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 phenomenon. 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ix</td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>A. Ground water contamination, the general</td>
<td>1</td>
</tr>
<tr>
<td>problem</td>
<td></td>
</tr>
<tr>
<td>B. Contaminant tailing, the specific problem</td>
<td>2</td>
</tr>
<tr>
<td>C. Improved contaminant transport models, the</td>
<td>6</td>
</tr>
<tr>
<td>focus of this report</td>
<td></td>
</tr>
<tr>
<td>II. THEORETICAL BACKGROUND AND LITERATURE</td>
<td>8</td>
</tr>
<tr>
<td>REVIEW</td>
<td></td>
</tr>
<tr>
<td>A. The advective-dispersive equation</td>
<td>9</td>
</tr>
<tr>
<td>B. The sorption expression</td>
<td>11</td>
</tr>
<tr>
<td>C. Related research</td>
<td>23</td>
</tr>
<tr>
<td>III. EXPERIMENTAL METHODS</td>
<td>25</td>
</tr>
<tr>
<td>A. Description of laboratory studies</td>
<td>25</td>
</tr>
<tr>
<td>B. Improvement of numerical models</td>
<td>26</td>
</tr>
<tr>
<td>C. Studies with the numerical models</td>
<td>36</td>
</tr>
<tr>
<td>IV. EXPERIMENTAL RESULTS AND DISCUSSION</td>
<td>39</td>
</tr>
<tr>
<td>A. Laboratory-scale analysis</td>
<td>39</td>
</tr>
<tr>
<td>B. Field-scale analysis</td>
<td>85</td>
</tr>
<tr>
<td>Number</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>I-1</td>
<td>The Tailing Phenomenon</td>
</tr>
<tr>
<td>IV-1</td>
<td>LED Validation: Analytic Solution</td>
</tr>
<tr>
<td>IV-2</td>
<td>LED Validation: Numerical Prediction</td>
</tr>
<tr>
<td>IV-3</td>
<td>LSO Validation: Analytic Solution</td>
</tr>
<tr>
<td>IV-4</td>
<td>LSO Validation: Numerical Prediction</td>
</tr>
<tr>
<td>IV-5</td>
<td>FED Validation: Analytic Solution</td>
</tr>
<tr>
<td>IV-6</td>
<td>FED Validation: Numerical Prediction</td>
</tr>
<tr>
<td>IV-7</td>
<td>DUAL Validation: Analytic Solution</td>
</tr>
<tr>
<td>IV-8</td>
<td>DUAL Validation: Numerical Prediction</td>
</tr>
<tr>
<td>IV-9</td>
<td>Tracer Test 15-1 Fit with 21 Nodes</td>
</tr>
<tr>
<td>IV-10</td>
<td>Tracer Test 16-1 Fit with 21 Nodes</td>
</tr>
<tr>
<td>IV-11</td>
<td>Tracer Test 16-2 Fit with 21 Nodes</td>
</tr>
<tr>
<td>IV-12</td>
<td>15-1, Linear LED, 1 Parameter Fit</td>
</tr>
<tr>
<td>IV-13</td>
<td>15-1, Freundlich LED, 1 Parameter Fit</td>
</tr>
<tr>
<td>IV-14</td>
<td>15-1, LSO, 1 Parameter Fit</td>
</tr>
<tr>
<td>IV-15</td>
<td>15-1, FED, 1 Parameter Fit</td>
</tr>
<tr>
<td>IV-16</td>
<td>15-1, DUAL, 1 Parameter Fit</td>
</tr>
<tr>
<td>IV-17</td>
<td>15-1, Linear LED, DH Sensitivity</td>
</tr>
<tr>
<td>IV-18</td>
<td>15-1, Freundlich LED, DH Sensitivity</td>
</tr>
<tr>
<td>IV-19</td>
<td>15-1, LSO, SOK Sensitivity</td>
</tr>
<tr>
<td>IV-20</td>
<td>15-1, FED, Fast/Slow Sensitivity</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>III-1</td>
<td>Benchmark Testing in Various Computer Environments</td>
<td>33</td>
</tr>
<tr>
<td>IV-1</td>
<td>Fitted Hydrodynamic Parameters</td>
<td>53</td>
</tr>
<tr>
<td>IV-2</td>
<td>Laboratory Column Parameters Known at Start of Sorption Study</td>
<td>55</td>
</tr>
<tr>
<td>IV-3</td>
<td>Findings of Sorption Calibration Procedure</td>
<td>57</td>
</tr>
<tr>
<td>IV-4</td>
<td>Input Parameters for Models of Column 15-1, the Calibration Run</td>
<td>59</td>
</tr>
<tr>
<td>IV-5</td>
<td>Variance of Fitted Models of Column 15-1</td>
<td>66</td>
</tr>
<tr>
<td>IV-6</td>
<td>Input Parameters for Models of Column 16-1, A Predictive Run</td>
<td>75</td>
</tr>
<tr>
<td>IV-7</td>
<td>Input Parameters for Models of Column 16-2, A Predictive Run</td>
<td>76</td>
</tr>
<tr>
<td>IV-8</td>
<td>Variance of Predictive Models of Column 16-1</td>
<td>83</td>
</tr>
<tr>
<td>IV-9</td>
<td>Variance of Predictive Models of Column 16-2</td>
<td>83</td>
</tr>
<tr>
<td>IV-10</td>
<td>Input Parameters for Field-Scale Investigations</td>
<td>87</td>
</tr>
</tbody>
</table>
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 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 1-1

THE TAILING PHENOMENON

□ MEASURED VALUES  ○ PREDICTED VALUES

CONCENTRATION

TIME
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

\[ \Delta \text{mass} = \text{area} \times (\text{flux in}) - \text{area} \times (\text{flux out}) + (\text{source or sink}) + (\text{reaction}) \]  

(II-1)

where

- \( \Delta \text{mass} \) = the net rate of change of contaminant mass within the elemental volume (MT\(^{-1}\));
- \( \text{area} \) = area of element face normal to velocity of flow (L\(^2\));
- \( \text{flux in} \) = mass transfer into the elemental volume (ML\(^{-2}\)T\(^{-1}\));
- \( \text{flux out} \) = mass transfer from the elemental volume (ML\(^{-2}\)T\(^{-1}\));
- \( \text{source or sink} \) = contaminant mass added or removed (MT\(^{-1}\));
- \( \text{reaction} \) = 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:

$$\frac{\delta C}{\delta t} = -\vec{v} \cdot \text{grad} C + \text{div}(D_h \text{grad} C)$$  \hspace{1cm} (II-2)

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

where

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

$t = \text{time (T)}$;

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

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

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

$\rho_b = \text{bulk density of the soil phase (ML}^{-3}\text{)}$;

$\theta = \text{volume void fraction in the medium (dimensionless)}$;

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

$\text{rxn} = \text{subscript denoting a general chemical or mass-transfer reaction (dimensionless)}$;

$\text{srp} = \text{subscript denoting sorption reaction (dimensionless)}$.

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):

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

(II-3)

where

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

The local equilibrium approach

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,

$$\frac{\delta q}{\delta t} = \frac{\delta q}{\delta C} \frac{\delta C}{\delta t}.$$  \hspace{1cm} (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.

\[ q_e = K_F C_e^n \]  

where

- \( q_e \) = equilibrium volume-average soil-phase mass normalized by the solid-phase mass (MM\(^{-1}\));
- \( K_F \) = Freundlich isotherm sorption-capacity constant \((L^3 M^{-1})^n\);
- \( C_e \) = 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).

\[ \frac{dq}{dC} = \frac{\delta C}{\delta t} n K_F C_e^{n-1} \]  

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

$$\frac{\delta C}{\delta t} = \frac{\delta^2 C}{\delta z^2} - v \frac{\delta C}{\delta z} - \frac{\rho_{D}}{\theta} n K_f C^{n-1} \frac{\delta C}{\delta t}.$$ (II-8)
Rearrangement yields

\[ R_f \frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z^2} - v_z \frac{\delta C}{\delta z} \]  

(II-9)

where

\[ R_f = 1 + \frac{\rho_b}{\theta} nK_F C^{n-1} \]

\( R_f = \) retardation factor.

The term \( R_f \) 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

$$\frac{\delta q}{\delta t} = k_s[C(Q^o - q) - \frac{q}{b}]$$  \hspace{1cm} (II-10)

where

- $Q^o = \text{Langmuir isotherm sorption-capacity constant (MM}^{-1}\text{)}$;
- $k_s = \text{second-order Langmuir model rate constant (L}^3M^{-1}T^{-1}\text{)}$;
- $b = \text{Langmuir isotherm sorption-energy constant (L}^3M^{-1}\text{)}$.

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

$$q_e = \frac{bQ^oC_e}{1 + bC_e}$$  \hspace{1cm} (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

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

$$- \frac{\rho_b}{\theta} k_s[C(Q^o - q) - \frac{q}{b}]$$  \hspace{1cm} (II-12)
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,

$$\frac{\delta q}{\delta t} = \frac{\delta q_f}{\delta t} + \frac{\delta q_s}{\delta t}$$  \hspace{1cm} (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 soil phase for the slow sorption rate component of the equilibrium/first-order model (MM$^{-1}$).

Applying the chain rule,

$$\frac{\delta q}{\delta t} = \frac{\delta q_f}{\delta c} \frac{\delta c}{\delta t} + \frac{\delta q_s}{\delta t}$$  \hspace{1cm} (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

\[
\frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z^2} - v \frac{\delta C}{\delta z} - \frac{\rho_b}{\theta} \left( \frac{\delta q_s}{\delta C} \frac{\delta C}{\delta t} + \frac{\delta q_s}{\delta t} \right). \tag{II-15}
\]

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

\[
\left(1 + \frac{K_{F,f} n_f \rho_b}{\theta} C^{n_f-1}\right) \frac{\delta C}{\delta t} = D_h \frac{\delta^2 C}{\delta z^2} - v \frac{\delta C}{\delta z} - \frac{\rho_b}{\theta} \frac{\delta q_s}{\delta t} \tag{II-16}
\]

where

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

\[
n_f = \text{Freundlich isotherm sorption-energy constant for the rapid rate component of the equilibrium/first-order rate model (dimensionless).}
\]

The addition of a first-order expression for the slow-sorption term yields
\[
R_{f,f} \frac{\delta C}{\delta t} + D_h \frac{\delta^2 C}{\delta z^2} - v_z \frac{\delta C}{\delta z} - \frac{\rho b}{\theta} \alpha(K_{F,s} C^n_s - q_s) = 0
\]  

(II-17)

where

\( \alpha = \) 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\(^3\)M\(^{-1}\) 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

\[
\frac{\delta q_s}{\delta t} = \alpha(K_{F,s} C^n_s - q_s)
\]

(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

\[
q = \frac{3}{R^3} \left( \int_0^R r^2 q_r \, dr \right)
\]  \hspace{1cm} \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}).
By applying a mass balance to mass transfer among adjacent shells and assuming Fickian diffusion, the solid phase governing equation is

$$\frac{\delta q}{\delta t} = \frac{3}{R^3} \frac{\delta}{\delta t} \left( \int_0^R r^2 q_r \, dr \right) .$$  \hspace{1cm} (II-20)

where

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

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

$$\frac{\delta q_r}{\delta r} \bigg|_{r=0} = 0 . \hspace{1cm} (II-22)$$

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

$$\frac{\delta q_r}{\delta r} \bigg|_{r=R} = \frac{k_f}{D_s \rho} (C - C_s) . \hspace{1cm} (II-23)$$
where

\[ k_f = \text{film mass transfer coefficient for the DUAL model (LT}^{-1}) \]

\[ C = \text{fluid-phase, equilibrium-isotherm concentration} \]

\[ C_s = \text{corresponding to the soil-phase concentration at the} \]

\[ \text{particle boundary (ML}^{-3}) \]

\[ \rho = \text{density of the soil particle (ML}^{-3}) \]

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.

\[
\frac{\delta q}{\delta t} = \text{MASS FLUX} \times \text{SURFACE AREA} \times \frac{1}{\text{PARTICLE MASS}} \quad (\text{II-24})
\]

Substituting mathematical terms,

\[
\frac{\delta q}{\delta t} = k_f (C - C_s) \times 4\pi R^2 \times \frac{1}{\frac{4}{3} \pi R^3 \rho} \quad (\text{II-25})
\]

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

\[
\frac{\delta q}{\delta t} = \frac{3k_f (C - C_s)}{\rho R} \quad (\text{II-26})
\]
Now the advective-dispersive equation, equation (II-3) becomes

\[
\frac{\delta C}{\delta t} = D \frac{\delta^2 C}{\delta z^2} - \frac{k_f (1 - \theta)}{R \theta} (C - C_s),
\]

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 (C₆H₅NO₂), 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 (d₆₀/d₄₃) of 2.616, a hydraulic conductivity of 4.15x10⁻³ 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 by 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

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

where

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

\[
i = \text{column node index.}
\]

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
\[
\frac{dc_i}{dt} = \left( \frac{D_h}{\Delta z^2} + \frac{v_z}{2\Delta z} \right) c_{i-1} + \left( -\frac{2D_h}{\Delta z^2} \right) c_i
\]  
(III-2)

\[
\frac{D_h}{\Delta z^2} - \frac{v_z}{2\Delta z} c_{i+1}
\]

\[
-\frac{\rho_b}{\theta} k_s \left( Q^0 - q_i \right) c_i - \frac{q_i}{b}
\]

The LSO solid phase equation (II-10) becomes

\[
\frac{dq_i}{dt} = k_s \left( Q^0 - q_i \right) c_i - \frac{q_i}{b}
\]  
(III-3)

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

\[
\frac{dc_i}{dt} = \frac{1}{R_{f,f,i}} \left( \frac{D_h}{\Delta z^2} + \frac{v_z}{2\Delta z} \right) c_{i-1} + \frac{1}{R_{f,f,i}} \left( -\frac{2D_h}{\Delta z^2} \right) c_i
\]

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

\[
-\frac{\rho_b}{R_{f,f,i} \theta} \alpha (K_{F,s} c_i^{n_s} - q_s)
\]  
(III-4)
The FED solid phase equation (II-18) becomes

\[
\frac{dq_{s,i}}{dt} = \alpha(K_{F,s} c_{i}^{n_{s}} - q_{s,i}) . \tag{III-5}
\]

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

\[
\frac{dC_{i}}{dt} = \left(\frac{D_{h}}{\Delta z^{2}} + \frac{v_{z}}{2\Delta z}\right) C_{i-1} + \left(-\frac{2D_{h}}{\Delta z^{2}} - \frac{3k_{f}(1-\theta)}{R \theta}\right) C_{i}
\]

\[
+ \left(\frac{D_{h}}{\Delta z^{2}} - \frac{v_{z}}{2\Delta z}\right) C_{i+1} + \left(-\frac{3k_{f}(1-\theta)}{R \theta}\right) C_{s,i} . \tag{III-6}
\]

The DUAL solid phase equation (II-21) becomes

\[
\frac{dq_{r}}{dt} = \left(\frac{D_{s_{-}}}{\Delta r \Delta r_{-}} - \frac{D_{s}}{r \Delta r}\right) q_{r,l-1} + \left(-\frac{D_{s}}{\Delta r \Delta r_{-}} - \frac{D_{s_{+}}}{r \Delta r_{+}}\right) q_{r,l}
\]

\[
+ \left(\frac{D_{s_{+}}}{\Delta r \Delta r_{+}} + \frac{D_{s}}{r \Delta r}\right) q_{r,l+1} . \tag{III-7}
\]

where

\[l = \text{radial node index;}\]
\[ \Delta r = \frac{\Delta r_+ + \Delta r_-}{2}; \]

\[ \Delta r_- = \frac{r_{l+1} - r_l}{2}; \]

\[ \Delta r_+ = \frac{r_{l+1} - r_l}{2}. \]

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 or 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**

<table>
<thead>
<tr>
<th>Environment</th>
<th>Program</th>
<th>Solution Time (HH:MM:SS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>LED</td>
<td>00:05:07</td>
</tr>
<tr>
<td>PC</td>
<td>LSO</td>
<td>00:05:13</td>
</tr>
<tr>
<td>PC</td>
<td>FED</td>
<td>00:14:12</td>
</tr>
<tr>
<td>PC</td>
<td>DUAL</td>
<td>NOTE 1</td>
</tr>
<tr>
<td>PC-AT</td>
<td>LED</td>
<td>00:02:31</td>
</tr>
<tr>
<td>PC-AT</td>
<td>LSO</td>
<td>00:02:20</td>
</tr>
<tr>
<td>PC-AT</td>
<td>FED</td>
<td>00:05:22</td>
</tr>
<tr>
<td>PC-AT</td>
<td>DUAL</td>
<td>&gt;09:00:00 NOTE 2</td>
</tr>
<tr>
<td>MAINFRAME</td>
<td>LED</td>
<td>00:00:05</td>
</tr>
<tr>
<td>MAINFRAME</td>
<td>LSO</td>
<td>00:00:04</td>
</tr>
<tr>
<td>MAINFRAME</td>
<td>FED</td>
<td>00:00:22</td>
</tr>
<tr>
<td>MAINFRAME</td>
<td>DUAL</td>
<td>00:06:49</td>
</tr>
</tbody>
</table>

**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

NORMALIZED CONCENTRATION

BED VOLUMES
FIGURE IV-2

LED VALIDATION: NUMERICAL PREDICTION

NORMALIZED CONCENTRATION

BED VOLUMES
FIGURE IV-3
LSO VALIDATION: ANALYTIC SOLUTION

NORMALIZED CONCENTRATION

BED VOLUMES
FIGURE IV-4

LSO VALIDATION: NUMERICAL PREDICTION

NORMALIZED CONCENTRATION

BED VOLUMES
FIGURE IV-5

FED VALIDATION: ANALYTIC SOLUTION

NORMALIZED CONCENTRATION

BED VOLUMES
FIGURE IV-7

DUAL VALIDATION: ANALYTIC SOLUTION

NORMALIZED CONCENTRATION

BED VOLUMES
FIGURE IV-8

DUAL VALIDATION: NUMERICAL PREDICTION

NORMALIZED CONCENTRATION

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.

\[
\alpha_L = \frac{D_h}{v}
\]

(IV-1)

where

\( \alpha_L \) = 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

NORMALIZED CONCENTRATION

BED VOLUMES

- OBSERVED DATA
- NUMERICAL MODEL FIT
FIGURE IV-10

TRACER TEST 16-1 FIT WITH 21 NODES

NORMALIZED CONCENTRATION

BED VOLUMES

OBSERVED DATA

NUMERICAL MODEL FIT
FIGURE IV-11

TRACER TEST 16-2 FIT WITH 21 NODES

NORMALIZED CONCENTRATION

OBSERVED DATA

BED VOLUMES

NUMERICAL MODEL FIT
Table IV-1

Fitted Hydrodynamic Parameters

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

<table>
<thead>
<tr>
<th>Column Number</th>
<th>Hydrodynamic Dispersion Coefficient (cm$^2$/hr)</th>
<th>Longitudinal Dispersivity (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-1</td>
<td>2.22</td>
<td>0.453</td>
</tr>
<tr>
<td>16-1</td>
<td>1.86</td>
<td>0.587</td>
</tr>
<tr>
<td>16-2</td>
<td>1.34</td>
<td>0.391</td>
</tr>
</tbody>
</table>
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

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longitudinal dispersitivity for linear local equilibrium model</td>
<td>2.59 cm</td>
</tr>
<tr>
<td>Longitudinal dispersitivity for Freundlich local equilibrium model</td>
<td>2.27 cm</td>
</tr>
<tr>
<td>Second-order Langmuir model rate constant</td>
<td>5900 cm³/g/hr</td>
</tr>
<tr>
<td>Freundlich isotherm sorption-capacity constant for the rapid rate component of the equilibrium/first-order rate model</td>
<td>0.00 (cm³/g)⁰.⁷³</td>
</tr>
<tr>
<td>Freundlich isotherm sorption-capacity constant for the slow rate component of the equilibrium/first-order rate model</td>
<td>0.27 (cm³/g)⁰.⁷³</td>
</tr>
<tr>
<td>Film transport coefficient for the dual-resistance model</td>
<td>0.0062 cm/hr</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_v$ during contamination</td>
<td>$0.2970 \times 10^1$ cm$^2$/hr</td>
</tr>
<tr>
<td>$D_h$ for L-LED during contamination</td>
<td>$0.1700 \times 10^2$ cm$^2$/hr</td>
</tr>
<tr>
<td>$D_h$ for F-LED during contamination</td>
<td>$0.1490 \times 10^2$ cm$^2$/hr</td>
</tr>
<tr>
<td>$\theta$</td>
<td>$0.3730$ \times 10^0$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>$0.2670 \times 10^1$ g/cm$^3$</td>
</tr>
<tr>
<td>$v$ during contamination</td>
<td>$0.6560 \times 10^1$ cm/hr</td>
</tr>
<tr>
<td>$v$ during elutriation</td>
<td>$0.5070 \times 10^1$ cm/hr</td>
</tr>
<tr>
<td>$a$</td>
<td>$0.2450 \times 10^2$ cm</td>
</tr>
<tr>
<td>$K_p$</td>
<td>$0.6690 \times 10^1$ cm$^3$/g</td>
</tr>
<tr>
<td>$K_F$</td>
<td>$0.2170 \times 10^0$ (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n$</td>
<td>$0.7300 \times 10^0$</td>
</tr>
<tr>
<td>$K_{F,f}$</td>
<td>$0.0000 \times 10^0$ (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n_f$</td>
<td>$0.7300 \times 10^0$</td>
</tr>
<tr>
<td>$K_{F,s}$</td>
<td>$0.2170 \times 10^0$ (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n_s$</td>
<td>$0.7300 \times 10^0$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$0.3320 \times 10^1$ l/hr</td>
</tr>
<tr>
<td>$Q^0$</td>
<td>$0.4400 \times 10^{-4}$ g/g</td>
</tr>
<tr>
<td>$b$</td>
<td>$0.2910 \times 10^1$ cm$^3$/g</td>
</tr>
<tr>
<td>$k_s$</td>
<td>$0.5900 \times 10^1$ cm$^3$/g/hr</td>
</tr>
<tr>
<td>$D_s$</td>
<td>$0.3440 \times 10^{-6}$ cm$^3$/hr</td>
</tr>
<tr>
<td>$K_f$</td>
<td>$0.6200 \times 10^1$ cm/hr</td>
</tr>
<tr>
<td>$R$</td>
<td>$0.1160 \times 10^1$ cm</td>
</tr>
<tr>
<td>$C_0 \ @ t=0.00$ hr</td>
<td>$0.1564 \times 10^{-5}$ g/cm$^3$</td>
</tr>
<tr>
<td>$C_1 \ @ t=144$ hr</td>
<td>$0.1310 \times 10^{-5}$ g/cm$^3$</td>
</tr>
<tr>
<td>$C_2 \ @ t=191$ hr</td>
<td>$0.1310 \times 10^{-5}$ g/cm$^3$</td>
</tr>
<tr>
<td>$C_3 \ @ t=191+$ hr</td>
<td>$0.1510 \times 10^{-5}$ g/cm$^3$</td>
</tr>
<tr>
<td>$C_4 \ @ t=600$ hr</td>
<td>$0.1000 \times 10^{-5}$ g/cm$^3$</td>
</tr>
</tbody>
</table>

* dimensionless
FIGURE IV-13

15-1, FREUNDLICH LED, 1 PARAMETER FIT

NORMALIZED CONCENTRATION

BED VOLUMES

□ OBSERVED DATA

○ NUMERICAL MODEL FIT
FIGURE IV-14

15-1, LSO, 1 PARAMETER FIT

NORMALIZED CONCENTRATION

BED VOLUMES

OBSERVED DATA

NUMERICAL MODEL FIT
FIGURE IV-15

15-1, FED, 1 PARAMETER FIT

NORMALIZED CONCENTRATION

OBSERVED DATA
NUMERICAL MODEL FIT

BED VOLUMES

0 20 40 60 80 100 120
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

<table>
<thead>
<tr>
<th>Model</th>
<th>Model Variance (g/cm²)²</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-LED</td>
<td>0.0226</td>
</tr>
<tr>
<td>F-LED</td>
<td>0.0332</td>
</tr>
<tr>
<td>LSO</td>
<td>0.0036</td>
</tr>
<tr>
<td>FED</td>
<td>0.0043</td>
</tr>
<tr>
<td>DUAL</td>
<td>0.0033</td>
</tr>
</tbody>
</table>
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

NORMALIZED CONCENTRATION

DH=0.17E+02 cm²/hr                                  +   DH=0.17E+01 cm²/hr

BED VOLUMES

DH=0.17E+03 cm²/hr
FIGURE IV-18

15-1, FREUNDLICH LED, DH SENSITIVITY

NORMALIZED CONCENTRATION

DH = 0.149 E+02 cm²/hr

BED VOLUMES

DH = 0.149 E+01 cm²/hr

DH = 0.149 E+03 cm²/hr
FIGURE IV-19

15-1, LSO, SOK SENSITIVITY

NORMALIZED CONCENTRATION

BED VOLUMES

□ SOK=5.9E+03 cm³/g/hr  + SOK=5.9E+02 cm³/g/hr  ○ SOK=5.9E+04 cm³/g/hr
FIGURE IV-20

15-1, FED, FAST/SLOW SENSITIVITY

NORMALIZED CONCENTRATION

BED VOLUMES

FKF=0.0000 (cm$^3$/g)$^{0.73}$
FKF=0.1085 (cm$^3$/g)$^{0.73}$
FKF=0.2170 (cm$^3$/g)$^{0.73}$
FIGURE IV-21

15-1, DUAL, FILMK SENSITIVITY

FILMK = 6.2E-03 cm/hr
FILMK = 6.2E-04 cm/hr
FILMK = 6.2E-02 cm/hr
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 fourfold 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_h$ during contamination</td>
<td>0.1773 E+01 cm$^2$/hr</td>
</tr>
<tr>
<td>$D_h$ for L-LED during contamination</td>
<td>0.7820 E+01 cm$^2$/hr</td>
</tr>
<tr>
<td>$D_h$ for F-LED during contamination</td>
<td>0.6855 E+01 cm$^2$/hr</td>
</tr>
<tr>
<td>$\theta$</td>
<td>0.3730 E+00*</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.2670 E+01 g/cm$^3$</td>
</tr>
<tr>
<td>$v$ during contamination</td>
<td>0.3020 E+01 cm/hr</td>
</tr>
<tr>
<td>$v$ during elutriation</td>
<td>0.3140 E+01 cm/hr</td>
</tr>
<tr>
<td>$z$</td>
<td>0.2180 E+02 cm</td>
</tr>
<tr>
<td>$K_p$</td>
<td>0.6690 E+01 cm$^3$/g</td>
</tr>
<tr>
<td>$K_F$</td>
<td>0.2170 E+00 (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n$</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>$K_{F,f}$</td>
<td>0.0000 E+00 (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n$</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>$K_{F,s}$</td>
<td>0.2170 E+00 (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n_s$</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>$a$</td>
<td>0.3320 E-01 l/hr</td>
</tr>
<tr>
<td>$Q^o$</td>
<td>0.4400 E-04 g/g</td>
</tr>
<tr>
<td>$b$</td>
<td>0.2910 E+06 cm$^3$/g</td>
</tr>
<tr>
<td>$k_S$</td>
<td>0.5900 E+04 cm$^3$/g/hr</td>
</tr>
<tr>
<td>$D_S$</td>
<td>0.3440 E-06 cm$^2$/hr</td>
</tr>
<tr>
<td>$K_f$</td>
<td>0.6200 E-02 cm/hr</td>
</tr>
<tr>
<td>$R$</td>
<td>0.1160 E-01 cm/hr</td>
</tr>
<tr>
<td>$C_0$ @ t=5=0.00 hr</td>
<td>0.4653 E-05 g/cm$^3$</td>
</tr>
<tr>
<td>$C_1$ @ t=120 hr</td>
<td>0.4240 E-05 g/cm$^3$</td>
</tr>
<tr>
<td>$C_2$ @ t=482 hr</td>
<td>0.4140 E-05 g/cm$^3$</td>
</tr>
<tr>
<td>$C_3$ @ t=482+ hr</td>
<td>0.1000 E-19 g/cm$^3$</td>
</tr>
<tr>
<td>$C_4$ @ t=1000 hr</td>
<td>0.1000 E-19 g/cm$^3$</td>
</tr>
</tbody>
</table>

