOTHERM BY IMPROVED MIXING CELL
BY THE METHOD OF BISECTION (TWO SITE MODEL) WITH B. C. TYPE 3

DIMENSION X1(2),F(2)
DIMENSION C(14),CMIN(14),CMAX(14)
DIMENSION C1(0:200,2),S1(0:200,2),SS1(0:200,2),SC(200)
DIMENSION C10(0:2000),S10(0:2000),SKE(0:2,2)
DIMENSION TOT1(0:200),TOT2(0:200),TOT(200),XX(3,200)
DIMENSION AK(14),JJ1(14),GAMMA(14)
COMMON/MODAT1/X(3,200),INDEX(14),NOB,NVAR
COMMON/MODAT2/CLEN,C01,C10,S10,DELT,RHO,TIMET
COMMON/MODAT3/CMIN,CMAX
COMMON/MODAT4/DFT,MODE
COMMON/MODAT5/DT
COMMON/MODAT7/BETA
COMMON A1,A3,AC0,AC1,CR,ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,ALPHA6
EXTERNAL FCN2
DATA N,MAXIT,DELTA/2,20,1.E-12/
DATA XTOL,FTOL / 1.E-12, 1.E-12/
C00=C01

```

```

X1(1)=1.E-12
X1(2)=1.E-12
TOT1(0)=C01
TOT2(0)=C01

```

```

C
C
C
INPUT PARAMETERS

```

```

VEL=C(1)
DS=C(2)
PULSET=C(3)
ALPHA1=C(4)
ALPHA2=C(5)
ALPHA3=C(6)
ALPHA4=C(7)
ALPHA5=C(8)
ALPHA6=C(9)

```

```

C
C
C
COMPUTE GRID SIZE

```

```

DX=2.*DS/VEL
RCLEN=CLEN-INT(CLEN/DX)*DX
NSTEP=2000
IF(RCLEN.LE.(DX*1.E-4)) THEN
  NCELL=CLEN/DX+DX*1.E-4
ELSE
  NCELL=INT(CLEN/DX+DX*1.E-4)+1
ENDIF

```

```

C
C
C
NUMBER OF CELLS CONSTRAINT

```

```

IF(NCELL.GT.200) THEN
  WRITE(*,*)' VARIABLE ARRAY SIZE EXCEEDED'
  STOP
ELSEIF(NCELL.LT.3) THEN
  WRITE(*,*)' PECTET NUMBER TOO LOW: PROGRAM HALTED'
  STOP
ENDIF

```

```

C
C
C
INITIAL CONCENTRATION

```

```

DO 20 I=0,NCELL
  C1(I,1)=C10(I)
20 S1(I,1)=S10(I)

```

```

C
C
C
CALCULATION OF CONCENTRATION

```

```

CR=VEL*DT/DX
TOL=DMIN1((DELT*1.E-4),(DT*1.E-4))

```

```

C
C
C
CASES OF LINEAR AND NONLINEAR PROBLEMS TREATED SEPARATELY

```

```

IF(ALPHA3.EQ.1..AND.ALPHA4.EQ.1.) THEN

```

```

CASE OF LINEAR ISOTHERM

```

```

ALIN1=1.+RHO*ALPHA2*ALPHA6
ALIN2=RHO*ALPHA5*DT/2./(1.+ALPHA5*DT/2.)
COEF1=ALIN1+CR/2.+ALIN2*(1.-ALPHA6)*ALPHA2
COEF2=ALIN1-CR/2.-ALIN2*(1.-ALPHA6)*ALPHA2
ABC31=COEF1+CR*DX*VEL/DS-CR**2/4./COEF1
ABC32=COEF2-CR*DX*VEL/DS+CR**2/4./COEF1
ABC33=CR/2.*(1.+COEF2/COEF1)
TIME=0.
DO 195 J=1,NSTEP
  TIME=TIME+DT
  FPULSE=FPULSE+DT
  IF(FPULSE.GT.PULSET) THEN
    C00=0.
  ENDIF

```

```

C
C
C
COMPUTING EQUILIBRIUM CONCENTRATION AT POINTS 0 AND 1

```

```

1 C1(0,2)=(ABC32*C1(0,1)+ABC33*C1(1,1)
2 +2.*ALIN2*(CR/2./COEF1*S1(1,1)+S1(0,1))
1 C1(1,2)=(CR/2.*(C1(0,2)+C1(0,1))
1 +COEF2*C1(1,1)+2.*ALIN2*S1(1,1))/COEF1
92 DO 92 I=0,1
1 S1(I,2)=ALIN2/RHO*((1.-ALPHA6)
1 *ALPHA2*(C1(I,2)+C1(I,1))-S1(I,1))

```

