Numerical Simulation of Advective-Dispersive Multisolute Transport with Sorption, Ion Exchange and Equilibrium Chemistry

By Frank M. Lewis, Clifford I. Voss, and Jacob Rubin

U.S. GEOLOGICAL SURVEY
Water-Resources Investigations Report 86—4022

Reston, Virginia
1986

UNITED STATES DEPARTMENT OF THE INTERIOR

Donald Paul Hodel, Secretary

GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

Project Chief
U.S. Geological Survey
431 National Center
Reston, Virginia 22092

Copies of this report can be
purchased from:

Open-File Services Section
Western Distribution Branch
U.S. Geological Survey
Box 25425, Federal Center
Denver, Colorado 80225

CONTENTS

ILLUSTRATIONS

Figure Page

CONVERSION FACTORS

The metric (International) system of measurement used in this report may be converted to inch-pound units by using the following conversion factors:

Multiply Metric Unit	By	To Obtain Inch-Pound Unit
meter (m)	3.281	foot (ft)
kilometer (km)	0.6214	mile (mi)
meter per second (m/s)	3.281	foot per second (ft/s)
gram (g)	2.205×10^{-3}	pound-mass (lb-m)
	6.854×10^{-5}	slug
kilogram (kg)	2.2105	pound-mass (lb-m)
	6.854×10^{-2}	slug
gram per cubic centimeter (g/cm^3)	62.59	pound-mass per cubic foot ($lb\text{-}m/ft^3$)
	1.942	slug per cubic foot ($slug/ft^3$)

NUMERICAL SIMULATION OF ADVECTION-DISPERSION MULTISOLUTE TRANSPORT WITH SORPTION, ION EXCHANGE AND EQUILIBRIUM CHEMISTRY

by

Frank M. Lewis, Clifford I. Voss, and Jacob Rubin

ABSTRACT

A model is introduced that can simulate the effect of certain chemical and sorption reactions simultaneously among solutes involved in advective-dispersive transport through porous media. The model is based on a methodology that utilizes physical-chemical relationships in the development of the basic solute mass-balance equations; however, the form of these equations allows their solution to be obtained by methods that do not depend on the chemical processes. The chemical environment is governed by the condition of local chemical equilibrium, and may be defined either by the linear sorption of a single species and two soluble complexation reactions which also involve that species, or binary ion exchange and one complexation reaction involving a common ion. Partial differential equations that describe solute mass balance entirely in the liquid phase are developed for each tenad (a chemical entity whose total mass is independent of the reaction process) in terms of their total dissolved concentration. These equations are solved numerically in two dimensions through the modification of an existing ground-water flow/transport computer code.

Following the solution of the transport equations, a system of algebraic expressions that define the chemical interactions is solved simultaneously to obtain the particular species concentration from the transported total quantities. Examples are presented that demonstrate the individual and combined effects of aqueous reactions and either linear sorption or ion exchange on the simultaneous transport of multiple solutes in chemical equilibrium.

INTRODUCTION

Purpose

The principal objectives of this report are to (1) introduce a methodology for incorporating aqueous equilibrium-controlled reactions, and either linear sorption or binary ion exchange, into the partial differential equations that describe advective-dispersive solute transport; (2) present a model, SATRA-CHEM, that utilizes this methodology; and (3) use the results produced by the model when simulating various combinations of reactions to demonstrate some of the processes important in the simultaneous transport of multiple interacting solutes.

Previous Work

The traditional approach to modeling problems of ground-water contamination involves a numerical solution to the advection-dispersion equation describing the transport of a single, nonreactive, solute (e.g.,Bredehoeft and Pinder, 1973; Pinder, 1973; Konikow and Bredehoeft, 1974). Additionally, several investigators (e.g., Holly and Fenske, 1968; Lai and Jurinak, 1972; Gupta and Greenkorn, 1973; Pickens and Lennox, 1976) have incorporated solute-porous media interactions through a source/sink term in their respective mass-balance equations. Consequently, mechanisms such as adsorption and ion exchange have been accounted for, accommodating an additional level of complexity in the simulation of single-solute transport.

Problems posed in natural systems, however, frequently involve several solutes that not only interact with the porous media, but among themselves. Investigations of multicomponent reactive transport are not new. Solution techniques have been developed since the early 1970's that account for multiple solutes participating in various chemical reactions. Jennings et al. [1982], Miller and Benson [1983] and Cederberg et al. [1985] have all noted in their work that two fundamentally different approaches have been followed in the course of developing models to simulate reactive-solute transport. The first approach separates the partial differential equations (PDEs) that track solute-mass balances through advection and dispersion from the algebraic expressions that define various chemical processes such as sorption and chemical equilibria.

In this approach, the solution is obtained by iterating between the two sets of equations: solving first for physical transport alone, followed by the simulation of chemical interaction until a satisfactory level of convergence is reached. This "two-step" approach has been employed in studies by Grove and Wood [1979], Narasimhan et al. [1984], and Cederberg et al. [1985] among others. The approach is potentially very powerful especially when nonreactive transport models are coupled with available computer codes such as WATEQF (Plummer et al., 1978) and PHREEQE (Parkhurst et al., 1982) which facilitate equilibrium calculations for large chemical systems.

There are certain disadvantages with this approach, however, and some of the principal ones have been discussed by Jennings et al. [1982]. These authors report difficulty in obtaining convergent solutions when iterating between the two types of equation sets, and note that the mechanics of this iteration process have been analyzed only recently. Another disadvantage arises when the chemical equilibria models such as those mentioned above are employed. These computer codes are large in order to be general, and may not be highly efficient inasmuch as the solution generated by the model represents equilibrium in a physically static system. Their use in conjunction with transport models, where the equilibrium code must be activated once for each node, on every iteration, and for all time steps, may potentially lead to excessive computer time and storage requirements.

The second approach involves incorporating the mathematics that define the various chemical interactions directly into the transport equations. The resulting set of equations all have the same mathematical form (either PDE or Algebraic), and hence solutions may be obtained simultaneously in a "one-step" procedure. This general approach has been employed in studies by several investigators including Rubin and James [1973], Jennings et al. [1982], Miller and Benson [1983], and Kirkerner et al. [1984].

The Current Study

This report outlines the development and some applications of a new model, SATRA-CHEM. This model is actually a modified version of the computer code SATRA, which itself is a simplified version of the U. S. Geological Survey flow and solute-transport model SUTRA (Voss, 1984) for the case of fully saturated porous media and constant density fluid. The principal modification in SATRA-CHEM is the incorporation of physical-chemical relationships that describe the interaction among participants in certain classes of chemical reactions developed by Rubin [1983]. Specifically, in its present form, SATRA-CHEM can simulate transport with two types of chemical environments under the condition of local equilibrium: (1) linear sorption of a single species accompanied by up to two aqueous complexation reactions which also involve that species, and (2) binary ion exchange and single complexation reaction involving one of the exchanging species.

