ABSTRACT

ROBERT H. GILBERTSEN. Application of contaminant fate and transport models in saturated soil (Under the direction of CASS T. MILLER.)

Systems that remediate an aquifer by purging contaminated water often operate for prolonged periods because contamination stubbornly lingers. This is the tailing phenonmenon. For nondegradable, nonionic organics in ground water, sorption is the prominent reaction. Sorption is crucial to the tailing phenomenon.

Types of sorption rate expressions include (1) local equilibrium, (2) Langmuir second order, (3) equilibrium/ first order, and (4) dual resistance sorption. Types of equilibrium isotherms include (1) linear, (2) Freundlich, or (3) Langmuir equilibrium.

This work improves four existing contaminant transport simulator models; each model incorporates one sorption rate assumption.

Based on modeling of the ground water contaminant nitrobenzene on Ann Arbor granular aquifer material in laboratory soil column reactors, the rate-controlled models predict contaminant breakthrough better than the local equilibrium models.

In one field-scale simulation all nonlinear and nonequilibrium models display the tailing phenomenon. The linear local equilibrium model does not.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank the citizens of North Carolina for supporting my research through the Water Resources Research Institute. Their grant titled "Modeling Organic Contaminant Sorption Impacts on Aquifer Restoration" funded this research.

I. INTRODUCTION

A. Ground water contamination, the general problem

Millions of Americans depend on ground water. At least 73 million people in the United States drink ground water (JAWWA, 1986), or about one-third of the U.S. population. This figure is not constant from state to state. Florida, for instance, provides 90 percent of its residents with drinking water from ground water (Tschinkel, 1986).

Because Americans depend so much on ground water, we wish it were always safe and pure. Unfortunately this is not the case. A great deal of ground water is contaminated. A commonly quoted estimate of ground water contamination is 2 percent of the contents of U.S. aquifers, but even this figure may underestimate the magnitude of the problem. The National Research Council recently reviewed estimates of the nation's ground water contamination. They dismissed earlier claims that 2 percent of America's ground water is contaminated as a "rough estimate based on oversimplified assumptions." The Council believes that the actual level is probably higher. Even if the percentage appears small, the tendency for contamination to exist near population centers magnifies the gravity of the problem (ENR, 1986a).

One rough gauge of the extent of serious ground water contamination is the number of Superfund sites on the National Priorities List. That number is now approaching 1000 and is expected to at least double (GWMR, 1985; SN, 1986; Grisham, et al., 1986). Presently there are 703 formal Superfund sites and 248 under consideration (USWN, 1987). EPA has no data on the volume of ground water contaminated by these sites, but estimates that 75 percent of the sites have produced observed ground water contamination (ENR, 1986b).

Whatever the estimate of current contamination may be, it is probably an underestimate. That is because a long lag typically occurs between a contamination incident and the eventual discovery of contamination. In ground water these lags often stretch for decades. Even if people begin using only safe waste disposal practices now, an increasing rate of contaminant discovery will almost certainly continue (Roberts et al., 1982).

B. Contaminant tailing, the specific problem

After someone discovers that an aquifer is contaminated, the land owner or the government often begins a cleanup effort. Although a variety of cleanup options is available, the most widely used option is the purge well method (Canter and Knox, 1986). The concept of purge wells is simple. The engineer deliberately places one or more withdrawal wells in the contaminated region of the aquifer. As pumping progresses over a period of time, the flow of water carries away the contaminated water. The soil in the aquifer poses a special problem because some contaminant sticks or "sorbs" to the soil. This contaminated soil can continue to cause problems by releasing sorbed contaminants even after wells remove the the original contaminated water. In the purge well method pure water does eventually wash even the contaminated soil of the aquifer clean. Several possibilities exist for disposing the withdrawn contaminated water. Treatment and discharge to a surface water, treatment and recharge to the ground water, and direct discharge to sewers are typical disposal options.

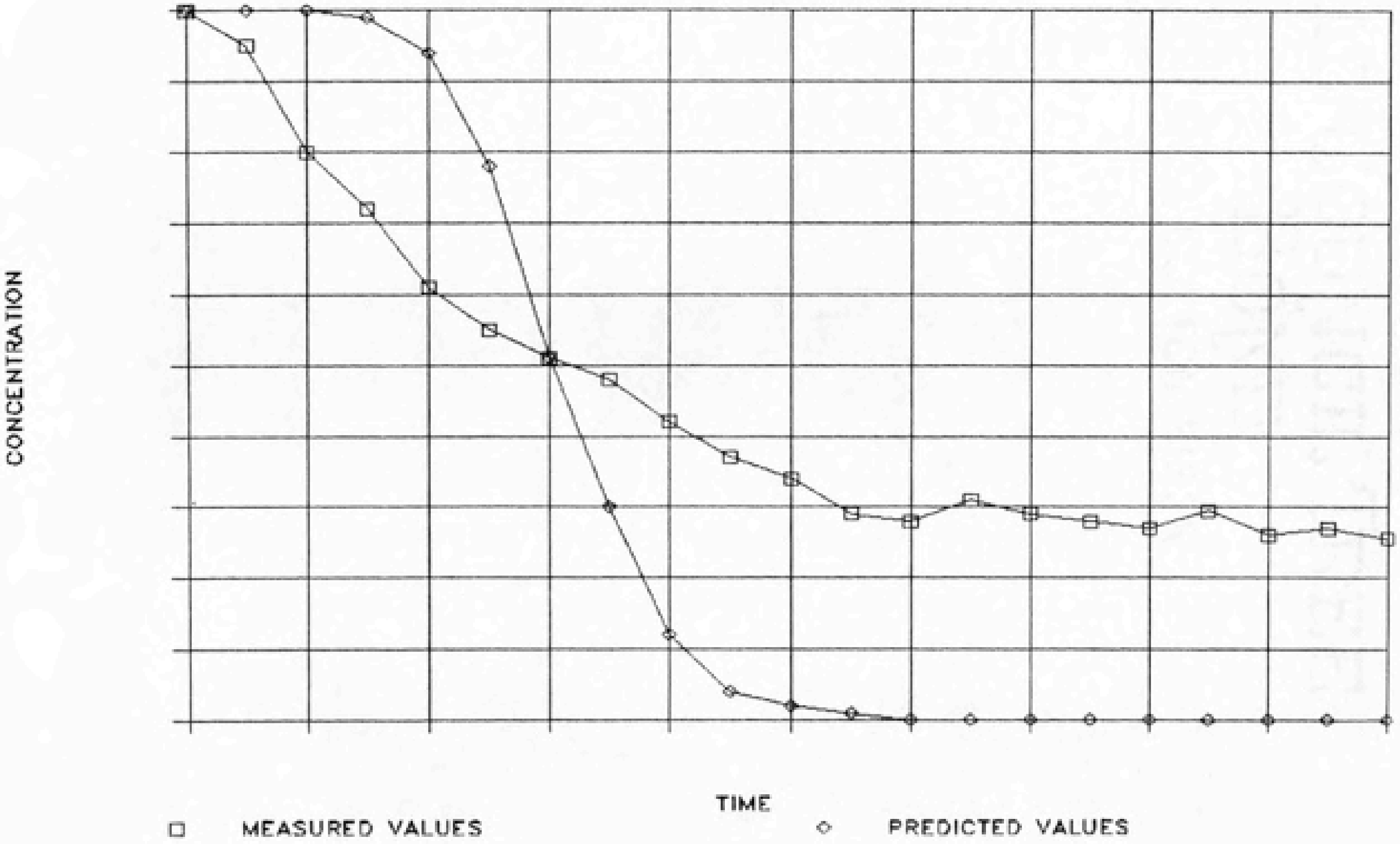
Such cleanup operations can cost a great deal. The average cost of cleaning up a Superfund site by whatever method is about $8 million, but due to the great range of aquifer and contaminant situations possible, that cost may range from $200,000 to $2 billion (Kavanaugh, 1986). For the specific case of well recovery systems, the single greatest cost may be the energy cost of maintaining the pumping for the years required to cleanse an aquifer (Canter and Knox, 1986).

Given the high cost of cleaning contaminated aquifers in general, and given the high energy cost of purge well systems in particular, accurate prediction of contaminant movement is crucial. Only by accurately understanding the movement of a ground water contaminant plume can a ground water professional provide a satisfactory measure to remedy the ground water problem. In the case of purge wells in particular, accurate forecasts of cleanup time are vital to economical design.

One drawback of many models currently in use, models that couple the most commonly applied physical modeling assumptions with the most commonly applied chemical modeling assumptions, is the so-called "tailing" phenomenon (Roberts et al., 1982). In the field and even in the lab researchers find that breakthrough response extends for considerably longer than expected. It appears that the contaminant sorbed to the soil takes longer to release than the conventional model would predict. Figure I-1 is an illustrative example of what tailing might look like for a purge well system. As the figure shows, at the start of pumping the ground water is contaminated, but as pumping progresses over time the ground water withdrawn becomes increasingly clean. A conventional model that made the conventional assumption that ground water and soil are always in local equilibrium would suggest a fairly rapid cleanup, but contaminant actually lingers for many times the expected period.

FIGURE I-1

THE TAILING PHENOMENON



Clearly, a field scale purge well design based on such a misleading forecast would incorporate grossly unrealistic operation and maintenance costs. Since cleanup costs are typically so large, this imprecision is important to examine.

C. Improved contaminant transport models, the focus of this report.

This work explores the hypothesis that the tailing phenomenon results from a number of subtleties of contaminant sorption. These subtleties are easily understood, and mathematical modeling can anticipate them. A discussion of two areas of concern follows.

First, most current ground water contaminant transport models depend on the notion of instantaneous equilibration of contaminant levels between the ground water and nearby soil. This notion is generally based on batch reactor experiments in which a small quantity of soil is stirred in a solution of water containing a contaminant of interest. The change in fluid phase contaminant concentration versus contact time is customarily plotted in a graph. When this measured change drops below experimental error, the system is assumed to be at equilibrium. This approach is less than rigorous, and may considerably underestimate the true time required for attainment of equilibrium. That is because the solid phase available for sorption comprises a nearly inconsequential portion of the total system. A good deal of activity may still occur in the solid without a noticeable effect on the fluid phase. The conventional approach may indicate that the system attains equilibrium within a few hours when in fact, the equilibration process may continue indefinitely (Coates and Elzerman, 1986).

Second, most current ground water contaminant transport models hypothesize that the soil-water equilibrium isotherm is linear. Recent work using low-organic soils over wide ranges of contaminant concentration suggests that various nonlinear isotherm models better define the equilibrium condition (Miller and Weber, 1986).

This paper, then, is devoted to the improved modeling of contaminant fate and transport by taking rate-controlled kinetics, intraparticle diffusion, and nonlinear sorption equilibrium into account. The technical report explores the effect of rate-controlled kinetics, intraparticle diffusion, and nonlinear sorption equilibrium on simulations of laboratory soil columns and on simulations of a simple field-scale application. Improvements of existing simulator programs allow improved modeling precision and reliability.

II. THEORETICAL BACKGROUND AND LITERATURE REVIEW

A. The advective-dispersive equation

The mathematical specification of physical and chemical phenomena related to contaminant fate and transport makes development of computerized, mathematical, ground-water models possible. Such development begins here with the fundamental physical processes -- hydrodynamics -- and then adds the relevant chemical process -- sorption.

Consider first an elemental volume. Within this unit volume, conservation of mass dictates

$$\Delta\text{mass} = \text{area}*(\text{flux in}) - \text{area}*(\text{flux out}) \pm (\text{source or sink}) \pm (\text{reaction}) \qquad \text{(II-1)}$$

where

Δmass = the net rate of change of contaminant mass within the elemental volume (MT^{-1});

area = area of element face normal to velocity of flow (L^2);

flux in = mass transfer into the elemental volume ($ML^{-2}T^{-1}$);

flux out = mass transfer from the elemental volume ($ML^{-2}T^{-1}$);

source or sink = contaminant mass added or removed (MT^{-1});

reaction = gain or loss of mass due to reaction of any kind (MT^{-1}).

The physical forces responsible for flux into and out of the element are advection, mechanical dispersion, and molecular diffusion. Most engineers assume Fick's law governs mechanical dispersion and molecular diffusion. In three dimensions, detailed consideration of mass balance yields the advective-dispersive equation:

$$\frac{\delta C}{\delta t} = - \vec{v} \cdot \mathrm{grad}C + \mathrm{div}(D_h \mathrm{grad}C) \qquad \text{(II-2)}$$

$$+ \left(\frac{\delta C}{\delta T}\right)_{rxn} + \gamma(C) + \frac{\rho_b}{\theta} \left(\frac{\delta q}{\delta t}\right)_{srp}$$

where

C = solution-phase solute concentration (ML^{-3});

t = time (T);

$\vec{v}$ = pore velocity vector (LT^{-1});

D_h = second-rank hydrodynamic dispersion tensor (L^2T^{-1});

$\gamma(C)$ = fluid-phase solute source ($ML^{-3}T^{-1}$);

ρ_b = bulk density of the soil phase (ML^{-3});

θ = volume void fraction in the medium (dimensionless);

q = volume-average soil-phase mass normalized by the mass of the solid phase (MM^{-1});

rxn = subscript denoting a general chemical or mass-transfer reaction (dimensionless);

srp = subscript denoting sorption reaction (dimensionless).

This research considers the one-dimensional case. The previous general development (adapted from Bear, 1979) reduces to a simpler form in one-dimensional systems. In one dimension without sources or sinks and without reactions other than sorption, the mass-balance approach yields a simpler equation. Physical chemists working with chromatographic columns discovered this equation almost four decades ago (adapted from Lapidus and Amundson, 1952):

$$\frac{\delta C}{\delta t} = -v_z \frac{\delta C}{\delta z} + D_h \frac{\delta^2 C}{\delta z^2} - \frac{\rho_b}{\theta} \left(\frac{\delta q}{\delta t}\right)_{srp} \qquad \text{(II-3)}$$

where

D_h = longitudinal hydrodynamic dispersion coefficient (L^2T^{-1});

v_z = pore velocity in the longitudinal direction (LT^{-1});

z = longitudinal distance variable (L).

Lapidus and Amundson performed their pioneering work in ion exchange and chromatographic columns, but the basic principles also apply to one-dimensional ground water flow and ultimately to higher-dimensional, ground-water systems. Interestingly, the chemists' early work examined the relative applicability of rate and equilibrium controlled sorption modeling, which is a central focus of this technical report.

Although some researchers (van Genuchten, et al., 1984; Crittenden, et al., 1986; Goltz and Roberts, 1986; and

Parker and Valocchi, 1986) argue that the advective-dispersive equation presented here can oversimplify ground water hydrodynamics, this research uses it for two reasons. First, it is by far the most common means of analyzing ground water fate and transport. Secondly, laboratory tracer tests using a nonreactive, nonsorbing tracer suggest that for the laboratory systems studied in this report the advective-dispersive equation reasonably approximates the hydrodynamics.

B. The sorption expression

Once system hydrodynamics are assumed, the only remaining task in modeling the fate and transport of a nondegradable contaminant is to specify the nature of the sorption phenomenon. Because of the huge surface area over which ground water and soil contact each other, accurate characterization of sorption is indispensable to the construction of a realistic ground-water contaminant-transport model. Sorption is a general term used to describe the uptake of a contaminant by soil without specific reference to any particular mechanism. Sorption encompasses both surface adsorption of a contaminant onto the exterior of soil particles and also the partitioning of a contaminant between water and the interior of soil particles (Chiou, in press).

Considerable disagreement surrounds the sorption expression. Debate centers on two issues, the shape of equilibrium isotherm plots and the rate of equilibrium attainment. The question of equilibrium isotherm shape centers on whether increasing fluid-phase contaminant concentration in the ground water causes a linear increase in solid-phase contaminant concentration. The question of rate centers on whether the contaminant in ground water rapidly equilibrates with the contaminant sorbed in the soil particles.

The next few pages detail the development of four fundamentally different ways of looking at the sorption expression. These formulations form the basis of the computer programs revised for this research.

<u>The local equilibrium approach</u>

This approach (Freeze and Cherry, 1979) assumes that interphase mass transfer occurs so rapidly that the solid phase -- the particles that compose the aquifer -- and the fluid phase -- the ground water -- are always in local equilibrium.

Focusing specifically on the sorption term of equation (II-3) and invoking the chain rule,

$$\frac{\delta q}{\delta t} = \frac{\delta q}{\delta C} \frac{\delta C}{\delta t} . \qquad \text{(II-4)}$$

Equation (II-4) creates a need to define the relation between the solid-phase concentration and the fluid-phase concentration. For the case of instantaneous local equilibrium, the sorption isotherm provides the needed information. One possible isotherm is the Freundlich isotherm.

$$q_e = K_F C_e{}^n \tag{II-5}$$

where

q_e = equilibrium volume-average soil-phase mass normalized by the solid-phase mass (MM^{-1});

K_F = Freundlich isotherm sorption-capacity constant ($(L^3M^{-1})^n$);

C_e = equilibrium solution-phase concentration (ML^{-3});

n = Freundlich isotherm sorption-energy constant (dimensionless).

Assuming local equilibrium, any solid-phase concentration and fluid-phase concentration combination that satisfies the equation is an equilibrium solution. This permits differentiation of (II-5).

$$\frac{dq}{dC} = \frac{\delta C}{\delta t} \; nK_F C^{n-1} \; . \tag{II-6}$$

Putting this new information to work, equation (II-4) becomes

$$\frac{\delta q}{\delta t} = \frac{\delta C}{\delta t} nK_F C^{n-1} \quad . \tag{II-7}$$

Some researchers (Chiou, in press; Chiou, et al., 1979; Chiou, et al., 1983) maintain that the Freundlich coefficient equals unity for nonionic polar organics, the class of contaminants of interest in this report. These researchers suggest that nonionic polar organics sorb by means of partitioning to the organic matter in aquifer materials. In other words they believe that the sorption isotherm is linear. This is also the conventional wisdom among practicing engineers. Other researchers (Saltzman et al., 1972; Mingelgrin and Gerstl, 1983; and Miller and Weber, 1986) claim that the Freundlich form with n not equal to unity is more likely to be true, particularly in low-organic aquifer materials over wide contaminant concentration ranges. Such an extreme situation, they maintain, is common in ground water, but rare in sorption studies.

Regardless of whether the linear or nonlinear isotherm is true, the derivative of the general Freundlich isotherm, equation (II-7), may enter the advective dispersive equation, equation (II-3) to yield

$$\frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z} - v_z \frac{\delta C}{\delta z} - \frac{\rho_b}{\theta} n K_f C^{n-1} \frac{\delta C}{\delta t} \quad . \tag{II-8}$$

Rearrangement yields

$$R_f \frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z^2} - v_z \frac{\delta C}{\delta z} \qquad \text{(II-9)}$$

where

$$R_f = 1 + \frac{\rho_b}{\theta} n K_f C^{n-1} \quad ;$$

R_f = retardation factor.

The term Rf is the retardation factor. It has an interesting property: If the Freundlich coefficient equals unity the retardation factor is a constant. The constant provides the ratio of the average ground water velocity to the average contaminant velocity. This is a compelling reason why engineers may wish to assume linear equilibrium, quite apart from the question of whether the assumption of linearity is rigorously true.

The Langmuir second-order approach

This model -- which Hiester and Vermeulen (1952) pioneered in packed-bed adsorbers -- carries two assumptions: (1) Sorption is a function of the product of the fluid-phase concentration and the difference between the sorption capacity and the contaminant concentration on the solid phase. (2) Desorption is simply a function of the solid- phase concentration.

Specifically, the solid-phase governing equation is

$$\frac{\delta q}{\delta t} = k_s[C(Q^o - q) - \frac{q}{b}] \qquad \text{(II-10)}$$

where

Q^o = Langmuir isotherm sorption-capacity constant (MM^{-1});

k_s = second-order Langmuir model rate constant ($L^3M^{-1}T^{-1}$);

b = Langmuir isotherm sorption-energy constant (L^3M^{-1}).

As conditions approach steady state, the above equation yields the Langmuir isotherm,

$$q_e = \frac{bQ^oC_e}{1 + bC_e} \quad . \qquad \text{(II-11)}$$

As before, this development is for a nondegradable, nonreactive contaminant.

Substituting equation (II-10), the solid-phase governing equation, into equation (II-3), the advective-dispersive equation, yields

$$\frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z^2} - v_z \frac{\delta C}{\delta z} \qquad \text{(II-12)}$$

$$- \frac{\rho_b}{\theta} k_s[C(Q^o - q) - \frac{q}{b}] \quad .$$

Simultaneous solution of equation (II-12) and equation (II- 10) links the solid-phase and fluid-phase solutions.

The first-order rate controlled with parallel equilibrium approach

Cameron and Klute (1977) were among the first to propose this approach. It assumes that there are two types of sites on the aquifer material where sorption may occur. The first group of sites allows fast sorption. The model assumes sorption there occurs so quickly that the fast sites locally equilibrate with nearby ground water. At the second group of sites sorption is slower. Sorption there proceeds according to both rate and equilibrium expressions.

In equation form,

$$\frac{\delta q}{\delta t} = \frac{\delta q_f}{\delta t} + \frac{\delta q_s}{\delta t} \qquad \text{(II-13)}$$

where

q_f = volume-average soil-phase mass normalized by the mass of the solid phase for the rapid sorption rate component of the equilibrium/first-order model (MM^{-1});

q_s = volume-average soil-phase mass normalized by the mass of the soild phase for the slow sorption rate component of the equilibrium/first-order model (MM^{-1}).

Applying the chain rule,

$$\frac{\delta q}{\delta t} = \frac{\delta q_f}{\delta C} \frac{\delta C}{\delta t} + \frac{\delta q_s}{\delta t} \quad . \qquad \text{(II-14)}$$

Equation II-14 is the solid-phase governing equation. It is ready to substitute into equation (II-3), the advective-dispersive equation. This yields

$$\frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z^2} - v_z \frac{\delta C}{\delta z} - \frac{\rho_b}{\theta} \left(\frac{\delta q_f}{\delta C} \frac{\delta C}{\delta t} + \frac{\delta q_s}{\delta t}\right) . \qquad \text{(II-15)}$$

Assuming that the Freundlich equilibrium model governs the fast sites, equation (II-15) develops just as the Freundlich local-equilibrium model did to produce

$$\left(1 + \frac{K_{F,f} n_f \rho_b}{\theta} C^{n_f - 1}\right) \frac{\delta C}{\delta t} \qquad \text{(II-16)}$$

$$= D_h \frac{\delta^2 C}{\delta z^2} - v_z \frac{\delta C}{\delta z} - \frac{\rho_b}{\theta} \frac{\delta q_s}{\delta t}$$

where

$K_{F,f}$ = Freundlich isotherm sorption-capacity constant for the rapid rate component of the equilibrium/first-order rate model $((L^3M^{-1})^{n_f})$;

n_f = Freundlich isotherm sorption-energy constant for the rapid rate component of the equilibrium/first-order rate model (dimensionless).

The addition of a first-order expression for the slow-sorption term yields

$$R_{f,f} \frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z^2} - v_z \frac{\delta C}{\delta z} - \frac{\rho_b}{\theta} \alpha(K_{F,s} C^{n_s} - q_s) \qquad \text{(II-17)}$$

where

α = equilibrium/first-order mass-transfer coefficient (T^{-1});

$R_{f,f}$ = retardation factor for the rapidly sorbing fraction in the equilibrium/first-order model (dimensionless);

$K_{F,s}$ = Freundlich isotherm sorption-capacity constant for the slow rate component of the equilibrium/first-order rate model ($(L^3M^{-1})^{n_s}$);

n_s = Freundlich isotherm sorption-energy constant for the slow rate component of the equilibrium/first-order model (dimensionless).

The slow-sorption solid-phase equation is

$$\frac{\delta q_s}{\delta t} = \alpha(K_{F,s} C^{n_s} - q_s) \quad . \qquad \text{(II-18)}$$

Simultaneous solution links equations (II-17) and (II-18).

The dual-resistance approach

Crittenden and Weber (1978) were among those who performed early work with the dual-resistance model in porous adsorbants. This approach assumes that sorption is a two-step process. For a molecule of contaminant to sorb to the soil it must first pass through a film layer which surrounds each soil particle and then diffuse into the soil particle proper. Even though aquifer soil is mostly impenetrable mineral, this does not detract from the model's applicability. The diffusion process simply shifts to the patches of organic matter which dot the exterior of the soil particles, so the porous adsorbant model also applies to sand grains as long as there is some organic matter on the grains' exterior.

If the concentration of contaminant on a soil particle composed of spherical shells equals the integral of contaminant mass on each shell over the particle radius divided by the particle volume, then

$$q = \frac{3}{R^3} \left(\int_o^R r^2 q_r dr \right) \qquad \text{(II-19)}$$

where

R = radius of soil particle (L);

r = radial distance variable for dual-resistance model (L);

q_r = solid-phase solute mass normalized by the mass of the solid phase as a function of radial position (MM^{-1}),

and

$$\frac{\delta q}{\delta t} = \frac{3}{R^3} \frac{\delta}{\delta t} \left(\int_o^R r^2 q_r dr \right) \quad . \tag{II-20}$$

By applying a mass balance to mass transfer among adjacent shells and assuming Fickian diffusion, the solid phase governing equation is

$$D_s \left(\frac{\delta^2 q_r}{\delta r^2} + \frac{2}{r} \frac{\delta q_r}{\delta r} \right) = \frac{\delta q_s}{\delta t} \tag{II-21}$$

where

D_s = intraparticle surface-diffusion coefficient for dual-resistance model (L^2T^{-1}).

The solid phase governing equation has two boundary conditions. The first follows from radial symmetry at the particle center.

$$\frac{\delta q_r}{\delta r} \Big|_{r=o} = 0 \quad . \tag{II-22}$$

The second combines Fickian diffusion at the films's exterior with flux through the boundary-layer film.

$$\frac{\delta q_r}{\delta r} \Big|_{r=R} = \frac{k_f}{D_s \rho} (C - C_s) \tag{II-23}$$

where

k_f = film mass transfer coefficient for the DUAL model (LT^{-1});

C_s = fluid-phase, equilibrium-isotherm concentration corresponding to the soil-phase concentration at the particle boundary (ML^{-3});

ρ = density of the soil particle (ML^{-3}).

Although equation (II-20) defines the change of solid-phase concentration with respect to time, a more practical definition to substitute into the fluid-phase governing equation comes from examination of flux.

$$\frac{\delta q}{\delta t} = \text{MASS FLUX} * \text{SURFACE AREA} * \frac{1}{\text{PARTICLE MASS}} \quad . \qquad \text{(II-24)}$$

Substituting mathematical terms,

$$\frac{\delta q}{\delta t} = k_f(C - C_s) * 4\pi R^2 * \frac{1}{\frac{4}{3}\pi R^3 \rho} \quad . \qquad \text{(II-25)}$$

Simplification yields the change in the solid-phase concentration with respect to time to plug into the advective-dispersive equation.

$$\frac{\delta q}{\delta t} = \frac{3k_f(C - C_s)}{\rho R} \quad . \qquad \text{(II-26)}$$

Now the advective-dispersive equation, equation (II-3) becomes

$$\frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z^2} - \frac{3\ k_f\ (1 - \theta)}{R\theta} (C - C_s)\ , \qquad \text{(II-27)}$$

which serves as the fluid-phase governing equation.

Simultaneous solution links the fluid-phase governing equation, equation (II-27) with the solid-phase governing equation, equation (II-21).

C. Related research

Valocchi (1985) examined the relative performance of rate-controlled versus local-equilibrium-controlled sorption in simulated one-dimensional ground water column experiments. The study included not only chemical nonequilibrium but also physical nonequilibrium models. Valocchi did not examine the importance of isotherm shape because he used the method of time moments, an analytical analysis method that requires use of linear isotherms.

Chiou (in press) has conducted numerous batch laboratory experiments to determine equilibrium sorption isotherm shapes for a variety of nonionic organic solvents. His studies have encompassed a range of organic matter content, mineral matter, and moisture content.

Miller (1984) studied and modeled sorption in batch reactor laboratory systems and in laboratory soil column reactors. His work included the contaminants lindane and nitrobenzene, but focused on lindane. He demonstrated that lindane exhibits nonlinear, rate-controlled sorption on several granular aquifer materials.

This technical report is closely allied with the original work of Miller. The chief distinctions of this technical report are refinements of the numerical model simulator programs and the modeling of the contaminant nitrobenzene in ground water columns.

III. EXPERIMENTAL METHODS

A. Description of laboratory studies

This technical report is an offshoot of the dissertation of Miller (1984). All laboratory work and many programs used in this technical report come from the work of Miller. Clearly, a basic understanding of that work is a prerequisite to understanding this work. This first section of Chapter II provides a framework for such a basic understanding. A reader interested in details should refer to the original dissertation.

The original study is titled Modeling of Sorption and Desorption Phenomena for Hydrophobic Organic Contaminants in Saturated Soil Environments. The work consists of three major parts: bottle-point equilibrium studies, completely mixed batch reactor (CMBR) studies, and soil-column reactor studies. It focuses on the sorption of two hydrophobic organic contaminants taken one at a time on a variety of glacially deposited Wisconsin Age sands.

Of particular pertinence to the present technical report is the work on the contaminant nitrobenzene on Ann Arbor soil. Miller's modeling thoroughly covers only bottle point and CMBR reactors. The dissertation does not detail the modeling of the soil-column reactor for this particular contaminant-soil combination even though a complete set of laboratory data, rate parameters, and equilibrium parameters is at hand.

Nitrobenzene ($C_6H_5NO_2$), the contaminant used in the experiments of interest, is also known as oil of mirbane. It is a product of the organic chemical industry. Solvent recovery plants use it, as do manufacturers of dyestuffs. It is a solvent used in TNT production. It is also a constituent in shoe polish. It has the odor of bitter almonds. Miller studied it because of its common occurrence in the environment, its moderately nonvolatile nature, its resistance to degradation, and its partitioning properties.

The Ann Arbor soil had a median grain diameter of 0.232 mm, a grain size uniformity coefficient (d_{60}/d_{10}) of 2.616, a hydraulic conductivity of 4.15×10^{-3} cm/sec, a total organic carbon content of 1.14 percent, and a cation exchange capacity of 6.9 meq/100gr. Miller studied Ann Arbor soil because of its sandy character, a character shared my many aquifers tapped for potable water supplies.

This remainder of this chapter details how the student used the information from Miller's original studies, modified and wrote computer programs, and then modeled nitrobenzene on Ann Arbor soil in soil columns.

B. Improvement of numerical models

Computer model improvement represents by far the largest portion of the time budget of this endeavor. Copies of the programs are located in the appendix.

At the start of this study, this research inherited four FORTRAN computer programs which contained the basic algorithms for finite difference solution of the four types of sorption models presented in the previous chapter. The programs were descendants of the programs Miller used in his dissertation. These programs were named local equilibrium with dispersion (LED); Langmuir second order sorption (LSO); equilibrium / first-order sorption (FED); and dual-resistance (DUAL).

The finite difference models are, of course, numerical models, not analytical solutions. The nonlinearity of the problems and the desire to have flexible control over boundary conditions force the use of numerical models.

This paper often refers to five, not four models. That is because the model LED does double duty. Depending on whether the input data deck specifies linear isotherm equilibrium or or Freundlich isotherm equilibrium, the model operates under the name linear local equilibrium with dispersion (L-LED) or Freundlich local equilibrium with dispersion (F-LED).

The heart of each model is a set of governing finite difference equations. These are spatially centered finite difference approximations of the differential equations presented in the previous chapter. The equations that appear here are not the direct finite difference analogies to the governing equations, but instead the "sorted" versions. That is, coefficients associated with the dependent variable at like nodes appear together. DGEAR, an IMSL subroutine, handles the time-stepping tasks. DGEAR is a pre-packaged FORTRAN subroutine that solves differential equations using Gear's method.

For LED the governing equation (II-8) becomes

$$\frac{dC_i}{dt} = \frac{1}{R_{f,i}} \left[\left(\frac{D_h}{\Delta z^2} + \frac{v_z}{2\Delta z} \right) C_{i-1} + \left(- \frac{2\ D_h}{\Delta z^2} \right) C_i + \left(\frac{D_h}{\Delta z^2} - \frac{v_z}{2\Delta z} \right) C_{i+1} \right] \qquad \text{(III-1)}$$

where

$$R_{f,i} = 1 + \frac{\rho_b}{\theta} \ nK_F \ C_i^{n-1} \ ,$$

i = column node index.

For each of the other models there are two governing equations, a solid-phase and a fluid-phase equation. The LSO fluid-phase equation (II-12) becomes

$$\frac{dC_i}{dt} = \left(\frac{D_h}{\Delta z^2} + \frac{v_z}{2\Delta z}\right) C_{i-1} + \left(-\frac{2D_h}{\Delta z^2}\right) C_i \qquad \text{(III-2)}$$

$$+ \left(\frac{D_h}{\Delta z^2} - \frac{v_z}{2\Delta z}\right) C_{i+1}$$

$$- \frac{\rho_b}{\theta} k_s[(Q^o - q_i) C_i - \frac{q_i}{b}] \quad .$$

The LSO solid phase equation (II-10) becomes

$$\frac{dq_i}{dt} = k_s[(Q^o - q_i) C_i - \frac{q_i}{b}] \quad . \qquad \text{(III-3)}$$

The FED fluid-phase equation (II-17) becomes

$$\frac{dC_i}{dt} = \frac{1}{R_{f,f,i}} \left(\frac{D_h}{\Delta z^2} + \frac{v_z}{2\Delta z}\right) C_{i-1} + \frac{1}{R_{f,f,i}} \left(-\frac{2D_h}{\Delta z^2}\right) C_i$$

$$+ \frac{1}{R_{f,f,i}} \left(\frac{D_h}{\Delta z^2} - \frac{v_z}{2\Delta z}\right) C_{i+1}$$

$$- \frac{\rho_b}{R_{f,f,i}\theta} [\alpha(K_{F,s}C_i^{n_s} - q_s)] \quad . \qquad \text{(III-4)}$$

The FED solid phase equation (II-18) becomes

$$\frac{dq_{s,i}}{dt} = \alpha(K_{F,s} \; C_i^{n_s} - q_{s,i}) \quad . \tag{III-5}$$

The DUAL fluid-phase equation (II-27) becomes

$$\frac{dC}{dt} = \left(\frac{D_h}{\Delta z^2} + \frac{v_z}{2\Delta z}\right) C_{i-1} + \left(- \frac{2D_h}{\Delta z^2} - \frac{3k_f(1-\theta)}{R\,\theta}\right) C_i$$

$$+ \left(\frac{D_h}{\Delta z^2} - \frac{v_z}{2\Delta z}\right) C_{i+1} + \left(\frac{3k_f(1-\theta)}{R\theta}\right) C_{s,i} \quad . \tag{III-6}$$

The DUAL solid phase equation (II-21) becomes

$$\frac{dq_r}{dt} = \left(\frac{D_{s_-}}{\Delta r \Delta r_-} - \frac{D_s}{r\Delta r}\right) q_{r,l-1} + \left(- \frac{D_s}{\Delta r\Delta r_-} - \frac{D_{s_+}}{\Delta r\Delta r_+}\right) q_{r,l}$$

$$+ \left(\frac{D_{s_+}}{\Delta r\Delta r_+} + \frac{D_s}{r\Delta r}\right) q_{r,l+1} \tag{III-7}$$

where

l = radial node index;

$$\Delta r = \frac{\Delta r_+ + \Delta r_-}{2} \quad ;$$

$$\Delta r_- = \frac{r_l - r_{l+1}}{2} \quad ;$$

$$\Delta r_+ = \frac{r_{l+1} - r_l}{2} \quad .$$

The changes to the existing programs follow.

1. Commenting of programs

The first change made to the four existing programs was the addition of comment lines. These comment lines did not, of course, affect the operation of the programs, but they did improve the clarity of the programs. The task also familiarized the student with the programs.

2. Transportability of programs

The second change was the conversion of the programs from personal computer programs to a form that could easily operate in either a mainframe or2 microcomputer environment. The increased solution speed was a welcome improvement, particularly for models such as DUAL which required many hours to execute, even on a PC-AT. The needed changes included restructuring of the program common blocks and development of easily redirectable input and output units. Several mainframe versions also ran on the Triangle University Computing Center's Floating Point Systems - 164 scientific computer. The FPS-164 offered no improvement in speed using the standard FORTRAN code developed for the mainframe and microcomputer. Improved FPS-164 performance would have required code specifically designed to take advantage of the FPS-optimized library subroutines. Program transportability took precedence over speed so further programming efforts stayed in the mainframe and microcomputer environments.

An early task performed to learn about the relative usefulness of the various computer environments was benchmark testing. Table III-1 presents the performance of a benchmark run in three different environments: the PC-AT, an ordinary PC with a math coprocessor, and the mainframe. The comparison attempted to isolate CPU time from input/output time in order to report just the time spent doing the math.

Table III-1

BENCHMARK TESTING IN VARIOUS COMPUTER ENVIRONMENTS

Environment	Program	Solution Time (HH:MM:SS)
PC	LED	00:05:07
PC	LSO	00:05:13
PC	FED	00:14:12
PC	DUAL	NOTE 1
PC-AT	LED	00:02:31
PC-AT	LSO	00:02:20
PC-AT	FED	00:05:22
PC-AT	DUAL	>09:00:00 NOTE2
MAINFRAME	LED	00:00:05
MAINFRAME	LSO	00:00:04
MAINFRAME	FED	00:00:22
MAINFRAME	DUAL	00:06:49

NOTE 1: THE DUAL EXECUTABLE MODULE IS TOO LARGE TO FIT ON A FLOPPY DISK. SINCE IT CANNOT FIT ON A DISK IT CANNOT RUN.

NOTE 2: EVEN AN OVERNIGHT RUN DID NOT ALLOW SUFFICIENT TIME FOR PROGRAM COMPLETION.

3. Variation of influent

As the present research began, the computer simulation models assumed constant soil column influent concentration.

The third change and the first major alteration of the codes was the addition of variable influent concentration capability. The modifications to the code allowed the programs to accept both minor fluctuations in influent concentration and drastic changes in influent concentration. The user could now specify a constant influent concentration or precisely steer the influent concentration. The effluent boundary condition, the partial of concentration with respect to location equals zero, remained unchanged. The ability to control precisely the upstream boundary condition was convenient when reproducing small experimental fluctuations in the influent concentration. The control was indispensable when studying sorption/desorption systems where an elutriation phase follows a period of contaminant feed.

A companion modification to the variable influent provision was to add subroutines, INFO1 and INFO2, that provide a detailed report of simulation status. As program complexity increased the likelihood of blunders increased dramatically. INFO1 and INFO2 helped to counteract this natural tendency by displaying such relevant factors as influent concentrations and the derivative of influent concentration with respect to time to assure that the programs were functioning properly, and to aid in troubleshooting if they were not.

4. Variation of velocity

The fourth change was to accommodate a single update in the soil column flow velocity at the time of contaminant shutoff, should one occur. This was required because of a peculiarity in the laboratory experimental design which resulted in a change in contaminant velocity when contaminant flow stopped and elutriation began.

5. Calculation of mass balance

The fifth and final change was the addition of an automatic mass balance check. Numerical models, even ones that are technically validated, always threaten an unexpected numerical breakdown. These breakdowns were a constant source of worry for the program operator since they were not always obvious. Subroutines MASS1 and MASS2 provided relief. The subroutines kept track of the contaminant mass in the influent, contaminant mass in the effluent, contaminant left in the soil, and contaminant left in the fluid. By computing the ratio of input to accounted mass at the end of each soil column simulation, these subroutines assured the program operator that the just-completed run operated smoothly, at least to the extent that matter had neither been created nor destroyed.

C. Studies with the numerical models

A six-step protocol for model development and application drew ideas about the modeling process from a paper by Thomann (1982).

MODEL DEVELOPMENT PROTOCOL

1. Formulate a model -- a set of governing differential equations plus boundary conditions -- from consideration of fundamental principles. Recast the model in terms of a numerical scheme for solution.
2. Check the numerical model in an analytically tractable scenario against an analytic solution. If it checks, the model is technically validated.
3. Isolate and identify as many parameters as possible through independent parameter determination.
4. Use an experimentally derived data set and force a fit using parameters that could not be independently established. Check that the parameters are plausible. If the parameters are plausible and the prediction fits, the model is calibrated, and operationally validated.
5. Take at least one more data set from a different set of conditions and make a predictive run. If the prediction fits the new data, the model is dynamically validated.
6. After every run perform a mass balance. This affords yet another partial technical validation each time the model runs.

Upon completion of the full set of activities from the model development protocol at the lab scale, the research returned to the original question of purge well analysis by preparing an illustrative field-scale example. Unfortunately, the lack of actual field-scale data prevented application of the entire modeling process. The model was, however, well suited to demonstrate the effect of the scale of the problem on the prediction of the model.

The application of these modeling activities to Miller's soil-column laboratory experiments and the illustrative field scale model compose the subject matter of the next chapter.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Laboratory-scale analysis

This section details the results of the model development protocol for the laboratory-scale analysis.

Technical validation precedes actual model use. One way to make this validation is to compare predictions of the numerical models against predictions of existing analytical simulators from Miller (1984). When the predictions match, it demonstrates that the central components of the numerical models function correctly. Because this research uses existing programs, the use of the technical validation is a conservative approach.

The breakthrough curves produced in simplified, analytically tractable situations follow in Figures IV-1 through IV-8. The curves appear in matched pairs that present a numerical prediction and the corresponding analytic prediction. The LED and FED predictions depict a situation of contamination followed by elutriation. The LSO and DUAL predictions depict a continuous contaminant input.

FIGURE IV-1

LED VALIDATION: ANALYTIC SOLUTION

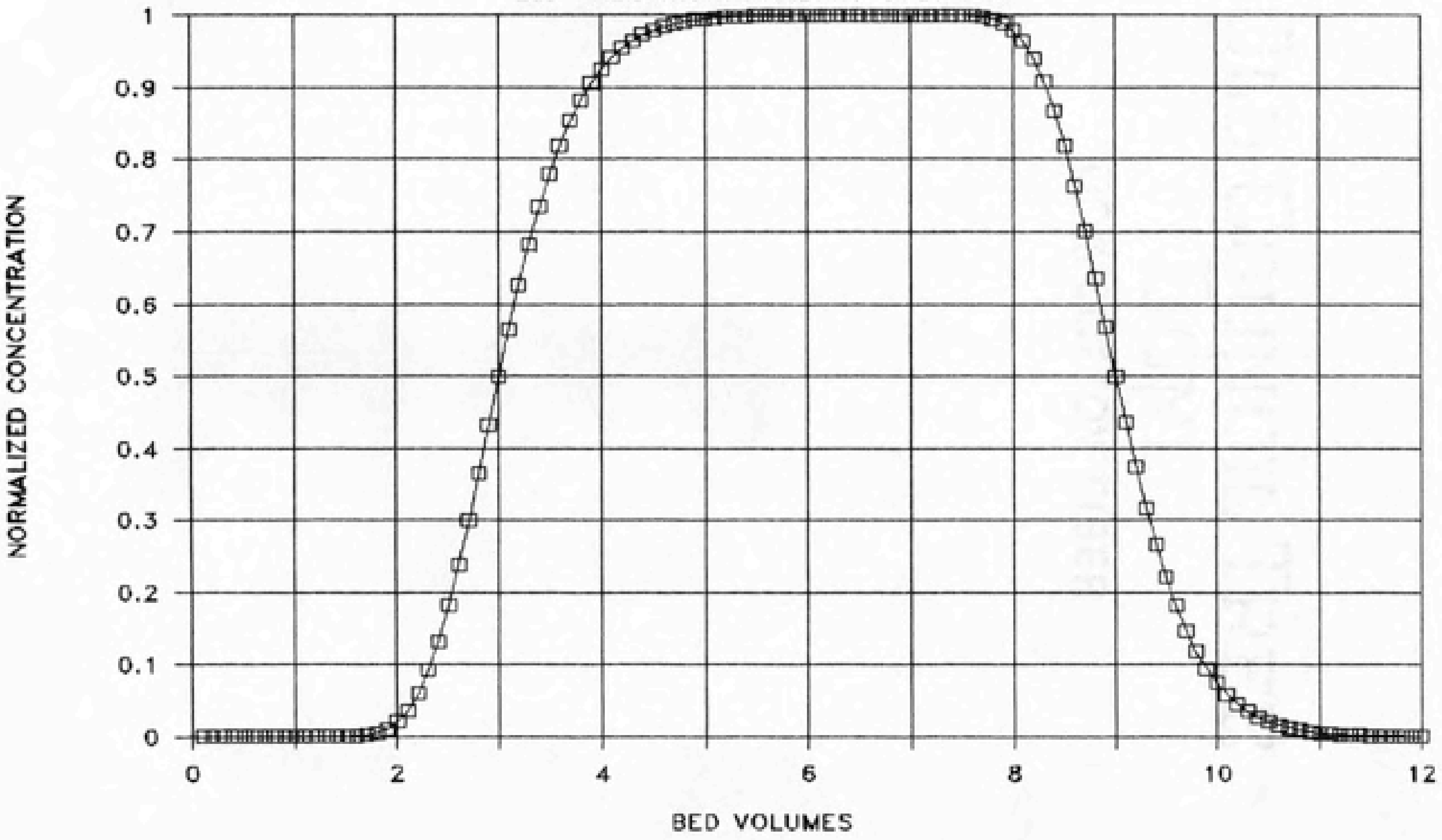


FIGURE IV-2

LED VALIDATION: NUMERICAL PREDICTION

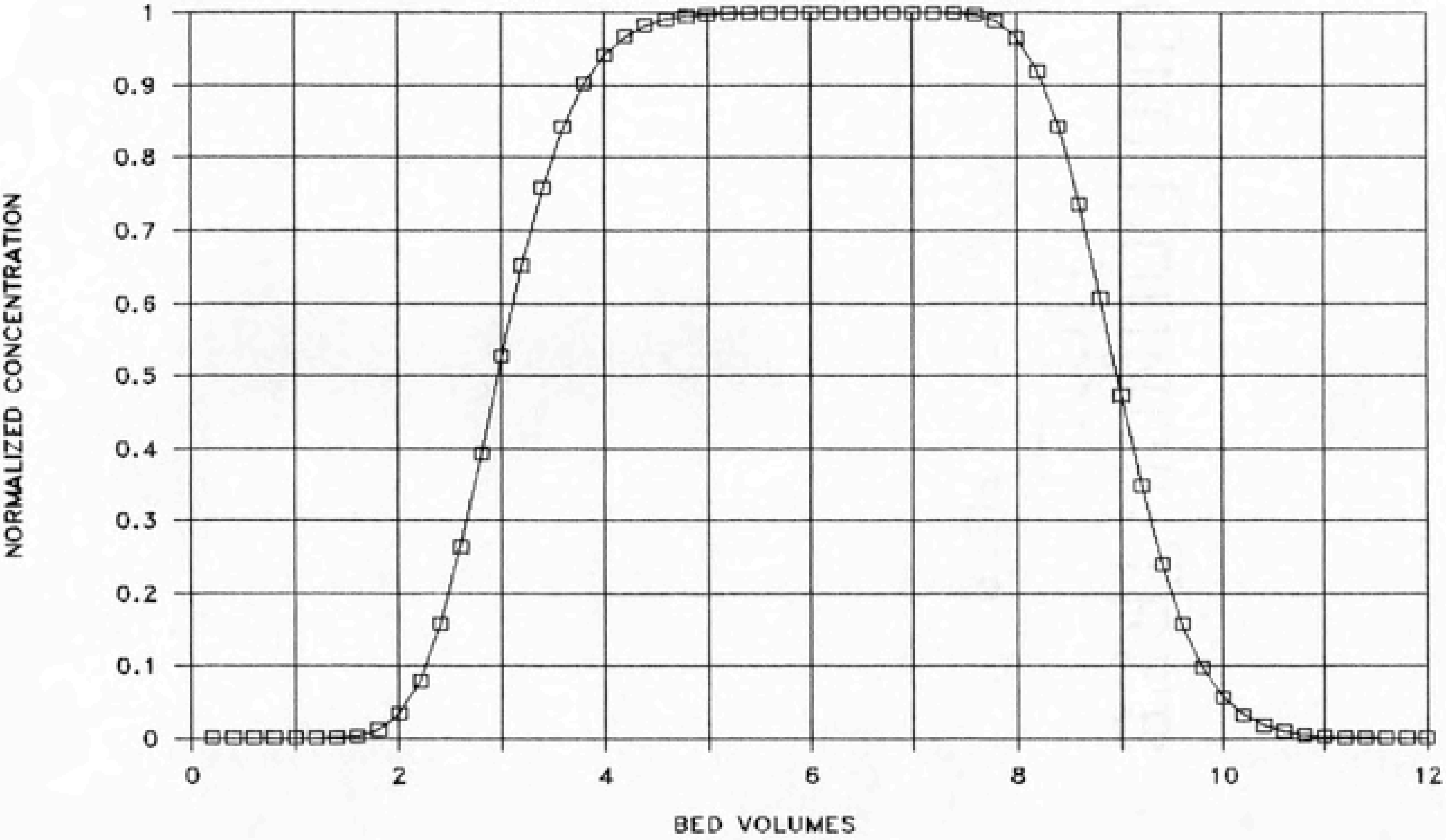


FIGURE IV-3

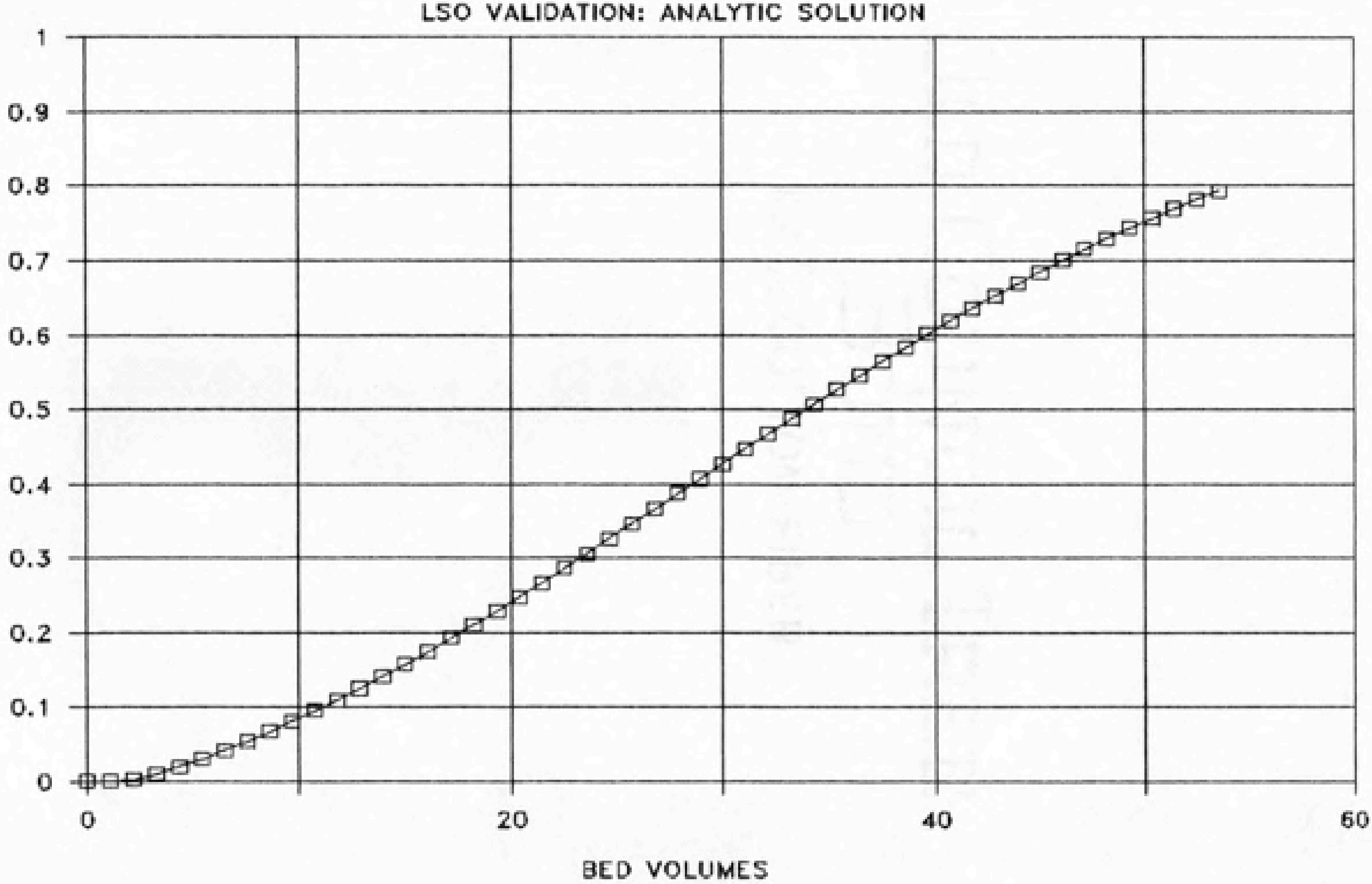

LSO VALIDATION: ANALYTIC SOLUTION
NORMALIZED CONCENTRATION
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
BED VOLUMES