```

C
C
C
SOLVE FOR REMAINING POINTS

```

```

DO 95 I=2,NCELL
1 C1(I,2)=(CR/2.*(C1(I-1,1)+C1(I-1,2))
95 +COEF2*C1(I,1)+2.*ALIN2*S1(I,1))/COEF1
1 S1(I,2)=ALIN2/RHO*((1.-ALPHA6)
1 *ALPHA2*(C1(I,2)+C1(I,1))-S1(I,1))
2 +S1(I,1)/(1.+ALPHA5*DT/2.)

```

```

C
C
C
STOP COMPUTING IF TIME IS EQUAL TO OR GREATER THAN TIMET

```

```

IF((TIME-TIMET).GE.TOL.OR.(TIMET-TIME).LE.TOL) GOTO 210
DO 97 I=0,NCELL
  C1(I,1)=C1(I,2)
97 S1(I,1)=S1(I,2)
195 CONTINUE
ELSE

```


NONLINEAR CASE

```

C
C
A1=1.+CR/2.+CR*VEL*DX/DS
A2=1.-CR/2.-CR*VEL*DX/DS
A3=RHO*ALPHA5*DT/2./(1.+ALPHA5*DT/2.)
TIME=0.
DO 200 J=1,NSTEP
  TIME=TIME+DT
  FPULSE=FPULSE+DT
  IF (FPULSE.GE.PULSET) THEN
    C00=0.
  ENDIF

```

```

C
C
C
COMPUTING EQUILIBRIUM CONCENTRATION AT POINTS 0 AND 1

```

```

DO 125 I=0,1
  SKE(I,1)=(1.-ALPHA6)*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,1))*ALPHA3)**ALPHA4)
125 1 SS1(I,1)=ALPHA6*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,1))*ALPHA3)**ALPHA4)
1 AC0=A2*C1(0,1)+CR/2.*C1(1,1)+2.*CR*VEL*DX/DS*C00
1 -A3*SKE(0,1)+2.*A3*S1(0,1)+RHO*SS1(0,1)
1 AC1=(1-CR/2.)*C1(1,1)+CR/2.*C1(0,1)
1 -A3*SKE(1,1)+2.*A3*S1(1,1)+RHO*SS1(1,1)

```

```

C
C
C
SOLVE EQUATIONS

```

```

CALL NLSYST(FCN,N,MAXIT,X1,F,DELTA,XTOL,FTOL)
DO 127 I=0,1
  C1(I,2)=X1(I+1)
  SKE(I,2)=(1.-ALPHA6)*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,2))*ALPHA3)**ALPHA4)
  SS1(I,2)=ALPHA6*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,2))*ALPHA3)**ALPHA4)
127 1 S1(I,2)=(DT*ALPHA5/2*(SKE(I,1)+SKE(I,2)-S1(I,1))
  +S1(I,1))/(1.+ALPHA5*DT/2.)

```