* dimensionless
Table IV-7

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_h$ during contamination</td>
<td>0.1290 E+01 cm$^2$/hr</td>
</tr>
<tr>
<td>$D_h$ for L-LED during contamination</td>
<td>0.8550 E+01 cm$^2$/hr</td>
</tr>
<tr>
<td>$D_h$ for F-LED during contamination</td>
<td>0.7490 E+01 cm$^3$/hr</td>
</tr>
<tr>
<td>$\theta$</td>
<td>0.3730 E+00*</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.2670 E+01 g/cm$^3$</td>
</tr>
<tr>
<td>$v$ during contamination</td>
<td>0.3300 E+01 cm/hr</td>
</tr>
<tr>
<td>$v$ during elutriation</td>
<td>0.3670 E+01 cm/hr</td>
</tr>
<tr>
<td>$z$</td>
<td>0.2180 E+02 cm</td>
</tr>
<tr>
<td>$K_p$</td>
<td>0.6690 E+01 cm$^3$/g</td>
</tr>
<tr>
<td>$K_F$</td>
<td>0.2170 E+00 (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n$</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>$K_{F,f}$</td>
<td>0.0000 E+00 (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n_f$</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>$K_{F,s}$</td>
<td>0.2170 E+00 (cm$^3$/g)$^{0.73}$</td>
</tr>
<tr>
<td>$n_s$</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>$a$</td>
<td>0.3320 E-01 l/hr</td>
</tr>
<tr>
<td>$Q^0$</td>
<td>0.4400 E-04 g/g</td>
</tr>
<tr>
<td>$b$</td>
<td>0.2910 E+06 cm$^3$/g</td>
</tr>
<tr>
<td>$K_S$</td>
<td>0.5900 E+04 cm$^3$/g/hr</td>
</tr>
<tr>
<td>$D_S$</td>
<td>0.3440 E-06 cm$^2$/hr</td>
</tr>
<tr>
<td>$K_f$</td>
<td>0.6200 E-02 cm/hr</td>
</tr>
<tr>
<td>$R$</td>
<td>0.1160 E-01 cm</td>
</tr>
<tr>
<td>$C_0$ $t=0.00$</td>
<td>0.3191 E-05 g/cm$^3$</td>
</tr>
<tr>
<td>$C_1$ $t=120$ hr</td>
<td>0.3130 E-05 g/cm$^3$</td>
</tr>
<tr>
<td>$C_2$ $t=482$ hr</td>
<td>0.2480 E-05 g/cm$^3$</td>
</tr>
<tr>
<td>$C_3$ $t=482+$ hr</td>
<td>0.1000 E-19 g/cm$^3$</td>
</tr>
<tr>
<td>$C_4$ $t=1000$ hr</td>
<td>0.1000 E-19 g/cm$^3$</td>
</tr>
</tbody>
</table>
FIGURE IV-22
16-1, LSO, PREDICTIVE MODE

NORMALIZED CONCENTRATION

BED VOLUMES

OBSERVED DATA  MODEL PREDICTION
FIGURE IV-23

16-1, FED, PREDICTIVE MODE

NORMALIZED CONCENTRATION

BED VOLUMES

□ OBSERVED DATA

○ MODEL PREDICTION
FIGURE IV-25

16-2, LSO, PREDICTIVE MODE

NORMALIZED CONCENTRATION

BED VOLUMES

□ OBSERVED DATA

◊ MODEL PREDICTION
FIGURE IV-27

16-2, DUAL, PREDICTIVE MODE

NORMALIZED CONCENTRATION

BED VOLUMES

□ OBSERVED DATA  ○ MODEL PREDICTION
### Table IV-8
Variance of Fitted Models of Column 16-1

<table>
<thead>
<tr>
<th>Model</th>
<th>Model Variance (g/cm$^3$)$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSO</td>
<td>0.0203</td>
</tr>
<tr>
<td>FED</td>
<td>0.0224</td>
</tr>
<tr>
<td>DUAL</td>
<td>0.0197</td>
</tr>
</tbody>
</table>

### Table IV-9
Variance of Fitted Models of Column 16-2

<table>
<thead>
<tr>
<th>Model</th>
<th>Model Variance (g/cm$^3$)$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSO</td>
<td>0.0375</td>
</tr>
<tr>
<td>FED</td>
<td>0.0414</td>
</tr>
<tr>
<td>DUAL</td>
<td>0.0381</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>D&lt;sub&gt;H&lt;/sub&gt; during contamination</td>
<td>0.2970 E+03 cm&lt;sup&gt;2&lt;/sup&gt;/hr</td>
</tr>
<tr>
<td>D&lt;sub&gt;H&lt;/sub&gt; for L-LED during contamination</td>
<td>0.2970 E+03 cm&lt;sup&gt;2&lt;/sup&gt;/hr</td>
</tr>
<tr>
<td>D&lt;sub&gt;H&lt;/sub&gt; for F-LED during contamination</td>
<td>0.2970 E+03 cm&lt;sup&gt;2&lt;/sup&gt;/hr</td>
</tr>
<tr>
<td>θ</td>
<td>0.3970 E+00*</td>
</tr>
<tr>
<td>ρ</td>
<td>0.2670 E+01 g/cm&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>v during contamination</td>
<td>0.6564 E+01 cm/hr</td>
</tr>
<tr>
<td>v during elutriation</td>
<td>0.6564 E+01 cm/hr</td>
</tr>
<tr>
<td>z</td>
<td>0.2450 E+04 cm</td>
</tr>
<tr>
<td>K&lt;sub&gt;P&lt;/sub&gt;</td>
<td>0.6690 E+01 cm&lt;sup&gt;3&lt;/sup&gt;/g</td>
</tr>
<tr>
<td>K&lt;sub&gt;f&lt;/sub&gt;</td>
<td>0.2170 E+00 (cm&lt;sup&gt;3&lt;/sup&gt;/g)&lt;sup&gt;0.73&lt;/sup&gt;</td>
</tr>
<tr>
<td>n</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>K&lt;sub&gt;F,f&lt;/sub&gt;</td>
<td>0.2170 E-01 (cm&lt;sup&gt;3&lt;/sup&gt;/g)&lt;sup&gt;0.73&lt;/sup&gt;</td>
</tr>
<tr>
<td>n&lt;sub&gt;f&lt;/sub&gt;</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>K&lt;sub&gt;F,s&lt;/sub&gt;</td>
<td>0.1953 E+00 (cm&lt;sup&gt;3&lt;/sup&gt;/g)&lt;sup&gt;0.73&lt;/sup&gt;</td>
</tr>
<tr>
<td>n&lt;sub&gt;s&lt;/sub&gt;</td>
<td>0.7300 E+00*</td>
</tr>
<tr>
<td>α</td>
<td>0.3320 E-01 l/hr</td>
</tr>
<tr>
<td>Q&lt;sup&gt;0&lt;/sup&gt;</td>
<td>0.4400 E-04 g/g</td>
</tr>
<tr>
<td>b</td>
<td>0.2910 E+06 cm&lt;sup&gt;3&lt;/sup&gt;/g</td>
</tr>
<tr>
<td>k&lt;sub&gt;S&lt;/sub&gt;</td>
<td>0.5900 E+04 cm&lt;sup&gt;3&lt;/sup&gt;/g/hr</td>
</tr>
<tr>
<td>D&lt;sub&gt;S&lt;/sub&gt;</td>
<td>0.3440 E-06 cm&lt;sup&gt;2&lt;/sup&gt;/hr</td>
</tr>
<tr>
<td>k&lt;sub&gt;f&lt;/sub&gt;</td>
<td>0.6200 E-02 cm/hr</td>
</tr>
<tr>
<td>R</td>
<td>0.1160 E-01 cm</td>
</tr>
<tr>
<td>C&lt;sub&gt;o&lt;/sub&gt; @ t=0.00 hr</td>
<td>0.1564 E-05 g/cm&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>C&lt;sub&gt;1&lt;/sub&gt; @ t=5=10,000 hr</td>
<td>0.1564 E-05 g/cm&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>C&lt;sub&gt;2&lt;/sub&gt; @ t=10,000+ hr</td>
<td>0.1000 E-10 g/cm&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>C&lt;sub&gt;3&lt;/sub&gt; @ t=60,000 hr</td>
<td>0.1000 E-10 g/cm&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

* dimensionsless
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 \((L^2M^{-1})\).

\(c\)     solution-phase solute concentration \((ML^{-3})\).

\(c_e\)   equilibrium solution-phase concentration \((ML^{-3})\).

\(c_0\)   initial solution-phase solute concentration \((ML)\).

\(c_s\)   fluid-phase, equilibrium-isotherm concentration corresponding to the soil-phase concentration at the particle boundary \((ML^{-1})\).

\(D_h\)   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 \((ML^2T^{-1})\).

flux out mass transfer out of the elemental volume \((ML^2T^{-1})\).

\(i\)     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 ($L^{-1} M^{-1}$).

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

$l$ 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 ($MM^{-1}$).

$Q^0$ 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 ($MT^{-1}$).
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_f$</td>
<td>retardation factor - defined by equation II-9 (dimensionless).</td>
</tr>
<tr>
<td>$R_{f,f}$</td>
<td>retardation factor for the rapidly sorbing fraction in the equilibrium/first-order model - equation II-17 (dimensionless).</td>
</tr>
<tr>
<td>rxn</td>
<td>subscript denoting a general chemical or mass-transfer reaction (dimensionless).</td>
</tr>
<tr>
<td>source or sink</td>
<td>contaminant mass added or removed ($MT^{-1}$).</td>
</tr>
<tr>
<td>srp</td>
<td>subscript denoting sorption reaction (dimensionless).</td>
</tr>
<tr>
<td>$t$</td>
<td>time ($T$).</td>
</tr>
<tr>
<td>$\vec{v}$</td>
<td>pore velocity vector ($LT^{-1}$).</td>
</tr>
<tr>
<td>$v_z$</td>
<td>pore velocity in the longitudinal direction ($LT^{-1}$).</td>
</tr>
<tr>
<td>$z$</td>
<td>longitudinal distance variable ($L$).</td>
</tr>
<tr>
<td>$Z$</td>
<td>length of column ($L$).</td>
</tr>
</tbody>
</table>
\( \alpha \)  
equilibrium/first-order model mass-transfer coefficient (T\(^{-1}\)).

\( \alpha_L \)  
longitudinal dispersivity (L).

\( \Delta \text{mass} \)  
the net rate of change of contaminant mass within the elemental value (MT\(^{-1}\)).

\( \gamma(C) \)  
fluid-phase solution source (ML\(^{-3}\)T\(^{-1}\)).

\( \theta \)  
volume void fraction of the medium (dimensionless).

\( \rho \)  
density of the soil particle (ML\(^{-3}\)).

\( \rho_b \)  
bulk density of the soil phase (ML\(^{-3}\)).
REFERENCES


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


SORPTION MODEL:
LOCAL EQUILIBRIUM WITH DISPERSION
MAINFRAME VERSION

THIS PROGRAM USES THE DGEAR METHOD TO SOLVE THE ADVECTIVE
DISPERSE EQUATION FOR THE CASE OF LOCAL-EQUILIBRIUM SORPTION.
THE PROGRAM IS BASED ON THE ASSUMPTIONS THAT THE SOLID AND FLUID
PHASES ATTAIN INSTANTANEOUS EQUILIBRIUM AND THAT THE EQUILIBRIUM
RELATION IS THE SINGLE COMPONENT FREUNDLICH ISOTHERM. THE EQUATIONS
ARE SPATIALLY APPROXIMATED USING THE FINITE DIFFERENCE METHOD
RESULTING IN A SET OF DC/DT EQUATIONS. THESE EQUATIONS
ARE THEN SOLVED SIMULTANEOUSLY USING DGEAR---AN IMSL SUBROUTINE.

IN ITS PRESENT FORM THE PROGRAM IS LIMITED TO 100 NODES.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>IDUMMY</td>
<td>INTEGER*4</td>
<td>DUMMY WORKING ARRAY FOR DGEAR</td>
</tr>
<tr>
<td>IER</td>
<td>INTEGER*4</td>
<td>ERROR CODE RETURN INDEX FOR DGEAR</td>
</tr>
<tr>
<td>INDEX</td>
<td>INTEGER*4</td>
<td>CALL INDICATOR INDEX FOR GEAR</td>
</tr>
<tr>
<td>IGIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>IWK</td>
<td>INTEGER*4</td>
<td>WORKING ARRAY FOR DGEAR</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NLC</td>
<td>INTEGER*4</td>
<td>NUMBER OF LOWER CODIAGONALS-DGEAR</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>NUC</td>
<td>INTEGER*4</td>
<td>NUMBER OF UPPER CODIAGONALS-DGEAR</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS TO BE CALCULATED</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>VECTOR OF KNOWN INFLUENT CONCENTRATIONS (SEE VECTOR T)</td>
</tr>
<tr>
<td>CNORM</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>CPRIME</td>
<td>REAL*8</td>
<td>NORMALIZED EXIT CONCENTRATION</td>
</tr>
<tr>
<td>CTEMP</td>
<td>REAL*8</td>
<td>EXTERNAL NAME OF DC/DT SUBROUTINE</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXIT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCD/DT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHANGE IN INFLUENT CONCENTRATION WITH RESPECT TO TIME</td>
</tr>
<tr>
<td>DCODT</td>
<td>REAL*8</td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>DDDVDT</td>
<td>REAL*8</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL*8</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL*8</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>SORPTION PHASE HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
</tbody>
</table>
REAL*8       DEPENDENT VARIABLE VECTOR C
REAL*8       ALLOWABLE ERROR FOR DGEAR C
REAL*8       EXTERNAL NAME OF JACOBIAN ROUTINE C
REAL*8       FREUNDLICH ISOTHERM COEFFICIENT C
REAL*8       FLUID-PHASE LOSS RATE C
REAL*8       FREUNDLICH ISOTHERM EXPONENT C
REAL*8       STEP SIZE FOR DGEAR C
REAL*8       VECTOR OF EXIT CONCENTRATIONS C
REAL*4       A MATRIX OF SIMULATION STATUS C

REAL*8       SUM OF RESIDUALS SQUARED C
REAL*8       RETARDATION FACTOR GROUPING C
REAL*8       SOLID-PHASE LOSS RATE C
REAL*8       SOLID-PHASE LOSS GROUP C
REAL*8       WORKING ARRAY NEEDED BY DGEAR C
REAL*8       LENGTH OF COLUMN (Z DIRECTION) C
CHARACTER*72 TITLE OF SIMULATION RUN

PROGRAM LED
IMPLICIT REAL*8 (A-H, O-Z)
REAL*4 SDUMMY, DATA, OUTPUT
CHARACTER*72 TITLE

DIMENSION F(IOO), X(1)
DIMENSION IKW (101)
DIMENSION DATA (100,4), DV (101), DDDVDT (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, DGNLOU, DGNUP, RFGRP, SPLGRP
COMMON /INPUT/ TMESIM, 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, I00UT
EXTERNAL CPRIME, FCNJ
IOIN = 1
I00UT = 3
CALL READ
VEL1 = VEL
CALL ECHO
CALL ERRSET (206, 256, -1, 1)

C.....COMPUTE CONCENTRATIONS AT DATA POINTS IF DATA ENTERED
C
IF (NOUT .GT. 0) THEN
  TO = 0.000
  CALL INITL (DDVDT, DV, OUTPUT, TO)
  CALL GRPFRM
  CALL INFOl (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 SIMULATION 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.
      DATA (I, 3) .GT. 1.2 * DATA ((I - 1), 3)) THEN
      H = 1.D-08
      INDEX = 1
      DCDT = 0.000
      DV (1) = DATA (1, 3)
      WRITE (IOOUT, 1300)
      IF (DV (1) .LT. 1.OE-10) THEN
        VEL = VEL2
        DH = DH1 * VEL2 / VEL1
        CALL GRPFRM
        WRITE (IOOUT, 1400)
      END IF
    ELSE
      DCDT = DATA (1, 4)
    END IF
  ELSE
    DCDT = DATA (1, 4)
  ENDIF
  TIME = DATA (1, 1)
  CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
             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,
             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.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE

NDOF = NCOL
H = 1.D-08
INDEX = 1
DV (1) = CO
NLC = 2
NUC = 2

C.....TIME LOOP FOR SIMULATION AND OUTPUT OF COMPUTED POINTS

WRITE (10OUT, 1000)
WRITE (10OUT, 1100)
DO 110 I = 1, NSTEPS
  IF (I .GT. 1) THEN
    IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
      H = 1.D-08
      INDEX = 1
      DCODT = 0.000
      DV (1) = OUTPUT (1,4)
      WRITE (10OUT, 1300)
      IF (DV (1) .LT. 1.0E-10) THEN
        VEL = VEL2
        DH = DH1 = VEL2 / VEL1
        CALL GRPFRM
        WRITE (10OUT, 1400)
      END IF
    ELSE
      DCODT = OUTPUT (1, 5)
    END IF
  ELSE
    DCODT = OUTPUT (I, 5)
  ENDIF
END IF
110    WRITE (IOOUT,1200) (OUTPUT (I, J), J = 1, 3), CTEMP, CNORM

C.....RESTORE ORIGINAL PARAMETERS

DH = DH1
VEL = VEL1

C.....CHECK THE MASS BALANCE

CALL MASS2 (OUTCON, DV, CO, ZCOL, THETA, FK, FN, VEL, VEL2, RHO, OUTPUT, NSTEPS, NCOL)

END IF

1000 FORMAT ('**', 42X, 'SIMULATION OUTPUT FOR PLOTTING'),
1100 FORMAT ('------------------------------------------------------------------------------------',
             '------------------------------------------------------------------------------------',
             '------------------------------------------------------------------------------------'),
1200 FORMAT (5E12.5)
1300 FORMAT (' ', 20X, '***** ', 'ABRUPT INFLUENT CONCENTRATION ',
             'CHANGE. ', '***** ', 'DGEAR WAS RESET. ', '*****')
1400 FORMAT (' ', 23X, '***** ', 'DESORPTION DETECTED. ', 1
THIS SUBROUTINE READS THE BASIC INPUT REQUIRED FOR THE LOCAL EQUILIBRIUM SORPTION VERSION OF THE ADVECTIVE-DISPERSIVE EQUATION FROM UNIT NUMBER IOIN.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IDOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS TO BE CALCULATED</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*8</td>
<td>VECTOR OF KNOWN INFLUENT CONCENTRATIONS (SEE VECTOR T)</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*8</td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*8</td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*8</td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*8</td>
<td>COLUMN 4 GIVES DC0/DT</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>EPSLON</td>
<td>REAL*8</td>
<td>ALLOWABLE ERROR FOR DGEAR</td>
</tr>
<tr>
<td>FK</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>FN</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>RHO</td>
<td>REAL*8</td>
<td>SOLID PHASE PARTICLE DENSITY</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>SOLID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>T</td>
<td>REAL*8</td>
<td>VECTOR OF ELAPSED TIME CORRESPONDING TO CONCENTRATIONS IN THE VECTOR C</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>THESIM</td>
<td>REAL*8</td>
<td>NUMBER OF BED VOLUMES TO SIMULATE</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
<tr>
<td>TITLE</td>
<td>CHARACTER*72</td>
<td>TITLE OF SIMULATION</td>
</tr>
</tbody>
</table>
SUBROUTINE ECHO
IMPLICIT REAL*8 (A - H, O - Z)
REAL*4 DATA

COMMON /WORDS/ TITLE
COMMON /UNITS/ I0IN, I0OUT
EXTERNAL CPRIME, FCNJ
READ (I0IN, 1000) TITLE
READ (I0IN, 1100) NCOL, NOUT, NSTEPS, THESIM, ICON
IF (ABS (NOUT) .GT. 0 ) THEN
  NTEMP = ABS (NOUT)
  READ (I0IN, 1200) (DATA (I,1), I = 1, NTEMP)
  READ (I0IN, 1200) (DATA (I,2), I = 1, NTEMP)
ENDIF
READ (I0IN, 1200) VEL, DH, RHO, THETA, ZCOL, EPSLON
READ (I0IN, 1200) CO, FK, FN, FKAPPA, SKAPPA, VEL2
READ (I0IN, 1200) (C (I), I = 1, ICON)
READ (I0IN, 1200) (T (I), I = 1, ICON)
1000 FORMAT (A72)
1100 FORMAT (315, E12.5, 15)
1200 FORMAT ((6E12.5))
RETURN
END

THIS SUBROUTINE ECHOS THE INPUT TO DEVICE IOOUT.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I0IN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN 2 DIRECTION</td>
</tr>
<tr>
<td>C</td>
<td>REAL*8</td>
<td>VECTOR OF KNOWN INFLUENT CONCENTRATIONS (SEE VECTOR T)</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>EPSLON</td>
<td>REAL*8</td>
<td>ALLOWABLE ERROR FOR DGEAR</td>
</tr>
<tr>
<td>FK</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>FN</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>PE</td>
<td>REAL*8</td>
<td>SYSTEM PECLET NUMBER</td>
</tr>
<tr>
<td>RHO</td>
<td>REAL*8</td>
<td>SOLID PHASE PARTICLE DENSITY</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>SOLID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>T</td>
<td>REAL*8</td>
<td>VECTOR OF ELAPSED TIME CORRESPONDING TO CONCENTRATIONS IN THE VECTOR C</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
<tr>
<td>TITLE</td>
<td>CHARACTER*72</td>
<td>TITLE OF SIMULATION</td>
</tr>
</tbody>
</table>

SUBROUTINE ECHO
100 FORMAT ( ' ', 50X, 'PLUG FLOW DISPERSION REACTOR')
110 FORMAT ( ' ', 47X, 'LOCAL EQUILIBRIUM KINETIC SIMULATION')
120 FORMAT ( ' ', 20X, A72)
130 FORMAT (' ', 15X, 'VARIABLE', 16X, 'DESCRIPTION', 16X, 'UNITS', 16X)
140 FORMAT (' ', 15X, 'CO', 18X, 'INITIAL CONCENTRATION', E10.4, 16X)
150 FORMAT (' ', 15X, 'D', 18X, 'HYDRODYNAMIC DISPERSION', E10.4, 16X)
160 FORMAT (' ', 15X, 'NDZ', 16X, 'NUMBER OF COLUMN STEPS', E10.4)
170 FORMAT (' ', 15X, 'RHO', 18X, 'SOLID PHASE DENSITY', E10.4, 16X)
180 FORMAT (' ', 15X, 'VEL', 18X, 'PORE VELOCITY', E10.4, 16X)
190 FORMAT (' ', 15X, 'DESORPTION VELOCITY', E10.4, 16X)
200 FORMAT (' ', 15X, 'VOID', 18X, 'VOID VOLUME FRACTION', E10.4)
210 FORMAT (' ', 15X, 'ZL', 18X, 'LENGTH OF COLUMN', E10.4)
220 FORMAT (' ', 15X, 'PE', 18X, 'PECLET NUMBER', E10.4)
230 FORMAT (' ', 15X, 'KFRS', 18X, 'FREUNDLICH COEFFICIENT', E10.4)
240 FORMAT (' ', 15X, 'NS', 18X, 'FREUNDLICH EXPONENT', E10.4)
250 FORMAT (' ', 15X, 'DIA', 18X, 'FLUID PHASE LOSS RATE', E10.4, 16X)
260 FORMAT (' ', 15X, 'NDR', 18X, 'SOLID PHASE LOSS RATE', E10.4)
THIS SUBROUTINE WRITES THE CONTENTS OF THE DATA MATRIX TO UNIT 100UT.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IDOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>INTEGER*4</td>
<td>LOOP COUNTER</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
</tbody>
</table>

SUBROUTINE INFOl (DATA, NOUT)

REAL*4 DATA
DIMENSION DATA (100,4)
COMMON /UNITS/ IDIN, IDOUT
WRITE (IDOUT, 1000)
WRITE (IDOUT, 1100)
WRITE (IDOUT, 1200)
WRITE (IDOUT, 1100)
DO 100 NCOUNT = 1, NOUT
   WRITE (IDOUT, 1300) ( DATA (NCOUNT, I), I = 1, 4)
100 CONTINUE
1000 FORMAT ('1', 42X, 'EXPERIMENTAL COMPARISON MODE INFORMATION')
1100 FORMAT ('0----------------------------------------',
           '1', '----------------------------------------',
           '2', '----------------------------------------')
1200 FORMAT ('1', 26X, 'TIME',5X,'CONCENTRATION OUT', 6X,
           'CONCENTRATION IN', 16X, 'DCO/DT')
1300 FORMAT ('1', 20X, E12.5, 10X, E12.5, 10X, E12.5, 10X, E12.5)
RETURN
END

THIS SUBROUTINE WRITES THE CONTENTS OF THE OUTPUT MATRIX TO UNIT 100UT.
VARIABLE MAP:

---

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>INTEGER*4</td>
<td>LOOP COUNTER</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS TO BE CALCULATED</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*4</td>
<td>A MATRIX OF SIMULATION STATUS COL 1 CONTAINS REPORT TIME</td>
</tr>
</tbody>
</table>

SUBROUTINE INFO2 (OUTPUT, NSTEPS)

REAL*4 OUTPUT

DIMENSION OUTPUT (250, 5)

COMMON /UNITS/ IN, OUT

WRITE (OUT, 1000)

WRITE (OUT, 1100)

WRITE (OUT, 1200)

WRITE (OUT, 1300)

DO 100 NCOUNT = 1, NSTEPS

WRITE (OUT, 1300) (OUTPUT (NCOUNT, I), I = 1, 5)

CONTINUE

RETURN
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.....INITIALIZE DEPENDENT VARIABLE AND DERIVATIVE ARRAYS
C
DO 100 I = 1, 101
100 DDVDT (I) = 0.000
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 = 0.000
DO 110 I = 1, NSTEPS
TEMPT = TEMPT + DT
OUTPUT (1,1) = TEMPT
OUTPUT (1,2) = TEMPT / TAU
110 OUTPUT (1,3) = TEMPT / (TAU * DG)
ENDIF