The Methodology employed by SATRA-CHEM follows neither of the general approaches discussed above completely, but may be considered more of a hybrid of the two. The method resembles the second ("one-step") approach because all nonlinear terms, which result from the particular simultaneous reaction combinations, are incorporated into coefficients of the transport equation for species that sorb. All of the chemical calcu lations, however, are performed external to the basic solution of the PDEs. An entirely separate solution procedure is used to solve the algebraic equations, which define the aqueous chemistry, than is used to solve the transport equations. Additionally, the coefficients that contain the nonlinear terms are determined explicitly and independently of transport.

The result is a modular mathematical structure that permits the incorporation of chemical mathematics such that the resulting PDEs remain in a form that is compatible with solution algorithms utilized by transport models in general, not only the code SATRA. In other words, the formulation of the basic reactive-transport equations is done in such a way that the particular method used to solve them is minimally dependent on the chemistry. In this way, solute-transport codes developed for general field applications may be altered to accommodate multiple solutes, and various chemical reactions, without significantly altering the basic solution process within the model.

GENERAL MASS-BALANCE FORMULATION

As mentioned above, SATRA-CHEM is a version of the computer code SATRA that has been modified to accommodate the transport of more than one solute as well as two specific sets of chemical reaction combinations. SATRA is a two-dimensional finite-element model that is capable of simulating both groundwater flow and solute transport under fully saturated density-independent conditions. The method through which the model obtains hydraulic heads, however, is not discussed here. (For a complete description of the flow equation and its solution by SATRA (SUTRA) see Voss [1984]).

In general SATRA-CHEM solves a modified form of the classic advection-dispersion equation for fully saturated, heterogeneous porous media which may be anisotropic with respect to hydraulic conductivity, but assumes dispersivities are direction-independent following Voss [1984]. A general form of this equation may be written in terms of a dissolved concentration, C, as

$$\frac{\partial(\varepsilon C)}{\partial t} = \underline{\nabla}\cdot(\varepsilon\underline{\underline{D}}\cdot\underline{\nabla}C) - \underline{\nabla}\cdot(\varepsilon\underline{v}C) - f + QC^{*} \qquad (1)$$

where $\underline{\nabla}$ is a two-dimensional differential operator, ε is the porosity, C is the solute concentration, $\underline{\underline{D}}$ is the total dispersion tensor (which includes the components of both molecular diffusion and hydrodynamic dispersion), $\underline{v}$ is the average fluid velocity, f is the adsorbate source, Q is the volumetric fluid source, and C* is the solute concentration of the fluid source. A listing of all notation used in this report may be found in Appendix A. For reference, equation (1) is equivalent to equation

(2.29) in Voss [1984]; however, the terms are simplified due to the assumptions of (1) full saturation (i.e., S_w=1), (2) no solute mass sources resulting from production reactions (i.e., Γ_w=0), and (3) constant fluid density (ρ). The latter assumption implies ρ can be moved outside the derivatives and subsequently devided out of the equation.

The principal difference between the model SATRA and the modified version, SATRA-CHEM, is the composition of f in equation (1). The term f denotes the transfer of solute mass from solution onto the solid surface as adsorbate, and hence may be viewed as a volumetric adsorbate source. In general, f is a function of solute concentration, adsorbate concentration, and the rate of change in solute concentration, depending on the appropriate sorption isotherm. In SATRA-CHEM, however, f also accounts for the aqueous chemical interactions among species that sorb.

To understand what the adsorbate source represents, f may be written in terms of bulk density (ρ_b) and an adsorbate flux (f_s) as:

$$f = \rho_b f_s \tag{2}$$

The adsorbate flux, in turn, represents the solute-mass adsorption rate per unit mass of solid matrix $[(M/M_s)/t]$, and may be expanded into a general form, (see equation (2.32b) in Voss [1984]), as:

$$f_s = k_1 \frac{\partial C}{\partial t} + k_2 C + k_3 \tag{3}$$

such that a variety of sorption models may be accommodated through the the coefficients k_1 $[L^3/M_s]$, k_2 $[(M_s/M)/t]$ and k_3 $[(M/M_s)/t]$. In this study, however, the adsorbate flux is defined only for either linear sorption or binary ion exchange. In these cases, k_2C vanishes and equation (3) becomes

$$f_s = k_1 \frac{\partial C}{\partial t} + k_3$$

or, because k_3 is merely a constant and can be renamed k_2,

$$f_s = k_1 \frac{\partial C}{\partial t} + k_2 \qquad (4)$$

As we will show, the coefficients k_1 and k_2 not only account for these sorptive processes, but they account for aqueous interactions among dissolved solutes that also sorb. As a result, we will fully define these coefficients in the next section according to the nature of the chemical interaction.

For now, the adsorbate source may be expressed in general as

$$f = (1-\varepsilon)\rho_s \left(k_1 \frac{\partial C}{\partial t} + k_2\right) \qquad (5)$$

where ρ_s is the solid-grain density $[M_s/L^3]$. Substituting (3) into (1) and rearranging, results in

$$\frac{\partial(\varepsilon C)}{\partial t} + [(1-\varepsilon)\rho_s k_1]\frac{\partial C}{\partial t} - \underline{\nabla}\cdot(\varepsilon\underline{\underline{D}}\cdot\underline{\nabla}C) + \underline{\nabla}\cdot(\varepsilon\underline{v}C) = QC^* - (1-\varepsilon)\rho_s k_2 \qquad (6)$$

Further modification of the basic transport equation into the form solved by the model requires the concept of mass-conservative solute balance to be introduced (Voss, 1984). At present, equation (6) contains redundant expressions for the time rate-of-change in solute mass per unit total solid matrix volume due to the way in which the head solutions are obtained prior to transport. The flow simulation is performed basically through a fluid-mass balance at every point in the ground-water system. In its most fundamental form this fluid-mass balance is defined on a per unit volume basis as

$$\frac{\partial(\varepsilon\rho)}{\partial t} = - \underline{\nabla}\cdot(\varepsilon\rho\underline{v}) + Q\rho \qquad (7)$$

where ρ is the fluid density. An important factor in determining the amount of total fluid mass in a unit volume of porous media is the temporal change in the storage capability of the media due to fluctuations in hydraulic head. If the fluid density is assumed to remain constant, this factor is implicitly accounted for in the mass-balance relationship (7) when that expression is multiplied by concentration, C. The resulting product represents the changes in solute mass caused by variations in fluid volume within a unit volume of porous media:

$$C \frac{\partial(\varepsilon)}{\partial t} + C\underline{\nabla}\cdot(\varepsilon\underline{v}) = QC \qquad (8)$$