```

C
C
C
SOLVE FOR REMAINING POINTS

```

```

DO 100 I=2,NCELL
  C11=(C1(I,1)+CR/2.*(C1(I-1,2)
  +C1(I-1,1)-C1(I,1)))/(1.+CR/2.)

```

```

C
C
C
COMPUTING EQUILIBRIUM CONCENTRATION BY METHOD OF BISECTION

```

```

A=0.
B=C01
NNN=0
80 C1(I,2)=A
  NNN=NNN+1
  DO 85 K=1,2
    SC(K)=(1.-ALPHA6)*ALPHA1
    * (1.-(1.+(ALPHA2*C1(I,K))*ALPHA3)**ALPHA4)
  85 1 SS1(I,K)=ALPHA6*ALPHA1
    * (1.-(1.+(ALPHA2*C1(I,K))*ALPHA3)**ALPHA4)
  1 S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
  +S1(I,1))/(1.+ALPHA5*DT/2.)
  1 FCA=-C1(I,2)+C11+RHO
  * (S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
  1 C1(I,2)=B
  SC(2)=(1.-ALPHA6)*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,2))*ALPHA3)**ALPHA4)
  1 SS1(I,2)=ALPHA6*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,2))*ALPHA3)**ALPHA4)
  1 S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
  +S1(I,1))/(1.+ALPHA5*DT/2.)
  1 FCC=-C1(I,2)+C11+RHO
  * (S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
  IF (FCA*FCC.GT.0) THEN
    C1(I,2)=0.
    S1(I,2)=0.
    GOTO 100
  ENDIF
  C1(I,2)=(A+B)/2.
  SC(2)=(1.-ALPHA6)*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,2))*ALPHA3)**ALPHA4)
  1 SS1(I,2)=ALPHA6*ALPHA1
  * (1.-(1.+(ALPHA2*C1(I,2))*ALPHA3)**ALPHA4)
  1 S1(I,2)=(DT*ALPHA5/2.*(SC(1)+SC(2)-S1(I,1))
  +S1(I,1))/(1.+ALPHA5*DT/2.)
  1 FCC=-C1(I,2)+C11+RHO
  * (S1(I,1)-S1(I,2)+SS1(I,1)-SS1(I,2))/(1.+CR/2.)
  IF (FCC.EQ.0.OR.(ABS(FCC).LT.1E-8)) THEN
    GOTO 100
  ELSEIF (ABS(A-B).LT.1.E-8) THEN
    GOTO 100
  ELSEIF (FCA*FCC.LT.0) THEN
    A=A
    B=C1(I,2)
  ELSE
    A=C1(I,2)
    B=B
  ENDIF
  IF (NNN.GT.500) THEN
    WRITE(*,*)
    1 'INTERRUPTING BISECTION AFTER 500 ITERATIONS'
    STOP
  ENDIF
  GOTO 80

```