C.....CALCULATE THE MACHINE EPSILON FOR USE AS A LOWER ERROR BOUND
C
EPSMCN = 1.0
120 EPSMCN = EPSMCN / 2.0000
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)) THEN
          DATA (NCOUNT, 3) = DNCON + (UPCON - DNCON)
          * (TEND - DNTIME) / (UPTIME - DNTIME)
          DONE = .TRUE.
        ENDIF
      ENDIF
    140 CONTINUE
  IF (NCOUNT .GT. 1) THEN
    DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) - DATA (NCOUNT - 1, 3)) / (DATA (NCOUNT, 1) - DATA (NCOUNT - 1, 1))
  ELSE
    DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) - CO) / (DATA (NCOUNT, 1) - TO)
  ENDIF
  130 CONTINUE
ENDIF

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)) THEN
          OUTPUT (NCOUNT, 4) = DNCON + (UPCON - DNCON)
          * (TEND - DNTIME) / (UPTIME - DNTIME)
          DONE = .TRUE.
        ENDIF
      ENDIF
    160 CONTINUE
  IF (NCOUNT .GT. 1) THEN
    OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - OUTPUT (NCOUNT - 1, 4)) / (OUTPUT (NCOUNT, 1) - OUTPUT (NCOUNT - 1, 1))
  ELSE
    OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - CO) / (OUTPUT (NCOUNT, 1) - TO)
  ENDIF
  150 CONTINUE
C......COMPUTE BED VOLUMES FOR VARIABLE VELOCITY CASE

OLDVEL = VEL
TEMPT = 0.00
DO 170 I = 1, NSTEPS
  TEMPT = TEMPT + DT
  IF (OUTPUT (1, 4) .GT. 1.0E-10) THEN
    CURVEL = VEL
  ELSE
    CURVEL = VEL2
  END IF
  CURVEL = VEL2
  AVGVEL = (OLDVEL * (TEMPT - DT) + CURVEL * DT) / TEMPT
  TAU = ZCOL / AVGVEL
  OUTPUT (1, 2) = TEMPT / TAU
  OUTPUT (1, 3) = TEMPT / (TAU * DG)
  OLDVEL = AVGVEL
  CONTINUE
END IF
RETURN
END

THIS SUBROUTINE COMPUTES VARIABLE GROUPINGS TO MINIMIZE EFFORT IN
THE SOLUTION OF THE SET OF SIMULTANEOUS NONLINEAR EQUATIONS.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDIF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS TO BE CALCULATED</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL*8</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL*8</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL*8</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRO_DYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>DZ</td>
<td>REAL*8</td>
<td>COLUMN STEP SIZE IN Z DIRECTION</td>
</tr>
<tr>
<td>FK</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>FN</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>REALND</td>
<td>REAL*8</td>
<td>NUMBER OF NODES IN COLUMN</td>
</tr>
<tr>
<td>RFGRP</td>
<td>REAL*8</td>
<td>RETARDATION FACTOR GROUPING</td>
</tr>
<tr>
<td>RHO</td>
<td>REAL*8</td>
<td>SOLID PHASE PARTICLE DENSITY</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>SOLID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>SPLGRP</td>
<td>REAL*8</td>
<td>SOLID PHASE LOSS GROUP</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
</tbody>
</table>
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/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
1 EPSLON, CO, FK, FN, FKAPPA, SKAPPA, C, T,
2 ICON, NCOL, NOUT, NSTEPS
SPLGRP

THETA, ZCOL, VEL2,

C.....COMPUTE MACROSCOPIC ADVECTIVE-DISPERSIVE EQUATION FLUID PHASE GROUPS
C
REALND = NCOL
DZ = ZCOL / (REALND - 1.000)
DGNLOW = DH / (DZ ** 2.000) + VEL / (2.000 * DZ)
DGN = -2.000 * DH / (DZ ** 2.000) - FKAPPA
DGNUP = DH / (DZ ** 2.000) - VEL / (2.000 * DZ)
RFGRP = FK * FN * (1 - THETA) * RHO / THETA
SPLGRP = (RHO * (1 - THETA) / THETA) * SKAPPA * FK
RETURN
END

THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE
DEPENDENT VARIABLE AS A FUNCTION OF TIME. THE EQUATIONS ARE
SOLVED USING THE VARIABLE GROUPINGS PREVIOUSLY DERIVED IN ORDER
TO SAVE COMPUTATIONAL EFFORT.

ALL DEPENDENT VARIABLE ARE STORED IN A SINGLE ARRAY DV, AND ALL
TEMPORAL DERIVATIVES OF THE DEPENDENT VARIABLE ARE STORED IN A
SINGLE ARRAY TERMED DDVDT.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TEMPORARY VARIABLE</td>
</tr>
<tr>
<td>J</td>
<td>INTEGER*4</td>
<td>TEMPORARY VARIABLE</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDIF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NEND</td>
<td>INTEGER*4</td>
<td>VARIABLE INDEX FOR LAST COLUMN NODE</td>
</tr>
<tr>
<td>NSTART</td>
<td>INTEGER*4</td>
<td>LOOP STARTING POSITION</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS TO BE CALCULATED</td>
</tr>
<tr>
<td>NSTOP</td>
<td>INTEGER*4</td>
<td>LOOP STOPPING POSITION</td>
</tr>
<tr>
<td>DCDT</td>
<td>REAL*8</td>
<td>CHANGE IN INFLUENT CONCENTRATION WITH RESPECT TO TIME</td>
</tr>
<tr>
<td>DDVDT</td>
<td>REAL*8</td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL*8</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL*8</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL*8</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
</tbody>
</table>
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(IOO)
 COMMON /GROUPS/ DGN, DGNLOW, DGNUP, RFGRP, SPLGRP
 COMMON /INPUT/ TMESIM, 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.....FORM FLUID PHASE DERIVATIVES
C
NDOF = NCOL

C.....FIRST THE INLET CONDITION, A DIRICHLET BOUNDARY
C
DDVDT (1) = DCODT

C.....SECOND THE INTERIOR FLUID PHASE DERIVATIVES
C
NEND = NDOF - 1
DO 100 I = 2, NEND
IF (DV (I) .GT. 0.000) THEN
 RF1 = 1.000 + RFGRP * (DV (I) ** (FN - 1.000))
 SPL = SPLGRP * (DV (I) ** FN)
ELSE
 RF1 = 1.000
 SPL = 0.000
ENDIF
100 DDVDT (I) = ((DGNLOW + DGNUP) * DV (I-1) + DGN * DV (I) + DGNUP * DV (I+1)
 1 - SPL) / RF1

C.....THIRD THE OUTLET BOUNDARY CONDITION, A NEUMANN BOUNDARY
C
I = NDOF
IF (DV (I) .GT. 0.000) THEN
 RF1 = 1.000 + RFGRP * (DV (I) ** (FN - 1.000))
 SPL = SPLGRP * (DV (I) ** FN)
ELSE
 RF1 = 1.000
 SPL = 0.000
ENDIF
 DDVDT (I) = ((DGNLOW + DGNUP) * DV (I - 1) + DGN * DV (I)
 1 - SPL) / RF1
RETURN
END
This subroutine computes the values of the derivative of the DDDVT array as a function of the dependent variable (DV). Since the finite difference option is used in the call to DGEAR (I.E. MTER=-2) this routine is a blank.

SUBROUTINE FCNJ (N, X, Y, PD)
REAL*8 Y(N), PD(N,N), X
RETURN
END

This subroutine uses the experimental data input and the model simulation output to compute the model residual, residual squared, sum of residuals, and sum of residuals squared. These computations are output to unit 100OUT.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>IOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NDV</td>
<td>INTEGER*4</td>
<td>INDEX OF OUTLET CONCENTRATION</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF POINTS SIMULATED</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS TO BE CALCULATED</td>
</tr>
<tr>
<td>CNORM</td>
<td>REAL*8</td>
<td>NORMALIZED OUTLET CONCENTRATION</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>RDSQSM</td>
<td>REAL*8</td>
<td>SUM OF RESIDUALS SQUARED</td>
</tr>
<tr>
<td>RSD</td>
<td>REAL*8</td>
<td>RESIDUAL OF POINT AT OUTLET</td>
</tr>
<tr>
<td>RSDSQ</td>
<td>REAL*8</td>
<td>RESIDUAL SQUARED OF POINT AT OUTLET</td>
</tr>
<tr>
<td>RDSUM</td>
<td>REAL*8</td>
<td>SUM OF RESIDUALS</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>SOLID-PHASE LOSS RATE</td>
</tr>
</tbody>
</table>

SUBROUTINE REPORT (I, DV, RDSQSM, RDSQSM, RSD)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 DATA
DIMENSION DATA (100,4), DV(101), C (100), T (100)
COMMON /INPUT/ THESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
```plaintext
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 (1,2) - CNORM
RSDSQ = RSD * RSD
RSDSUM = RSDSUM + RSD
RDSQSM = RDSQSM + RSDSQ
WRITE (IOOUT, 1300) DATA (1,1), DATA (1,2), 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-------------------------------0'),
1 '-----------------------------------',
2 '-------------------------------')
1200 FORMAT ('0',19X,'TIME',10X,'C/Co (INPUT)',10X,'C/Co (PREDICTED)',
1 'RESIDUAL',10X,'RESIDUAL SQUARED'),
1400 FORMAT (' ',81X,E12.5,14X,E12.5)
RETURN
END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE
COMPLETION OF SIMULATION FOR EXPERIMENTAL COMPARISON.

VARIABLE MAP:
-----------

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>LOOP INDEX</td>
</tr>
<tr>
<td>IFLAG1</td>
<td>INTEGER*4</td>
<td>FLAG FOR ABRUPT CONCENTRATION CHANGE</td>
</tr>
<tr>
<td>IFLAG2</td>
<td>INTEGER*4</td>
<td>FLAG FOR DESORPTION</td>
</tr>
<tr>
<td>IOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>BAL</td>
<td>REAL*8</td>
<td>RATIO OF MASS IN TO MASS ACCOUNTED</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
</tbody>
</table>
```

This subroutine performs a mass balance check after the completion of simulation for experimental comparison. The variable map includes variables such as loop index, flag for abrupt concentration change, flag for desorption, unit numbers to read from and write to, number of nodes in z direction, number of temporal steps to output, ratio of mass in to mass accounted, initial concentration, experimental input data, as well as dependent variable vector.
SUBROUTINE MASS1 (OUTCON, DV, CO, ZCOL, THETA, FK, FN, VEL, VEL2, RHO, DATA, NOUT, NCOL)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*4 DATA
DIMENSION DATA (100, 4), DV (101), OUTCON (250)
COMMON /UNITS/ I0IN, I0OUT

IFLAG1 = 0
IFLAG2 = 0

C.....STORE THE INITIAL VELOCITY
VEL1 = VEL

C.....CALCULATE TOTAL CONTAMINANT INJECTED

CALCULATE CONTAMINANT INJECTED DURING FIRST REPORT PERIOD.

TOTINF = CO * DATA (1, 1) * VEL * THETA *
1.0 * DATA (1, 2)
2.0 * DATA (1, 3)

CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS

DO 100 I = 2, (NOUT - 1)
IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
1.0 * DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3)) THEN
IFLAG1 = 1
TOTINF = TOTINF +
1.0 * DATA (I, 3) * (DATA (1, 1) - DATA (I - 1, 1)) * VEL * THETA
ELSE
TOTINF = TOTINF +
1.0 * DATA (1, 3) * (DATA (1, 1) - DATA (I - 1, 1)) * VEL * THETA
2.0 * DATA (1, 1) - DATA (I - 1, 1)
3.0 * DATA (1, 3) - DATA (I - 1, 3)
4.0 * DATA (1, 1) - DATA (I - 1, 1)
ENDIF
100 CONTINUE

C.....CALCULATE TOTAL EFFLUENT RELEASED

CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
THE FIRST REPORT PERIOD

TOTOUT = (0.75 * 0.0 DO + 0.25 * OUTFCON (1)) * 
1 DATA (1, 1) / 2.0 DO * VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
PERIOD

TOTOUT = TOTOUT + 
1 ((0.75 * OUTFCON (1) + 0.25 * 0.0 DO) * 
2 (DATA (1, 1) - DATA (1, 1)) / 2.0 DO + 
3 (0.75 * OUTFCON (1) + 0.25 * OUTFCON (2)) * 
4 (DATA (2, 1) - DATA (1, 1)) / 2.0 DO) * 
5 VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
PERIOD

DO 110 I = 2, (NOUT - 1)

RESET VELOCITY IF DESORPTION BEGINS

IF (DATA (I, 3) .LT. 1.0E-10) THEN
VEL = VEL2
IFLAG2 = 1
ENDIF

TOTOUT = TOTOUT + 
1 ((0.75 * OUTFCON (1) + 0.25 * OUTFCON (1 - 1)) * 
2 (DATA (I, 1) - DATA (I - 1, 1)) / 2.0 DO + 
3 (0.75 * OUTFCON (I) + 0.25 * OUTFCON (I + 1)) * 
4 (DATA (I + 1, 1) - DATA (I, 1)) / 2.0 DO) * 
5 VEL * THETA

110 CONTINUE

CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
REPORT PERIOD

TOTOUT = TOTOUT + 
1 (0.75 * OUTFCON (NOUT) + 
2 0.25 * OUTFCON (NOUT - 1)) * 
3 (DATA (NOUT, 1) - DATA (NOUT - 1, 1)) / 2.0 DO + 
4 VEL * THETA

RESTORE VELOCITY

VEL = VEL1

CALCULATE CONTAMINANT REMAINING IN FLUID PHASE

CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE

RCOL = REAL (NCOL)
TOTWET = (0.75 * DV (1) + 0.25 * DV (2)) * 
1 ZCOL / (RCOL - 1.0 DO) / 2.0 DO * THETA

CALCULATE CONTAMINANT IN FLUID NEAR TYPICAL NODE

DO 120 I = 2, (NCOL - 1)

TOTWET = TOTWET + 
1 (0.75 * DV (1) + 0.125 * DV (1 - 1)) * 
2 0.125 * DV (1 + 1)) * 
3 ZCOL / (RCOL - 1.0 DO) * THETA

120 CONTINUE

CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
TOTWET = TOTWET + (0.25 * DV (NCOL - 1) + 0.75 * DV (NCOL)) * 
1 ZCOL / (RCOL - 1.DO) / 2.DO * THETA

CALCULATE CONTAMINANT REMAINING ON SOLID PHASE

CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE

TOTDRY = FK * (0.75 * DV (1) + 0.25 * DV (2)) * FN *
1 ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO - THETA)

CALCULATE CONTAMINANT ON TYPICAL NODES

DO 130 1 = 2, (NCOL - 1)

TOTDRY = TOTDRY + FK *
1 (0.75 * DV (I) + 0.125 * DV (I - 1) + 0.125 * DV (I + 1)) * FN *
2 ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)

130 CONTINUE

CALCULATE CONTAMINANT NEAR EXIT

TOTDRY = TOTDRY *
1 FK * (0.25 * DV (NCOL - 1) + 0.75 * DV (NCOL)) * FN *
2 ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)

COMPUTE MASS BALANCE

BAL = TOTINF / (TOTOUT + TOTWET + TOTDRY)

OUTPUT THE MASS BALANCE FINDINGS

WRITE (I0OUT, 1000)
WRITE (I0OUT, 1100)
WRITE (I0OUT, 1200) TOTINF
WRITE (I0OUT, 1300) TOTOUT
WRITE (I0OUT, 1400) TOTWET
WRITE (I0OUT, 1500) TOTDRY
WRITE (I0OUT, 1600) BAL
IF (IFLAG1 .EQ. 1) THEN
WRITE (I0OUT, 1700)
ENDIF
IF (IFLAG2 .EQ. 1) THEN
WRITE (I0OUT, 1800)
ENDIF

1000 FORMAT ('1', 35X, 'EXPERIMENTAL MASS BALANCE REPORT')
1100 FORMAT ('---------------------------------------------', 1)
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 DESORPTION.')

RETURN
END

THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE 
COMPLETION OF SIMULATION IN SIMULATION MODE.

VARIABLE MAP:

---

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>LOOP INDEX</td>
</tr>
<tr>
<td>IFLAG1</td>
<td>INTEGER*4</td>
<td>FLAG FOR ABRUPT CONCENTRATION CHANGE</td>
</tr>
<tr>
<td>IFLAG2</td>
<td>INTEGER*4</td>
<td>FLAG FOR DESORPTION</td>
</tr>
<tr>
<td>ION</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS</td>
</tr>
<tr>
<td>BAL</td>
<td>REAL*8</td>
<td>RATIO OF MASS IN TO MASS ACCOUNTED</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>FK</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FN</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>OUTCON</td>
<td>REAL*8</td>
<td>EXIT CONCENTRATION VECTOR</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*4</td>
<td>SIMULATION STATUS MATRIX</td>
</tr>
<tr>
<td>RCOL</td>
<td>REAL*8</td>
<td>COL 1 GIVES REPORT TIMES</td>
</tr>
<tr>
<td>Theta</td>
<td>REAL*8</td>
<td>COL 2 GIVES BED VOLUMES</td>
</tr>
<tr>
<td>TOTDRY</td>
<td>REAL*8</td>
<td>COL 3 GIVES THROUGHPUT</td>
</tr>
<tr>
<td>TOTINF</td>
<td>REAL*8</td>
<td>COL 4 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td>TOTOUT</td>
<td>REAL*8</td>
<td>COL 5 GIVES DCD/DT</td>
</tr>
<tr>
<td>TOTVET</td>
<td>REAL*8</td>
<td>REAL EQUIVALENT OF NCOL</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>TOTAL MASS LEFT ON SOLID PHASE</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>TOTAL MASS INJECTED INTO COLUMN</td>
</tr>
<tr>
<td></td>
<td>REAL*8</td>
<td>TOTAL MASS TO EXIT COLUMN</td>
</tr>
<tr>
<td></td>
<td>REAL*8</td>
<td>TOTAL MASS LEFT IN LIQUID PHASE</td>
</tr>
<tr>
<td></td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td></td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td></td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (2 DIRECTION)</td>
</tr>
</tbody>
</table>

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, IOUT

IFLAG1 = 0
IFLAG2 = 0

C.....STORE THE INITIAL VELOCITY
VEL1 = VEL

C.....CALCULATE TOTAL CONTAMINANT INJECTED

CALCULATE CONTAMINANT INJECTED DURING FIRST REPORT PERIOD.

TOTINF = CO * OUTPUT (1, 1) * VEL * THETA +
          0.5 * (OUTPUT (1, 4) - CO) * OUTPUT (1, 1) *
          VEL * THETA

CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS

DO 100 I = 2, (NSTEPS - 1)
  IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1),4)) .OR.
     (OUTPUT (1, 4) .GT. 1.2 * OUTPUT ((I - 1),4))) THEN
    IFLAG1 = 1
    TOTINF = TOTINF +
              OUTPUT (I, 4) *
              (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
              VEL * THETA
  ELSE
    TOTINF = TOTINF +
              OUTPUT (I - 1, 4) *
              (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
              VEL * THETA +
              0.5 * (OUTPUT (I, 4) - OUTPUT (I - 1, 4)) *
              (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) *
              VEL * THETA
  END IF
100  CONTINUE

C.....CALCULATE TOTAL EFFLUENT RELEASED

CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF THE FIRST REPORT PERIOD

TOTOUT = (0.75 * O.DO + 0.25 * OUTCON (1)) *
         OUTPUT (1, 1) / 2.DO * VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT PERIOD

TOTOUT = TOTOUT +
         ((0.75 * OUTCON (1) + 0.25 * O.DO) *
          (OUTPUT (1, 1) - O.DO) / 2.DO +
          (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
          (OUTPUT (2, 1) - OUTPUT (1, 1)) / 2.DO) *
          VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT PERIOD

DO 110 I = 2, (NSTEPS - 1)
C RESET VELOCITY IF DESORPTION BEGINS
IF (OUTPUT (1, 4) .LT. 1.0E-10) THEN
VEL = VEL2
IFLAG2 = 1
ENDIF

TOTOUT = TOTOUT +
1 (0.75 * OUTCON (1) + 0.25 * OUTCON (1 - 1)) *
2 (OUTPUT (1, 1) - OUTPUT (1 - 1, 1)) / 2.DO *
3 (0.75 * OUTCON (1) + 0.25 * OUTCON (1 + 1)) *
4 (OUTPUT (1 + 1, 1) - OUTPUT (1, 1)) / 2.DO *
5 VEL * THETA

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.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C     CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
RCOL = REAL (NCOL)
TOTWET = (0.75 * DV (1) + 0.25 * DV (2)) *
1 ZCOL / (RCOL - 1.DO) / 2.DO * THETA

C
C     CALCULATE CONTAMINANT IN FLUID NEAR TYPICAL NODE
DO 120 I = 2, (NCOL - 1)
   TOTWET = TOTWET +
   1 (0.75 * DV (1) + 0.125 * DV (1 - 1) +
   2 0.125 * DV (1 + 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 - 1) + 0.75 * DV (NCOL)) *
1 ZCOL / (RCOL - 1.DO) / 2.DO * THETA

C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE
C
C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE
TOTDRY = FK * (0.75 * DV (1) + 0.25 * DV (2)) ** FN *
1 ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO - THETA)

C
C     CALCULATE CONTAMINANT ON TYPICAL NODES
DO 130 I = 2, (NCOL - 1)
   TOTDRY = TOTDRY + FK *
   1 (0.75 * DV (1) + 0.125 * DV (1 - 1) +
   2 0.125 * DV (1 + 1)) ** FN *
CONTINUE

CALCULATE CONTAMINANT NEAR EXIT

TOTDRY = TOTDRY +

1 FK * (0.25 * DV (NCOL - 1) + 0.75 * DV (NCOL)) * FN
2 * ZCOL / (RCOL - l.DO) / 2.DO * RHO * (1.DO - THETA)

.. COMPUTE MASS BALANCE

BAL = TOTINF / (TOTOUT + TOTWET + TOTDRY)

.. OUTPUT THE MASS BALANCE FINDINGS

WRITE (lOOUT, 1000)
WRITE (lOOUT, 1100)
WRITE (lOOUT, 1200) TOTINF
WRITE (lOOUT, 1300) TOTOUT
WRITE (lOOUT, 1400) TOTWET
WRITE (lOOUT, 1500) TOTDRY
WRITE (lOOUT, 1600) BAL
BAR (IFLAG1 .EQ. 1) THEN
  WRITE (lOOUT, 1700)
END IF
BAR (IFLAG2 .EQ. 1) THEN
  WRITE (lOOUT, 1800)
END IF

1000 FORMAT ('1', 35X, 'SIMULATION MASS BALANCE REPORT ')
1100 FORMAT ('Q-----------------------------',
             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 DESORPTION.')

RETURN
END

THE EQUATIONS ARE APPROXIMATED USING THE FINITE DIFFERENCE METHOD RESULTING IN A SET OF DQ/DT AND DC/DT EQUATIONS. THESE ARE SOLVED USING DGEAR --- AN IMSL SUBROUTINE.

THE PROGRAM IS PRESENTLY LIMITED TO 100 NODES.

**VARIABLE MAP:**

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>IDUMMY</td>
<td>INTEGER4</td>
<td>DUMMY WORKING ARRAY FOR DGEAR</td>
</tr>
<tr>
<td>IER</td>
<td>INTEGER4</td>
<td>ERROR CODE RETURN INDEX FOR DGEAR</td>
</tr>
<tr>
<td>INDEX</td>
<td>INTEGER4</td>
<td>CALL INDICATOR INDEX FOR GEAR</td>
</tr>
<tr>
<td>IOIN</td>
<td>INTEGER4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOUT</td>
<td>INTEGER4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>IWK</td>
<td>INTEGER4</td>
<td>WORKING ARRAY FOR DGEAR</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NLC</td>
<td>INTEGER4</td>
<td>NUMBER OF LOWER CODIAGONALS-DGEAR</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>NUC</td>
<td>INTEGER4</td>
<td>NUMBER OF UPPER CODIAGONALS-DGEAR</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER4</td>
<td>NUMBER OF BED VOLUME STEP TO OUTPUT</td>
</tr>
<tr>
<td>B</td>
<td>REAL8</td>
<td>LANGMUIR ENERGY CONSTANT</td>
</tr>
<tr>
<td>CO</td>
<td>REAL8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>CNORM</td>
<td>REAL8</td>
<td>NORMALIZED EXIT CONCENTRATION</td>
</tr>
<tr>
<td>CPRIME</td>
<td>REAL8</td>
<td>EXTERNAL NAME OF DC/DT SUBROUTINE</td>
</tr>
<tr>
<td>CTEMP</td>
<td>REAL8</td>
<td>EXIT CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td>DDVDT</td>
<td>REAL8</td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL8</td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL8</td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL8</td>
<td>COLUMN 4 GIVES DC/DT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
</tbody>
</table>
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 SDUMMY, DATA, OUTPUT
CHARACTER*72 TITLE

DIMENSION F(100), X(1)
DIMENSION IWK (101)
DIMENSION DATA (100,4), DV (101), DDVDT (101), OUTPUT (250,5),
1 UK (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/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
1 EPSLON, CO, QQ, B, SOK, FKAPPA, SKAPPA, C, T,
2 ICON, NCOL, NOUT, NSTEPS
COMMON /WORDS/ TITLE
COMMON /FLUID/ DCODT
COMMON /UNITS/ I0IN, I0OUT
EXTERNAL CPRiME, FCNJ
I0IN = 1
I0OUT = 3
CALL HEAD
VEL1 = VEL
DH1 = DH
CALL ECHO
CALL ERRSET (208, 256, -1, 1)

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.D0
H = 1.D-08
INDEX = 1
DV (1) = CO
NLC = 2
NUC = 2
C
C.....TIME LOOP FOR SIMULATION 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 (10OUT, 1300)
      IF (DV (1) .LT. 1.OE-10) THEN
        VEL = VEL2
        DH = DH1 * VEL2 / VEL1
        CALL GRPFRM
        WRITE (10OUT, 1400)
      END IF.
    ELSE
      DCODT = DATA (1, 4)
      ENDIF
  ELSE
    DCODT = DATA (1, 4)
    ENDIF
  TIME = DATA (1, 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
  ENDF
C...compute model output at calculated points if desired

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 = NDOF - 1
  TO = 0.0
  H = 1.D-08
  INDEX = 1
  DV (1) = CO
  NLC = 2
  NUC = 2
C
C...time loop for simulation and output at computed points
C
  WRITE (IOUT, 1000)
  WRITE (IOUT, 1100)
  DO 110 I = 1, NSTEPS
   IF (I .GT. 1) THEN
    IF ((OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1), 4)) .OR.
     (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 (IOUT, 1300)
      IF (DV (1) .LT. 1.E-10) THEN
       VEL = VEL2
       DH = DH1 * VEL2 / VEL1
       CALL GRPFRM
       WRITE (IOUT, 1400)
      END IF
      ELSE
       DCODT = OUTPUT (I, 5)
      END IF
   ELSE
    DCODT = OUTPUT (I, 5)
   END IF
   TIME = OUTPUT (I, 1)
   CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
    EPSLON, 2, -2, INDEX, IWK, WK, IER)
   CTEMP = DV (NCOLEX)
   CNORM = CTEMP / CO
   OUTCON (I) = DV (NCOLEX)
  110   WRITE (IOUT, 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,
   RH0, OUTPUT, NSTEPS, NCOL)
C
END IF

1000 FORMAT ('!1$, 42X, 'SIMULATION OUTPUT FOR PLOTTING')
1100 FORMAT ('0-----------------------------',
1 '-------------------------------------------------------------',
2 '-------------------------------------------------------------')
THIS SUBROUTINE READS THE BASIC INPUT REQUIRED FOR THE SECOND-ORDER SORPTION VERSION OF THE ADVECTIVE-DISPERSIVE EQUATION.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICON</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF BED VOLUME STEP TO OUTPUT</td>
</tr>
<tr>
<td>B</td>
<td>REAL*8</td>
<td>LANGMUIR ENERGY CONSTANT</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>EPSLON</td>
<td>REAL*8</td>
<td>ALLOWABLE ERROR FOR DGEAR</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>QO</td>
<td>REAL*8</td>
<td>LANGMUIR CAPACITY CONSTANT</td>
</tr>
<tr>
<td>RHO</td>
<td>REAL*8</td>
<td>SOLID PHASE PARTICLE DENSITY</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>SOLID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>SOK</td>
<td>REAL*8</td>
<td>SECOND-ORDER RATE CONSTANT</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>THESIM</td>
<td>REAL*8</td>
<td>NUMBER OF BED VOLUMES TO SIMULATE</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
<tr>
<td>TITLE</td>
<td>CHARACTER*72</td>
<td>TITLE OF SIMULATION RUN</td>
</tr>
</tbody>
</table>

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,
               EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
               ICON, NCOL, NOUT, NSTEPS
COMMON /WORDS/ TITLE
COMMON /UNITS/ IOIN, IOOUT
READ (IOIN, 1000) TITLE
READ (IOIN, 1100) NCOL, NOUT, NSTEPS, TMESIM, ICON
IF (ABS (NOUT) .GT. 0) THEN
   NTEMP = ABS (NOUT)
   READ (IOIN, 1200) DATA (1,1), I = 1, NTEMP)
   READ (IOIN, 1200) DATA (1,2), I = 1, NTEMP)
ENDIF
READ (IOIN, 1200) VEL, DH, RHO, THETA, ZCOL, EPSLON
READ (IOIN, 1200) CO, QQ, B, SOK, FKAPPA, SKAPPA, VEL2
READ (IOIN, 1200) (C (I), I = ICON)
READ (IOIN, 1200) (T (I), I = ICON)
1000 FORMAT (A72)
1100 FORMAT (315, E12.5, 15)
1200 FORMAT ((6E12.5))
RETURN
END

THIS SUBROUTINE ECHOS THE INPUT TO DEVICE IOOUT.