FIGURE IV-4

LSO VALIDATION: NUMERICAL PREDICTION

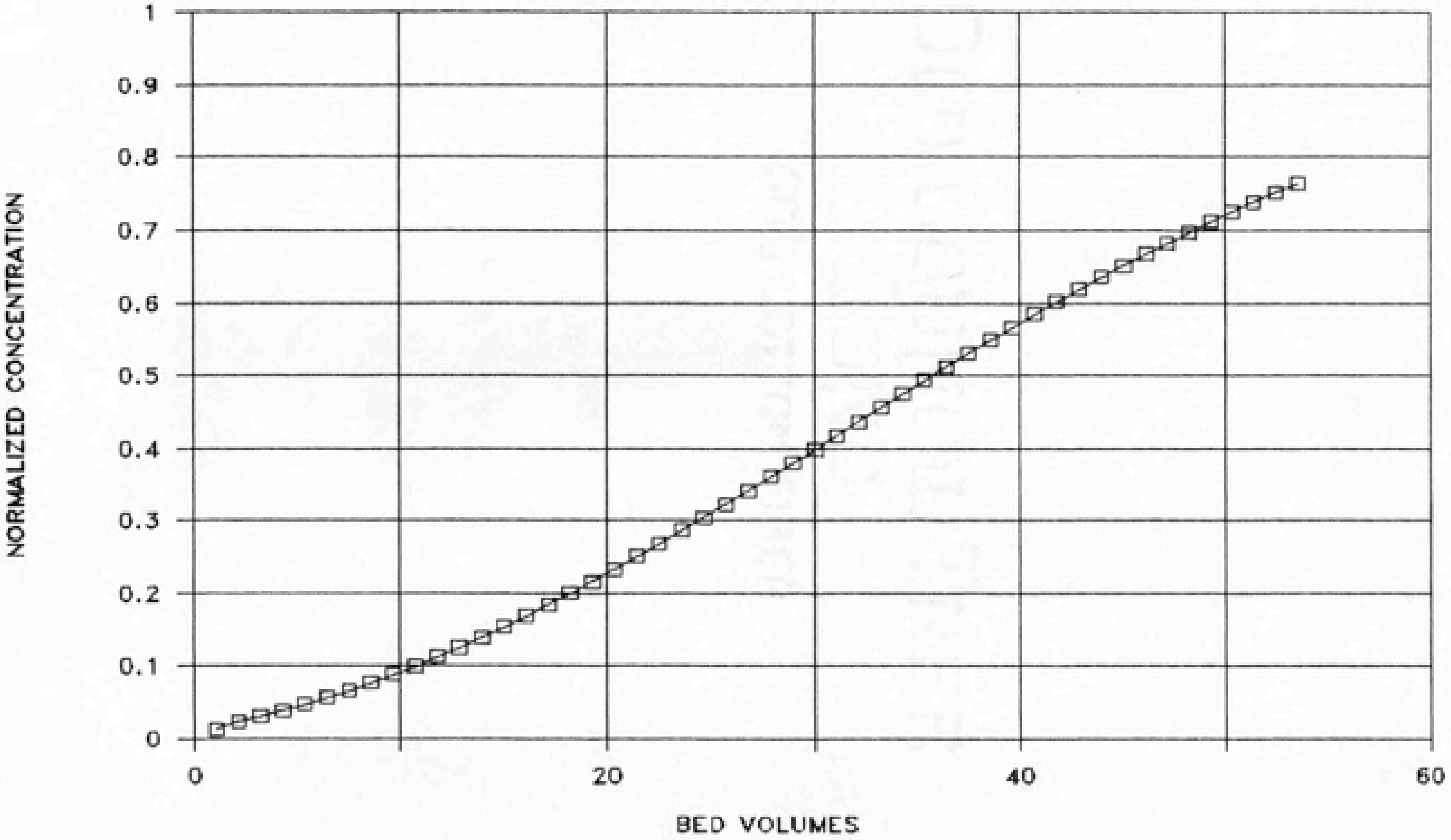


FIGURE IV-5

FED VALIDATION: ANALYTIC SOLUTION

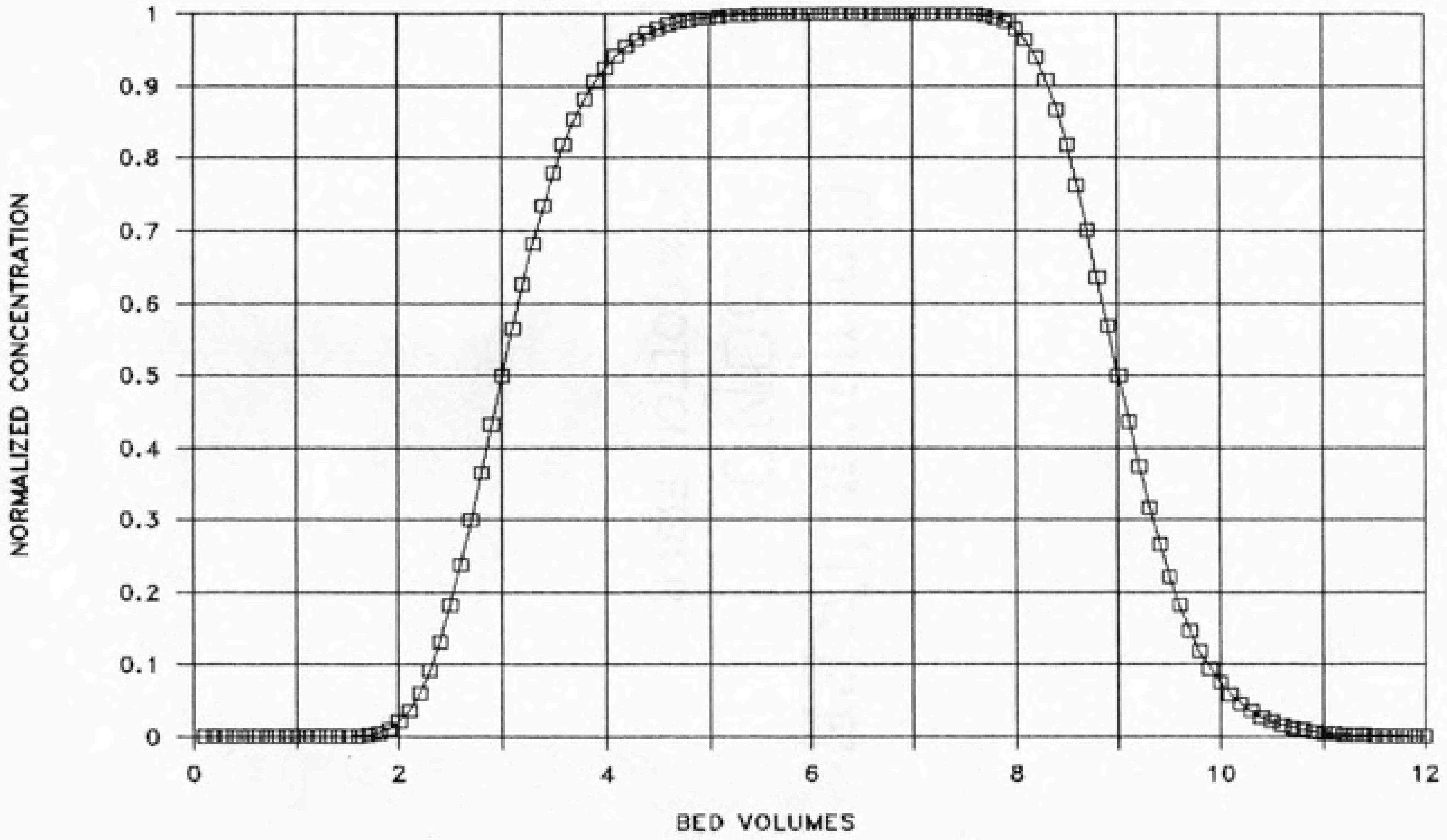


FIGURE IV-6

FED VALIDATION: NUMERICAL PREDICTION

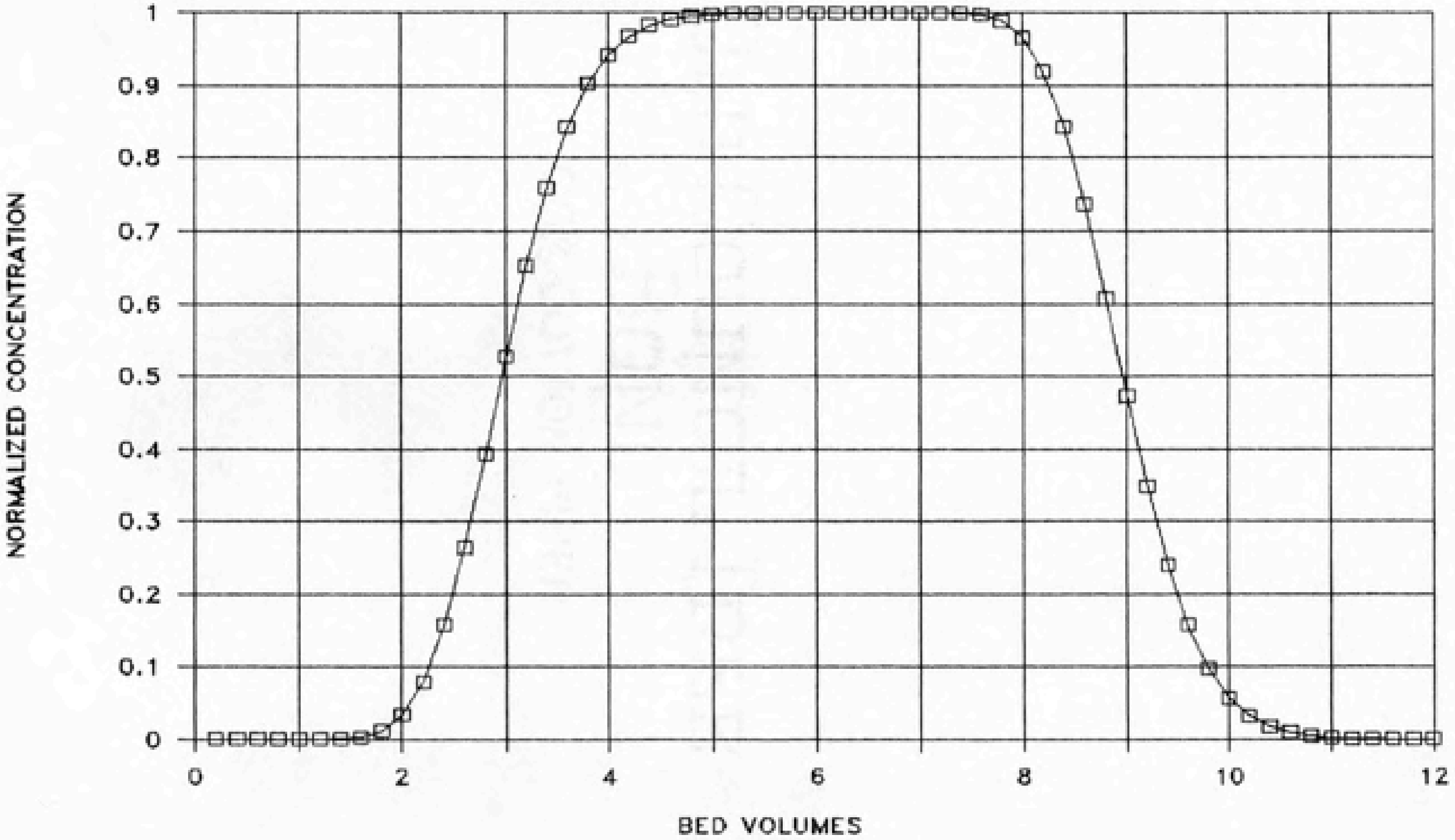


FIGURE IV-7

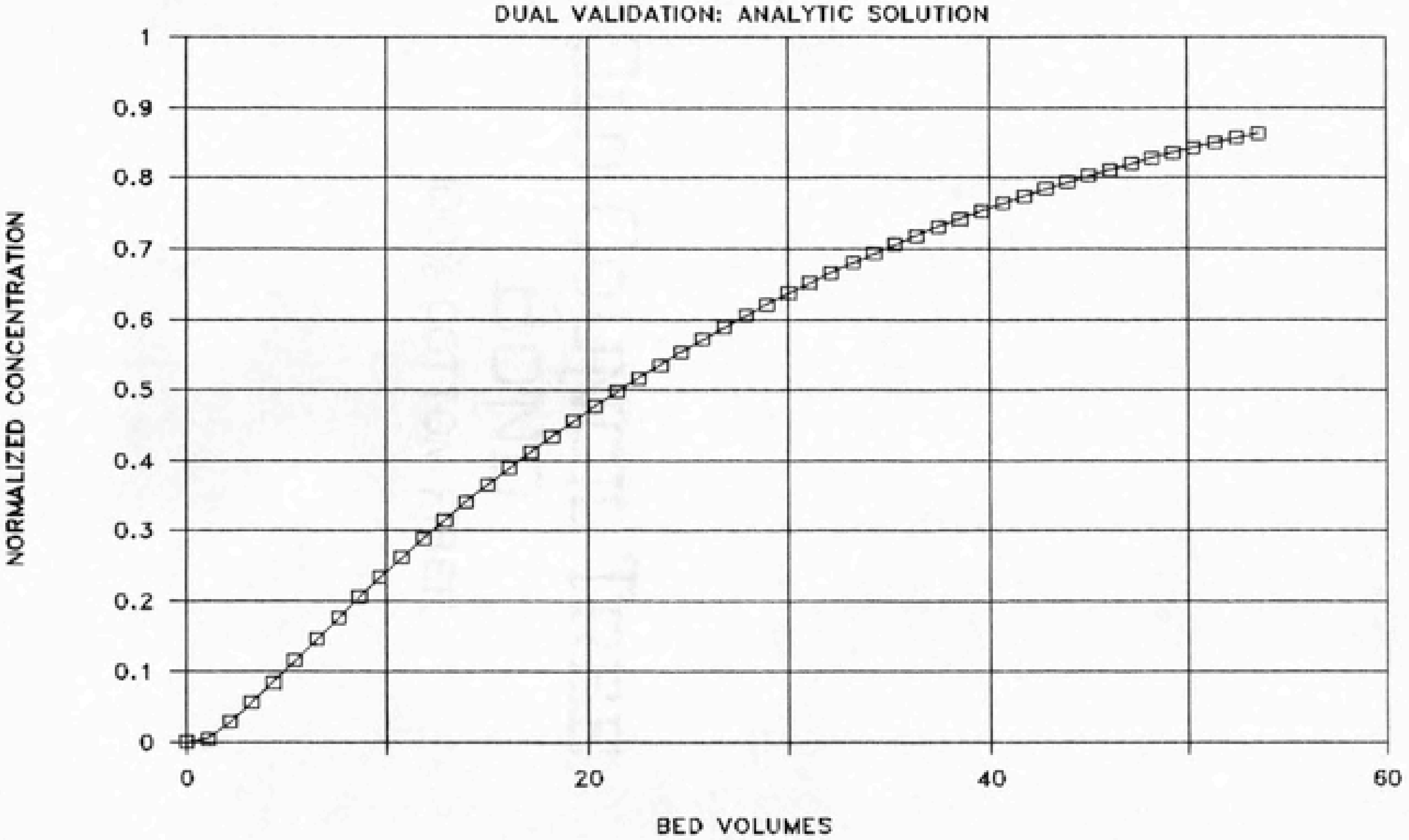
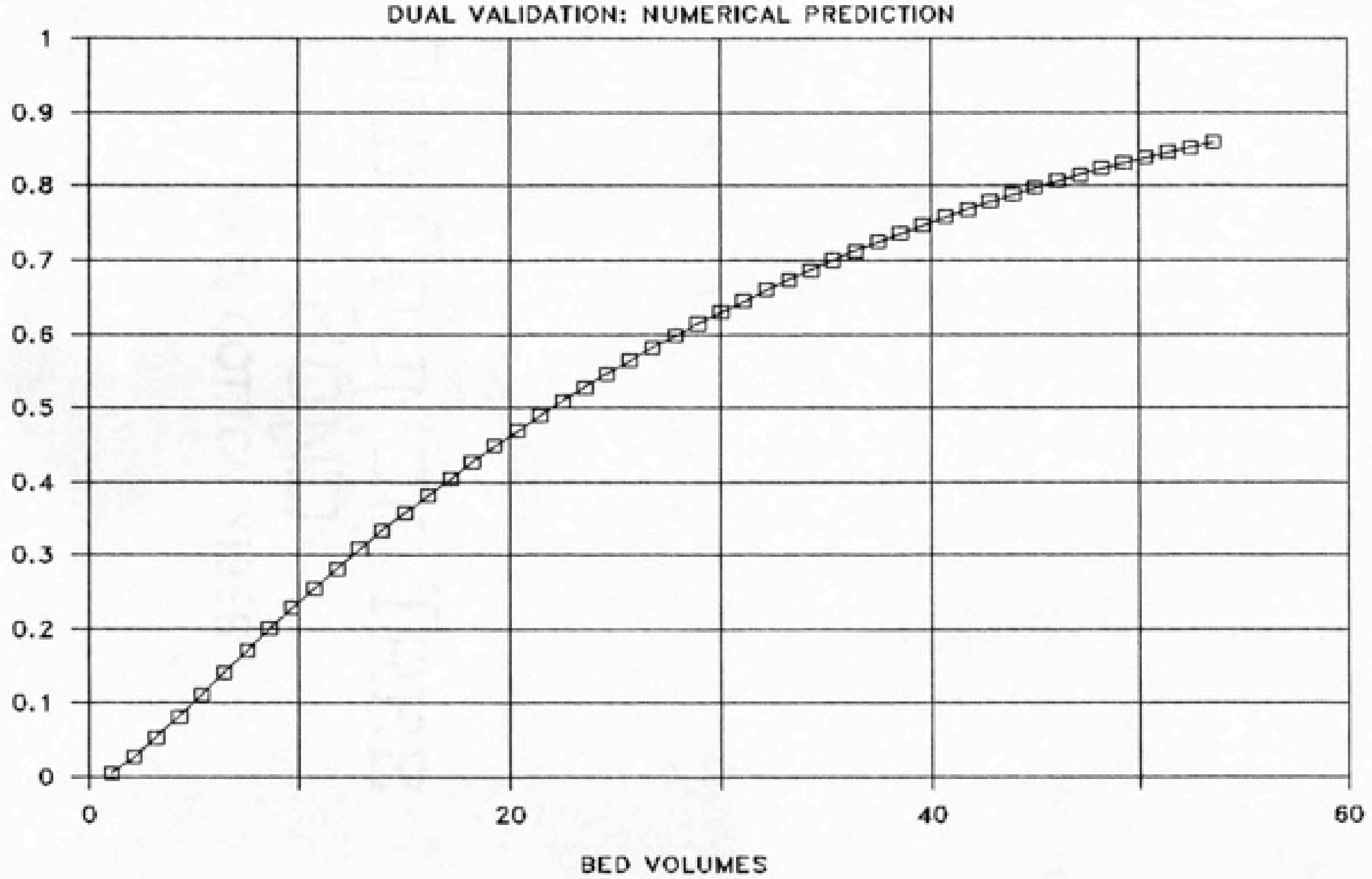

DUAL VALIDATION: ANALYTIC SOLUTION
NORMALIZED CONCENTRATION
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
BED VOLUMES

FIGURE IV-8


DUAL VALIDATION: NUMERICAL PREDICTION
NORMALIZED CONCENTRATION
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
BED VOLUMES

The analytic and numerical models match almost perfectly; the models are technically validated.

The next step -- isolation and independent estimation of as many parameters as possible -- begins with characterization of system hydrodynamics. The system is a suite of three soil column reactors packed with Ann Arbor granular aquifer material. The operational names of these columns are column 15-1, column 16-1, and column 16-2.

Miller's laboratory tracer tests conducted with the chloride ion provide the means to characterize each column's hydrodynamics. An automatic nonlinear parameter estimator program attached to the front end of one of the improved simulator programs (LED) drives the simulator. The parameter estimator program uses the IMSL subroutine ZXSSQ. ZXSSQ uses the Levenberg-Marquardt algorithm for nonlinear optimization. The estimator drives the simulator, varying only hydrodynamic dispersion until the sum of the squares of the residuals of the model prediction with respect to the observed data reach a minimum.

Limitations on the size of the DUAL program restrict analysis to models with 21 column nodes, although for ideal numerical model operation theoretical considerations would call for as many as 36 column nodes. A comparison with a theoretically more favorable situation not presented here demonstrates that the consequences of the 21-node limitation are minimal. Figures IV-9 through IV-11 illustrate the best- fit tracer test breakthrough curves for columns with 21 nodes.

Table IV-1 presents the optimal hydrodynamic dispersion coefficient and dispersivity for each column as determined by this research.

$$\alpha_L = \frac{D_h}{v} \qquad \text{(IV-1)}$$

where

α_L = longitudinal dispersivity (cm)

Dispersivity measures the tendency of the aquifer material to cause dispersion and is independent of ground water velocity if molecular diffusion is negligible.

FIGURE IV-9

TRACER TEST 15-1 FIT WITH 21 NODES

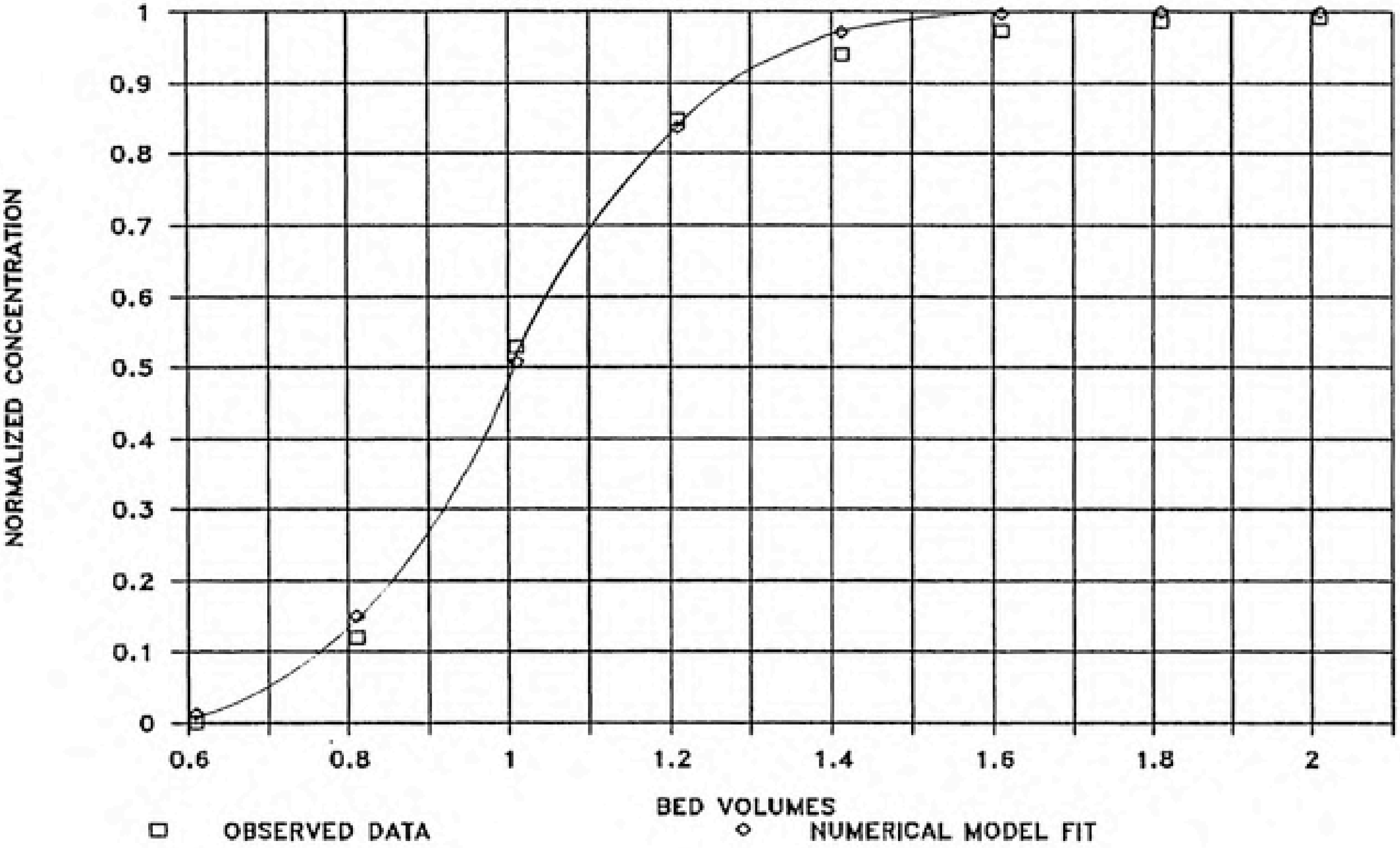


FIGURE IV-10

TRACER TEST 16-1 FIT WITH 21 NODES

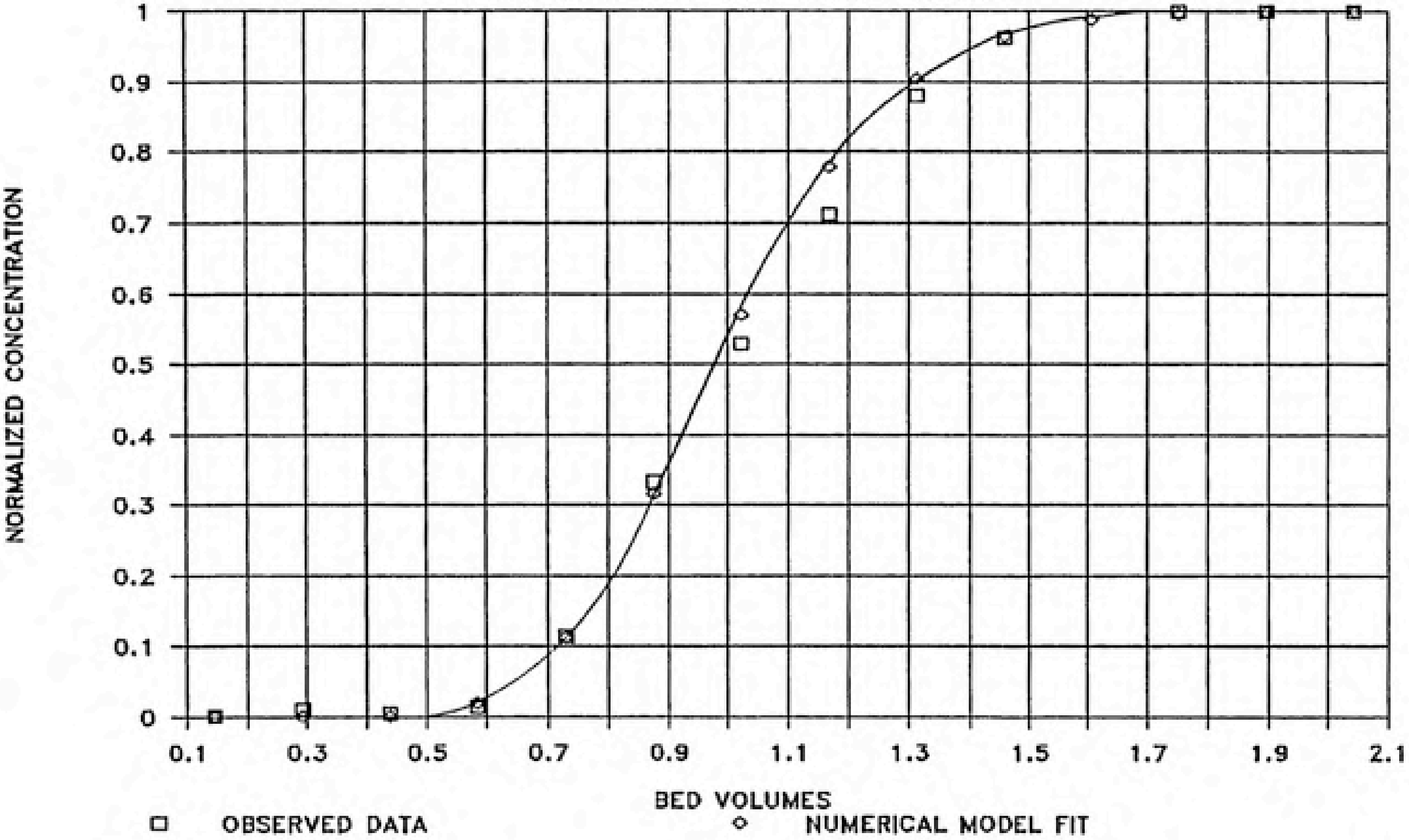
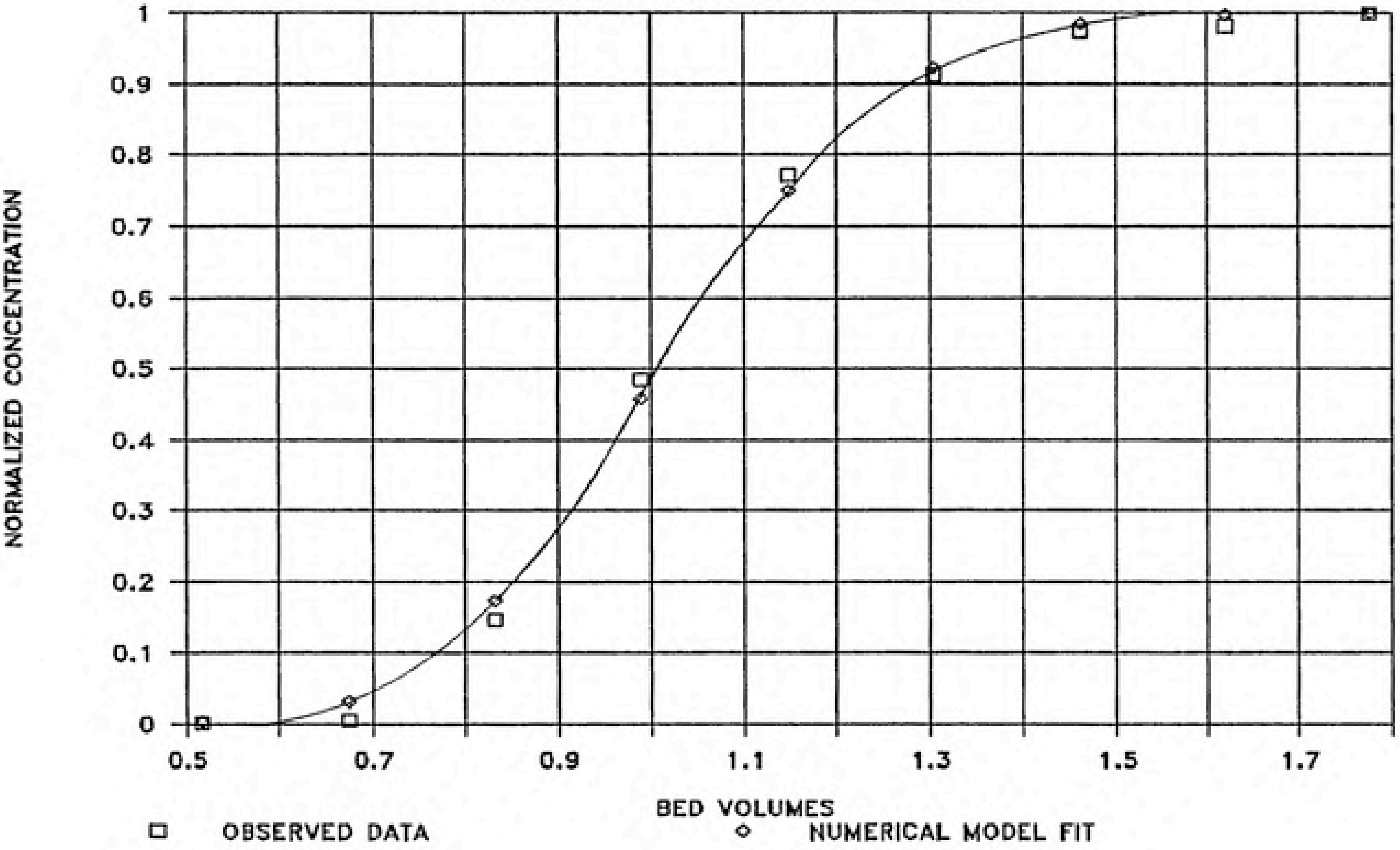


FIGURE IV-11



TRACER TEST 16-2 FIT WITH 21 NODES

Table IV-1

Fitted Hydrodynamic Parameters

The table lists optimal hydrodynamic parameters for columns modeled with 21 nodes.

Column Number	Hydrodynamic Dispersion Coefficient (cm^2/hr)	Longitudinal Dispersitivity (cm)
15-1	2.22	0.453
16-1	1.86	0.587
16-2	1.34	0.391

The tracer test results, together with Miller's soil analysis, bottle-point studies, and completely mixed batch-reactor studies yield a nearly complete set of parameters for modeling the breakthrough of nitrobenzene in the soil columns. A complete parameter set already exists for L-LED, F-LED, and LSO. FED and DUAL each lack one parameter. FED lacks the relative distribution of fast sorption sites to slow sorption sites. DUAL lacks the film transport coefficient.

Table IV-2 summarizes the sorption parameter information that is available for nitrobenzene on Ann Arbor granular aquifer material at the outset of the nitrobenzene/Ann Arbor column modeling.

Table IV-2

Laboratory Sorption Parameters Known at Start of Sorption Study

Parameter	Value
Linear sorption isotherm model coefficient	6.69 (cm^3/g)
Freundlich isotherm sorption-capacity constant	0.217 ($(cm^3/g)^{0.73}$)
Freundlich isotherm sorption-energy constant	0.73 (Dimensionless)
Equilibrium/first-order model mass-transfer coefficient	3.32×10^{-2} (1/hr)
Langmuir isotherm sorption-capacity constant	4.40×10^{-5} (g/g)
Langmuir isotherm sorption-energy constant	$2.91 \times 10^{+5}$ (cm^3/g)
Second-order Langmuir model rate constant	$1.86 \times 10^{+4}$ ($cm^3/g/hr$)
Intraparticle surface-diffusion coefficient for dual-resistance model	3.44×10^{-7} (cm^2/hr)

Source: Miller (1984),
Miller and Weber (1984), and
Weber and Miller (in press)

Model calibration to determine the values of the remaining parameters for nitrobenzene modeling follows independent parameter estimation. In order to preserve a fair comparison among the models, the runs treat one parameter from each model as an unknown. This allows L-LED, F-LED, and LSO to share the improved prediction capability afforded by calibration. The runs treat hydrodynamic dispersion as unknown for L-LED and F-LED, and treat the second-order rate constant as unknown for LSO.

Calibration begins with the first nitrobenzene / Ann Arbor data set, number 15-1. Calibration fits the one missing parameter for each model. The objective function is to minimize the sum of squares of model residuals. The IMSL automatic estimator ZXSSQ encounters operational difficulties for LSO and FED models so progressively finer grid searches find the best fit. A grid search on k_f also finds the best fit for the DUAL model because an automatic parameter estimation would require an unacceptably large amount of computer time -- something in the range of several CPU-hours for a single run.

Table IV-3 presents the findings of the calibration procedure.

Table IV-3

Findings of Sorption Calibration Procedure

Parameter	Value
Longitudinal dispersitivity for linear local equilibrium model	2.59 cm
Longitudinal dispersitivity for Freundlich local equilibrium model	2.27 cm
Second-order Langmuir model rate constant	5900 $cm^3/g/hr$
Freundlich isotherm sorption-capacity constant for the rapid rate component of the equilibrium/first-order rate model	0.00 $(cm^3/g)^{0.73}$
Freundlich isotherm sorption-capacity constant for the slow rate component of the equilibrium/first-order rate model	0.27 $(cm^3/g)^{0.73}$
Film transport coefficient for the dual-resistance model	0.0062 cm/hr

Comparison of Table IV-2 with Table IV-3 provides favorable information: For the three models where parameter estimation is not absolutely necessary, the parameter values do not shift dramatically.

Plotting the model predictions for column 15-1 along with the experimental breakthrough measurements provides means to assess whether the calibrations are successful. Table IV-4 presents the input parameters for each model. Figures IV-12 through IV-16 display the breakthrough curves predicted by the force-fit models.

Table IV-4

Input Parameters for Models of Column 15-1, The Calibration Run

Variable	Value
D_h during contamination	0.2970 E+01 cm^2/hr
D_h for L-LED during contamination	0.1700 E+02 cm^2/hr
D_h for F-LED during contamination	0.1490 E+02 cm^2/hr
θ	0.3730 E+00*
ρ	0.2670 E+01 g/cm^3
v during contamination	0.6560 E+01 cm/hr
v during elutriation	0.5070 E+01 cm/hr
z	0.2450 E+02 cm
K_p	0.6690 E+01 cm^3/g
K_F	0.2170 E+00 $(cm^3/g)^{0.73}$
n	0.7300 E+00*
$K_{F,f}$	0.0000 E+00 $(cm^3/g)^{0.73}$
n_f	0.7300 E+00*
$K_{F,s}$	0.2170 E+00 $(cm^3/g)^{0.73}$
n_s	0.7300 E+00*
α	0.3320 E-01 1/hr
Q^o	0.4400 E-04 g/g
b	0.2910 E+06 cm^3/g
k_s	0.5900 E+04 $cm^3/g/hr$
D_s	0.3440 E-06 cm^3/hr
k_f	0.6200 E-02 cm/hr
R	0.1160 E-01 cm
C_o @ t=0.00 hr	0.1564 E-05 g/cm^3
C_1 @ t=144 hr	0.1310 E-05 g/cm^3
C_2 @ t=191 hr	0.1510 E-05 g/cm^3
C_3 @ t=191+ hr	0.1000 E-19 g/cm^3
C_4 @ t=600 hr	0.1000 E-19 g/cm^3

* dimensionless

FIGURE IV-12

15-1, LINEAR LED, 1 PARAMETER FIT

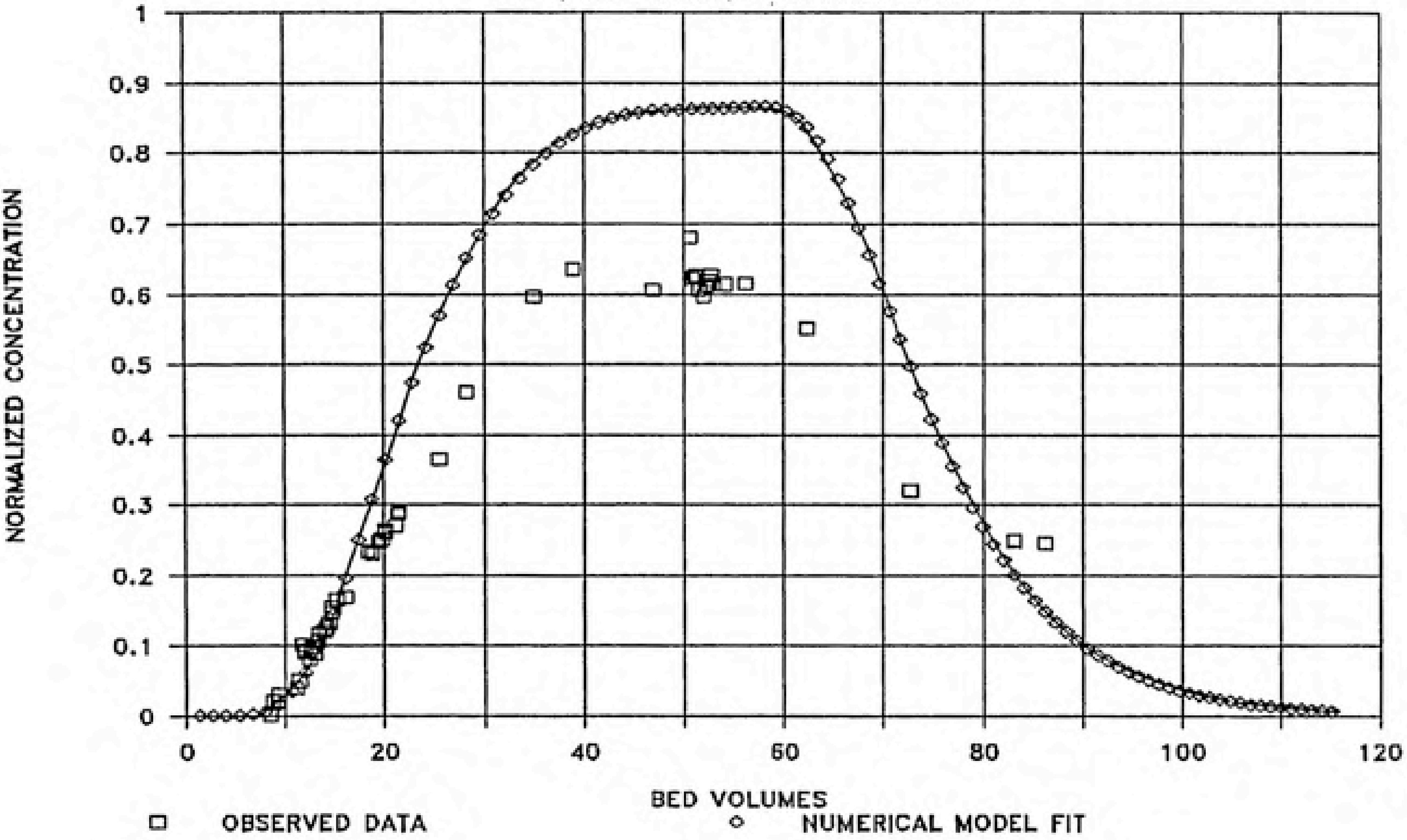


FIGURE IV-13

15-1, FREUNDLICH LED, 1 PARAMETER FIT

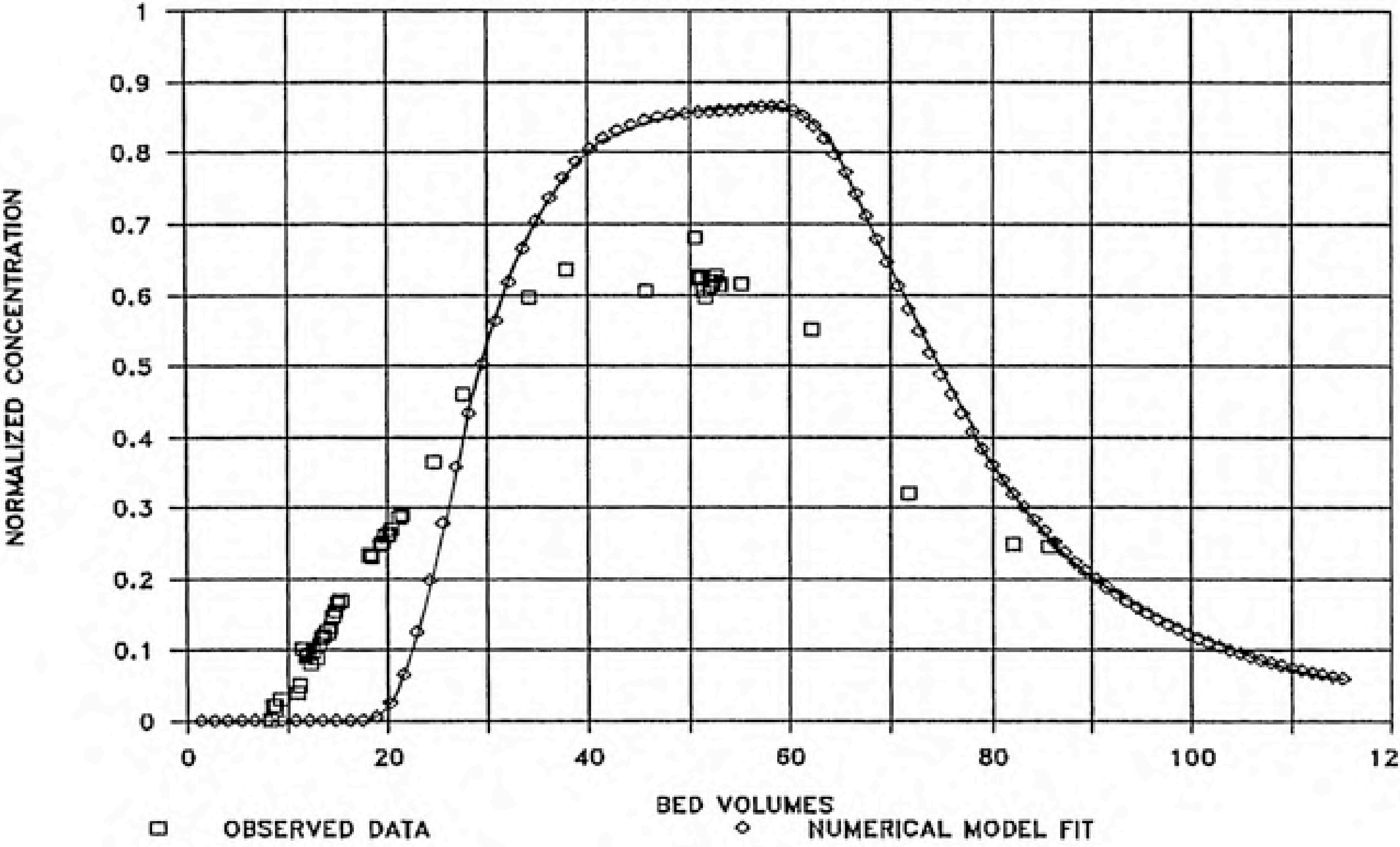


FIGURE IV-14

15-1, LSO, 1 PARAMETER FIT

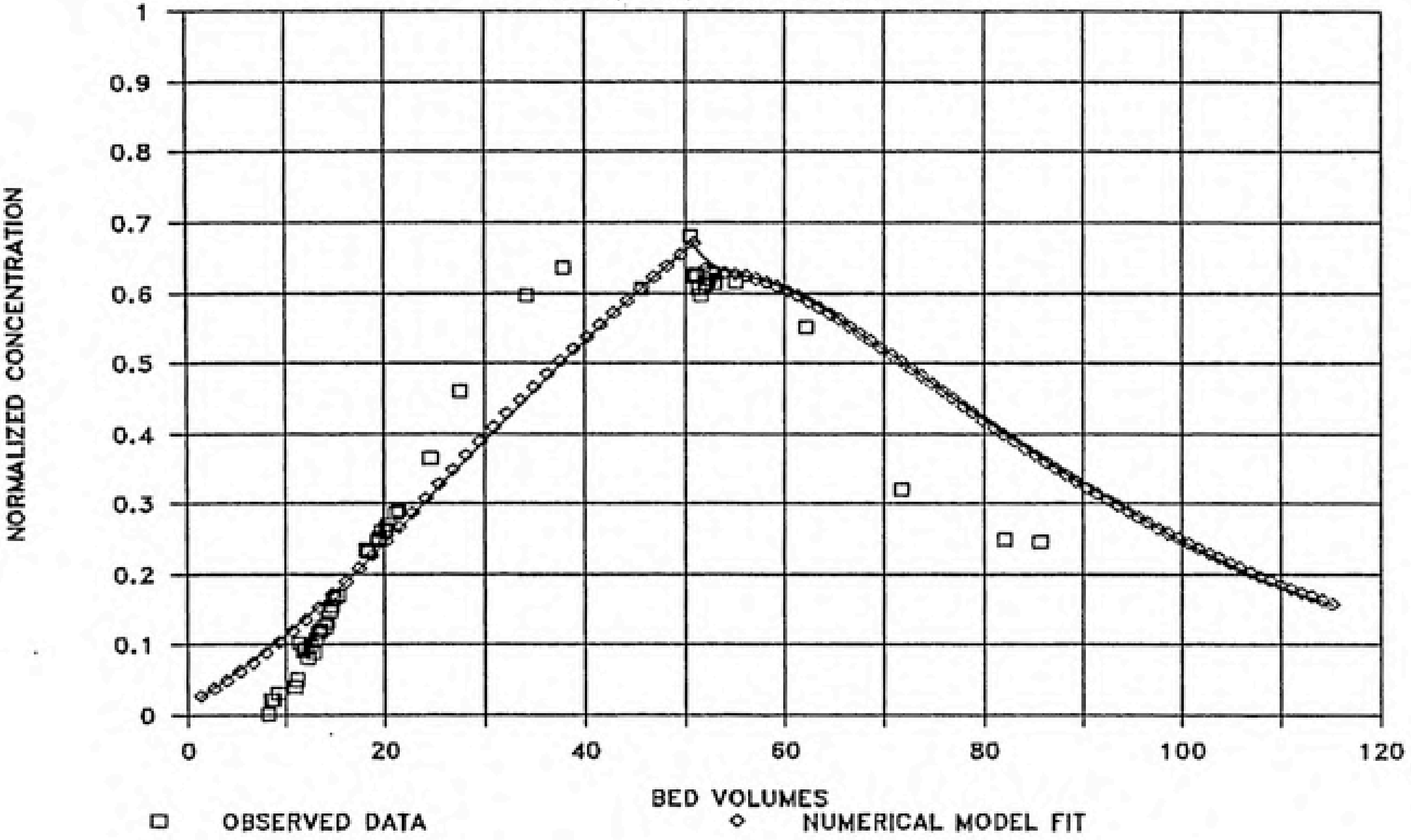


FIGURE IV-15

15-1, FED, 1 PARAMETER FIT

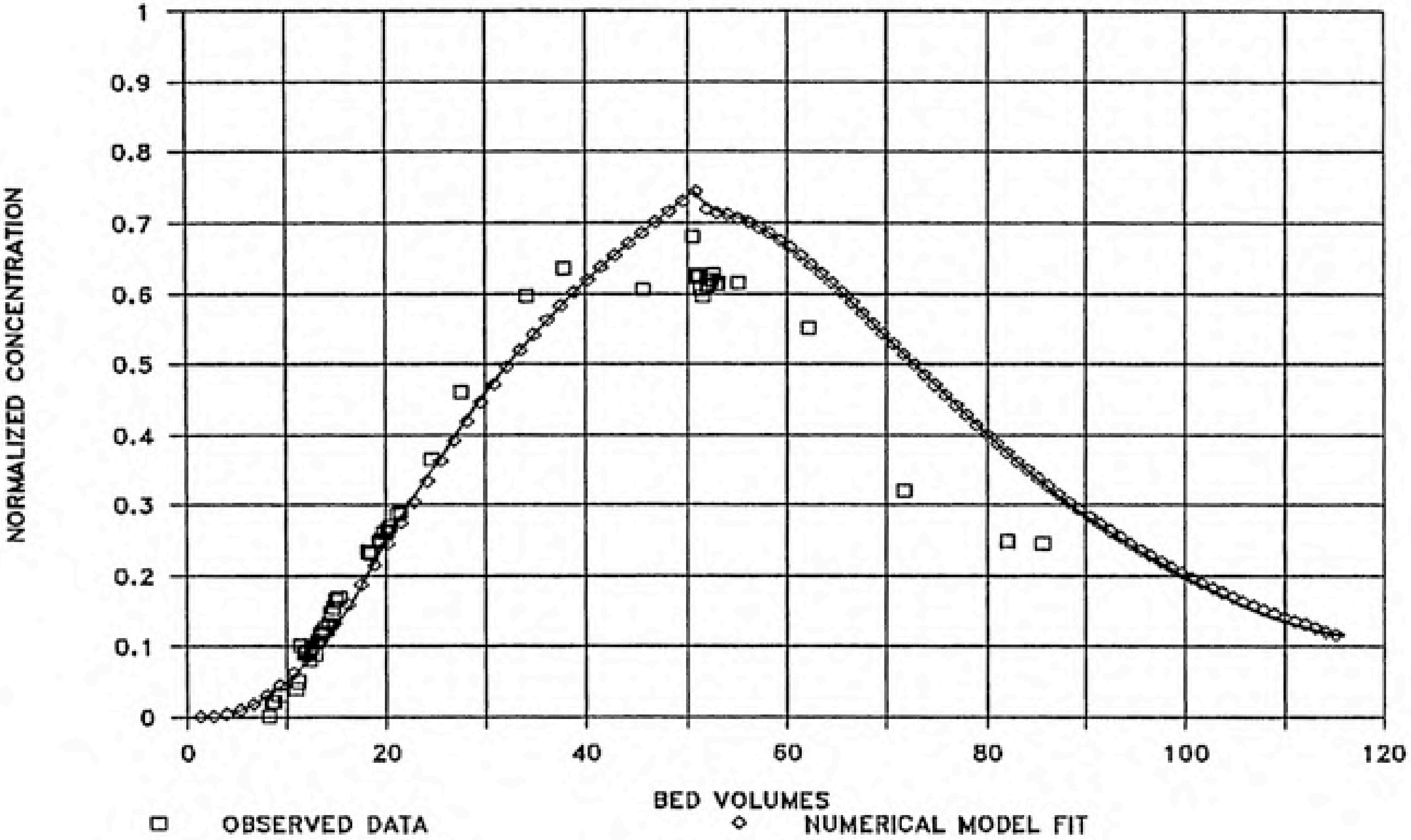
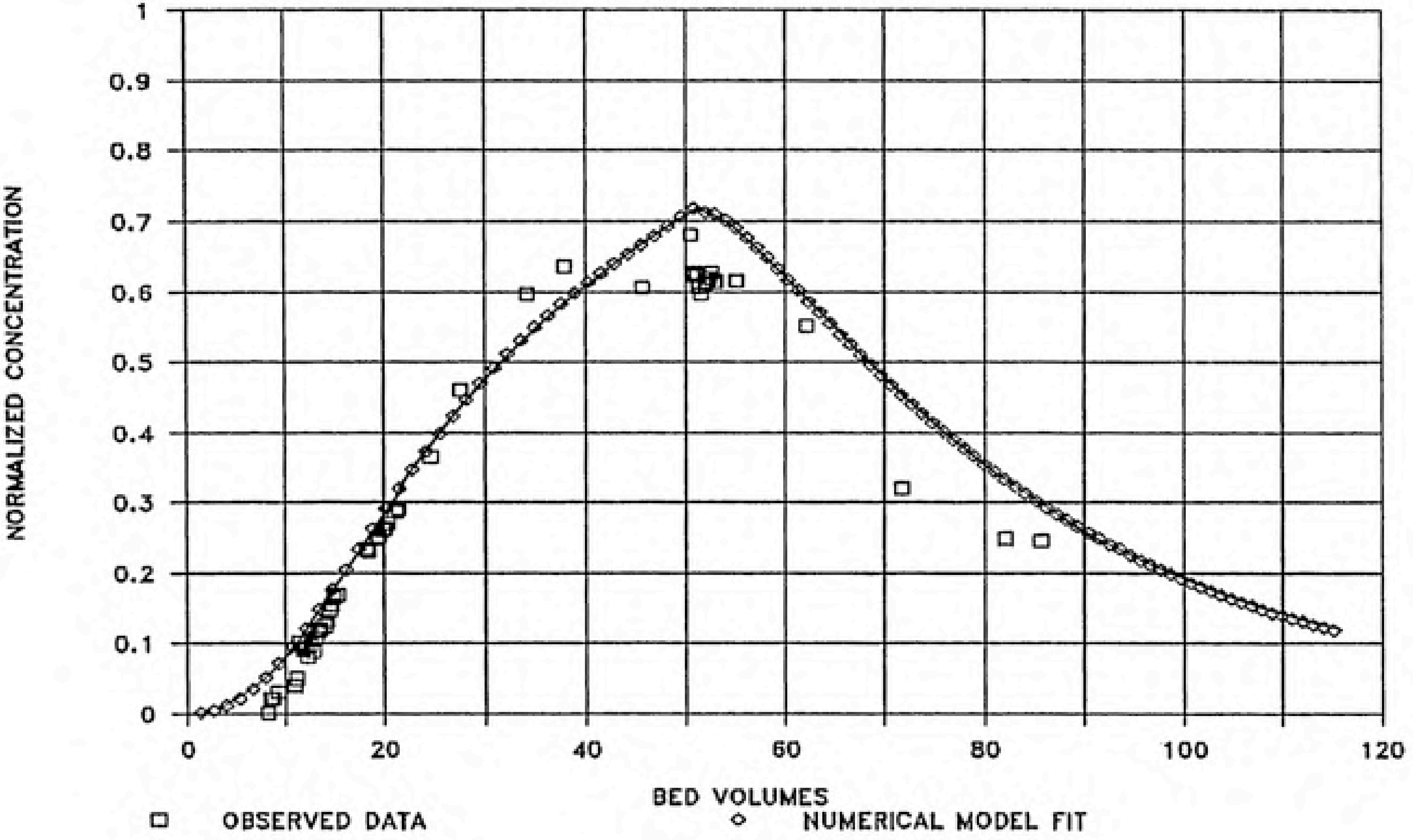


FIGURE IV-16

15-1, DUAL, 1 PARAMETER FIT



The predictions all do a reasonable job of anticipating the timing of the main breakthrough events around 30 and 70 bed volumes. L-LED and F-LED fail to capture either the correct shape of the breakthrough curve or the peak concentration. LSO and FED do better on both counts but still contain periods of consistent overprediction and underprediction. DUAL predictions appear most accurate, capturing both the shape and the timing of the breakthrough curve.

From a more objective standpoint, model variance supports the suggestion that DUAL outperforms LSO and FED which in turn outperform F-LED and L-LED. Model variance equals the sum of the squares of the model residuals divided by the number of observations. Table IV-5 presents the variance of each model.