```

100      CONTINUE
C
C      STOP COMPUTING IF TIME IS EQUAL TO OR GREATER THAN TIMET
C
      IF ((TIME-TIMET).GE.TOL .OR.(TIMET-TIME).LE.TOL) GOTO 210
      DO 102 I=0,NCELL
          C1(I,1)=C1(I,2)
102      S1(I,1)=S1(I,2)
200      CONTINUE
      ENDIF
C
C      LINEAR INTERPOLATION IN SPACE AND TIME
C
210     DFT=TIME-TIMET
      IF (ABS(TIME-TIMET).LE.TOL) THEN
          DO 220 I=0,NCELL
220      TOT2(I)=C1(I,2)
C
C      LINEAR INTERPOLATION IN SPACE
C
          DO 230 I=1,NOB-1
              JJ=(XX(1,I))/DX+1
              IF (XX(1,I).LE.DX) JJ=1
              IF (XX(1,I).GE.(NCELL-1)*DX) JJ=NCELL
              X0=(JJ-1)*DX
              IF (TOT2(JJ).GT.TOT2(JJ-1)) THEN
                  CDIF=TOT2(JJ)-TOT2(JJ-1)
                  TOT(I)=TOT2(JJ-1)+(XX(1,I)-X0)*CDIF/DX
              ELSE
                  CDIF=TOT2(JJ-1)-TOT2(JJ)
                  TOT(I)=TOT2(JJ)+(DX-XX(1,I)+X0)*CDIF/DX
              ENDIF
230      CONTINUE
          ELSE
              DO 260 I=0,NCELL
                  TOT1(I)=C1(I,1)
260      TOT2(I)=C1(I,2)
C
C      LINEAR INTERPOLATION IN TIME
C
          DO 300 I=0,NCELL
              IF (TOT2(I).GT.TOT1(I)) THEN
                  TOT2(I)=TOT1(I)+(TOT2(I)-TOT1(I))/DT*DFT
              ELSEIF (TOT2(I).LE.TOT1(I)) THEN
                  TOT2(I)=TOT2(I)+(TOT1(I)-TOT2(I))/DT*DFT
              ENDIF
300      CONTINUE
C
C      LINEAR INTERPOLATION IN SPACE
C
          DO 310 I=1,NOB-1
              JJ=(XX(1,I))/DX+1
              IF (XX(1,I).LE.DX) JJ=1
              IF (XX(1,I).GE.(NCELL-1)*DX) JJ=NCELL
              X0=(JJ-1)*DX
              IF (TOT2(JJ).GT.TOT2(JJ-1)) THEN
                  CDIF=TOT2(JJ)-TOT2(JJ-1)
                  TOT(I)=TOT2(JJ-1)+(XX(1,I)-X0)*CDIF/DX
              ELSE
                  CDIF=TOT2(JJ-1)-TOT2(JJ)
                  TOT(I)=TOT2(JJ)+(DX-XX(1,I)+X0)*CDIF/DX
              ENDIF
310      CONTINUE
          ENDIF
C
C      FITTING ADDITIONAL DATA POINT TO SATISFY CONSTRAINTS
C
      TOT(NOB)=0.
      DO 2 I=1,NVAR
          IF(INDEX(I).EQ.0) GOTO 2
C
C      STARTING GAMMA VALUE
C
          IF(JJ1(I).EQ.0) GAMMA(I)=10.
C
C      INCREASE BY MULTIPLES OF 10
C
          IF(C(I).GT.CMAX(I).OR.C(I).LT.CMIN(I)) THEN
              JJ1(I)=JJ1(I)+1
              GAMMA(I)=10.**(JJ1(I)+1)
          ENDIF
C
C      CALCULATE AK'S AND THE PREDICTED ADDED DATA POINT
C
          AK(I)=(CMAX(I)-CMIN(I))/GAMMA(I)
          IF(C(I).GE.CMIN(I).AND.C(I).LE.CMAX(I)) THEN
              TOTT=0.
              ELSEIF((C(I)-CMAX(I)).GT.1.E-12) THEN
                  TOT(NOB)=TOT(NOB)+BETA*EXP(AK(I)/(1.-C(I)/CMAX(I)))
              ELSEIF((CMIN(I)-C(I)).GT.1.E-12) THEN
                  TOT(NOB)=TOT(NOB)+BETA*EXP(-AK(I)/(1.-C(I)/CMIN(I)))
              ENDIF
2      CONTINUE
      RETURN
      END
C
      SUBROUTINE NLSYST(FCN,N,MAXIT,X,F,DELTA,XTOL,FTOL)
C
      SUBROUTINE FOR SOLVING SYSTEM OF NONLINEAR EQUATIONS BY NEWTON'S

```

METHOD. THE PARTIAL DERIVATIVES OF THE FUNCTIONS ARE ESTIMATED BY DIFFERENCE QUOTIENTS WHEN A VARIABLE IS PERTURBATED BY AN AMOUNT EQUAL TO DELTA (DELTA ADDED). THIS IS DONE FOR EACH VARIABLE IN EACH FUNCTION. INCREMENTS TO IMPROVE THE ESTIMATES FOR THE X-VALUES ARE COMPUTED BY SOLVING DIRECTLY.

FCN: SUBROUTINE THAT COMPUTES VALUES OF THE FUNCTIONS.
 N: NUMBER OF EQUATIONS.
 MAXIT: LIMIT TO THE NUMBER OF ITERATIONS.
 X: ARRAY TO HOLD X VALUES
 F: ARRAY TO HOLD VALUES OF THE FUNCTIONS
 DELTA: A SMALL VALUE USED TO PERTURB THE X VALUES SO PARTIAL DERIVATIVES CAN BE COMPUTED BY DIFFERENCE QUOTIENTS
 XTOL: TOLERANCE VALUE FOR CHANGE IN X VALUES TO STOP ITERATIONS
 FTOL: TOLERANCE VALUE ON F TO TERMINATE
 K: RETURNS VALUES TO INDICATE HOW THE ROUTINE TERMINATED THIS WILL NOT BE SHOWN HERE; CAN BE MODIFIED IF WANTED
 K=1 XTOL WAS MET
 K=2 FTOL WAS MET
 K=-1 MAXIT EXCEEDED BUT TOLERANCES NOT MET
 K=-2 VERY SMALL PIVOT ENCOUNTERED IN GAUSSIAN ELIMINATION
 K=-3 INCORRECT VALUE OF N

DIMENSION X(N),F(N), A(10,11),XSAVE(10),FSAVE(10)

CHECK VALIDITY OF VALUE OF N

IF (N .LT. 2 .OR. N .GT. 10) GOTO 999

BEGIN ITERATIONS
 SAVE X VALUES, THEN GET F VALUES

NP=N+1
 DO 100 IT=1,MAXIT
 DO 10 IVBL=1,N
 XSAVE(IVBL)=X(IVBL)
 10 CONTINUE
 CALL FCN2(X,F)

TEST F VALUES AND SAVE THEM

ITEST=0
 DO 20 IFCN=1,N
 IF (ABS(F(IFCN)) .GT. FTOL) ITEST=ITEST+1
 FSAVE(IFCN)=F(IFCN)
 20 CONTINUE

SEE IF FTOL IS MET. IF NOT CONTINUE. IF SO, SET K=2 AND RETURN

30 IF (ITEST .NE. 0) GOTO 35
 K=2
 RETURN

THIS DOUBLE LOOP COMPUTES THE PARTIAL DERIVATIVES OF EACH FUNCTION FOR EACH VARIABLE AND STORES THEM IN A COEFF. ARRAY

35 DO 50 JCOL=1,N
 X(JCOL)=XSAVE(JCOL)+DELTA
 CALL FCN2(X,F)
 DO 40 IROW=1,N
 A(IROW,JCOL)=(F(IROW)-FSAVE(IROW))/DELTA
 40 CONTINUE

RESET X VALUES FOR NEXT COLUMN OF PARTIALS

X(JCOL)=XSAVE(JCOL)
 50 CONTINUE

NOW NEGATIVE F VALUES ARE EQUATED AS RIGHT HAND SIDES AND SOLVE LINEAR SYSTEM OF EQUATIONS

DO 60 IROW=1,N
 A(IROW,NP)=-FSAVE(IROW)
 60 CONTINUE
 ROOT2=(A(2,3)*A(1,1)-A(2,1)*A(1,3))/
 1 (A(2,2)*A(1,1)-A(2,1)*A(1,2))
 ROOT1=(A(1,3)-A(1,2)*ROOT2)/A(1,1)
 A(1,3)=ROOT1
 A(2,3)=ROOT2

BE SURE THAT COEFFICIENT MATRIX ARE NOT TOO ILL-CONDITIONED

DO 70 IROW=1,N
 IF(ABS(A(IROW,IROW)).LE.1.E-12) GOTO 998
 70 CONTINUE

APPLY THE CORRECTIONS TO THE X VALUES, ALSO SEE IF XTOL IS MET

ITEST=0
 DO 80 IVBL=1,N
 X(IVBL)=XSAVE(IVBL)+A(IVBL,NP)
 IF (ABS(A(IVBL,NP)).GT.XTOL) ITEST=ITEST+1
 80 CONTINUE

CHECK

DO 82 IVBL=1,N
 IF (X(IVBL).LT.0.) THEN
 X(IVBL)=0.