VARIABLE MAP:
---------------------
<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>B</td>
<td>REAL*8</td>
<td>LANGMUIR ENERGY CONSTANT</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>EPSLON</td>
<td>REAL*8</td>
<td>ALLOWABLE ERROR FOR DGEAR</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>PE</td>
<td>REAL*8</td>
<td>SYSTEM PECLET NUMBER</td>
</tr>
<tr>
<td>QQ</td>
<td>REAL*8</td>
<td>LANGMUIR CAPACITY CONSTANT</td>
</tr>
<tr>
<td>RHO</td>
<td>REAL*0</td>
<td>SOLID PHASE PARTICLE DENSITY</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>SOLID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>SOK</td>
<td>REAL*8</td>
<td>SECOND-ORDER RATE CONSTANT</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>THESIM</td>
<td>REAL*0</td>
<td>NUMBER OF BED VOLUMES TO SIMULATE</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
<tr>
<td>TITLE</td>
<td>CHARACTER*72</td>
<td>TITLE OF SIMULATION RUN</td>
</tr>
</tbody>
</table>

SUBROUTINE ECHO
IMPLICIT REAL*8 (A - H.O - Z)
REAL DATA
CHARACTER TITLE
DIMENSION DATA (100,4), C (100), T (100)
COMMON /INPUT/ TMESIM, 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/ I0IN, I0OUT
PE = VEL / ZCOL / DH
WRITE (I0OUT, 1000)
WRITE (I0OUT, 1100)
WRITE (I0OUT, 1200) TITLE
WRITE (I0OUT, 1300)
WRITE (I0OUT, 1400)
WRITE (I0OUT, 1400)
WRITE (I0OUT, 1500) CO
WRITE (I0OUT, 1600) DH
WRITE (I0OUT, 1700) NCOL
WRITE (I0OUT, 1800) RHO
WRITE (I0OUT, 1900) VEL
WRITE (I0OUT, 1950) B
WRITE (I0OUT, 2000) SOK
WRITE (I0OUT, 2100) FKAPPA
WRITE (I0OUT, 2200) THETA
WRITE (I0OUT, 2300) ZCOL
WRITE (I0OUT, 2400) PE
WRITE (I0OUT, 2500) QO
WRITE (I0OUT, 2500) B
WRITE (I0OUT, 2600) SOK
WRITE (I0OUT, 2700) SKAPPA
WRITE (I0OUT, 2800) ION
WRITE (I0OUT, 2900)
WRITE (I0OUT, 3000) (C (I), I = 1, ICON)
WRITE (I0OUT, 3100)
WRITE (I0OUT, 3200) (T (I), I = 1, ICON)
1000 FORMAT ("1", 50X, 'PLUG FLOW DISPERSION REACTOR')
1100 FORMAT (' ', 47X, 'LANGMUIR SECOND ORDER KINETIC SIMULATION')
1200 FORMAT (' ', 26X, 'DESCRIPTION', 16X, 'VALUE', 16X)
1300 FORMAT (' ', 15X, 'UNITS', 16X)
1400 FORMAT (''
1500 FORMAT ('O', 15X, 'CO', 16X, 'INITIAL CONCENTRATION', 14X, 1E10.4, 16X, 'G/CM*3')
1600 FORMAT (' ', 15X, 'DH', 18X, 'HYDRODYNAMIC DISPERSION', 14X, 1E10.4, 16X, 'CM/HOUR')
1700 FORMAT (' ', 15X, 'NCOL', 18X, 'NUMBER OF COLUMN STEPS', 16X, 15)
1800 FORMAT (' ', 15X, 'RHO', 18X, 'SOLID PHASE DENSITY', 14X, 1E10.4, 16X, 'G/CM*3')
1900 FORMAT (' ', 15X, 'VEL', 18X, 'PORE VELOCITY', 14X, 1E10.4, 16X, 'CM/HOUR')
2000 FORMAT (' ', 15X, 'THETA', 18X, 'VOID VOLUME FRACTION', 14X, 1E10.4)
2100 FORMAT (' ', 15X, 'ZCOL', 18X, 'LENGTH OF COLUMN', 14X, 1E10.4, 16X, 'CM')
2200 FORMAT (' ', 15X, 'PE', 18X, 'PECLET NUMBER', 14X, 1E10.4)
2300 FORMAT (' ', 15X, 'QO', 18X, 'LANGMUIR CAPACITY CONSTANT', 14X, 1E10.4, 16X, 'G/G')
2500 FORMAT (' ', 15X, 'SOK', 18X, 'SECOND-ORDER RATE CONSTANT', 14X, 1E10.4, 16X, 'CM#3/G-HOUR')
THIS SUBROUTINE INITIALIZES THE MODEL VARIABLES FOR THE SIMULATION
OF THE ADVECTIVE-DISPERSIVE EQUATION.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>WORKING VARIABLE</td>
</tr>
<tr>
<td>J</td>
<td>INTEGER*4</td>
<td>WORKING VARIABLE</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS TO BE CALCULATED</td>
</tr>
<tr>
<td>B</td>
<td>REAL*8</td>
<td>LANGMUIR ENERGY CONSTANT</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*8</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td>DDVDT</td>
<td>REAL*8</td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td>DG</td>
<td>REAL*8</td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td>DT</td>
<td>REAL*8</td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td>DTN</td>
<td>REAL*8</td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>RATIO OF TOTAL MASS ADSORBABLE TO TOTAL MASS IN</td>
</tr>
<tr>
<td>EPSMCN</td>
<td>REAL*8</td>
<td>FLUID PHASE</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>SOLID PHASE</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*4</td>
<td>DURATION OF EACH TIME STEP</td>
</tr>
<tr>
<td>QO</td>
<td>REAL*8</td>
<td>REAL EQUIVALENT OF NSTEPS</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>SOK</td>
<td>REAL*8</td>
<td>MACHINE EPSILON</td>
</tr>
<tr>
<td>TAU</td>
<td>REAL*8</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>TEMP</td>
<td>REAL*8</td>
<td>A MATRIX OF SIMULATION STATUS</td>
</tr>
<tr>
<td>TEPHT</td>
<td>REAL*8</td>
<td>COL 1 CONTAINS REPORT TIME</td>
</tr>
<tr>
<td>TESIM</td>
<td>REAL*8</td>
<td>COL 2 CONTAINS BED VOLUMES</td>
</tr>
<tr>
<td>QIN</td>
<td>REAL*8</td>
<td>COL 3 CONTAINS THROUGHPUT</td>
</tr>
<tr>
<td>CONC</td>
<td>REAL*8</td>
<td>COL 4 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td>TIMES</td>
<td>REAL*8</td>
<td>COL 5 GIVES DCO/DT</td>
</tr>
<tr>
<td>QIN</td>
<td>REAL*8</td>
<td>NUMBER OF BED VOLUMES TO SIMULATE</td>
</tr>
</tbody>
</table>
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, SQK, FKAPPA, SKAPPA, C, T,
2 ICON, NCOL, NOUT, NSTEPS
C.....initialize dependent variable and derivative vectors
C
DO 100 I = 1, 101
DV (I) = 0.00
100 DDVDT (I) = 0.00
C.....compute simulation output points if desired
C
IF (NSTEPS .GT. 0) THEN
TAU = ZCOL / VEL
DG = 1 + (RHO * (1 - THETA) / THETA) * ((QO * B * CO) / (1 + B * CO)) / CO
DTN = NSTEPS
DT = THESIM / DTN
TEMPT = 0.00
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.....calculate the machine epsilon for use as a lower error bound
C
EPSMCN = 1.0
120 EPSMCN = EPSMCN / 2.000
TEMP = 1.000 + EPSMCN
IF (TEMP .GT. 1.000) GO TO 120
C.....calculate the influent concentrations for use in the experimental comparison mode by interpolating from the 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)
1 IF ((DNTIME .LE. TEND) .AND. (UPTIME .GE. TEND)) THEN
DATA (NCOUNT, 3) = DNCON * (UCON - DNCON) / (UPTIME - DNTIME)
1 = .TRUE.
ENDIF
ENDIF
CONTINUE
DO 130 NCOUNT = 1, NOUT, 1
DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) -
ELSE
130  DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) - CO) / (DATA (NCOUNT, 1) - TO)
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
END IF
CONTINUE
RETURN
END

C

CC...CALCULATE INFLUENT CONCENTRATION FOR USE IN THE
CC...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)) THEN
OUTPUT (NCOUNT, 4) = DNCON + (UPCON - DNCON) + (TEND - DNTIME) / (UPTIME - DNTIME)
DONE = .TRUE.
END IF
ENDIF
160 CONTINUE
IF (NCOUNT .GT. 1) THEN
OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - OUTPUT (NCOUNT, 1)) / (OUTPUT (NCOUNT, 1) - TO)
ELSE
OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - CO)
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
END IF
RETURN
END

C

C

C

CC THIS SUBROUTINE WRITES THE CONTENTS OF THE DATA MATRIX TO
CC UNIT 100 OUT.
VARIABLE MAP:
----------------

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>INTEGER*4</td>
<td>LOOP COUNTER</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
</tbody>
</table>

SUBROUTINE INFO (DATA, NOUT)
REAL*4 DATA
DIMENSION DATA (100,4)
COMMON /UNITS/ IOIN, IOUT
WRITE (IOUT, 1000)
WRITE (IOUT, 1100)
WRITE (IOUT, 1200)
WRITE (IOUT, 1100)
DO 100 NCOUNT = 1, NOUT
WRITE (IOUT, 1300) ( DATA (NCOUNT, I), I = 1, 4)
100 CONTINUE
1000 FORMAT ('1', 42X, 'EXPERIMENTAL COMPARISON MODE INFORMATION')
1100 FORMAT ('0'-----------------------------------------------',
           '--------------------------------------------------------',
           '--------------------------------------------------------',
           2 '--------------------------------------------------------'
1200 FORMAT (' ', 28X, 'TIME', 5X, 'CONCENTRATION OUT', 6X,
           'CONCENTRATION IN', 16X, 'DCO/DT')
1300 FORMAT (' ', 20X, E12.5, 10X, E12.5, 10X, E12.5, 10X, E12.5)
RETURN
END

VARIABLE MAP:
--------------

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>INTEGER*4</td>
<td>LOOP COUNTER</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS TO BE CALCULATED</td>
</tr>
</tbody>
</table>
SUBROUTINE INFO2 (OUTPUT, NSTEPS)

REAL*4 OUTPUT
DIMENSION OUTPUT (250, 5)
COMMON /UNITS/ IOIN, IOUT
WRITE (IOUT, 1000)
WRITE (IOUT, 1100)
WRITE (IOUT, 1200)
WRITE (IOUT, 1300)
DO 100 NCOUNT = 1, NSTEPS
WRITE (IOUT, 1300) (OUTPUT (NCOUNT, I), I = 1, 5)
100  CONTINUE
1000 FORMAT ('!', 48X, 'SIMULATION MODE INFORMATION')
1100 FORMAT ('-----------------------------------------------',
1             ' ', 'REPORT TIME', IIX, 'BED VOLUMES', 12X,'THROUGHPUT',
2             ' ', 'CONCENTRATION IN',
2             ' ', 'DCO/DT')
1200 FORMAT (' ', IIX, E12.5, lOX, E12.5, lOX, E12.5, lOX, E12.5, lOX)
1300 FORMAT (' ', IIX, E12.5, 10X, E12.5, 10X, E12.5, 10X,
1             ' ', E12.5)
RETURN
END

THIS SUBROUTINE COMPUTES VARIABLE GROUPINGS TO MINIMIZE EFFORT IN
THE SOLUTION OF THE SET OF SIMULTANEOUS NONLINEAR EQUATIONS.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO SIMULATE</td>
</tr>
<tr>
<td>B</td>
<td>REAL*8</td>
<td>LANGMUIR ENERGY CONSTANT</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL*8</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGNLOV</td>
<td>REAL*8</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL*8</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>DZ</td>
<td>REAL*8</td>
<td>COLUMN STEP SIZE IN Z DIRECTION</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
</tbody>
</table>
SUBROUTINE GRPFRM
IMPLICIT REAL*8(A-H, O-Z)
REAL*8 DATA
DIMENSION DATA (100,4), C (100), T (100)
COMMON /GROUPS/ DGN, DGNLOU, DGNUP
COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
1 EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
2 ICON, NCOL, NOUT, NSTEPS
C.....COMPUTE MACROSCOPIC ADVECTIVE-DISPERSIVE EQUATION FLUID PHASE GROUP INS
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

THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE
DEPENDENT VARIABLE AS A FUNCTION OF TIME. THE EQUATIONS ARE
SOLVED USING THE VARIABLE GROUPINGS PREVIOUSLY DERIVED IN ORDER
TO SAVE COMPUTATIONAL EFFORT.

ALL DEPENDENT VARIABLE ARE STORED IN A SINGLE ARRAY DV, AND ALL
TEMPORAL DERIVATIVES OF THE DEPENDENT VARIABLE ARE STORED IN A
SINGLE ARRAY TERMED DDVDT. THE INDEX NUMBERING OF THESE ARRAYS
MAY BE DESCRIBED AS FOLLOWS:
A) THE FIRST ENTRY (1) IS THE INLET NODE OF THE COLUMN.
B) THE SECOND ENTRY (2) IS LEFT BLANK.
C) THE SUCCEEDING REFERENCES TO THE FLUID PHASE ARE AT THE
ODD-NUMBERED INDICES.
D) THE SOLID PHASE REFERENCES ARE AT THE EVEN-NUMBERED INDEX
FOLLOWING THE ASSOCIATED FLUID REFERENCES.
E) THE TOTAL DEGREES OF FREEDOM ARE EQUAL TO 2 * NCOL.
F) THE NEXT-TO-LAST ENTRY IS THE FLUID PHASE EXIT CON'N.
G) THE LAST ENTRY IS THE SOLID PHASE CONCENTRATION AT THE EXIT.
VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>INTEGER</td>
<td>TEMPORARY VARIABLE</td>
</tr>
<tr>
<td>c</td>
<td>INTEGER</td>
<td>TEMPORARY VARIABLE</td>
</tr>
<tr>
<td>C</td>
<td>INTEGER</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>c</td>
<td>INTEGER</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>C</td>
<td>INTEGER</td>
<td>VARIABLE INDEX FOR LAST COLUMN NODE</td>
</tr>
<tr>
<td>c</td>
<td>INTEGER</td>
<td>LOOP STARTING POSITION</td>
</tr>
<tr>
<td>c</td>
<td>INTEGER</td>
<td>STEP SIZE FOR LOOP</td>
</tr>
<tr>
<td>c</td>
<td>INTEGER</td>
<td>LOOP STOPPING POSITION</td>
</tr>
<tr>
<td>B</td>
<td>REAL</td>
<td>LANGMUIR ENERGY CONSTANT</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCODT</td>
</tr>
<tr>
<td>DDVDT</td>
<td>REAL</td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DV</td>
<td>REAL</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL</td>
<td>FLUID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>FLSP</td>
<td>REAL</td>
<td>FLUID-PHASE LOSS GROUPING</td>
</tr>
<tr>
<td>FLDSP</td>
<td>REAL</td>
<td>FLUID-PHASE LOSS GROUPING</td>
</tr>
<tr>
<td>QO</td>
<td>REAL</td>
<td>LANGMUIR CAPACITY CONSTANT</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL</td>
<td>SOLID-PHASE LOSS RATE</td>
</tr>
<tr>
<td>SLDSP</td>
<td>REAL</td>
<td>SOLID-PHASE LOSS GROUPING</td>
</tr>
<tr>
<td>SOK</td>
<td>REAL</td>
<td>SECOND-ORDER RATE CONSTANT</td>
</tr>
<tr>
<td>TMESIM</td>
<td>REAL</td>
<td>NUMBER OF BED VOLUMES TO SIMULATE</td>
</tr>
</tbody>
</table>

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/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2, EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
2 ICON, NCOL, NOUT, NSTEPS
COMMON /FLUID/ DCODT

C.....FORM FLUID PHASE DERIVATIVES
C
NDOF = NCOL * 2
C.....FIRST THE INLET CONDITION
C
DDVDT (1) = DCODT
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..... FORM THE OUTLET BOUNDARY CONDITION FOR THE FLUID PHASE

1 = 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..... FORM THE SOLID PHASE EQUATIONS

DO 110 I = 4, NDOF, 2

110 DDVDT (I) = SOK * (QO - DV (I)) * DV (I-1) - SLDSP * DV (I)

RETURN

END

C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE
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 (I.E. MITER = -2) THIS ROUTINE IS A BLANK.

C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C SUBROUTINE FCNJ (N, X, Y, PD)
REAL*8 Y(N), PD(N,N), X
RETURN
END

C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C THIS SUBROUTINE USES THE EXPERIMENTAL DATA INPUT AND THE MODEL
C SIMULATION OUTPUT TO COMPUTE THE MODEL RESIDUAL, RESIDUAL SQUARED,
C SUM OF RESIDUALS, AND SUM OF RESIDUALS SQUARED. THESE COMPUTATIONS
C ARE OUTPUT TO UNIT I0OUT.

C VARIABLE MAP:

-----
<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>I0IN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NDV</td>
<td>INTEGER*4</td>
<td>INDEX OF OUTLET CONCENTRATION</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF POINTS SIMULATED</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF STEPS TO SIMULATE</td>
</tr>
<tr>
<td>B</td>
<td>REAL*8</td>
<td>LANGMUIR ENERGY CONSTANT</td>
</tr>
<tr>
<td>CNORM</td>
<td>REAL*8</td>
<td>NORMALIZED OUTLET CONCENTRATION</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT</td>
</tr>
</tbody>
</table>
SUBROUTINE REPORT (I, DV, RSDSUM, RDSQSM, RSD)
IMPLICIT REAL*8 (A-H, O-Z)

REAL*8 DATA
DIMENSION DATA (100,4), DV(101), C (100), T (100)
COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
1 EPSLON, CO, QO, B, SOK, FKAPPA, SKAPPA, C, T,
2 ICON, NCOL, NOUT, NSTEPS
COMMON /UNITS/ lOIN, lOOUT

NDV = (NCOL » 2) - 1
IF (I .EQ. 1) THEN
WRITE (1OOUT, 1000)
WRITE (1OOUT, 1100)
WRITE (1OOUT, 1200)
WRITE (1OOUT, 1100)
RSDSUM = 0.D0
RDSQSM = 0.D0
ENDIF
CNORM = DV (NDV) / CO
RSD = DATA (1,2) - CNORM
RSDSQ = RSD * RSD
RSDSUM = RSDSUM + RSD
RDSQSM = RDSQSM + RSDSQ
WRITE (1OOUT, 1300) DATA (1,1), DATA (1,2), CNORM, RSD, RSDSQ
IF (I .GE. NOUT) THEN
WRITE (1OOUT, 1400) RSDSUM, RDSQSM
END IF
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')
1400 FORMAT ('*',81X,E12.5,14X,E12.5)
RETURN
END

This subroutine performs a mass balance check after the completion of simulation for experimental comparison.

variable map:
### VARIABLE NAME | PRECISION | DESCRIPTION
--- | --- | ---
I | INTEGER*4 | LOOP INDEX
IFLAG1 | INTEGER*4 | FLAG FOR ABRUPT CONCENTRATION CHANGE
IFLAG2 | INTEGER*4 | FLAG FOR DESORPTION
I0IN | INTEGER*4 | UNIT NUMBER TO READ FROM
I0OUT | INTEGER*4 | UNIT NUMBER TO WRITE TO
NCELL | INTEGER*4 | NUMBER OF NODES IN Z DIRECTION
NOUT | INTEGER*4 | NUMBER OF TEMPORAL STEPS TO OUTPUT
BAL | REAL*8 | RATIO OF MASS IN TO MASS ACCOUNTED
CO | REAL*8 | INITIAL CONCENTRATION
DATA | REAL*4 | EXPERIMENTAL INPUT DATA
DV | REAL*8 | DEPENDENT VARIABLE VECTOR
OUTCON | REAL*8 | EXIT CONCENTRATION VECTOR
RCOL | REAL*8 | REAL EQUIVALENT OF NCOL
THETA | REAL*8 | POROSITY
TOTDRY | REAL*8 | TOTAL MASS LEFT ON SOLID PHASE
TOTINF | REAL*8 | TOTAL MASS INJECTED INTO COLUMN
TOTOUT | REAL*8 | TOTAL MASS TO EXIT COLUMN
TOTWET | REAL*8 | TOTAL MASS LEFT IN LIQUID PHASE
VEL | REAL*8 | PORE VELOCITY
VEL2 | REAL*8 | DESORPTION PORE VELOCITY
ZCOL | REAL*8 | LENGTH OF COLUMN (Z DIRECTION)

---

```fortran
SUBROUTINE MASS1 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2, 
                  RHO, DATA, NOUT, NCOL)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 DATA
DIMENSION DATA (100, 4), DV (101), OUTCON (250)
COMMON /UNITS/ I0IN, I0OUT

IFLAG1 = 0
IFLAG2 = 0

C.....STORE THE INITIAL VELOCITY
VEL1 = VEL

C.....CALCULATE TOTAL CONTAMINANT INJECTED

CALCULATE CONTAMINANT INJECTED DURING FIRST REPORT PERIOD.

TOTINF = CO * DATA (1, 1) * VEL * THETA
1   0.5 * (DATA (1, 3) - CO) * DATA (1, 1) * 
2   VEL * THETA
```
CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS

DO 100 I = 2, (NOUT - 1)
  IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
    (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
    IFLAG1 = 1
    TOTINF = TOTINF +
    DATA (I, 3) * (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
  ELSE
    TOTINF = TOTINF +
    DATA (I, 3) * (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA +
    0.5 * (DATA (I, 3) - DATA (I - 1, 3)) * (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
  ENDIF
100 CONTINUE

CALCULATE TOTAL EFFLUENT RELEASED

CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF THE FIRST REPORT PERIOD

TOTOUT = (0.75 * O.DO + 0.25 * OUTCON (1)) *
DATA (1, 1) / 2.DO * VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT PERIOD

TOTOUT = TOTOUT +
((0.75 * OUTCON (1) + 0.25 * O.DO) *
(DATA (1, 1) - O.DO) / 2.DO +
(0.75 * OUTCON (1) + 0.25 * OUTCON (2)) *
(DATA (2, 1) - DATA (1, 1)) / 2.DO) *
VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT PERIOD

DO 110 I = 2, (NOUT - 1)

RESET VELOCITY IF DESORPTION BEGINS

IF (DATA (I, 3) .LT. 1.0E-10) THEN
  VEL = VEL2
  IFLAG2 = 1
ENDIF

TOTOUT = TOTOUT +
((0.75 * OUTCON (1) + 0.25 * OUTCON (I - 1)) *
(DATA (I, 1) - DATA (I - 1, 1)) / 2.DO +
(0.75 * OUTCON (1) + 0.25 * OUTCON (I + 1)) *
(DATA (I + 1, 1) - DATA (1, 1)) / 2.DO) *
VEL * THETA
110 CONTINUE

CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST REPORT PERIOD

TOTOUT = TOTOUT +
(0.75 * OUTCON (NOUT) +
0.25 * OUTCON (NOUT - 1)) *
(DATA (NOUT, 1) - DATA (NOUT - 1, 1)) / 2.DO *
VEL * THETA
C RESTORE VELOCITY
VEL = VEL1
C
C.....CALCULATE CONTAMINANT REMAINING IN FLUID PHASE
C
C CALCIULATE CONTAMINANT IN FLUID NEAR FIRST NODE
RCOL = REAL (NCOL)
TOTWET = (0.75 * DV (1) + 0.25 * DV (3)) * ZCOL / (RCOL - 1.DO) / 2.DO * THETA
C
C CALCIULATE CONTAMINANT IN FLUID NEAR TYPICAL NODE
DO 120 I = 3, (2 * NCOL - 3), 2
TOTWET = TOTWET + (0.75 * DV (I) + 0.125 * DV (I - 2) + 0.125 * DV (I + 2)) * ZCOL / (RCOL - 1.DO) * THETA
120 CONTINUE
C
C CALCIULATE CONTAMINANT IN FLUID NEAR EXIT NODE
TOTWET = TOTWET + (0.25 * DV (2 * NCOL - 3) + 0.75 * DV (2 * NCOL - 1)) * ZCOL / (RCOL - 1.DO) * THETA
C
C.....CALCULATE CONTAMINANT REMAINING ON SOLID PHASE
C
C CALCIULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE
TOTDRY = 0.DO * 0.75 + DV (4) * 0.25
C
C CALCIULATE CONTAMINANT ON TYPICAL NODES
DO 130 I = 4, (2 * NCOL - 2), 2
TOTDRY = TOTDRY + (0.75 * DV (I) + 0.125 * DV (I - 2) + 0.125 * DV (I + 2)) * ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)
130 CONTINUE
C
C CALCIULATE CONTAMINANT NEAR EXIT
TOTDRY = TOTDRY + (0.25 * DV (2 * NCOL - 2) + 0.75 * DV (2 * NCOL)) * ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO - THETA)
C
C.....COMPUTE MASS BALANCE
BAL = TOTINF / (TOTOUT + TOTWET + TOTDRY)
C
C.....OUTPUT THE MASS BALANCE FINDINGS
WRITE (100OUT, 1000)
WRITE (100OUT, 1100)
WRITE (100OUT, 1200) TOTINF
WRITE (100OUT, 1300) TOTOUT
WRITE (100OUT, 1400) TOTWET
WRITE (100OUT, 1500) TOTDRY
WRITE (100UT, 1600) BAL
IF (IFLAG1 .EQ. 1) THEN
WRITE (100UT, 1700)
ENDIF
IF (IFLAG2 .EQ. 1) THEN
WRITE (100UT, 1800)
END IF
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 DESORPTION. ')
C
RETURN
END
C

THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE COMPLETION OF SIMULATION IN SIMULATION MODE.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>LOOP INDEX</td>
</tr>
<tr>
<td>IFLAG1</td>
<td>INTEGER*4</td>
<td>FLAG FOR ABRUPT CONCENTRATION CHANGE</td>
</tr>
<tr>
<td>IFLAG2</td>
<td>INTEGER*4</td>
<td>FLAG FOR DESORPTION</td>
</tr>
<tr>
<td>I0IN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS</td>
</tr>
<tr>
<td>BAL</td>
<td>REAL*8</td>
<td>RATIO OF MASS IN TO MASS ACCOUNTED</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>OUTCON</td>
<td>REAL*8</td>
<td>EXIT CONCENTRATION VECTOR</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*4</td>
<td>SIMULATION STATUS MATRIX</td>
</tr>
<tr>
<td>RCOL</td>
<td>REAL*8</td>
<td>REAL EQUIVALENT OF NCOL</td>
</tr>
</tbody>
</table>
SUBROUTINE MASS2 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2, RHO, OUTPUT, NSTEPS, NCOL)

IMPLICIT REAL*8 (A-H, O-Z)
REAL*4 OUTPUT
DIMENSION OUTPUT (250, 5), DV (101), OUTCON (250)
COMMON /UNITS/ I0IN, I0OUT

IFLAG1 = 0
IFLAG2 = 0

C.....STORE THE INITIAL VELOCITY
VEL1 = VEL

C.....CALCULATE TOTAL CONTAMINANT INJECTED

CALCULATE CONTAMINANT INJECTED DURING FIRST REPORT PERIOD.