Table IV-5

Variance of Fitted Models of Column 15-1

Model	Model Variance $(g/cm^3)^2$
L-LED	0.0226
F-LED	0.0332
ISO	0.0036
FED	0.0043
DUAL	0.0033

An additional valuable exercise is to examine the sensitivity of the model predictions to perturbations of the fitted parameters. Figures IV-17 through IV-21 present the result of shifting the fitted parameter in each model up and down one order of magnitude. Figures IV-17 and IV-18 demonstrate the importance of hydrodynamic dispersion in controlling smearing and timing of the breakthrough in the LED models. Figure IV-19 shows the profound effect of the second order constant in LSO on contaminant breakthrough predictions. Figure IV-20 demonstrates how the relative distribution of fast versus slow sites affects FED's predictions. Finally for the DUAL model Figure IV-21 shows the small impact of a film constant that is too large and the large impact of a film constant that is too small, at least for this run. This is an important point. It suggests that the sorption process is internal diffusion controlled.

FIGURE IV-17

15-1, LINEAR LED, DH SENSITIVITY

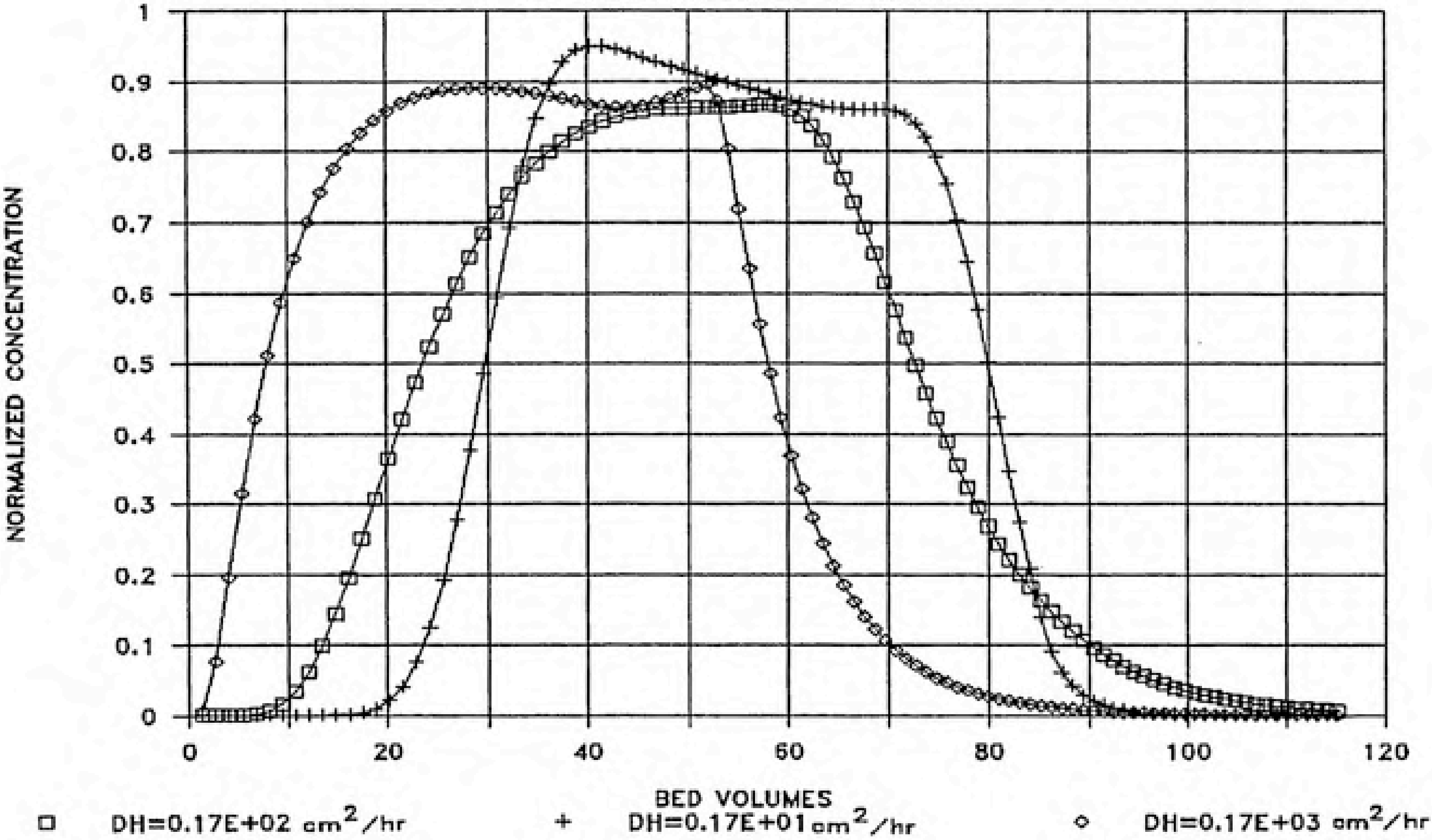


FIGURE IV-18

15-1, FREUNDLICH LED, DH SENSITIVITY

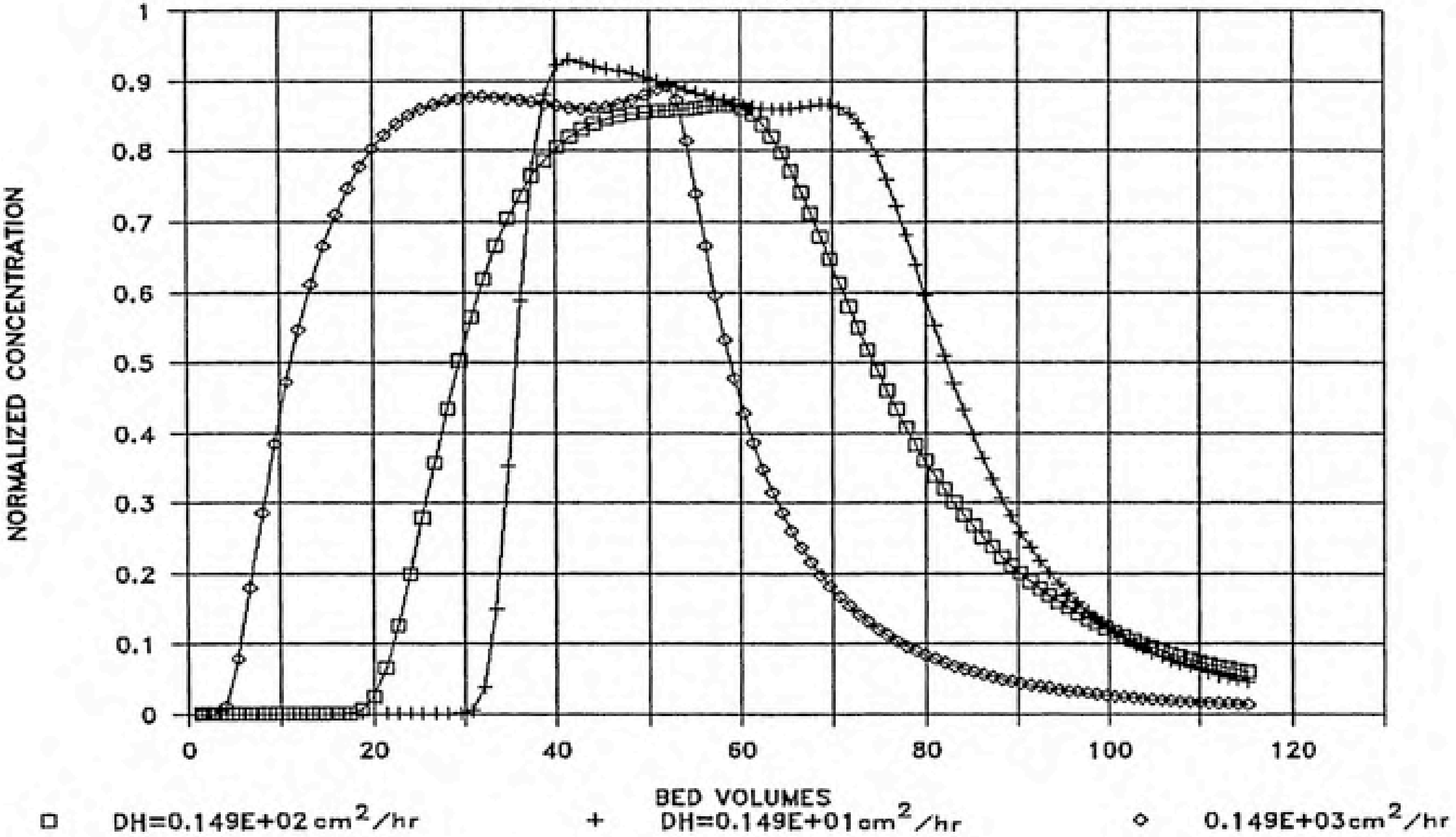


FIGURE IV-19

15-1, LSO, SOK SENSITIVITY

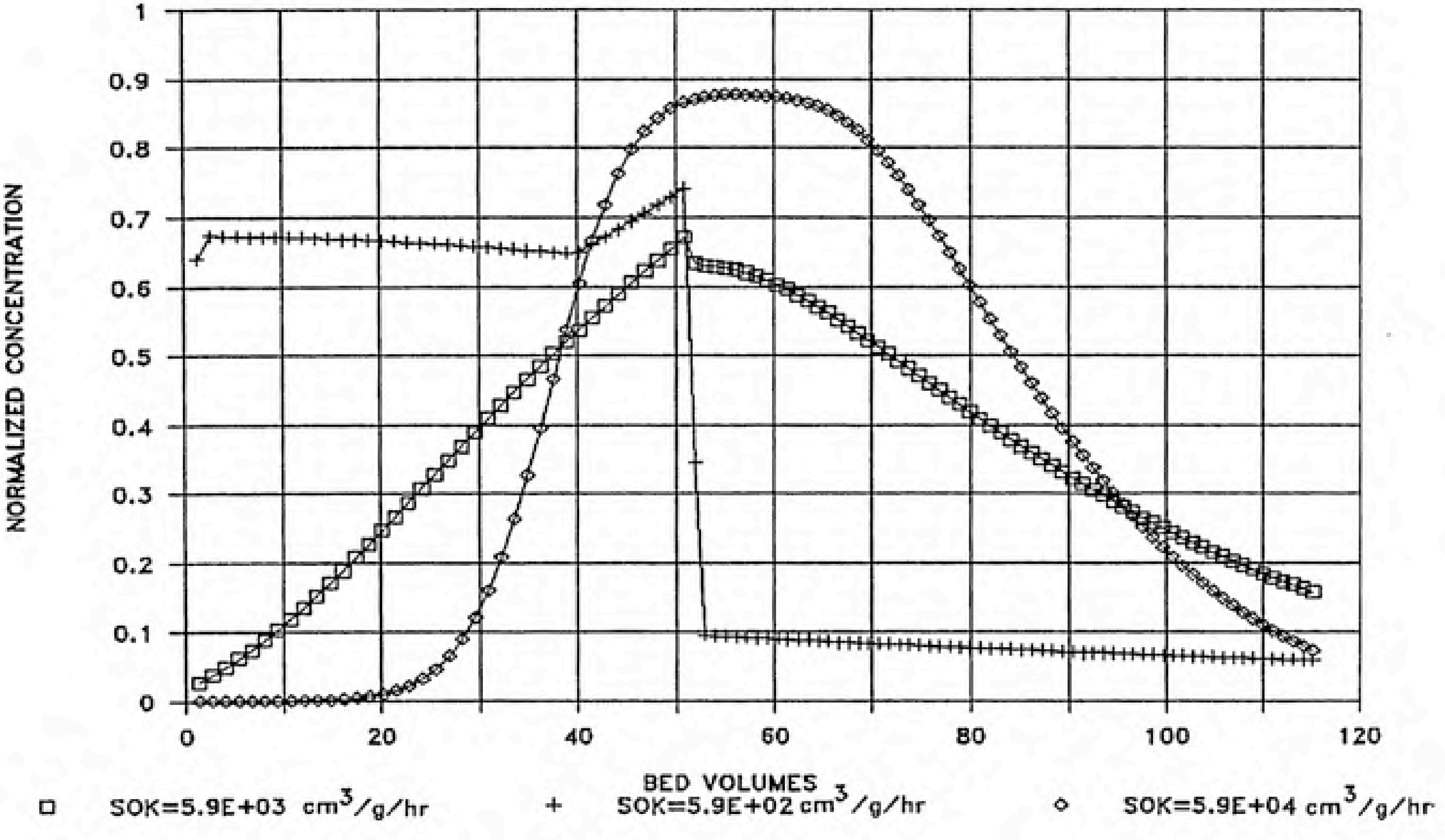


FIGURE IV-20

15-1, FED, FAST/SLOW SENSITIVITY

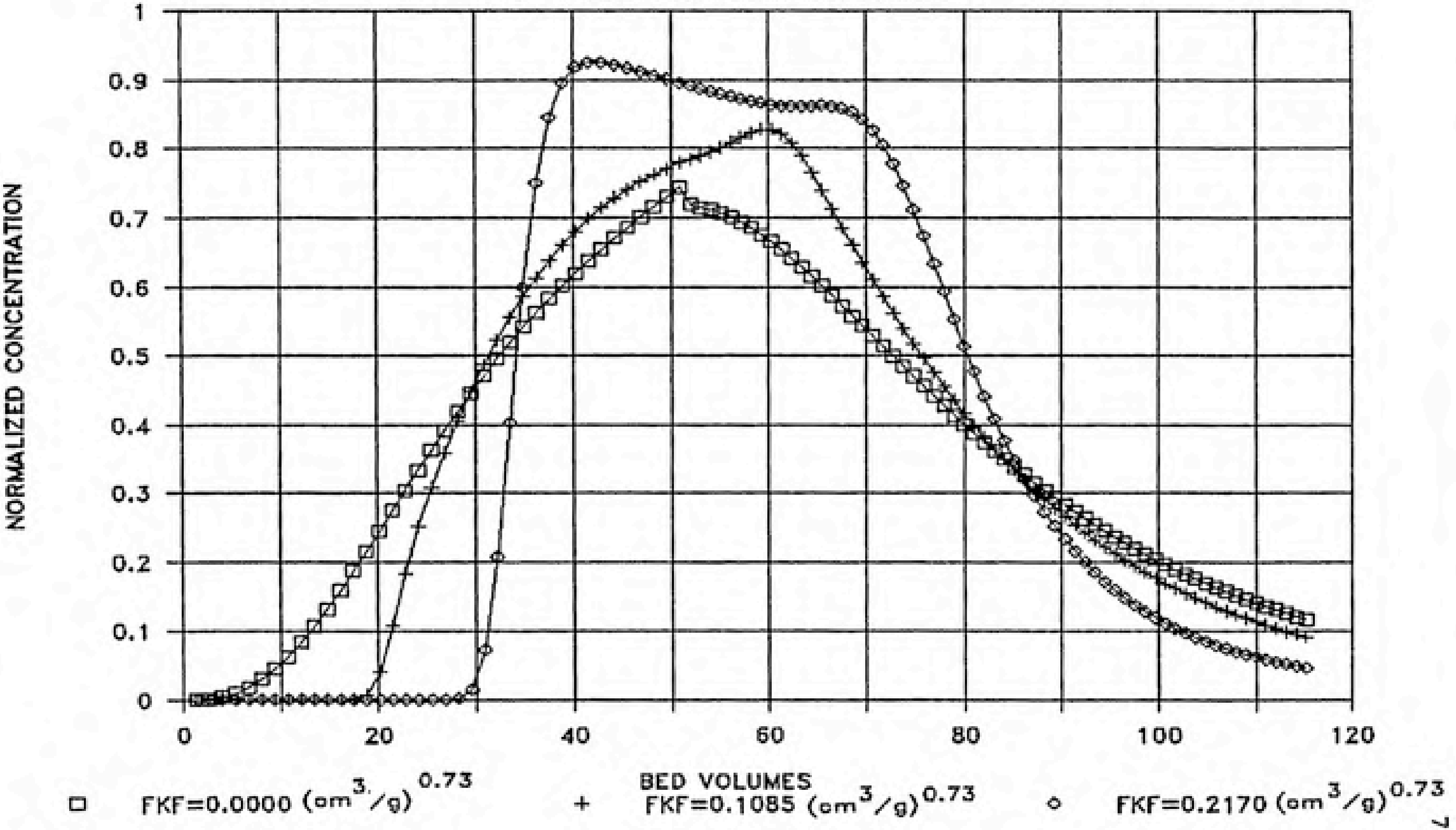
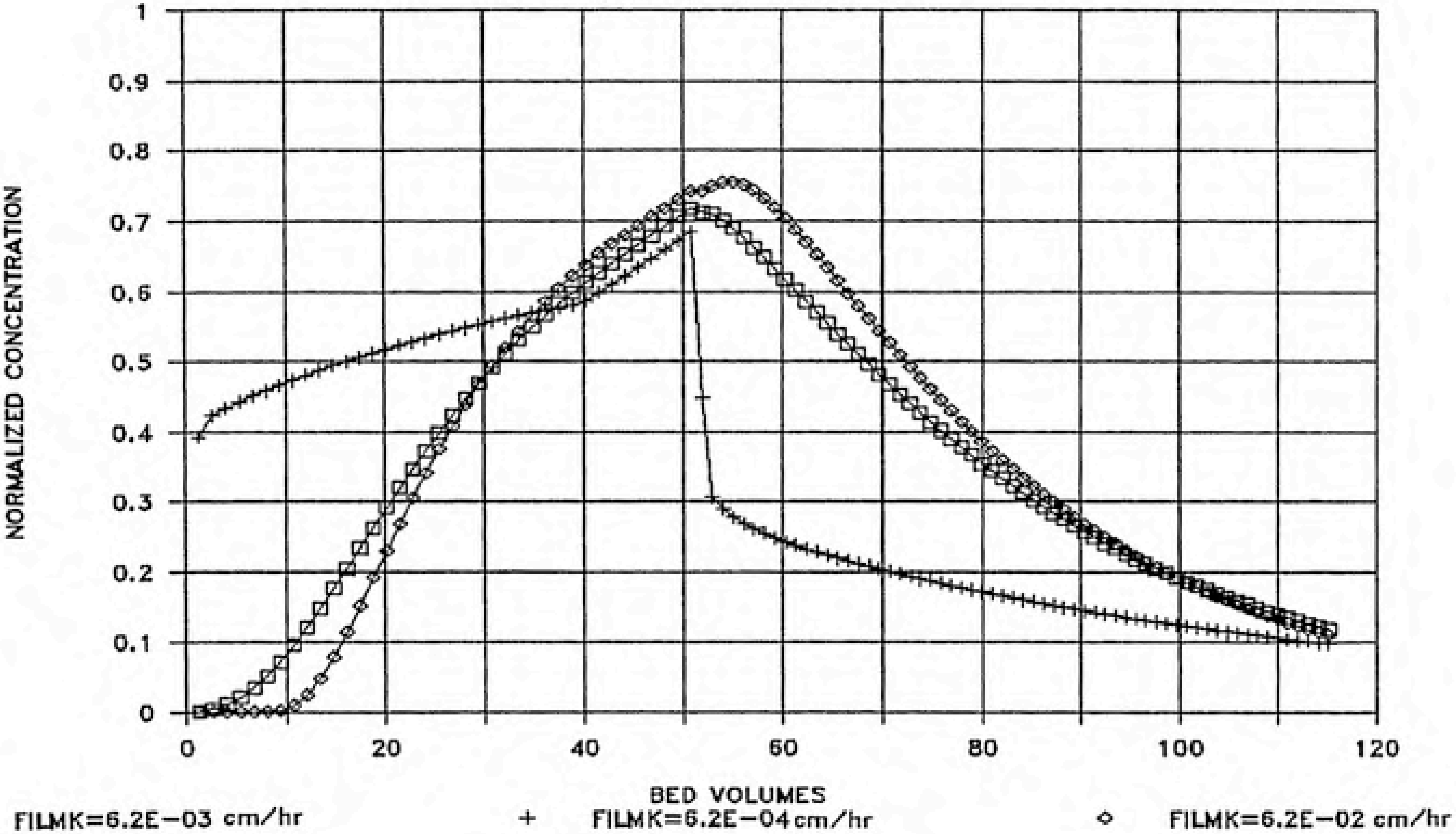


FIGURE IV-21

15-1, DUAL, FILMK SENSITIVITY



On the basis of the preceding analysis, the model DUAL appears calibrated and operationally validated. The situation for FED and LSO is not as clear. Arguably, FED and LSO are also calibrated and operationally validated. By any but the most generous analysis, L-LED and F-LED are not calibrated or operationally validated. In accordance with the model development protocol, the LED models drop from further consideration. The elimination of L-LED is particularly noteworthy because L-LED is the most widely used sorption model in practice.

The next step for the remaining models is to apply the models in predictive mode on new data sets under different conditions to establish dynamic validation. This brings the remaining two data sets involving nitrobenzene into consideration. The first, 16-1, is distinguished by a four-fold increase in influent concentration and a halving of velocity as compared to data set 15-1. The second, 16-2, is quite similar to 16-1 and has only a slightly lower influent concentration than 16-1.

Tables IV-6 and IV-7 present the input parameters for the purely predictive runs. Figures IV-22 through IV-27 display the resulting breakthrough predictions together with the experimental observations. Tables IV-8 and IV-9 list the variance of each of the predictive models for each experiment.

For both soil columns all three remaining models -- LSO, FED, and DUAL -- overpredict the peak concentration. The models do, however, capture the general shape and timing of the breakthrough curve. Variance, the selected quantitative measure of the models' fit, increases in all cases from the calibration run.

Table IV-6

Input Parameters for Models of Column 16-1
A Predictive Run

Variable	Value
D_h during contamination	0.1773 E+01 cm^2/hr
D_h for L-LED during contamination	0.7820 E+01 cm^2/hr
D_h for F-LED during contamination	0.6855 E+01 cm^2/hr
θ	0.3730 E+00*
ρ	0.2670 E+01 g/cm^3
v during contamination	0.3020 E+01 cm/hr
v during elutriation	0.3140 E+01 cm/hr
z	0.2180 E+02 cm
K_p	0.6690 E+01 cm^3/g
K_F	0.2170 E+00 (cm^3/g)$^{0.73}$
n	0.7300 E+00*
$K_{F,f}$	0.0000 E+00 (cm^3/g)$^{0.73}$
n	0.7300 E+00*
$K_{F,s}$	0.2170 E+00 (cm^3/g)$^{0.73}$
n_s	0.7300 E+00*
α	0.3320 E-01 1/hr
Q^o	0.4400 E-04 g/g
b	0.2910 E+06 cm^3/g
k_s	0.5900 E+04 cm^3/g/hr
D_s	0.3440 E-06 cm^2/hr
k_f	0.6200 E-02 cm/hr
R	0.1160 E-01 cm/hr
C_o @ 5=0.00 hr	0.4653 E-05 g/cm^3
C_1 @ t=120 hr	0.4240 E-05 g/cm^3
C_2 @ t=482 hr	0.4140 E-05 g/cm^3
C_3 @ t=482+ hr	0.1000 E-19 g/cm^3
C_4 @ t=1000 hr	0.1000 E-19 g/cm^3

* dimensionless

Table IV-7

Input Parameters for Models of Column 16-2, A Predictive Run

Variable	Value
D_h during contamination	0.1290 E+01 cm^2/hr
D_h for L-LED during contamination	0.8550 E+01 cm^2/hr
D_h for F-LED during contamination	0.7490 E+01 cm^3/hr
θ	0.3730 E+00*
ρ	0.2670 E+01 g/cm^3
v during contamination	0.3300 E+01 cm/hr
v during elutriation	0.3670 E+01 cm/hr
z	0.2180 E+02 cm
K_p	0.6690 E+01 cm^3/g
K_F	0.2170 E+00 $(cm^3/g)^{0.73}$
n	0.7300 E+00*
$K_{F,f}$	0.0000 E+00 $(cm^3/g)^{0.73}$
n_f	0.7300 E+00*
$K_{F,s}$	0.2170 E+00 $(cm^3/g)^{0.73}$
n_s	0.7300 E+00*
α	0.3320 E-01 1/hr
Q^o	0.4400 E-04 g/g
b	0.2910 E+06 cm^3/g
k_s	0.5900 E+04 $cm^3/g/hr$
D_s	0.3440 E-06 cm^2/hr
k_f	0.6200 E-02 cm/hr
R	0.1160 E-01 cm
C_o @ t=0.00	0.3191 E-05 g/cm^3
C_1 @ t=120 hr	0.3130 E-05 g/cm^3
C_2 @ t=482 hr	0.2480 E-05 g/cm^3
C_3 @ t=482+ hr	0.1000 E-19 g/cm^3
C_4 @ t=1000 hr	0.1000 E-19 g/cm^3

FIGURE IV-22

16-1, LSO, PREDICTIVE MODE

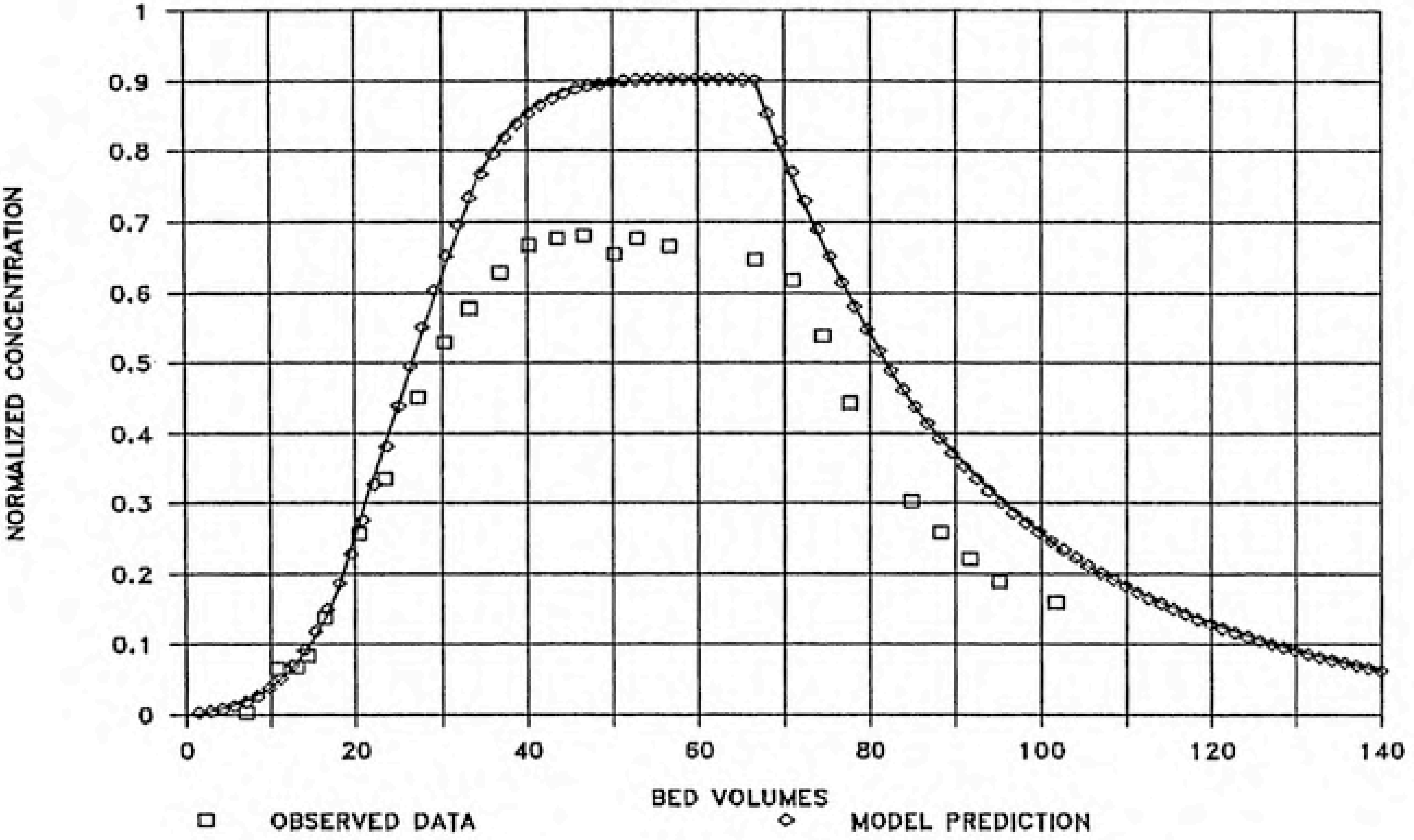


FIGURE IV-23

16-1, FED, PREDICTIVE MODE

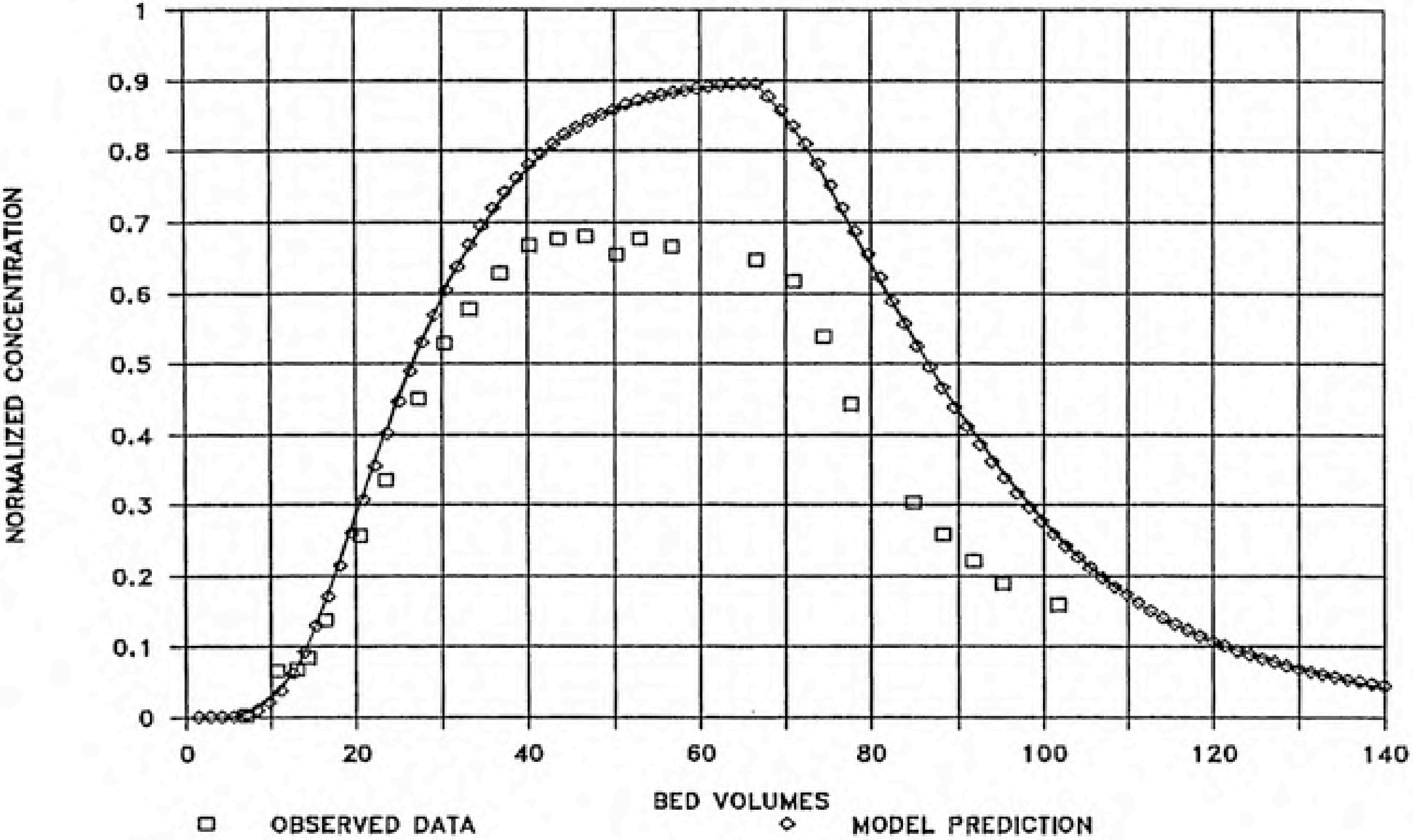


FIGURE IV-24

16-1, DUAL, PREDICTIVE MODE

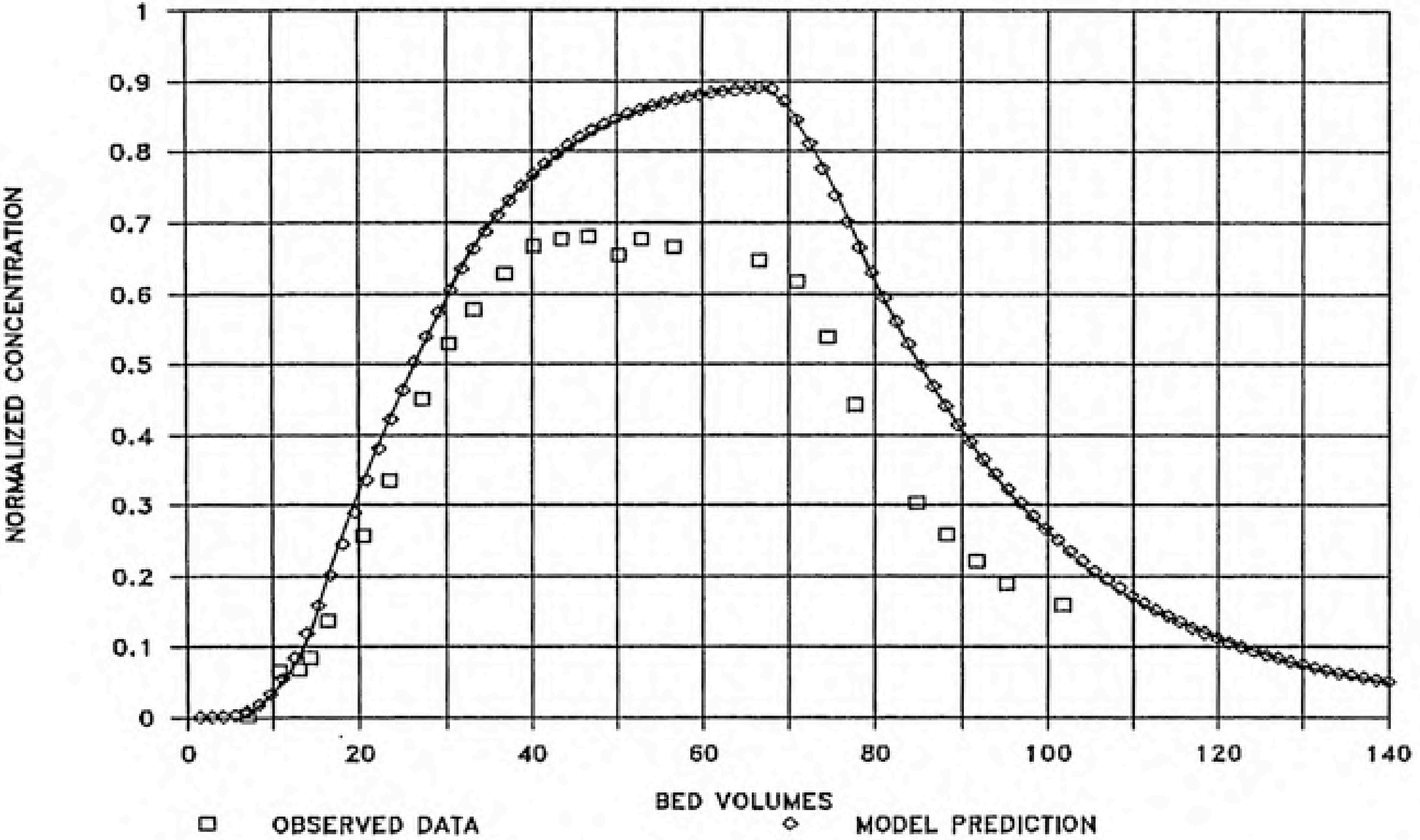


FIGURE IV-25

16-2, LSO, PREDICTIVE MODE

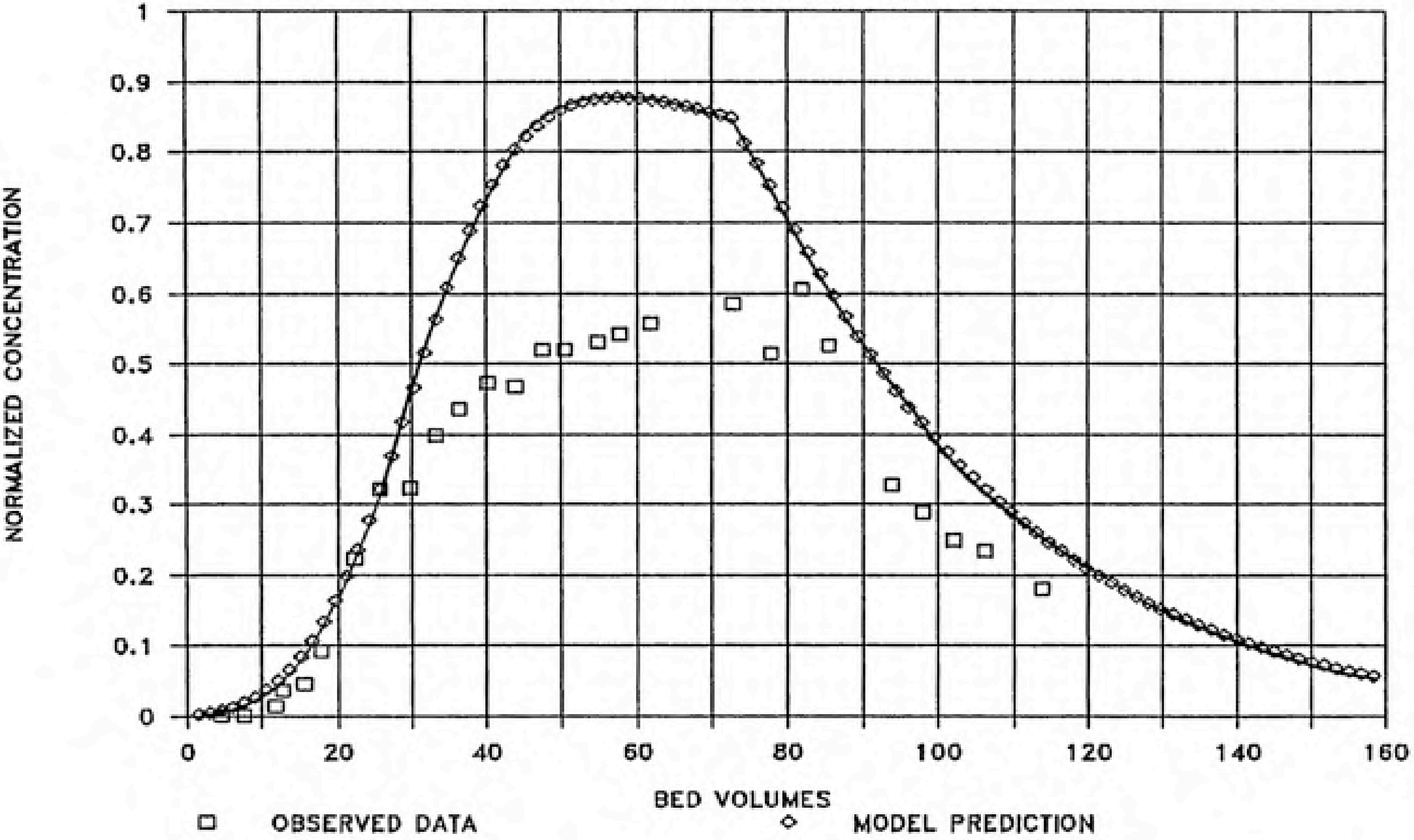


FIGURE IV–26

16–2, FED, PREDICTIVE MODE

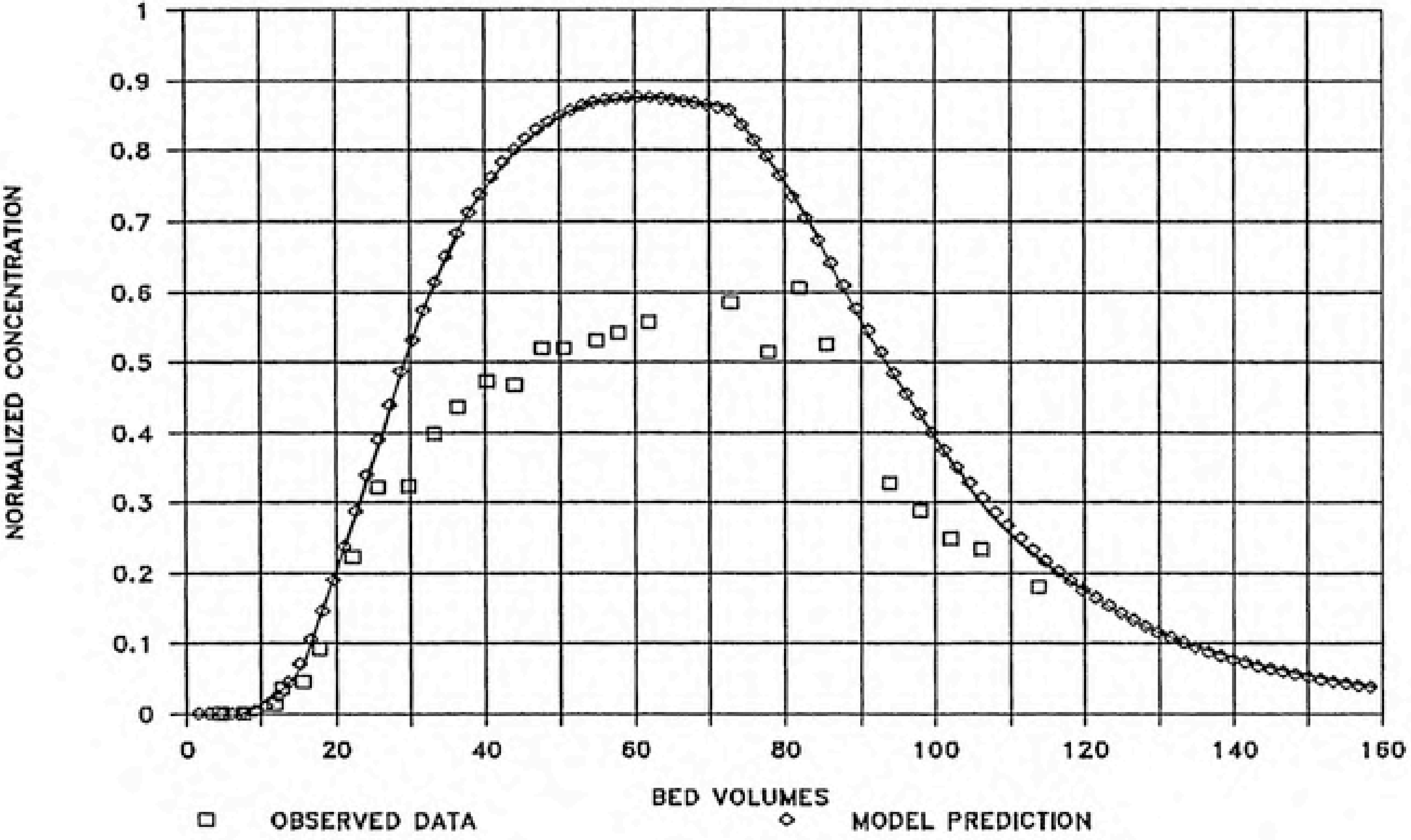


FIGURE IV-27

16-2, DUAL, PREDICTIVE MODE

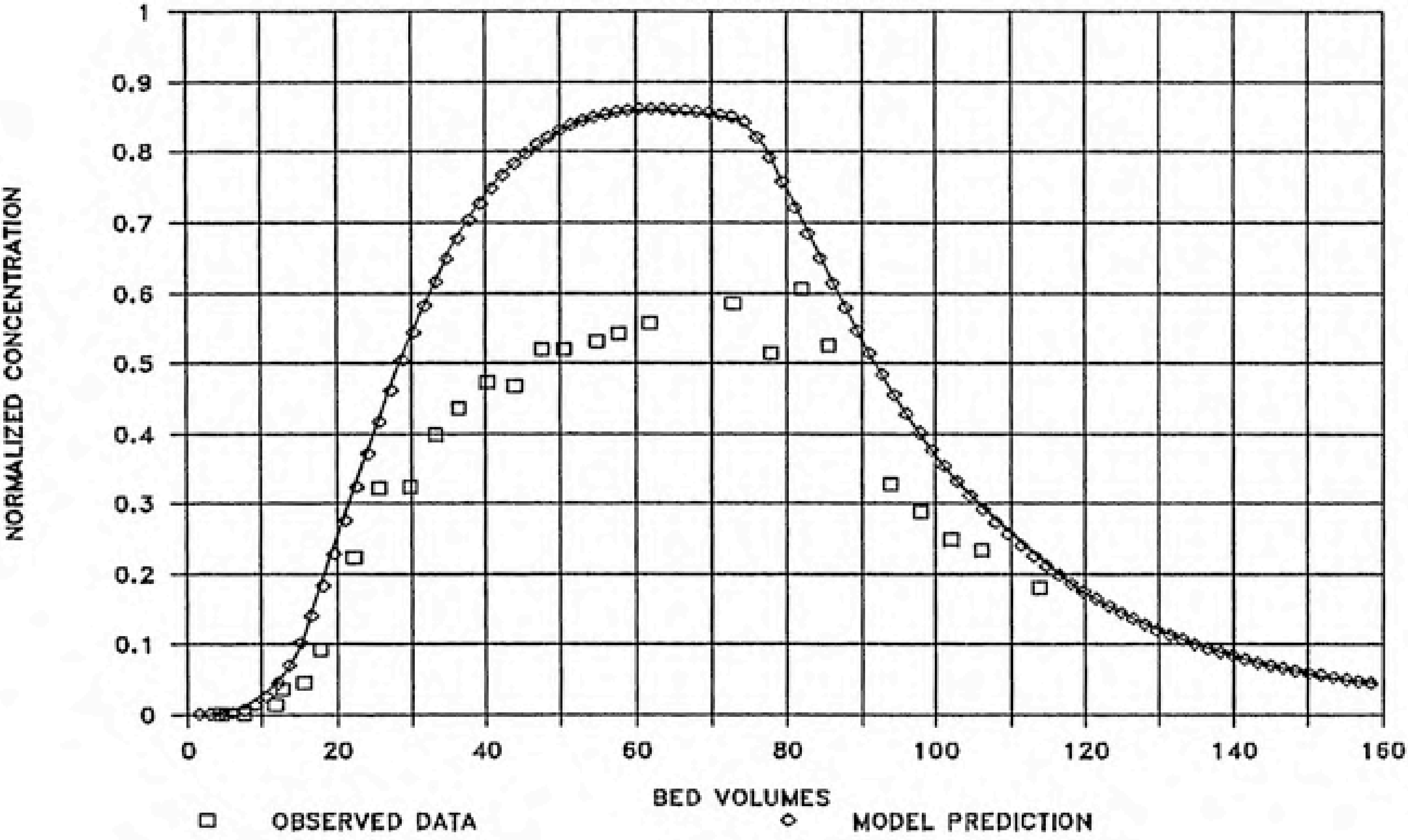


Table IV-8

Variance of Fitted Models of Column 16-1

Model	Model Variance $(g/cm^3)^2$
LSO	0.0203
FED	0.0224
DUAL	0.0197

Table IV-9

Variance of Fitted Models of Column 16-2

Model	Model Variance $(g/cm^3)^2$
LSO	0.0375
FED	0.0414
DUAL	0.0381

Clearly the models appear somewhat lacking in their ability to predict contaminant breakthrough accurately in the soil columns. These predictive runs fail to establish dynamic validation.

A number of possible explanations for this unfortunate situation exist. Although the models themselves may be incorrect, experience of Miller (1984) suggests they should perform better. One area of concern is the original lab work. First, the equilibrium nitrobenzene isotherms studies in the lab never encompass the high concentrations in the soil column experiments 16-1 and 16-2. Since the columns operate in the range of extrapolation beyond the range of earlier experience, the sorption isotherm input parameters may be inadequate. Since the column models do much better in experiment 15-1, a run at much lower concentrations, this hypothesis is consistent with the facts. Second, the lab work may suffer from a mass balance error. Due to the automatic simulation mass balance check, it is clear the simulation mass balances are in order. Unfortunately there is no way to check the mass balance of the actual laboratory experiment since the experiment stops while a considerable concentration of contaminant continues to exit. Third, there are many potential sources of error in the laboratory setup ranging from sorption into the tubing to leaky joints. Fourth, the laboratory study includes relatively few nitrobenzene experiments, so lack of experience may play a role.

The above discussion does not suggest that the laboratory work is substandard. Given the considerable practical difficulties posed by the experiment, the problems the technician appears to have encountered are understandable. This just suggests that the discrepancy between the laboratory observations and the model predictions may not stem from any problems with the models. If the laboratory results are real, possible causes of the apparent mass loss include hysteretic or irreversible sorption, contaminant degradation, or volatile losses.

B. Field-scale analysis

The field-scale problem depicts a highly simplified, homogeneous, one-dimensional situation that resembles column 15-1 in most respects except size. It portrays a column 100 times longer than the laboratory column, a system almost 25 meters long. Hydrodynamic dispersion increases 100 times. The run simulates a constant release of nitrobenzene for 10 thousand hours (417 days), followed by an abrupt contaminant cutoff and 50 thousand hours (6 years) of elutriation. Table IV-10 lists the input parameters for all the models.

Figure IV-28 illustrates the resulting contaminant breakthrough predicted by each model. As expected the conventional linear-local-equilibrium model (L-LED) predicts no tailing, while the other models do predict tailing. It is interesting to note the convergence of three of the models -- F-LED, FED, and DUAL -- to virtually the same prediction. This is not inevitable but instead an interesting consequence of the particular input function and system parameters. Comparison of the field scale predictions with experimental findings is impossible at present because no field-scale data are available, but the tailing predictions agree in a qualitative sense with the findings of other observers.

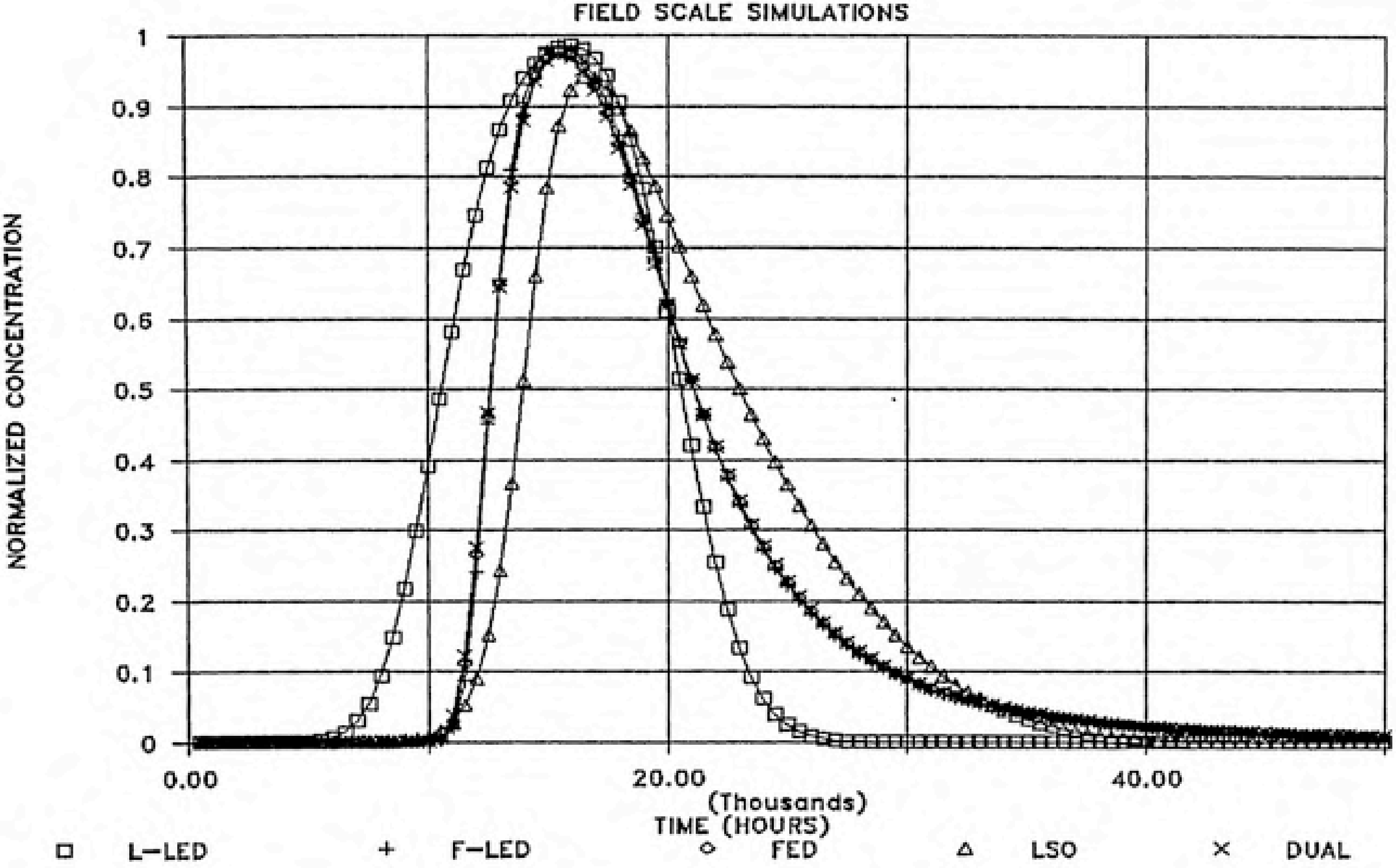
Table IV-10

Input Parameters for Field-Scale Investigations

Variable	Value
D_h during contamination	0.2970 E+03 cm^2/hr
D_h for L-LED during contamination	0.2970 E+03 cm^2/hr
D_h for F-LED during contamination	0.2970 E+03 cm^2/hr
θ	0.3970 E+00*
ρ	0.2670 E+01 g/cm^3
v during contamination	0.6564 E+01 cm/hr
v during elutriation	0.6564 E+01 cm/hr
z	0.2450 E+04 cm
K_p	0.6690 E+01 cm^3/g
K_f	0.2170 E+00 (cm^3/g)$^{0.73}$
n	0.7300 E+00*
$K_{F,f}$	0.2170 E-01 (cm^3/g)$^{0.73}$
n_f	0.7300 E+00*
$K_{F,s}$	0.1953 E+00 (cm^3/g)$^{0.73}$
n_s	0.7300 E+00*
α	0.3320 E-01 1/hr
Q^o	0.4400 E-04 g/g
b	0.2910 E+06 cm^3/g
k_s	0.5900 E+04 cm^3/g/hr
D_s	0.3440 E-06 cm^2/hr
k_f	0.6200 E-02 cm/hr
R	0.1160 E-01 cm
C_o @ t=0.00 hr	0.1564 E-05 g/cm^3
C_1 @ 5=10,000 hr	0.1564 E-05 g/cm^3
C_2 @ t=10,000+ hr	0.1000 E-10 g/cm^3
C_3 @ t=60,000 hr	0.1000 E-10 g/cm^3

* dimensionless

FIGURE IV-28


FIELD SCALE SIMULATIONS
NORMALIZED CONCENTRATION
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.00
20.00
40.00
(Thousands)
TIME (HOURS)
□ L-LED
+ F-LED
◇ FED
△ LSO
× DUAL

V. CONCLUSIONS AND RECOMMENDATIONS

A. Conclusions

The product of this research is a set of four improved ground water contaminant transport simulators. The programs are useful for investigating the effects of nonlinear and rate-controlled sorption in one-dimensional systems. The programs are flexible, mechanically reliable, and well documented.