When (8) is subtracted from equation (6), the redundant mass-balance information is eliminated and the result is

$$[\varepsilon + (1-\varepsilon)\rho_s k_1] \frac{\partial C}{\partial t} - \underline{\nabla}\cdot(\varepsilon\underline{\underline{D}}\cdot\underline{\nabla}C) + \varepsilon\underline{v}\cdot\underline{\nabla}C = Q(C^*-C)-(1-\varepsilon)\rho_s k_2 \qquad (9)$$

For simplicity, the principal advective-dispersive components may be combined into the following two-dimensional differential operator:

$$L(C) = \underline{\nabla}\cdot(\varepsilon\underline{\underline{D}}\cdot\underline{\nabla}C) - \varepsilon\underline{v}\cdot\underline{\nabla}C \qquad (10)$$

Using this relationship, the final general form of the solute-transport equation solved by SATRA-CHEM may be written as

$$[\varepsilon + (1-\varepsilon)\rho_s k_1] \frac{\partial C}{\partial t} = L(C) + Q(C^*-C) - (1-\varepsilon)\rho_s k_2 \qquad (11)$$

SPECIFIC MASS-BALANCE FORMULATION

The focus of this study is on the simultaneous transport of several dissolved constituents. Specifically, the constituents involved are expressed in terms of tenads. In a given chemical system, a tenad is best defined as a chemical entity whose total mass is not influenced by the reaction process, regardless of whether it is an active participant or not (Rubin, 1983). In other words, the total mass of a tenad is always conserved within a given system, even if it is sorbed. If a tenad is not sorbed, the definition implies the total dissolved concentration of that tenad is transported conservatively. (Further explanation and specific examples of tenads in chemical systems may be found in Rubin [1983].)

The current version of SATRA-CHEM was developed to simulate the transport of multiple solutes in two different hybrid reaction systems. In the first system, three tenads are allowed to interact through aqueous equilibrium-controlled reactions while one of the three is additionally able to sorb onto the solid matrix. The second system involves four tenads that simultaneously interact through binary ion exchange and an aqueous equilibrium reaction. Both chemical transport systems are simulated by the same computer code. The governing equations have exactly the same form, but each reaction system requires a different mathematical analysis to define the coefficients k_1 and k_2 in equation (11).

Assumptions

In order to facilitate the demonstration of a general approach to reactive transport, the mathematical formulation employed in this study is based on physical conditions that represent a relatively simple chemical environment. The model assumes isothermal conditions with constant fluid density and viscosity. In addition, the porous media considered chemically homogeneous in terms of its reactive capabilities. All reactions are assumed to be reversible and sufficiently fast such that local chemical equilibrium is continually maintained. This implies that the reaction rate greatly exceeds that of the actual physical transport process.

The assumption that local chemical equilibrium prevails is only an approximation of natural systems. In practice, the actual conditions required for instantaneous equilibrium are difficult to determine, and complete equilibrium is rarely attained (Drever, 1982). The issue of whether local equilibrium is an appropriate assumption in modeling reactive solute transport is the subject of recent studies by Valocchi [1985] and Jennings and Kirkner [1984].

The alternative to assuming continuous chemical equilibrium is to account for reaction kinetics which is generally both conceptually and mathematically more complex. Nevertheless, methodologies describing transport involving specific classes of reactions that are not governed by local chemical equilibrium have been developed (e.g., Rubin, 1983).

Further assumptions in this model involve the stoichiometric and activity coefficients all of which, for simplicity, are set equal to one. Additionally, the chemical equilibrium constants are assumed to be independent of space and time.

Equilibrium Sorption and Aqueous Reactions

Following the notation found in Rubin [1983], the first system of chemical reactions that the model can simulate individually, or in any simultaneous combination, is, in generic form:

$$M_1 + \bar{M}_x \qquad \overline{M_1M_x} \qquad \text{(R1)}$$

$$M_1 + M_2 \qquad M_1M_2 \qquad \text{(R2)}$$

$$M_1 + M_4 \qquad M_1M_4 \qquad \text{(R3)}$$

where M_1, M_2, and M_4 are dissolved chemical species composed of only one tenad, namely $\{M_1\}$, $\{M_2\}$, and $\{M_4\}$ respectively, and M_1M_2 and M_1M_4 are dissolved chemical species comprised of two tenads $\{M_1\}$ and either $\{M_2\}$ or $\{M_4\}$. The component $\bar{M}_x$ represents a reactive surface of the porous media, and when combined with $\bar{M}_1$ represents the sorption of M_1. The odd and even nature of the subscripts denotes opposite electrical charges on the chemical components. For reference, even subscripted tenads may be viewed as anions and odd numbered ones as cations. The mathematics of reactive transport as discussed here, however, are independent of these designations.

The concentration of each chemical species in solution will be designated C_i $i=1,2,4$ (e.g., moles per unit volume of fluid) for M_i, and C_{1j} $j=2,4$ for M_1M_j. Accordingly, adsorbate concentrations (e.g., moles per unit mass of solids) will be referred to as $\bar{C}_i$ $i=1,2,4$ and $\bar{C}_{1j}$ $j=2,4$.

Reaction R1 describes the sorption of M_1 from solution onto the surface of the solid matrix and, thus, is considered a surface-heterogeneous reaction (Rubin, 1983). Reactions R2 and R3, on the other hand, occur entirely in the aqueous phase and are, therefore, considered homogeneous reactions. Reactions of this latter sort may describe such processes as complexation, dissociation, oxidation or reduction. In addition, reactions R2 and R3 are each governed by an associated chemical equilibrium constant defined respectively as follows.

$$K_{12} = \frac{C_{12}}{C_1 C_2} \tag{12A}$$

$$K_{14} = \frac{C_{14}}{C_1 C_2} \tag{12B}$$

Reactions Rl through R3 are incorporated into expressions of advective-dispersive transport through the procedure outlined below. The result is a set of PDEs that are identical in form with equation (11). In each case, the final form of the equation is expressed entirely in the liquid phase and in terms of partial derivatives of a single dependent variable. This allows the total dissolved concentrations of the two tenads that occur only in the homogeneous reactions to be transported conservatively and independently, and implies that the mathematics that describe the solid phase interaction of the sorbing tenad are accounted for implicitly.