```

      ENDIF
82    CONTINUE
C
C      IF XTOL IS MET, RETURN, ELSE DO ANOTHER ITERATION
C
C      IF (ITEST.EQ.0) GOTO 997
100  CONTINUE
C
C      WHEN DONE MAXIT ITERATIONS, SET K=-1 AND RETURN
C
C      K=-1
C      RETURN
C
C      WHEN XTOL IS MET, SET K=1, RETURN
C
997  K=1
110  RETURN
C
C      WHEN PARTIALS FORM A NEARLY SINGULAR MATRIX. SET K=-2
C
998  K=-2
      WRITE(*,1003)
1003 FORMAT('CANNOT SOLVE SYSTEM, MATRIX NEARLY SINGULAR')
      RETURN
C
C      WHEN NUMBER OF EQUATIONS PASSED TO NLSYST IS INVALID
C
999  K=-3
      WRITE(*,1004) N
1004 FORMAT('NUMBER OF EQUATIONS PASSED TO NLSYST IS INVALID'
           $ ' MUST BE 1 < N < 11. VALUE WAS,' I3)
      RETURN
      END
C
C      FUNCTION TO GENERATE RANDOM NUMBERS OBTAINED FROM
C      NUMERICAL RECIPES
C
      FUNCTION RAN3(IDUM)
      PARAMETER (MBIG=1000000000, MSEED=161803398, FAC=1./MBIG)
      INTEGER I, IFF, II, INEXT, INEXTP, K
      INTEGER MJ, MK, MA(55)
      SAVE IFF, INEXT, INEXTP, MA
      DATA IFF /0/
      IF(IDUM.LT.0.OR.IFF.EQ.0) THEN
          IFF=1
          MJ=MSEED-IABS(IDUM)
          MJ=MOD(MJ,MBIG)
          MA(55)=MJ
          MK=1
          DO 11 I=1,54
              II=MOD(21*I,55)
              MA(II)=MK
              MK=MJ-MK
              IF(MK.LT.MZ) MK=MK+MBIG
              MJ=MA(II)
11         CONTINUE
          DO 13 K=1,4
              DO 12 I=1,55
                  MA(I)=MA(I)-MA(1+MOD(I+30,55))
                  IF(MA(I).LT.MZ) MA(I)=MA(I)+MBIG
12         CONTINUE
13         CONTINUE
          INEXT=0
          INEXTP=31
          IDUM=1
      ENDIF
      INEXT=INEXT+1
      IF(INEXT.EQ.56) INEXT=1
      INEXTP=INEXTP+1
      IF(INEXTP.EQ.56) INEXTP=1
      MJ=MA(INEXT)-MA(INEXTP)
      IF(MJ.LT.MZ) MJ=MJ+MBIG
      MA(INEXT)=MJ
      RAN3=MJ*FAC
      RETURN
      END

```