All three rate-controlled simulators (FED, LSO and DUAL) and particularly the dual-resistance simulator (DUAL) can fit the experimentally determined nitrobenzene breakthrough measurements for the calibration run on Ann Arbor granular aquifer material. None of the rate-controlled models does a good job of predicting the experimentally determined nitrobenzene breakthrough in the predictive runs on Ann Arbor granular aquifer material. The discrepancies between the model predictions and the laboratory measurements may stem from inaccurate and incomplete laboratory work rather than from any shortcoming of the models.

The models can simulate contaminant breakthrough in simplified field-scale scenarios. Only nonlinear or rate-controlled models anticipate the tailing phenomenon. Depending on system characteristics and input functions, several models may converge on a single breakthrough curve prediction.

B. Recommendations

The simulators in their present state are well suited for additional theoretical work. An extension of the chemical nonequilibrium portion of the Valocchi (1985) paper is clearly desirable. Valocchi's work deals only with linear systems, and these numerical models can extend the investigation to nonlinear systems. Since the simulators are already mature, such a study could readily expose the interplay among the model input functions, dimensionless groupings, and column breakthrough curves.

Since previous studies (Miller, 1984) demonstrate that the rate-controlled models accurately predict lindane breakthrough curves in soil column reactors packed with a variety of granular aquifer materials, and since this technical report demonstrates partial success with nitrobenzene despite weak laboratory data, the next logical models to develop are two-or three-dimensional nonlinear, nonequilibrium models expressly designed for field use. The complexity of field-scale dispersion in realistic multidimensional scenarios may require a more sophisticated treatment of dispersion than the current simulators afford.

Such work may someday provide ground water professionals with an improved ability to predict the fate and transport of ground water contaminants.

NOTATION

area	area of elemental volume normal to flow (L^2).
b	Langmuir isotherm sorption-energy constant (L^3M^{-1}).
C	solution-phase solute concentration (ML^{-3}).
C_e	equilibrium solution-phase concentration (ML^{-3}).
C_o	initial solution-phase solute concentration (ML^{-3}).
C_s	fluid-phase, equilibrium-isotherm concentration corresponding to the soil-phase concentration at the particle boundary (ML^{-3}).
D_h	second-rank hydrodynamic dispersion tensor (L^2T^{-1}).
D_h	longitudinal hydrodynamic dispersion coefficient (L^2T^{-1}).
D_s	intraparticle surface-diffusion coefficient for dual-resistance model (L^2T^{-1}).
flux in	mass transfer into the elemental volume ($ML^{-2}T^{-1}$).
flux out	mass transfer out of the elemental volume ($ML^{-2}T^{-1}$).
i	column node index (dimensionless).
k_f	film mass-transfer coefficient for dual-resistance model (LT^{-1})
k_p	linear isotherm sorption partition constant (L^3M^{-1})
k_s	second-order Langmuir model rate constant ($L^3M^{-1}T^{-1}$).
K_F	Freundlich isotherm sorption-capacity constant ($(L^3M^{-1})^n$).

$K_{F,f}$	Freundlich isotherm sorption-capacity constant for the rapid rate component of the equilibrium/first-order rate model $((L^3M^{-1})^{n_s})$.
$K_{F,s}$	Freundlich isotherm sorption-capacity constant for the slow rate component of the equilibrium/first-order rate model $((L^3M^{-1})^{ns})$.
l	radial node index for the dual-resistance model (dimensionless)
n	Freundlich isotherm sorption-energy constant (dimensionless).
n_f	Freundlich isotherm sorption-energy constant for the rapid rate component of the equilibrium/first-order rate model (dimensionless).
n_s	Freundlich isotherm sorption-energy constant for the slow rate component of the equilibrium/first-order rate model (dimensionless).
q	volume-average solid-phase mass normalized by the mass of the solid phase (MM^{-1}).
q_e	equilibrium volume-average soil-phase mass normalized by the solid phase mass (MM^{-1}).
q_f	volume-average soil-phase mass normalized by the mass of the solid phase for the rapid sorption rate component of the equilibrium/first-order model (MM^{-1}).
q_r	solid-phase solute mass normalized by the mass of the solid phase as a function of radial position (MM^{-1}).
q_s	volume-average soil-phase mass normalized by the mass of the solid phase for the slow sorption rate component of the equilibrium/first-order model (MM^{-1}).
Q^o	Langmuir isotherm sorption-capacity constant (MM^{-1}).
r	radial distance variable for dual-resistance model (L).
R	radius of soil particle (L).
reaction	gain or loss of mass due to reaction of any kind (MT^{-1}).

R_f	retardation factor - defined by equation II-9 (dimensionless).
$R_{f,f}$	retardation factor for the rapidly sorbing fraction in the equilibrium/first-order model - equation II-17 (dimensionless).
rxn	subscript denoting a general chemical or mass-transfer reaction (dimensionless).
source or sink	contaminant mass added or removed (MT^{-1}).
srp	subscript denoting sorption reaction (dimensionless).
t	time (T).
$\vec{v}$	pore velocity vector (LT^{-1}).
v_z	pore velocity in the longitudinal direction (LT^{-1}).
z	longitudinal distance variable (L).
Z	length of column (L)

α	equilibrium/first-order model mass-transfer coefficient (T^{-1}).
α_L	longitudinal dispersivity (L).
Δmass	the net rate of change of contaminant mass within the elemental value (MT^{-1}).
$\Gamma(C)$	fluid-phase solution source $(ML^{-3}T^{-1})$.
θ	volume void fraction of the medium (dimensionless).
ρ	density of the soil particle (ML^{-3}).
ρ_b	bulk density of the soil phase (ML^{-3}).

REFERENCES

Bear, J., Hydraulics of Groundwater, McGraw-Hill Book Company, New York, NY, 1979.

Cameron, D.R. and A. Klute, "Convective-dispersive solute transport with a combined equilibrium and kinetic adsorption model," Water Resources Research, vol. 13, no. 1, pp. 183-188, February 1977.

Canter, L.W., and R.C. Knox, Ground Water Pollution Control, Lewis Publishers, Chelsea, MI, 1986.

Chiou, C.T., "Roles of organic matter, minerals and moisture in sorption of nonionic compounds and pesticides by soil," in press.

Chiou, C.T., L.J. Peters, and V.H. Freed, "A physical concept of soil-water equilibria for nonionic organic compounds," Science, vol 206, pp. 831-832, November 16, 1979.

Chiou, C.T., P.E. Porter, and D.W. Schmedding, "Partition equilibria of nonionic organic compounds between soil organic matter and water," Environmental Science and Technology, vol. 17, no. 4, pp. 227-231, 1983.

Coates, J.T. and A.W. Elzerman, "Desorption kinetics for selected PCB cogeners from river sediments," Journal of Contaminant Hydrology, vol. 1, pp. 191-210, 1986.

Crittenden, J.C., J.L. Oravitz, and G. Friedman, "Transport of organic compounds with saturated groundwater flow: model development and parameter sensitivity," Water Resources Research, vol. 22, no. 3, pp. 271-284, March 1986.

Crittenden, J.C. and W.J. Weber, Jr., "Predictive model for design of fixed-bed adsorbers: parameter estimation and model development," Journal of the Environmental Engineering Division, vol. 104, p. 1175, 1978.

Engineering News Record, "Groundwater needs defined," vol. 217, no. 11, p. 16, September 11, 1986a.

Engineering News Record, "Strict rules dampen water cleanup," vol. 217, no. 14, pp. 36-37, October 2, 1986b.

Freeze, R.A. and J.A. Cherry, Groundwater, Prentice-Hall, Englewood Cliffs, NJ, 1979.

Goltz, M.N. and Roberts, P.V. "Interpreting organic solute transport data from a field experiment using physical nonequilibrium models," Journal of Contaminant Hydrology, vol. 1, pp. 77-93, 1986.

Grisham, J.W. (Panel Chairman) et al., "Health aspects of hazardous waste disposal," Environment, vol. 28, no. 3, pp 38-45, April 1986.

Ground Water Monitoring Review, "Priority superfund sites added to EPA list," vol. 5, no. 1, p. 9, Winter 1985.

Hiester, N.K. and T. Vermeulen, "Saturation performance of ion-exchange and adsorption columns," Chemical Engineering Progress, vol. 48, p. 505, 1952.

Journal of the American Water Works Association, "Safeguarding groundwater," vol. 78, no. 9, p.61, September 1986.

Kavanaugh, M.C., "Aquifer remediation case studies," Water World News, pp. 10-12, November/December 1986.

Lapidus,L. and N.R. Amundson, "Mathematics of adsorption in beds," Journal of Physical Chemistry, vol. 56, pp. 984-988, November, 1952.

Miller, C.T. and W.J. Weber, "Sorption of hydrophobic organic pollutants in saturated soil systems," Journal of Contaminant Hydrology, vol. 1, pp. 243-261, 1986.

Miller, C.T. and W.J. Weber, Jr., "Modeling the sorption of hydrophobic compounds by soils - II. Soil column reactor studies," in press.

Miller, C.T., Modeling of Sorption and Desorption Phenomena for Hydrophobic Organic Contaminats in Saturated Soil Environments, Ph.D. Dissertation, University of Michigan, Ann Arbor, MI, 1984.

Mingelgrin, U., and Z. Gerstl, "Reevaluation of partitioning as a mechanism of nonionic chemicals adsorption in soils," Journal of Environmental Quality, vol. 12, pp. 1-11, 1983.

Parker, J.C. and A.J. Valocchi, "Constraints on the validity of equilibrium and first-order kinetic transport models in structured soils," Water Resources Research, vol. 22, no. 3, pp. 399-407, March 1986.

Roberts, P.V., M. Reinhard, A.J. Valocchi, "Movement of organic contaminants in groundwater: implications for water supply," Journal of the American Water Works Association, vol. 74, no. 8, pp. 408-413, August 1982.

Saltzman, S., L. Klinger, and B. Yaron, "Adsorption-desorption of parathion as affected by soil organic matter," Journal of Agricultural Food Chemistry, vol.20, pp. 330-336, 1972.

Science News, "Late-night legislation: New laws include revised Superfund," vol. 130, no. 17, p. 264, October 25, 1986.

Thomann, R.V., "Verification of water quality models," Journal of the Environmental Engineering Division, Proceedings of the American Society of Civil Engineers, vol. 108, no. EE5, pp.923-939, October 1982.

Tschinkel, V.J., "The transition toward long-term management," Environment, vol. 28, no. 3, p. 19, April 1986.

U.S. Water News, "Superfund list lengthens," vol. 3., no. 10, p.1, April 1987.

Valocchi, A.J., "Validity of the local equilibrium assumption for modeling sorbing solute transport through homogenous soils," Water Resources Research, vol. 21, no. 6, pp. 808-820, June 1985.

van Genuchten, M.T., D.H. Tang, and R. Guennelon, "Some exact solutions for solute transport through soils containing large cylindrical macropores," Water Resources Research, vol. 20, no. 3, pp. 335-346, March 1984.

Verschuern, K., Handbook of Environmental Data on Organic Chemicals, Van Nostrand Reinhold Company, New York, NY, 1977.

Weber, W.J., Jr. and C.T. Miller, "Modeling the sorption of hydrophobic compounds by soils - I. Rates and equilibria," in press.

APPENDIX

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C                          SORPTION MODEL :                             C
C                  LOCAL EQUILIBRIUM WITH DISPERSION                    C
C                       MAINFRAME  VERSION                              C
C                                                                       C
C     THIS PROGRAM USES THE DGEAR METHOD TO SOLVE THE ADVECTIVE         C
C     DISPERSIVE EQUATION FOR THE CASE OF LOCAL-EQUILIBRIUM SORPTION.   C
C     THE PROGRAM IS BASED ON THE ASSUMPTIONS THAT THE SOLID AND FLUID  C
C     PHASES ATTAIN INSTANTANEOUS EQUILIBRIUM AND THAT THE EQUILIBRIUM  C
C     RELATION IS THE SINGLE COMPONENT FREUNDLICH ISOTHERM. THE EQUATIONS C
C     ARE SPATIALLY APPROXIMATED USING THE FINITE DIFFERENCE METHOD     C
C     RESULTING IN A SET OF DC/DT EQUATIONS. THESE EQUATIONS            C
C     ARE THEN SOLVED SIMULTANEOUSLY USING DGEAR---AN IMSL SUBROUTINE.  C
C                                                                       C
C     IN ITS PRESENT FORM THE PROGRAM IS LIMITED TO 100 NODES.          C
C                                                                       C
C                                                                       C
C     VARIABLE MAP:                                                     C
C     ------------                                                      C
C                                                                       C
C     VARIABLE NAME      PRECISION               DESCRIPTION            C
C     -------------      ---------    --------------------------------- C
C     I                  INTEGER*4    TIME STEP INDEX                   C
C     IDUMMY             INTEGER*4    DUMMY WORKING ARRAY FOR DGEAR     C
C     IER                INTEGER*4    ERROR CODE RETURN INDEX FOR DGEAR C
C     INDEX              INTEGER*4    CALL INDICATOR INDEX FOR GEAR     C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM          C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO           C
C     IWK                INTEGER*4    WORKING ARRAY FOR DGEAR           C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION    C
C     NDOF               INTEGER*4    NUMBER OF DEGREES OF FREEDOM      C
C     NLC                INTEGER*4    NUMBER OF LOWER CODIAGONALS-DGEAR C
C     NOUT               INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT C
C     NUC                INTEGER*4    NUMBER OF UPPER CODIAGONALS-DGEAR C
C     NSTEPS             INTEGER*4    NUMBER OF EQUAL BED VOLUME STEPS  C
C                                     TO BE CALCULATED                  C
C                                                                       C
C                                                                       C
C     C                  REAL*8       VECTOR OF KNOWN INFLUENT          C
C                                     CONCENTRATIONS (SEE VECTOR T)     C
C     CO                 REAL*8       INITIAL CONCENTRATION             C
C     CNORM              REAL*8       NORMALIZED EXIT CONCENTRATION     C
C     CPRIME             REAL*8       EXTERNAL NAME OF DC/DT SUBROUTINE C
C     CTEMP              REAL*8       EXIT CONCENTRATION                C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA           C
C                                     COLUMN 1 CONTAINS REPORT TIMES    C
C                                     COLUMN 2 GIVES EXPERIMENTAL       C
C                                     CONCENTRATION                     C
C                                     COLUMN 3 GIVES INFLUENT           C
C                                     CONCENTRATION                     C
C                                     COLUMN 4 GIVES DCO/DT             C
C     DCODT              REAL*8       CHANGE IN INFLUENT CONCENTRATION  C
C                                     WITH RESPECT TO TIME              C
C     DDVDT              REAL*8       DERIVATIVE OF DEPENDENT VARIABLES C
C     DGN                REAL*8       DIAGONAL VARIABLE GROUPING TERMS  C
C     DGNLOW             REAL*8       LOWER DIAGONAL VARAIBLE GROUPINGS C
C     DGNUP              REAL*8       UPPER DIAGONAL VARIABLE GROUPINGS C
C     DH                 REAL*8       HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DH1                REAL*8       SORPTION PHASE HYDRODYNAMIC       C
C                                     DISPERSION COEFFICIENT            C
```

```
C     DV                REAL*8       DEPENDENT VARIABLE VECTOR          C
C     EPSLON            REAL*8       ALLOWABLE ERROR FOR DGEAR          C
C     FCNJ              REAL*8       EXTERNAL NAME OF JACOBIAN ROUTINE  C
C     FK                REAL*8       FREUNDLICH ISOTHERM COEFFICIENT    C
C     FKAPPA            REAL*8       FLUID-PHASE LOSS RATE              C
C     FN                REAL*8       FREUNDLICH ISOTHERM EXPONENT       C
C     H                 REAL*8       STEP SIZE FOR DGEAR                C
C     OUTCON            REAL*8       VECTOR OF EXIT CONCENTRATIONS      C
C     OUTPUT            REAL*4       A MATRIX OF SIMULATION STATUS      C
C                                    COL 1 CONTAINS REPORT TIME         C
C                                    COL 2 CONTAINS BED VOLUMES         C
C                                    COL 3 CONTAINS THROUGHPUT          C
C                                    COL 4 GIVES INFLUENT CONCENTRATION C
C                                    COL 5 GIVES DCO/DT                 C
C     RDSQSM            REAL*8       SUM OF RESIDUALS SQUARED           C
C     RFGRP             REAL*8       RETARDATION FACTOR GROUPING        C
C     RHO               REAL*8       SOLID PHASE PARTICLE DENSITY       C
C     RSDSUM            REAL*8       SUM OF RESIDUALS                   C
C     SDUMMY            REAL*8       DUMMY ROUTINE FOR DGEAR            C
C     SKAPPA            REAL*8       SOLID-PHASE LOSS RATE              C
C     SPLGRP            REAL*8       SOLID PHASE LOSS GROUP             C
C     T                 REAL*8       VECTOR OF ELAPSED TIME             C
C                                    CORRESPONDING TO CONCENTRATIONS    C
C                                    IN THE VECTOR C                    C
C     THETA             REAL*8       POROSITY                           C
C     THESIM            REAL*8       NUMBER OF BED VOLUMES TO SIMULATE  C
C     TIME              REAL*8       TIME OF OUTPUT SIMULATION          C
C     TO                REAL*8       INITIAL TIME FOR DGEAR             C
C     VEL               REAL*8       PORE VELOCITY                      C
C     VEL1              REAL*8       SORPTION PORE VELOCITY             C
C     VEL2              REAL*8       DESORPTION PORE VELOCITY           C
C     WK                REAL*8       WORKING ARRAY NEEDED BY DGEAR      C
C     ZCOL              REAL*8       LENGTH OF COLUMN (Z DIRECTION)     C
C                                                                       C
C     TITLE             CHARACTER*72 TITLE OF SIMULATION RUN            C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      PROGRAM LED
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 SDUMMY, DATA, OUTPUT
      CHARACTER*72 TITLE
C
C
      DIMENSION F(100), X(1)
      DIMENSION IWK (101)
      DIMENSION DATA (100,4), DV (101), DDVDT (101), OUTPUT (250,5),
     1          WK (1919),C (100), T (100), OUTCON (250)
      COMMON /DBAND/ NLC, NUC
      COMMON /GEAR/ DUMMY (48), SDUMMY (4), IDUMMY (38)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, RFGRP, SPLGRP
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1              EPSLON, CO, FK, FN, FKAPPA, SKAPPA, C, T,
     2              ICON, NCOL, NOUT, NSTEPS
      COMMON /WORDS/ TITLE
      COMMON /FLUID/ DCODT
      COMMON /UNITS/ IOIN, IOOUT
      EXTERNAL CPRIME, FCNJ
      IOIN = 1
      IOOUT = 3
      CALL READ
      VEL1 = VEL
```

```
      CALL ECHO
      CALL ERRSET (208, 256, -1, 1)
C
C.....COMPUTE CONCENTRATIONS AT DATA POINTS IF DATA ENTERED
C
      IF (NOUT .GT. 0) THEN
         TO = 0.D00
         CALL INITL (DDVDT, DV, OUTPUT, TO)
         CALL GRPFRM
         CALL INFO1 (DATA, NOUT)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
         NDOF = NCOL
         H = 1.D-08
         INDEX = 1
         DV (1) = CO
         NLC = 2
         NUC = 2
C
C.....TIME LOOP FOR SIMULTATION AND OUTPUT AT EXPERIMENTAL POINTS
C
         DO 100 I = 1,NOUT
            IF (I .GT. 1) THEN
               IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
     1             (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
                  H = 1.D-08
                  INDEX = 1
                  DCODT = 0.000
                  DV (1) = DATA (I, 3)
                  WRITE (IODUT, 1300)
                  IF (DV (1) .LT. 1.0E-10) THEN
                     VEL = VEL2
                     DH = DH1 * VEL2 / VEL1
                     CALL GRPFRM
                     WRITE (IODUT, 1400)
                  ENDIF
               ELSE
                  DCODT = DATA (I, 4)
               ENDIF
            ELSE
               DCODT = DATA (I, 4)
            ENDIF
            TIME = DATA (I,1)
            CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
     1                  EPSLON, 2, -2, INDEX, IWK, WK, IER)
            CALL REPORT (I, DV, RSDSUM, RDSQSM, RSD)
            OUTCON (I) = DV (NDOF)
  100    F(I) = RSD
C
C.....RESTORE ORIGINAL PARAMETERS
C
         DH = DH1
         VEL = VEL1
C
C.....CHECK THE MASS BALANCE
C
         CALL MASS1 (OUTCON, DV, CO, ZCOL, THETA, FK, FN, VEL, VEL2,
     1               RHO, DATA, NOUT, NCOL)
C
      ENDIF
C
C.....IF REGULAR TIME INTERVAL SIMULATION OUTPUT IS REQUESTED, COMPUTE
C
      IF (NSTEPS .GT. 0) THEN
```

```
         CALL INITL (DDVDT, DV, OUTPUT, TO)
         CALL GRPFRM
         CALL INFO2 (OUTPUT, NSTEPS)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
         NDOF = NCOL
         H = 1.D-08
         INDEX = 1
         DV (1) = CO
         NLC = 2
         NUC = 2
C
C.....TIME LOOP FOR SIMULTATION AND OUTPUT OF COMPUTED POINTS
C
         WRITE (IOOUT, 1000)
         WRITE (IOOUT, 1100)
         DO 110 I = 1, NSTEPS
            IF (I .GT. 1) THEN
               IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     1             (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
                   H = 1.D-08
                   INDEX = 1
                   DCODT = 0.000
                   DV (1) = OUTPUT (I, 4)
                   WRITE (IOOUT, 1300)
                   IF (DV (1) .LT. 1.0E-10) THEN
                      VEL = VEL2
                      DH = DH1 * VEL2 / VEL1
                      CALL GRPFRM
                      WRITE (IOOUT, 1400)
                   ENDIF
               ELSE
                   DCODT = OUTPUT (I, 5)
               ENDIF
            ELSE
               DCODT = OUTPUT (I, 5)
            ENDIF
            TIME = OUTPUT (I,1)
            CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
     1                  EPSLON, 2, -2, INDEX, IWK, WK, IER)
            CTEMP = DV (NDOF)
            CNORM = CTEMP / CO
            OUTCON (I) = DV (NDOF)
  110    WRITE (IOOUT,1200) (OUTPUT (I,J), J = 1, 3), CTEMP, CNORM
C
C.....RESTORE ORIGINAL PARAMETERS
C
         DH = DH1
         VEL = VEL1
C
C.....CHECK THE MASS BALANCE
C
         CALL MASS2 (OUTCON, DV, CO, ZCOL, THETA, FK, FN, VEL, VEL2,
     1            RHO, OUTPUT, NSTEPS, NCOL)
C
      ENDIF
 1000 FORMAT ('1', 42X, 'SIMULATION OUTPUT FOR PLOTTING')
 1100 FORMAT ('0---------------------------------------------------',
     1        '---------------------------------------------------',
     2        '-----------------------------------')
 1200 FORMAT (5E12.5)
 1300 FORMAT (' ', 20X, '***** '. 'ABRUPT INFLUENT CONCENTRATION ',
     1        'CHANGE. ', '***** ', 'DGEAR WAS RESET. ', '*****')
 1400 FORMAT (' ', 23X, '***** ', 'DESORPTION DETECTED. ',
```

```
     *         ***** ', 'HYDRODYNAMICS WERE RESET. ', '*****')
      STOP
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE READS THE BASIC INPUT REQUIRED FOR THE LOCAL     C
C     EQUILIBRIUM SORPTION VERSION OF THE ADVECTIVE-DISPERSIVE EQUATION I C
C     FROM UNIT NUMBER IOIN.                                           C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME      PRECISION               DESCRIPTION           C
C     -------------      ---------    ---------------------------------  C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM         C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO          C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION   C
C     NOUT               INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT C
C     NSTEPS             INTEGER*4    NUMBER OF EQUAL BED VOLUME STEPS  C
C                                     TO BE CALCULATED                 C
C                                                                      C
C                                                                      C
C     C                  REAL*8       VECTOR OF KNOWN INFLUENT         C
C                                     CONCENTRATIONS (SEE VECTOR T)    C
C     CO                 REAL*8       INITIAL CONCENTRATION            C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA          C
C                                     COLUMN 1 CONTAINS REPORT TIMES   C
C                                     COLUMN 2 GIVES EXPERIMENTAL      C
C                                     CONCENTRATION                    C
C                                     COLUMN 3 GIVES INFLUENT          C
C                                     CONCENTRATION                    C
C                                     COLUMN 4 GIVES DCO/DT            C
C     DH                 REAL*8       HYDRODYNAMIC DISPERSION COEFFICIENT C
C     EPSLON             REAL*8       ALLOWABLE ERROR FOR DGEAR        C
C     FK                 REAL*8       FREUNDLICH ISOTHERM COEFFICIENT  C
C     FKAPPA             REAL*8       FLUID-PHASE LOSS RATE            C
C     FN                 REAL*8       FREUNDLICH ISOTHERM EXPONENT     C
C     RHO                REAL*8       SOLID PHASE PARTICLE DENSITY     C
C     SKAPPA             REAL*8       SOLID-PHASE LOSS RATE            C
C     T                  REAL*8       VECTOR OF ELAPSED TIME           C
C                                     CORRESPONDING TO CONCENTRATIONS  C
C                                     IN THE VECTOR C                  C
C     THETA              REAL*8       POROSITY                         C
C     THESIM             REAL*8       NUMBER OF BED VOLUMES TO SIMULATE C
C     VEL                REAL*8       PORE VELOCITY                    C
C     VEL2               REAL*8       DESORPTION PORE VELOCITY         C
C     ZCOL               REAL*8       LENGTH OF COLUMN (Z DIRECTION)   C
C                                                                      C
C                                                                      C
C     TITLE              CHARACTER*72 TITLE OF SIMULATION              C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE READ
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      CHARACTER*72 TITLE
      DIMENSION DATA (100,4), C (100), T (100)
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, FKAPPA, SKAPPA, C, T.
```

```
     2              ICON, NCOL, NOUT, NSTEPS
      COMMON /WORDS/ TITLE
      COMMON /UNITS/ IOIN, IOOUT
      EXTERNAL CPRIME, FCNJ
      READ (IOIN, 1000) TITLE
      READ (IOIN, 1100) NCOL, NOUT, NSTEPS, THESIM, ICON
      IF (ABS (NOUT) .GT. 0 ) THEN
         NTEMP = ABS (NOUT)
         READ (IOIN, 1200) (DATA (I,1), I = 1, NTEMP)
         READ (IOIN, 1200) (DATA (I,2), I = 1, NTEMP)
      ENDIF
      READ (IOIN, 1200) VEL, DH, RHO, THETA, ZCOL, EPSLON
      READ (IOIN, 1200) CO, FK, FN, FKAPPA, SKAPPA,  VEL2
      READ (IOIN, 1200) (C (I), I = 1, ICON)
      READ (IOIN, 1200) (T (I), I = 1, ICON)
 1000 FORMAT (A72)
 1100 FORMAT (3I5, E12.5, I5)
 1200 FORMAT ((6E12.5))
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C     THIS SUBROUTINE ECHOS THE INPUT TO DEVICE IOOUT.                  C
C                                                                       C
C     VARIABLE MAP:                                                     C
C     ------------                                                      C
C                                                                       C
C     VARIABLE NAME     PRECISION                DESCRIPTION            C
C     -------------     ---------   ---------------------------------   C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM            C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO             C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION      C
C                                                                       C
C                                                                       C
C     C                 REAL*8      VECTOR OF KNOWN INFLUENT            C
C                                   CONCENTRATIONS (SEE VECTOR T)       C
C     CO                REAL*8      INITIAL CONCENTRATION               C
C     DH                REAL*8      HYDRODYNAMIC DISPERSION COEFFICIENT C
C     EPSLON            REAL*8      ALLOWABLE ERROR FOR DGEAR           C
C     FK                REAL*8      FREUNDLICH ISOTHERM COEFFICIENT     C
C     FKAPPA            REAL*8      FLUID-PHASE LOSS RATE               C
C     FN                REAL*8      FREUNDLICH ISOTHERM EXPONENT        C
C     PE                REAL*8      SYSTEM PECLET NUMBER                C
C     RHO               REAL*8      SOLID PHASE PARTICLE DENSITY        C
C     SKAPPA            REAL*8      SOLID-PHASE LOSS RATE               C
C     T                 REAL*8      VECTOR OF ELAPSED TIME              C
C                                   CORRESPONDING TO CONCENTRATIONS     C
C                                   IN THE VECTOR C                     C
C     THETA             REAL*8      POROSITY                            C
C     VEL               REAL*8      PORE VELOCITY                       C
C     VEL2              REAL*8      DESORPTION PORE VELOCITY            C
C     ZCOL              REAL*8      LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                       C
C                                                                       C
C     TITLE             CHARACTER*72 TITLE OF SIMULATION                C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE ECHO
      IMPLICIT REAL*8 (A - H,O - Z)
      REAL*4 DATA
```

```
     CHARACTER*72 TITLE
     DIMENSION  DATA (100,4), C (100), T (100)
     COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
    1               EPSLON, CO, FK, FN, FKAPPA, SKAPPA, C, T,
    2               ICON, NCOL, NOUT, NSTEPS
     COMMON /WORDS/ TITLE
     COMMON /UNITS/ IOIN, IOOUT
     PE = VEL * ZCOL / DH
     WRITE (IOOUT, 1000)
     WRITE (IOOUT, 1100)
     WRITE (IOOUT, 1200) TITLE
     WRITE (IOOUT, 1400)
     WRITE (IOOUT, 1300)
     WRITE (IOOUT, 1400)
     WRITE (IOOUT, 1500) CO
     WRITE (IOOUT, 1600) DH
     WRITE (IOOUT, 1700) NCOL
     WRITE (IOOUT, 1800) RHO
     WRITE (IOOUT, 1900) VEL
     WRITE (IOOUT, 1950) VEL2
     WRITE (IOOUT, 2000) THETA
     WRITE (IOOUT, 2100) ZCOL
     WRITE (IOOUT, 2200) PE
     WRITE (IOOUT, 2300) FK
     WRITE (IOOUT, 2400) FN
     WRITE (IOOUT, 2500) FKAPPA
     WRITE (IOOUT, 2600) SKAPPA
     WRITE (IOOUT, 2700) ICON
     WRITE (IOOUT, 2800)
     WRITE (IOOUT, 2900) ( C (I), I = 1, ICON )
     WRITE (IOOUT, 3000)
     WRITE (IOOUT, 2900) ( T (I), I = 1, ICON )
1000 FORMAT ('1', 50X, 'PLUG FLOW DISPERSION REACTOR')
1100 FORMAT (' ', 47X, 'LOCAL EQUILIBRIUM KINETIC SIMULATION')
1200 FORMAT (' ', 20X, A72)
1300 FORMAT (' ', 15X, 'VARIABLE', 16X, 'DESCRIPTION', 30X, '   VALUE',
    1        16X, 'UNITS   ', 16X)
1400 FORMAT ('O---------------------------------------------------',
    1        '----------------------------------------------------',
    2        '---------------------------------')
1500 FORMAT ('O', 15X, 'CO    ', 18X, 'INITIAL CONCENTRATION    ', 14X,
    1        E10.4, 16X, 'G/CM**3')
1600 FORMAT (' ', 15X, 'D     ', 18X, 'HYDRODYNAMIC DISPERSION  ', 14X,
    1        E10.4, 16X, 'CM**2/HOUR')
1700 FORMAT (' ', 15X, 'NDZ   ', 18X, 'NUMBER OF COLUMN STEPS   ', 16X,
    1        I5)
1800 FORMAT (' ', 15X, 'RHO   ', 18X, 'SOLID PHASE DENSITY      ', 14X,
    1        E10.4, 16X, 'GR/CM**3')
1900 FORMAT (' ', 15X, 'VEL   ', 18X, 'PORE VELOCITY            ', 14X,
    1        E10.4, 16X, 'CM/HOUR')
1950 FORMAT (' ', 15X, 'VEL   ', 18X, 'DESORPTION VELOCITY      ', 14X,
    1        E10.4, 16X, 'CM/HOUR')
2000 FORMAT (' ', 15X, 'VOID  ', 18X, 'VOID VOLUME FRACTION     ', 14X,
    1        E10.4)
2100 FORMAT (' ', 15X, 'ZL    ', 18X, 'LENGTH OF COLUMN         ', 14X,
    1        E10.4, 16X, 'CM')
2200 FORMAT (' ', 15X, 'PE    ', 18X, 'PECLET NUMBER            ', 14X,
    1        E10.4)
2300 FORMAT (' ', 15X, 'KFRS  ', 18X, 'FREUNDLICH COEFFICIENT   ',14X,
    1        E10.4, 16X, '((CM**3/GR)**NFR)')
2400 FORMAT (' ', 15X, 'NS    ', 18X, 'FREUNDLICH EXPONENT      ', 14X,
    1        E10.4)
2500 FORMAT (' ', 15X, 'DIA   ', 18X, 'FLUID PHASE LOSS RATE    ', 14X,
    1        E10.4, 16X, '1/HR')
2600 FORMAT (' ', 15X, 'NDR   ', 18X, 'SOLID PHASE LOSS RATE    ', 14X,
    1        E10.4, 16X, '1/HR')
```

```
 2700 FORMAT (' ', 15X, 'ICON  ', 18X, 'NUMBER OF INFLUENTS      ', 14X,
     1        I5 )
 2800 FORMAT (' ', 39X, 'INFLUENT CONCENTRATIONS' )
 2900 FORMAT (' ', 14X, 10E10.3)
 3000 FORMAT (' ', 39X, 'INFLUENT TIMES')
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE DATA MATRIX TO        C
C     UNIT IOOUT.                                                      C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION                DESCRIPTION           C
C     -------------     ---------   ---------------------------------- C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     NCOUNT            INTEGER*4   LOOP COUNTER                       C
C     NOUT              INTEGER*4   NUMBER OF TEMPORAL STEPS TO OUTPUT C
C                                                                      C
C     DATA              REAL*4      EXPERIMENTAL INPUT DATA            C
C                                   COLUMN 1 CONTAINS REPORT TIMES     C
C                                   COLUMN 2 GIVES EXPERIMENTAL        C
C                                   CONCENTRATION                      C
C                                   COLUMN 3 GIVES INFLUENT            C
C                                   CONCENTRATION                      C
C                                   COLUMN 4 GIVES DCO/DT              C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INFO1 (DATA, NOUT)
      REAL*4 DATA
      DIMENSION DATA (100,4)
      COMMON /UNITS/ IOIN, IOOUT
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200)
      WRITE (IOOUT, 1100)
      DO 100 NCOUNT = 1, NOUT
        WRITE (IOOUT, 1300) ( DATA (NCOUNT, I), I = 1, 4)
 100  CONTINUE
 1000 FORMAT ('1', 42X, 'EXPERIMENTAL COMPARISON MODE INFORMATION')
 1100 FORMAT ('0----------------------------------------------',
     1        '-----------------------------------------------',
     2        '-------------------------------')
 1200 FORMAT (' ', 28X, 'TIME',5X,'CONCENTRATION OUT', 6X,
     1        'CONCENTRATION IN', 16X, 'DCO/DT')
 1300 FORMAT (' ', 20X, E12.5, 10X, E12.5, 10X, E12.5, 10X, E12.5)
      RETURN
      END

C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE OUTPUT MATRIX TO      C
C     UNIT IOOUT.                                                      C
```

```
-                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                 DESCRIPTION            C
C     -------------      ---------    ----------------------------------  C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM            C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO             C
C     NCOUNT             INTEGER*4    LOOP COUNTER                        C
C     NSTEPS             INTEGER*4    NUMBER OF EQUAL BED VOLUME STEPS    C
C                                     TO BE CALCULATED                    C
C                                                                         C
C     OUTPUT             REAL*4       A MATRIX OF SIMULATION STATUS       C
C                                     COL 1 CONTAINS REPORT TIME          C
C                                     COL 2 CONTAINS BED VOLUMES          C
C                                     COL 3 CONTAINS THROUGHPUT           C
C                                     COL 4 GIVES INFLUENT CONCENTRATION  C
C                                     COL 5 GIVES DCO/DT                  C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INFO2 (OUTPUT, NSTEPS)
      REAL*4 OUTPUT
      DIMENSION OUTPUT (250, 5)
      COMMON /UNITS/ IOIN, IOOUT
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200)
      WRITE (IOOUT, 1100)
      DO 100 NCOUNT = 1, NSTEPS
          WRITE (IOOUT, 1300) (OUTPUT (NCOUNT, I), I = 1, 5)
 100  CONTINUE
 1000 FORMAT ('1', 48X, 'SIMULATION MODE INFORMATION')
 1100 FORMAT ('0------------------------------------------------',
     1        '-------------------------------------------------',
     2        '--------------------------------')
 1200 FORMAT (' ', 12X, 'REPORT TIME', 11X, 'BED VOLUMES', 12X,
     1      'THROUGHPUT', 6X, 'CONCENTRATION IN',
     2       16X, 'DCO/DT')
 1300 FORMAT (' ', 11X, E12.5, 10X, E12.5, 10X, E12.5, 10X,
     1       E12.5, 10X, E12.5)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C                                                                         C
C     THIS SUBROUTINE INITIALIZES THE MODEL VARIABLES FOR THE SIMULATION  C
C     OF THE ADVECTIVE-DISPERSIVE EQUATION.                               C
C                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                 DESCRIPTION            C
C     -------------      ---------    ----------------------------------  C
C     I                  INTEGER*4    WORKING VARIABLE                    C
C     J                  INTEGER*4    WORKING VARIABLE                    C
C     NSTEPS             INTEGER*4    NUMBER OF EQUAL BED VOLUME STEPS    C
C                                     TO BE CALCULATED                    C
C                                                                         C
C     C                  REAL*8       VECTOR OF KNOWN INFLUENT            C
C                                     CONCENTRATIONS (SEE VECTOR T)       C
C     DDVDT              REAL*8       DERIVATIVE OF DEPENDENT VARIABLES   C
C     DG                 REAL*8       RATIO OF TOTAL MASS ADSORBABLE      C
```

```
C                                   ON SOLID PHASE TO TOTAL MASS IN    C
C                                   FLUID PHASE                        C
C     DT                REAL*8      DURATION OF EACH TIME STEP         C
C     DTN               REAL*8      REAL EQUIVALENT OF NSTEPS          C
C     DV                REAL*8      DEPENDENT VARIABLE VECTOR          C
C     EPSMCN            REAL*8      MACHINE EPSILON                    C
C     FKAPPA            REAL*8      FLUID-PHASE LOSS RATE              C
C     OUTPUT            REAL*4      A MATRIX OF SIMULATION STATUS      C
C                                   COL 1 CONTAINS REPORT TIME         C
C                                   COL 2 CONTAINS BED VOLUMES         C
C                                   COL 3 CONTAINS THROUGHPUT          C
C                                   COL 4 GIVES INFLUENT CONCENTRATION C
C                                   COL 5 GIVES DCO/DT                 C
C     SKAPPA            REAL*8      SOLID-PHASE LOSS RATE              C
C     T                 REAL*8      VECTOR OF ELAPSED TIME             C
C                                   CORRESPONDING TO CONCENTRATIONS    C
C                                   IN THE VECTOR C                    C
C     TAU               REAL*8      TRAVEL TIME THROUGH COLUMN         C
C     TEMP              REAL*8      TEMPORARY VARIABLE FOR EPSMCN CALC C
C     TEMPT             REAL*8      TEMPORARY VALUE OF REPORT TIME     C
C                                                                      C
C                                                                      C
C     DONE              LOGICAL     FLAG TO ALLOW CONDITIONAL LOOPING  C
C                                   WITHOUT SOFTWARE SUPPORT FOR WHILE C
C                                   LOOPS                              C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INITL (DDVDT, DV, OUTPUT, TO)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA, OUTPUT
      LOGICAL DONE
      DIMENSION DDVDT (101), DV (101), DATA (100,4),
     1          OUTPUT (250,5), C (100), T (100)
      COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, FKAPPA, SKAPPA, C, T,
     2               ICON, NCOL, NOUT, NSTEPS
C
C.....INITIALIZE DEPENDENT VARIABLE AND DERIVATIVE ARRAYS
C
      DO 100 I = 1, 101
         DV (I) = 0.D00
  100 DDVDT (I) = 0.D00
C
C.....COMPUTE SIMULATION OUTPUT POINTS IF DESIRED
C
      IF (NSTEPS .GT. 0) THEN
         TAU = ZCOL / VEL
         DG = 1 + (RHO * (1-THETA) / THETA) * (FK * (CO ** FN)) / CO
         DTN = NSTEPS
         DT = TMESIM / DTN
         TEMPT = 0.D00
         DO 110 I = 1, NSTEPS
            TEMPT = TEMPT + DT
            OUTPUT (I,1) = TEMPT
            OUTPUT (I,2) = TEMPT / TAU
  110    OUTPUT (I,3) = TEMPT / (TAU * DG)
      ENDIF
C
C.....CALCULATE THE MACHINE EPSILON FOR USE AS A LOWER ERROR BOUND
C
      EPSMCN = 1.0
  120 EPSMCN = EPSMCN / 2.0D00
```

```
      TEMP = 1.0D00 + EPSMCN
      IF (TEMP .GT. 1.0D00) GO TO 120
C
C.....CALCULATE THE INFLUENT CONCENTRATIONS FOR USE IN THE
C.....EXPERIMENTAL COMPARISON MODE BY INTERPOLATING FROM THE
C.....LIST OF KNOWN CONCENTRATIONS.
C
      IF (NOUT .GT. 0) THEN
          DO 130 NCOUNT = 1, NOUT, 1
              TEND = DATA (NCOUNT, 1)
              DONE = .FALSE.
              DO 140 NLOOP = 1, (ICON - 1)
              IF (DONE .EQV. .FALSE.) THEN
                  DNTIME = T (NLOOP)
                  UPTIME = T (NLOOP + 1)
                  DNCON  = C (NLOOP)
                  UPCON  = C (NLOOP + 1)
                  IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND))
     1            THEN
                      DATA (NCOUNT, 3) = DNCON + (UPCON - DNCON)
     1                * (TEND - DNTIME) / (UPTIME - DNTIME)
                      DONE = .TRUE.
                  ENDIF
               ENDIF
  140          CONTINUE
               IF (NCOUNT .GT. 1) THEN
                   DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) -
     1             DATA (NCOUNT - 1, 3)) / (DATA (NCOUNT, 1) -
     2             DATA (NCOUNT - 1, 1))
               ELSE
                   DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) - C0)
     1             / (DATA ( NCOUNT, 1) - T0)
               ENDIF
  130      CONTINUE
      ENDIF
C
C.....CALCULATE INFLUENT CONCENTRATION FOR USE IN THE
C.....SIMULATION MODE.
C
      IF (NSTEPS .GT. 0) THEN
          DO 150 NCOUNT = 1, NSTEPS, 1
              TEND = OUTPUT(NCOUNT, 1)
              DONE = .FALSE.
              DO 160 NLOOP = 1, (ICON - 1)
              IF (DONE .EQV. .FALSE.) THEN
                  DNTIME = T (NLOOP)
                  UPTIME = T (NLOOP + 1)
                  DNCON  = C (NLOOP)
                  UPCON  = C (NLOOP + 1)
                  IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND))
     1            THEN
                      OUTPUT (NCOUNT, 4) = DNCON + (UPCON - DNCON)
     1                * (TEND - DNTIME) / (UPTIME - DNTIME)
                      DONE = .TRUE.
                  ENDIF
              ENDIF
  160         CONTINUE
              IF (NCOUNT .GT. 1) THEN
                  OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) -
     1            OUTPUT (NCOUNT - 1, 4)) / (OUTPUT (NCOUNT, 1) -
     2            OUTPUT (NCOUNT - 1, 1))
              ELSE
                  OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - C0)
     1            / (OUTPUT ( NCOUNT, 1) - T0)
              ENDIF
  150     CONTINUE
```

```
C
C.....COMPUTE BED VOLUMES FOR VARIABLE VELOCITY CASE
C
          OLDVEL = VEL
          TEMPT = 0.D00
          DO 170 I = 1, NSTEPS
              TEMPT = TEMPT + DT
              IF (OUTPUT (I, 4) . GT. 1.0E-10) THEN
                  CURVEL = VEL
              ELSE
                  CURVEL = VEL2
              ENDIF
              AVGVEL = (OLDVEL * (TEMPT - DT) + CURVEL * DT) / TEMPT
              TAU = ZCOL / AVGVEL
              OUTPUT (I, 2) = TEMPT / TAU
              OUTPUT (I, 3) = TEMPT / (TAU * DG)
              OLDVEL = AVGVEL
 170      CONTINUE
      ENDIF
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C                                                                         C
C     THIS SUBROUTINE COMPUTES VARIABLE GROUPINGS TO MINIMIZE EFFORT IN   C
C     THE SOLUTION OF THE SET OF SIMULTANEOUS NONLINEAR EQUATIONS.        C
C                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME       PRECISION              DESCRIPTION              C
C     -------------       ---------    ---------------------------------  C
C     I                   INTEGER*4    TIME STEP INDEX                    C
C     NCOL                INTEGER*4    NUMBER OF NODES IN Z DIRECTION     C
C     NDOF                INTEGER*4    NUMBER OF DEGREES OF FREEDOM       C
C     NOUT                INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT C
C     NSTEPS              INTEGER*4    NUMBER OF EQUAL BED VOLUME STEPS   C
C                                      TO BE CALCULATED                   C
C                                                                         C
C                                                                         C
C     DATA                REAL*4       EXPERIMENTAL INPUT DATA            C
C                                      COLUMN 1 CONTAINS REPORT TIMES     C
C                                      COLUMN 2 GIVES EXPERIMENTAL        C
C                                      CONCENTRATION                      C
C                                      COLUMN 3 GIVES INFLUENT            C
C                                      CONCENTRATION                      C
C                                      COLUMN 4 GIVES DCO/DT              C
C     DGN                 REAL*8       DIAGONAL VARIABLE GROUPING TERMS   C
C     DGNLOW              REAL*8       LOWER DIAGONAL VARAIBLE GROUPINGS  C
C     DGNUP               REAL*8       UPPER DIAGONAL VARIABLE GROUPINGS  C
C     DH                  REAL*8       HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DZ                  REAL*8       COLUMN STEP SIZE IN Z DIRECTION    C
C     FK                  REAL*8       FREUNDLICH ISOTHERM COEFFICIENT    C
C     FKAPPA              REAL*8       FLUID-PHASE LOSS RATE              C
C     FN                  REAL*8       FREUNDLICH ISOTHERM EXPONENT       C
C     REALND              REAL*8       NUMBER OF NODES IN COLUMN          C
C     RFGRP               REAL*8       RETARDATION FACTOR GROUPING        C
C     RHO                 REAL*8       SOLID PHASE PARTICLE DENSITY       C
C     SKAPPA              REAL*8       SOLID-PHASE LOSS RATE              C
C     SPLGRP              REAL*8       SOLID PHASE LOSS GROUP             C
C     THETA               REAL*8       POROSITY                           C
C     VEL                 REAL*8       PORE VELOCITY                      C
C     VEL2                REAL*8       DESORPTION PORE VELOCITY           C
```

```
C     ZCOL                REAL*8       LENGTH OF COLUMN (Z DIRECTION)    C
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE GRPFRM
      IMPLICIT REAL*8(A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), C (100), T (100)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, RFGRP, SPLGRP
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, FKAPPA, SKAPPA, C, T,
     2               ICON, NCOL, NOUT, NSTEPS
C
C.....COMPUTE MACROSCOPIC ADVECTIVE-DSIPERSIVE EQUATION FLUID PHASE GROUPINS
C
      REALND = NCOL
      DZ = ZCOL / (REALND - 1.D00)
      DGNLOW = DH / (DZ ** 2.D00) + VEL / (2.D00 * DZ)
      DGN = -2.D00 * DH / (DZ ** 2.D00) - FKAPPA
      DGNUP = DH / (DZ ** 2.D00) - VEL / (2.D00 *DZ)
      RFGRP = FK * FN * (1 - THETA) * RHO / THETA
      SPLGRP = (RHO * (1 - THETA) / THETA) * SKAPPA * FK
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE     C
C     DEPENDENT VARIABLE AS A FUNCTION OF TIME. THE EQUATIONS ARE      C
C     SOLVED USING THE VARIABLE GROUPINGS PREVIOUSLY DERIVED IN ORDER  C
C     TO SAVE COMPUTATIONAL EFFORT.                                    C
C                                                                      C
C     ALL DEPENDENT VARIABLE ARE STORED IN A SINGLE ARRAY DV, AND ALL  C
C     TEMPORAL DERIVATIVES OF THE DEPENDENT VARIABLE ARE STORED IN A   C
C     SINGLE ARRAY TERMED DDVDT.                                       C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION          DESCRIPTION                 C
C     -------------     ---------    --------------------------------  C
C     I                 INTEGER*4    TEMPORARY VARAIBLE                C
C     J                 INTEGER*4    TEMPORARY VARAIBLE                C
C     NCOL              INTEGER*4    NUMBER OF NODES IN Z DIRECTION    C
C     NDOF              INTEGER*4    NUMBER OF DEGREES OF FREEDOM      C
C     NEND              INTEGER*4    VARIABLE INDEX FOR LAST COLUMN NODE C
C     NSTART            INTEGER*4    LOOP STARTING POSITION            C
C     NSTEPS            INTEGER*4    NUMBER OF EQUAL BED VOLUME STEPS  C
C                                    TO BE CALCULATED                  C
C     NSTOP             INTEGER*4    LOOP STOPPING POSITION            C
C                                                                      C
C                                                                      C
C     DCODT             REAL*8       CHANGE IN INFLUENT CONCENTRAION   C
C                                    WITH RESPECT TO TIME              C
C     DDVDT             REAL*8       DERIVATIVE OF DEPENDENT VARIABLES C
C     DGN               REAL*8       DIAGONAL VARIABLE GROUPING TERMS  C
C     DGNLOW            REAL*8       LOWER DIAGONAL VARAIBLE GROUPINGS C
C     DGNUP             REAL*8       UPPER DIAGONAL VARIABLE GROUPINGS C
C     DV                REAL*8       DEPENDENT VARIABLE VECTOR         C
```

```
C     FK                REAL*8      FREUNDLICH ISOTHERM COEFFICIENT   C
C     FKAPPA            REAL*8      FLUID-PHASE LOSS RATE             C
C     RFGRP             REAL*8      RETARDATION FACTOR GROUPING       C
C     RFI               REAL*8      RETARDATION FACTOR AT NODE I      C
C     SKAPPA            REAL*8      SOLID-PHASE LOSS RATE             C
C     SPL               REAL*8      SOLID PHASE LOSS AT NODE I        C
C     SPLGRP            REAL*8      SOLID-PHASE LOSS GROUPING         C
C                                                                     C
C                                                                     C
C                                                                     C
C                                                                     C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE CPRIME (NVRB, TIME, DV, DDVDT)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DV (101), DDVDT (101), DATA (100,4), C (100), T(100)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, RFGRP, SPLGRP
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, FKAPPA, SKAPPA, C, T,
     2               ICON, NCOL, NOUT, NSTEPS
      COMMON /FLUID/ DCODT
C
C.....FORM FLUID PHASE DERIVATIVES
C
      NDOF = NCOL
C
C.....FIRST THE INLET CONDITION, A DIRICHLET BOUNDARY
C
      DDVDT (1) =  DCODT
C
C.....SECOND THE INTERIOR FLUID PHASE DERIVATIVES
C
      NEND = NDOF - 1
      DO 100 I = 2, NEND
      IF (DV (I) .GT. 0.D00) THEN
         RFI = 1.D00 + RFGRP * (DV (I) ** (FN - 1.D00))
         SPL = SPLGRP * (DV (I) ** FN)
      ELSE
         RFI = 1.D00
         SPL = 0.D00
      ENDIF
  100 DDVDT (I) = (DGNLOW * DV (I-1) + DGN * DV (I) + DGNUP * DV (I+1)
     1            - SPL) / RFI
C
C.....THIRD THE OUTLET BOUNDARY CONDITION, A NEUMANN BOUNDARY
C
      I = NDOF
      IF (DV (I) .GT. 0.D00) THEN
         RFI = 1.D00 + RFGRP * (DV (I) ** (FN - 1.D00))
         SPL = SPLGRP * (DV (I) ** FN)
      ELSE
         RFI = 1.D00
         SPL = 0.D00
      ENDIF
      DDVDT (I) = ((DGNLOW + DGNUP) * DV (I - 1) + DGN * DV (I)
     1            - SPL) / RFI
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C                                                                     C
```

```
C     THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE           C
C     DDVDT ARRAY AS A FUNCTION OF THE DEPENDENT VARIABLE (DV).              C
C     SINCE THE FINITE DIFFERENCE OPTION IS USED IN THE CALL TO DGEAR        C
C     (I.E. MITER= -2) THIS ROUTINE IS A BLANK.                              C
C                                                                            C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE FCNJ (N, X, Y, PD)
      REAL*8 Y(N), PD(N,N), X
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                            C
C     THIS SUBROUTINE USES THE EXPERIMENTAL DATA INPUT AND THE MODEL          C
C     SIMULATION OUTPUT TO COMPUTE THE MODEL RESIDUAL, RESIDUAL SQUARED,     C
C     SUM OF RESIDUALS, AND SUM OF RESIDUALS SQUARED. THESE COMPUTATIONS     C
C     ARE OUTPUT TO UNIT IOOUT.                                              C
C                                                                            C
C                                                                            C
C     VARIABLE MAP:                                                          C
C     ------------                                                           C
C                                                                            C
C     VARIABLE NAME      PRECISION              DESCRIPTION                  C
C     -------------      ---------     ----------------------------------    C
C     I                  INTEGER*4     TIME STEP INDEX                       C
C     IOIN               INTEGER*4     UNIT NUMBER TO READ FROM              C
C     IOOUT              INTEGER*4     UNIT NUMBER TO WRITE TO               C
C     NCOL               INTEGER*4     NUMBER OF NODES IN Z DIRECTION        C
C     NDOF               INTEGER*4     NUMBER OF DEGREES OF FREEDOM          C
C     NDV                INTEGER*4     INDEX OF OUTLET CONCENTRATION         C
C     NOUT               INTEGER*4     NUMBER OF POINTS SIMULATED            C
C     NSTEPS             INTEGER*4     NUMBER OF EQUAL BED VOLUME STEPS      C
C                                      TO BE CALCULATED                      C
C                                                                            C
C                                                                            C
C     CNORM              REAL*8        NORMALIZED OUTLET CONCENTRATION       C
C     CO                 REAL*8        INITIAL CONCENTRATION                 C
C     DATA               REAL*4        EXPERIMENTAL INPUT DATA               C
C                                      COLUMN 1 CONTAINS REPORT TIMES        C
C                                      COLUMN 2 GIVES EXPERIMENTAL           C
C                                      CONCENTRATION                         C
C                                      COLUMN 3 GIVES INFLUENT               C
C                                      CONCENTRATION                         C
C                                      COLUMN 4 GIVES DCO/DT                 C
C     DV                 REAL*8        DEPENDENT VARIABLE VECTOR             C
C     FKAPPA             REAL*8        FLUID-PHASE LOSS RATE                 C
C     RDSQSM             REAL*8        SUM OF RESIDUALS SQUARRED             C
C     RSD                REAL*8        RESIDUAL OF POINT AT OUTLET           C
C     RSDSQ              REAL*8        RESIDUAL SQUARED OF POINT AT OUTLET C
C     RSDSUM             REAL*8        SUM OF RESIDUALS                      C
C     SKAPPA             REAL*8        SOLID-PHASE LOSS RATE                 C
C                                                                            C
C                                                                            C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE REPORT (I, DV, RSDSUM, RDSQSM, RSD)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), DV(101), C (100), T (100)
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
```