The first step is to define the basic mass-balance relationships, using the differential operator (10), that describe the two-dimensional transport of each of the three tenads:

for $\{M_1\}$,

$$\varepsilon\frac{\partial C_1}{\partial t} + \varepsilon\frac{\partial C_{12}}{\partial t} + \varepsilon\frac{\partial C_{14}}{\partial t} + \rho_b\frac{\partial \bar{C}_1}{\partial t} =$$

$$L(C_1 + C_{12} + C_{14}) + Q(C_1^* + C_{12}^* + C_{14}^* - C_1 - C_{12} - C_{14}) \tag{13}$$

for $\{M_2\}$,

$$\varepsilon\frac{\partial C_2}{\partial t} + \varepsilon\frac{\partial C_{12}}{\partial t} = L(C_2 + C_{12}) + Q(C_2^* + C_{12}^* - C_2 - C_{12}) \tag{14}$$

and for $\{M_4\}$,

$$\epsilon\frac{\partial C_4}{\partial t} + \epsilon\frac{\partial C_{14}}{\partial t} = L(C_4 + C_{14}) + Q(C_4^* + C_{14}^* - C_4 - C_{14}) \qquad (15)$$

These three equations, along with the two reaction quotients (12A and 12B) comprise the basic set of five equations from which the five unknown dissolved concentrations can be determined (Rubin, 1983).

Each mass-balance expression may be simplified by defining the total dissolved concentration of the tenad in question as a single variable comprised of both the free ion concentration and the concentration in compound form. These new variables are designated U, V and W, where

$$U = C_2 + C_{12} \qquad (16)$$

$$V = C_4 + C_{14} \qquad (17)$$

$$W = C_1 + C_{12} + C_{14} \qquad (18)$$

Source concentrations are designated similarly using the superscript, *. By substituting the relations (16), (17) and (18) into equations (14), (15) and (13) respectively, the transport equations can be expressed more simply in terms of the concentrations U, V and W as

$$\epsilon\frac{\partial W}{\partial t} + \rho_b\frac{\partial \bar{C}_1}{\partial t} = L(W) + Q(W^* - W) \qquad (19)$$

$$\epsilon\frac{\partial U}{\partial t} = L(U) + Q(U^* - U) \qquad (20)$$

$$\epsilon\frac{\partial V}{\partial t} = L(V) + Q(V^* - V) \qquad (21)$$

The adsorbate contribution, $\rho_b \frac{\partial \bar{C}_1}{\partial t}$, in equation (19) is equivalent to f in equation (1), and can be redefined in the solute phase through the appropriate sorption model. If the sorbed concentration $\bar{C}_1$ is assumed to be a function of the concentration in solution, i.e.,

$$\bar{C}_1 = f(C_1)$$

then it follows that

$$\frac{\partial \bar{C}_1}{\partial t} = \frac{\partial \bar{C}_1}{\partial C_1} \frac{\partial C_1}{\partial t} \tag{22}$$

The derivative $\frac{\partial \bar{C}_1}{\partial C_1}$ mathematically represents the partitioning of M_1 between the sorbed and solute phase, and will be called F in all future references. F may directly conform to various sorption isotherms (Bear, 1979); however, in this study, a linear isotherm is assumed. In linear sorption, F is constant and analogous to the equilibrium distribution coefficient, K_d, as defined in Freeze and Cherry [1979] and Reardon [1981] among others, i.e.,

$$\begin{aligned} F &= K_d \\ &= \frac{\text{mass of solute on solid per unit mass of solid}}{\text{mass of solute per unit volume of water}} \\ &= \frac{\bar{C}_1}{C_1} \end{aligned} \tag{23}$$

With F defined, the adsorbate time derivative can now be expressed simply as

$$\frac{\partial \bar{C}_1}{\partial t} = F\frac{\partial C_1}{\partial t} \tag{24}$$

This allows the transport equation (19) for $\{M_1\}$ to be expressed entirely in the liquid phase:

$$\varepsilon\frac{\partial W}{\partial t} + \rho_b F\frac{\partial C_1}{\partial t} = L(W) + Q(W^* - W) \tag{25}$$

Equation (25) may be solved in this form by iterating on C_1 until a solution for W has converged, but it may also be expressed in terms of the partial derivatives a single dependent variable by using the chemical-relation equations and the definitions of U, V and W. $\frac{\partial C_1}{\partial t}$ can be described as a function of $\frac{\partial W}{\partial t}$ by first rearranging the chemical-relation equations (12A) and (12B) respectively, to read:

$$C_{12} = K_{12} C_1 C_2 \tag{26}$$

$$C_{14} = K_{14} C_1 C_4 \tag{27}$$

Next, by combining (12A) with (13), and (12B) with (15), the following expressions for C_2 and C_4 are obtained:

$$C_2 = \frac{U}{(1+K_{12}C_1)} \tag{28}$$

$$C_4 = \frac{V}{(1+K_{14}C_1)} \tag{29}$$

The subsequent derivatives of these equations with respect to time yield:

$$\frac{\partial C_2}{\partial t} = \left(\frac{1}{(1+K_{12}C_1)}\right)\frac{\partial U}{\partial t} - \frac{K_{12}U}{(1+K_{12}C_1)^2}\ \frac{\partial C_1}{\partial t} \tag{30}$$

and

$$\frac{\partial C_4}{\partial t} = \left(\frac{1}{(1+K_{14}C_1)}\right)\frac{\partial V}{\partial t} - \frac{K_{14}\ V}{(1+K_{14}C_1)^2}\ \frac{\partial C_1}{\partial t} \tag{31}$$

Similarly, if

$$\begin{aligned} W &= C_1 + C_{12} + C_{14} \\ &= C_1(1 + K_{12}C_2 + K_{14}C_4) \end{aligned} \tag{32}$$

then it follows that

$$\frac{\partial W}{\partial t} = \frac{\partial C_1}{\partial t} (1 + K_{12}C_2 + K_{14}C_4) + C_1K_{12}\frac{\partial C_2}{\partial t} + C_1K_{14}\frac{\partial C_4}{\partial t} \qquad (33)$$

Equations (30) through (33) clearly illustrate the origins of the nonlinear components that complicate the final form of the transport equation for W.

The derivation continues with the substitution of equations (30) and (31) for $\frac{\partial C_2}{\partial t}$ and $\frac{\partial C_4}{\partial t}$ in equation (33), and arriving at

$$\frac{\partial W}{\partial t} = \frac{\partial C_1}{\partial t} H + G \qquad (34)$$

where

$$H = 1 + K_{12}C_2 + K_{14}C_4 - \left(\frac{K_{12}}{1+K_{12}C_1}\right)^2 UC_1 - \left(\frac{K_{14}}{1+K_{14}C_1}\right)^2 VC_1 \qquad (35)$$

and

$$G = \left(\frac{K_{12}C_1}{1+K_{12}C_1}\right)\frac{\partial U}{\partial t} + \left(\frac{K_{14}C_1}{1+K_{14}C_1}\right)\frac{\partial V}{\partial t} \qquad (36)$$

Equation (34) may be rewritten explicitly for $\frac{\partial C_1}{\partial t}$ as

$$\frac{\partial C_1}{\partial t} = \frac{1}{H}\frac{\partial W}{\partial t} - \frac{G}{H} \qquad (37)$$

which, in turn, may be introduced into equation (25) such that the transport equation for W becomes

$$\varepsilon\frac{\partial W}{\partial t} + \rho_b F[(\frac{1}{H}) \frac{\partial W}{\partial t} - \frac{G}{H}] = L(W) + Q(W^* - W) \qquad (38)$$