TOTINF = CO * OUTPUT (1, 1) * VEL * THETA +
0.5 * (OUTPUT (1, 4) - CO) * OUTPUT (1, 1) * VEL * THETA

CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS

DO 100 I = 2, (NSTEPS - 1)
  IF ((OUTPUT (1, 4) .LT. 0.8 * OUTPUT ((I - 1), 4)) OR. (OUTPUT (1, 4) .GT. 1.2 * OUTPUT ((I - 1), 4)) THEN
    IFLAG1 = 1
    TOTINF = TOTINF +
           1. OUTPUT (1, 4) *
           2. (OUTPUT (1, 1) - OUTPUT (I - 1, 1)) * VEL * THETA
  ELSE
    TOTINF = TOTINF +
           1. OUTPUT (I - 1, 4) *
           2. (OUTPUT (1, 1) - OUTPUT (I - 1, 1)) *
           3. VEL * THETA *
           4. 0.5 * (OUTPUT (1, 4) - OUTPUT (I - 1, 4)) *
           5. (OUTPUT (1, 1) - OUTPUT (I - 1, 1)) *
           6. VEL * THETA
  ENDIF
100 CONTINUE

C.....CALCULATE TOTAL EFFLUENT RELEASED

CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
THE FIRST REPORT PERIOD

TOTOUT = (0.75 * O.DO + 0.25 * OUTCON (1)) * 
1 OUTPUT (1, 1) / 2.DO * VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT PERIOD

TOTOUT = TOTOUT + 
1 ((0.75 * OUTCON (1) + 0.25 * O.DO) * 
2 (OUTPUT (1, 1) - O.DO) / 2.DO + 
3 (0.75 * OUTCON (1) + 0.25 * OUTCON (2)) * 
4 (OUTPUT (2, 1) - OUTPUT (1, 1)) / 2.DO) * 
5 VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT PERIOD

DO 110 I = 2, (NSTEPS - 1)

RESET VELOCITY IF DESORPTION BEGINS

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

CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST REPORT PERIOD

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

RESTORE VELOCITY

VEL = VEL1

CALCULATE CONTAMINANT REMAINING IN FLUID PHASE

CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE

RCOL = REAL (NCOL)
TOTWET = (0.75 * DV (1) + 0.25 * DV (3)) * 
1 ZCOL / (RCOL - 1.DO) / 2.DO * THETA

CALCULATE CONTAMINANT IN FLUID NEAR TYPICAL NODE

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

CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
TOTWET = TOTWET + (0.25 * DV (2 * NCOL - 3) * 
1.0.75 * DV (2 * NCOL - 1)) * 
2. ZCOL / (RCOL - 1.DO) / 2.DO * THETA

.... CALCULATE CONTAMINANT REMAINING ON SOLID PHASE

CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE
TOTDRY = 0.DO * 0.75 + DV (4) * 0.25

CALCULATE CONAMINANT ON TYPICAL NODES
DO 130 1 = 4, (2 * NCOL - 2), 2
TOTDRY = TOTDRY +
1 (0.75 * DV (1) + 0.125 * DV (1 - 2) 
2 + 0.125 * DV (1 + 2)) * 
3 ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)
130 CONTINUE

CALCULATE CONTAMINANT NEAR EXIT
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)

.... COMPUTE MASS BALANCE

BAL = TOTINF / (TOTOUT + TOTWET + TOTDRY)

.... OUTPUT THE MASS BALANCE FINDINGS

WRITE (IOUT, 1000)
WRITE (IOUT, 1100)
WRITE (IOUT, 1200) TOTINF
WRITE (IOUT, 1300) TOTOUT
WRITE (IOUT, 1400) TOTWET
WRITE (IOUT, 1500) TOTDRY
WRITE (IOUT, 1600) BAL
IF (IFLAG1 .EQ. 1) THEN
WRITE (IOUT, 1700)
END IF
IF (IFLAG2 .EQ. 1) THEN
WRITE (IOUT, 1800)
END IF

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 = '')
1700 FORMAT (' ', 20X, 'MASS BALANCE TAKES ACCOUNT OF ABRUPT CHANGE',
  ' IN INFLUENT CONCENTRATION."
1800 FORMAT (' ', 20X, 'MASS BALANCE USED DESORPTION PORE VELOCITY',
  ' DURING DESORPTION."
C
C
RETURN
END
SORPTION MODEL:
FIRST ORDER EQUILIBRIUM WITH DISPERSION
MAINFRAME VERSION

THIS PROGRAM USES THE DGEAR METHOD TO SOLVE THE ADVECTIVE
DISPERSSIVE EQUATION FOR THE CASE OF FIRST ORDER EQUILIBRIUM WITH
DISPERSION.

THE FIRST ORDER AND PARALLEL EQUILIBRIUM SORPTION MODEL (FOPESM)
ASSUMES THAT INTERPHASE MASS TRANSFER OCCURS BETWEEN THE FLUID
PHASE AND TWO DIFFERENT TYPES OF SOLID PHASE SITES: FAST SITES
AND SLOW SITES. THE MASS TRANSFER TO THE FAST SITES OCCURS
RAPIDLY AND CAN BE ASSUMED TO BE AT EQUILIBRIUM. MASS TRANSFER
TO THE SLOW SITES IS TESTUDINATE AND REQUIRES BOTH A RATE AND
EQUILIBRIUM EXPRESSION TO COMPLETELY DESCRIBE THE INTERPHASE
MASS TRANSFER.

THE EQUATIONS ARE SPATIALLY APPROXIMATED USING THE FINITE
DIFFERENCE METHOD RESULTING IN A SET OF DC/DT AND DQ/DT EQUATIONS.
THESE EQUATIONS ARE THEN SOLVED SIMULTANEOUSLY USING DGEAR---
AN IMSL subroutine.

IN ITS PRESENT FORM THE PROGRAM IS LIMITED TO 100 NODES.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>I DUMMY</td>
<td>INTEGER*4</td>
<td>DUMMY WORKING ARRAY FOR DGEAR</td>
</tr>
<tr>
<td>IER</td>
<td>INTEGER*4</td>
<td>ERROR CODE RETURN INDEX FOR DGEAR</td>
</tr>
<tr>
<td>INDEX</td>
<td>INTEGER*4</td>
<td>CALL INDICATOR INDEX FOR GEAR</td>
</tr>
<tr>
<td>JOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>IKW</td>
<td>INTEGER*4</td>
<td>WORKING ARRAY FOR DGEAR</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NCOLEX</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF LOWER CODIAGONALS-DGEAR</td>
</tr>
<tr>
<td>NLC</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF UPPER CODIAGONALS-DGEAR</td>
</tr>
<tr>
<td>NUC</td>
<td>INTEGER*4</td>
<td>NUMBER OF BED VOLUME STEP TO OUTPUT</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF BED VOLUME STEP TO OUTPUT</td>
</tr>
<tr>
<td>ALPHA</td>
<td>REAL*8</td>
<td>SLOW SORPTION PROPORPORTIONALLY</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>CNORM</td>
<td>REAL*8</td>
<td>NORMALIZED EXIT CONCENTRATION</td>
</tr>
<tr>
<td>CPRIME</td>
<td>REAL*8</td>
<td>EXTERNAL NAME OF DC/DT SUBROUTINE</td>
</tr>
<tr>
<td>CTEMP</td>
<td>REAL*8</td>
<td>EXIT CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
</tbody>
</table>

COLUMN 1 CONTAINS REPORT TIMES
COLUMN 2 GIVES EXPERIMENTAL
CONCENTRATION
COLUMN 3 GIVES INFLUENT
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDVDT</td>
<td>REAL*8</td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL*8</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL*8</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL*8</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>DH1</td>
<td>REAL*8</td>
<td>SORPTION PHASE HYDRO_DYNAMIC</td>
</tr>
<tr>
<td>DUMMY</td>
<td>REAL*8</td>
<td>DUMMY ROUTINE REQUIRED BY DGEAR</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>EPSLON</td>
<td>REAL*8</td>
<td>ALLOWABLE ERROR FOR DGEAR</td>
</tr>
<tr>
<td>FCNJ</td>
<td>REAL*8</td>
<td>EXTERNAL NAME OF JACOBIAN ROUTINE</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID PHASE LOSS RATE</td>
</tr>
<tr>
<td>FKF</td>
<td>REAL*8</td>
<td>FAST FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKS</td>
<td>REAL*8</td>
<td>SLOW FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKT</td>
<td>REAL*8</td>
<td>THROUGHPUT FREUNDLICH COEFFICIENT</td>
</tr>
<tr>
<td>FNF</td>
<td>REAL*8</td>
<td>FAST FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>FNS</td>
<td>REAL*8</td>
<td>SLOW FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>FNT</td>
<td>REAL*8</td>
<td>THROUGHPUT FREUNDLICH EXPONENT</td>
</tr>
<tr>
<td>FSSDL</td>
<td>REAL*8</td>
<td>FIRST ORDER DECAY LOSS ON SOLID</td>
</tr>
<tr>
<td>FSDSL</td>
<td>REAL*8</td>
<td>FIRST ORDER DECAY LOSS ON SOLID</td>
</tr>
<tr>
<td>H</td>
<td>REAL*8</td>
<td>STEP SIZE FOR DGEAR</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*8</td>
<td>A MATRIX OF SIMULATION STATUS</td>
</tr>
<tr>
<td>RDSQSM</td>
<td>REAL*8</td>
<td>SUM OF RESIDUALS SQUARED</td>
</tr>
<tr>
<td>RFGRP</td>
<td>REAL*8</td>
<td>RETARDATION VARIABLE GROUPING</td>
</tr>
<tr>
<td>RHO</td>
<td>REAL*8</td>
<td>SOLID PHASE PARTICLE DENSITY</td>
</tr>
<tr>
<td>RSDSUM</td>
<td>REAL*8</td>
<td>SUM OF RESIDUALS</td>
</tr>
<tr>
<td>SDUMMY</td>
<td>REAL*8</td>
<td>DUMMY ROUTINE FOR DGEAR</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>SLOW SOLID PHASE ACTUAL</td>
</tr>
<tr>
<td>SLSPEQ</td>
<td>REAL*8</td>
<td>SLOW SOLID PHASE EQUILIB GROUPING</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>TIME</td>
<td>REAL*8</td>
<td>TIME OF OUTPUT SIMULATION POINT</td>
</tr>
<tr>
<td>THESIM</td>
<td>REAL*8</td>
<td>NUMBER OF BED VOLUMES TO SIMULATE</td>
</tr>
<tr>
<td>TO</td>
<td>REAL*8</td>
<td>INITIAL TIME FOR DGEAR</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL1</td>
<td>REAL*8</td>
<td>SORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>UK</td>
<td>REAL*8</td>
<td>WORKING ARRY NEEDED BY DGEAR</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
</tbody>
</table>

**Variables**

- **Title**: CHARACTER*72 TITLE OF SIMULATION RUN

**Implicit Types**

- IMPLICIT REAL*8 (A-H, O-Z)
- REAL*4 SDUMMY, DATA, OUTPUT
- CHARACTER*72 TITLE

**Dimension**

- DIMENSION F(100), Y(2)
DIMENSION IWK (101), DW (101), DDDVT (101), OUTPUT (250,5), 1 \( \text{DIMENSION DATA (100,4), DV (101), DDDVT (101), OUTPUT (250,5),} \)
1 \( \text{COMMON /DBAND/ NLC, NUC} \)
COMMON /GEAR/ DUMMY(48), SDUMMY(4), IDUMMY(38)  
COMMON /GROUPS/ DGN, DGNLOU, DGNUP, RFGRP, FSSDLS, SLSPEQ, 1 \( \text{COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,} \)
1 \( \text{EPSLON, CO, FKF, FNF, FKS, FNS, ALPHA,} \)
1 \( \text{FKAPPA, SKAPPA, FKT, FNT, C, T,} \)
1 \( \text{ICON, NCOL, NOUT, NSTEPS} \)
COMMON /WORDS/ TITLE  
COMMON /FLUID/ DCODT  
COMMON /UNITS/ I0IN, I0OUT  
EXTERNAL CPRIME, FCNJ  
I0IN = 1  
I0OUT = 3  
CALL READ  
VEL1 = VEL  
DH1 = DH  
CALL ECHO  
CALL ERRSET (208, 256, -1, 1)  
C.....COMPUTE OUTPUT FOR EXPERIMENTAL DATA IF NUMBER OF POINTS > 0  
C  
IF (NOUT .GT. 0) THEN  
CALL INIITL (DDVDVT, DV, OUTPUT, TO)  
CALL GRPFRM  
CALL INFO1 (DATA, NOUT)  
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE  
C  
NDOF = 2 * NCOL  
NCOLEX = 2 * NCOL - 1  
TO = 0.000  
H = 1.0-08  
INDEX = 1  
DV (1) = CO  
NLC = 2  
NUC = 2  
C.....TIME LOOP FOR SIMULATION AND OUTPUT AT EXPERIMENTAL POINTS  
C  
DO 100 I = 1, NOUT  
IF (I .GT. 1) THEN  
IF ((DATA (I, 3) .LT. 0.8 .AND. DATA ((I-1),3)) .OR. 1 \( \text{(DATA (I, 3) .GT. 1.2 .AND. DATA ((I-1),3)}) \)) THEN  
H = 1.0-08  
INDEX = 1  
DCODT = 0.000  
DV (1) = DATA (1, 3)  
WRITE (I0OUT, 1300)  
IF (DV (1) .LT. 1.0E-10) THEN  
VEL = VEL2  
DH = DH1 * VEL2 / VEL1  
CALL GRPFRM  
WRITE (I0OUT, 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).
CALL REPORT (I, DV, RSDSUM, RDSQSM, RSD)
OUTCON (I) = DV (NCOLEX)

100  F(I) = RSD

C.....RESTORE ORIGINAL PARAMETERS

C    DH = DHl
   VEL = VELl

C.....CHECK THE MASS BALANCE

C    CALL MASSl (OUTCON, DV, C0, ZCOL, THETA, VEL, VEL2,
   1 RHO, DATA, NOUT, NCOL)

C ENDIF

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 INF02 (OUTPUT, NSTEPS)

C ENDIF

C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE

C    NDOF = 2 * NCOL
    NCOLEX = 2 * NCOL - 1
    TO = 0.000
    H = 1.0D-08
    INDEX = 1
    DV (1) = C0
    NLC = 2
    NUC = 2

C.....TIME LOOP FOR SIMULATION AND OUTPUT AT COMPUTED POINTS

C WRITE (10OUT, 1000)
WRITE (10OUT, 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 = 1.0D-08
         INDEX = 1
         DCODT = 0.000
         DV (1) = OUTPUT (1, 4)
         WRITE (10OUT, 1300)
      IF (DV (1) .LT. 1.0E-10) THEN
         VEL = VEL2
         DH = DHl * VEL2 / VELl
         CALL GRPFRM
         WRITE (10OUT, 1400)
      ENDIF
   ELSE
      DCODT = OUTPUT (1, 5)
   ENDIF
   ELSE
      DCODT = OUTPUT (1, 5)
   ENDIF
   TIME = OUTPUT (1, 1)
   CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
   1 EPSLON, 2, -2, INDEX, IWK, WK, IER)
    CTEMP = DV (NCOLEX)
   CNORM = CTEMP / C0
   OUTCON (I) = DV (NCOLEX)
WRITE (100, 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 ('---------------------------------',
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

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIT NUMBER TO READ FROM</td>
<td>INTEGER*4</td>
<td>SLOW SORPTION PROPORTIONALITY CONSTANT</td>
</tr>
<tr>
<td>UNIT NUMBER TO WRITE TO</td>
<td>INTEGER*4</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>NUMBER OF NODES IN Z DIRECTION</td>
<td>INTEGER*4</td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
<td>INTEGER*4</td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td>NUMBER OF BED VOLUME STEP TO OUTPUT</td>
<td>INTEGER*4</td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td>COLUMN 4 GIVES DCO/DT</td>
<td>REAL*8</td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>EXPERIMENTAL INPUT DATA</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
<td>REAL*8</td>
<td>ALLOWABLE ERROR FOR DGEAR</td>
</tr>
<tr>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
<td>REAL*8</td>
<td>FLUID PHASE LOSS RATE</td>
</tr>
<tr>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
<td>REAL*8</td>
<td>FAST FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>COLUMN 4 GIVES DCO/DT</td>
<td>REAL*8</td>
<td>SLOW FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
<td>REAL*8</td>
<td>THROUGHPUT FREUNDLICH COEFFICIENT</td>
</tr>
<tr>
<td>ALLOWABLE ERROR FOR DGEAR</td>
<td>REAL*8</td>
<td>THROUGHPUT FREUNDLICH EXPONENT</td>
</tr>
<tr>
<td>FLUID PHASE LOSS RATE</td>
<td>REAL*8</td>
<td>SOLID PHASE PARTICLE DENSITY</td>
</tr>
<tr>
<td>FAST FREUNDLICH ISOTHERM COEFFICIENT</td>
<td>REAL*8</td>
<td>SOLID PHASE LOSS RATE</td>
</tr>
<tr>
<td>SLOW FREUNDLICH ISOTHERM COEFFICIENT</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>THROUGHPUT FREUNDLICH COEFFICIENT</td>
<td>REAL*8</td>
<td>NUMBER OF BED VOLUME STEPS TO SIMULATE</td>
</tr>
</tbody>
</table>
SUBROUTINE READ
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 DATA
CHARACTER*72 TITLE
DIMENSION DATA (100,4), C (100), T (100)
COMMON /INPUT/ THETAM, DATA, VEL, DH, RHO, ZCOL, VEL2,
1 EPSLON, CO, FKAPPA, FNF, FKS, FNS, ALPHA,
2 FKF, SKAPPA, FKT, FNT, C, T,
3 ICON, NCOL, NOUT, NSTEPS
COMMON /WORDS/ TITLE
COMMON /UNITS/ I01N, I00UT
READ (101N, 1000) TITLE
READ (101N, 1100) NCOL, NOUT, NSTEPS, THETAM, ICON
IF (ABS (NOUT) .GT. 0) THEN
   NTEMP = ABS (NOUT)
   READ (101N, 1200) (DATA (I), I = 1, NTEMP)
   READ (101N, 1200) (DATA (I), I = 1, NTEMP)
ENDIF
READ (101N, 1200) VEL, DH, RHO, THETA, ZCOL, EPSLON
READ (101N, 1200) CO, FKAPPA, FNF, FKS, FNS, ALPHA
READ (101N, 1200) FKF, SKAPPA, FKT, FNT, VEL2
READ (101N, 1200) (C (I), I = 1, ICON)
READ (101N, 1200) (T (I), I = 1, ICON)
1000 FORMAT (A72)
1100 FORMAT (3I5, E12.5, I5)
1200 FORMAT ((6E12.5))
RETURN
END

THIS SUBROUTINE ECHOS THE INPUT TO DEVICE I00UT.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I01N</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I00UT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>ALPHA</td>
<td>REAL*8</td>
<td>SLOW SORPTION PROPORTIONALITY</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
</tbody>
</table>
SUBROUTINE ECHO
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 DATA
CHARACTER*72 TITLE
DIMENSION DATA (100, A), 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, FKAPPA, FKT, FNT, C, T,
3 ICON, NCOL, NOUT, NSTEPS
COMMON /WORDS/ TITLE
COMMON /UNITS/ 101N, I0OUT
PE = VEL * ZCOL / DH
WRITE (I0OUT, 1000)
WRITE (I0OUT, 1100)
WRITE (I0OUT, 1200) TITLE
WRITE (I0OUT, 1400)
WRITE (I0OUT, 1300)
WRITE (I0OUT, 1400)
WRITE (I0OUT, 1500) CD
WRITE (I0OUT, 1600) DH
WRITE (I0OUT, 1700) NCOL
WRITE (I0OUT, 1800) RHO
WRITE (I0OUT, 1900) VEL
WRITE (I0OUT, 1950) VEL2
WRITE (I0OUT, 2000) THETA
WRITE (I0OUT, 2100) ZCOL
WRITE (I0OUT, 2200) PE
WRITE (I0OUT, 2300) FKF
WRITE (I0OUT, 2400) FNF
WRITE (I0OUT, 2500) FKS
WRITE (I0OUT, 2600) FNS
WRITE (I0OUT, 2700) ALPHA
WRITE (I0OUT, 2800) FKAPPA
WRITE (I0OUT, 2900) SKAPPA
WRITE (I0OUT, 3000) ICON
WRITE (I0OUT, 3100)
WRITE (I0OUT, 3200) (C (I), 1 * I, ICON)
WRITE (I0OUT, 3300)
WRITE (10OUT, 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, 'UNIT', 16X)
1400 Format ('O-----',
1 '-----------------------------------------------'
1 '-----------------------------------------------'
1500 FORMAT ('O', 15X, 'CD', 16X, '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 15)
1800 FORMAT ('*', 15X, 'RHO', 18X, 'SOLID PHASE DENSITY', 14X,
1 E10.4, 16X, 'GR/CM**3')
1900 FORMAT ('*', 15X, 'VOL', 18X, 'PORE VELOCITY', 14X,
1 E10.4, 16X, 'CM/HOUR')
1950 FORMAT ('*', 15X, 'VOL', 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, 'l/HR')
2800 FORMAT ('*', 15X, 'FKAPPA', 18X, 'FLUID PHASE LOSS RATE', 14X,
1 E10.4, 16X, 'l/HR')
2900 FORMAT ('*', 15X, 'SKAPPA', 18X, 'SOLID PHASE LOSS RATE', 14X,
1 E10.4, 16X, 'l/HR')
3000 FORMAT ('*', 15X, 'ICON', 18X, 'NUMBER OF INFLUENTS', 14X,
1 15)
3100 FORMAT ('*', 39X, 'INFLUENT CONCENTRATIONS')
3200 FORMAT ('*', 14X, 10E10.3)
3300 FORMAT ('*', 39X, 'INFLUENT TIMES')
RETURN
END

THIS SUBROUTINE WRITES THE CONTENTS OF THE DATA MATRIX TO
UNIT 10OUT.