```
     1                EPSLON, CO, FK, FN, FKAPPA, SKAPPA, C, T,
     2                ICON, NCOL, NOUT, NSTEPS
      COMMON /UNITS/ IOIN, IOOUT
      IF (I .EQ. 1) THEN
         WRITE (IOOUT, 1000)
         WRITE (IOOUT, 1100)
         WRITE (IOOUT, 1200)
         WRITE (IOOUT, 1100)
         RSDSUM = 0.D00
         RDSQSM = 0.D00
      ENDIF
      CNORM = DV (NCOL) / CO
      RSD = DATA (I,2) - CNORM
      RSDSQ = RSD * RSD
      RSDSUM = RSDSUM + RSD
      RDSQSM = RDSQSM + RSDSQ
      WRITE (IOOUT, 1300) DATA (I,1), DATA (I,2), CNORM, RSD, RSDSQ
      IF (I .GE. NOUT) THEN
         WRITE (IOOUT, 1100)
         WRITE (IOOUT, 1400) RSDSUM, RDSQSM
      ENDIF
 1000 FORMAT ('1',58X,'RESIDUAL REPORT')
 1100 FORMAT ('0--------------------------------------------------',
     1        '---------------------------------------------------',
     2        '---------------------------------')
 1200 FORMAT ('0',19X,'TIME',10X,'C/Co (INPUT)',10X,'C/Co (PREDICTED)',
     1        14X,'RESIDUAL',10X,'RESIDUAL SQUARED')
 1300 FORMAT (' ',11X,E12.5,10X,E12.5,14X,E12.5,10X,E12.5,14X,E12.5)
 1400 FORMAT (' ',81X,E12.5,14X,E12.5)
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C                                                                              C
C     THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE                  C
C     COMPLETION OF SIMULATION FOR EXPERIMENTAL COMPARISON.                    C
C                                                                              C
C                                                                              C
C     VARIABLE MAP:                                                            C
C     ------------                                                             C
C                                                                              C
C     VARIABLE NAME       PRECISION                   DESCRIPTION              C
C     -------------       ---------     ------------------------------------   C
C     I                   INTEGER*4     LOOP INDEX                             C
C     IFLAG1              INTEGER*4     FLAG FOR ABRUPT CONCENTRATION          C
C                                       CHANGE                                 C
C     IFLAG2              INTEGER*4     FLAG FOR DESORPTION                    C
C     IOIN                INTEGER*4     UNIT NUMBER TO READ FROM               C
C     IOOUT               INTEGER*4     UNIT NUMBER TO WRITE TO                C
C     NCOL                INTEGER*4     NUMBER OF NODES IN Z DIRECTION         C
C     NOUT                INTEGER*4     NUMBER OF TEMPORAL STEPS TO OUTPUT     C
C                                                                              C
C                                                                              C
C                                                                              C
C     BAL                 REAL*8        RATIO OF MASS IN TO MASS ACCOUNTED     C
C     CO                  REAL*8        INITIAL CONCENTRATION                  C
C     DATA                REAL*4        EXPERIMENTAL INPUT DATA                C
C                                       COLUMN 1 CONTAINS REPORT TIMES         C
C                                       COLUMN 2 GIVES EXPERIMENTAL            C
C                                       CONCENTRATION                          C
C                                       COLUMN 3 GIVES INFLUENT                C
C                                       CONCENTRATION                          C
C                                       COLUMN 4 GIVES DCO/DT                  C
C     DV                  REAL*8        DEPENDENT VARIABLE VECTOR              C
```

```
C      FK              REAL*8       FREUNDLICH ISOTHERM COEFFICIENT     C
C      FN              REAL*8       FREUNDLICH ISOTHERM EXPONENT        C
C      OUTCON          REAL*8       EXIT CONCENTRATION VECTOR           C
C      RCOL            REAL*8       REAL EQUIVALENT OF NCOL             C
C      THETA           REAL*8       POROSITY                            C
C      TOTDRY          REAL*8       TOTAL MASS LEFT ON SOLID PHASE      C
C      TOTINF          REAL*8       TOTAL MASS INJECTED INTO COLUMN     C
C      TOTOUT          REAL*8       TOTAL MASS TO EXIT COLUMN           C
C      TOTWET          REAL*8       TOTAL MASS LEFT IN LIQUID PHASE     C
C      VEL             REAL*8       PORE VELOCITY                     . C
C      VEL2            REAL*8       DESORPTION PORE VELOCITY            C
C      ZCOL            REAL*8       LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                       C
C                                                                       C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE MASS1 (OUTCON, DV, CO, ZCOL, THETA, FK, FN, VEL, VEL2,
     1             RHO, DATA, NOUT, NCOL)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100, 4), DV (101), OUTCON (250)
      COMMON /UNITS/ IOIN, IOOUT
C
      IFLAG1 = 0
      IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
      VEL1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
C
C     CALCULATE  CONTAMINANT INJECTED DURING FIRST
C     REPORT PERIOD.
C
          TOTINF = CO * DATA (1, 1) * VEL * THETA +
     1              0.5 * (DATA (1, 3) - CO) * DATA (1, 1) *
     2              VEL * THETA
C
C     CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
C
      DO 100 I = 2, (NOUT - 1)
         IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
     1       (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
             IFLAG1 = 1
             TOTINF = TOTINF +
     1                 DATA (I, 3) * (DATA (I, 1) - DATA (I - 1, 1)) *
     2                 VEL * THETA
         ELSE
             TOTINF = TOTINF +
     1                 DATA (I - 1, 3) *
     2                 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA +
     3                 0.5 * (DATA (I, 3) - DATA (I - 1, 3)) *
     4                 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
         ENDIF
 100  CONTINUE
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
C     CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
```

```
C      THE FIRST REPORT PERIOD
C
       TOTOUT = (0.75 * 0.D0 + 0.25 * OUTCON (1)) *
     1          DATA (1, 1) / 2.D0 * VEL * THETA
C
C      CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C      PERIOD
C
       TOTOUT = TOTOUT +
     1          ((0.75 * OUTCON (1) + 0.25 * 0.D0) *
     2          (DATA (1, 1) - 0.D0) / 2.D0 +
     3          (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
     4          (DATA (2, 1) - DATA (1, 1)) / 2.D0) *
     5          VEL * THETA
C
C      CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
C      PERIOD
C
       DO 110 I = 2, (NOUT - 1)
C
C      RESET VELOCITY IF DESORPTION BEGINS
C
          IF (DATA (I, 3) .LT. 1.0E-10) THEN
             VEL = VEL2
             IFLAG2 = 1
          ENDIF
C
          TOTOUT = TOTOUT +
     1             ((0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
     2             (DATA (I, 1) - DATA (I - 1, 1)) / 2.D0 +
     3             (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
     4             (DATA (I + 1, 1) - DATA (I, 1)) / 2.D0) *
     5             VEL * THETA
 110   CONTINUE
C
C      CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C      REPORT PERIOD
C
       TOTOUT = TOTOUT +
     1          (0.75 * OUTCON (NOUT) +
     2          0.25 * OUTCON (NOUT - 1)) *
     3          (DATA (NOUT, 1) - DATA (NOUT - 1, 1)) / 2.D0 *
     4          VEL * THETA
C
C      RESTORE VELOCITY
C
       VEL = VEL1
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C      CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
C
       RCOL = REAL (NCOL)
       TOTWET = (0.75 * DV (1) + 0.25 * DV (2)) *
     1          ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C      CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE
C
       DO 120 I = 2, (NCOL - 1)
          TOTWET = TOTWET +
     1             (0.75 * DV (I) + 0.125 * DV (I - 1) +
     2             0.125 * DV (I + 1)) *
     3             ZCOL / (RCOL - 1.D0) * THETA
 120   CONTINUE
C
C      CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
```

```
C
      TOTWET = TOTWET + (0.25 * DV (NCOL - 1) + 0.75 * DV (NCOL)) *
     1         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE
C
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE
C
      TOTDRY = FK * (0.75 * DV (1) + 0.25 * DV (2)) ** FN *
     1         ZCOL / (RCOL - 1.D0) / 2.D0 * RHO * (1.D0 - THETA)
C
C     CALCULATE CONAMINANT ON TYPICAL NODES
C
      DO 130 I = 2, (NCOL - 1)
         TOTDRY = TOTDRY + FK *
     1            (0.75 * DV (I) + 0.125 * DV (I - 1)
     2            + 0.125 * DV (I + 1)) ** FN *
     3            ZCOL / (RCOL - 1.D0) * RHO * (1.D0 - THETA)
 130  CONTINUE
C
C     CALCULATE CONTAMINANT NEAR EXIT
C
      TOTDRY = TOTDRY +
     1         FK * (0.25 * DV (NCOL - 1) + 0.75 * DV (NCOL)) ** FN
     2         * ZCOL / (RCOL - 1.D0) / 2.D0 * RHO * (1.D0 -THETA)
C
C
C.....COMPUTE MASS BALANCE
C
C
      BAL = TOTINF / ( TOTOUT + TOTWET + TOTDRY )
C
C
C.....OUTPUT THE MASS BALANCE FINDINGS
C
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TOTINF
      WRITE (IOOUT, 1300) TOTOUT
      WRITE (IOOUT, 1400) TOTWET
      WRITE (IOOUT, 1500) TOTDRY
      WRITE (IOOUT, 1600) BAL
      IF (IFLAG1 .EQ. 1) THEN
         WRITE (IOOUT, 1700)
      ENDIF
      IF (IFLAG2 .EQ. 1) THEN
         WRITE (IOOUT, 1800)
      ENDIF
C
C
 1000 FORMAT ('1', 35X, 'EXPERIMENTAL MASS BALANCE REPORT')
 1100 FORMAT ('0---------------------------------------------------',
     1        '----------------------------------------------------',
     2        '----------------------------------')
 1200 FORMAT (' ', 20X, 'TOTAL MASS INJECTED = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1300 FORMAT (' ', 20X, 'TOTAL MASS RELEASED = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = ',
     1       E10.4)
```

```
 1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
     1        ' IN INFLUENT CONCENTRATION.')
 1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
     1        ' DURING DESORPITON.')
C
C
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE          C
C     COMPLETION OF SIMULATION IN SIMULATION MODE.                     C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME      PRECISION               DESCRIPTION           C
C     -------------      ---------    ---------------------------------C
C     I                  INTEGER*4    LOOP INDEX                       C
C     IFLAG1             INTEGER*4    FLAG FOR ABRUPT CONCENTRATION    C
C                                     CHANGE                           C
C     IFLAG2             INTEGER*4    FLAG FOR DESORPTION              C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM         C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO          C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION   C
C     NSTEPS             INTEGER*4    NUMBER OF TEMPORAL STEPS         C
C                                                                      C
C                                                                      C
C                                                                      C
C     BAL                REAL*8       RATIO OF MASS IN TO MASS ACCOUNTED C
C     CO                 REAL*8       INITIAL CONCENTRATION            C
C     DV                 REAL*8       DEPENDENT VARIABLE VECTOR        C
C     FK                 REAL*8       FREUNDLICH ISOTHERM COEFFICIENT  C
C     FN                 REAL*8       FREUNDLICH ISOTHERM EXPONENT     C
C     OUTCON             REAL*8       EXIT CONCENTRATION VECTOR        C
C     OUTPUT             REAL*4       SIMULATION STATUS MATRIX         C
C                                     COL 1 GIVES REPORT TIMES         C
C                                     COL 2 GIVES BED VOLUMES          C
C                                     COL 3 GIVES THROUGHPUT           C
C                                     COL 4 GIVES INFLUENT CONCENTRATION C
C                                     COL 5 GIVES DCO/DT               C
C     RCOL               REAL*8       REAL EQUIVALENT OF NCOL          C
C     THETA              REAL*8       POROSITY                         C
C     TOTDRY             REAL*8       TOTAL MASS LEFT ON SOLID PHASE   C
C     TOTINF             REAL*8       TOTAL MASS INJECTED INTO COLUMN  C
C     TOTOUT             REAL*8       TOTAL MASS TO EXIT COLUMN        C
C     TOTWET             REAL*8       TOTAL MASS LEFT IN LIQUID PHASE  C
C     VEL                REAL*8       PORE VELOCITY                    C
C     VEL2               REAL*8       DESORPTION PORE VELOCITY         C
C     ZCOL               REAL*8       LENGTH OF COLUMN (Z DIRECTION)   C
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE MASS2 (OUTCON, DV, CO, ZCOL, THETA, FK, FN, VEL, VEL2,
     1             RHO, OUTPUT, NSTEPS , NCOL)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 OUTPUT
```

```
      DIMENSION OUTPUT (250, 5), DV (101), OUTCON (250)
      COMMON /UNITS/ IOIN, IOOUT
C
      IFLAG1 = 0
      IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
      VEL1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
C
C     CALCULATE  CONTAMINANT INJECTED DURING FIRST
C     REPORT PERIOD.
C
          TOTINF = CO * OUTPUT (1, 1) * VEL * THETA +
     1             0.5 * (OUTPUT (1, 4) - CO) * OUTPUT (1, 1) *
     2             VEL * THETA
C
C     CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
C
      DO 100 I = 2, (NSTEPS - 1)
         IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     1       (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
             IFLAG1 = 1
             TOTINF = TOTINF +
     1                OUTPUT (I, 4) *
     2                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     3                VEL * THETA
         ELSE
             TOTINF = TOTINF +
     1                OUTPUT (I - 1, 4) *
     2                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     3                VEL * THETA +
     4                0.5 * (OUTPUT (I, 4) - OUTPUT (I - 1, 4)) *
     5                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     6                VEL * THETA
         ENDIF
 100  CONTINUE
C
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
C     CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
C     THE FIRST REPORT PERIOD
C
      TOTOUT = (0.75 * 0.D0 + 0.25 * OUTCON (1)) *
     1         OUTPUT (1, 1) / 2.D0 * VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C     PERIOD
C
      TOTOUT = TOTOUT +
     1         ((0.75 * OUTCON (1) + 0.25 * 0.D0) *
     2         (OUTPUT (1, 1) - 0.D0) / 2.D0 +
     3         (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
     4         (OUTPUT (2, 1) - OUTPUT (1, 1)) / 2.D0) *
     5         VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
C     PERIOD
C
      DO 110 I = 2, (NSTEPS - 1)
C
C     RESET VELOCITY IF DESORPTION BEGINS
```

```
C
         IF (OUTPUT (I, 4) .LT. 1.0E-10) THEN
            VEL = VEL2
            IFLAG2 = 1
         ENDIF
C
         TOTOUT = TOTOUT +
     1            ((0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
     2            (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) / 2.D0 +
     3            (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
     4            (OUTPUT (I + 1, 1) - OUTPUT (I, 1)) / 2.D0) *
     5            VEL * THETA
  110 CONTINUE
C
C     CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C     REPORT PERIOD
C
      TOTOUT = TOTOUT +
     1         (0.75 * OUTCON (NSTEPS) +
     2         0.25 * OUTCON (NSTEPS - 1)) *
     3         (OUTPUT (NSTEPS, 1) - OUTPUT (NSTEPS - 1, 1)) / 2.D0 *
     4         VEL * THETA
C
C     RESTORE VELOCITY
C
      VEL = VEL1
C
C
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
C
      RCOL = REAL (NCOL)
      TOTWET = (0.75 * DV (1) + 0.25 * DV (2)) *
     1         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C     CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE
C
      DO 120 I = 2, (NCOL - 1)
         TOTWET = TOTWET +
     1            (0.75 * DV (I) + 0.125 * DV (I - 1) +
     2            0.125 * DV (I + 1)) *
     3            ZCOL / (RCOL - 1.D0) * THETA
  120 CONTINUE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
C
      TOTWET = TOTWET + (0.25 * DV (NCOL - 1) + 0.75 * DV (NCOL)) *
     1         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE
C
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE
C
      TOTDRY = FK * (0.75 * DV (1) + 0.25 * DV (2)) ** FN *
     1         ZCOL / (RCOL - 1.D0) / 2.D0 * RHO * (1.D0 - THETA)
C
C     CALCULATE CONAMINANT ON TYPICAL NODES
C
      DO 130 I = 2, (NCOL - 1)
         TOTDRY = TOTDRY + FK *
     1            (0.75 * DV (I) + 0.125 * DV (I - 1)
     2            + 0.125 * DV (I + 1)) ** FN *
```

```
     3             ZCOL / (NCOL - 1.DO) * RHO * (1.DO - THETA)
 130  CONTINUE
C
C     CALCULATE CONTAMINANT NEAR EXIT
C
      TOTDRY = TOTDRY +
     1         FK * (0.25 * DV (NCOL - 1) + 0.75 * DV (NCOL)) ** FN
     2         * ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO -THETA)
C
C
C.....COMPUTE MASS BALANCE
C
C
      BAL = TOTINF / ( TOTOUT + TOTWET + TOTDRY )
C
C
C.....OUTPUT THE MASS BALANCE FINDINGS
C
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TOTINF
      WRITE (IOOUT, 1300) TOTOUT
      WRITE (IOOUT, 1400) TOTWET
      WRITE (IOOUT, 1500) TOTDRY
      WRITE (IOOUT, 1600) BAL
      IF (IFLAG1 .EQ. 1) THEN
         WRITE (IOOUT, 1700)
      ENDIF
      IF (IFLAG2 .EQ. 1) THEN
         WRITE (IOOUT, 1800)
      ENDIF
C
C
 1000 FORMAT ('1', 35X, 'SIMULATION MASS BALANCE REPORT ')
 1100 FORMAT ('0--------------------------------------------------',
     1        '--------------------------------------------------',
     2        '---------------------------------')
 1200 FORMAT (' ', 20X, 'TOTAL MASS INJECTED = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1300 FORMAT (' ', 20X, 'TOTAL MASS RELEASED = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = ',
     1       E10.4)
 1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
     1        ' IN INFLUENT CONCENTRATION.')
 1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
     1        ' DURING DESORPITON.')
C
C
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                     SORPTION MODEL :                                 C
C               LANGMUIR SECOND ORDER MODEL                            C
C                      MAINFRAME VERSION                               C
C                                                                      C
C                                                                      C
C                                                                      C
C     THIS PROGRAM USES THE DGEAR METHOD TO SOLVE THE ADVECTIVE        C
C     DISPERSIVE EQUATION FOR THE CASE OF THE LANGMUIR SECOND ORDER    C
C     SORPTION MODEL.  THIS MODEL ASSUMES THAT SORPTION IS A SECOND-   C
C     ORDER PROCESS DEPENDENT ON THE PRODUCT OF THE FLUID-PHASE        C
C     CONCENTRATION AND THE DIFFERENCE BETWEEN THE SOLID-PHASE CAPACITYC
C     AND THE SOLID PHASE CONCENTRAION, AND DESORPTION IS A FUNCTION   C
C     OF THE SOLID PHASE CONCENTRATION ALONE.                          C
C                                                                      C
C                                                                      C
C     THE EQUATIONS ARE APPROXIMATED USING THE FINITE DIFFERENCE METHODC
C     RESULTING IN A SET OF DQ/DT AND DC/DT EQUATIONS.  THESE ARE      C
C     SOLVED USING DGEAR --- AN IMSL SUBROUTINE.                       C
C                                                                      C
C                                                                      C
C     THE PROGRAM IS PRESENTLY LIMITED TO 100 NODES.                   C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION               DESCRIPTION            C
C     -------------     ---------   ---------------------------------  C
C     I                 INTEGER*4   TIME STEP INDEX                    C
C     IDUMMY            INTEGER*4   DUMMY WORKING ARRAY FOR DGEAR      C
C     IER               INTEGER*4   ERROR CODE RETURN INDEX FOR DGEAR  C
C     INDEX             INTEGER*4   CALL INDICATOR INDEX FOR GEAR      C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     IWK               INTEGER*4   WORKING ARRAY FOR DGEAR            C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION     C
C     NDOF              INTEGER*4   NUMBER OF DEGRFES OF FREEDOM       C
C     NLC               INTEGER*4   NUMBER OF LOWER CODIAGONALS-DGEAR  C
C     NOUT              INTEGER*4   NUMBER OF TEMPORAL STEPS TO OUTPUT C
C     NUC               INTEGER*4   NUMBER OF UPPER CODIAGONALS-DGEAR  C
C     NSTEPS            INTEGER*4   NUMBER OF BED VOLUME STEP TO OUTPUTC
C                                                                      C
C                                                                      C
C     B                 REAL*8      LANGMUIR ENERGY CONSTANT           C
C     CO                REAL*8      INITIAL CONCENTRATION              C
C     CNORM             REAL*8      NORMALIZED EXIT CONCENTRATION      C
C     CPRIME            REAL*8      EXTERNAL NAME OF DC/DT SUBROUTINE  C
C     CTEMP             REAL*8      EXIT CONCENTRATION                 C
C     DATA              REAL*4      EXPERIMENTAL INPUT DATA            C
C                                   COLUMN 1 CONTAINS REPORT TIMES     C
C                                   COLUMN 2 GIVES EXPERIMENTAL        C
C                                   CONCENTRATION                      C
C                                   COLUMN 3 GIVES INFLUENT            C
C                                   CONCENTRATION                      C
C                                   COLUMN 4 GIVES DCO/DT              C
C     DDVDT             REAL*8      DERIVATIVE OF DEPENDENT VARIABLES  C
C     DGN               REAL*8      DIAGONAL VARIABLE GROUPING TERMS   C
C     DGNLOW            REAL*8      LOWER DIAGONAL VARAIBLE GROUPINGS  C
C     DGNUP             REAL*8      UPPER DIAGONAL VARIABLE GROUPINGS  C
```

```
C     DH               REAL*8       HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DH1              REAL*8       SORPTION PHASE HYDRODYNAMIC         C
C                                   DISPERSION COEFFICIENT              C
C     DUMMY            REAL*8       DUMMY ROUTINE REQUIRED BY DGEAR     C
C     DV               REAL*8       DEPENDENT VARIABLE VECTOR           C
C     EPSLON           REAL*8       ALLOWABLE ERROR FOR DGEAR           C
C     FCNJ             REAL*8       EXTERNAL NAME OF JACOBIAN ROUTINE   C
C     FKAPPA           REAL*8       FLUID-PHASE LOSS RATE               C
C     FLSP             REAL*8       FLUID-PHASE LOSS GROUPING           C
C     FLDSP            REAL*8       FLUID-PHASE LOSS GROUPING           C
C     H                REAL*8       STEP SIZE FOR DGEAR                 C
C     OUTPUT           REAL*8       A MATRIX OF SIMULATION STATUS       C
C                                   COL 1 CONTAINS REPORT TIME          C
C                                   COL 2 CONTAINS BED VOLUMES          C
C                                   COL 3 CONTAINS THROUGHPUT           C
C                                   COL 4 GIVES INFLUENT CONCENTRATION  C
C                                   COL 5 GIVES DCO/DT                  C
C     QO               REAL*8       LANGMUIR CAPACITY CONSTANT          C
C     RDSQSM           REAL*8       SUM OF RESIDUALS SQUARED            C
C     RHO              REAL*8       SOLID PHASE PARTICLE DENSITY        C
C     RSDSUM           REAL*8       SUM OF RESIDUALS                    C
C     SDUMMY           REAL*8       DUMMY ROUTINE FOR DGEAR             C
C     SKAPPA           REAL*8       SOLID-PHASE LOSS RATE               C
C     SLDSP            REAL*8       SOLID-PHASE LOSS GROUPING           C
C     SOK              REAL*8       SECOND-ORDER RATE CONSTANT          C
C     THETA            REAL*8       POROSITY                            C
C     TIME             REAL*8       TIME OF OUTPUT SIMULATION POINT     C
C     THESIM           REAL*8       NUMBER OF BED VOLUMES TO SIMULATE   C
C     TO               REAL*8       INITIAL TIME FOR DGEAR              C
C     VEL              REAL*8       PORE VELOCITY                       C
C     VEL1             REAL*8       SORPTION PORE VELOCITY              C
C     VEL2             REAL*8       DESORPTION PORE VELOCITY            C
C     WK               REAL*8       WORKING ARRY NEEDED BY DGEAR        C
C     ZCOL             REAL*8       LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                       C
C                                                                       C
C     TITLE            CHARACTER*72 TITLE OF SIMULATION RUN             C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 SDUMMY, DATA, OUTPUT
      CHARACTER*72 TITLE
C
C
      DIMENSION F(100), X(1)
      DIMENSION IWK (101)
      DIMENSION DATA (100,4), DV (101), DDVDT (101), OUTPUT (250,5),
     1        WK (1919), C (100), T(100), OUTCON (250)
      COMMON /DBAND/ NLC, NUC
      COMMON /GEAR/ DUMMY(48), SDUMMY(4), IDUMMY(38)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, FLSP, FLDSP, SLDSP
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1              EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
     2              ICON, NCOL, NOUT, NSTEPS
      COMMON /WORDS/ TITLE
      COMMON /FLUID/ DCODT
      COMMON /UNITS/ IOIN, IOOUT
      EXTERNAL CPRIME, FCNJ
      IOIN = 1
      IOOUT = 3
```

```
      CALL READ
      VEL1 = VEL
      DH1 = DH
      CALL ECHO
      CALL ERRSET (208, 256, -1, 1)
C
C.....COMPUTE MODEL OUTPUT AT EXPERIMENTAL POINTS IF DESIRED
C
      IF (NOUT .GT. 0) THEN
        CALL INITL (DDVDT, DV, OUTPUT, TO)
        CALL GRPFRM
        CALL INFO1 (DATA, NOUT)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
         NDOF = 2 * NCOL
         NCOLEX = 2 * NCOL - 1
         TO = 0.D00
         H = 1.D-08
         INDEX = 1
         DV (1) = CO
         NLC = 2
         NUC = 2
C
C.....TIME LOOP FOR SIMULTATION AND OUTPUT AT EXPERIMENTAL POINTS
C
         DO 100 I = 1,NOUT
            IF (I .GT. 1) THEN
               IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
     1             (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
                   H = 1.D-08
                   INDEX = 1
                   DCODT = 0.000
                   DV (1) = DATA (I, 3)
                   WRITE (IOOUT, 1300)
                   IF (DV (1) .LT. 1.0E-10) THEN
                      VEL = VEL2
                      DH = DH1 * VEL2 / VEL1
                      CALL GRPFRM
                      WRITE (IOOUT, 1400)
                   ENDIF.
               ELSE
                   DCODT = DATA (I, 4)
               ENDIF
            ELSE
               DCODT = DATA (I, 4)
            ENDIF
            TIME = DATA (I, 1)
            CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
     1                  EPSLON, 2, -2, INDEX, IWK, WK, IER)
            CALL REPORT (I, DV, RSDSUM, RDSQSM, RSD)
            OUTCON (I) = DV (NCOLEX)
  100    F(I) = RSD
C
C.....RESTORE ORIGINAL PARAMETERS
C
         DH = DH1
         VEL = VEL1
C
C.....CHECK THE MASS BALANCE
C
         CALL MASS1 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
     1               RHO, DATA, NOUT, NCOL)
C
      ENDIF
C
```

```
C.....COMPUTE MODEL OUTPUT AT CALCULATED POINTS IF DESIRED
C
      IF (NSTEPS .GT. 0) THEN
         CALL INITL (DDVDT, DV, OUTPUT, T0)
         CALL GRPFRM
         CALL INFO2 (OUTPUT, NSTEPS)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
         NDOF = 2 * NCOL
         NCOLEX = NDOF - 1
         T0 = 0.D00
         H = 1.D-08
         INDEX = 1
         DV (1) = C0
         NLC = 2
         NUC = 2
C
C.....TIME LOOP FOR SIMULTATION AND OUTPUT AT COMPUTED POINTS
C
         WRITE (IOOUT, 1000)
         WRITE (IOOUT, 1100)
         DO 110 I = 1, NSTEPS
            IF (I .GT. 1) THEN
               IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     1            (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
                  H = 1.D-08
                  INDEX = 1
                  DCODT = 0.D00
                  DV (1) = OUTPUT (I,4)
                  WRITE (IOOUT, 1300)
                  IF (DV (1) .LT. 1.0E-10) THEN
                     VEL = VEL2
                     DH = DH1 * VEL2 / VEL1
                     CALL GRPFRM
                     WRITE (IOOUT, 1400)
                  ENDIF
               ELSE
                  DCODT = OUTPUT (I, 5)
               ENDIF
            ELSE
               DCODT = OUTPUT (I, 5)
            ENDIF
            TIME = OUTPUT (I,1)
            CALL DGEAR (NDOF, CPRIME, FCNJ, T0, H, DV, TIME,
     1                  EPSLON, 2, -2, INDEX, IWK, WK, IER)
            CTEMP = DV (NCOLEX)
            CNORM = CTEMP / C0
            OUTCON (I) = DV (NCOLEX)
  110       WRITE (IOOUT, 1200) (OUTPUT (I,J), J = 1, 3), CTEMP, CNORM
C
C.....RESTORE ORIGINAL PARAMETERS
C
         DH = DH1
         VEL = VEL1
C
C.....CHECK THE MASS BALANCE
C
         CALL MASS2 (OUTCON, DV, C0, ZCOL, THETA, VEL, VEL2,
     1               RHO, OUTPUT, NSTEPS, NCOL)
C
      ENDIF
 1000 FORMAT ('1', 42X, 'SIMULATION OUTPUT FOR PLOTTING')
 1100 FORMAT ('0-------------------------------------------------',
     1        '--------------------------------------------------',
     2        '-------------------------------')
```

```
 1200 FORMAT (5E12.5)
 1300 FORMAT (' ', 20X, '***** ', 'ABRUPT INFLUENT CONCENTRATION ',
     1           'CHANGE. ', '***** ', 'DGEAR WAS RESET. ', '*****')
 1400 FORMAT (' ', 23X, '***** ', 'DESORPTION DETECTED. ',
     1           '***** ', 'HYDRODYNAMICS WERE RESET. ', '*****')
      STOP
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                           C
C                                                                           C
C     THIS SUBROUTINE READS THE BASIC INPUT REQUIRED FOR THE SECOND-        C
C     ORDER  SORPTION VERSION OF THE ADVECTIVE-DISPERSIVE EQUATION.         C
C                                                                           C
C                                                                           C
C     VARIABLE MAP:                                                         C
C     ------------                                                          C
C                                                                           C
C     VARIABLE NAME       PRECISION                 DESCRIPTION             C
C     -------------       ---------     ----------------------------------  C
C     IOIN                INTEGER*4     UNIT NUMBER TO READ FROM            C
C     IODUT               INTEGER*4     UNIT NUMBER TO WRITE TO             C
C     NCOL                INTEGER*4     NUMBER OF NODES IN Z DIRECTION      C
C     NOUT                INTEGER*4     NUMBER OF TEMPORAL STEPS TO OUTPUT  C
C     NSTEPS              INTEGER*4     NUMBER OF BED VOLUME STEP TO OUTPUT C
C                                                                           C
C                                                                           C
C     B                   REAL*8        LANGMUIR ENERGY CONSTANT            C
C     CO                  REAL*8        INITIAL CONCENTRATION               C
C     DATA                REAL*4        EXPERIMENTAL INPUT DATA             C
C                                       COLUMN 1 CONTAINS REPORT TIMES      C
C                                       COLUMN 2 GIVES EXPERIMENTAL         C
C                                       CONCENTRATION                       C
C                                       COLUMN 3 GIVES INFLUENT             C
C                                       CONCENTRATION                       C
C                                       COLUMN 4 GIVES DCO/DT               C
C     DH                  REAL*8        HYDRODYNAMIC DISPERSION COEFFICIENT C
C     EPSLON              REAL*8        ALLOWABLE ERROR FOR DGEAR           C
C     FKAPPA              REAL*8        FLUID-PHASE LOSS RATE               C
C     QO                  REAL*8        LANGMUIR CAPACITY CONSTANT          C
C     RHO                 REAL*8        SOLID PHASE PARTICLE DENSITY        C
C     SKAPPA              REAL*8        SOLID-PHASE LOSS RATE               C
C     SOK                 REAL*8        SECOND-ORDER RATE CONSTANT          C
C     THETA               REAL*8        POROSITY                            C
C     THESIM              REAL*8        NUMBER OF BED VOLUMES TO SIMULATE   C
C     VEL                 REAL*8        PORE VELOCITY                       C
C     VEL2                REAL*8        DESORPTION PORE VELOCITY            C
C     ZCOL                REAL*8        LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                           C
C                                                                           C
C     TITLE               CHARACTER*72 TITLE OF SIMULATION RUN              C
C                                                                           C
C                                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE READ
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      CHARACTER*72 TITLE
      DIMENSION DATA (100,4), C (100), T(100)
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
     2               ICON, NCOL, NOUT, NSTEPS
      COMMON /WORDS/ TITLE
```

```
      COMMON /UNITS/ IOIN, IOOUT
      READ (IOIN, 1000) TITLE
      READ (IOIN, 1100) NCOL, NOUT, NSTEPS, THESIM, ICON
      IF (ABS (NOUT) .GT. 0) THEN
         NTEMP = ABS (NOUT)
         READ (IOIN, 1200) (DATA (I,1), I = 1, NTEMP)
         READ (IOIN, 1200) (DATA (I,2), I = 1, NTEMP)
      ENDIF
      READ (IOIN, 1200) VEL, DH, RHO, THETA, ZCOL, EPSLON
      READ (IOIN, 1200) CO, QO, B, SOK, FKAPPA, SKAPPA, VEL2
      READ (IOIN, 1200) (C (I), I = 1, ICON)
      READ (IOIN, 1200) (T (I), I = 1, ICON)
 1000 FORMAT (A72)
 1100 FORMAT (3I5, E12.5, I5)
 1200 FORMAT ((6E12.5))
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                           C
C     THIS SUBROUTINE ECHOS THE INPUT TO DEVICE IOOUT.                      C
C                                                                           C
C     VARIABLE MAP:                                                         C
C     ------------                                                          C
C                                                                           C
C     VARIABLE NAME      PRECISION                 DESCRIPTION              C
C     -------------      ---------     ----------------------------------   C
C     IOIN               INTEGER*4     UNIT NUMBER TO READ FROM             C
C     IOOUT              INTEGER*4     UNIT NUMBER TO WRITE TO              C
C     NCOL               INTEGER*4     NUMBER OF NODES IN Z DIRECTION       C
C                                                                           C
C                                                                           C
C     B                  REAL*8        LANGMUIR ENERGY CONSTANT             C
C     CO                 REAL*8        INITIAL CONCENTRATION                C
C     DATA               REAL*4        EXPERIMENTAL INPUT DATA              C
C                                      COLUMN 1 CONTAINS REPORT TIMES       C
C                                      COLUMN 2 GIVES EXPERIMENTAL          C
C                                      CONCENTRATION                        C
C                                      COLUMN 3 GIVES INFLUENT              C
C                                      CONCENTRATION                        C
C                                      COLUMN 4 GIVES DCO/DT                C
C     DH                 REAL*8        HYDRODYNAMIC DISPERSION COEFFICIENT  C
C     EPSLON             REAL*8        ALLOWABLE ERROR FOR DGEAR            C
C     FKAPPA             REAL*8        FLUID-PHASE LOSS RATE                C
C     PE                 REAL*8        SYSTEM PECLET NUMBER                 C
C     QO                 REAL*8        LANGMUIR CAPACITY CONSTANT           C
C     RHO                REAL*8        SOLID PHASE PARTICLE DENSITY         C
C     SKAPPA             REAL*8        SOLID-PHASE LOSS RATE                C
C     SOK                REAL*8        SECOND-ORDER RATE CONSTANT           C
C     THETA              REAL*8        POROSITY                             C
C     THESIM             REAL*8        NUMBER OF BED VOLUMES TO SIMULATE    C
C     VEL                REAL*8        PORE VELOCITY                        C
C     VEL2               REAL*8        DESORPTION PORE VELOCITY             C
C     ZCOL               REAL*8        LENGTH OF COLUMN (Z DIRECTION)       C
C                                                                           C
C                                                                           C
C     TITLE              CHARACTER*72 TITLE OF SIMULATION RUN               C
C                                                                           C
C                                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE ECHO
      IMPLICIT REAL*8 (A - H,O - Z)
```

```
      REAL*4 DATA
      CHARACTER*72 TITLE
      DIMENSION  DATA (100,4), C (100), T (100)
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C,T,
     2               ICON, NCOL, NOUT, NSTEPS
      COMMON /WORDS/ TITLE
      COMMON /UNITS/ IOIN, IOOUT
      PE = VEL * ZCOL / DH
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TITLE
      WRITE (IOOUT, 1400)
      WRITE (IOOUT, 1300)
      WRITE (IOOUT, 1400)
      WRITE (IOOUT, 1500) CO
      WRITE (IOOUT, 1600) DH
      WRITE (IOOUT, 1700) NCOL
      WRITE (IOOUT, 1800) RHO
      WRITE (IOOUT, 1900) VEL
      WRITE (IOOUT, 1950) VEL2
      WRITE (IOOUT, 2000) THETA
      WRITE (IOOUT, 2100) ZCOL
      WRITE (IOOUT, 2200) PE
      WRITE (IOOUT, 2300) QO
      WRITE (IOOUT, 2400) B
      WRITE (IOOUT, 2500) SOK
      WRITE (IOOUT, 2600) FKAPPA
      WRITE (IOOUT, 2700) SKAPPA
      WRITE (IOOUT, 2800) ICON
      WRITE (IOOUT, 2900)
      WRITE (IOOUT, 3000) (C (I), I = 1, ICON)
      WRITE (IOOUT, 3100)
      WRITE (IOOUT, 3000) (T (I), I = 1, ICON)
 1000 FORMAT ('1', 50X, 'PLUG FLOW DISPERSION REACTOR')
 1100 FORMAT (' ', 47X, 'LANGMUIR SECOND ORDER KINETIC SIMULATION')
 1200 FORMAT (' ', 26X, A72)
 1300 FORMAT (' ', 15X, 'VARIABLE', 16X, 'DESCRIPTION', 30X, '   VALUE',
     1        16X, 'UNITS   ', 16X)
 1400 FORMAT ('0-------------------------------------------------',
     1        '--------------------------------------------------',
     2        '---------------------------------')
 1500 FORMAT ('0', 15X, 'CO    ', 18X, 'INITIAL CONCENTRATION    ', 14X,
     1        E10.4, 16X, 'G/CM**3')
 1600 FORMAT (' ', 15X, 'DH    ', 18X, 'HYDRODYNAMIC DISPERSION  ', 14X,
     1        E10.4, 16X, 'CM**2/HOUR')
 1700 FORMAT (' ', 15X, 'NCOL  ', 18X, 'NUMBER OF COLUMN STEPS   ', 16X,
     1        I5)
 1800 FORMAT (' ', 15X, 'RHO   ', 18X, 'SOLID PHASE DENSITY      ', 14X,
     1        E10.4, 16X, 'G/CM**3')
 1900 FORMAT (' ', 15X, 'VEL   ', 18X, 'PORE VELOCITY            ', 14X,
     1        E10.4, 16X, 'CM/HOUR')
 1950 FORMAT (' ', 15X, 'VEL   ', 18X, 'DESORPTION VELOCITY      ', 14X,
     1        E10.4, 16X, 'CM/HOUR')
 2000 FORMAT (' ', 15X, 'THETA ', 18X, 'VOID VOLUME FRACTION     ', 14X,
     1        E10.4)
 2100 FORMAT (' ', 15X, 'ZCOL  ', 18X, 'LENGTH OF COLUMN         ', 14X,
     1        E10.4, 16X, 'CM')
 2200 FORMAT (' ', 15X, 'PE    ', 18X, 'PECLET NUMBER            ', 14X,
     1        E10.4)
 2300 FORMAT (' ', 15X, 'QO    ', 18X, 'LANGMUIR CAPACITY CONSTANT',14X,
     1        E10.4, 16X, 'G/G')
 2400 FORMAT (' ', 15X, 'B     ', 18X, 'LANGMUIR ENERGY CONSTANT ', 14X,
     1        E10.4, 16X, 'CM**3/G')
 2500 FORMAT (' ', 15X, 'SOK   ', 18X, 'SECOND-ORDER RATE CONSTANT ',
     1       12X, E10.4, 16X, 'CM**3/G-HR')
```

```
 2600 FORMAT (' ', 15X, 'FKAPPA', 18X, 'FLUID PHASE LOSS RATE    ', 14X,
     1        E10.4, 16X, '1/HR')
 2700 FORMAT (' ', 15X, 'SKAPPA', 18X, 'SOLID PHASE LOSS RATE    ', 14X,
     1        E10.4, 16X, '1/HR')
 2800 FORMAT (' ', 15X, 'ICON  ', 18X, 'NUMBER OF INFLUENTS      ', 14X,
     1        I5)
 2900 FORMAT (' ', 39X, 'INFLUENT CONCENTRATIONS')
 3000 FORMAT (' ', 14X, 10E10.3)
 3100 FORMAT (' ', 39X, 'INFLUENT TIMES')
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C                                                                         C
C     THIS SUBROUTINE INITIALIZES THE MODEL VARIABLES FOR THE SIMULATION  C
C     OF THE ADVECTIVE-DISPERSIVE EQUATION.                               C
C                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                  DESCRIPTION           C
C     -------------      ---------    ----------------------------------  C
C     I                  INTEGER*4    WORKING VARIABLE                    C
C     J                  INTEGER*4    WORKING VARIABLE                    C
C     NSTEPS             INTEGER*4    NUMBER OF EQUAL BED VOLUME STEPS    C
C                                     TO BE CALCULATED                    C
C                                                                         C
C                                                                         C
C     B                  REAL*8       LANGMUIR ENERGY CONSTANT            C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA             C
C                                     COLUMN 1 CONTAINS REPORT TIMES      C
C                                     COLUMN 2 GIVES EXPERIMENTAL         C
C                                     CONCENTRATION                       C
C                                     COLUMN 3 GIVES INFLUENT             C
C                                     CONCENTRATION                       C
C                                     COLUMN 4 GIVES DCO/DT               C
C     DDVDT              REAL*8       DERIVATIVE OF DEPENDENT VARIABLES   C
C     DG                 REAL*8       RATIO  OF TOTAL MASS ADSORBABLE     C
C                                     ON SOLID PHASE TO TOTAL MASS IN     C
C                                     FLUID PHASE                         C
C     DT                 REAL*8       DURATION OF EACH TIME STEP          C
C     DTN                REAL*8       REAL EQUIVALENT OF NSTEPS           C
C     DV                 REAL*8       DEPENDENT VARIABLE VECTOR           C
C     EPSMCN             REAL*8       MACHINE EPSILON                     C
C     FKAPPA             REAL*8       FLUID-PHASE LOSS RATE               C
C     OUTPUT             REAL*4       A MATRIX OF SIMULATION STATUS       C
C                                     COL 1 CONTAINS REPORT TIME          C
C                                     COL 2 CONTAINS BED VOLUMES          C
C                                     COL 3 CONTAINS THROUGHPUT           C
C                                     COL 4 GIVES INFLUENT CONCENTRATION  C
C                                     COL 5 GIVES DCO/DT                  C
C     QO                 REAL*8       LANGMUIR CAPACITY CONSTANT          C
C     SKAPPA             REAL*8       SOLID-PHASE LOSS RATE               C
C     SOK                REAL*8       SECOND-ORDER RATE CONSTANT          C
C     TAU                REAL*8       TRAVEL TIME THROUGH COLUMN          C
C     TEMP               REAL*8       TEMPORARY VARIABLE FOR EPSMCN CALC  C
C     TEMPT              REAL*8       TEMPORARY VALUE OF REPORT TIME      C
C     THESIM             REAL*8       NUMBER OF BED VOLUMES TO SIMULATE   C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
C
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C
C
      SUBROUTINE INITL (DDVDT, DV, OUTPUT, TO)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA, OUTPUT
      LOGICAL DONE
      DIMENSION DATA (100,4), DDVDT (101), DV (101), OUTPUT (250,5),
     1          C (100), T (100)
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
     2               ICON, NCOL, NOUT, NSTEPS
C
C.....INITIALIZE DEPENDENT VARIABLE AND DERIVATIVE VECTORS
C
      DO 100 I = 1, 101
         DV (I) = 0.D00
  100 DDVDT (I) = 0.D00
C
C.....COMPUTE SIMULATION OUTPUT POINTS IF DESIRED
C
      IF (NSTEPS .GT. 0) THEN
         TAU = ZCOL / VEL
         DG = 1 + (RHO * (1 - THETA) / THETA) * ((Q0 * B * CO) /
     1        (1 + B * CO)) / CO
         DTN = NSTEPS
         DT = THESIM / DTN
         TEMPT = 0.D00
         DO 110 I = 1, NSTEPS
            TEMPT = TEMPT + DT
            OUTPUT (I,1) = TEMPT
            OUTPUT (I,2) = TEMPT / TAU
  110    OUTPUT (I,3) = TEMPT / (TAU * DG)
      ENDIF
C
C.....CALCULATE THE MACHINE EPSILON FOR USE AS A LOWER ERROR BOUND
C
      EPSMCN = 1.0
  120 EPSMCN = EPSMCN / 2.0D00
      TEMP = 1.0D00 + EPSMCN
      IF (TEMP .GT. 1.0D00) GO TO 120
C
C.....CALCULATE THE INFLUENT CONCENTRATIONS FOR USE IN THE
C.....EXPERIMENTAL COMPARISON MODE BY INTERPOLATING FROM THE
C.....LIST OF KNOWN CONCENTRATIONS.
C
      IF (NOUT .GT. 0) THEN
          DO 130 NCOUNT = 1, NOUT, 1
              TEND = DATA (NCOUNT, 1)
              DONE = .FALSE.
              DO 140 NLOOP = 1, (ICON - 1)
              IF (DONE .EQV. .FALSE.) THEN
                  DNTIME = T (NLOOP)
                  UPTIME = T (NLOOP + 1)
                  DNCON  = C (NLOOP)
                  UPCON  = C (NLOOP + 1)
                  IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND))
     1            THEN
                      DATA (NCOUNT, 3) = DNCON + (UPCON - DNCON)
     1                * (TEND - DNTIME) / (UPTIME - DNTIME)
                      DONE = .TRUE.
                  ENDIF
               ENDIF
  140          CONTINUE
               IF (NCOUNT .GT. 1) THEN
                   DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) -
```

```
     1              DATA (NCOUNT - 1, 3)) / (DATA (NCOUNT, 1) -
     2              DATA (NCOUNT - 1, 1))
               ELSE
                   DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) - CO)
     1              / (DATA ( NCOUNT, 1) - TO)
               ENDIF
 130       CONTINUE
      ENDIF
C
C.....CALCULATE INFLUENT CONCENTRATION FOR USE IN THE
C.....SIMULATION MODE.
C
      IF (NSTEPS .GT. 0) THEN
          DO 150 NCOUNT = 1, NSTEPS, 1
              TEND = OUTPUT(NCOUNT, 1)
              DONE = .FALSE.
              DO 160 NLOOP = 1, (ICON - 1)
              IF (DONE .EQV. .FALSE.) THEN
                  DNTIME = T (NLOOP)
                  UPTIME = T (NLOOP + 1)
                  DNCON  = C (NLOOP)
                  UPCON  = C (NLOOP + 1)
                  IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND))
     1            THEN
                      OUTPUT (NCOUNT, 4) = DNCON + (UPCON - DNCON)
     1                * (TEND - DNTIME) / (UPTIME - DNTIME)
                      DONE = .TRUE.
                  ENDIF
              ENDIF
 160          CONTINUE
              IF (NCOUNT .GT. 1) THEN
                  OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) -
     1            OUTPUT (NCOUNT - 1, 4)) / (OUTPUT (NCOUNT, 1) -
     2            OUTPUT (NCOUNT - 1, 1))
              ELSE
                  OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - CO)
     1            / (OUTPUT ( NCOUNT, 1) - TO)
              ENDIF
 150      CONTINUE
C
C.....COMPUTE BED VOLUMES FOR VARIABLE VELOCITY CASE
C
          OLDVEL = VEL
          TEMPT = 0.D00
          DO 170 I = 1, NSTEPS
              TEMPT = TEMPT + DT
              IF (OUTPUT (I, 4) . GT. 1.0E-10) THEN
                  CURVEL = VEL
              ELSE
                  CURVEL = VEL2
              ENDIF
              AVGVEL = (OLDVEL * (TEMPT - DT) + CURVEL * DT) / TEMPT
              TAU = ZCOL / AVGVEL
              OUTPUT (I, 2) = TEMPT / TAU
              OUTPUT (I, 3) = TEMPT / (TAU * DG)
              OLDVEL = AVGVEL
 170      CONTINUE
      ENDIF
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C                                                                          C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE DATA MATRIX TO            C
C     UNIT IOOUT.                                                          C
```

```
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION               DESCRIPTION            C
C     -------------     ---------   ---------------------------------  C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     NCOUNT            INTEGER*4   LOOP COUNTER                       C
C     NOUT              INTEGER*4   NUMBER OF TEMPORAL STEPS TO OUTPUT C
C                                                                      C
C     DATA              REAL*4      EXPERIMENTAL INPUT DATA            C
C                                   COLUMN 1 CONTAINS REPORT TIMES     C
C                                   COLUMN 2 GIVES EXPERIMENTAL        C
C                                   CONCENTRATION                      C
C                                   COLUMN 3 GIVES INFLUENT            C
C                                   CONCENTRATION                      C
C                                   COLUMN 4 GIVES DCO/DT              C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INFO1 (DATA, NOUT)
      REAL*4 DATA
      DIMENSION DATA (100,4)
      COMMON /UNITS/ IOIN, IOOUT
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200)
      WRITE (IOOUT, 1100)
      DO 100 NCOUNT = 1, NOUT
          WRITE (IOOUT, 1300) ( DATA (NCOUNT, I), I = 1, 4)
 100  CONTINUE
 1000 FORMAT ('1', 42X, 'EXPERIMENTAL COMPARISON MODE INFORMATION')
 1100 FORMAT ('0----------------------------------------------',
     1        '-----------------------------------------------',
     2        '--------------------------------')
 1200 FORMAT (' ', 28X, 'TIME',5X,'CONCENTRATION OUT', 6X,
     1        'CONCENTRATION IN', 16X, 'DCO/DT')
 1300 FORMAT (' ', 20X, E12.5, 10X, E12.5, 10X, E12.5, 10X, E12.5)
      RETURN
      END
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE OUTPUT MATRIX TO      C
C     UNIT IOOUT.                                                      C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION               DESCRIPTION            C
C     -------------     ---------   ---------------------------------  C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     NCOUNT            INTEGER*4   LOOP COUNTER                       C
C     NSTEPS            INTEGER*4   NUMBER OF EQUAL BED VOLUME STEPS   C
C                                   TO BE CALCULATED                   C
C                                                                      C
```

```
C     OUTPUT            REAL*4      A MATRIX OF SIMULATION STATUS      C
C                                   COL 1 CONTAINS REPORT TIME         C
C                                   COL 2 CONTAINS BED VOLUMES         C
C                                   COL 3 CONTAINS THROUGHPUT          C
C                                   COL 4 GIVES INFLUENT CONCENTRATION C
C                                   COL 5 GIVES DCO/DT                 C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INFO2 (OUTPUT, NSTEPS)
      REAL*4 OUTPUT
      DIMENSION OUTPUT (250, 5)
      COMMON /UNITS/ IOIN, IOOUT
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200)
      WRITE (IOOUT, 1100)
      DO 100 NCOUNT = 1, NSTEPS
          WRITE (IOOUT, 1300) (OUTPUT (NCOUNT, I), I = 1, 5)
 100  CONTINUE
 1000 FORMAT ('1', 48X, 'SIMULATION MODE INFORMATION')
 1100 FORMAT ('0------------------------------------------------',
     1        '-------------------------------------------------',
     2        '--------------------------------')
 1200 FORMAT (' ', 12X, 'REPORT TIME', 11X, 'BED VOLUMES', 12X,
     1       'THROUGHPUT', 6X, 'CONCENTRATION IN',
     2       16X, 'DCO/DT')
 1300 FORMAT (' ', 11X, E12.5, 10X, E12.5, 10X, E12.5, 10X,
     1       E12.5, 10X, E12.5)
      RETURN
      END
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE COMPUTES VARIABLE GROUPINGS TO MINIMIZE EFFORT IN C
C     THE SOLUTION OF THE SET OF SIMULTANEOUS NONLINEAR EQUATIONS.     C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION                DESCRIPTION           C
C     -------------     ---------   ---------------------------------- C
C     I                 INTEGER*4   TIME STEP INDEX                    C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION     C
C     NDOF              INTEGER*4   NUMBER OF DEGREES OF FREEDOM       C
C     NOUT              INTEGER*4   NUMBER OF TEMPORAL STEPS TO OUTPUT C
C     NSTEPS            INTEGER*4   NUMBER OF TEMPORAL STEPS TO SIMULATEC
C                                                                      C
C                                                                      C
C     B                 REAL*8      LANGMUIR ENERGY CONSTANT           C
C     DATA              REAL*4      EXPERIMENTAL INPUT DATA            C
C                                   COLUMN 1 CONTAINS REPORT TIMES     C
C                                   COLUMN 2 GIVES EXPERIMENTAL        C
C                                   CONCENTRATION                      C
C                                   COLUMN 3 GIVES INFLUENT            C
C                                   CONCENTRATION                      C
C                                   COLUMN 4 GIVES DCO/DT              C
C     DGN               REAL*8      DIAGONAL VARIABLE GROUPING TERMS   C
C     DGNLOW            REAL*8      LOWER DIAGONAL VARAIBLE GROUPINGS  C
C     DGNUP             REAL*8      UPPER DIAGONAL VARIABLE GROUPINGS  C
C     DH                REAL*8      HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DZ                REAL*8      COLUMN STEP SIZE IN Z DIRECTION    C
C     FKAPPA            REAL*8      FLUID-PHASE LOSS RATE              C
```