After substituting grain density for bulk density, this equation may be rearranged to read:

$$[\varepsilon + (1-\varepsilon)\rho_s \frac{F}{H}]\frac{\partial W}{\partial t} = L(W) + Q(W^* - W) + (1-\varepsilon)\rho_s\frac{FG}{H} \qquad (39)$$

Equation (39) along with equations (20) and (21) comprise the basic set of transport equations solved by SATRA-CHEM when linear sorption and homogeneous aqueous reactions occur together. All three equations have the same form as the more general equation (11), but are derived specifically to account for the simultaneous reactions R1 through R3. In this case, the coefficients k_1 and k_2 in (11) are both zero for equations (20) and (21), but in equation (39) they are defined through the relationships:

$$k_1 = F/H \tag{40}$$

$$k_2 = -FG/H \tag{41}$$

All nonlinear components resulting from the combined sorption and chemical interactions are contained in the variables G and H, and the process of linear sorption is incorporated through F.

Ion Exchange and Aqueous Complexation

The second chemical environment the model is capable of simulating during transport is represented by the following simultaneous reactions in generic form:

$$M_1 + \overline{M_3M_e} \quad \overline{M_1M_e} + M_3 \tag{R4}$$

$$M_1 + M_2 \quad M_1M_2 \tag{R5}$$

Reaction R4 describes binary exchange between M_1 and M_3 in which a tenad in solution interchanges with another tenad in the sorbed phase. The common factor for both tenads is the cation exchanger, $\overline{M}_e$. Like R1, R4 is a surface heterogeneous reaction, and R5, which is identical to R2, is a homogeneous reaction (Rubin, 1983) representing such processes as aqueous complexation.

The algebraic equations that further define this chemical system are the chemical-relation equations

$$K_{13} = \frac{\bar{C}_1 C_3}{C_1 \bar{C}_3} \qquad (42)$$

$$K_{12} = \frac{C_{12}}{C_1 C_2} \qquad (43)$$

associated with reactions R4 and R5 respectively, and the exchange capacity of the media

$$\bar{C}_T = \bar{C}_1 + \bar{C}_3 \qquad (44)$$

which is assumed constant.

As in the previous chemical system, the environment defined by R4 and R5 can be coupled with advective-dispersive mass-balance expressions to formulate a set of partial differential equations identical in form with equation (11), and expressed entirely in the liquid phase. In this case, the mass-balance relationships for each of the four tenads are:

for $\{M_1\}$,

$$\varepsilon\frac{\partial C_1}{\partial t} + \varepsilon\frac{\partial C_{12}}{\partial t} + \rho_b\frac{\partial \bar{C}_1}{\partial t} = L(C_1 + C_{12}) + Q(C_1^* + C_{12}^* - C_1 - C_{12}) \qquad (45)$$

$\{M_2\}$,

$$\varepsilon\frac{\partial C_2}{\partial t} + \varepsilon\frac{\partial C_{12}}{\partial t} = L(C_2 + C_{12}) + Q(C_2^* + C_{12}^* - C_2 - C_{12}) \qquad (46)$$

$\{M_3\}$,

$$\varepsilon\frac{\partial C_3}{\partial t} + \rho_b\frac{\partial \bar{C}_3}{\partial t} = L(C_3) + Q(C_3^* - C_3) \qquad (47)$$

and $\{\bar{M}_e\}$,

$$\rho_b\frac{\partial \bar{C}_1}{\partial t} + \rho_b\frac{\partial \bar{C}_3}{\partial t} = 0 \qquad (48)$$

These four expressions, together with relationships (42) through (44), comprise the basic set of equations from which the unknown solute concentrations are determined.

The number of partial differential equations required to define this system may be reduced by one if a mass-balance expression is developed in terms of the total dissolved concentration of chemical species that contain the sorbing tenads. This is accomplished by adding equations (47) to (45). In the process, the relationship in equation (48) is accounted for, and the adsorbate contribution of $\bar{M}_3$ is eliminated from the mass balance. The result is an expression containing three unknown solute concentrations:

$\{M_1\}+\{M_3\}$:

$$\varepsilon\frac{\partial C_1}{\partial t} + \varepsilon\frac{\partial C_{12}}{\partial t} + \varepsilon\frac{\partial C_3}{\partial t} = L(C_1 + C_{12} + C_3) + Q(C_1^* + C_{12}^* + C_3^* - C_1 - C_{12} - C_3) \quad (49)$$

To facilitate the solution process, the combined dissolved concentrations associated with each of the three differential equations are expressed as a single variable. This allows equations (45), (46) and (49), to be transformed into the following:

$$\varepsilon\frac{\partial W}{\partial t} + \rho_b\frac{\partial \bar{C}_1}{\partial t} = L(W) + Q(W^* - W) \quad (50)$$

$$\varepsilon\frac{\partial U}{\partial t} = L(U) + Q(U^* - U) \quad (51)$$

$$\varepsilon\frac{\partial V}{\partial t} = L(V) + Q(V^* - V) \quad (52)$$

where

$$W = C_1 + C_{12} \tag{53}$$

$$U = C_2 + C_{12} \tag{54}$$

$$\begin{aligned} V &= C_1 + C_{12} + C_3 \\ &= W + C_3 \end{aligned} \tag{55}$$

Equation (50) may be expressed entirely in the liquid phase by replacing the time rate-of-change in $\bar{C}_1$ with expressions that are functions of the transported concentrations U, V and W. From the chemical-relation equation (42), it follows that

$$\frac{\partial}{\partial t}(K_{13}C_1\bar{C}_3) = \frac{\partial}{\partial t}(\bar{C}_1 C_3) \tag{56}$$

Rearranging and applying the product rule of differentiation yields:

$$K_{13}C_1\frac{\partial \bar{C}_3}{\partial t} + K_{13}\bar{C}_3\frac{\partial C_1}{\partial t} - \bar{C}_1\frac{\partial C_3}{\partial t} - C_3\frac{\partial \bar{C}_1}{\partial t} = 0 \tag{57}$$