VARIABLE MAP:
----------------
VARIABLE NAME    PRECISION      DESCRIPTION
--------------------------
10IN INTEGER*4       UNIT NUMBER TO READ FROM
10OUT INTEGER*4      UNIT NUMBER TO WRITE TO
NCOUNT INTEGER*4     LOOP COUNTER
NOUT INTEGER*4       NUMBER OF TEMPORAL STEPS TO OUTPUT
DATA REAL*4

EXPERIMENTAL INPUT DATA
COLUMN 1 CONTAINS REPORT TIMES
COLUMN 2 GIVES EXPERIMENTAL
CONCENTRATION
COLUMN 3 GIVES INFLUENT
CONCENTRATION
COLUMN 4 GIVES DCO/DT

SUBROUTINE INFO1 (DATA, NOUT)
REAL*4 DATA
DIMENSION DATA (100,4)
COMMON /UNITS/ lOIN, lOOUT
WRITE (lOOUT, 1000)
WRITE (lOOUT, 1100)
WRITE (lOOUT, 1200)
WRITE (lOOUT, 1300)
DO 100 NCOUNT = 1, NOUT
WRITE (lOOUT, 1300) ( DATA (NCOUNT, I), I = 1, A)
100  CONTINUE
1000 FORMAT ('', 42X, 'EXPERIMENTAL COMPARISON MODE INFORMATION')
1100 FORMAT ('--------------------------',
1 '-------------------------------',
2 '-----------------------------------')
1200 FORMAT (' ', 'TIME', 5X, 'CONCENTRATION OUT', 6X,
1 'CONCENTRATION IN', 16X, 'DCO/DT')
1300 FORMAT (' ', 20X, E12.5, 1OX, E12.5, 1OX, E12.5, 1OX, E12.5)
RETURN
END

VARIABLE MAP:
---------------------

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>lOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>lOOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>INTEGER*4</td>
<td>LOOP COUNTER</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF EQUAL BED VOLUME STEPS</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*4</td>
<td>A MATRIX OF SIMULATION STATUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 1 CONTAINS REPORT TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 2 CONTAINS BED VOLUMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 3 CONTAINS THROUGHPUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 4 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 5 GIVES DCO/DT</td>
</tr>
</tbody>
</table>

SUBROUTINE INFO2 (OUTPUT, NSTEPS)
REAL*4 OUTPUT
DIMENSION OUTPUT (250, 5)
COMMON /UNITS/ IOIN, IOUT
WRITE (IOUT, 1000)
WRITE (IOUT, 1100)
WRITE (IOUT, 1200)
WRITE (IOUT, 1300)
DO 100 NCOUNT = 1, NSTEPS
WRITE (IOUT, 1000) (OUTPUT (NCOUNT, I), I = 1, 5)
100  CONTINUE
1000 FORMAT ('1', 48X, 'SIMULATION MODE INFORMATION')
1100 FORMAT ('------------------------------',
  1 '-------------------------------.)
1200 FORMAT (' ', 12X, 'REPORT TIME', IIX, 'BED VOLUMES', 12X,
  1 'THROUGHPUT', 6X, 'CONCENTRATION IN',
  2 16X, 'DCO/DT')
1300 FORMAT (' ', IIX, E12.5, IIX, E12.5, IIX, E12.5, IIX,
  1       E12.5, IIX, E12.5)
RETURN
END

THIS SUBROUTINE Initializes THE MODEL VARIABLES FOR THE SIMULATION
OF THE ADVECTIVE-DISPERSIVE EQUATION.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>WORKING VARIABLE</td>
</tr>
<tr>
<td>J</td>
<td>INTEGER*4</td>
<td>WORKING VARIABLE</td>
</tr>
<tr>
<td>ALPHA</td>
<td>REAL*8</td>
<td>SLOW SORPTION PROPORTIONALITY</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>DDVDT</td>
<td>REAL*8</td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>DG</td>
<td>REAL*8</td>
<td>RATIO OF TOTAL MASS ADSORBABLE ON</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOLID PHASE TO TOTAL MASS IN FLUID</td>
</tr>
<tr>
<td>DT</td>
<td>REAL*8</td>
<td>DURATION OF EACH TIME STEP</td>
</tr>
<tr>
<td>DTN</td>
<td>REAL*8</td>
<td>REAL EQUIVALENT OF NSTEPS</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>EPSMCN</td>
<td>REAL*8</td>
<td>MACHINE EPSILON</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID PHASE LOSS RATE</td>
</tr>
<tr>
<td>FKF</td>
<td>REAL*8</td>
<td>FAST FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKS</td>
<td>REAL*8</td>
<td>SLOW FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKT</td>
<td>REAL*8</td>
<td>THROUGHPUT FREUNDLICH COEFFICIENT</td>
</tr>
<tr>
<td>FNF</td>
<td>REAL*8</td>
<td>FAST FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>FNS</td>
<td>REAL*8</td>
<td>SLOW FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>FNT</td>
<td>REAL*8</td>
<td>THROUGHPUT FREUNDLICH EXPONENT</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*8</td>
<td>A MATRIX OF SIMULATION STATUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 1 CONTAINS REPORT TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 2 CONTAINS BED VOLUMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 3 CONTAINS THROUGHPUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 4 GIVES INFLUENT CONCENTRATION</td>
</tr>
</tbody>
</table>
SUBROUTINE INITL (DDVDT, DV, OUTPUT, TO)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 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, 2C0L, VEL2,
1 EPSLON, CO, FKAPPA, FNF, FKS, FNS, ALPHA,
2 FKAPPA, SKAPPA, FKT, FNT, C, T,
3 ICON, NCOL, NOUT, NSTEPS
DO 100 I = 1, 101
   DV (I) = 0.00
   DDVDT (I) = 0.00
100 DDVDT (1) = 0.00

C.....COMPUTE SIMULATION OUTPUT POINTS IF DESIRED
IF (NSTEPS .GT. 0) THEN
   TAU = ZCOL / VEL
   DG = 1 + (RHO * (1 - THETA) / THETA) * (FKT * (CO ** FNT)) / CO
   DTN = NSTEPS
   DT = TMESIM / DTN
   TEMPT = 0.00
   DO 110 I = 1, NSTEPS
      TEMPT = TEMPT + DT
      OUTPUT (1,1) = TEMPT
      OUTPUT (1,2) = TEMPT / TAU
      OUTPUT (1,3) = TEMPT / (TAU * DG)
   110    OUTPUT (1,3) = TEMPT / (TAU * DG)
ENDIF

C.....CALCULATE THE MACHINE EPSILON FOR USE AS A LOWER ERROR BOUND
EPSMCN = 1.0
120 EPSMCN = EPSMCN / 2.0000
   TEMP = 1.0000 + EPSMCN
   IF (TEMP .GT. 1.0000) GO TO 120

C.....CALCULATE THE INFLUENT CONCENTRATIONS FOR USE IN THE
C.....EXPERIMENTAL COMPARISON MODE BY INTERPOLATING FROM THE
C.....LIST OF KNOWN CONCENTRATIONS.
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)) THEN
               DATA (NCOUNT, 3) = DNCON * (UPCON - DNCON)
            1               * (TEND - DNTIME) / (UPTIME - DNTIME)
   130      DATA (NCOUNT, 3) = DNCON * (UPCON - DNCON)
   140      DATA (NCOUNT, 3) = DNCON * (UPCON - DNCON)
130      DATA (NCOUNT, 3) = DNCON * (UPCON - DNCON)

156

ENDF

140 CONTINUE
   IF (NCOUNT .GT. 1) THEN
      DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) -
                             DATA (NCOUNT - 1, 3)) / (DATA (NCOUNT, 1) -
                             DATA (NCOUNT - 1, 1))
   ELSE
      DATA (NCOUNT, 4) = (DATA (NCOUNT, 3) - CO) /
                           (DATA (NCOUNT, 1) - TO)
   ENDIF
   CONTINUE
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)) THEN
               OUTPUT (NCOUNT, 4) = DNCON + (UPCON - DNCON)
               * (TEND - DNTIME) / (UPTIME - DNTIME)
               DONE = .TRUE.
            ENDIF
         ENDIF
      ENDIF
   ENDIF
   CONTINUE
   IF (NCOUNT .GT. 1) THEN
      OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) -
                               OUTPUT (NCOUNT - 1, 4)) / (OUTPUT (NCOUNT, 1) -
                               OUTPUT (NCOUNT - 1, 1))
   ELSE
      OUTPUT (NCOUNT, 5) = (OUTPUT (NCOUNT, 4) - CO) /
                            (OUTPUT (NCOUNT, 1) - TO)
   ENDIF
   CONTINUE
C
C.....COMPUTE BED VOLUMES FOR VARIABLE VELOCITY CASE
C
OLDVEL = VEL
TEMPT = 0.0 DOO
   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
   CONTINUE
END
THIS SUBROUTINE COMPUTES VARIABLE GROUPINGS TO MINIMIZE EFFORT IN
THE SOLUTION OF THE SET OF SIMULTANEOUS NONLINEAR EQUATIONS.

VARIABLE MAP:
-------------------------

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>ALPHA</td>
<td>REAL*8</td>
<td>SLOW SORPTION PROPORTIONALITY</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DCO/DT</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL*8</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL*8</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL*8</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>DZ</td>
<td>REAL*8</td>
<td>COLUMN STEP SIZE IN Z DIRECTION</td>
</tr>
<tr>
<td>FKAPPA</td>
<td>REAL*8</td>
<td>FLUID PHASE LOSS RATE</td>
</tr>
<tr>
<td>FKF</td>
<td>REAL*8</td>
<td>FAST FREUNDLICH IsoTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKS</td>
<td>REAL*8</td>
<td>SLOW FREUNDLICH IsoTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FKT</td>
<td>REAL*8</td>
<td>THROUGHPUT FREUNDLICH COEFFICIENT</td>
</tr>
<tr>
<td>FNP</td>
<td>REAL*8</td>
<td>FAST FREUNDLICH IsoTHERM EXPONENT</td>
</tr>
<tr>
<td>FNS</td>
<td>REAL*8</td>
<td>SLOW FREUNDLICH IsoTHERM EXPONENT</td>
</tr>
<tr>
<td>FNT</td>
<td>REAL*8</td>
<td>THROUGHPUT FREUNDLICH EXPONENT</td>
</tr>
<tr>
<td>FSSDLS</td>
<td>REAL*8</td>
<td>FIRST ORDER DECAY LOSS ON SOLID</td>
</tr>
<tr>
<td>REALND</td>
<td>REAL*8</td>
<td>NUMBER OF NODES IN COLUMN</td>
</tr>
<tr>
<td>RFGRP</td>
<td>REAL*8</td>
<td>RETARDATION VARIABLE GROUPING</td>
</tr>
<tr>
<td>RMDO</td>
<td>REAL*8</td>
<td>SOLID PHASE PARTICLE DENSITY</td>
</tr>
<tr>
<td>SKAPPA</td>
<td>REAL*8</td>
<td>SOLID PHASE LOSS RATE</td>
</tr>
<tr>
<td>SLSPEC</td>
<td>REAL*8</td>
<td>SLOW SOLID PHASE ACTUAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONCENTRATION GROUPING</td>
</tr>
<tr>
<td>SLSPEQ</td>
<td>REAL*8</td>
<td>SLOW SOLID PHASE EQUILIB GROUPING</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>THESIM</td>
<td>REAL*8</td>
<td>NUMBER OF BED VOLUMES TO SIMULATE</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
</tbody>
</table>

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, REALND, RMDO, SKAPPA, SLSPEC, THETA, THESIM, VEL, VEL2, ZCOL
COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
EPSLON, C, FK, FNF, FKS, FNS, ALPHA,
FKAPPA, SKAPPA, FKT, FNT, C, T,
ICON, NCOL, NOUT, NSTEPS

C......COMPUTE MACROSCOPIC ADVECTIVE-DISPERSSIVE EQUATION FLUID PHASE GROUPINS

REALND = NCOL
DZ = ZCOL / (REALND - 1.D0)
DGNLOW = DH / (DZ ** 2.D0) + VEL / (2.D0 * DZ)
DGN = -2.D0 / (D2 ** 2.D0) + FKAPPA
DGNUP = DH / (D2 ** 2.D0) - VEL / (2.D0 * DZ)

RFGRP = FKF * FNF * (1 - THETA) * RHO / THETA

SLSPEQ = (RHO * (1 - THETA) / THETA) * ALPHA * FKS
SLSFAC = (RHO * (1 - THETA) / THETA) * ALPH
RETURN
END

THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE
DEPENDENT VARIABLE AS A FUNCTION OF TIME. THE EQUATIONS ARE
SOLVED USING THE VARIABLE GROUPINGS PREVIOUSLY DERIVED IN ORDER
TO SAVE COMPUTATIONAL EFFORT.

ALL DEPENDENT VARIABLE ARE STORED IN A SINGLE ARRAY DV, AND ALL
TEMPORAL DERIVATIVES OF THE DEPENDENT VARIABLE ARE STORED IN A
SINGLE ARRAY TERMED DDVDT. THE INDEX NUMBERING OF THESE ARRAYS
MAY BE DESCRIBED AS FOLLOWS:
A) THE FIRST ENTRY (1) IS THE INLET NODE OF THE COLUMN.
B) THE SECOND ENTRY (2) IS LEFT BLANK.
C) THE SUCCEEDING REFERENCES TO THE FLUID PHASE ARE AT THE
ODD-NUMBERED INDICES.
D) THE SOLID PHASE REFERENCES ARE AT THE EVEN-NUMBERED INDEX
FOLLOWING THE ASSOCIATED FLUID REFERENCES.
E) THE TOTAL DEGREES OF FREEDOM ARE EQUAL TO 2 * NCOL.
F) THE THE NEXT-TO-LAST ENTRY IS THE FLUID PHASE EXIT CON’N.
G) THE LAST ENTRY IS THE SOLID PHASE CONCENTRATION AT THE EXIT.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TEMPORARY VARIABLE</td>
</tr>
<tr>
<td>J</td>
<td>INTEGER*4</td>
<td>TEMPORARY VARIABLE</td>
</tr>
<tr>
<td>K</td>
<td>INTEGER*4</td>
<td>INDEX FOR SPHERICAL GROUPINGS</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NEND</td>
<td>INTEGER*4</td>
<td>VARIABLE INDEX FOR LAST COLUMN NODE</td>
</tr>
<tr>
<td>NSTART</td>
<td>INTEGER*4</td>
<td>LOOP STARTING POSITION</td>
</tr>
<tr>
<td>NSTEP</td>
<td>INTEGER*4</td>
<td>LOOP STOPPING POSITION</td>
</tr>
<tr>
<td>ALPHA</td>
<td>REAL*8</td>
<td>SLOW SORPTION PROPORTIONALITY</td>
</tr>
<tr>
<td>CNF</td>
<td>REAL*8</td>
<td>FAST EQUILIBRIUM PROPORTIONALITY</td>
</tr>
<tr>
<td>CNS</td>
<td>REAL*8</td>
<td>SLOW EQUILIBRIUM CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL</td>
</tr>
</tbody>
</table>
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 SLSPC
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 /FLUID/ DCODT

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. 0.000) THEN
  RFI = 1.000 + RFGRP * (DV (I) ** (FNF - 1.000))
  CNF = DV (I) ** FNF
  CNS = DV (I) ** FNS
ELSE
  RFI = 1.000
  CNF = 0.000
  CNS = 0.000
100 CONTINUE
100 DDVDT (I) = (DGNLOW * DV (I-2) + DGN * DV (I) + DGNUP * DV (I+2) 
1 + FSSDLS * CNF - SLSPEQ * CNS + SLSPAC * DV (I + 1)) 
2 / RFCI

C FORM THE OUTLET BOUNDARY CONDITION FOR THE FLUID PHASE

I = NDOF - 1
IF (DV (I) .GT. O.DOO) 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 / RFCI

C FORM THE SOLID PHASE EQUATIONS

DO 110 I = 4, NDOF, 2
J = I - 1
IF (DV (J) .GT. O.DOO) THEN
CNS = DV (J) ** FNS
ELSE
CNS = 0.D00
ENDIF
110 DDVDT (I) = ALPHA * (FKS * CNS - DV (I)) - SKAPPA * DV (I)
RETURN
END

THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE 
DDVDT ARRAY AS A FUNCTION OF THE DEPENDENT VARIABLE (DV). 
SINCE THE FINITE DIFFERENCE OPTION IS USED IN THE CALL TO DGEAR 
(I.E. MITER = -2) THIS ROUTINE IS A BLANK.

SUBROUTINE FCNJ (N, X, Y, PD)
REAL*8 Y(N), PD(N,N), X
RETURN
END

THIS SUBROUTINE USES THE EXPERIMENTAL DATA INPUT AND THE MODEL 
SIMULATION OUTPUT TO COMPUTE THE MODEL RESIDUAL, RESIDUAL SQUARED, 
SUM OF RESIDUALS, AND SUM OF RESIDUALS SQUARED. THESE COMPUTATIONS 
ARE OUTPUT TO UNIT I00OUT.

VARIABLE MAP:
-----------------

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
</table>

...
SUBROUTINE REPORT (I, DV, RDSUM, RDSQSM, RSD)

IMPLICIT REAL*8 (A-H, O-Z)

REAL*4 DATA
DIMENSION DATA (100,*), DV(100), C (100), T (100)

COMMON /INPUT/ TMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2,
1 EPSLON, CO, FKF, FNF, FKS, FNT, FNT, FKAPPA, SKAPPA, FKT, FNT, C, T,
2 ICON, NCOL, NOUT, NSTEPS

COMMON /UNITS/ IOIN, IOUT

NDV = (NCOL * 2) - 1

IF (I .EQ. 1) THEN
  WRITE (IOUT, 1000)
  WRITE (IOUT, 1100)
  WRITE (IOUT, 1100)
  RDSUM = 0.D00
  RDSQSM = 0.D00
END IF

CNORM = DV (NDV) / CO

RSD = DATA (1,2) - CNORM

RSDSQ = RSD * RSD

RDSQSM = RDSQSM + RSDSQ

WRITE (IOUT, 1300) DATA (1,1), DATA (1,2), CNORM, RSD, RDSQSM

IF (I .GE. NOUT) THEN
  WRITE (IOUT, 1100)
WRITE (1000, 1400) RSDSUM, RDSQM
ENDIF
1000 FORMAT ('1',56X,'RESIDUAL REPORT')
1100 FORMAT ('----------------------------------------------------------------------------------',
1 '----------------------------------------------------------------------------------',
2 '----------------------------------------------------------------------------------')
1200 FORMAT ('0',19X,'TIME','C/Co (INPUT)','C/Co (PREDICTED),'
14X,'RESIDUAL','14X,'RESIDUAL SQUARED')
1300 FORMAT (' ',19X,'TIME',1OX,'C/Co (INPUT),1OX,'C/Co (PREDICTED),
14X,'RESIDUAL',1OX,'RESIDUAL SQUARED')
1400 FORMAT (' ',81X,E12.5,14X,E12.5)
RETURN
END

THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE COMPLETION OF SIMULATION FOR EXPERIMENTAL COMPARISON.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>LOOP INDEX</td>
</tr>
<tr>
<td>IFLAG1</td>
<td>INTEGER*4</td>
<td>FLAG FOR ABRUPT CONCENTRATION CHANGE</td>
</tr>
<tr>
<td>IFLAG2</td>
<td>INTEGER*4</td>
<td>FLAG FOR DESORPTION</td>
</tr>
<tr>
<td>I0IN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>BAL</td>
<td>REAL*8</td>
<td>RATIO OF MASS IN TO MASS ACCOUNTED</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*8</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>OUTCON</td>
<td>REAL*8</td>
<td>EXIT CONCENTRATION VECTOR</td>
</tr>
<tr>
<td>RCOL</td>
<td>REAL*8</td>
<td>REAL EQUIVALENT OF NCOL</td>
</tr>
<tr>
<td>Theta</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>TOTDRY</td>
<td>REAL*8</td>
<td>TOTAL MASS LEFT ON SOLID PHASE</td>
</tr>
<tr>
<td>TOTINF</td>
<td>REAL*8</td>
<td>TOTAL MASS INJECTED INTO COLUMN</td>
</tr>
<tr>
<td>TOTOUT</td>
<td>REAL*8</td>
<td>TOTAL MASS TO EXIT COLUMN</td>
</tr>
<tr>
<td>TOTWET</td>
<td>REAL*8</td>
<td>TOTAL MASS LEFT IN LIQUID PHASE</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
</tbody>
</table>
SUBROUTINE MASS1 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
1 RHO, DATA, NOUT, NCOL)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 DATA
DIMENSION DATA (100, 4), DV (101), OUTCON (250)
COMMON /UNITS/ lOIN, lOOUT
C
C. IFLAG1 = 0
1 IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
C
VE L1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED
C
CALCULATE CONTAMINANT INJECTED DURING FIRST
REPORT PERIOD.
1 TOTINF = CO * DATA (1, 1) * VEL * THETA +
2 0.5 * (DATA (1, 3) - CO) * DATA (1, 1) *
2 VEL * THETA
C
CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
DO 100 I = 2, (NOUT - 1)
1 IF ((DATA (I, 3) .LT. 0.8 * DATA ((I - 1),3)) .OR.
2 (DATA (I, 3) .GT. 1.2 * DATA ((I - 1),3))) THEN
3 IFLAG1 = 1
4 TOTINF = TOTINF +
5 DATA (I, 3) * (DATA (I, 1) - DATA (I - 1, 1)) *
6 VEL * THETA
ELSE
7 TOTINF = TOTINF +
8 DATA (I - 1, 3) *
9 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA +
10 0.5 * (DATA (I, 3) - DATA (I - 1, 3)) *
11 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
END IF
100 CONTINUE
C
C.....CALCULATE TOTAL EFFLUENT RELEASED
C
CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
THE FIRST REPORT PERIOD
1 TOTOUT = (0.75 * O.DO + 0.25 * OUTCON (1)) *
2 DATA (1, 1) / 2.DO * VEL * THETA
C
CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
PERIOD
C
TOTOUT = TOTOUT +
1 ((0.75 * OUTCON (1) + 0.25 * O.DO) *
2 (DATA (1, 1) - O.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
CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
PERIOD
C
DO 110 I = 2, (NOUT - 1)
C
RESET VELOCITY IF DESORPTION BEGINS
IF (DATA (I, 3) < 1.0E-10) THEN
  VEL = VEL2
  IFLAG2 = 1
ENDIF

TOTOUT = TOTOUT +
  1 (0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
  2 (DATA (I, 1) - DATA (I - 1, 1)) / 2.0D0 *
  3 (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
  4 (DATA (I + 1, 1) - DATA (I, 1)) / 2.0D0) *
  5 VEL * THETA

CONTINUE

C     CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST
C     REPORT PERIOD

TOTOUT = TOTOUT +
  1 (0.75 * OUTCON (NOUT) +
  2 0.25 * OUTCON (NOUT - 1)) *
  3 (DATA (NOUT, 1) - DATA (NOUT - 1, 1)) / 2.0D0 *
  4 VEL * THETA

C     RESTORE VELOCITY

VEL = VEL1

C     CALCULATE CONTAMINANT REMAINING IN FLUID PHASE

RCOL = REAL (NCOL)
TOTWET = (0.75 * DV (1) + 0.25 * DV (3)) *
  1 ZCOL / (RCOL - 1.0D0) / 2.0D0 * THETA

C     CALCULATE CONTAMINANT IN FLUID NEAR TYPICAL NODE

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 - 1.0D0) * THETA

120 CONTINUE

C     CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE

TOTWET = TOTWET + (0.25 * DV (2 * NCOL - 3) +
  1 0.75 * DV (2 * NCOL - 1)) *
  2 ZCOL / (RCOL - 1.0D0) / 2.0D0 * THETA

C     CALCULATE CONTAMINANT REMAINING ON SOLID PHASE (SLOW)

C     CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE (SLOW)

TOTDRY = 0.0D0 * 0.75 * DV (4) * 0.25

C     CALCULATE CONTAMINANT ON TYPICAL NODES (SLOW)

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)) *
CONTINUE

CALCULATE CONTAMINANT NEAR EXIT (SLOW)

\[ \text{TOTDRY} = \text{TOTDRY} + \]
\[ (0.25 \times \text{DV} (2 \times \text{NCOL} - 2) + 0.75 \times \text{DV} (2 \times \text{NCOL})) \]
\[ \times \frac{\text{ZCOL}}{\text{RCOL} - \text{L.DO}} \times \frac{1}{2 \times \text{DO}} \times \text{RHO} \times (\text{L.DO} - \text{THETA}) \]

CALCULATE CONTAMINANT REMAINING ON SOLID PHASE (FAST SITES)

CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE (FAST)

\[ \text{TOTDRY} = \text{TOTDRY} + \text{FKF} \times (0.75 \times \text{DV} (1) + 0.25 \times \text{DV} (3)) \times \frac{\text{ZCOL}}{\text{RCOL} - \text{L.DO}} \times \frac{1}{2 \times \text{DO}} \times \text{RHO} \times (\text{L.DO} - \text{THETA}) \]

CALCULATE CONTAMINANT ON TYPICAL NODES (FAST)

DO 140 1 = 3, (2 \times \text{NCOL} - 3), 2

\[ \text{TOTDRY} = \text{TOTDRY} + \text{FKF} \times \]
\[ (0.75 \times \text{DV} (1) + 0.125 \times \text{DV} (1 - 2)) \]
\[ + 0.125 \times \text{DV} (1 + 2)) \times \frac{\text{FKF}}{\text{FKF}} \times \frac{\text{ZCOL}}{\text{RCOL} - \text{L.DO}} \times \frac{1}{2 \times \text{DO}} \times \text{RHO} \times (\text{L.DO} - \text{THETA}) \]

140 CONTINUE

CALCULATE CONTAMINANT NEAR EXIT (FAST)

\[ \text{TOTDRY} = \text{TOTDRY} + \text{FKF} \times (0.25 \times \text{DV} (2 \times \text{NCOL} - 3) \]
\[ + 0.75 \times \text{DV} (2 \times \text{NCOL} - 1)) \times \text{FKF} \times \frac{\text{ZCOL}}{\text{RCOL} - \text{L.DO}} \times \frac{1}{2 \times \text{DO}} \times \text{RHO} \times (\text{L.DO} - \text{THETA}) \]

CALCULATE MASS BALANCE

\[ \text{BAL} = \frac{\text{TOTINF}}{\text{TOTOUT} + \text{TOTWET} + \text{TOTDRY}} \]

OUTPUT THE MASS BALANCE FINDINGS

WRITE (100OUT, 1000)
WRITE (100OUT, 1100)
WRITE (100OUT, 1200) TOTINF
WRITE (100OUT, 1300) TOTOUT
WRITE (100OUT, 1400) TOTWET
WRITE (100OUT, 1500) TOTDRY
WRITE (100OUT, 1600) BAL
IF (IFLAG1 .EQ. 1) THEN
  WRITE (100OUT, 1700)
ENDIF
IF (IFLAG2 .EQ. 1) THEN
  WRITE (100OUT, 1800)
ENDIF

1000 FORMAT ('1', 35X, 'EXPERIMENTAL MASS BALANCE REPORT')
1100 FORMAT ('0----------------------------------------'),
1   '----------------------------------------'),
2   '----------------------------------------'),
1200 FORMAT ('1', 20X, 'TOTAL MASS INJECTED = ',
1    E10.4, ' GRAMS / SQ. CM ')
1300 FORMAT ('1', 20X, 'TOTAL MASS RELEASED = '
1    E10.4, ' GRAMS / SQ. CM ')
RETURN
END

THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE COMPLETION OF SIMULATION IN SIMULATION MODE.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>LOOP INDEX</td>
</tr>
<tr>
<td>IFLAG1</td>
<td>INTEGER*4</td>
<td>FLAG FOR ABRUPT CONCENTRATION CHANGE</td>
</tr>
<tr>
<td>IFLAG2</td>
<td>INTEGER*4</td>
<td>FLAG FOR DESORPTION</td>
</tr>
<tr>
<td>I0IN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS</td>
</tr>
<tr>
<td>BAL</td>
<td>REAL*8</td>
<td>RATIO OF MASS IN TO MASS ACCOUNTED</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>OUTCON</td>
<td>REAL*8</td>
<td>EXIT CONCENTRATION VECTOR</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*8</td>
<td>SIMULATION STATUS MATRIX</td>
</tr>
<tr>
<td>RCOL</td>
<td>REAL*8</td>
<td>COL 1 GIVES REPORT TIMES</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>COL 2 GIVES BED VOLUMES</td>
</tr>
<tr>
<td>TOTDRY</td>
<td>REAL*8</td>
<td>COL 3 GIVES THROUGHPUT</td>
</tr>
<tr>
<td>TOTINF</td>
<td>REAL*8</td>
<td>COL 4 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td>TOTOUT</td>
<td>REAL*8</td>
<td>COL 5 GIVES DCO/DT</td>
</tr>
<tr>
<td>TOTWET</td>
<td>REAL*8</td>
<td>COL 1 GIVES REPORT TIMES</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>COL 2 GIVES BED VOLUMES</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>COL 3 GIVES THROUGHPUT</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>COL 4 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COL 5 GIVES DCO/DT</td>
</tr>
</tbody>
</table>

1400 FORMAT (' ', 20X, 'MASS LEFT IN FLUID PHASE = ', 1E10.4, ' GRAMS / SQ. CM ')
1500 FORMAT (' ', 20X, 'MASS LEFT ON SOLID PHASE = ', 1E10.4, ' GRAMS / SQ. CM ')
1600 FORMAT (' ', 20X, 'RATIO OF INPUT TO ACCOUNTED MASS = ', 1E10.4)
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 DESORPTION. ')

RETURN
END
SUBROUTINE MASSZ (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2, 
RHO, OUTPUT, NSTEPS, NCOL)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*4 OUTPUT
DIMENSION OUTPUT (250, 5), DV (101), OUTCON (250)
COMMON /UNITS/ I0IN, I0OUT

C
IFLAG1 = 0
IFLAG2 = 0
C.....STORE THE INITIAL VELOCITY
VEL1 = VEL
C.....CALCULATE TOTAL CONTAMINANT INJECTED

C
C    CALCULATE CONTAMINANT INJECTED DURING FIRST
C    REPORT PERIOD.