```
C      FLSP                 REAL*8       FLUID-PHASE LOSS GROUPING           C
C      FLDSP                REAL*8       FLUID-PHASE LOSS GROUPING           C
C      QO                   REAL*8       LANGMUIR CAPACITY CONSTANT          C
C      REALND               REAL*8       NUMBER OF NODES IN COLUMN           C
C      RHO                  REAL*8       SOLID PHASE PARTICLE DENSITY        C
C      SKAPPA               REAL*8       SOLID-PHASE LOSS RATE               C
C      SLDSP                REAL*8       SOLID-PHASE LOSS GROUPING           C
C      SOK                  REAL*8       SECOND-ORDER RATE CONSTANT          C
C      THETA                REAL*8       POROSITY                            C
C      THESIM               REAL*8       NUMBER OF BED VOLUMES TO SIMULATE   C
C      VEL                  REAL*8       PORE VELOCITY                       C
C      VEL2                 REAL*8       DESORPTION PORE VELOCITY            C
C      ZCOL                 REAL*8       LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                            C
C                                                                            C
C                                                                            C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE GRPFRM
      IMPLICIT REAL*8(A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), C (100), T (100)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, FLSP, FLDSP, SLDSP
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
     2               ICON, NCOL, NOUT, NSTEPS
C
C.....COMPUTE MACROSCOPIC ADVECTIVE-DISPERSIVE EQUATION FLUID PHASE GROUPINS
C
      REALND = NCOL
      DZ = ZCOL / (REALND - 1.D00)
      DGNLOW = DH / (DZ ** 2.D00) + VEL / (2.D00 * DZ)
      DGN = -2.D00 * DH / (DZ ** 2.D00) - FKAPPA
      DGNUP = DH / (DZ ** 2.D00) - VEL / (2.D00 *DZ)
      FLSP = (RHO * (1 - THETA) / THETA) * SOK
      FLDSP = (RHO * (1 - THETA) / THETA) * (SOK / B)
      SLDSP = SOK / B
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                            C
C                                                                            C
C      THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE          C
C      DEPENDENT VARIABLE AS A FUNCTION OF TIME. THE EQUATIONS ARE           C
C      SOLVED USING THE VARIABLE GROUPINGS PREVIOUSLY DERIVED IN ORDER       C
C      TO SAVE COMPUTATIONAL EFFORT.                                         C
C                                                                            C
C      ALL DEPENDENT VARIABLE ARE STORED IN A SINGLE ARRAY DV, AND ALL       C
C      TEMPORAL DERIVATIVES OF THE DEPENDENT VARIABLE ARE STORED IN A        C
C      SINGLE ARRAY TERMED DDVDT. THE INDEX NUMBERING OF THESE ARRAYS        C
C      MAY BE DESCRIBED AS FOLLOWS:                                          C
C           A) THE FIRST ENTRY (1) IS THE INLET NODE OF THE COLUMN.          C
C           B) THE SECOND ENTRY (2) IS LEFT BLANK.                           C
C           C) THE SUCCEEDING REFERENCES TO THE FLUID PHASE ARE AT THE       C
C              ODD-NUMBERED INDICES.                                         C
C           D) THE SOLID PHASE REFERENCES ARE AT THE EVEN-NUMBERED INDEX     C
C              FOLLOWING THE ASSOCIATED FLUID REFERENCES.                    C
C           E) THE TOTAL DEGREES OF FREEDON ARE EQUAL TO 2 * NCOL.           C
C           F) THE NEXT-TO-LAST ENTRY IS THE FLUID PHASE EXIT CON'N.         C
C           G) THE LAST ENTRY IS THE SOLID PHASE CONCENTRATION AT THE EXIT.C
C                                                                            C
C                                                                            C
```

```
C     VARIABLE MAP:                                                     C
C     ------------                                                      C
C                                                                       C
C     VARIABLE NAME     PRECISION               DESCRIPTION             C
C     -------------     ---------   ----------------------------------  C
C     I                 INTEGER*4   TEMPORARY VARAIBLE                  C
C     J                 INTEGER*4   TEMPORARY VARAIBLE                  C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION      C
C     NDOF              INTEGER*4   NUMBER OF DEGREES OF FREEDOM        C
C     NEND              INTEGER*4   VARIABLE INDEX FOR LAST COLUMN NODE C
C     NSTART            INTEGER*4   LOOP STARTING POSITION              C
C     NSTEP             INTEGER*4   STEP SIZE FOR LOOP                  C
C     NSTOP             INTEGER*4   LOOP STOPPING POSITION              C
C                                                                       C
C                                                                       C
C     B                 REAL*8      LANGMUIR ENERGY CONSTANT            C
C     DATA              REAL*4      EXPERIMENTAL INPUT DATA             C
C                                   COLUMN 1 CONTAINS REPORT TIMES      C
C                                   COLUMN 2 GIVES EXPERIMENTAL         C
C                                   CONCENTRATION                       C
C                                   COLUMN 3 GIVES INFLUENT             C
C                                   CONCENTRATION                       C
C                                   COLUMN 4 GIVES DCO/DT               C
C     DDVDT             REAL*8      DERIVATIVE OF DEPENDENT VARIABLES   C
C     DGN               REAL*8      DIAGONAL VARIABLE GROUPING TERMS    C
C     DGNLOW            REAL*8      LOWER DIAGONAL VARAIBLE GROUPINGS   C
C     DGNUP             REAL*8      UPPER DIAGONAL VARIABLE GROUPINGS   C
C     DV                REAL*8      DEPENDENT VARIABLE VECTOR           C
C     FKAPPA            REAL*8      FLUID-PHASE LOSS RATE               C
C     FLSP              REAL*8      FLUID-PHASE LOSS GROUPING           C
C     FLDSP             REAL*8      FLUID-PHASE LOSS GROUPING           C
C     QO                REAL*8      LANGMUIR CAPACITY CONSTANT          C
C     SKAPPA            REAL*8      SOLID-PHASE LOSS RATE               C
C     SLDSP             REAL*8      SOLID-PHASE LOSS GROUPING           C
C     SOK               REAL*8      SECOND-ORDER RATE CONSTANT          C
C     THESIM            REAL*8      NUMBER OF BED VOLUMES TO SIMULATE   C
C                                                                       C
C                                                                       C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE CPRIME (NVRB, TIME, DV, DDVDT)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DV (101), DDVDT (101), DATA (100,4), C (100), T (100)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, FLSP, FLDSP, SLDSP
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
     2               ICON, NCOL, NOUT, NSTEPS
      COMMON /FLUID/ DCODT
C
C.....FORM FLUID PHASE DERIVATIVES
C
      NDOF = NCOL * 2
C
C.....FIRST THE INLET CONDITION
C
      DDVDT (1) = DCODT
C
C.....SECOND THE INTERIOR FLUID PHASE DERIVATIVES
C
      NEND = NDOF - 3
      DO 100 I = 3, NEND, 2
  100 DDVDT (I) = DGNLOW * DV (I-2) + DGN * DV (I) + DGNUP * DV (I+2)
```

```
     1            - FLSP * (QO - DV (I+1)) * DV (I) + FLDSP * DV (I+1)
C
C.....FORM THE OUTLET BOUNDARY CONDITION FOR THE FLUID PHASE
C
      I = NDOF - 1
      DDVDT (I) = (DGNLOW + DGNUP) * DV (I - 2) + DGN * DV (I)
     1            - FLSP * (QO - DV (I+1)) * DV (I) + FLDSP * DV (I+1)
C
C.....FORM THE SOLID PHASE EQUATIONS
C
      DO 110 I = 4, NDOF, 2
  110 DDVDT (I) = SOK * (QO - DV (I)) * DV (I-1) - SLDSP * DV (I)
     1            - SKAPPA * DV (I)
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C                                                                          C
C     THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE         C
C     DDVDT ARRAY AS A FUNCTION OF THE DEPENDENT VARIABLE (DV).            C
C     SINCE THE FINITE DIFFERENCE OPTION IS USED IN THE CALL TO DGEAR      C
C     (I.E. MITER= -2) THIS ROUTINE IS A BLANK.                            C
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE FCNJ (N, X, Y, PD)
      REAL*8 Y(N), PD(N,N), X
      RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C     THIS SUBROUTINE USES THE EXPERIMENTAL DATA INPUT AND THE MODEL       C
C     SIMULATION OUTPUT TO COMPUTE THE MODEL RESIDUAL, RESIDUAL SQUARED,   C
C     SUM OF RESIDUALS, AND SUM OF RESIDUALS SQUARED. THESE COMPUTATIONS   C
C     ARE OUTPUT TO UNIT IOOUT.                                            C
C                                                                          C
C                                                                          C
C     VARIABLE MAP:                                                        C
C     ------------                                                         C
C                                                                          C
C     VARIABLE NAME      PRECISION                 DESCRIPTION             C
C     -------------      ---------    ----------------------------------   C
C     I                  INTEGER*4    TIME STEP INDEX                      C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM             C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO              C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION       C
C     NDOF               INTEGER*4    NUMBER OF DEGREES OF FREEDOM         C
C     NDV                INTEGER*4    INDEX OF OUTLET CONCENTRATION        C
C     NOUT               INTEGER*4    NUMBER OF POINTS SIMULATED           C
C     NSTEPS             INTEGER*4    NUMBER OF STEPS TO SIMULATE          C
C                                                                          C
C                                                                          C
C     B                  REAL*8       LANGMUIR ENERGY CONSTANT             C
C     CNORM              REAL*8       NORMALIZED OUTLET CONCENTRATION      C
C     CO                 REAL*8       INITIAL CONCENTRATION                C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA              C
C                                     COLUMN 1 CONTAINS REPORT TIMES       C
C                                     COLUMN 2 GIVES EXPERIMENTAL          C
C                                     CONCENTRATION                        C
C                                     COLUMN 3 GIVES INFLUENT              C
```

```
C                                  CONCENTRATION                       C
C                                  COLUMN 4 GIVES DCO/DT               C
C     DV                REAL*8     DEPENDENT VARIABLE VECTOR           C
C     FKAPPA            REAL*8     FLUID-PHASE LOSS RATE               C
C     QO                REAL*8     LANGMUIR CAPACITY CONSTANT          C
C     RDSQSM            REAL*8     SUM OF RESIDUALS SQUARED            C
C     RSD               REAL*8     RESIDUAL OF POINT AT OUTLET         C
C     RSDSQ             REAL*8     RESIDUAL SQUARED OF POINT AT OUTLET C
C     RSDSUM            REAL*8     SUM OF RESIDUALS                    C
C     SKAPPA            REAL*8     SOLID-PHASE LOSS RATE               C
C     SOK               REAL*8     SECOND-ORDER RATE CONSTANT          C
C     THESIM            REAL*8     NUMBER OF BED VOLUMES TO SIMULATE   C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE REPORT (I, DV, RSDSUM, RDSQSM, RSD)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), DV(101), C (100), T (100)
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1              EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
     2              ICON, NCOL, NOUT, NSTEPS
      COMMON /UNITS/ IOIN, IOOUT
      NDV = (NCOL * 2) - 1
      IF (I .EQ. 1) THEN
         WRITE (IOOUT, 1000)
         WRITE (IOOUT, 1100)
         WRITE (IOOUT, 1200)
         WRITE (IOOUT, 1100)
         RSDSUM = 0.DOO
         RDSQSM = 0.DOO
      ENDIF
      CNORM = DV (NDV) / CO
      RSD = DATA (I,2) - CNORM
      RSDSQ = RSD * RSD
      RSDSUM = RSDSUM + RSD
      RDSQSM = RDSQSM + RSDSQ
      WRITE (IOOUT, 1300) DATA (I,1), DATA (I,2), CNORM, RSD, RSDSQ
      IF (I .GE. NOUT) THEN
         WRITE (IOOUT, 1100)
         WRITE (IOOUT, 1400) RSDSUM, RDSQSM
      ENDIF
 1000 FORMAT ('1',58X,'RESIDUAL REPORT')
 1100 FORMAT ('0-----------------------------------------------',
     1        '------------------------------------------------',
     2        '-------------------------------')
 1200 FORMAT ('0',19X,'TIME',10X,'C/Co (INPUT)',10X,'C/Co (PREDICTED)',
     1        14X,'RESIDUAL',10X,'RESIDUAL SQUARED')
 1300 FORMAT (' ',11X,E12.5,10X,E12.5,14X,E12.5,10X,E12.5,14X,E12.5)
 1400 FORMAT (' ',81X,E12.5,14X,E12.5)
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE          C
C     COMPLETION OF SIMULATION FOR EXPERIMENTAL COMPARISON.            C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
```

```
C                                                                          C
C     VARIABLE NAME       PRECISION                DESCRIPTION             C
C     -------------       ---------     -------------------------------    C
C     I                   INTEGER*4     LOOP INDEX                         C
C     IFLAG1              INTEGER*4     FLAG FOR ABRUPT CONCENTRATION      C
C                                       CHANGE                             C
C     IFLAG2              INTEGER*4     FLAG FOR DESORPTION                C
C     IOIN                INTEGER*4     UNIT NUMBER TO READ FROM           C
C     IOOUT               INTEGER*4     UNIT NUMBER TO WRITE TO            C
C     NCOL                INTEGER*4     NUMBER OF NODES IN Z DIRECTION     C
C     NOUT                INTEGER*4     NUMBER OF TEMPORAL STEPS TO OUTPUT C
C                                                                          C
C                                                                          C
C                                                                          C
C     BAL                 REAL*8        RATIO OF MASS IN TO MASS ACCOUNTED C
C     CO                  REAL*8        INITIAL CONCENTRATION              C
C     DATA                REAL*4        EXPERIMENTAL INPUT DATA            C
C                                       COLUMN 1 CONTAINS REPORT TIMES     C
C                                       COLUMN 2 GIVES EXPERIMENTAL        C
C                                       CONCENTRATION                      C
C                                       COLUMN 3 GIVES INFLUENT            C
C                                       CONCENTRATION                      C
C                                       COLUMN 4 GIVES DCO/DT              C
C     DV                  REAL*8        DEPENDENT VARIABLE VECTOR          C
C     OUTCON              REAL*8        EXIT CONCENTRATION VECTOR          C
C     RCOL                REAL*8        REAL EQUIVALENT OF NCOL            C
C     THETA               REAL*8        POROSITY                           C
C     TOTDRY              REAL*8        TOTAL MASS LEFT ON SOLID PHASE     C
C     TOTINF              REAL*8        TOTAL MASS INJECTED INTO COLUMN    C
C     TOTOUT              REAL*8        TOTAL MASS TO EXIT COLUMN          C
C     TOTWET              REAL*8        TOTAL MASS LEFT IN LIQUID PHASE    C
C     VEL                 REAL*8        PORE VELOCITY                      C
C     VEL2                REAL*8        DESORPTION PORE VELOCITY           C
C     ZCOL                REAL*8        LENGTH OF COLUMN (Z DIRECTION)     C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE MASS1 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
     1                  RHO, DATA, NOUT, NCOL)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100, 4), DV (101), OUTCON (250)
      COMMON /UNITS/ IOIN, IOOUT
C
      IFLAG1 = 0
      IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
      VEL1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
C
C     CALCULATE  CONTAMINANT INJECTED DURING FIRST
C     REPORT PERIOD.
C
          TOTINF = CO * DATA (1, 1) * VEL * THETA +
     1             0.5 * (DATA (1, 3) - CO) * DATA (1, 1) *
     2             VEL * THETA
```

```
C
C     CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
C
      DO 100 I = 2, (NOUT - 1)
         IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
     1        (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
              IFLAG1 = 1
              TOTINF = TOTINF +
     1                 DATA (I, 3) * (DATA (I, 1) - DATA (I - 1, 1)) *
     2                 VEL * THETA
         ELSE
              TOTINF = TOTINF +
     1                 DATA (I - 1, 3) *
     2                 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA +
     3                 0.5 * (DATA (I, 3) - DATA (I - 1, 3)) *
     4                 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
         ENDIF
 100  CONTINUE
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
C     CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
C     THE FIRST REPORT PERIOD
C
      TOTOUT = (0.75 * 0.DO + 0.25 * OUTCON (1)) *
     1         DATA (1, 1) / 2.DO * VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C     PERIOD
C
      TOTOUT = TOTOUT +
     1         ((0.75 * OUTCON (1) + 0.25 * 0.DO) *
     2         (DATA (1, 1) - 0.DO) / 2.DO +
     3         (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
     4         (DATA (2, 1) - DATA (1, 1)) / 2.DO) *
     5         VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
C     PERIOD
C
      DO 110 I = 2, (NOUT - 1)
C
C     RESET VELOCITY IF DESORPTION BEGINS
C
         IF (DATA (I, 3) .LT. 1.0E-10) THEN
            VEL = VEL2
            IFLAG2 = 1
         ENDIF
C
         TOTOUT = TOTOUT +
     1            ((0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
     2            (DATA (I, 1) - DATA (I - 1, 1)) / 2.DO +
     3            (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
     4            (DATA (I + 1, 1) - DATA (I, 1)) / 2.DO) *
     5            VEL * THETA
 110  CONTINUE
C
C     CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C     REPORT PERIOD
C
      TOTOUT = TOTOUT +
     1         (0.75 * OUTCON (NOUT) +
     2         0.25 * OUTCON (NOUT - 1)) *
     3         (DATA (NOUT, 1) - DATA (NOUT - 1, 1)) / 2.DO *
     4         VEL * THETA
C
```

```
C     RESTORE VELOCITY
C
      VEL = VEL1
C
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
C
      RCOL = REAL (NCOL)
      TOTWET = (0.75 * DV (1) + 0.25 * DV (3)) *
     1          ZCOL / (RCOL - 1.DO) / 2.DO * THETA
C
C     CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE
C
      DO 120 I = 3, (2 * NCOL - 3), 2
         TOTWET = TOTWET +
     1             (0.75 * DV (I) + 0.125 * DV (I - 2) +
     2             0.125 * DV (I + 2)) *
     3             ZCOL / (RCOL - 1.DO) * THETA
 120  CONTINUE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
C
      TOTWET = TOTWET + (0.25 * DV (2 * NCOL - 3) +
     1          0.75 * DV (2 * NCOL - 1)) *
     2          ZCOL / (RCOL - 1.DO) / 2.DO * THETA
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE
C
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE
C
      TOTDRY = 0.DO * 0.75 + DV (4) * 0.25
C
C     CALCULATE CONAMINANT ON TYPICAL NODES
C
      DO 130 I = 4, (2 * NCOL - 2), 2
         TOTDRY = TOTDRY +
     1             (0.75 * DV (I) + 0.125 * DV (I - 2)
     2             + 0.125 * DV (I + 2)) *
     3             ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)
 130  CONTINUE
C
C     CALCULATE CONTAMINANT NEAR EXIT
C
      TOTDRY = TOTDRY +
     1          (0.25 * DV (2 * NCOL - 2) + 0.75 * DV (2 * NCOL))
     2          * ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO -THETA)
C
C
C.....COMPUTE MASS BALANCE
C
C
      BAL = TOTINF / ( TOTOUT + TOTWET + TOTDRY )
C
C
C.....OUTPUT THE MASS BALANCE FINDINGS
C
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TOTINF
      WRITE (IOOUT, 1300) TOTOUT
      WRITE (IOOUT, 1400) TOTWET
      WRITE (IOOUT, 1500) TOTDRY
```

```
      WRITE (IOOUT, 1600) BAL
      IF (IFLAG1 .EQ. 1) THEN
         WRITE (IOOUT, 1700)
      ENDIF
      IF (IFLAG2 .EQ. 1) THEN
         WRITE (IOOUT, 1800)
      ENDIF
C
C
 1000 FORMAT ('1', 35X, 'EXPERIMENTAL MASS BALANCE REPORT')
 1100 FORMAT ('0-------------------------------------------------',
     1       '--------------------------------------------------',
     2       '--------------------------------')
 1200 FORMAT (' ', 20X, 'TOTAL MASS INJECTED = ',
     1      E10.4, ' GRAMS / SQ. CM ')
 1300 FORMAT (' ', 20X, 'TOTAL MASS RELEASED = ',
     1      E10.4, ' GRAMS / SQ. CM ')
 1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ',
     1      E10.4, ' GRAMS / SQ. CM ')
 1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ',
     1      E10.4, ' GRAMS / SQ. CM ')
 1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = ',
     1      E10.4)
 1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
     1       ' IN INFLUENT CONCENTRATION.')
 1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
     1       ' DURING DESORPITON.')
C
C
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE          C
C     COMPLETION OF SIMULATION IN SIMULATION MODE.                     C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION                DESCRIPTION           C
C     -------------     ---------   ----------------------------------  C
C     I                 INTEGER*4   LOOP INDEX                         C
C     IFLAG1            INTEGER*4   FLAG FOR ABRUPT CONCENTRATION      C
C                                   CHANGE                             C
C     IFLAG2            INTEGER*4   FLAG FOR DESORPTION                C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION     C
C     NSTEPS            INTEGER*4   NUMBER OF TEMPORAL STEPS           C
C                                                                      C
C                                                                      C
C                                                                      C
C     BAL               REAL*8      RATIO OF MASS IN TO MASS ACCOUNTED C
C     CO                REAL*8      INITIAL CONCENTRATION              C
C     DV                REAL*8      DEPENDENT VARIABLE VECTOR          C
C     OUTCON            REAL*8      EXIT CONCENTRATION VECTOR          C
C     OUTPUT            REAL*4      SIMULATION STATUS MATRIX           C
C                                   COL 1 GIVES REPORT TIMES           C
C                                   COL 2 GIVES BED VOLUMES            C
C                                   COL 3 GIVES THROUGHPUT             C
C                                   COL 4 GIVES INFLUENT CONCENTRATION C
C                                   COL 5 GIVES DCO/DT                 C
C     RCOL              REAL*8      REAL EQUIVALENT OF NCOL            C
```

```
C     THETA              REAL*8        POROSITY                            C
C     TOTDRY             REAL*8        TOTAL MASS LEFT ON SOLID PHASE      C
C     TOTINF             REAL*8        TOTAL MASS INJECTED INTO COLUMN     C
C     TOTOUT             REAL*8        TOTAL MASS TO EXIT COLUMN           C
C     TOTWET             REAL*8        TOTAL MASS LEFT IN LIQUID PHASE     C
C     VEL                REAL*8        PORE VELOCITY                       C
C     VEL2               REAL*8        DESORPTION PORE VELOCITY            C
C     ZCOL               REAL*8        LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE MASS2 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
     1            RHO, OUTPUT, NSTEPS , NCOL)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 OUTPUT
      DIMENSION OUTPUT (250, 5), DV (101), OUTCON (250)
      COMMON /UNITS/ IOIN, IOOUT
C
      IFLAG1 = 0
      IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
      VEL1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
C
C     CALCULATE  CONTAMINANT INJECTED DURING FIRST
C     REPORT PERIOD.
C
          TOTINF = CO * OUTPUT (1, 1) * VEL * THETA +
     1             0.5 * (OUTPUT (1, 4) - CO) * OUTPUT (1, 1) *
     2             VEL * THETA
C
C     CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
C
      DO 100 I = 2, (NSTEPS - 1)
         IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     1       (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
             IFLAG1 = 1
             TOTINF = TOTINF +
     1                OUTPUT (I, 4) *
     2                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     3                VEL * THETA
         ELSE
             TOTINF = TOTINF +
     1                OUTPUT (I - 1, 4) *
     2                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     3                VEL * THETA +
     4                0.5 * (OUTPUT (I, 4) - OUTPUT (I - 1, 4)) *
     5                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     6                VEL * THETA
         ENDIF
 100  CONTINUE
C
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
C     CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
```

```
C       THE FIRST REPORT PERIOD
C
        TOTOUT = (0.75 * 0.DO + 0.25 * OUTCON (1)) *
     1          OUTPUT (1, 1) / 2.DO * VEL * THETA
C
C       CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C       PERIOD
C
        TOTOUT = TOTOUT +
     1          ((0.75 * OUTCON (1) + 0.25 * 0.DO) *
     2          (OUTPUT (1, 1) - 0.DO) / 2.DO +
     3          (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
     4          (OUTPUT (2, 1) - OUTPUT (1, 1)) / 2.DO) *
     5          VEL * THETA
C
C       CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
C       PERIOD
C
        DO 110 I = 2, (NSTEPS - 1)
C
C       RESET VELOCITY IF DESORPTION BEGINS
C
           IF (OUTPUT (I, 4) .LT. 1.0E-10) THEN
              VEL = VEL2
              IFLAG2 = 1
           ENDIF
           TOTOUT = TOTOUT +
     1             ((0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
     2             (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) / 2.DO +
     3             (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
     4             (OUTPUT (I + 1, 1) - OUTPUT (I, 1)) / 2.DO) *
     5             VEL * THETA
  110   CONTINUE
C
C       CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C       REPORT PERIOD
C
        TOTOUT = TOTOUT +
     1          (0.75 * OUTCON (NSTEPS) +
     2          0.25 * OUTCON (NSTEPS - 1)) *
     3          (OUTPUT (NSTEPS, 1) - OUTPUT (NSTEPS - 1, 1)) / 2.DO *
     4          VEL * THETA
C
C       RESTORE VELOCITY
C
        VEL = VEL1
C
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C       CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
C
        RCOL = REAL (NCOL)
        TOTWET = (0.75 * DV (1) + 0.25 * DV (3)) *
     1          ZCOL / (RCOL - 1.DO) / 2.DO * THETA
C
C       CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE
C
        DO 120 I = 3, (2 * NCOL - 3), 2
           TOTWET = TOTWET +
     1             (0.75 * DV (I) + 0.125 * DV (I - 2) +
     2             0.125 * DV (I + 2)) *
     3             ZCOL / (RCOL - 1.DO) * THETA
  120   CONTINUE
C
C       CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
```

```
C
      TOTWET = TOTWET + (0.25 * DV (2 * NCOL - 3) +
     1          0.75 * DV (2 * NCOL - 1)) *
     2          ZCOL / (RCOL - 1.DO) / 2.DO * THETA
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE
C
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE
C
      TOTDRY = 0.DO * 0.75 + DV (4) * 0.25
C
C     CALCULATE CONAMINANT ON TYPICAL NODES
C
      DO 130 I = 4, (2 * NCOL - 2), 2
         TOTDRY = TOTDRY +
     1             (0.75 * DV (I) + 0.125 * DV (I - 2)
     2             + 0.125 * DV (I + 2)) *
     3             ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)
 130  CONTINUE
C
C     CALCULATE CONTAMINANT NEAR EXIT
C
      TOTDRY = TOTDRY +
     1          (0.25 * DV (2 * NCOL - 2) + 0.75 * DV (2 * NCOL))
     2          * ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO -THETA)
C
C
C
C.....COMPUTE MASS BALANCE
C
C
      BAL = TOTINF / ( TOTOUT + TOTWET + TOTDRY )
C
C
C.....OUTPUT THE MASS BALANCE FINDINGS
C
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TOTINF
      WRITE (IOOUT, 1300) TOTOUT
      WRITE (IOOUT, 1400) TOTWET
      WRITE (IOOUT, 1500) TOTDRY
      WRITE (IOOUT, 1600) BAL
      IF (IFLAG1 .EQ. 1) THEN
         WRITE (IOOUT, 1700)
      ENDIF
      IF (IFLAG2 .EQ. 1) THEN
         WRITE (IOOUT, 1800)
      ENDIF
C
C
 1000 FORMAT ('1', 35X, 'SIMULATION MASS BALANCE REPORT ')
 1100 FORMAT ('0-------------------------------------------------',
     1        '-------------------------------------------------',
     2        '--------------------------------')
 1200 FORMAT (' ', 20X, 'TOTAL MASS INJECTED = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1300 FORMAT (' ', 20X, 'TOTAL MASS RELEASED = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = '
```

```
     1          E10.4)
 1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
     1          ' IN INFLUENT CONCENTRATION.')
 1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
     1          ' DURING DESORPITON.')
C
C
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                           C
C                        SORPTION MODEL :                                   C
C                  FIRST ORDER EQUILIBRIUM WITH DISPERSION                  C
C                       MAINFRAME VERSION                                   C
C                                                                           C
C                                                                           C
C     THIS PROGRAM USES THE DGEAR METHOD TO SOLVE THE ADVECTIVE             C
C     DISPERSIVE EQUATION FOR THE CASE OF FIRST ORDER EQUILIBRIUM WITH      C
C     DISPERSION.                                                           C
C                                                                           C
C                                                                           C
C     THE FIRST ORDER AND PARALLEL EQUILIBRIUM SORPTION MODEL (FOPESM)      C
C     ASSUMES THAT INTERPHASE MASS TRANSFER OCCURS BETWEEN THE FLUID        C
C     PHASE AND TWO DIFFERENT TYPES OF SOLID PHASE SITES:  FAST SITES       C
C     AND SLOW SITES.  THE MASS TRANSFER TO THE FAST SITES OCCURS           C
C     RAPIDLY AND CAN BE ASSUMED TO BE AT EQUILIBRIUM.  MASS TRANSFER       C
C     TO THE SLOW SITES IS TESTUDINATE  AND REQUIRES BOTH A RATE AND        C
C     EQUILIBRIUM EXPRESSION TO COMPLETELY DESCRIBE THE INTERPHASE          C
C     MASS TRANSFER.                                                        C
C                                                                           C
C                                                                           C
C     THE EQUATIONS ARE SPATIALLY APPROXIMATED USING THE FINITE             C
C     DIFFERENCE METHOD RESULTING IN A SET OF DC/DT AND DQ/DT EQUATIONS.    C
C     THESE EQUATIONS ARE THEN SOLVED SIMULTANEOUSLY USING DGEAR---         C
C     AN IMSL SUBROUTINE.                                                   C
C                                                                           C
C     IN ITS PRESENT FORM THE PROGRAM IS LIMITED TO 100 NODES.              C
C                                                                           C
C                                                                           C
C     VARIABLE MAP:                                                         C
C     ------------                                                          C
C                                                                           C
C     VARIABLE NAME      PRECISION                DESCRIPTION               C
C     -------------      ---------    ----------------------------------   C
C     I                  INTEGER*4    TIME STEP INDEX                       C
C     IDUMMY             INTEGER*4    DUMMY WORKING ARRAY FOR DGEAR         C
C     IER                INTEGER*4    ERROR CODE RETURN INDEX FOR DGEAR     C
C     INDEX              INTEGER*4    CALL INDICATOR INDEX FOR GEAR         C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM              C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO               C
C     IWK                INTEGER*4    WORKING ARRAY FOR DGEAR               C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION        C
C     NCOLEX             INTEGER*4    COLUMN EXIT NODE                      C
C     NDOF               INTEGER*4    NUMBER OF DEGREES OF FREEDOM          C
C     NLC                INTEGER*4    NUMBER OF LOWER CODIAGONALS-DGEAR     C
C     NOUT               INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT    C
C     NUC                INTEGER*4    NUMBER OF UPPER CODIAGONALS-DGEAR     C
C     NSTEPS             INTEGER*4    NUMBER OF BED VOLUME STEP TO OUTPUT  C
C                                                                           C
C                                                                           C
C     ALPHA              REAL*8       SLOW SORPTION PROPORTIONALITY         C
C     CO                 REAL*8       INITIAL CONCENTRATION                 C
C     CNORM              REAL*8       NORMALIZED EXIT CONCENTRATION         C
C     CPRIME             REAL*8       EXTERNAL NAME OF DC/DT SUBROUTINE     C
C     CTEMP              REAL*8       EXIT CONCENTRATION                    C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA               C
C                                     COLUMN 1 CONTAINS REPORT TIMES        C
C                                     COLUMN 2 GIVES EXPERIMENTAL           C
C                                     CONCENTRATION                         C
C                                     COLUMN 3 GIVES INFLUENT               C
```

```
C                                  CONCENTRATION                       C
C                                  COLUMN 4 GIVES DCO/DT               C
C     DDVDT            REAL*8      DERIVATIVE OF DEPENDENT VARIABLES   C
C     DGN              REAL*8      DIAGONAL VARIABLE GROUPING TERMS    C
C     DGNLOW           REAL*8      LOWER DIAGONAL VARAIBLE GROUPINGS   C
C     DGNUP            REAL*8      UPPER DIAGONAL VARIABLE GROUPINGS   C
C     DH               REAL*8      HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DH1              REAL*8      SORPTION PHASE HYDRODYNAMIC         C
C                                  DISPERSION COEFFICIENT              C
C     DUMMY            REAL*8      DUMMY ROUTINE REQUIRED BY DGEAR     C
C     DV               REAL*8      DEPENDENT VARIABLE VECTOR           C
C     EPSLON           REAL*8      ALLOWABLE ERROR FOR DGEAR           C
C     FCNJ             REAL*8      EXTERNAL NAME OF JACOBIAN ROUTINE   C
C     FKAPPA           REAL*8      FLUID PHASE LOSS RATE               C
C     FKF              REAL*8      FAST FREUNDLICH ISOTHERM COEFFICIENTC
C     FKS              REAL*8      SLOW FREUNDLICH ISOTHERM COEFFICIENTC
C     FKT              REAL*8      THROUGHPUT FREUNDLICH COEFFICIENT   C
C     FNF              REAL*8      FAST FREUNDLICH ISOTHERM EXPONENT   C
C     FNS              REAL*8      SLOW FREUNDLICH ISOTHERM EXPONENT   C
C     FNT              REAL*8      THROUGHPUT FREUNDLICH EXPONENT      C
C     FSSDLS           REAL*8      FIRST ORDER DECAY LOSS ON SOLID     C
C     FSSDLS           REAL*8      FIRST ORDER DECAY LOSS ON SOLID     C
C     H                REAL*8      STEP SIZE FOR DGEAR                 C
C     OUTPUT           REAL*8      A MATRIX OF SIMULATION STATUS       C
C                                  COL 1 CONTAINS REPORT TIME          C
C                                  COL 2 CONTAINS BED VOLUMES          C
C                                  COL 3 CONTAINS THROUGHPUT           C
C                                  COL 4 GIVES INFLUENT CONCENTRATION  C
C                                  COL 5 GIVES DCO/DT                  C
C     RDSQSM           REAL*8      SUM OF RESIDUALS SQUARED            C
C     RFGRP            REAL*8      RETARDATION VARIABLE GROUPING       C
C     RHO              REAL*8      SOLID PHASE PARTICLE DENSITY        C
C     RSDSUM           REAL*8      SUM OF RESIDUALS                    C
C     SDUMMY           REAL*8      DUMMY ROUTINE FOR DGEAR             C
C     SKAPPA           REAL*8      SOLID PHASE LOSS RATE               C
C     SLSPAC           REAL*8      SLOW SOLID PHASE ACTUAL             C
C                                  CONCENTRATION GROUPING              C
C     SLSPEQ           REAL*8      SLOW SOLID PHASE EQULIB GROUPING    C
C     THETA            REAL*8      POROSITY                            C
C     TIME             REAL*8      TIME OF OUTPUT SIMULATION POINT     C
C     TMESIM           REAL*8      NUMBER OF BED VOLUMES TO SIMULATE   C
C     TO               REAL*8      INITIAL TIME FOR DGEAR              C
C     VEL              REAL*8      PORE VELOCITY                       C
C     VEL1             REAL*8      SORPTION PORE VELOCITY              C
C     VEL2             REAL*8      DESORPTION PORE VELOCITY            C
C     WK               REAL*8      WORKING ARRY NEEDED BY DGEAR        C
C     ZCOL             REAL*8      LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                      C
C                                                                      C
C     TITLE            CHARACTER*72 TITLE OF SIMULATION RUN            C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 SDUMMY, DATA, OUTPUT
      CHARACTER*72 TITLE
C
C
C
C
      DIMENSION F(100), X(2)
```

```
      DIMENSION IWK (101)
      DIMENSION DATA (100,4), DV (101), DDVDT (101), OUTPUT (250,5),
     1          WK (1919), C (100), T (100), OUTCON (250)
      COMMON /DBAND/ NLC, NUC
      COMMON /GEAR/ DUMMY(48), SDUMMY(4), IDUMMY(38)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, RFGRP, FSSDLS, SLSPEQ,
     1                SLSPAC
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FKF, FNF, FKS, FNS, ALPHA,
     2               FKAPPA, SKAPPA, FKT, FNT, C, T,
     3               ICON, NCOL, NOUT, NSTEPS
      COMMON /WORDS/ TITLE
      COMMON /FLUID/ DCODT
      COMMON /UNITS/ IOIN, IOOUT
      EXTERNAL CPRIME, FCNJ
      IOIN = 1
      IOOUT = 3
      CALL READ
      VEL1 = VEL
      DH1 = DH
      CALL ECHO
      CALL ERRSET (208, 256, -1, 1)
C
C.....COMPUTE OUTPUT FOR EXPERIMENTAL DATA IF NUMBER OF POINTS > 0
C
      IF (NOUT .GT. 0) THEN
         CALL INITL (DDVDT, DV, OUTPUT, TO)
         CALL GRPFRM
         CALL INFO1 (DATA, NOUT)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
         NDOF = 2 * NCOL
         NCOLEX = 2 * NCOL - 1
         TO = 0.D00
         H = 1.D-08
         INDEX = 1
         DV (1) = CO
         NLC = 2
         NUC = 2
C
C.....TIME LOOP FOR SIMULTATION AND OUTPUT AT EXPERIMENTAL POINTS
C
         DO 100 I = 1,NOUT
            IF (I .GT. 1) THEN
               IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
     1             (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
                   H = 1.D-08
                   INDEX = 1
                   DCODT = 0.000
                   DV (1) = DATA (I, 3)
                   WRITE (IOOUT, 1300)
                   IF (DV (1) .LT. 1.0E-10) THEN
                      VEL = VEL2
                      DH = DH1 * VEL2 / VEL1
                      CALL GRPFRM
                      WRITE (IOOUT, 1400)
                   ENDIF
               ELSE
                   DCODT = DATA (I, 4)
               ENDIF
            ELSE
               DCODT = DATA (I, 4)
            ENDIF
            TIME = DATA (I,1)
            CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME
```

```
     1                  EPSLON, 2, -2, INDEX, IWK, WK, IER)
            CALL REPORT (I, DV, RSDSUM, RDSQSM, RSD)
            OUTCON (I) = DV (NCOLEX)
  100    F(I) = RSD
C
C.....RESTORE ORIGINAL PARAMETERS
C
         DH = DH1
         VEL = VEL1
C
C.....CHECK THE MASS BALANCE
C
         CALL MASS1 (OUTCON, DV, C0, ZCOL, THETA, VEL, VEL2,
     1            RHO, DATA, NOUT, NCOL)
C
      ENDIF
C
C.....COMPUTE OUTPUT FOR COMPUTED POINTS IF NUMBER OF POINTS > 0
C
      IF (NSTEPS .GT. 0) THEN
         CALL INITL (DDVDT, DV, OUTPUT, TO)
         CALL GRPFRM
         CALL INFO2 (OUTPUT, NSTEPS)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
         NDOF = 2 * NCOL
         NCOLEX = 2 * NCOL - 1
         TO = 0.DOO
         H = 1.D-08
         INDEX = 1
         DV (1) = CO
         NLC = 2
         NUC = 2
C
C.....TIME LOOP FOR SIMULTATION AND OUTPUT AT COMPUTED POINTS
C
         WRITE (IOOUT, 1000)
         WRITE (IOOUT, 1100)
         DO 110 I = 1,NSTEPS
            IF (I .GT. 1) THEN
               IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     1             (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
                   H = 1.D-08
                   INDEX = 1
                   DCODT = 0.000
                   DV (1) = OUTPUT (I, 4)
                   WRITE (IOOUT, 1300)
                   IF (DV (1) .LT. 1.0E-10) THEN
                      VEL = VEL2
                      DH = DH1 * VEL2 / VEL1
                      CALL GRPFRM
                      WRITE (IOOUT, 1400)
                   ENDIF
               ELSE
                   DCODT = OUTPUT (I, 5)
               ENDIF
            ELSE
               DCODT = OUTPUT (I, 5)
            ENDIF
            TIME = OUTPUT (I,1)
            CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
     1                  EPSLON, 2, -2, INDEX, IWK, WK, IER)
            CTEMP = DV (NCOLEX)
            CNORM = CTEMP / CO
            OUTCON (I) = DV (NCOLEX)
```

```
  110    WRITE (IOOUT, 1200) (OUTPUT (I, J), J = 1, 3), CTEMP, CNORM
C
C.....RESTORE ORIGINAL PARAMETERS
C
         DH = DH1
         VEL = VEL1
C
C.....CHECK THE MASS BALANCE
C
         CALL MASS2 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
     1            RHO, OUTPUT, NSTEPS, NCOL)
C
      ENDIF
 1000 FORMAT ('1', 42X, 'SIMULATION OUTPUT FOR PLOTTING')
 1100 FORMAT ('O-------------------------------------------------',
     1        '--------------------------------------------------',
     2        '--------------------------------')
 1200 FORMAT (5E12.5)
 1300 FORMAT (' ', 20X, '***** ', 'ABRUPT INFLUENT CONCENTRATION ',
     1         'CHANGE. ', '***** ', 'DGEAR WAS RESET. ', '*****')
 1400 FORMAT (' ', 23X, '***** ', 'DESORPTION DETECTED. ',
     1        '***** ', 'HYDRODYNAMICS WERE RESET. ', '*****')
      STOP
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C                                                                         C
C     THIS SUBROUTINE READS THE BASIC INPUT REQUIRED FOR THE DUAL-         C
C     RESISTANCE SORPTION VERSION OF THE ADVECTIVE-DISPERSIVE EQUATION.   C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                 DESCRIPTION            C
C     -------------      ---------    ----------------------------------  C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM            C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO             C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION      C
C     NOUT               INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT  C
C     NSTEPS             INTEGER*4    NUMBER OF BED VOLUME STEP TO OUTPUT C
C                                                                         C
C                                                                         C
C     ALPHA              REAL*8       SLOW SORPTION PROPORTIONALITY CONST C
C     CO                 REAL*8       INITIAL CONCENTRATION               C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA             C
C                                     COLUMN 1 CONTAINS REPORT TIMES      C
C                                     COLUMN 2 GIVES EXPERIMENTAL         C
C                                     CONCENTRATION                       C
C                                     COLUMN 3 GIVES INFLUENT             C
C                                     CONCENTRATION                       C
C                                     COLUMN 4 GIVES DCO/DT               C
C     DH                 REAL*8       HYDRODYNAMIC DISPERSION COEFFICIENT C
C     EPSLON             REAL*8       ALLOWABLE ERROR FOR DGEAR           C
C     FKAPPA             REAL*8       FLUID PHASE LOSS RATE               C
C     FKF                REAL*8       FAST FREUNDLICH ISOTHERM COEFFICIENTC
C     FKS                REAL*8       SLOW FREUNDLICH ISOTHERM COEFFICIENTC
C     FKT                REAL*8       THROUGHPUT FREUNDLICH COEFFICIENT   C
C     FNF                REAL*8       FAST FREUNDLICH ISOTHERM EXPONENT   C
C     FNS                REAL*8       SLOW FREUNDLICH ISOTHERM EXPONENT   C
C     FNT                REAL*8       THROUGHPUT FREUNDLICH EXPONENT      C
C     RHO                REAL*8       SOLID PHASE PARTICLE DENSITY        C
C     SKAPPA             REAL*8       SOLID PHASE LOSS RATE               C
C     THETA              REAL*8       POROSITY                            C
C     THESIM             REAL*8       NUMBER OF BED VOLUMES TO SIMULATE   C
```

```
C     VEL               REAL*8     PORE VELOCITY                       C
C     VEL2              REAL*8     DESORPTION PORE VELOCITY            C
C     ZCOL              REAL*8     LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                      C
C                                                                      C
C     TITLE             CHARACTER*72 TITLE OF SIMULATION RUN           C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE READ
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      CHARACTER*72 TITLE
      DIMENSION DATA (100,4) , C (100), T (100)
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FKF, FNF, FKS, FNS, ALPHA,
     2               FKAPPA, SKAPPA, FKT, FNT, C, T,
     3               ICON, NCOL, NOUT, NSTEPS
      COMMON /WORDS/ TITLE
      COMMON /UNITS/ IOIN, IOOUT
      READ (IOIN, 1000) TITLE
      READ (IOIN, 1100) NCOL, NOUT, NSTEPS, THESIM, ICON
      IF (ABS (NOUT) .GT. 0) THEN
         NTEMP = ABS (NOUT)
         READ (IOIN, 1200) (DATA (I,1), I = 1, NTEMP)
         READ (IOIN, 1200) (DATA (I,2), I = 1, NTEMP)
      ENDIF
      READ (IOIN, 1200) VEL, DH, RHO, THETA, ZCOL, EPSLON
      READ (IOIN, 1200) CO, FKF, FNF, FKS, FNS, ALPHA
      READ (IOIN, 1200) FKAPPA, SKAPPA, FKT, FNT, VEL2
      READ (IOIN, 1200) (C (I), I = 1, ICON)
      READ (IOIN, 1200) (T (I), I = 1, ICON)
 1000 FORMAT (A72)
 1100 FORMAT (3I5, E12.5, I5)
 1200 FORMAT ((6E12.5))
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     THIS SUBROUTINE ECHOS THE INPUT TO DEVICE IOOUT.                 C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION               DESCRIPTION            C
C     -------------     ---------  ---------------------------------   C
C     IOIN              INTEGER*4  UNIT NUMBER TO READ FROM            C
C     IOOUT             INTEGER*4  UNIT NUMBER TO WRITE TO             C
C     NCOL              INTEGER*4  NUMBER OF NODES IN Z DIRECTION      C
C                                                                      C
C                                                                      C
C     ALPHA             REAL*8     SLOW SORPTION PROPORTIONALITY       C
C     CO                REAL*8     INITIAL CONCENTRATION               C
C     DATA              REAL*4     EXPERIMENTAL INPUT DATA             C
C                                  COLUMN 1 CONTAINS REPORT TIMES      C
C                                  COLUMN 2 GIVES EXPERIMENTAL         C
C                                  CONCENTRATION                       C
C                                  COLUMN 3 GIVES INFLUENT             C
C                                  CONCENTRATION                       C
C                                  COLUMN 4 GIVES DCO/DT               C
C     DH                REAL*8     HYDRODYNAMIC DISPERSION COEFFICIENT C
```

```
C     EPSLON            REAL*8       ALLOWABLE ERROR FOR DGEAR          C
C     FKAPPA            REAL*8       FLUID PHASE LOSS RATE              C
C     FKF               REAL*8       FAST FREUNDLICH ISOTHERM COEFFICIENTC
C     FKF               REAL*8       FAST FREUNDLICH ISOTHERM COEFFICIENTC
C     FKS               REAL*8       SLOW FREUNDLICH ISOTHERM COEFFICIENTC
C     FKT               REAL*8       THROUGHPUT FREUNDLICH COEFFICIENT  C
C     FNF               REAL*8       FAST FREUNDLICH ISOTHERM EXPONENT  C
C     FNS               REAL*8       SLOW FREUNDLICH ISOTHERM EXPONENT  C
C     FNT               REAL*8       THROUGHPUT FREUNDLICH EXPONENT     C
C     FKS               REAL*8       SLOW FREUNDLICH ISOTHERM COEFFICIENTC
C     PE                REAL*8       SYSTEM PECLET NUMBER               C
C     RHO               REAL*8       SOLID PHASE PARTICLE DENSITY       C
C     SKAPPA            REAL*8       SOLID PHASE LOSS RATE              C
C     THETA             REAL*8       POROSITY                           C
C     TMESIM            REAL*8       NUMBER OF BED VOLUMES TO SIMULATE  C
C     VEL               REAL*8       PORE VELOCITY                      C
C     VEL2              REAL*8       DESORPTION PORE VELOCITY           C
C     ZCOL              REAL*8       LENGTH OF COLUMN (Z DIRECTION)     C
C                                                                       C
C                                                                       C
C     TITLE             CHARACTER*72 TITLE OF SIMULATION RUN            C
C                                                                       C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE ECHO
      IMPLICIT REAL*8 (A - H,O - Z)
      REAL*4 DATA
      CHARACTER*72 TITLE
      DIMENSION DATA (100,4), C (100), T (100)
      COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FKF, FNF, FKS, FNS, ALPHA,
     2               FKAPPA, SKAPPA, FKT, FNT, C, T,
     3               ICON, NCOL, NOUT, NSTEPS
      COMMON /WORDS/ TITLE
      COMMON /UNITS/ IOIN, IOOUT
      PE = VEL * ZCOL / DH
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TITLE
      WRITE (IOOUT, 1400)
      WRITE (IOOUT, 1300)
      WRITE (IOOUT, 1400)
      WRITE (IOOUT, 1500) CO
      WRITE (IOOUT, 1600) DH
      WRITE (IOOUT, 1700) NCOL
      WRITE (IOOUT, 1800) RHO
      WRITE (IOOUT, 1900) VEL
      WRITE (IOOUT, 1950) VEL2
      WRITE (IOOUT, 2000) THETA
      WRITE (IOOUT, 2100) ZCOL
      WRITE (IOOUT, 2200) PE
      WRITE (IOOUT, 2300) FKF
      WRITE (IOOUT, 2400) FNF
      WRITE (IOOUT, 2500) FKS
      WRITE (IOOUT, 2600) FNS
      WRITE (IOOUT, 2700) ALPHA
      WRITE (IOOUT, 2800) FKAPPA
      WRITE (IOOUT, 2900) SKAPPA
      WRITE (IOOUT, 3000) ICON
      WRITE (IOOUT, 3100)
      WRITE (IOOUT, 3200) (C (I), I = 1, ICON)
      WRITE (IOOUT, 3300)
```

```
      WRITE (IOOUT, 3200) (T (I), I = 1, ICON)
 1000 FORMAT ('1', 50X, 'PLUG FLOW DISPERSION REACTOR')
 1100 FORMAT (' ', 47X, 'FIRST ORDER PARALLEL EQUILIBRIUM SIMULATION')
 1200 FORMAT (' ', 26X, A72)
 1300 FORMAT (' ', 15X, 'VARIABLE', 16X, 'DESCRIPTION', 30X, '   VALUE',
     1         16X, 'UNITS   ', 16X)
 1400 FORMAT ('0-------------------------------------------------',
     1         '--------------------------------------------------',
     2         '--------------------------------')
 1500 FORMAT ('0', 15X, 'CO    ', 18X, 'INITIAL CONCENTRATION    ', 14X,
     1         E10.4, 16X, 'G/CM**3')
 1600 FORMAT (' ', 15X, 'DH    ', 18X, 'HYDRODYNAMIC DISPERSION  ', 14X,
     1         E10.4, 16X, 'CM**2/HOUR')
 1700 FORMAT (' ', 15X, 'NCOL  ', 18X, 'NUMBER OF COLUMN STEPS   ', 16X,
     1         I5)
 1800 FORMAT (' ', 15X, 'RHO   ', 18X, 'SOLID PHASE DENSITY      ', 14X,
     1         E10.4, 16X, 'GR/CM**3')
 1900 FORMAT (' ', 15X, 'VEL   ', 18X, 'PORE VELOCITY            ', 14X,
     1         E10.4, 16X, 'CM/HOUR')
 1950 FORMAT (' ', 15X, 'VEL   ', 18X, 'DESORPTION VELOCITY      ', 14X,
     1         E10.4, 16X, 'CM/HOUR')
 2000 FORMAT (' ', 15X, 'THETA ', 18X, 'VOID VOLUME FRACTION     ', 14X,
     1         E10.4)
 2100 FORMAT (' ', 15X, 'ZCOL  ', 18X, 'LENGTH OF COLUMN         ', 14X,
     1         E10.4, 16X, 'CM')
 2200 FORMAT (' ', 15X, 'PE    ', 18X, 'PECLET NUMBER            ', 14X,
     1         E10.4)
 2300 FORMAT (' ', 15X, 'FKF   ', 18X, 'FREUNDLICH FAST COEFFICIENT',
     1         12X, E10.4, 16X, '((CM**3/GR)**FNF)')
 2400 FORMAT (' ', 15X, 'FNF   ', 18X, 'FREUNDLICH FAST EXPONENT ', 14X,
     1         E10.4)
 2500 FORMAT (' ', 15X, 'FKS   ', 18X, 'FREUNDLICH SLOW COEFFICIENT',
     1         12X, E10.4, 16X, '((CM**3/GR)**FNS)')
 2600 FORMAT (' ', 15X, 'FNS   ', 18X, 'FREUNDLICH SLOW EXPONENT ', 14X,
     1         E10.4)
 2700 FORMAT (' ', 15X, 'ALPHA ', 18X, 'FIRST ORDER RATE COEFFICIENT',
     1         11X, E10.4, 16X, '1/HR')
 2800 FORMAT (' ', 15X, 'FKAPPA', 18X, 'FLUID PHASE LOSS RATE    ', 14X,
     1         E10.4, 16X, '1/HR')
 2900 FORMAT (' ', 15X, 'SKAPPA', 18X, 'SOLID PHASE LOSS RATE    ', 14X,
     1         E10.4, 16X, '1/HR')
 3000 FORMAT (' ', 15X, 'ICON  ', 18X, 'NUMBER OF INFLUENTS      ', 14X,
     1          I5)
 3100 FORMAT (' ', 39X, 'INFLUENT CONCENTRATIONS')
 3200 FORMAT (' ', 14X, 10E10.3)
 3300 FORMAT (' ', 39X, 'INFLUENT TIMES')
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE DATA MATRIX TO        C
C     UNIT IOOUT.                                                      C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME      PRECISION             DESCRIPTION             C
C     -------------      ---------    -------------------------------- C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM         C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO          C
C     NCOUNT             INTEGER*4    LOOP COUNTER                     C
C     NOUT               INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT C
C                                                                      C
```

```
C     DATA                  REAL*4      EXPERIMENTAL INPUT DATA            C
C                                       COLUMN 1 CONTAINS REPORT TIMES     C
C                                       COLUMN 2 GIVES EXPERIMENTAL        C
C                                       CONCENTRATION                      C
C                                       COLUMN 3 GIVES INFLUENT            C
C                                       CONCENTRATION                      C
C                                       COLUMN 4 GIVES DCO/DT              C
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INFO1 (DATA, NOUT)
      REAL*4 DATA
      DIMENSION DATA (100,4)
      COMMON /UNITS/ IOIN, IOOUT
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200)
      WRITE (IOOUT, 1100)
      DO 100 NCOUNT = 1, NOUT
          WRITE (IOOUT, 1300) ( DATA (NCOUNT, I), I = 1, 4)
 100  CONTINUE
 1000 FORMAT ('1', 42X, 'EXPERIMENTAL COMPARISON MODE INFORMATION')
 1100 FORMAT ('0----------------------------------------------',
     1        '-----------------------------------------------',
     2        '-------------------------------')
 1200 FORMAT (' ', 28X, 'TIME',5X,'CONCENTRATION OUT', 6X,
     1        'CONCENTRATION IN', 16X, 'DCO/DT')
 1300 FORMAT (' ', 20X, E12.5, 10X, E12.5, 10X, E12.5, 10X, E12.5)
      RETURN
      END
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C                                                                          C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE OUTPUT MATRIX TO          C
C     UNIT IOOUT.                                                          C
C                                                                          C
C                                                                          C
C     VARIABLE MAP:                                                        C
C     ------------                                                         C
C                                                                          C
C     VARIABLE NAME     PRECISION                DESCRIPTION               C
C     -------------     ---------   ----------------------------------     C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM               C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO                C
C     NCOUNT            INTEGER*4   LOOP COUNTER                           C
C     NSTEPS            INTEGER*4   NUMBER OF EQUAL BED VOLUME STEPS       C
C                                   TO BE CALCULATED                       C
C                                                                          C
C     OUTPUT            REAL*4      A MATRIX OF SIMULATION STATUS          C
C                                   COL 1 CONTAINS REPORT TIME             C
C                                   COL 2 CONTAINS BED VOLUMES             C
C                                   COL 3 CONTAINS THROUGHPUT              C
C                                   COL 4 GIVES INFLUENT CONCENTRATION     C
C                                   COL 5 GIVES DCO/DT                     C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INFO2 (OUTPUT, NSTEPS)
      REAL*4 OUTPUT
```