The number of derivatives is reduced by using equation (48) and rewriting $\frac{\partial \bar{C}_3}{\partial t}$ in terms of $\frac{\partial \bar{C}_1}{\partial t}$. This enables common coefficients to be combined such that equation (57) becomes

$$(K_{13}C_1 + C_3)\frac{\partial \bar{C}_1}{\partial t} = K_{13}\bar{C}_3\frac{\partial C_1}{\partial t} - \bar{C}_1\frac{\partial C_3}{\partial t} \tag{58}$$

or

$$\frac{\partial \bar{C}_1}{\partial t} = \frac{1}{g}\left[f_1\frac{\partial C_1}{\partial t} - f_2\frac{\partial C_3}{\partial t}\right] \tag{59}$$

where

$$g = K_{13}C_1 + C_3$$

$$f_1 = K_{13}\bar{C}_3$$

$$f_2 = \bar{C}_1$$

To be of use, however, these coefficients need to be expressed in terms of known or explicitly defined variables. Accordingly, equation (55) together with (42) and (44) are used to replace C_3, $\bar{C}_3$ and $\bar{C}_1$, such that

$$g = K_{13}C_1 + V - W \tag{60}$$

$$f_2 = (K_{13}C_1\bar{C}_T)/g \tag{61}$$

$$f_1 = K_{13}(\bar{C}_T - f_2) \tag{62}$$

The final step is to define explicit expressions for the derivatives of dissolved concentration in equation (59). The derivative of C_3 is easily converted: from equation (55), it follows that

$$\frac{\partial C_3}{\partial t} = \frac{\partial V}{\partial t} - \frac{\partial W}{\partial t} \tag{63}$$

The conversion of $\frac{\partial C_1}{\partial t}$ is slightly more involved. An explicit relationship for C_1 in terms of the combined variables U and W must first be developed prior to determining the differential. By combining equations (43) and (35), C_1 may be defined as

$$C_1 = \frac{W}{(1+K_{12}C_2)} \tag{64}$$

Similarly, using equations (43) and (54),

$$C_2 = \frac{U}{(1+K_{12}C_1)} \tag{65}$$

Equation (65) can now be introduced into equation (64) for C_2, and the result expanded into the following quadratic equation,

$$K_{12}C_1^2 + (1 + K_{12}U - K_{12}W)C_1 - W = 0 \tag{66}$$

which may be solved directly for C_1. Applying the quadratic formula yields:

$$C_1 = \tfrac{1}{2}[W - U - (1/K_{12})] + \tfrac{1}{2}\{[W - U - (1/K_{12})]^2 + 4(1/K_{12})W]\}^{\frac{1}{2}}$$

$$= \tfrac{1}{2}A + \tfrac{1}{2}B$$

$$= \tfrac{1}{2}(A + B) \qquad (67)$$

where

$$A = W - U - (1/K_{12}) \qquad (68)$$

$$B = [A^2 + 4(1/K_{12})W]^{\frac{1}{2}} \qquad (69)$$

The time derivative of C_1 required for equation (59) is best obtained from the quadratic equation. Accordingly, differentiating (66) with respect to time yields:

$$2C_1 \frac{\partial C_1}{\partial t} + [(1/K_{12}) + U - W] \frac{\partial C_1}{\partial t} + C_1\left(\frac{\partial U}{\partial t} - \frac{\partial W}{\partial t}\right) - (1/K_{12})\frac{\partial W}{\partial t} = 0 \quad (70)$$

or

$$[2C_1 + (1/K_{12}) + U - W] \frac{\partial C_1}{\partial t} = [(1/K_{12}) + C_1] \frac{\partial W}{\partial t} - C_1 \frac{\partial U}{\partial t} \qquad (71)$$

This can be simplified in terms of B (equation (69)) as

$$B \frac{\partial C_1}{\partial t} = [(1/K_{12}) + C_1] \frac{\partial W}{\partial t} - C_1 \frac{\partial U}{\partial t} \qquad (72)$$

which implies

$$\frac{\partial C_1}{\partial t} = \frac{1}{B} [(1/K_{12}) + C_1] \frac{\partial W}{\partial t} - \frac{C_1}{B} \frac{\partial U}{\partial t} \qquad (73)$$

By substituting equations (63) and (73) for $\frac{\partial C_3}{\partial t}$ and $\frac{\partial C_1}{\partial t}$ respectively, equation (59) may be rewritten as

$$\frac{\partial \bar{C}_1}{\partial t} = \left(\frac{f_1}{g}\right)\left[\left(\frac{(1/K_{12}) + C_1}{B}\right)\frac{\partial W}{\partial t} - \frac{C_1}{B}\frac{\partial U}{\partial t}\right] - \left(\frac{f_2}{g}\right)\left(\frac{\partial V}{\partial t} - \frac{\partial W}{\partial t}\right)$$

$$= \left(\frac{1}{g}\right)\left\{\frac{f_1[(1/K_{12}) + C_1]}{B} + f_2\right\}\frac{\partial W}{\partial t} - \left(\frac{f_1}{g}\,\frac{C_1}{B}\right)\frac{\partial U}{\partial t} - \left(\frac{f_2}{g}\right)\frac{\partial V}{\partial t} \quad (74)$$

This latter expression may, in turn, be substituted into equation (50), and rearranged to read:

$$\left\{\varepsilon + \rho_b(1/g)\left[f_2 + \frac{f_1[(1/K_{12}) + C_1]}{B}\right]\right\}\frac{\partial W}{\partial t} =$$

$$L(W) + Q(W^* - W) + \rho_b(1/g)\left[\left(\frac{f_1 C_1}{B}\right)\frac{\partial U}{\partial t} + f_2\,\frac{\partial V}{\partial t}\right] \quad (75)$$

Equation (75) together with equations (51) and (52) comprise the basic set of transport equations solved by SATRA-CHEM when binary ion exchange occurs simultaneously with an aqueous complexation reaction involving one of the exchanging ions. Note equation (75) is in the same form as both equations (11) and (39). In this case, the coefficients k_1 and k_2 are defined as

$$k_1 = (1/g)\left[f_2 + \frac{f_1[(1/K_{12}) + C_1]}{B}\right] \quad (76)$$

$$k_2 = (1/g)\left[\left(\frac{f_1 C_1}{B}\right)\frac{\partial U}{\partial t} + f_2\,\frac{\partial V}{\partial t}\right] \quad (77)$$

When binary ion exchange occurs without an accompanying aqueous reaction, the development just described must be adjusted to allow K_{12} to equal zero. As expressed in equation (76) k_1 approaches infinity when $K_{12} = 0$. In chemical systems defined only by R4 (ion exchange only),

$W = C_1$ (from equation (53)); therefore, the quadratic equation (66) is not required, and the variables A and B, both of which contain the term $(1/K_{12})$, are not defined. Moreover, it follows that

$$\frac{\partial W}{\partial t} = \frac{\partial C_1}{\partial t} \tag{78}$$

thus equation (73) is similarly omitted. Consequently, when $K_{12} = 0$, equation (75) becomes

$$\left[\varepsilon + \frac{\rho_b}{g}(f_1 + f_2)\right]\frac{\partial W}{\partial t} = L(W) + Q(W^* - W) + \left(\frac{\rho_b\, f_2}{g}\right)\frac{\partial V}{\partial t} \tag{79}$$

and the coefficients of equations (76) and (77) are redefined as

$$k_1 = (1/g)[f_1 + f_2] \tag{80}$$

$$k_2 = (1/g)f_2\,\frac{\partial V}{\partial t} \tag{81}$$

OVERVIEW OF SOLUTION ALGORITHMS

Sorption and Aqueous Complexation

A schematic diagram of the solution algorithm for the case of transport with reactions R1 through R3 is shown in figure 1. The model first solves the linear transport equations (20) and (21) at the end of a given time step (t^k) for U and V respectively. Equation (39), which is nonlinear due to the variables G and H, is then solved for W at t^k using a 'global' Picard iteration scheme. The term global is used to distinguish this simple (first-order) process from a localized Newton-Raphson procedure contained within each iteration to solve the algebraic equation set.