1 TOTINF = CO • OUTPUT (1, 1) • VEL • THETA +
1 0.5 • (OUTPUT (1, 4) - CO) • OUTPUT (1, 1) •
2 VEL • THETA
2
C
C    CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS

DO 100 I = 2, (NSTEPS - 1)
1 IF ((OUTPUT (I, 4) .LT. 0.8 • OUTPUT ((I - 1),4)) .OR.
2 (OUTPUT (I, 4) .GT. 1.2 • OUTPUT ((I - 1),4))) THEN
3 IFLAG1 = 1
4 TOTINF = TOTINF +
5 OUTPUT (I, 4) •
6 (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) •
7 VEL • THETA
8 ELSE
9 TOTINF = TOTINF +
10 OUTPUT (I - 1, 4) •
11 (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) •
12 VEL • THETA +
13 0.5 • (OUTPUT (I, 4) - OUTPUT (I - 1, 4)) •
14 (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) •
15 VEL • THETA
16 ENDIF
100 CONTINUE
C
C.....CALCULATE TOTAL EFFLUENT RELEASED

C
C    CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
C    THE FIRST REPORT PERIOD

1 TOTOUT = (0.75 • O.DO + 0.25 • OUTCON (1)) •
2 OUTPUT (1, 1) / 2.DO • VEL • THETA
2
C
C    CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT
C    PERIOD

1 TOTOUT = TOTOUT +
2 ((0.75 • OUTCON (1) + 0.25 • O.DO) •
3 (OUTPUT (1, 1) - O.DO) / 2.DO +
4 (0.75 • OUTCON (1) + 0.25 • OUTCON (2)) •
5 (OUTPUT (2, 1) - OUTPUT (1, 1)) / 2.DO •
6 VEL • THETA
6
C
C    CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT
C    PERIOD


DO 110 I = 2, (NSTEPS - 1)
RESET VELOCITY IF DESORPTION BEGINS
   IF (OUTPUT(I, 4) .LT. 1.0E-10) THEN
     VEL = VEL2
     IFLAG2 = 1
   END IF
   TOTOUT = TOTOUT +
     ((0.75 * OUTCON(I) + 0.25 * OUTCON(I - 1)) *
      (OUTPUT(I, 1) - OUTPUT(I - 1, 1)) / 2.0D0 +
     (0.75 * OUTCON(I) + 0.25 * OUTCON(I + 1)) *
      (OUTPUT(I + 1, 1) - OUTPUT(I, 1)) / 2.0D0) *
     VEL * THETA
110  CONTINUE

CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST REPORT PERIOD
   TOTOUT = TOTOUT +
     (0.75 * OUTCON(NSTEPS) +
      0.25 * OUTCON(NSTEPS - 1)) *
     (OUTPUT(NSTEPS, 1) - OUTPUT(NSTEPS - 1, 1)) / 2.0D0 *
     VEL * THETA
   VEL = VEL1

CALCULATE CONTAMINANT REMAINING IN FLUID PHASE

CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE
   RCOL = REAL(NCOL)
   TOTWET = (0.75 * DV(1) + 0.25 * DV(3)) *
     ZCOL / (RCOL - 1.0D0) / 2.0D0 * THETA

CALCULATE CONTAMINANT IN FLUID NEAR TYPICAL NODE
   DO 120 I = 3, (2 * NCOL - 3), 2
     TOTWET = TOTWET +
       (0.75 * DV(1) + 0.125 * DV(1 - 2) +
        0.125 * DV(1 + 2)) *
       ZCOL / (RCOL - 1.0D0) * THETA
120  CONTINUE

CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE
   TOTWET = TOTWET + (0.25 * DV(2 * NCOL - 3) +
     0.75 * DV(2 * NCOL - 1)) *
     ZCOL / (RCOL - 1.0D0) / 2.0D0 * THETA

CALCULATE CONTAMINANT REMAINING ON SOLID PHASE (SLOW SITES)

CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE (SLOW)
   TOTDRY = 0.0D0 * 0.75 * DV(4) * 0.25

CALCULATE CONTAMINANT ON TYPICAL NODES (SLOW)
   DO 130 I = 4, (2 * NCOL - 2), 2
TOTDRY = TOTDRY +
1 (0.75 * DV (1) + 0.125 * DV (1 - 2))
2 * 0.125 * DV (1 + 2)) *
3 ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)

CONTINUE

CALCULATE CONTAMINANT NEAR EXIT (SLOW)

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)

CALCULATE CONTAMINANT REMAINING ON SOLID PHASE (FAST SITES)

CALCULATE CONTAMINANT ON THE SOLID NEAR THE FIRST NODE (FAST)

TOTDRY = TOTDRY + FKF * (0.75 * DV (1) + 0.25 * DV (3)) * FNF *
1 ZCOL / (RCOL - 1.DO) / 2.DO * RHO * (1.DO - THETA)

CALCULATE CONAMINANT ON TYPICAL NODES (FAST)

DO 140 I = 3, (2 * NCOL - 3), 2

TOTDRY = TOTDRY + FKF *
1 (0.75 * DV (1) + 0.125 * DV (1 - 2))
2 * 0.125 * DV (1 + 2)) * FNF *
3 ZCOL / (RCOL - 1.DO) * RHO * (1.DO - THETA)

CONTINUE

CALCULATE CONTAMINANT NEAR EXIT (FAST)

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)

COMPUTE MASS BALANCE

BAL = TOTINF / (TOTOUT + TOTWET + TOTDRY)

OUTPUT THE MASS BALANCE FINDINGS

WRITE (IOUT, 1000)
WRITE (IOUT, 1100)
WRITE (IOUT, 1200) TOTINF
WRITE (IOUT, 1300) TOTOUT
WRITE (IOUT, 1400) TOTWET
WRITE (IOUT, 1500) TOTDRY
WRITE (IOUT, 1600) BAL
IF (IFLAG1 .EQ. 1) THEN
  WRITE (IOUT, 1700)
ENDIF
IF (IFLAG2 .EQ. 1) THEN
  WRITE (IOUT, 1800)
ENDIF
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 DESORPTION. ')
C
C
RETURN
END
SORPTION MODEL:
DUAL RESISTANCE
MAINFRAME VERSION

This program uses the DGEAR method to solve the advective dispersive equation for the case of dual-resistance sorption. The two resistances simulated include external film transport and internal solid-phase surface diffusion. The equilibrium model used is the single component Freundlich isotherm. The equations are spatially approximated using the finite difference method resulting in a set of DC/DT and DQ/DT equations. These equations are then solved simultaneously using DGEAR---an IMSL subroutine.

The limits of this program are three:
1) number of column nodes = 21
2) number of radial nodes = 21
3) number of data points = 100

These limits are imposed chiefly because of the resultant size of working array for DGEAR (WK). These limits may be relaxed at the expense of increased simulation time by using the diagonally dominant assumption for calls to DGEAR (i.e. MITER = 3). Much larger problems can be simulated using the diagonal dominant form, and the only changes required are to the arrays DV, DDVDT, RADGRP, and WK.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>IDUMMY</td>
<td>INTEGER*4</td>
<td>DUMMY WORKING ARRAY FOR DGEAR</td>
</tr>
<tr>
<td>IER</td>
<td>INTEGER*4</td>
<td>ERROR CODE RETURN INDEX FOR DGEAR</td>
</tr>
<tr>
<td>INDEX</td>
<td>INTEGER*4</td>
<td>CALL INDICATOR INDEX FOR GEAR</td>
</tr>
<tr>
<td>I0IN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>IWK</td>
<td>INTEGER*4</td>
<td>WORKING ARRAY FOR DGEAR</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NLC</td>
<td>INTEGER*4</td>
<td>NUMBER OF LOWER CODIAGONALS-DGEAR</td>
</tr>
<tr>
<td>NRAD</td>
<td>INTEGER*4</td>
<td>NUMBER OF RADIAL STEPS</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>NUC</td>
<td>INTEGER*4</td>
<td>NUMBER OF UPPER CODIAGONALS-DGEAR</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>CPRIME</td>
<td>REAL*8</td>
<td>EXTERNAL NAME OF DC/DT SUBROUTINE</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 1 CONTAINS REPORT TIMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 3 GIVES INFLUENT CONCENTRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLUMN 4 GIVES DC0/DT</td>
</tr>
<tr>
<td>DDVDT</td>
<td>REAL*8</td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
</tbody>
</table>
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), DDVD (462), OUTPUT (250,5),
 1 RADGRP (21,3), WK (3649), 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, NOUT, ICON
COMMON /WORDS/ TITLE
COMMON /FLUIDS/ DCO/DT

DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
DIMENSION REAL*8 (A-H, O-Z)
COMMON /UNITS/ IDIN, IDOUT
EXTERNAL CPRIME, FCNJ
IDIN = 1
IDOUT = 3
CALL READ
VEL1 = VEL
DH1 = DH
CALL ECHO
CALL ERRSET (208, 256, -1, 1)

C.....COMPUTE CONCENTRATION AT EXPERIMENTAL POINTS IF DESIRED
C
IF (NOUT .GT. 0) THEN
CALL INITL (DDVDT, DV, RADGRP, OUTPUT, TO)
CALL GRPFRM
CALL INFO1 (DATA, NOUT)
C
C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE
C
TO = O.DOO
H = 1.D-05
INDEX = 1
DV (1) = CO
NDOF = NCOL * (NRAD + 1)
NDV = NCOL * (NCOL - 1) * NRAD
NLC = NRAD + 1
NUC = NRAD + 1
C
C.....TIME LOOP FOR SIMULATION AND OUTPUT AT EXPERIMENTAL POINTS
C
DO 100 I = l,NOUT
IF (1 .GT. 1) THEN
IF (((DATA (1, 3) .LT. 0.8 * DATA ((1 - 1),3)) .OR.
     (DATA (1, 3) .GT. 1.2 * DATA ((1 - 1),3))) THEN
    H = 1.D-08
    INDEX = 1
    DCDT = 0.000
    DV (1) = DATA (1, 3)
    WRITE (IDOUT, 1300)
    IF (DV (1) .LT. 1.0E-10) THEN
       VEL = VEL2
       DH = DH1 * VEL2 / VEL1
       CALL GRPFRM
       WRITE (10OUT, 1400)
    END IF
ELSE
   DCDT = DATA (1, 4)
ENDIF
ELSE
   DCDT = DATA (1, 4)
ENDIF
TIME = DATA (1,1)
CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
    EPSLON, 2, -2, INDEX, IWK, WK, IER)
CALL REPORT (I, DV, RDSUM, 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 MASSl (OUTCON, DV, CO, 7001, THETA, VEL, Vel2)
RHO, DATA, NOUT, NCOL, NRAD)

ENDIF

C.....COMPUTE OUTPUT AT COMPUTED OUTPUT POINTS IF DESIRED

IF (NSTEPS .GT. 0) THEN
    CALL INITL (DDVDT, DV, RADGRP, OUTPUT, TO)
    CALL GRPFRM
    CALL INFO2 (OUTPUT, NSTEPS)

C.....INITIALIZE VARIABLE FOR FIRST CALL TO IMSL DGEAR ROUTINE

TO = 0.000
H = 1.0D-05
INDEX = 1
DV (1) = CO
NDOF = NCOL * (NRAD + 1)
NDV = NCOL * (NCOL - 1) * NRAD
NLC = NRAD + 1
NUC = NRAD + 1

C.....TIME LOOP FOR SIMULATION AND OUTPUT AT COMPUTED POINTS

WRITE (10OUT, 1000)
WRITE (10OUT, 1100)
DO 110 I = 1, NSTEPS
    IF (I .GT. 1) THEN
        IF (OUTPUT (I, 4) .LT. 0.8 * OUTPUT ((I - 1), 4)) OR.
    (OUTPUT (I, 4) .GT. 1.2 * OUTPUT ((I - 1), 4)) THEN
            H = 1.0D-05
            INDEX = 1
            DCODT = 0.000
            DV (1) = OUTPUT (I, 4)
            WRITE (I0OUT, 1300)
            IF (DV (1) .LT. 1.0E-10) THEN
                VEL = VEL2
                DH = DH1 * VEL2 / VEL1
                CALL GRPFRM
                WRITE (I0OUT, 1400)
            END IF
        ELSE
            DCODT = OUTPUT (I, 5)
        END IF
    ELSE
        DCODT = OUTPUT (I, 5)
    END IF
    TIME = OUTPUT (I, 1)
    CALL DGEAR (NDOF, CPRIME, FCNJ, TO, H, DV, TIME,
                EPSLON, 2, -2, INDEX, IWK, WK, IER)
    CTEMP = DV (NDV)
    CNORM = CTEMP / CO
    OUTCON (I) = DV (NDV)
110    WRITE (I0OUT, 1200) (OUTPUT (I, J), J = 1, 3), CTEMP, CNORM

C.....RESTORE ORIGINAL PARAMETERS

DH = DH1
VEL = VEL1

C.....CHECK THE MASS BALANCE

CALL MASS2 (OUTCON, DV, CO, ZCOL, THETA, VEL, VEL2,
            RHO, OUTPUT, NSTEPS, NCOL, NRAD)

ENDIF
THIS SUBROUTINE READS THE BASIC INPUT REQUIRED FOR THE DUAL-RESISTANCE SORPTION VERSION OF THE ADVECTIVE-DISPERSIVE EQUATION.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NRAD</td>
<td>INTEGER*4</td>
<td>NUMBER OF RADIAL STEPS</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
</tbody>
</table>

| CO            | REAL*8   | INITIAL CONCENTRATION |
| DATA          | REAL*4   | EXPERIMENTAL INPUT DATA |

| COLUMN 1 CONTAINS REPORT TIMES |
| COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION |
| COLUMN 3 GIVES INFLUENT CONCENTRATION |
| COLUMN 4 GIVES DCO/DT |

| DH            | REAL*8   | HYDRODYNAMIC DISPERSION COEFFICIENT |
| DS            | REAL*8   | SURFACE DIFFUSION COEFFICIENT |
| EPSLON        | REAL*8   | ALLOWABLE ERROR FOR DGEAR |
| FILMK         | REAL*8   | FILM TRANSFER COEFFICIENT |
| FK            | REAL*8   | FREUNDLICH ISOTHERM COEFFICIENT |
| FN            | REAL*8   | FREUNDLICH ISOTHERM EXPONENT |
| RADIUS        | REAL*8   | RADIUS OF SPHERICAL PARTICLE |
| RHO           | REAL*8   | SOLID PHASE PARTICLE DENSITY |
| THETA         | REAL*8   | POROSITY |
| VEL           | REAL*8   | PORE VELOCITY |
| VEL2          | REAL*8   | DESORPTION PORE VELOCITY |
| ZCOL          | REAL*8   | LENGTH OF COLUMN (Z DIRECTION) |

| TITLE         | CHARACTER*72 | TITLE OF SIMULATION RUN |

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/ lOIN, lOOUT
READ (lOIN, 1000) TITLE
READ (lOIN, 1100) NCOL, NRAD, NOUT, NSTEPS, TMESIM, ICON
IF (ABS (NOUT) .GT. 0) THEN
  NTEMP = ABS (NOUT)
  READ (lOIN, 1200) (DATA (I,1), I = 1, NTEMP)
ENDIF
READ (lOIN, 1100) VELO, DH, RHO, THETA, ZCOL, EPSLON
READ (lOIN, 1200) CO, FK, FN, RADIUS, DS, FILMK
READ (lOIN, 1200) VEL2
READ (lOIN, 1200) (C (I), I = 1, ICON)
READ (lOIN, 1200) (T (I), I = 1, ICON)
1000 FORMAT (A72)
1100 FORMAT (415,E12.5,15)
1200 FORMAT ((6E12.5))
RETURN
END

THIS SUBROUTINE ECHOS THE INPUT TO DEVICE lOOUT.

VARIABLE MAP:
-------------------
   VARIABLE NAME  PRECISION            DESCRIPTION
   ---------------  ---------------            -------------------
   lOIN            INTEGER*4            UNIT NUMBER TO READ FROM
   lOOUT           INTEGER*4            UNIT NUMBER TO WRITE TO
   NCOL            INTEGER*4            NUMBER OF NODES IN Z DIRECTION
   NRAD            INTEGER*4            NUMBER OF RADIAL STEPS
   CO              REAL*8               INITIAL CONCENTRATION
   DATA            REAL*8               EXPERIMENTAL INPUT DATA
   COLUMN 1 CONTAINS REPORT TIMES
   COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION
   COLUMN 3 GIVES INFLUENT CONCENTRATION
   COLUMN 4 GIVES DCO/DT
   DH              REAL*8               HYDRODYNAMIC DISPERSION COEFFICIENT
   DS              REAL*8               SURFACE DIFFUSION COEFFICIENT
   EPSLON          REAL*8               ALLOWABLE ERROR FOR DGEAR
   FILMK           REAL*8               FILM TRANSFER COEFFICIENT
   FK              REAL*8               FREUNDLICH ISOTHERM COEFFICIENT
   FN              REAL*8               FREUNDLICH ISOTHERM EXPONENT
   PE              REAL*8               SYSTEM PECLET NUMBER
   RADIUS          REAL*8               RADIUS OF SPHERICAL PARTICLE
   RHO             REAL*8               SOLID PHASE PARTICLE DENSITY
   THETA           REAL*8               POROSITY
   VEL             REAL*8               PORE VELOCITY
   VEL2            REAL*8               DESORPTION PORE VELOCITY
   ZCOL            REAL*8               LENGTH OF COLUMN (2 DIRECTION)
SUBROUTINE ECHO
IMPLICIT REAL*8 (A-H, O-Z)
REAL*4 DATA
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/ lOIN, lOOUT
PE = VEL * ZCOL / DH
WRITE (lOOUT, 1000)
WRITE (lOOUT, 1100)
WRITE (lOOUT, 1200) TITLE
WRITE (lOOUT, 1400)
WRITE (lOOUT, 1300)
WRITE (lOOUT, 1400)
WRITE (lOOUT, 1500) CO
WRITE (lOOUT, 1600) DH
WRITE (lOOUT, 1700) NCOL
WRITE (lOOUT, 1800) RHO
WRITE (lOOUT, 1900) VEL
WRITE (lOOUT, 1950) VEL2
WRITE (lOOUT, 2000) THETA
WRITE (lOOUT, 2100) ZCOL
WRITE (lOOUT, 2200) PE
WRITE (lOOUT, 2300) FK
WRITE (lOOUT, 2400) FN
WRITE (lOOUT, 2500) FN
WRITE (lOOUT, 2600) FILMK
WRITE (lOOUT, 2700) DS
WRITE (lOOUT, 2800) RADUS
WRITE (lOOUT, 2900) NRAD
WRITE (lOOUT, 3000) ICON
WRITE (lOOUT, 3100)
WRITE (lOOUT, 3200) (C (I), I = 1, ICON)
WRITE (lOOUT, 3300)
WRITE (lOOUT, 3200) (T (I), I = 1, ICON)
1000 FORMAT ('1', 5OX, 'PLUG FLOW DISPERSION REACTOR')
1100 FORMAT ('1', 47X, 'DUAL RESISTANCE KINETIC SIMULATION')
1200 FORMAT ('1', 26X, A72)
1300 FORMAT ('1', 15X, 'VARIABLE', 16X, 'DESCRIPTION', 30X, 'VALUE',
1 16X, 'UNITS ', 16X)
1400 FORMAT ('0-----------------------------',
1 '-----------------------------------')
1500 FORMAT ('0', 15X, 'CO ', 16X, 'INITIAL CONCENTRATION ', 14X,
1 E10.4, 16X, 'G/CM**3')
1600 FORMAT ('', 15X, 'DH ', 16X, 'HYDRODYNAMIC DISPERSION ', 14X,
1 E10.4, 16X, 'CM**2/HOUR')
1700 FORMAT ('', 15X, 'NCOL ', 18X, 'NUMBER OF COLUMN STEPS ', 16X,
1 15)
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/HR')
THIS SUBROUTINE WRITES THE CONTENTS OF THE DATA MATRIX TO UNIT 10OUT.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>lOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>lOOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>INTEGER*4</td>
<td>LOOP COUNTER</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
</tbody>
</table>

COLUMN 1 CONTAINS REPORT TIMES
COLUMN 2 GIVES EXPERIMENTAL CONCENTRATION
COLUMN 3 GIVES INFLUENT CONCENTRATION
COLUMN 4 GIVES DCO/DT

SUBROUTINE INFO1 (DATA, NOUT)
REAL*4 DATA
DIMENSION DATA (100,4)
COMMON /UNITS/ lOIN, lOOUT
WRITE (lOOUT, 1000)
WRITE (lOOUT, 1100)
WRITE (lOOUT, 1200)
WRITE (100OUT, 1100)
DO 100 NCOUNT = 1, NSTEPS
   WRITE (100OUT, 1300) (DATA (NCOUNT, I), I = 1, 4)
100  CONTINUE
1000 FORMAT (1X, 42X, 'EXPERIMENTAL COMPARISON MODE INFORMATION')
1100 FORMAT ('O---------------------------------------------------------------',
              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)
RETURN
END

This subroutine writes the contents of the output matrix to unit 100OUT.

Variable Map:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Precision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>loin</td>
<td>integer*4</td>
<td>Unit number to read from</td>
</tr>
<tr>
<td>l00out</td>
<td>integer*4</td>
<td>Unit number to write to</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>integer*4</td>
<td>Loop counter</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>integer*4</td>
<td>Number of equal bed volume steps to be calculated</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>real*4</td>
<td>A matrix of simulation status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Col 1 contains report time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Col 2 contains bed volumes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Col 3 contains throughput</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Col 4 gives influent concentration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Col 5 gives dco/dt</td>
</tr>
</tbody>
</table>

Subroutine INFO2 (OUTPUT, NSTEPS)

REAL*4 OUTPUT
DIMENSION OUTPUT (250, 5)
COMMON /UNITS/ loin, l00out
WRITE (l00OUT, 1000)
WRITE (l00OUT, 1100)
WRITE (l00OUT, 1200)
WRITE (l00OUT, 1100)
DO 100 NCOUNT = 1, NSTEPS
   WRITE (l00OUT, 1300) (OUTPUT (NCOUNT, I), I = 1, 5)
100  CONTINUE
1000 FORMAT (1X, 48X, 'SIMULATION MODE INFORMATION')
1100 FORMAT ('O---------------------------------------------------------------',
              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
THIS SUBROUTINE INITIALIZES THE MODEL VARIABLES FOR THE SIMULATION OF THE ADVECTIVE-DISPERSIVE EQUATION.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>WORKING VARIABLE</td>
</tr>
<tr>
<td>J</td>
<td>INTEGER*4</td>
<td>WORKING VARIABLE</td>
</tr>
<tr>
<td>DDVDT</td>
<td>REAL*8</td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>EPSMCN</td>
<td>REAL*8</td>
<td>MACHINE EPSILON</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*4</td>
<td>MATRIX OF SIMULATION STATUS</td>
</tr>
<tr>
<td>RADGRP</td>
<td>REAL*8</td>
<td>SPHERICAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>TEMP</td>
<td>REAL*8</td>
<td>TEMPORARY VARIABLE FOR EPSMCN CALC</td>
</tr>
</tbody>
</table>

SUBROUTINE INITL (DDVDT, DV, RADGRP, OUTPUT, TO)
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, DM, 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 (*) = 0.D00
100 DDVDT () = 0.D00
DO 110 I = 1, 21
1 DO 110 J = 1, 3
110 RADGRP (I,J) = 0.D00
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 120 I = 1, NSTEPS
1 TEMPT = TEMPT + DT
120 OUTPUT (1,1) = TEMPT
OUTPUT (1,2) = TEMPT / TAU
120 OUTPUT (1,3) = 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)) THEN
            DATA (NCOUNT, 3) = DNCON + (UPCON - DNCON)
            DATA (NCOUNT, 4) = (TEND - DNTIME) / (UPTIME - DNTIME)
            DONE = .TRUE.
            ENDIF
150 CONTINUE
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)) THEN
            OUTPUT (NCOUNT, 4) = DNCON + (UPCON - DNCON)
            DATA (NCOUNT, 4) = (TEND - DNTIME) / (UPTIME - DNTIME)
            DONE = .TRUE.
            ENDIF
170 CONTINUE
ENDIF
160 CONTINUE
ENDIF
C
C.....CALCULATE INFLUENT CONCENTRATION FOR USE IN THE
C.....SIMULATION MODE.
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)) THEN
            DATA (NCOUNT, 3) = DNCON + (UPCON - DNCON)
            DATA (NCOUNT, 4) = (TEND - DNTIME) / (UPTIME - DNTIME)
            DONE = .TRUE.
            ENDIF
150 CONTINUE
ENDIF
140 CONTINUE
ENDIF
C
C.....CALCULATE INFLUENT CONCENTRATION FOR USE IN THE
C.....SIMULATION MODE.
C

....COMPUTE BED VOLUMES FOR VARIABLE VELOCITY CASE

OLDVEL = VEL
TEMP = 0.000
DO 180 I = 1, NSTEPS
   TEMPT = TEMPT + DT
   IF (OUTPUT (I, 6) GT. 1.0E-10) THEN
       CURVEL = VEL
   ELSE
       CURVEL = VEL2
   END IF
   AVGVEL = (OLDVEL • (TEMPT - DT) + CURVEL • DT) / TEMPT
   TAU = 2.0COL / AVGVEL
   OUTPUT (1, 2) = TEMPT / TAU
   OUTPUT (1, 3) = TEMPT / (TAU • DG)
   OLDVEL = AVGVEL
180   CONTINUE
END IF
RETURN
END

THESE SUBROUTINES COMPUTES VARIABLE GROUPINGS TO MINIMIZE EFFORT IN
THE SOLUTION OF THE SET OF SIMULTANEOUS NONLINEAR EQUATIONS.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDFF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>DELTAR</td>
<td>REAL*8</td>
<td>DELTA RADIUS FOR SPHERICAL EQUATION</td>
</tr>
<tr>
<td>DG</td>
<td>REAL*8</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL*8</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL*8</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DH</td>
<td>REAL*8</td>
<td>HYDRODYNAMIC DISPERSION COEFFICIENT</td>
</tr>
<tr>
<td>DS</td>
<td>REAL*8</td>
<td>SURFACE DIFFUSION COEFFICIENT</td>
</tr>
<tr>
<td>DZ</td>
<td>REAL*8</td>
<td>COLUMN STEP SIZE IN Z DIRECTION</td>
</tr>
<tr>
<td>FILMK</td>
<td>REAL*8</td>
<td>FILM TRANSFER COEFFICIENT</td>
</tr>
<tr>
<td>FLUXCF</td>
<td>REAL*8</td>
<td>MACROSCOPIC FLUX TERM GROUPING</td>
</tr>
<tr>
<td>GRPBDN</td>
<td>REAL*8</td>
<td>DIAGONAL GROUPING FOR SPHERICAL EQ</td>
</tr>
<tr>
<td>GRPOFF</td>
<td>REAL*8</td>
<td>OFF DIAGONAL GROUP FOR SPHERICAL EQ</td>
</tr>
<tr>
<td>RADGRP</td>
<td>REAL*8</td>
<td>SPHERICAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>RADIN</td>
<td>REAL*8</td>
<td>SPHERICAL INTERNAL BOUNDARY GROUP</td>
</tr>
<tr>
<td>RADIUS</td>
<td>REAL*8</td>
<td>RADIUS OF SPHERICAL PARTICLE</td>
</tr>
<tr>
<td>RADND</td>
<td>REAL*8</td>
<td>NUMBER OF RADIAL NODES</td>
</tr>
<tr>
<td>RADOUT</td>
<td>REAL*8</td>
<td>SPHERICAL EXTERNAL BOUNDARY GROUP</td>
</tr>
</tbody>
</table>
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/ 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
NDOF = NCOL * (NRAD + 1)
C.....COMPUTE MACROSCOPIC ADVECTIVE-DSIPERSIVE EQUATION FLUID PHASE GROUPINS C
REALND = NCOL
DZ = ZCOL / (REALND + 1.0D0)
DGNLOW = DH / (DZ ** 2.0D0) + VEL / (2.0D0 * DZ)
DGN = -2.0D0 * DH / (DZ ** 2.0D0) - 3.0D0 * FILMK * (1 - THETA)
1 / (RADIUS * THETA)
DGNUP = DH / (DZ ** 2.0D0) - VEL / (2.0D0 * DZ)
C.....COMPUTE SPHERICAL SOLID PHASE VARIABLE GROUPINGS C
FLUXCF = 3.0D0 * FILMK * (1 - THETA) / (RADIUS * THETA)
RADND = NRAD
DELTAR = RADIUS / (RADND - 1.0D0)
GRPDGN = -2.0D0 * DS / (DELTAR ** 2.0D0)
GRPOFF = DS / (DELTAR ** 2.0D0)
RADTMP = DELTAR
DO 100 1 = 2, NRAD
RADGRP (1,1) = GRPOFF - DS / (RADTMP * DELTAR)
RADGRP (1,2) = GRPDGN
RADGRP (1,3) = GRPOFF + DS / (RADTMP * DELTAR)
100 RADTMP = RADTMP + DELTAR
RADIN = 6.0D0 * DS / (DELTAR ** 2.0D0)
RADOUT = 2.0D0 * FILMK * DELTAR / (DS * RHO)
RETURN
END

THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE
DEPENDENT VARIABLE AS A FUNCTION OF TIME. THE EQUATIONS ARE
SOLVED USING THE VARIABLE GROUPINGS PREVIOUSLY DERIVED IN ORDER
TO SAVE COMPUTATIONAL EFFORT.