```
      DIMENSION OUTPUT (250, 5)
      COMMON /UNITS/ IOIN, IOOUT
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200)
      WRITE (IOOUT, 1100)
      DO 100 NCOUNT = 1, NSTEPS
          WRITE (IOOUT, 1300) (OUTPUT (NCOUNT, I), I = 1, 5)
 100  CONTINUE
 1000 FORMAT ('1', 48X, 'SIMULATION MODE INFORMATION')
 1100 FORMAT ('0---------------------------------------------------',
     1         '---------------------------------------------------',
     2         '---------------------------------')
 1200 FORMAT (' ', 12X, 'REPORT TIME', 11X, 'BED VOLUMES', 12X,
     1       'THROUGHPUT', 6X, 'CONCENTRATION IN',
     2       16X, 'DCO/DT')
 1300 FORMAT (' ', 11X, E12.5, 10X, E12.5, 10X, E12.5, 10X,
     1       E12.5, 10X, E12.5)
      RETURN
      END
C                                                                          C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C                                                                          C
C     THIS SUBROUTINE INITIALIZES THE MODEL VARIABLES FOR THE SIMULATION   C
C     OF THE ADVECTIVE-DISPERSIVE EQUATION.                                C
C                                                                          C
C                                                                          C
C     VARIABLE MAP:                                                        C
C     ------------                                                         C
C                                                                          C
C     VARIABLE NAME       PRECISION                 DESCRIPTION            C
C     -------------       ---------    -----------------------------------  C
C     I                   INTEGER*4    WORKING VARIABLE                    C
C     J                   INTEGER*4    WORKING VARIABLE                    C
C                                                                          C
C                                                                          C
C     ALPHA               REAL*8       SLOW SORPTION PROPORTIONALITY       C
C     DATA                REAL*4       EXPERIMENTAL INPUT DATA             C
C                                      COLUMN 1 CONTAINS REPORT TIMES      C
C                                      COLUMN 2 GIVES EXPERIMENTAL         C
C                                      CONCENTRATION                       C
C                                      COLUMN 3 GIVES INFLUENT             C
C                                      CONCENTRATION                       C
C                                      COLUMN 4 GIVES DCO/DT               C
C     DDVDT               REAL*8       DERIVATIVE OF DEPENDENT VARIABLES   C
C     DG                  REAL*8       RATIO OF TOTAL MASSS ADSORBABLE ON  C
C                                      SOLID PHASE TO TOTAL MASS IN FLUID  C
C                                      PHASE                               C
C     DT                  REAL*8       DURATION OF EACH TIME STEP          C
C     DTN                 REAL*8       REAL EQUIVALENT OF NSTEPS           C
C     DV                  REAL*8       DEPENDENT VARIABLE VECTOR           C
C     EPSMCN              REAL*8       MACHINE EPSILON                     C
C     FKAPPA              REAL*8       FLUID PHASE LOSS RATE               C
C     FKF                 REAL*8       FAST FREUNDLICH ISOTHERM COEFFICIENTC
C     FKS                 REAL*8       SLOW FREUNDLICH ISOTHERM COEFFICIENTC
C     FKT                 REAL*8       THROUGHPUT FREUNDLICH COEFFICIENT   C
C     FNF                 REAL*8       FAST FREUNDLICH ISOTHERM EXPONENT   C
C     FNS                 REAL*8       SLOW FREUNDLICH ISOTHERM EXPONENT   C
C     FNT                 REAL*8       THROUGHPUT FREUNDLICH EXPONENT      C
C     OUTPUT              REAL*8       A MATRIX OF SIMULATION STATUS       C
C                                      COL 1 CONTAINS REPORT TIME          C
C                                      COL 2 CONTAINS BED VOLUMES          C
C                                      COL 3 CONTAINS THROUGHPUT           C
C                                      COL 4 GIVES INFLUENT CONCENTRATION  C
```

```
C                                     COL 5 GIVES DCO/DT                  C
C     SKAPPA             REAL*8       SOLID PHASE LOSS RATE               C
C     TEMP               REAL*8       TEMPORARY VARIABLE FOR EPSMCN CALC  C
C     TMESIM             REAL*8       NUMBER OF BED VOLUMES TO SIMULATE   C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INITL (DDVDT, DV, OUTPUT, TO)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA, OUTPUT
      LOGICAL DONE
      DIMENSION DATA (100,4), DDVDT (101), DV (101), OUTPUT (250,5)
      DIMENSION C (100), T (100)
      COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FKF, FNF, FKS, FNS, ALPHA,
     2               FKAPPA, SKAPPA, FKT, FNT, C, T,
     3               ICON, NCOL, NOUT, NSTEPS
      DO 100 I = 1, 101
         DV (I) = 0.D00
  100 DDVDT (I) = 0.D00
C
C.....COMPUTE SIMULATION OUTPUT POINTS IF DESIRED
C
      IF (NSTEPS .GT. 0) THEN
         TAU = ZCOL / VEL
         DG = 1 + (RHO * (1 - THETA) / THETA) * (FKT * (CO ** FNT)) / CO
         DTN = NSTEPS
         DT = TMESIM / DTN
         TEMPT = 0.D00
         DO 110 I = 1, NSTEPS
            TEMPT = TEMPT + DT
            OUTPUT (I,1) = TEMPT
            OUTPUT (I,2) = TEMPT / TAU
  110    OUTPUT (I,3) = TEMPT / (TAU * DG)
      ENDIF
C
C.....CALCULATE THE MACHINE EPSILON FOR USE AS A LOWER ERROR BOUND
C
      EPSMCN = 1.0
  120 EPSMCN = EPSMCN / 2.0D00
      TEMP = 1.0D00 + EPSMCN
      IF (TEMP .GT. 1.0D00) GO TO 120
C
C.....CALCULATE THE INFLUENT CONCENTRATIONS FOR USE IN THE
C.....EXPERIMENTAL COMPARISON MODE BY INTERPOLATING FROM THE
C.....LIST OF KNOWN CONCENTRATIONS.
C
      IF (NOUT .GT. 0) THEN
          DO 130 NCOUNT = 1, NOUT, 1
              TEND = DATA (NCOUNT, 1)
              DONE = .FALSE.
              DO 140 NLOOP = 1, (ICON - 1)
              IF (DONE .EQV. .FALSE.) THEN
                  DNTIME = T (NLOOP)
                  UPTIME = T (NLOOP + 1)
                  DNCON  = C (NLOOP)
                  UPCON  = C (NLOOP + 1)
                  IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND))
     1            THEN
                      DATA (NCOUNT, 3) = DNCON + (UPCON - DNCON)
     1                * (TEND - DNTIME) / (UPTIME - DNTIME)
```

```
                      DONE = .TRUE.
                  ENDIF
               ENDIF
 140           CONTINUE
               IF (NCOUNT .GT. 1) THEN
                   DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) -
     1             DATA (NCOUNT - 1, 3)) / (DATA (NCOUNT, 1) -
     2             DATA (NCOUNT - 1, 1))
               ELSE
                   DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) - CO)
     1             / (DATA ( NCOUNT, 1) - TO)
               ENDIF
 130       CONTINUE
      ENDIF
C
C.....CALCULATE INFLUENT CONCENTRATION FOR USE IN THE
C.....SIMULATION MODE.
C
      IF (NSTEPS .GT. 0) THEN
          DO 150 NCOUNT = 1, NSTEPS, 1
              TEND = OUTPUT(NCOUNT, 1)
              DONE = .FALSE.
              DO 160 NLOOP = 1, (ICON - 1)
              IF (DONE .EQV. .FALSE.) THEN
                  DNTIME = T (NLOOP)
                  UPTIME = T (NLOOP + 1)
                  DNCON  = C (NLOOP)
                  UPCON  = C (NLOOP + 1)
                  IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND))
     1            THEN
                      OUTPUT (NCOUNT, 4) = DNCON + (UPCON - DNCON)
     1                * (TEND - DNTIME) / (UPTIME - DNTIME)
                      DONE = .TRUE.
                  ENDIF
              ENDIF
 160          CONTINUE
              IF (NCOUNT .GT. 1) THEN
                  OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) -
     1            OUTPUT (NCOUNT - 1, 4)) / (OUTPUT (NCOUNT, 1) -
     2            OUTPUT (NCOUNT - 1, 1))
              ELSE
                  OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - CO)
     1            / (OUTPUT ( NCOUNT, 1) - TO)
              ENDIF
 150      CONTINUE
C
C.....COMPUTE BED VOLUMES FOR VARIABLE VELOCITY CASE
C
          OLDVEL = VEL
          TEMPT = 0.D00
          DO 170 I = 1, NSTEPS
              TEMPT = TEMPT + DT
              IF (OUTPUT (I, 4) . GT. 1.0E-10) THEN
                  CURVEL = VEL
              ELSE
                  CURVEL = VEL2
              ENDIF
              AVGVEL = (OLDVEL * (TEMPT - DT) + CURVEL * DT) / TEMPT
              TAU = ZCOL / AVGVEL
              OUTPUT (I, 2) = TEMPT / TAU
              OUTPUT (I, 3) = TEMPT / (TAU * DG)
              OLDVEL = AVGVEL
 170      CONTINUE
      ENDIF
      RETURN
      END
```

```
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C                                                                         C
C     THIS SUBROUTINE COMPUTES VARIABLE GROUPINGS TO MINIMIZE EFFORT IN   C
C     THE SOLUTION OF THE SET OF SIMULTANEOUS NONLINEAR EQUATIONS.        C
C                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                DESCRIPTION             C
C     -------------      ---------    ----------------------------------  C
C     I                  INTEGER*4    TIME STEP INDEX                     C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION      C
C     NDOF               INTEGER*4    NUMBER OF DEGREES OF FREEDOM        C
C     NOUT               INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT  C
C                                                                         C
C                                                                         C
C     ALPHA              REAL*8       SLOW SORPTION PROPORTIONALITY       C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA             C
C                                     COLUMN 1 CONTAINS REPORT TIMES      C
C                                     COLUMN 2 GIVES EXPERIMENTAL         C
C                                     CONCENTRATION                       C
C                                     COLUMN 3 GIVES INFLUENT             C
C                                     CONCENTRATION                       C
C                                     COLUMN 4 GIVES DCO/DT               C
C     DGN                REAL*8       DIAGONAL VARIABLE GROUPING TERMS    C
C     DGNLOW             REAL*8       LOWER DIAGONAL VARAIBLE GROUPINGS   C
C     DGNUP              REAL*8       UPPER DIAGONAL VARIABLE GROUPINGS   C
C     DH                 REAL*8       HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DZ                 REAL*8       COLUMN STEP SIZE IN Z DIRECTION     C
C     FKAPPA             REAL*8       FLUID PHASE LOSS RATE               C
C     FKF                REAL*8       FAST FREUNDLICH ISOTHERM COEFFICIENTC
C     FKS                REAL*8       SLOW FREUNDLICH ISOTHERM COEFFICIENTC
C     FKT                REAL*8       THROUGHPUT FREUNDLICH COEFFICIENT   C
C     FNF                REAL*8       FAST FREUNDLICH ISOTHERM EXPONENT   C
C     FNS                REAL*8       SLOW FREUNDLICH ISOTHERM EXPONENT   C
C     FNT                REAL*8       THROUGHPUT FREUNDLICH EXPONENT      C
C     FSSDLS             REAL*8       FIRST ORDER DECAY LOSS ON SOLID     C
C     REALND             REAL*8       NUMBER OF NODES IN COLUMN           C
C     RFGRP              REAL*8       RETARDATION VARIABLE GROUPING       C
C     RHO                REAL*8       SOLID PHASE PARTICLE DENSITY        C
C     SKAPPA             REAL*8       SOLID PHASE LOSS RATE               C
C     SLSPAC             REAL*8       SLOW SOLID PHASE ACTUAL             C
C                                     CONCENTRATION GROUPING              C
C     SLSPEQ             REAL*8       SLOW SOLID PHASE EQULIB GROUPING    C
C     THETA              REAL*8       POROSITY                            C
C     TMESIM             REAL*8       NUMBER OF BED VOLUMES TO SIMULATE   C
C     VEL                REAL*8       PORE VELOCITY                       C
C     VEL2               REAL*8       DESORPTION PORE VELOCITY            C
C     ZCOL               REAL*8       LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE GRPFRM
      IMPLICIT REAL*8(A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), C (100), T (100)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, RFGRP, FSSDLS, SLSPEQ
```

```
     1                    SLSPAC
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FKF, FNF, FKS, FNS, ALPHA,
     2               FKAPPA, SKAPPA, FKT, FNT, C, T,
     3               ICON, NCOL, NOUT, NSTEPS
C
C.....COMPUTE MACROSCOPIC ADVECTIVE-DSIPERSIVE EQUATION FLUID PHASE GROUPINS
C
      REALND = NCOL
      DZ = ZCOL / (REALND - 1.DOO)
      DGNLOW = DH / (DZ ** 2.DOO) + VEL / (2.DOO * DZ)
      DGN = -2.DOO * DH / (DZ ** 2.DOO) - FKAPPA
      DGNUP = DH / (DZ ** 2.DOO) - VEL / (2.DOO *DZ)
      RFGRP = FKF * FNF * (1 - THETA) * RHO / THETA
      FSSDLS = (RHO * (1 - THETA) / THETA) * SKAPPA * FKF
      SLSPEQ = (RHO * (1 - THETA) / THETA) * ALPHA * FKS
      SLSPAC = (RHO * (1 - THETA) / THETA) * ALPHA
      RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C                                                                       C
C     THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE      C
C     DEPENDENT VARIABLE AS A FUNCTION OF TIME. THE EQUATIONS ARE       C
C     SOLVED USING THE VARIABLE GROUPINGS PREVIOUSLY DERIVED IN ORDER   C
C     TO SAVE COMPUTATIONAL EFFORT.                                     C
C                                                                       C
C     ALL DEPENDENT VARIABLE ARE STORED IN A SINGLE ARRAY DV, AND ALL   C
C     TEMPORAL DERIVATIVES OF THE DEPENDENT VARIABLE ARE STORED IN A    C
C     SINGLE ARRAY TERMED DDVDT. THE INDEX NUMBERING OF THESE ARRAYS    C
C     MAY BE DESCRIBED AS FOLLOWS:                                      C
C          A) THE FIRST ENTRY (1) IS THE INLET NODE OF THE COLUMN.      C
C          B) THE SECOND ENTRY (2) IS LEFT BLANK .                      C
C          C) THE SUCCEEDING REFERENCES TO THE FLUID PHASE ARE AT THE   C
C             ODD-NUMBERED INDICES.                                     C
C          D) THE SOLID PHASE REFERENCES ARE AT THE EVEN-NUMBERED INDEX C
C             FOLLOWING THE ASSOCIATED FLUID REFERENCES.                C
C          E) THE TOTAL DEGREES OF FREEDON ARE EQUAL TO 2 * NCOL.       C
C          F) THE THE NEXT-TO-LAST ENTRY IS THE FLUID PHASE EXIT CON'N. C
C          G) THE LAST ENTRY IS THE SOLID PHASE CONCENTRATION AT THE EXIT.C
C                                                                       C
C                                                                       C
C     VARIABLE MAP:                                                     C
C     ------------                                                      C
C                                                                       C
C     VARIABLE NAME      PRECISION                DESCRIPTION           C
C     -------------      ---------    --------------------------------- C
C     I                  INTEGER*4    TEMPORARY VARAIBLE                C
C     J                  INTEGER*4    TEMPORARY VARAIBLE                C
C     K                  INTEGER*4    INDEX FOR SPHERICAL GROUPINGS     C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION    C
C     NDOF               INTEGER*4    NUMBER OF DEGREES OF FREEDOM      C
C     NEND               INTEGER*4    VARIABLE INDEX FOR LAST COLUMN NODE C
C     NSTART             INTEGER*4    LOOP STARTING POSITION            C
C     NSTEP              INTEGER*4    STEP SIZE FOR LOOP                C
C     NSTOP              INTEGER*4    LOOP STOPPING POSITION            C
C                                                                       C
C                                                                       C
C     ALPHA              REAL*8       SLOW SORPTION PROPORTIONALITY     C
C     CNF                REAL*8       FAST EQUILIBRIUM CONCENTRATION    C
C     CNS                REAL*8       SLOW EQUILIBRIUM CONCENTRATION    C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA           C
C                                     COLUMN 1 CONTAINS REPORT TIMES    C
C                                     COLUMN 2 GIVES EXPERIMENTAL       C
```

```
C                                  CONCENTRATION                       C
C                                  COLUMN 3 GIVES INFLUENT             C
C                                  CONCENTRATION                       C
C                                  COLUMN 4 GIVES DCO/DT               C
C     DDVDT            REAL*8      DERIVATIVE OF DEPENDENT VARIABLES   C
C     DGN              REAL*8      DIAGONAL VARIABLE GROUPING TERMS    C
C     DGNLOW           REAL*8      LOWER DIAGONAL VARAIBLE GROUPINGS   C
C     DGNUP            REAL*8      UPPER DIAGONAL VARIABLE GROUPINGS   C
C     DV               REAL*8      DEPENDENT VARIABLE VECTOR           C
C     FKAPPA           REAL*8      FLUID PHASE LOSS RATE               C
C     FKF              REAL*8      FAST FREUNDLICH ISOTHERM COEFFICIENTC
C     FKS              REAL*8      SLOW FREUNDLICH ISOTHERM COEFFICIENTC
C     FKT              REAL*8      THROUGHPUT FREUNDLICH COEFFICIENT   C
C     FNF              REAL*8      FAST FREUNDLICH ISOTHERM EXPONENT   C
C     FNS              REAL*8      SLOW FREUNDLICH ISOTHERM EXPONENT   C
C     FNT              REAL*8      THROUGHPUT FREUNDLICH EXPONENT      C
C     FSSDLS           REAL*8      FIRST ORDER DECAY LOSS ON SOLID     C
C     RFGRP            REAL*8      RETARDATION VARIABLE GROUPING       C
C     RFI              REAL*8      RETARDATION AT NODE I               C
C     SKAPPA           REAL*8      SOLID PHASE LOSS RATE               C
C     SLSPAC           REAL*8      SLOW SOLID PHASE ACTUAL             C
C                                  CONCENTRATION GROUPING              C
C     SLSPEQ           REAL*8      SLOW SOLID PHASE EQULIB GROUPING    C
C     THESIM           REAL*8      NUMBER OF BED VOLUMES TO SIMULATE   C
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE CPRIME (NVRB, TIME, DV, DDVDT)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      CHARACTER*72 TITLE
      DIMENSION DV (101), DDVDT (101), DATA (100,4)
      DIMENSION C (100), T (100)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, RFGRP, FSSDLS, SLSPEQ,
     1                SLSPAC
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FKF, FNF, FKS, FNS, ALPHA,
     2               FKAPPA, SKAPPA, FKT, FNT, C, T,
     3               ICON, NCOL, NOUT, NSTEPS
      COMMON /FLUID/ DCODT
C
C.....FORM FLUID PHASE DERIVATIVES
C
      NDOF = NCOL * 2
C
C.....FIRST THE INLET CONDITION
C
      DDVDT (1) = DCODT
C
C.....SECOND THE INTERIOR FLUID PHASE DERIVATIVES
C
      NEND = NDOF - 3
      DO 100 I = 3, NEND, 2
      IF (DV (I) .GT. 0.D00) THEN
         RFI = 1.D00 + RFGRP * (DV (I) ** (FNF - 1.D00))
         CNF = DV (I) ** FNF
         CNS = DV (I) ** FNS
      ELSE
         RFI = 1.D00
         CNF = 0.D00
         CNS = 0.D00
```

```
      ENDIF
  100 DDVDT (I) = (DGNLOW * DV (I-2) + DGN * DV (I) + DGNUP * DV (I+2)
     1              - FSSDLS * CNF - SLSPEQ * CNS + SLSPAC * DV (I + 1))
     2              / RFI
C
C.....FORM THE OUTLET BOUNDARY CONDITION FOR THE FLUID PHASE
C
      I = NDOF - 1
      IF (DV (I) .GT. 0.D00) THEN
         RFI = 1.D00 + RFGRP * (DV (I) ** (FNF - 1.D00))
         CNF = DV (I) ** FNF
         CNS = DV (I) ** FNS
      ELSE
         RFI = 1.D00
         CNF = 0.D00
         CNS = 0.D00
      ENDIF
      DDVDT (I) = ((DGNLOW + DGNUP) * DV (I - 2) + DGN * DV (I)
     1              - FSSDLS * CNF - SLSPEQ * CNS + SLSPAC * DV (I + 1))
     2              / RFI
C
C.....FORM THE SOLID PHASE EQUATIONS
C
      DO 110 I = 4, NDOF, 2
      J = I - 1
      IF (DV (J) .GT. 0.D00) THEN
         CNS = DV (J) ** FNS
      ELSE
         CNS = 0.D00
      ENDIF
  110 DDVDT (I) = ALPHA * (FKS * CNS - DV (I)) - SKAPPA * DV (I)
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE     C
C     DDVDT ARRAY AS A FUNCTION OF THE DEPENDENT VARIABLE (DV).        C
C     SINCE THE FINITE DIFFERENCE OPTION IS USED IN THE CALL TO DGEAR  C
C     (I.E. MITER= -2) THIS ROUTINE IS A BLANK.                        C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C
      SUBROUTINE FCNJ (N, X, Y, PD)
      REAL*8 Y(N), PD(N,N), X
      RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     THIS SUBROUTINE USES THE EXPERIMENTAL DATA INPUT AND THE MODEL    C
C     SIMULATION OUTPUT TO COMPUTE THE MODEL RESIDUAL, RESIDUAL SQUARED, C
C     SUM OF RESIDUALS, AND SUM OF RESIDUALS SQUARED. THESE COMPUTATIONS C
C     ARE OUTPUT TO UNIT IOOUT.                                        C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME      PRECISION               DESCRIPTION           C
C     -------------      ---------     ------------------------------- C
```

```
C     I                 INTEGER*4  TIME STEP INDEX                     C
C     IOIN              INTEGER*4  UNIT NUMBER TO READ FROM            C
C     IOOUT             INTEGER*4  UNIT NUMBER TO WRITE TO             C
C     NCOL              INTEGER*4  NUMBER OF NODES IN Z DIRECTION      C
C     NDOF              INTEGER*4  NUMBER OF DEGREES OF FREEDOM        C
C     NDV               INTEGER*4  INDEX OF OUTLET CONCENTRATION       C
C     NOUT              INTEGER*4  NUMBER OF POINTS SIMULATED          C
C                                                                      C
C                                                                      C
C     ALPHA             REAL*8     SLOW SORPTION PROPORTIONALITY       C
C     CNORM             REAL*8     NORMALIZED OUTLET CONCENTRATION     C
C     CO                REAL*8     INITIAL CONCENTRATION               C
C     DATA              REAL*4     EXPERIMENTAL INPUT DATA             C
C                                  COLUMN 1 CONTAINS REPORT TIMES      C
C                                  COLUMN 2 GIVES EXPERIMENTAL         C
C                                  CONCENTRATION                       C
C                                  COLUMN 3 GIVES INFLUENT             C
C                                  CONCENTRATION                       C
C                                  COLUMN 4 GIVES DCO/DT               C
C     DV                REAL*8     DEPENDENT VARIABLE VECTOR           C
C     FKAPPA            REAL*8     FLUID PHASE LOSS RATE               C
C     FKF               REAL*8     FAST FREUNDLICH ISOTHERM COEFFICIENTC
C     FKS               REAL*8     SLOW FREUNDLICH ISOTHERM COEFFICIENTC
C     FKT               REAL*8     THROUGHPUT FREUNDLICH COEFFICIENT   C
C     FNF               REAL*8     FAST FREUNDLICH ISOTHERM EXPONENT   C
C     FNS               REAL*8     SLOW FREUNDLICH ISOTHERM EXPONENT   C
C     FNT               REAL*8     THROUGHPUT FREUNDLICH EXPONENT      C
C     RDSQSM            REAL*8     SUM OF RESIDUALS SQUARED            C
C     RSD               REAL*8     RESIDUAL OF POINT AT OUTLET         C
C     RSDSQ             REAL*8     RESIDUAL SQUARED OF POINT AT OUTLET C
C     RSDSUM            REAL*8     SUM OF RESIDUALS                    C
C     SKAPPA            REAL*8     SOLID PHASE LOSS RATE               C
C     TMESIM            REAL*8     NUMBER OF BED VOLUMES TO SIMULATE   C
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE REPORT (I, DV, RSDSUM, RDSQSM, RSD)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), DV(101), C (100), T (100)
      COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FKF, FNF, FKS, FNS, ALPHA,
     2               FKAPPA, SKAPPA, FKT, FNT, C, T,
     3               ICON, NCOL, NOUT, NSTEPS
      COMMON /UNITS/ IOIN, IOOUT
      NDV = (NCOL * 2) - 1
      IF (I .EQ. 1) THEN
         WRITE (IOOUT, 1000)
         WRITE (IOOUT, 1100)
         WRITE (IOOUT, 1200)
         WRITE (IOOUT, 1100)
         RSDSUM = 0.D00
         RDSQSM = 0.D00
      ENDIF
      CNORM = DV (NDV) / CO
      RSD = DATA (I,2) - CNORM
      RSDSQ = RSD * RSD
      RSDSUM = RSDSUM + RSD
      RDSQSM = RDSQSM + RSDSQ
      WRITE (IOOUT, 1300) DATA (I,1), DATA (I,2), CNORM, RSD, RSDSQ
      IF (I .GE. NOUT) THEN
         WRITE (IOOUT, 1100)
```

```
         WRITE (IOOUT, 1400) RSDSUM, RDSQSM
      ENDIF
 1000 FORMAT ('1',58X,'RESIDUAL REPORT')
 1100 FORMAT ('O--------------------------------------------------',
     1        '---------------------------------------------------',
     2        '---------------------------------')
 1200 FORMAT ('O',19X,'TIME',10X,'C/Co (INPUT)',10X,'C/Co (PREDICTED)',
     1        14X,'RESIDUAL',10X,'RESIDUAL SQUARED')
 1300 FORMAT (' ',11X,E12.5,10X,E12.5,14X,E12.5,10X,E12.5,14X,E12.5)
 1400 FORMAT (' ',81X,E12.5,14X,E12.5)
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE          C
C     COMPLETION OF SIMULATION FOR EXPERIMENTAL COMPARISON.            C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION                DESCRIPTION           C
C     -------------     ---------   ---------------------------------  C
C     I                 INTEGER*4   LOOP INDEX                         C
C     IFLAG1            INTEGER*4   FLAG FOR ABRUPT CONCENTRATION      C
C                                   CHANGE                             C
C     IFLAG2            INTEGER*4   FLAG FOR DESORPTION                C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION     C
C     NOUT              INTEGER*4   NUMBER OF TEMPORAL STEPS TO OUTPUT C
C                                                                      C
C                                                                      C
C                                                                      C
C     BAL               REAL*8      RATIO OF MASS IN TO MASS ACCOUNTED C
C     CO                REAL*8      INITIAL CONCENTRATION              C
C     DATA              REAL*4      EXPERIMENTAL INPUT DATA            C
C                                   COLUMN 1 CONTAINS REPORT TIMES     C
C                                   COLUMN 2 GIVES EXPERIMENTAL        C
C                                   CONCENTRATION                      C
C                                   COLUMN 3 GIVES INFLUENT            C
C                                   CONCENTRATION                      C
C                                   COLUMN 4 GIVES DCO/DT              C
C     DV                REAL*8      DEPENDENT VARIABLE VECTOR          C
C     OUTCON            REAL*8      EXIT CONCENTRATION VECTOR          C
C     RCOL              REAL*8      REAL EQUIVALENT OF NCOL            C
C     THETA             REAL*8      POROSITY                           C
C     TOTDRY            REAL*8      TOTAL MASS LEFT ON SOLID PHASE     C
C     TOTINF            REAL*8      TOTAL MASS INJECTED INTO COLUMN    C
C     TOTOUT            REAL*8      TOTAL MASS TO EXIT COLUMN          C
C     TOTWET            REAL*8      TOTAL MASS LEFT IN LIQUID PHASE    C
C     VEL               REAL*8      PORE VELOCITY                      C
C     VEL2              REAL*8      DESORPTION PORE VELOCITY           C
C     ZCOL              REAL*8      LENGTH OF COLUMN (Z DIRECTION)     C
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
```

```
      SUBROUTINE MASS1 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
     1            RHO, DATA, NOUT, NCOL)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100, 4), DV (101), OUTCON (250)
      COMMON /UNITS/ IOIN, IOOUT
C
      IFLAG1 = 0
      IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
      VEL1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
C
C     CALCULATE  CONTAMINANT INJECTED DURING FIRST
C     REPORT PERIOD.
C
          TOTINF = CO * DATA (1, 1) * VEL * THETA +
     1             0.5 * (DATA (1, 3) - CO) * DATA (1, 1) *
     2             VEL * THETA
C
C     CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
C
      DO 100 I = 2, (NOUT - 1)
         IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
     1       (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
            IFLAG1 = 1
            TOTINF = TOTINF +
     1                DATA (I, 3) * (DATA (I, 1) - DATA (I - 1, 1)) *
     2                VEL * THETA
         ELSE
            TOTINF = TOTINF +
     1                DATA (I - 1, 3) *
     2                (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA +
     3                0.5 * (DATA (I, 3) - DATA (I - 1, 3)) *
     4                (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
         ENDIF
 100  CONTINUE
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
C     CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
C     THE FIRST REPORT PERIOD
C
      TOTOUT = (0.75 * 0.D0 + 0.25 * OUTCON (1)) *
     1         DATA (1, 1) / 2.D0 * VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C     PERIOD
C
      TOTOUT = TOTOUT +
     1         ((0.75 * OUTCON (1) + 0.25 * 0.D0) *
     2         (DATA (1, 1) - 0.D0) / 2.D0 +
     3         (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
     4         (DATA (2, 1) - DATA (1, 1)) / 2.D0) *
     5         VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
C     PERIOD
C
      DO 110 I = 2, (NOUT - 1)
C
C     RESET VELOCITY IF DESORPTION BEGINS
```

```
C
         IF (DATA (I, 3) .LT. 1.0E-10) THEN
            VEL = VEL2
            IFLAG2 = 1
         ENDIF
C
         TOTOUT = TOTOUT +
     1            ((0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
     2            (DATA (I, 1) - DATA (I - 1, 1)) / 2.D0 +
     3            (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
     4            (DATA (I + 1, 1) - DATA (I, 1)) / 2.D0) *
     5            VEL * THETA
 110  CONTINUE
C
C     CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C     REPORT PERIOD
C
      TOTOUT = TOTOUT +
     1         (0.75 * OUTCON (NOUT) +
     2         0.25 * OUTCON (NOUT - 1)) *
     3         (DATA (NOUT, 1) - DATA (NOUT - 1, 1)) / 2.D0 *
     4         VEL * THETA
C
C
C     RESTORE VELOCITY
C
      VEL = VEL1
C
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
C
      RCOL = REAL (NCOL)
      TOTWET = (0.75 * DV (1) + 0.25 * DV (3)) *
     1         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C     CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE
C
      DO 120 I = 3, (2 * NCOL - 3), 2
         TOTWET = TOTWET +
     1            (0.75 * DV (I) + 0.125 * DV (I - 2) +
     2            0.125 * DV (I * 2)) *
     3            ZCOL / (RCOL - 1.D0) * THETA
 120  CONTINUE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
C
      TOTWET = TOTWET + (0.25 * DV (2 * NCOL - 3) +
     1         0.75 * DV (2 * NCOL - 1)) *
     2         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE (SLOW)
C
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE (SLOW)
C
      TOTDRY = 0.D0 * 0.75 + DV (4) * 0.25
C
C     CALCULATE CONAMINANT ON TYPICAL NODES (SLOW)
C
      DO 130 I = 4, (2 * NCOL - 2), 2
         TOTDRY = TOTDRY +
     1            (0.75 * DV (I) + 0.125 * DV (I - 2)
     2            + 0.125 * DV (I + 2)) *
```

```
     3            ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)
 130  CONTINUE
C
C     CALCULATE CONTAMINANT NEAR EXIT (SLOW)
C
      TOTDRY = TOTDRY +
     1         (0.25 * DV (2 * NCOL - 2) + 0.75 * DV (2 * NCOL))
     2         * ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO -THETA)
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE (FAST SITES)
C
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE (FAST)
C
      TOTDRY = TOTDRY + FKF * (0.75 * DV (1) + 0.25 * DV (3)) ** FNF *
     1         ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO - THETA)
C
C     CALCULATE CONAMINANT ON TYPICAL NODES (FAST)
C
      DO 140 I = 3, (2 * NCOL - 3), 2
         TOTDRY = TOTDRY + FKF *
     1            (0.75 * DV (I) + 0.125 * DV (I - 2)
     2            + 0.125 * DV (I + 2)) ** FNF *
     3            ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)
 140  CONTINUE
C
C     CALCULATE CONTAMINANT NEAR EXIT (FAST)
C
      TOTDRY = TOTDRY +
     1         FKF * (0.25 * DV (2 * NCOL - 3)
     2         + 0.75 * DV (2 * NCOL - 1)) ** FNF
     3         * ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO -THETA)
C
C
C.....COMPUTE MASS BALANCE
C
C
      BAL = TOTINF / ( TOTOUT + TOTWET + TOTDRY )
C
C
C.....OUTPUT THE MASS BALANCE FINDINGS
C
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TOTINF
      WRITE (IOOUT, 1300) TOTOUT
      WRITE (IOOUT, 1400) TOTWET
      WRITE (IOOUT, 1500) TOTDRY
      WRITE (IOOUT, 1600) BAL
      IF (IFLAG1 .EQ. 1) THEN
         WRITE (IOOUT, 1700)
      ENDIF
      IF (IFLAG2 .EQ. 1) THEN
         WRITE (IOOUT, 1800)
      ENDIF
C
C
 1000 FORMAT ('1', 35X, 'EXPERIMENTAL MASS BALANCE REPORT')
 1100 FORMAT ('0-----------------------------------------------------',
     1         '---------------------------------------------------',
     2         '---------------------------------')
 1200 FORMAT (' ', 20X, 'TOTAL MASS INJECTED = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1300 FORMAT (' ', 20X, 'TOTAL MASS RELEASED = ',
     1        E10.4, ' GRAMS / SQ. CM ')
```

```
 1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = ',
     1        E10.4)
 1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
     1        ' IN INFLUENT CONCENTRATION.')
 1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
     1        ' DURING DESORPITON.')
C
C
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE          C
C     COMPLETION OF SIMULATION IN SIMULATION MODE.                     C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION               DESCRIPTION            C
C     -------------     ---------   ---------------------------------  C
C     I                 INTEGER*4   LOOP INDEX                         C
C     IFLAG1            INTEGER*4   FLAG FOR ABRUPT CONCENTRATION      C
C                                   CHANGE                             C
C     IFLAG2            INTEGER*4   FLAG FOR DESORPTION                C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION     C
C     NSTEPS            INTEGER*4   NUMBER OF TEMPORAL STEPS           C
C                                                                      C
C                                                                      C
C                                                                      C
C     BAL               REAL*8      RATIO OF MASS IN TO MASS ACCOUNTED C
C     CO                REAL*8      INITIAL CONCENTRATION              C
C     DV                REAL*8      DEPENDENT VARIABLE VECTOR          C
C     OUTCON            REAL*8      EXIT CONCENTRATION VECTOR          C
C     OUTPUT            REAL*4      SIMULATION STATUS MATRIX           C
C                                   COL 1 GIVES REPORT TIMES           C
C                                   COL 2 GIVES BED VOLUMES            C
C                                   COL 3 GIVES THROUGHPUT             C
C                                   COL 4 GIVES INFLUENT CONCENTRATION C
C                                   COL 5 GIVES DCO/DT                 C
C     RCOL              REAL*8      REAL EQUIVALENT OF NCOL            C
C     THETA             REAL*8      POROSITY                           C
C     TOTDRY            REAL*8      TOTAL MASS LEFT ON SOLID PHASE     C
C     TOTINF            REAL*8      TOTAL MASS INJECTED INTO COLUMN    C
C     TOTOUT            REAL*8      TOTAL MASS TO EXIT COLUMN          C
C     TOTWET            REAL*8      TOTAL MASS LEFT IN LIQUID PHASE    C
C     VEL               REAL*8      PORE VELOCITY                      C
C     VEL2              REAL*8      DESORPTION PORE VELOCITY           C
C     ZCOL              REAL*8      LENGTH OF COLUMN (Z DIRECTION)     C
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
```

```
      SUBROUTINE MASS2 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
     1            RHO, OUTPUT, NSTEPS , NCOL)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 OUTPUT
      DIMENSION OUTPUT (250, 5), DV (101), OUTCON (250)
      COMMON /UNITS/ IOIN, IOOUT
C
      IFLAG1 = 0
      IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
      VEL1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
C
C     CALCULATE  CONTAMINANT INJECTED DURING FIRST
C     REPORT PERIOD.
C
          TOTINF = CO * OUTPUT (1, 1) * VEL * THETA +
     1            0.5 * (OUTPUT (1, 4) - CO) * OUTPUT (1, 1) *
     2            VEL * THETA
C
C     CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
C
      DO 100 I = 2, (NSTEPS - 1)
         IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     1       (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
             IFLAG1 = 1
             TOTINF = TOTINF +
     1                OUTPUT (I, 4) *
     2                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     3                VEL * THETA
         ELSE
             TOTINF = TOTINF +
     1                OUTPUT (I - 1, 4) *
     2                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     3                VEL * THETA +
     4                0.5 * (OUTPUT (I, 4) - OUTPUT (I - 1, 4)) *
     5                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     6                VEL * THETA
         ENDIF
 100  CONTINUE
C
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
C     CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
C     THE FIRST REPORT PERIOD
C
      TOTOUT = (0.75 * 0.D0 + 0.25 * OUTCON (1)) *
     1         OUTPUT (1, 1) / 2.D0 * VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C     PERIOD
C
      TOTOUT = TOTOUT +
     1         ((0.75 * OUTCON (1) + 0.25 * 0.D0) *
     2         (OUTPUT (1, 1) - 0.D0) / 2.D0 +
     3         (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
     4         (OUTPUT (2, 1) - OUTPUT (1, 1)) / 2.D0) *
     5         VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
C     PERIOD
```

```
C
      DO 110 I = 2, (NSTEPS - 1)
C
C     RESET VELOCITY IF DESORPTION BEGINS
C
         IF (OUTPUT (I, 4) .LT. 1.0E-10) THEN
            VEL = VEL2
            IFLAG2 = 1
         ENDIF
C
         TOTOUT = TOTOUT +
     1            ((0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
     2            (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) / 2.D0 +
     3            (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
     4            (OUTPUT (I + 1, 1) - OUTPUT (I, 1)) / 2.D0) *
     5            VEL * THETA
 110  CONTINUE
C
C     CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C     REPORT PERIOD
C
      TOTOUT = TOTOUT +
     1         (0.75 * OUTCON (NSTEPS) +
     2         0.25 * OUTCON (NSTEPS - 1)) *
     3         (OUTPUT (NSTEPS, 1) - OUTPUT (NSTEPS - 1, 1)) / 2.D0 *
     4         VEL * THETA
C
C     RESTORE VELOCITY
C
      VEL = VEL1
C
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
C
      RCOL = REAL (NCOL)
      TOTWET = (0.75 * DV (1) + 0.25 * DV (3)) *
     1         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C     CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE
C
      DO 120 I = 3, (2 * NCOL - 3), 2
         TOTWET = TOTWET +
     1            (0.75 * DV (I) + 0.125 * DV (I - 2) +
     2            0.125 * DV (I + 2)) *
     3            ZCOL / (RCOL - 1.D0) * THETA
 120  CONTINUE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
C
      TOTWET = TOTWET + (0.25 * DV (2 * NCOL - 3) +
     1         0.75 * DV (2 * NCOL - 1)) *
     2         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE (SLOW SITES)
C
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE (SLOW)
C
      TOTDRY = 0.D0 * 0.75 + DV (4) * 0.25
C
C     CALCULATE CONAMINANT ON TYPICAL NODES (SLOW)
C
      DO 130 I = 4, (2 * NCOL - 2), 2
```

```
         TOTDRY = TOTDRY +
     1            (0.75 * DV (I) + 0.125 * DV (I - 2)
     2            + 0.125 * DV (I + 2)) *
     3            ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)
 130  CONTINUE
C
C     CALCULATE CONTAMINANT NEAR EXIT (SLOW)
C
      TOTDRY = TOTDRY +
     1         (0.25 * DV (2 * NCOL - 2) + 0.75 * DV (2 * NCOL))
     2         * ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO -THETA)
C
C
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE (FAST SITES)
C
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE (FAST)
C
      TOTDRY = TOTDRY + FKF * (0.75 * DV (1) + 0.25 * DV (3)) ** FNF *
     1         ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO - THETA)
C
C     CALCULATE CONAMINANT ON TYPICAL NODES (FAST)
C
      DO 140 I = 3, (2 * NCOL - 3), 2
         TOTDRY = TOTDRY + FKF *
     1            (0.75 * DV (I) + 0.125 * DV (I - 2)
     2            + 0.125 * DV (I + 2)) ** FNF *
     3            ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)
 140  CONTINUE
C
C     CALCULATE CONTAMINANT NEAR EXIT (FAST)
C
      TOTDRY = TOTDRY +
     1         FKF * (0.25 * DV (2 * NCOL - 3)
     2         + 0.75 * DV (2 * NCOL - 1)) ** FNF
     3         * ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO -THETA)
C
C.....COMPUTE MASS BALANCE
C
C
      BAL = TOTINF / ( TOTOUT + TOTWET + TOTDRY )
C
C
C.....OUTPUT THE MASS BALANCE FINDINGS
C
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TOTINF
      WRITE (IOOUT, 1300) TOTOUT
      WRITE (IOOUT, 1400) TOTWET
      WRITE (IOOUT, 1500) TOTDRY
      WRITE (IOOUT, 1600) BAL
      IF (IFLAG1 .EQ. 1) THEN
         WRITE (IOOUT, 1700)
      ENDIF
      IF (IFLAG2 .EQ. 1) THEN
         WRITE (IOOUT, 1800)
      ENDIF
C
C
 1000 FORMAT ('1', 35X, 'SIMULATION MASS BALANCE REPORT ')
 1100 FORMAT ('0--------------------------------------------------',
     1        '---------------------------------------------------',
     2        '---------------------------------')
```

```
 1200 FORMAT (' ', 20X, 'TuTAL MASS INJECTED = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1300 FORMAT (' ', 20X, 'TOTAL MASS RELEASED = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ',
     1       E10.4, ' GRAMS / SQ. CM ')
 1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = ',
     1       E10.4)
 1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
     1        ' IN INFLUENT CONCENTRATION.')
 1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
     1        ' DURING DESORPITON.')
C
C
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C                     SORPTION MODEL:                                     C
C                     DUAL RESISTANCE                                     C
C                    MAINFRAME VERSION                                    C
C                                                                         C
C                                                                         C
C     THIS PROGRAM USES THE DGEAR METHOD TO SOLVE THE ADVECTIVE           C
C     DISPERSIVE EQUATION FOR THE CASE OF DUAL-RESISTANCE SORPTION.       C
C     THE TWO RESISTANCES SIMULATED INCLUDE EXTERNAL FILM TRANSPORT AND   C
C     INTERNAL SOLID-PHASE SURFACE DIFFUSION. THE EQUILIBRIUM MODEL       C
C     USED IS THE SINGLE COMPONENT FREUNDLICH ISOTHERM. THE EQUATIONS     C
C     ARE SPATIALLY APPROXIMATED USING THE FINITE DIFFERENCE METHOD       C
C     RESULTING IN A SET OF DC/DT AND DQ/DT EQUATIONS. THESE EQUATIONS    C
C     ARE THEN SOLVED SIMULTANEOUSLY USING DGEAR---AN IMSL SUBROUTINE.    C
C                                                                         C
C     THE LIMITS OF THIS PROGRAM ARE THREE:                               C
C          1) NUMBER OF COLUMN NODES =  21                                C
C          2) NUMBER OF RADIAL NODES =  21                                C
C          3) NUMBER OF DATA POINTS  = 100                                C
C                                                                         C
C     THESE LIMITS ARE IMPOSED CHIEFLY BECAUSE OF THE RESULTANT SIZE      C
C     OF WORKING ARRAY FOR DGEAR (WK). THESE LIMITS MAY BE RELAXED        C
C     AT THE EXPENSE OF INCREASED SIMULATION TIME BY USING THE DIAGONALLY C
C     DOMINANT ASSUMPTION FOR CALLS TO DGEAR (I.E. MITER = 3).            C
C     MUCH LARGER PROBLEMS CAN BE SIMULTAED USING THE DIAGONAL DOMINANT   C
C     FORM, AND THE ONLY CHANGES REQUIRED ARE TO THE ARRAYS DV, DDVDT,    C
C     RADGRP, AND WK.                                                     C
C                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                 DESCRIPTION            C
C     -------------      ---------    ----------------------------------  C
C     I                  INTEGER*4    TIME STEP INDEX                     C
C     IDUMMY             INTEGER*4    DUMMY WORKING ARRAY FOR DGEAR       C
C     IER                INTEGER*4    ERROR CODE RETURN INDEX FOR DGEAR   C
C     INDEX              INTEGER*4    CALL INDICATOR INDEX FOR GEAR       C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM            C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO             C
C     IWK                INTEGER*4    WORKING ARRAY FOR DGEAR             C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION      C
C     NDOF               INTEGER*4    NUMBER OF DEGREES OF FREEDOM        C
C     NLC                INTEGER*4    NUMBER OF LOWER CODIAGONALS-DGEAR   C
C     NRAD               INTEGER*4    NUMBER OF RADIAL STEPS              C
C     NOUT               INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT  C
C     NUC                INTEGER*4    NUMBER OF UPPER CODIAGONALS-DGEAR   C
C                                                                         C
C                                                                         C
C     CO                 REAL*8       INITIAL CONCENTRATION               C
C     CPRIME             REAL*8       EXTERNAL NAME OF DC/DT SUBROUTINE   C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA             C
C                                     COLUMN 1 CONTAINS REPORT TIMES      C
C                                     COLUMN 2 GIVES EXPERIMENTAL         C
C                                     CONCENTRATION                       C
C                                     COLUMN 3 GIVES INFLUENT             C
C                                     CONCENTRATION                       C
C                                     COLUMN 4 GIVES DCO/DT               C
C     DDVDT              REAL*8       DERIVATIVE OF DEPENDENT VARIABLES   C
```

```
C     DGN               REAL*8      DIAGONAL VARIABLE GROUPING TERMS    C
C     DGNLOW            REAL*8      LOWER DIAGONAL VARAIBLE GROUPINGS   C
C     DGNUP             REAL*8      UPPER DIAGONAL VARIABLE GROUPINGS   C
C     DH                REAL*8      HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DH1               REAL*8      SORPTION PHASE HYDRODYNAMIC         C
C                                   DISPERSION COEFFICIENT              C
C     DS                REAL*8      SURFACE DIFFUSION COEFFICIENT       C
C     DUMMY             REAL*8      DUMMY ROUTINE REQUIRED BY DGEAR     C
C     DV                REAL*8      DEPENDENT VARIABLE VECTOR           C
C     EPSLON            REAL*8      ALLOWABLE ERROR FOR DGEAR           C
C     FCNJ              REAL*8      EXTERNAL NAME OF JACOBIAN ROUTINE   C
C     FILMK             REAL*8      FILM TRANSFER COEFFICIENT           C
C     FK                REAL*8      FREUNDLICH ISOTHERM COEFFICIENT     C
C     FLUXCF            REAL*8      MACROSCOPIC FLUX TERM GROUPING      C
C     FN                REAL*8      FREUNDLICH ISOTHERM EXPONENT        C
C     H                 REAL*8      STEP SIZE FOR DGEAR                 C
C     OUTPUT            REAL*4      MATRIX OF SIMULATION STATUS         C
C                                   COL 1 CONTAINS REPORT TIME          C
C                                   COL 2 CONTAINS BED VOLUMES          C
C                                   COL 3 CONTAINS THROUGHPUT           C
C                                   COL 4 GIVES INFLUENT CONCENTRATION  C
C                                   COL 5 GIVES DCO/DT                  C
C     RADGRP            REAL*8      SPHERICAL VARIABLE GROUPINGS        C
C     RADIN             REAL*8      SPHERICAL INTERNAL BOUNDARY GROUP   C
C     RADIUS            REAL*8      RADIUS OF SPHERICAL PARTICLE        C
C     RADOUT            REAL*8      SPHERICAL EXTERNAL BOUNDARY GROUP   C
C     RDSQSM            REAL*8      SUM OF RESIDUALS SQUARED            C
C     RHO               REAL*8      SOLID PHASE PARTICLE DENSITY        C
C     RSDSUM            REAL*8      SUM OF RESIDUALS                    C
C     SDUMMY            REAL*8      DUMMY ROUTINE FOR DGEAR             C
C     THETA             REAL*8      POROSITY                            C
C     TIME              REAL*8      TIME OF OUTPUT SIMULATION POINT     C
C     TO                REAL*8      INITIAL TIME FOR DGEAR              C
C     VEL               REAL*8      PORE VELOCITY                       C
C     VEL1              REAL*8      SORPTION PORE VELOCITY              C
C     VEL2              REAL*8      DESORPTION PORE VELOCITY            C
C     WK                REAL*8      WORKING ARRAY NEEDED BY DGEAR       C
C     ZCOL              REAL*8      LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                       C
C                                                                       C
C     TITLE             CHARACTER*72 TITLE OF SIMULATION RUN            C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      PROGRAM DUAL
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 SDUMMY, DATA, OUTPUT
      CHARACTER*72 TITLE
      DIMENSION IWK (462)
      DIMENSION F (100), X(1)
      DIMENSION DATA (100,4), DV (462), DDVDT (462), OUTPUT (250,5),
     1          RADGRP (21,3),  WK (36498), C (100), T (100),
     2          OUTCON (250)
      COMMON /DBAND/ NLC, NUC
      COMMON /GEAR/ DUMMY(48), SDUMMY(4), IDUMMY(38)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, FLUXCF, RADGRP, RADOUT, RADIN
      COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, RADIUS, DS, FILMK,
     2               FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT,
     3               NSTEPS, ICON
      COMMON /WORDS/ TITLE
      COMMON /FLUID/ DCODT
```

```
      COMMON /UNITS/ IOIN, IOOUT
      EXTERNAL CPRIME, FCNJ
      IOIN = 1
      IOOUT = 3
      CALL READ
      VEL1 = VEL
      DH1 = DH
      CALL ECHO
      CALL ERRSET (208, 256, -1, 1)
C
C.....COMPUTE CONCENTRATION AT EXPERIMENTAL POINTS IF DESIRED
C
      IF (NOUT .GT. 0) THEN
         CALL INITL (DDVDT, DV, RADGRP, OUTPUT, T0)
         CALL GRPFRM
         CALL INFO1 (DATA, NOUT)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
         T0 = 0.D00
         H = 1.D-05
         INDEX = 1
         DV (1) = CD
         NDOF = NCOL * (NRAD + 1)
         NDV = NCOL + (NCOL - 1) * NRAD
         NLC = NRAD + 1
         NUC = NRAD + 1
C
C.....TIME LOOP FOR SIMULTATION AND OUTPUT AT EXPERIMENTAL POINTS
C
         DO 100 I = 1,NOUT
            IF (I .GT. 1) THEN
               IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
     1            (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
                   H = 1.D-08
                   INDEX = 1
                   DCODT = 0.000
                   DV (1) = DATA (I, 3)
                   WRITE (IOOUT, 1300)
                   IF (DV (1) .LT. 1.0E-10) THEN
                      VEL = VEL2
                      DH = DH1 * VEL2 / VEL1
                      CALL GRPFRM
                      WRITE (IOOUT, 1400)
                   ENDIF
               ELSE
                   DCODT = DATA (I, 4)
               ENDIF
            ELSE
               DCODT = DATA (I, 4)
            ENDIF
            TIME = DATA (I,1)
            CALL DGEAR (NDOF, CPRIME, FCNJ, T0, H, DV, TIME,
     1                  EPSLON, 2, -2, INDEX, IWK, WK, IER)
            CALL REPORT (I, DV, RSDSUM, RDSQSM, RSD)
            OUTCON (I) = DV (NDV)
  100    F(I) = RSD
C
C.....RESTORE ORIGINAL PARAMETERS
C
         DH = DH1
         VEL = VEL1
C
C.....CHECK THE MASS BALANCE
C
         CALL MASS1 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2
```

```
     1            RHO, DATA, NOUT, NCOL, NRAD)
C
      ENDIF
C
C.....COMPUTE OUTPUT AT COMPUTED OUTPUT POINTS IF DESIRED
C
      IF (NSTEPS .GT. 0) THEN
         CALL INITL (DDVDT, DV, RADGRP, OUTPUT, T0)
         CALL GRPFRM
         CALL INFO2 (OUTPUT, NSTEPS)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
         T0 = 0.D00
         H = 1.D-05
         INDEX = 1
         DV (1) = C0
         NDOF = NCOL * (NRAD + 1)
         NDV = NCOL + (NCOL - 1) * NRAD
         NLC = NRAD + 1
         NUC = NRAD + 1
C
C.....TIME LOOP FOR SIMULTATION AND OUTPUT AT COMPUTED POINTS
C
         WRITE (IOOUT, 1000)
         WRITE (IOOUT, 1100)
         DO 110 I = 1,NSTEPS
            IF (I .GT. 1) THEN
               IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     1             (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
                   H = 1.D-08
                   INDEX = 1
                   DCODT = 0.000
                   DV (1) = OUTPUT (I, 4)
                   WRITE (IOOUT, 1300)
                   IF (DV (1) .LT. 1.0E-10) THEN
                      VEL = VEL2
                      DH = DH1 * VEL2 / VEL1
                      CALL GRPFRM
                      WRITE (IOOUT, 1400)
                   ENDIF
               ELSE
                   DCODT = OUTPUT (I, 5)
               ENDIF
            ELSE
               DCODT = OUTPUT (I, 5)
            ENDIF
            TIME = OUTPUT (I,1)
            CALL DGEAR (NDOF, CPRIME, FCNJ, T0, H, DV, TIME,
     1                  EPSLON, 2, -2, INDEX, IWK, WK, IER)
            CTEMP = DV (NDV)
            CNORM = CTEMP / C0
            OUTCON (I) = DV (NDV)
  110    WRITE (IOOUT, 1200) (OUTPUT (I,J), J = 1, 3), CTEMP, CNORM
C
C.....RESTORE ORIGINAL PARAMETERS
C
         DH = DH1
         VEL = VEL1
C
C.....CHECK THE MASS BALANCE
C
         CALL MASS2 (OUTCON, DV, C0, ZCOL, THETA, VEL, VEL2,
     1            RHO, OUTPUT, NSTEPS, NCOL, NRAD)
C
      ENDIF
```