The first step in solving for W is to determine the values of G and H (equations (35) and (36)) on a given iteration. These variables require U and V at t^k, along with their derivatives with respect to time, and values of C_1, C_2 and C_4. The derivatives are expressed simply as the linear change during the time step (e.g., $\partial U/\partial t = (U^k - U^{k-1})/\Delta t$), and the values of C_1, C_2 and C_4 are obtained from the previous global iteration, or the previous time step if on the first iteration. With F known from the input data, all of the unknowns are accounted for and equation (39) can be solved for W.

Following the calculation of U, V and W, the individual reactant concentrations are obtained by simultaneously solving the algebraic expressions (16) through (18). Subtracting (16) and (17) from (18), and using the relationships (28) and (29), produces a single equation with C_1 as the only unknown:

$$\begin{aligned} W - U - V &= C_1 - C_2 - C_4 \\ &= C_1 - U/(1+K_{12}C_1) - V/(1+K_{14}C_1) \qquad (82) \end{aligned}$$

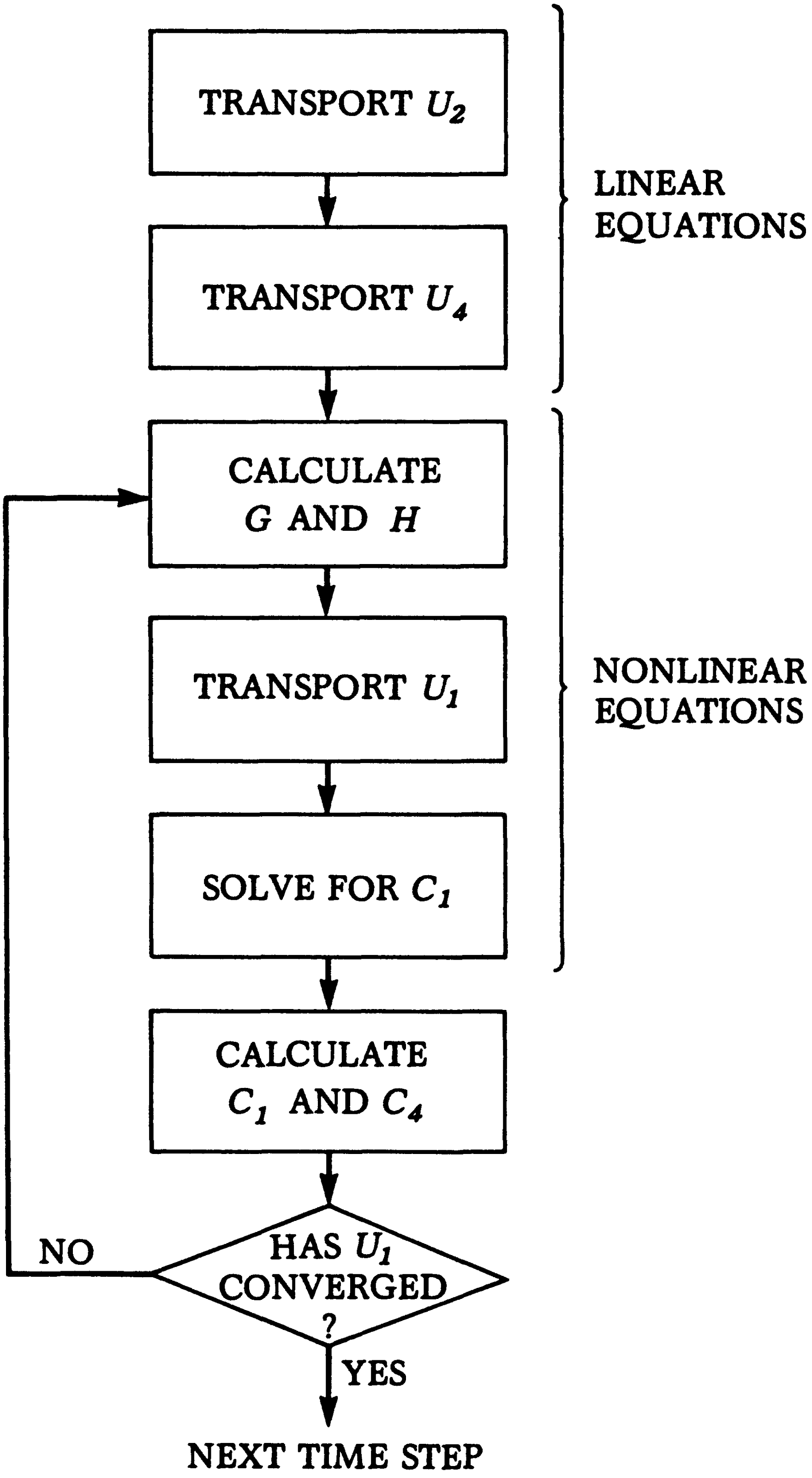

Figure 1. Schematic diagram of solution algorithm for transport with sorption and aqueous complexation.

This equation can be converted into the following cubic expression

$$(K_{12}K_{14})C_1^3 + [K_{12}K_{14}(U+V-W)+(K_{12}+ K_{14})]C_1^2$$

$$+ [1 + K_{12} (U-W) + K_{14}(V-W)]C_1 - W = 0 \qquad (83)$$

which the model solves for C_1 using a Newton-Raphson iteration procedure. This procedure requires the user to supply, at the beginning of each global iteration, an initial guess that is high enough to ensure that the greatest positive root is obtained by the iteration process. A guess on the same order of magnitude as the highest anticipated value of W is adequate. A guess that is too low may cause the Newton-Raphson solution to converge on a negative root. The iterations proceed quickly; so even a poor guess converges rapidly. Once C_1 is known, C_2 and C_4 can be back-calculated from equations (28) and (29). Note the cubic equation can also be solved directly using a formula; however, the coefficients in equation (83) are sufficiently complex that this does not provide a desirable alternative.