ALL DEPENDENT VARIABLE ARE STORED IN A SINGLE ARRAY DV, AND ALL
TEMPORAL DERIVATIVES OF THE DEPENDENT VARIABLE ARE STORED IN A
SINGLE ARRAY TERMED DDVDT. THE INDEX NUMBERING OF THESE ARRAYS
MAY BE DESCRIBED AS FOLLOWS:

A) THE FIRST ENTRY (1) IS THE INLET NODE OF THE COLUMN

B) THE SECOND ENTRY (2) IS THE INTERIOR NODE AT THE CENTER OF THE SPHERICAL PARTICLE CORRESPONDING TO THE INLET POSITION OF THE COLUMN

C) THE EXTERIOR BOUNDARY NODE OF THE SPHERICAL PARTICLE AT THE INLET OF COLUMN IS AT INDEX VALUE (NRAD + 1)

D) THE SECOND COLUMN NODE IS THEREFORE AT INDEX (NRAD + 2)

E) THE TOTAL DEGREES OF FREEDOM ARE EQUAL TO NCOL * (NRAD + 1)

F) THE LAST INDEX ENTRY IS THE BOUNDARY NODE OF THE SOLID PHASE SPHERE AT THE OUTLET POSITION OF THE COLUMN

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TEMPORARY VARIABLE</td>
</tr>
<tr>
<td>J</td>
<td>INTEGER*4</td>
<td>TEMPORARY VARIABLE</td>
</tr>
<tr>
<td>K</td>
<td>INTEGER*4</td>
<td>INDEX FOR SPHERICAL GROUPINGS</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NEND</td>
<td>INTEGER*4</td>
<td>VARIABLE INDEX FOR LAST COLUMN NODE</td>
</tr>
<tr>
<td>NRAD</td>
<td>INTEGER*4</td>
<td>NUMBER OF RADIAL STEPS</td>
</tr>
<tr>
<td>NSTART</td>
<td>INTEGER*4</td>
<td>LOOP STARTING POSITION</td>
</tr>
<tr>
<td>NSTEP</td>
<td>INTEGER*4</td>
<td>STEP SIZE FOR LOOP</td>
</tr>
<tr>
<td>NSTOP</td>
<td>INTEGER*4</td>
<td>LOOP STOPPING POSITION</td>
</tr>
<tr>
<td>CS</td>
<td>REAL*8</td>
<td>FLUID CONC AT EXTERIOR OF PARTICLE</td>
</tr>
<tr>
<td>DDVDT</td>
<td>REAL*8</td>
<td>DERIVATIVE OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>DGN</td>
<td>REAL*8</td>
<td>DIAGONAL VARIABLE GROUPING TERMS</td>
</tr>
<tr>
<td>DGNLOW</td>
<td>REAL*8</td>
<td>LOWER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DGNUP</td>
<td>REAL*8</td>
<td>UPPER DIAGONAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>FK</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM COEFFICIENT</td>
</tr>
<tr>
<td>FLUXCF</td>
<td>REAL*8</td>
<td>MACROSCOPIC FLUX TERM GROUPING</td>
</tr>
<tr>
<td>FN</td>
<td>REAL*8</td>
<td>FREUNDLICH ISOTHERM EXPONENT</td>
</tr>
<tr>
<td>RADGRP</td>
<td>REAL*8</td>
<td>SPHERICAL VARIABLE GROUPINGS</td>
</tr>
<tr>
<td>RADIN</td>
<td>REAL*8</td>
<td>SPHERICAL INTERNAL BOUNDARY GROUP</td>
</tr>
<tr>
<td>RADOUT</td>
<td>REAL*8</td>
<td>SPHERICAL EXTERNAL BOUNDARY GROUP</td>
</tr>
</tbody>
</table>

SUBROUTINE CPRIME (NVRB, TIME, DV, DDVDT)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 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/ TIMESIM, DATA, VEL, DH, RHO, THETA, ZCOL, VEL2, EPSLON, CO, FK, FN, RADIUS, DS, FILMK, FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT, NSTEPS, ICON COMMON /FLUID/ DCDT
$\text{NSTART} = \text{NRAD} + 2$

$\text{NSTOP} = \text{NDOF} - (2 \times \text{NRAD}) - 1$

$\text{NEND} = \text{NDOF} - \text{NRAD}$

$\text{NSTEP} = \text{NRAD} + 1$

$\text{DDVDT}(1) = \text{DCDT}$

$\text{DO 100 } I = \text{NSTART}, \text{NSTOP}, \text{NSTEP}$

$\text{IF (DV}(I+\text{NRAD})\. \text{GE.} \text{.O.DO0} \text{.THEN}$

$\text{CS} = ((\text{DV}(I+\text{NRAD}) / \text{FK}) ** (1.\text{DOO} / \text{FN}))$

$\text{ELSE}$

$\text{CS} = 0.\text{DOO}$

$\text{ENDIF}$

$\text{100 DDVDT}(1) = \text{DGNLOW} * \text{DV}(I-\text{NSTEP}) * \text{DGN} * \text{DV}(I)$

$\text{DO 120 } J = \text{NSTART}, \text{NSTOP}$

$\text{K} = J - \text{NSTART} + 2$

$\text{120 DDVDT}(J) = \text{RADGRP}(K,1) * \text{DV}(J-1) + \text{RADGRP}(K,2) * \text{DV}(J)$

$\text{RETURN}$

$\text{END}$

$\text{SUBROUTINE FCNJ(N, X, Y, PD)}$

$\text{REAL*8 Y(N), PD(N,N), X}$

$\text{RETURN}$

$\text{END}$

---

**THIS SUBROUTINE COMPUTES THE VALUES OF THE DERIVATIVE OF THE DDVDT ARRAY AS A FUNCTION OF THE DEPENDENT VARIABLE (DV).**

**SINCE THE FINITE DIFFERENCE OPTION IS USED IN THE CALL TO DGEAR (I.E. MITER=-2) THIS ROUTINE IS A BLANK.**

**SUBROUTINE FCNJ(N, X, Y, PD)**

**REAL*8 Y(N), PD(N,N), X**

**RETURN**

**END**

---
THIS SUBROUTINE USES THE EXPERIMENTAL DATA INPUT AND THE MODEL SIMULATION OUTPUT TO COMPUTE THE MODEL RESIDUAL, RESIDUAL SQUARED, SUM OF RESIDUALS, AND SUM OF RESIDUALS SQUARED. THESE COMPUTATIONS ARE OUTPUT TO UNIT 10OUT.

**VARIABLE MAP:**

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>TIME STEP INDEX</td>
</tr>
<tr>
<td>IOIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>IOUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NDOF</td>
<td>INTEGER*4</td>
<td>NUMBER OF DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>NDV</td>
<td>INTEGER*4</td>
<td>INDEX OF OUTLET CONCENTRATION</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF POINTS SIMULATED</td>
</tr>
<tr>
<td>NRAD</td>
<td>INTEGER*4</td>
<td>NUMBER OF RADIAL STEPS</td>
</tr>
<tr>
<td>CNORM</td>
<td>REAL*8</td>
<td>NORMALIZED OUTLET CONCENTRATION</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>RDSQSM</td>
<td>REAL*8</td>
<td>SUM OF RESIDUALS SQUARED</td>
</tr>
<tr>
<td>RSD</td>
<td>REAL*8</td>
<td>RESIDUAL OF POINT AT OUTLET</td>
</tr>
<tr>
<td>RSDSQ</td>
<td>REAL*8</td>
<td>RESIDUAL SQUARED OF POINT AT OUTLET</td>
</tr>
<tr>
<td>RSDSUM</td>
<td>REAL*8</td>
<td>SUM OF RESIDUALS</td>
</tr>
</tbody>
</table>

**SUBROUTINE REPORT (1, DV, RSDSUM, RDSQSM, RSD)**

```fortran
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, RHQ, THETA, ZCOL, VEL2, EPSLON, CO, FK, FN, RADIUS, DS, FILMK,
               FKAPPA, SKAPPA, C, T, NCOL, NRAD, NOUT,
               NSTEPS, ICON
COMMON /UNITS/ IOIN, IOUT
NDV = NCOL + (NCOL - 1) * NRAD
IF (I.EQ. 1) THEN
  WRITE (IOUT, 1000)
  WRITE (IOUT, 1100)
  WRITE (IOUT, 1200)
  WRITE (IOUT, 1100)
  RDSUM = 0.000
  RDSQSM = 0.000
ENDIF
CNORM = DV (NDV) / CO
RSD = DATA (1,2) - CNORM
RSDSQ = RSD * RSD
RSDSUM = RSDSUM + RSD
RDSQSM = RDSQSM + RSDSQ
```
```plaintext
WHITE (IOOUT, 1300) DATA (1,1), DATA (1,2), CNORM, RSD, RDSQ
IF (I .GE. NOUT) THEN
  WRITE (IOOUT, 1100)
  WRITE (IOOUT, 1400) RSDSUM, RDSQSM
END IF
1000 FORMAT ('1',58X,'RESIDUAL REPORT')
1100 FORMAT (•1',58X,'------------------------',
  '
  ...
  ')
1200 FORMAT ( ' 0M9X. ' TIME', lOX, 'C/Co (INPUT)', lOX, 'C/Co (PREDICTED)',
  '14X,'RESIDUAL',lOX,'RESIDUAL SQUARED')
1300 FORMAT (' ',IIX,E12.5,lOX,E12,5,14X,E12.5,lOX,E12.5,14X,E12.5)
1400 FORMAT (' ',SIX,E12.5,14X,E12.5)
RETURN
END

THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE
COMPLETION OF SIMULATION FOR EXPERIMENTAL COMPARISON.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTEGER*4</td>
<td>LOOP INDEX</td>
</tr>
<tr>
<td>IFLAG1</td>
<td>INTEGER*4</td>
<td>FLAG FOR ABRUPT CONCENTRATION CHANGE</td>
</tr>
<tr>
<td>IFLAG2</td>
<td>INTEGER*4</td>
<td>FLAG FOR DESORPTION</td>
</tr>
<tr>
<td>I0IN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NOUT</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS TO OUTPUT</td>
</tr>
<tr>
<td>BAL</td>
<td>REAL*8</td>
<td>RATIO OF MASS IN TO MASS ACCOUNTED</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DATA</td>
<td>REAL*4</td>
<td>EXPERIMENTAL INPUT DATA</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>OUTCON</td>
<td>REAL*8</td>
<td>EXIT CONCENTRATION VECTOR</td>
</tr>
<tr>
<td>RCOL</td>
<td>REAL*8</td>
<td>REAL EQUIVALENT OF NCOL</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>TOTDRY</td>
<td>REAL*8</td>
<td>TOTAL MASS LEFT ON SOLID PHASE</td>
</tr>
<tr>
<td>TOTINF</td>
<td>REAL*8</td>
<td>TOTAL MASS INJECTED INTO COLUMN</td>
</tr>
<tr>
<td>TOTOUT</td>
<td>REAL*8</td>
<td>TOTAL MASS TO EXIT COLUMN</td>
</tr>
<tr>
<td>TOTWET</td>
<td>REAL*8</td>
<td>TOTAL MASS LEFT IN LIQUID PHASE</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL1</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td></td>
</tr>
</tbody>
</table>
```

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, IOUT

C
IFLAG1 = 0
IFLAG2 = 0
C
C.....STORE THE INITIAL VELOCITY
VEL1 = VEL
C
C.....CALCULATE TOTAL CONTAMINANT INJECTED

CALCULATE CONTAMINANT INJECTED DURING FIRST REPORT PERIOD.

TOTINF = CO * DATA (1, 1) * VEL * THETA +
1 0.5 * (DATA (1, 3) - CO) * DATA (1, 1) * VEL * THETA

CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS

DO 100 I = 2, (NOUT - 1)
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
1 IFLAG1 = 1
1 TOTINF = TOTINF +
1 DATA (I, 3) * (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
ELSE
1 TOTINF = TOTINF +
1 DATA (I - 1, 3) *
1 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA +
1 0.5 * (DATA (I, 3) - DATA (I - 1, 3)) * 
1 (DATA (I, 1) - DATA (I - 1, 1)) * VEL * THETA
END IF

100 CONTINUE
C
C.....CALCULATE TOTAL EFFLUENT RELEASED

CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF
THE FIRST REPORT PERIOD

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 PERIOD

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
PERIOD

DO 110 I = 2, (NOUT - 1)

RESET VELOCITY IF DESORPTION BEGINS

IF (DATA (1, 3) .LT. 1.0E-10) THEN
VEL = VEL2
IFLAG2 = 1
ENDIF

TOTOUT = TOTOUT +
1 (0.75 * OUTCON (I) + 0.25 * OUTCON (I - 1)) *
2 (DATA (I, 1) - DATA (I - 1, 1)) / 2.0 *
3 (0.75 * OUTCON (I) + 0.25 * OUTCON (I + 1)) *
4 (DATA (I + 1, 1) - DATA (I, 1)) / 2.0 *
5 VEL * THETA

110 CONTINUE

CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST REPORT PERIOD

TOTOUT = TOTOUT +
1 (0.75 * OUTCON (NOUT) +
2 0.25 * OUTCON (NOUT - 1)) *
3 (DATA (NOUT, 1) - DATA (NOUT - 1, 1)) / 2.0 *
4 VEL * THETA

RESTORE VELOCITY
VEL = VEL1

..CALCULATE CONTAMINANT REMAINING IN FLUID PHASE

CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE

RCOL = REAL (NCOL)
TOTWET = (0.75 * DV (1) + 0.25 * DV (NRAD + 2)) *
1 ZCOL / (RCOL - 1.0) / 2.0 * THETA

CALCULATE CONAMINANT IN FLUID NEAR TYPICAL NODE

DO 120 I = (NRAD + 2), ((NCOL - 2) * NRAD + NCOL - 1), (NRAD + 1)
TOTWET = TOTWET +
1 (0.75 * DV (1) + 0.125 * DV (1 - NRAD - 1)) *
2 0.125 * DV (1 + NRAD + 1)) *
3 ZCOL / (RCOL - 1.0) * THETA
120 CONTINUE

CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE

TOTWET = TOTWET + (0.25 * DV ((NCOL - 2) * NRAD + NCOL - 1) *
1 0.75 * DV ((NRAD + 1) * (NCOL - 1) + 1)) *
2 ZCOL / (RCOL - 1.0) / 2.0 * THETA

..CALCULATE CONTAMINANT REMAINING ON SOLID PHASE

TOTDRY = 0.0
WGTSUM = 0.0
RRAD = REAL (NRAD)

CONSIDER EACH PARTICLE RADIUS ...
DO 140 IRAD = 2, NRAD + 1, 1
RIRAD = REAL (IRAD)
RADSUM = 0.0

...AND CONSIDER THAT RADIUS OVER THE ENTIRE COLUMN LENGTH

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)))
   THEN
   RADSUM = RADSUM + 0.50 * DV (ICOL)
   ELSE
   RADSUM = RADSUM + DV (ICOL)
   END IF
130 CONTINUE

CALCULATE THE AVERAGE CONCENTRATION AT THIS RADIUS
RADAVG = RADSUM / (RCOL - 1.DO)

CALCULATE APPROPRIATE WEIGHTING FOR THIS RADIUS BASED
ON ITS CONTRIBUTION TO THE OVERALL PARTICLE MASS
RADWGT = 3 * (((RIRAD - 1.DO) / RRAD) ** 2) * 1.DO / RRAD
WGTSUM = WGTSUM + RADWGT

CREDIT AN APPROPRIATE CONTRIBUTION FROM THIS RADIUS
TOTDRY = TOTDRY + RADAVG * ZCOL * RHO * RADWGT *
1   (1.DO - THETA)

140 CONTINUE

THE WEIGHTING FACTORS SHOULD SUM TO UNITY. IN COARSE
DISCRETIZATION THEY MAY NOT DO SO PRECISELY. APPLY A
BLANKET CORRECTION.

TOTDRY = TOTDRY / WGTSUM

....COMPUTE MASS BALANCE

BAL = TOTINF / (TOTOUT + TOTWET + TOTDRY)

....OUTPUT THE MASS BALANCE FINDINGS

WRITE (IOUT, 1000)
WRITE (IOUT, 1100)
WRITE (IOUT, 1200) TOTINF
WRITE (IOUT, 1300) TOTOUT
WRITE (IOUT, 1400) TOTWET
WRITE (IOUT, 1500) TOTDRY
WRITE (IOUT, 1600) BAL
   IF (IFLAG1 .EQ. 1) THEN
   WRITE (IOUT, 1700)
   ENDIF
   IF (IFLAG2 .EQ. 1) THEN
   WRITE (IOUT, 1800)
   ENDIF

1000 FORMAT ('1', 35X, 'EXPERIMENTAL MASS BALANCE REPORT')
1100 FORMAT ('0----------------------------------------------------------',
1   '----------------------------------------------------------');
C C C C C C C C C C C C C C C C C C C C C C C C C C C

RETURN END

THIS SUBROUTINE PERFORMS A MASS BALANCE CHECK AFTER THE COMPLETION OF SIMULATION IN SIMULATION MODE.

VARIABLE MAP:

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>PRECISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFLAG1</td>
<td>INTEGER*4</td>
<td>LOOP INDEX</td>
</tr>
<tr>
<td>IFLAG2</td>
<td>INTEGER*4</td>
<td>FLAG FOR DESORPTION</td>
</tr>
<tr>
<td>IDIN</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO READ FROM</td>
</tr>
<tr>
<td>I0OUT</td>
<td>INTEGER*4</td>
<td>UNIT NUMBER TO WRITE TO</td>
</tr>
<tr>
<td>NCOL</td>
<td>INTEGER*4</td>
<td>NUMBER OF NODES IN Z DIRECTION</td>
</tr>
<tr>
<td>NSTEPS</td>
<td>INTEGER*4</td>
<td>NUMBER OF TEMPORAL STEPS</td>
</tr>
<tr>
<td>BAL</td>
<td>REAL*8</td>
<td>RATIO OF MASS IN TO MASS ACCOUNTED</td>
</tr>
<tr>
<td>CO</td>
<td>REAL*8</td>
<td>INITIAL CONCENTRATION</td>
</tr>
<tr>
<td>DV</td>
<td>REAL*8</td>
<td>DEPENDENT VARIABLE VECTOR</td>
</tr>
<tr>
<td>OUTCON</td>
<td>REAL*8</td>
<td>EXIT CONCENTRATION VECTOR</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>REAL*4</td>
<td>SIMULATION STATUS MATRIX</td>
</tr>
<tr>
<td>RCOL</td>
<td>REAL*8</td>
<td>REAL EQUIVALENT OF NCOL</td>
</tr>
<tr>
<td>THETA</td>
<td>REAL*8</td>
<td>POROSITY</td>
</tr>
<tr>
<td>TOTALR</td>
<td>REAL*8</td>
<td>TOTAL MASS LEFT ON SOLID PHASE</td>
</tr>
<tr>
<td>TOTALINF</td>
<td>REAL*8</td>
<td>TOTAL MASS INJECTED INTO COLUMN</td>
</tr>
<tr>
<td>TOTALOUT</td>
<td>REAL*8</td>
<td>TOTAL MASS TO EXIT COLUMN</td>
</tr>
<tr>
<td>TOTALWET</td>
<td>REAL*8</td>
<td>TOTAL MASS LEFT IN LIQUID PHASE</td>
</tr>
<tr>
<td>VEL</td>
<td>REAL*8</td>
<td>PORE VELOCITY</td>
</tr>
<tr>
<td>VEL1</td>
<td>REAL*8</td>
<td>SORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>VEL2</td>
<td>REAL*8</td>
<td>DESORPTION PORE VELOCITY</td>
</tr>
<tr>
<td>ZCOL</td>
<td>REAL*8</td>
<td>LENGTH OF COLUMN (Z DIRECTION)</td>
</tr>
</tbody>
</table>
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/ 10IN, 10OUT
C
IFLAG1 = 0
IFLAG2 = 0
C.....STORE THE INITIAL VELOCITY
VEL1 = VEL
C.....CALCULATE TOTAL CONTAMINANT INJECTED
CALCULATE CONTAMINANT INJECTED DURING FIRST REPORT PERIOD.

TOTINF = CO * OUTPUT (1, 1) * VEL * THETA + 1 0.5 * (OUTPUT (1, 4) - CO) * OUTPUT (1, 1) * 2 VEL * THETA

CALCULATE CONTAMINANT INJECTED DURING TYPICAL REPORT PERIODS
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 1 IFLAG1 = 1 2 TOTINF = TOTINF + 1 OUTPUT (I, 4) * 2 (OUTPUT (I, 1) - OUTPUT (I - 1, 1)) * 3 VEL * THETA
ELSE
1 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.....CALCULATE TOTAL EFFLUENT RELEASED
CALCULATE EFFLUENT RELEASED DURING THE FIRST HALF OF THE FIRST REPORT PERIOD

TOTOUT = (0.75 * 0.25 * OUTCON (1)) * 1 OUTPUT (1, 1) / 2.0 * VEL * THETA

CALCULATE EFFLUENT RELEASED AROUND THE FIRST REPORT PERIOD

TOTOUT = TOTOUT + 1 (0.75 * OUTCON (1) + 0.25 * 0.25 * 0.25) * 2 (OUTPUT (1, 1) - 0.25) / 2.0 *
CALCULATE EFFLUENT RELEASED AROUND A TYPICAL REPORT PERIOD

DO 110 I = 2, (NSTEPS - 1)

RESET VELOCITY IF DESORPTION BEGINS

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

CALCULATE EFFLUENT RELEASED DURING LAST HALF OF LAST REPORT PERIOD

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

RESTORE VELOCITY

VEL = VEL1

CALCULATE CONTAMINANT REMAINING IN FLUID PHASE

CALCULATE CONTAMINANT IN FLUID NEAR FIRST NODE

RCOL = REAL (NCOL)
TOTWET = (0.75 * DV (1) + 0.25 * DV (NRAD + 2)) *
1 ZCOL / (RCOL - 1.DO) / 2.DO * THETA

CALCULATE CONTAMINANT IN FLUID NEAR TYPICAL NODE

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

CALCULATE CONTAMINANT IN FLUID NEAR EXIT NODE

TOTWET = TOTWET + (0.25 * DV ((NCOL - 2) + NRAD + NCOL - 1) +
1 0.75 * DV ((NRAD + 1) * (NCOL - 1)) *
2 ZCOL / (RCOL - 1.DO) / 2.DO * THETA

CALCULATE CONTAMINANT REMAINING ON SOLID PHASE

TOTDRY = O.DO
WGTSUM = O.DO
RRAD = REAL (NRAD)

CONSIDER EACH PARTICLE RADIUS ...

DO 140 IRAD = 2, NRAD + 1, 1
    RIRAD = REAL (IRAD)
    RADSUM = 0.DO

...AND CONSIDER THAT RADIUS OVER THE ENTIRE COLUMN LENGTH

DO 130 ICOL = IRAD, ((NRAD + 1) * (NCOL - 1) + IRAD), 1 (NRAD + 1)
    IF ((ICOL .LE. IRAD) .OR. (ICOL .GE. ((NRAD + 1) * (NCOL - 1) + IRAD)))
      THEN
      RADSUM = RADSUM + 0.50 * DV (ICOL)
    ELSE
      RADSUM = RADSUM + DV (ICOL)
    ENDIF
    CONTINUE

CALCULATE THE AVERAGE CONCENTRATION AT THIS RADIUS
    RADAVG = RADSUM / (RCOL - L.DO)

CALCULATE APPROPRIATE WEIGHTING FOR THIS RADIUS BASED
ON ITS CONTRIBUTION TO THE OVERALL PARTICLE MASS
    RADWGT = 3 * (((RIRAD - L.DO) / RRAD) ** 2) / RRAD
    WGTSUM = WGTSUM + RADWGT

CREDIT AN APPROPRIATE CONTRIBUTION FROM THIS RADIUS
    TOTDRY = TOTDRY + RADAVG * ZCOL * RHO * RADWGT * (1.DO - THETA)

CONTINUE

THE WEIGHTING FACTORS SHOULD SUM TO UNITY. IN COARSE
DISCRETIZATION THEY MAY NOT DO SO PRECISELY. APPLY A
BLANKET CORRECTION.
    TOTDRY = TOTDRY / WGTSUM

COMPUTE MASS BALANCE
    BAL = TOTINF / (TOTOUT + TOTWET + TOTDRY)

OUTPUT THE MASS BALANCE FINDINGS
    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

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 DESORPTION. ')

RETURN
END