```
 1000 FORMAT ('1', 42X, 'SIMULATION OUTPUT FOR PLOTTING')
 1100 FORMAT ('0-----------------------------------------------',
     1         '-----------------------------------------------',
     2         '-----------------------------')
 1200 FORMAT (5E12.5)
 1300 FORMAT (' ', 20X, '***** ', 'ABRUPT INFLUENT CONCENTRATION ',
     1         'CHANGE. ', '***** ', 'DGEAR WAS RESET. ', '*****')
 1400 FORMAT (' ', 23X, '***** ', 'DESORPTION DETECTED. ',
     1         '***** ', 'HYDRODYNAMICS WERE RESET. ', '*****')
      STOP
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                        C
C                                                                        C
C     THIS SUBROUTINE READS THE BASIC INPUT REQUIRED FOR THE DUAL-       C
C     RESISTANCE SORPTION VERSION OF THE ADVECTIVE-DISPERSIVE EQUATION.  C
C                                                                        C
C                                                                        C
C     VARIABLE MAP:                                                      C
C     ------------                                                       C
C                                                                        C
C     VARIABLE NAME     PRECISION                DESCRIPTION             C
C     -------------     ---------    ----------------------------------  C
C     IOIN              INTEGER*4    UNIT NUMBER TO READ FROM            C
C     IOOUT             INTEGER*4    UNIT NUMBER TO WRITE TO             C
C     NCOL              INTEGER*4    NUMBER OF NODES IN Z DIRECTION      C
C     NRAD              INTEGER*4    NUMBER OF RADIAL STEPS              C
C     NOUT              INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT  C
C                                                                        C
C                                                                        C
C     CO                REAL*8       INITIAL CONCENTRATION               C
C     DATA              REAL*4       EXPERIMENTAL INPUT DATA             C
C                                    COLUMN 1 CONTAINS REPORT TIMES      C
C                                    COLUMN 2 GIVES EXPERIMENTAL         C
C                                    CONCENTRATION                       C
C                                    COLUMN 3 GIVES INFLUENT             C
C                                    CONCENTRATION                       C
C                                    COLUMN 4 GIVES DCO/DT               C
C     DH                REAL*8       HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DS                REAL*8       SURFACE DIFFUSION COEFFICIENT       C
C     EPSLON            REAL*8       ALLOWABLE ERROR FOR DGEAR           C
C     FILMK             REAL*8       FILM TRANSFER COEFFICIENT           C
C     FK                REAL*8       FREUNDLICH ISOTHERM COEFFICIENT     C
C     FN                REAL*8       FREUNDLICH ISOTHERM EXPONENT        C
C     RADIUS            REAL*8       RADIUS OF SPHERICAL PARTICLE        C
C     RHO               REAL*8       SOLID PHASE PARTICLE DENSITY        C
C     THETA             REAL*8       POROSITY                            C
C     VEL               REAL*8       PORE VELOCITY                       C
C     VEL2              REAL*8       DESORPTION PORE VELOCITY            C
C     ZCOL              REAL*8       LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                        C
C                                                                        C
C                                                                        C
C                                                                        C
C     TITLE             CHARACTER*72 TITLE OF SIMULATION RUN             C
C                                                                        C
C                                                                        C
C                                                                        C
C                                                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE READ
      IMPLICIT REAL*8 (A-H, O-Z)
```

```
      REAL*4 DATA
      CHARACTER*72 TITLE
      DIMENSION DATA (100,4), C (100), T (100)
      COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, RADIUS, DS, FILMK,
     2               FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT,
     3               NSTEPS, ICON
      COMMON /WORDS/ TITLE
      COMMON /UNITS/ IOIN, IOOUT
      READ (IOIN, 1000) TITLE
      READ (IOIN, 1100) NCOL, NRAD, NOUT, NSTEPS, TMESIM, ICON
      IF (ABS (NOUT) .GT. 0) THEN
         NTEMP = ABS (NOUT)
         READ (IOIN, 1200) (DATA (I,1), I = 1, NTEMP)
         READ (IOIN, 1200) (DATA (I,2), I = 1, NTEMP)
      ENDIF
      READ (IOIN, 1200) VEL, DH, RHO, THETA, ZCOL, EPSLON
      READ (IOIN, 1200) CO, FK, FN, RADIUS, DS, FILMK
      READ (IOIN, 1200) VEL2
      READ (IOIN, 1200) (C (I), I = 1, ICON)
      READ (IOIN, 1200) (T (I), I = 1, ICON)
 1000 FORMAT (A72)
 1100 FORMAT (4I5,E12.5,I5)
 1200 FORMAT ((6E12.5))
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                           C
C     THIS SUBROUTINE ECHOS THE INPUT TO DEVICE IOOUT.                      C
C                                                                           C
C     VARIABLE MAP:                                                         C
C     -------------                                                         C
C                                                                           C
C     VARIABLE NAME      PRECISION                  DESCRIPTION             C
C     -------------      ---------    -----------------------------------   C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM              C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO               C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION        C
C     NRAD               INTEGER*4    NUMBER OF RADIAL STEPS                C
C                                                                           C
C                                                                           C
C     CO                 REAL*8       INITIAL CONCENTRATION                 C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA               C
C                                     COLUMN 1 CONTAINS REPORT TIMES        C
C                                     COLUMN 2 GIVES EXPERIMENTAL           C
C                                     CONCENTRATION                         C
C                                     COLUMN 3 GIVES INFLUENT               C
C                                     CONCENTRATION                         C
C                                     COLUMN 4 GIVES DCO/DT                 C
C     DH                 REAL*8       HYDRODYNAMIC DISPERSION COEFFICIENT   C
C     DS                 REAL*8       SURFACE DIFFUSION COEFFICIENT         C
C     EPSLON             REAL*8       ALLOWABLE ERROR FOR DGEAR             C
C     FILMK              REAL*8       FILM TRANSFER COEFFICIENT             C
C     FK                 REAL*8       FREUNDLICH ISOTHERM COEFFICIENT       C
C     FN                 REAL*8       FREUNDLICH ISOTHERM EXPONENT          C
C     PE                 REAL*8       SYSTEM PECLET NUMBER                  C
C     RADIUS             REAL*8       RADIUS OF SPHERICAL PARTICLE          C
C     RHO                REAL*8       SOLID PHASE PARTICLE DENSITY          C
C     THETA              REAL*8       POROSITY                              C
C     VEL                REAL*8       PORE VELOCITY                         C
C     VEL2               REAL*8       DESORPTION PORE VELOCITY              C
C     ZCOL               REAL*8       LENGTH OF COLUMN (Z DIRECTION)        C
C                                                                           C
C
```

```
C                                                                            C
C     TITLE               CHARACTER*72 TITLE OF SIMULATION RUN               C
C                                                                            C
C                                                                            C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE ECHO
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      CHARACTER*72 TITLE
      DIMENSION DATA (100,4), C (100), T (100)
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, RADIUS, DS, FILMK,
     2               FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT,
     3               NSTEPS, ICON
      COMMON /WORDS/ TITLE
      COMMON /UNITS/ IOIN, IOOUT
      PE = VEL * ZCOL / DH
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TITLE
      WRITE (IOOUT, 1400)
      WRITE (IOOUT, 1300)
      WRITE (IOOUT, 1400)
      WRITE (IOOUT, 1500) CO
      WRITE (IOOUT, 1600) DH
      WRITE (IOOUT, 1700) NCOL
      WRITE (IOOUT, 1800) RHO
      WRITE (IOOUT, 1900) VEL
      WRITE (IOOUT, 1950) VEL2
      WRITE (IOOUT, 2000) THETA
      WRITE (IOOUT, 2100) ZCOL
      WRITE (IOOUT, 2200) PE
      WRITE (IOOUT, 2300) FK
      WRITE (IOOUT, 2400) FN
      WRITE (IOOUT, 2500) FN
      WRITE (IOOUT, 2600) FILMK
      WRITE (IOOUT, 2700) DS
      WRITE (IOOUT, 2800) RADIUS
      WRITE (IOOUT, 2900) NRAD
      WRITE (IOOUT, 3000) ICON
      WRITE (IOOUT, 3100)
      WRITE (IOOUT, 3200) (C (I), I = 1, ICON)
      WRITE (IOOUT, 3300)
      WRITE (IOOUT, 3200) (T (I), I = 1, ICON)
 1000 FORMAT ('1', 50X, 'PLUG FLOW DISPERSION REACTOR')
 1100 FORMAT (' ', 47X, 'DUAL RESISTANCE KINETIC SIMULATION')
 1200 FORMAT (' ', 26X, A72)
 1300 FORMAT (' ', 15X, 'VARIABLE', 16X, 'DESCRIPTION', 30X, '   VALUE',
     1        16X, 'UNITS   ', 16X)
 1400 FORMAT ('0------------------------------------------------',
     1        '-------------------------------------------------',
     2        '-------------------------------')
 1500 FORMAT ('0', 15X, 'CO    ', 18X, 'INITIAL CONCENTRATION     ', 14X,
     1        E10.4, 16X, 'G/CM**3')
 1600 FORMAT (' ', 15X, 'DH    ', 18X, 'HYDRODYNAMIC DISPERSION   ', 14X,
     1        E10.4, 16X, 'CM**2/HOUR')
 1700 FORMAT (' ', 15X, 'NCOL  ', 18X, 'NUMBER OF COLUMN STEPS    ', 16X,
     1        I5)
 1800 FORMAT (' ', 15X, 'RHO   ', 18X, 'SOLID PHASE DENSITY       ', 14X,
     1        E10.4, 16X, 'GR/CM**3')
 1900 FORMAT (' ', 15X, 'VEL   ', 18X, 'PORE VELOCITY             ', 14X,
     1        E10.4, 16X, 'CM/HOUR')
```

```
 1950 FORMAT (' ', 15X, 'VEL   ', 18X, 'DESORPTION VELOCITY           ', 14X,
     1       E10.4, 16X, 'CM/HOUR')
 2000 FORMAT (' ', 15X, 'THETA ', 18X, 'VOID VOLUME FRACTION          ', 14X,
     1       E10.4)
 2100 FORMAT (' ', 15X, 'ZCOL  ', 18X, 'LENGTH OF COLUMN              ', 14X,
     1       E10.4, 16X, 'CM')
 2200 FORMAT (' ', 15X, 'PE    ', 18X, 'PECLET NUMBER                 ', 14X,
     1       E10.4)
 2300 FORMAT (' ', 15X, 'FK    ', 18X, 'FREUNDLICH COEFFICIENT        ', 14X,
     1       E10.4, 16X, '((CM**3/GR)**FN)')
 2400 FORMAT (' ', 15X, 'FN    ', 18X, 'FREUNDLICH EXPONENT           ', 14X,
     1       E10.4)
 2500 FORMAT (' ', 15X, 'FN    ', 18X, 'FREUNDLICH DESORPTION EXPONENT',
     1       9X, E10.4)
 2600 FORMAT (' ', 15X, 'FILMK ', 18X, 'FILM TRANSPORT COEFFICIENT',
     1       13X, E10.4, 16X, 'CM/HR')
 2700 FORMAT (' ', 15X, 'DS    ', 18X, 'INTRAPARTICLE COEFFICIENT', 14X,
     1       E10.4, 16X, 'CM**2/HR')
 2800 FORMAT (' ', 15X, 'RADIUS', 18X, 'PARTICLE RADIUS               ', 14X,
     1       E10.4, 16X, 'CM')
 2900 FORMAT (' ', 15X, 'NRAD  ', 18X, 'PARTICLE STEPS                ', 16X,
     1       I5)
 3000 FORMAT (' ', 15X, 'ICON  ', 18X, 'NUMBER OF INFLUENTS           ', 16X,
     1       I5)
 3100 FORMAT (' ', 39X, 'INFLUENT CONCENTRATIONS')
 3200 FORMAT (' ', 14X, 10E10.3)
 3300 FORMAT (' ', 39X, 'INFLUENT TIMES')
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C                                                                       C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE DATA MATRIX TO         C
C     UNIT IOOUT.                                                       C
C                                                                       C
C                                                                       C
C     VARIABLE MAP:                                                     C
C     ------------                                                      C
C                                                                       C
C     VARIABLE NAME     PRECISION                 DESCRIPTION           C
C     -------------     ---------    ---------------------------------- C
C     IOIN              INTEGER*4    UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4    UNIT NUMBER TO WRITE TO            C
C     NCOUNT            INTEGER*4    LOOP COUNTER                       C
C     NOUT              INTEGER*4    NUMBER OF TEMPORAL STEPS TO OUTPUT C
C                                                                       C
C     DATA              REAL*4       EXPERIMENTAL INPUT DATA            C
C                                    COLUMN 1 CONTAINS REPORT TIMES     C
C                                    COLUMN 2 GIVES EXPERIMENTAL        C
C                                    CONCENTRATION                      C
C                                    COLUMN 3 GIVES INFLUENT            C
C                                    CONCENTRATION                      C
C                                    COLUMN 4 GIVES DCO/DT              C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INFO1 (DATA, NOUT)
      REAL*4 DATA
      DIMENSION DATA (100,4)
      COMMON /UNITS/ IOIN, IOOUT
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200)
```

```
      WRITE (IOOUT, 1100)
      DO 100 NCOUNT = 1, NOUT
          WRITE (IOOUT, 1300) ( DATA (NCOUNT, I), I = 1, 4)
 100  CONTINUE
 1000 FORMAT ('1', 42X, 'EXPERIMENTAL COMPARISON MODE INFORMATION')
 1100 FORMAT ('0----------------------------------------------------',
     1         '----------------------------------------------------',
     2         '---------------------------------')
 1200 FORMAT (' ', 28X, 'TIME',5X,'CONCENTRATION OUT', 6X,
     1         'CONCENTRATION IN', 16X, 'DCO/DT')
 1300 FORMAT (' ', 20X, E12.5, 10X, E12.5, 10X, E12.5, 10X, E12.5)
      RETURN
      END
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                         C
C                                                                         C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE OUTPUT MATRIX TO         C
C     UNIT IOOUT.                                                         C
C                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                 DESCRIPTION            C
C     -------------      ---------    ----------------------------------  C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM            C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO             C
C     NCOUNT             INTEGER*4    LOOP COUNTER                        C
C     NSTEPS             INTEGER*4    NUMBER OF EQUAL BED VOLUME STEPS    C
C                                     TO BE CALCULATED                    C
C                                                                         C
C     OUTPUT             REAL*4       A MATRIX OF SIMULATION STATUS       C
C                                     COL 1 CONTAINS REPORT TIME          C
C                                     COL 2 CONTAINS BED VOLUMES          C
C                                     COL 3 CONTAINS THROUGHPUT           C
C                                     COL 4 GIVES INFLUENT CONCENTRATION  C
C                                     COL 5 GIVES DCO/DT                  C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INFO2 (OUTPUT, NSTEPS)
      REAL*4 OUTPUT
      DIMENSION OUTPUT (250, 5)
      COMMON /UNITS/ IOIN, IOOUT
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200)
      WRITE (IOOUT, 1100)
      DO 100 NCOUNT = 1, NSTEPS
          WRITE (IOOUT, 1300) (OUTPUT (NCOUNT, I), I = 1, 5)
 100  CONTINUE
 1000 FORMAT ('1', 48X, 'SIMULATION MODE INFORMATION')
 1100 FORMAT ('0----------------------------------------------------',
     1         '----------------------------------------------------',
     2         '---------------------------------')
 1200 FORMAT (' ', 12X, 'REPORT TIME', 11X, 'BED VOLUMES', 12X,
     1        'THROUGHPUT', 6X, 'CONCENTRATION IN',
     2        16X, 'DCO/DT')
 1300 FORMAT (' ', 11X, E12.5, 10X, E12.5, 10X, E12.5, 10X,
     1        E12.5, 10X, E12.5)
      RETURN
```

```
      END
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C                                                                          C
C     THIS SUBROUTINE INITIALIZES THE MODEL VARIABLES FOR THE SIMULATION   C
C     OF THE ADVECTIVE-DISPERSIVE EQUATION.                                C
C                                                                          C
C                                                                          C
C     VARIABLE MAP:                                                        C
C     ------------                                                         C
C                                                                          C
C     VARIABLE NAME       PRECISION                 DESCRIPTION            C
C     -------------       ---------    ----------------------------------- C
C     I                   INTEGER*4    WORKING VARIABLE                    C
C     J                   INTEGER*4    WORKING VARIABLE                    C
C                                                                          C
C     DDVDT               REAL*8       DERIVATIVE OF DEPENDENT VARIABLES   C
C     DV                  REAL*8       DEPENDENT VARIABLE VECTOR           C
C     EPSMCN              REAL*8       MACHINE EPSILON                     C
C     OUTPUT              REAL*4       MATRIX OF SIMULATION STATUS         C
C                                      COL 1 CONTAINS REPORT TIME          C
C                                      COL 2 CONTAINS BED VOLUMES          C
C                                      COL 3 CONTAINS THROUGHPUT           C
C                                      COL 4 GIVES INFLUENT CONCENTRATION  C
C                                      COL 5 GIVES DCO/DT                  C
C     RADGRP              REAL*8       SPHERICAL VARIABLE GROUPINGS        C
C     TEMP                REAL*8       TEMPORARY VARIABLE FOR EPSMCN CALC  C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE INITL (DDVDT, DV, RADGRP, OUTPUT, T0)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA, OUTPUT
      LOGICAL DONE
      DIMENSION DATA (100,4), DDVDT (462), DV (462), OUTPUT (250,5),
     1          RADGRP (21,3), C (100), T (100)
      COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, RADIUS, DS, FILMK,
     2               FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT,
     3               NSTEPS, ICON
      DO 100 I = 1, 462
         DV (I) = 0.D00
  100 DDVDT (I) = 0.D00
      DO 110 I = 1, 21
         DO 110 J = 1, 3
  110 RADGRP (I,J) = 0.D00
C
C.....COMPUTE SIMULATION OUTPUT POINTS IF DESIRED
C
      IF (NSTEPS .GT. 0) THEN
         TAU = ZCOL / VEL
         DG = 1 + (RHO * (1 - THETA) / THETA) * (FK * (CO ** FN)) / CO
         DTN = NSTEPS
         DT = THESIM / DTN
         TEMPT = 0.D00
         DO 120 I = 1, NSTEPS
            TEMPT = TEMPT + DT
            OUTPUT (I,1) = TEMPT
```

```
            OUTPUT (I,2) = TEMPT / TAU
  120    OUTPUT (I,3) = TEMPT / (TAU * DG)
      ENDIF
C
C.....CALCULATE THE MACHINE EPSILON FOR USE AS A LOWER ERROR BOUND
C
      EPSMCN = 1.0
  130 EPSMCN = EPSMCN / 2.0D00
      TEMP = 1.0D00 + EPSMCN
      IF (TEMP .GT. 1.0D00) GO TO 130
C
C.....CALCULATE THE INFLUENT CONCENTRATIONS FOR USE IN THE
C.....EXPERIMENTAL COMPARISON MODE BY INTERPOLATING FROM THE
C.....LIST OF KNOWN CONCENTRATIONS.
C
      IF (NOUT .GT. 0) THEN
          DO 140 NCOUNT = 1, NOUT, 1
              TEND = DATA (NCOUNT, 1)
              DONE = .FALSE.
              DO 150 NLOOP = 1, (ICON - 1)
              IF (DONE .EQV. .FALSE.) THEN
                  DNTIME = T (NLOOP)
                  UPTIME = T (NLOOP + 1)
                  DNCON  = C (NLOOP)
                  UPCON  = C (NLOOP + 1)
                  IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND))
     1            THEN
                      DATA (NCOUNT, 3) = DNCON + (UPCON - DNCON)
     1                * (TEND - DNTIME) / (UPTIME - DNTIME)
                      DONE = .TRUE.
                  ENDIF
               ENDIF
 150           CONTINUE
               IF (NCOUNT .GT. 1) THEN
                   DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) -
     1             DATA (NCOUNT - 1, 3)) / (DATA (NCOUNT, 1) -
     2             DATA (NCOUNT - 1, 1))
               ELSE
                   DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) - C0)
     1             / (DATA ( NCOUNT, 1) - T0)
               ENDIF
 140       CONTINUE
      ENDIF
C
C.....CALCULATE INFLUENT CONCENTRATION FOR USE IN THE
C.....SIMULATION MODE.
C
      IF (NSTEPS .GT. 0) THEN
          DO 160 NCOUNT = 1, NSTEPS, 1
              TEND = OUTPUT(NCOUNT, 1)
              DONE = .FALSE.
              DO 170 NLOOP = 1, (ICON - 1)
              IF (DONE .EQV. .FALSE.) THEN
                  DNTIME = T (NLOOP)
                  UPTIME = T (NLOOP + 1)
                  DNCON  = C (NLOOP)
                  UPCON  = C (NLOOP + 1)
                  IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND))
     1            THEN
                      OUTPUT (NCOUNT, 4) = DNCON + (UPCON - DNCON)
     1                * (TEND - DNTIME) / (UPTIME - DNTIME)
                      DONE = .TRUE.
                  ENDIF
              ENDIF
 170          CONTINUE
              IF (NCOUNT .GT. 1) THEN
```

```
                 OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) -
     1           OUTPUT (NCOUNT - 1, 4)) / (OUTPUT (NCOUNT, 1) -
     2           OUTPUT (NCOUNT - 1, 1))
             ELSE
                 OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - CO)
     1           / (OUTPUT ( NCOUNT, 1) - TO)
             ENDIF
 160     CONTINUE
C
C.....COMPUTE BED VOLUMES FOR VARIABLE VELOCITY CASE
C
         OLDVEL = VEL
         TEMPT = 0.D00
         DO 180 I = 1, NSTEPS
             TEMPT = TEMPT + DT
             IF (OUTPUT (I, 4) . GT. 1.0E-10) THEN
                 CURVEL = VEL
             ELSE
                 CURVEL = VEL2
             ENDIF
             AVGVEL = (OLDVEL * (TEMPT - DT) + CURVEL * DT) / TEMPT
             TAU = ZCOL / AVGVEL
             OUTPUT (I, 2) = TEMPT / TAU
             OUTPUT (I, 3) = TEMPT / (TAU * DG)
             OLDVEL = AVGVEL
 180     CONTINUE
      ENDIF
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE COMPUTES VARIABLE GROUPINGS TO MINIMIZE EFFORT IN C
C     THE SOLUTION OF THE SET OF SIMULTANEOUS NONLINEAR EQUATIONS.     C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION               DESCRIPTION            C
C     -------------     ---------   ---------------------------------  C
C     I                 INTEGER*4   TIME STEP INDEX                    C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION     C
C     NDOF              INTEGER*4   NUMBER OF DEGREES OF FREEDOM       C
C     NRAD              INTEGER*4   NUMBER OF RADIAL STEPS             C
C     NOUT              INTEGER*4   NUMBER OF TEMPORAL STEPS TO OUTPUT C
C                                                                      C
C                                                                      C
C     DELTAR            REAL*8      DELTA RADIUS FOR SPHERICAL EQUATION C
C     DGN               REAL*8      DIAGONAL VARIABLE GROUPING TERMS   C
C     DGNLOW            REAL*8      LOWER DIAGONAL VARAIBLE GROUPINGS  C
C     DGNUP             REAL*8      UPPER DIAGONAL VARIABLE GROUPINGS  C
C     DH                REAL*8      HYDRODYNAMIC DISPERSION COEFFICIENT C
C     DS                REAL*8      SURFACE DIFFUSION COEFFICIENT      C
C     DZ                REAL*8      COLUMN STEP SIZE IN Z DIRECTION    C
C     FILMK             REAL*8      FILM TRANSFER COEFFICIENT          C
C     FLUXCF            REAL*8      MACROSCOPIC FLUX TERM GROUPING     C
C     GRPDGN            REAL*8      DIAGONAL GROUPING FOR SPHERICAL EQ C
C     GRPOFF            REAL*8      OFF DIAGONAL GROUP FOR SPHERICAL EQ C
C     RADGRP            REAL*8      SPHERICAL VARIABLE GROUPINGS       C
C     RADIN             REAL*8      SHPERICAL INTERNAL BOUNDARY GROUP  C
C     RADIUS            REAL*8      RADIUS OF SPHERICAL PARTICLE       C
C     RADND             REAL*8      NUMBER OF RADIAL NODES             C
C     RADOUT            REAL*8      SPHERICAL EXTERNAL BOUNDARY GROUP  C
```

```
C     RADTMP              REAL*8       TEMPORARY VALUE OF SPHERICAL RADIUS C
C     REALND              REAL*8       NUMBER OF NODES IN COLUMN           C
C     RHO                 REAL*8       SOLID PHASE PARTICLE DENSITY        C
C     THETA               REAL*8       POROSITY                            C
C     VEL                 REAL*8       PORE VELOCITY                       C
C     VEL2                REAL*8       DESORPTION PORE VELOCITY            C
C     ZCOL                REAL*8       LENGTH OF COLUMN (Z DIRECTION)      C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE GRPFRM
      IMPLICIT REAL*8(A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), RADGRP (21,3), C (100), T (100)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, FLUXCF, RADGRP, RADOUT, RADIN
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, RADIUS, DS, FILMK,
     2               FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT,
     3               NSTEPS, ICON
      NDOF = NCOL * (NRAD + 1)
C
C.....COMPUTE MACROSCOPIC ADVECTIVE-DSIPERSIVE EQUATION FLUID PHASE GROUPINS
C
      REALND = NCOL
      DZ = ZCOL / (REALND - 1.D00)
      DGNLOW = DH / (DZ ** 2.D00) + VEL / (2.D00 * DZ)
      DGN = -2.D00 * DH / (DZ ** 2.D00) - 3.D00 * FILMK * (1 - THETA)
     1      / (RADIUS * THETA)
      DGNUP = DH / (DZ ** 2.D00) - VEL / (2.D00 *DZ)
C
C.....COMPUTE SPHERICAL SOLID PHASE VARIABLE GROUPINGS
C
      FLUXCF = 3.D00 * FILMK * (1 - THETA) / (RADIUS * THETA)
      RADND = NRAD
      DELTAR = RADIUS / (RADND - 1.D00)
      GRPDGN = -2.D00 * DS / (DELTAR ** 2.D00)
      GRPOFF = DS / (DELTAR ** 2.D00)
      RADTMP = DELTAR
      DO 100 I = 2, NRAD
      RADGRP (I,1) = GRPOFF - DS / (RADTMP * DELTAR)
      RADGRP (I,2) = GRPDGN
      RADGRP (I,3) = GRPOFF + DS / (RADTMP * DELTAR)
  100 RADTMP = RADTMP + DELTAR
      RADIN = 6.D00 * DS / (DELTAR ** 2.D00)
      RADOUT = 2.D00 *  FILMK * DELTAR / (DS * RHO)
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C                                                                          C
C     THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE         C
C     DEPENDENT VARIABLE AS A FUNCTION OF TIME. THE EQUATIONS ARE          C
C     SOLVED USING THE VARIABLE GROUPINGS PREVIOUSLY DERIVED IN ORDER      C
C     TO SAVE COMPUTATIONAL EFFORT.                                        C
C                                                                          C
C     ALL DEPENDENT VARIABLE ARE STORED IN A SINGLE ARRAY DV, AND ALL      C
C     TEMPORAL DERIVATIVES OF THE DEPENDENT VARIABLE ARE STORED IN A       C
C     SINGLE ARRAY TERMED DDVDT. THE INDEX NUMBERING OF THESE ARRAYS       C
```

```
C     MAY BE DESCRIBED AS FOLLOWS:                                        C
C          A) THE FIRST ENTRY (1) IS THE INLET NODE OF THE COLUMN         C
C          B) THE SECOND ENTRY (2) IS THE INTERIOR NODE AT THE CENTER     C
C             OF THE SPHERICAL PARTICLE CORRESPONDING TO THE INLET        C
C             POSITION OF THE COLUMN                                      C
C          C) THE EXTERIOR BOUNDARY NODE OF THE SPHERICAL PARTICLE AT THE C
C             INLET OF COLUMN IS AT INDEX VALUE (NRAD + 1)                C
C          D) THE SECOND COLUMN NODE IS THEREFORE AT INDEX (NRAD + 2)     C
C          E) THE TOTAL DEGREES OF FREEDOM ARE EQUAL TO NCOL * (NRAD + 1) C
C          F) THE LAST INDEX ENTRY IS THE BOUNDARY NODE OF THE SOLID      C
C             PHASE SPHERE AT THE OUTLET POSITION OF THE COLUMN           C
C                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                 DESCRIPTION            C
C     -------------      ---------    ----------------------------------- C
C     I                  INTEGER*4    TEMPORARY VARAIBLE                  C
C     J                  INTEGER*4    TEMPORARY VARAIBLE                  C
C     K                  INTEGER*4    INDEX FOR SPHERICAL GROUPINGS       C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION      C
C     NDOF               INTEGER*4    NUMBER OF DEGREES OF FREEDOM        C
C     NEND               INTEGER*4    VARIABLE INDEX FOR LAST COLUMN NODE C
C     NRAD               INTEGER*4    NUMBER OF RADIAL STEPS              C
C     NSTART             INTEGER*4    LOOP STARTING POSITION              C
C     NSTEP              INTEGER*4    STEP SIZE FOR LOOP                  C
C     NSTOP              INTEGER*4    LOOP STOPPING POSITION              C
C                                                                         C
C                                                                         C
C     CS                 REAL*8       FLUID CONC AT EXTERIOR OF PARTICLE  C
C     DDVDT              REAL*8       DERIVATIVE OF DEPENDENT VARIABLES   C
C     DGN                REAL*8       DIAGONAL VARIABLE GROUPING TERMS    C
C     DGNLOW             REAL*8       LOWER DIAGONAL VARAIBLE GROUPINGS   C
C     DGNUP              REAL*8       UPPER DIAGONAL VARIABLE GROUPINGS   C
C     DV                 REAL*8       DEPENDENT VARIABLE VECTOR           C
C     FK                 REAL*8       FREUNDLICH ISOTHERM COEFFICIENT     C
C     FLUXCF             REAL*8       MACROSCOPIC FLUX TERM GROUPING      C
C     FN                 REAL*8       FREUNDLICH ISOTHERM EXPONENT        C
C     RADGRP             REAL*8       SPHERICAL VARIABLE GROUPINGS        C
C     RADIN              REAL*8       SHPERICAL INTERNAL BOUNDARY GROUP   C
C     RADOUT             REAL*8       SPHERICAL EXTERNAL BOUNDARY GROUP   C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE CPRIME (NVRB, TIME, DV, DDVDT)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), DV (462), DDVDT (462), RADGRP (21,3)
      DIMENSION C (100), T (100)
      COMMON /GROUPS/ DGN, DGNLOW, DGNUP, FLUXCF, RADGRP, RADOUT, RADIN
      COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, RADIUS, DS, FILMK,
     2               FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT,
     3               NSTEPS, ICON
      COMMON /FLUID/ DCODT
C
C.....FORM COLUMN STEP DERIVATIVES
C
      NDOF = NCOL * (NRAD + 1)
```

```
      NSTART = NRAD + 2
      NSTOP = NDOF - (2 * NRAD) - 1
      NEND = NDOF - NRAD
      NSTEP = NRAD + 1
      DDVDT (1) = DCODT
      DO 100 I = NSTART, NSTOP, NSTEP
         IF (DV (I+NRAD) .GE. 0.D00) THEN
            CS = ((DV (I+NRAD) / FK) ** (1.D00 / FN))
         ELSE
            CS = 0.D00
         ENDIF
  100 DDVDT (I) = DGNLOW * DV (I-NSTEP) + DGN * DV (I)
     1            + DGNUP * DV (I+NSTEP)
     2            + FLUXCF * CS
      IF (DV (NDOF) .GE. 0.D00) THEN
         CS = ((DV (NDOF) / FK) ** (1.D00 / FN))
      ELSE
         CS = 0.D00
      ENDIF
      DDVDT (NEND) = (DGNLOW + DGNUP) * DV (NEND-NSTEP)
     1               + DGN * DV (NEND)
     2               + FLUXCF * CS
C
C.....FORM THE SOLID PHASE DERIVATIVES
C
      DO 110 I = 1, NEND, NSTEP
         DDVDT (I+1) = RADIN * (DV (I+2) - DV (I+1))
         NSTART = I + 2
         NSTOP  = I + NRAD - 1
         DO 120 J = NSTART, NSTOP
            K = J - NSTART + 2
  120    DDVDT (J) = RADGRP (K,1) * DV (J-1) + RADGRP (K,2) * DV (J)
     1                  + RADGRP (K,3) * DV (J+1)
         J = I + NRAD
         IF (DV (J) .GE. 0.D00) THEN
            CS = ((DV (J) / FK) ** (1.D00 / FN))
         ELSE
            CS = 0.D00
         ENDIF
  110 DDVDT (J) = RADGRP (NRAD,1) * DV (J-1) + RADGRP (NRAD,2) * DV (J)
     1            + RADGRP (NRAD,3) * (DV (J-1) + RADOUT * (DV (I)
     2            - CS))
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE     C
C     DDVDT ARRAY AS A FUNCTION OF THE DEPENDENT VARIABLE (DV).        C
C     SINCE THE FINITE DIFFERENCE OPTION IS USED IN THE CALL TO DGEAR  C
C     (I.E. MITER= -2) THIS ROUTINE IS A BLANK.                        C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE FCNJ (N, X, Y, PD)
      REAL*8 Y(N), PD(N,N), X
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
```

```
C     THIS SUBROUTINE USES THE EXPERIMENTAL DATA INPUT AND THE MODEL      C
C     SIMULATION OUTPUT TO COMPUTE THE MODEL RESIDUAL, RESIDUAL SQUARED,  C
C     SUM OF RESIDUALS, AND SUM OF RESIDUALS SQUARED. THESE COMPUTATIONS  C
C     ARE OUTPUT TO UNIT IOOUT.                                           C
C                                                                         C
C                                                                         C
C     VARIABLE MAP:                                                       C
C     ------------                                                        C
C                                                                         C
C     VARIABLE NAME      PRECISION                DESCRIPTION             C
C     -------------      ---------    ----------------------------------  C
C     I                  INTEGER*4    TIME STEP INDEX                     C
C     IOIN               INTEGER*4    UNIT NUMBER TO READ FROM            C
C     IOOUT              INTEGER*4    UNIT NUMBER TO WRITE TO             C
C     NCOL               INTEGER*4    NUMBER OF NODES IN Z DIRECTION      C
C     NDOF               INTEGER*4    NUMBER OF DEGREES OF FREEDOM        C
C     NDV                INTEGER*4    INDEX OF OUTLET CONCENTRATION       C
C     NOUT               INTEGER*4    NUMBER OF POINTS SIMULATED          C
C     NRAD               INTEGER*4    NUMBER OF RADIAL STEPS              C
C                                                                         C
C                                                                         C
C     CNORM              REAL*8       NORMALIZED OUTLET CONCENTRATION     C
C     CO                 REAL*8       INITIAL CONCENTRATION               C
C     DATA               REAL*4       EXPERIMENTAL INPUT DATA             C
C                                     COLUMN 1 CONTAINS REPORT TIMES      C
C                                     COLUMN 2 GIVES EXPERIMENTAL         C
C                                     CONCENTRATION                       C
C                                     COLUMN 3 GIVES INFLUENT             C
C                                     CONCENTRATION                       C
C                                     COLUMN 4 GIVES DCO/DT               C
C     DV                 REAL*8       DEPENDENT VARIABLE VECTOR           C
C     RDSQSM             REAL*8       SUM OF RESIDUALS SQUARED            C
C     RSD                REAL*8       RESIDUAL OF POINT AT OUTLET         C
C     RSDSQ              REAL*8       RESIDUAL SQUARED OF POINT AT OUTLET C
C     RSDSUM             REAL*8       SUM OF RESIDUALS                    C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
C                                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE REPORT (I, DV, RSDSUM, RDSQSM, RSD)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100,4), DV(462), C (100), T (100)
      COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
     1               EPSLON, CO, FK, FN, RADIUS, DS, FILMK,
     2               FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT,
     3               NSTEPS, ICON
      COMMON /UNITS/ IOIN, IOOUT
      NDV = NCOL + (NCOL - 1) * NRAD
      IF (I .EQ. 1) THEN
         WRITE (IOOUT, 1000)
         WRITE (IOOUT, 1100)
         WRITE (IOOUT, 1200)
         WRITE (IOOUT, 1100)
         RSDSUM = 0.D00
         RDSQSM = 0.D00
      ENDIF
      CNORM = DV (NDV) / CO
      RSD = DATA (I,2) - CNORM
      RSDSQ = RSD * RSD
      RSDSUM = RSDSUM + RSD
      RDSQSM = RDSQSM + RSDSQ
```

```
      WRITE (IOOUT, 1300) DATA (I,1), DATA (I,2), CNORM, RSD, RSDSQ
      IF (I .GE. NOUT) THEN
         WRITE (IOOUT, 1100)
         WRITE (IOOUT, 1400) RSDSUM, RDSQSM
      ENDIF
 1000 FORMAT ('1',58X,'RESIDUAL REPORT')
 1100 FORMAT ('0-----------------------------------------------',
     1        '------------------------------------------------',
     2        '-------------------------------')
 1200 FORMAT ('0',19X,'TIME',10X,'C/Co (INPUT)',10X,'C/Co (PREDICTED)',
     1        14X,'RESIDUAL',10X,'RESIDUAL SQUARED')
 1300 FORMAT (' ',11X,E12.5,10X,E12.5,14X,E12.5,10X,E12.5,14X,E12.5)
 1400 FORMAT (' ',81X,E12.5,14X,E12.5)
      RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE          C
C     COMPLETION OF SIMULATION FOR EXPERIMENTAL COMPARISON.            C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION                DESCRIPTION           C
C     -------------     ---------   ---------------------------------- C
C     I                 INTEGER*4   LOOP INDEX                         C
C     IFLAG1            INTEGER*4   FLAG FOR ABRUPT CONCENTRATION      C
C                                   CHANGE                             C
C     IFLAG2            INTEGER*4   FLAG FOR DESORPTION                C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION     C
C     NOUT              INTEGER*4   NUMBER OF TEMPORAL STEPS TO OUTPUT C
C                                                                      C
C                                                                      C
C                                                                      C
C     BAL               REAL*8      RATIO OF MASS IN TO MASS ACCOUNTED C
C     CO                REAL*8      INITIAL CONCENTRATION              C
C     DATA              REAL*4      EXPERIMENTAL INPUT DATA            C
C                                   COLUMN 1 CONTAINS REPORT TIMES     C
C                                   COLUMN 2 GIVES EXPERIMENTAL        C
C                                   CONCENTRATION                      C
C                                   COLUMN 3 GIVES INFLUENT            C
C                                   CONCENTRATION                      C
C                                   COLUMN 4 GIVES DCO/DT              C
C     DV                REAL*8      DEPENDENT VARIABLE VECTOR          C
C     OUTCON            REAL*8      EXIT CONCENTRATION VECTOR          C
C     RCOL              REAL*8      REAL EQUIVALENT OF NCOL            C
C     THETA             REAL*8      POROSITY                           C
C     TOTDRY            REAL*8      TOTAL MASS LEFT ON SOLID PHASE     C
C     TOTINF            REAL*8      TOTAL MASS INJECTED INTO COLUMN    C
C     TOTOUT            REAL*8      TOTAL MASS TO EXIT COLUMN          C
C     TOTWET            REAL*8      TOTAL MASS LEFT IN LIQUID PHASE    C
C     VEL               REAL*8      PORE VELOCITY                      C
C     VEL1              REAL*8      SORPTION PORE VELOCITY             C
C     VEL2              REAL*8      DESORPTION PORE VELOCITY           C
C     ZCOL              REAL*8      LENGTH OF COLUMN (Z DIRECTION)     C
C                                                                      C
C                                                                      C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C
C
C
C
      SUBROUTINE MASS1 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
     1             RHO, DATA, NOUT, NCOL, NRAD)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 DATA
      DIMENSION DATA (100, 4), DV (462), OUTCON (250)
      COMMON /UNITS/ IOIN, IOOUT
C
      IFLAG1 = 0
      IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
      VEL1 = VEL
C
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
C
C     CALCULATE  CONTAMINANT INJECTED DURING FIRST
C     REPORT PERIOD.
C
          TOTINF = CO * DATA (1, 1) * VEL * THETA +
     1             0.5 * (DATA (1, 3) - CO) * DATA (1, 1) *
     2             VEL * THETA
C
C     CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
C
      DO 100 I = 2, (NOUT - 1)
         IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
     1       (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
             IFLAG1 = 1
             TOTINF = TOTINF +
     1                 DATA (I, 3) * (DATA (I, 1) - DATA (I - 1, 1)) *
     2                 VEL * THETA
         ELSE
             TOTINF = TOTINF +
     1                 DATA (I - 1, 3) *
     2                 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA +
     3                 0.5 * (DATA (I, 3) - DATA (I - 1, 3)) *
     4                 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
         ENDIF
 100  CONTINUE
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
C     CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
C     THE FIRST REPORT PERIOD
C
      TOTOUT = (0.75 * 0.D0 + 0.25 * OUTCON (1)) *
     1         DATA (1, 1) / 2.D0 * VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C     PERIOD
C
      TOTOUT = TOTOUT +
     1         ((0.75 * OUTCON (1) + 0.25 * 0.D0) *
     2         (DATA (1, 1) - 0.D0) / 2.D0 +
     3         (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
     4         (DATA (2, 1) - DATA (1, 1)) / 2.D0) *
     5         VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
```

```
C     PERIOD
C
      DO 110 I = 2, (NOUT - 1)
C
C     RESET VELOCITY IF DESORPTION BEGINS
C
         IF (DATA (I, 3) .LT. 1.0E-10) THEN
            VEL = VEL2
            IFLAG2 = 1
         ENDIF
C
         TOTOUT = TOTOUT +
     1              ((0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
     2              (DATA (I, 1) - DATA (I - 1, 1)) / 2.DO +
     3              (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
     4              (DATA (I + 1, 1) - DATA (I, 1)) / 2.DO) *
     5              VEL * THETA
 110  CONTINUE
C
C     CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C     REPORT PERIOD
C
      TOTOUT = TOTOUT +
     1           (0.75 * OUTCON (NOUT) +
     2           0.25 * OUTCON (NOUT - 1)) *
     3           (DATA (NOUT, 1) - DATA (NOUT - 1, 1)) / 2.DO *
     4           VEL * THETA
C
C
C     RESTORE VELOCITY
C
      VEL = VEL1
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
C
      RCOL = REAL (NCOL)
      TOTWET = (0.75 * DV (1) + 0.25 * DV (NRAD + 2)) *
     1           ZCOL / (RCOL - 1.DO) / 2.DO * THETA
C
C     CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE
C
      DO 120 I = (NRAD + 2), ((NCOL - 2) * NRAD + NCOL - 1), (NRAD + 1)
         TOTWET = TOTWET +
     1              (0.75 * DV (I) + 0.125 * DV (I - NRAD - 1) +
     2              0.125 * DV (I + NRAD + 1)) *
     3              ZCOL / (RCOL - 1.DO) * THETA
 120  CONTINUE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
C
      TOTWET = TOTWET + (0.25 * DV ((NCOL - 2) * NRAD + NCOL - 1) +
     1           0.75 * DV ((NRAD + 1) * (NCOL - 1) + 1)) *
     2           ZCOL / (RCOL - 1.DO) / 2.DO * THETA
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE
C
C
      TOTDRY = 0.DO
      WGTSUM = 0.DO
      RRAD = REAL (NRAD)
C
C     CONSIDER EACH PARTICLE RADIUS ...
C
```

```
      DO 140 IRAD = 2, NRAD + 1, 1
         RIRAD = REAL (IRAD)
         RADSUM = 0.D0
C
C     ...AND CONSIDER THAT RADIUS OVER THE ENTIRE COLUMN LENGTH
C
         DO 130 ICOL = IRAD, ((NRAD + 1) * (NCOL - 1) + IRAD),
     1   (NRAD + 1)
            IF ((ICOL .LE. IRAD) .OR.
     1      (ICOL .GE. ((NRAD + 1) * (NCOL - 1) + IRAD)))
     2      THEN
               RADSUM = RADSUM + 0.50 * DV (ICOL)
            ELSE
               RADSUM = RADSUM + DV (ICOL)
            ENDIF
 130     CONTINUE
C
C     CALCULATE THE AVERAGE CONCENTRATION AT THIS RADIUS
C
         RADAVG = RADSUM / (RCOL - 1.D0)
C
C     CALCULATE APPROPRIATE WEIGHTING FOR THIS RADIUS BASED
C     ON ITS CONTRIBUTION TO THE OVERALL PARTICLE MASS
C
         RADWGT = 3 * (((RIRAD - 1.D0) / RRAD) ** 2) * 1.D0 / RRAD
         WGTSUM = WGTSUM + RADWGT
C
C     CREDIT AN APPROPRIATE CONTRIBUTION FROM THIS RADIUS
C
         TOTDRY = TOTDRY + RADAVG * ZCOL * RHO * RADWGT *
     1            (1.D0 - THETA)
C
 140  CONTINUE
C
C     THE WEIGHTING FACTORS SHOULD SUM TO UNITY.  IN COARSE
C     DISCRETIZATION THEY MAY NOT DO SO PRECISELY.  APPLY A
C     BLANKET CORRECTION.
C
      TOTDRY = TOTDRY / WGTSUM
C
C.....COMPUTE MASS BALANCE
C
C
      BAL = TOTINF / ( TOTOUT + TOTWET + TOTDRY )
C
C
C.....OUTPUT THE MASS BALANCE FINDINGS
C
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TOTINF
      WRITE (IOOUT, 1300) TOTOUT
      WRITE (IOOUT, 1400) TOTWET
      WRITE (IOOUT, 1500) TOTDRY
      WRITE (IOOUT, 1600) BAL
      IF (IFLAG1 .EQ. 1) THEN
         WRITE (IOOUT, 1700)
      ENDIF
      IF (IFLAG2 .EQ. 1) THEN
         WRITE (IOOUT, 1800)
      ENDIF
C
C
 1000 FORMAT ('1', 35X, 'EXPERIMENTAL MASS BALANCE REPORT')
 1100 FORMAT ('0--------------------------------------------------',
     1        '---------------------------------------------------',
```

```
     2       '---------------------------------')
 1200 FORMAT (' ', 20X, 'TOTAL MASS INJECTED = ',
     1      E10.4, ' GRAMS / SQ. CM ')
 1300 FORMAT (' ', 20X, 'TOTAL MASS RELEASED = ',
     1      E10.4, ' GRAMS / SQ. CM ')
 1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ',
     1      E10.4, ' GRAMS / SQ. CM ')
 1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ',
     1      E10.4, ' GRAMS / SQ. CM ')
 1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = ',
     1      E10.4)
 1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
     1       ' IN INFLUENT CONCENTRATION.')
 1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
     1       ' DURING DESORPITON.')
C
C
      RETURN
      END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE          C
C     COMPLETION OF SIMULATION IN SIMULATION MODE.                     C
C                                                                      C
C                                                                      C
C     VARIABLE MAP:                                                    C
C     ------------                                                     C
C                                                                      C
C     VARIABLE NAME     PRECISION                DESCRIPTION           C
C     -------------     ---------   ---------------------------------  C
C     I                 INTEGER*4   LOOP INDEX                         C
C     IFLAG1            INTEGER*4   FLAG FOR ABRUPT CONCENTRATION      C
C                                   CHANGE                             C
C     IFLAG2            INTEGER*4   FLAG FOR DESORPTION                C
C     IOIN              INTEGER*4   UNIT NUMBER TO READ FROM           C
C     IOOUT             INTEGER*4   UNIT NUMBER TO WRITE TO            C
C     NCOL              INTEGER*4   NUMBER OF NODES IN Z DIRECTION     C
C     NSTEPS            INTEGER*4   NUMBER OF TEMPORAL STEPS           C
C                                                                      C
C                                                                      C
C                                                                      C
C     BAL               REAL*8      RATIO OF MASS IN TO MASS ACCOUNTED C
C     CO                REAL*8      INITIAL CONCENTRATION              C
C     DV                REAL*8      DEPENDENT VARIABLE VECTOR          C
C     OUTCON            REAL*8      EXIT CONCENTRATION VECTOR          C
C     OUTPUT            REAL*4      SIMULATION STATUS MATRIX           C
C                                   COL 1 GIVES REPORT TIMES           C
C                                   COL 2 GIVES BED VOLUMES            C
C                                   COL 3 GIVES THROUGHPUT             C
C                                   COL 4 GIVES INFLUENT CONCENTRATION C
C                                   COL 5 GIVES DCO/DT                 C
C     RCOL              REAL*8      REAL EQUIVALENT OF NCOL            C
C     THETA             REAL*8      POROSITY                           C
C     TOTDRY            REAL*8      TOTAL MASS LEFT ON SOLID PHASE     C
C     TOTINF            REAL*8      TOTAL MASS INJECTED INTO COLUMN    C
C     TOTOUT            REAL*8      TOTAL MASS TO EXIT COLUMN          C
C     TOTWET            REAL*8      TOTAL MASS LEFT IN LIQUID PHASE    C
C     VEL               REAL*8      PORE VELOCITY                      C
C     VEL1              REAL*8      SORPTION PORE VELOCITY             C
C     VEL2              REAL*8      DESORPTION PORE VELOCITY           C
C     ZCOL              REAL*8      LENGTH OF COLUMN (Z DIRECTION)     C
C                                                                      C
C                                                                      C
C                                                                      C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE MASS2 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
     1            RHO, OUTPUT, NSTEPS , NCOL, NRAD)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*4 OUTPUT
      DIMENSION OUTPUT (250, 5), DV (462), OUTCON (250)
      COMMON /UNITS/ IOIN, IOOUT
C
      IFLAG1 = 0
      IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
      VEL1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
C
C     CALCULATE  CONTAMINANT INJECTED DURING FIRST
C     REPORT PERIOD.
C
          TOTINF = CO * OUTPUT (1, 1) * VEL *.THETA +
     1             0.5 * (OUTPUT (1, 4) - CO) * OUTPUT (1, 1) *
     2             VEL * THETA
C
C     CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
C
      DO 100 I = 2, (NSTEPS - 1)
         IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     1       (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
             IFLAG1 = 1
             TOTINF = TOTINF +
     1                OUTPUT (I, 4) *
     2                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     3                VEL * THETA
         ELSE
             TOTINF = TOTINF +
     1                OUTPUT (I - 1, 4) *
     2                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     3                VEL * THETA +
     4                0.5 * (OUTPUT (I, 4) - OUTPUT (I - 1, 4)) *
     5                (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
     6                VEL * THETA
         ENDIF
 100  CONTINUE
C
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
C     CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
C     THE FIRST REPORT PERIOD
C
      TOTOUT = (0.75 * 0.D0 + 0.25 * OUTCON (1)) *
     1         OUTPUT (1, 1) / 2.D0 * VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C     PERIOD
C
      TOTOUT = TOTOUT +
     1         ((0.75 * OUTCON (1) + 0.25 * 0.D0) *
     2         (OUTPUT (1, 1) - 0.D0) / 2.D0 *
```

```
     3         (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
     4         (OUTPUT (2, 1) - OUTPUT (1, 1)) / 2.D0) *
     5         VEL * THETA
C
C     CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
C     PERIOD
C
      DO 110 I = 2, (NSTEPS - 1)
C
C     RESET VELOCITY IF DESORPTION BEGINS
C
         IF (OUTPUT (I, 4) .LT. 1.0E-10) THEN
            VEL = VEL2
            IFLAG2 = 1
         ENDIF
         TOTOUT = TOTOUT +
     1            ((0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
     2            (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) / 2.D0 +
     3            (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
     4            (OUTPUT (I + 1, 1) - OUTPUT (I, 1)) / 2.D0) *
     5            VEL * THETA
 110  CONTINUE
C
C     CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C     REPORT PERIOD
C
      TOTOUT = TOTOUT +
     1         (0.75 * OUTCON (NSTEPS) +
     2         0.25 * OUTCON (NSTEPS - 1)) *
     3         (OUTPUT (NSTEPS, 1) - OUTPUT (NSTEPS - 1, 1)) / 2.D0 *
     4         VEL * THETA
C
C
C     RESTORE VELOCITY
C
      VEL = VEL1
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
C
      RCOL = REAL (NCOL)
      TOTWET = (0.75 * DV (1) + 0.25 * DV (NRAD + 2)) *
     1         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C     CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE
C
      DO 120 I = (NRAD + 2), ((NCOL - 2) * NRAD + NCOL - 1), (NRAD + 1)
         TOTWET = TOTWET +
     1            (0.75 * DV (I) + 0.125 * DV (I - NRAD - 1) +
     2            0.125 * DV (I + NRAD + 1)) *
     3            ZCOL / (RCOL - 1.D0) * THETA
 120  CONTINUE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
C
      TOTWET = TOTWET + (0.25 * DV ((NCOL - 2) * NRAD + NCOL - 1) +
     1         0.75 * DV ((NRAD + 1) * (NCOL - 1) + 1)) *
     2         ZCOL / (RCOL - 1.D0) / 2.D0 * THETA
C
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE
C
C
      TOTDRY = 0.D0
      WGTSUM = 0.D0
```

```
      RRAD = REAL (NRAD)
C
C     CONSIDER EACH PARTICLE RADIUS ...
C
      DO 140 IRAD = 2, NRAD + 1, 1
         RIRAD = REAL (IRAD)
         RADSUM = 0.DO
C
C     ...AND CONSIDER THAT RADIUS OVER THE ENTIRE COLUMN LENGTH
C
         DO 130 ICOL = IRAD, ((NRAD + 1) * (NCOL - 1) + IRAD),
     1   (NRAD + 1)
            IF ((ICOL .LE. IRAD) .OR.
     1      (ICOL .GE. ((NRAD + 1) * (NCOL - 1) + IRAD)))
     2      THEN
               RADSUM = RADSUM + 0.50 * DV (ICOL)
            ELSE
               RADSUM = RADSUM + DV (ICOL)
            ENDIF
 130     CONTINUE
C
C     CALCULATE THE AVERAGE CONCENTRATION AT THIS RADIUS
C
         RADAVG = RADSUM / (RCOL - 1.DO)
C
C     CALCULATE APPROPRIATE WEIGHTING FOR THIS RADIUS BASED
C     ON ITS CONTRIBUTION TO THE OVERALL PARTICLE MASS
C
         RADWGT = 3 * (((RIRAD - 1.DO) / RRAD) ** 2) * 1.DO / RRAD
         WGTSUM = WGTSUM + RADWGT
C
C     CREDIT AN APPROPRIATE CONTRIBUTION FROM THIS RADIUS
C
         TOTDRY = TOTDRY + RADAVG * ZCOL * RHO * RADWGT *
     1            (1.DO - THETA)
C
 140  CONTINUE
C
C
C     THE WEIGHTING FACTORS SHOULD SUM TO UNITY. IN COARSE
C     DISCRETIZATION THEY MAY NOT DO SO PRECISELY.  APPLY A
C     BLANKET CORRECTION.
C
      TOTDRY = TOTDRY / WGTSUM
C
C
C.....COMPUTE MASS BALANCE
C
C
      BAL = TOTINF / ( TOTOUT + TOTWET + TOTDRY )
C
C
C.....OUTPUT THE MASS BALANCE FINDINGS
C
      WRITE (IOOUT, 1000)
      WRITE (IOOUT, 1100)
      WRITE (IOOUT, 1200) TOTINF
      WRITE (IOOUT, 1300) TOTOUT
      WRITE (IOOUT, 1400) TOTWET
      WRITE (IOOUT, 1500) TOTDRY
      WRITE (IOOUT, 1600) BAL
      IF (IFLAG1 .EQ. 1) THEN
         WRITE (IOOUT, 1700)
      ENDIF
      IF (IFLAG2 .EQ. 1) THEN
         WRITE (IOOUT, 1800)
```

```
      ENDIF
C
C
 1000 FORMAT ('1', 35X, 'SIMULATION MASS BALANCE REPORT ')
 1100 FORMAT ('0---------------------------------------------------',
     1         '----------------------------------------------------',
     2         '----------------------------------')
 1200 FORMAT (' ', 20X, 'TOTAL MASS INJECTED = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1300 FORMAT (' ', 20X, 'TOTAL MASS RELEASED = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ',
     1        E10.4, ' GRAMS / SQ. CM ')
 1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = ',
     1        E10.4)
 1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
     1         ' IN INFLUENT CONCENTRATION.')
 1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
     1         ' DURING DESORPITON.')
C
C
      RETURN
      END
```