Finally, the most recent value of W is checked against the previous global iteration. If the difference between successive values of W is at every node within a user-set tolerance, the model proceeds to the next time step. If not, another global iteration is required and the solution procedure for W is repeated.

When sorption and aqueous complexations do not occur together the algorithm is simplified considerably. Specifically, without homogeneous aqueous reactions equations (16) through (18) reduce to

$$U = C_2$$

$$V = C_4$$

$$W = C_1$$

and without sorption (either with or without homogeneous reactions), equation (39) becomes

$$\epsilon\frac{\partial W}{\partial t} = L(W) + Q(W^{*}-W) \quad (84)$$

Consequently, in both of these cases the resulting transport equations are linear and no global iteration is required. However, in the latter case involving homogeneous reactions, Newton-Raphson iterations are necessary at each time step for which an output of the reactant concentrations is desired.

Ion Exchange and Aqueous Complexation

Although the coefficients k_1 and k_2 appear complex, the solution procedure is similar to that followed in the previous case involving equilibrium sorption. The model first solves the linear transport equations (51) and (52) at t^k for U and V respectively (Figure 2). Equation (75), which is nonlinear, is then solved for W using a first-order (Picard) iterative process. The first step in this process is to determine systematically all variables represented by k_1 and k_2. This is done by solving various explicit expressions, defined previously, in the following order:

1. $A = W - U - (1/K_{12}) \quad (85)$

2. $B = [A + 4(1/K_{12})W]^{\frac{1}{2}} \quad (86)$

3. $C_1 = \frac{1}{2}(A+B) \quad (87)$

4. $g = K_{13}C_1 + V - W \quad (88)$

5. $f_2 = (K_{13}C_1\bar{C}_T)/g \quad (89)$

6. $f_1 = K_{13}(\bar{C}_T - f_2) \quad (90)$

In these calculations values of W represent the previous iteration, or The previous time step if on the first iteration. With K_{12}, K_{13}, and $\bar{C}_T$

known from the input data, equations (76) and (77) can be solved for k_1 and k_2, and equation (75) is then ready to be solved for W. If the difference between values of W for successive iterations is greater than a specified tolerance at any node, new values of k_1 and k_2 are determined and the solution process for W is repeated. If the difference is less than the tolerance, the model proceeds to the next time step.

Once W has converged, the concentrations C_2 and C_3 are calculated from equations (54) and (55). Note this solution procedure does not require Newton-Raphson iterations for the calculation of C_1 because it can be determined directly from the quadratic equation (66) and the variables A and B.

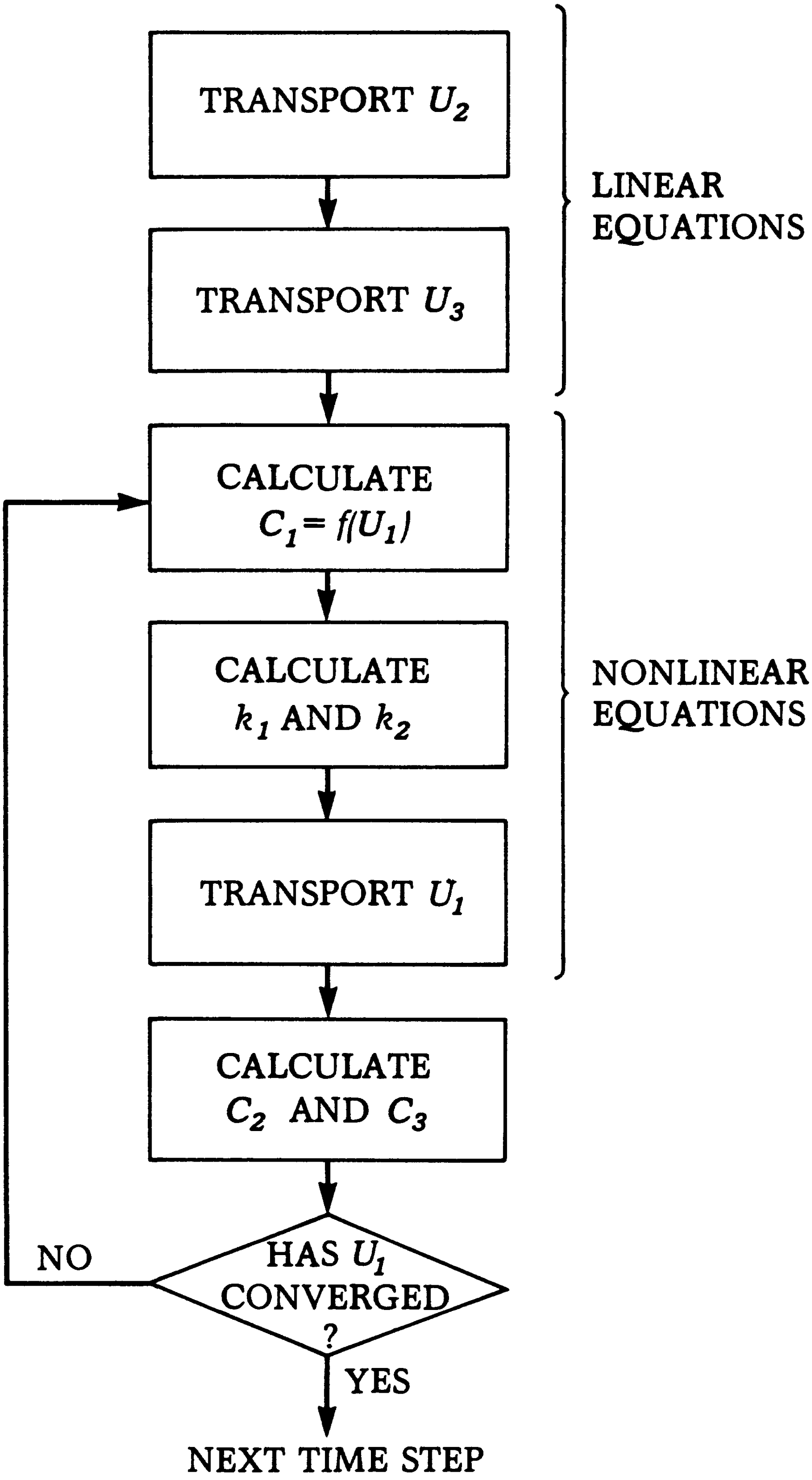

Figure 2. Schematic diagram of solution algorithm for transport with ion exchange and aqueous complexation.

NUMERICAL METHODOLOGY

The numerical technique used in the original transport model SATRA (SUTRA) remains unchanged in the present version of SATRA-CHEM. Equations (30) and (59) can be solved for flow regimes positioned either areally or in cross section using a mesh comprised of quadralateral finite elements.

SATRA-CHEM employs a Galerkin scheme for spatial approximations of the dependent variable that involves symmetric basis functions. Although it is not recommended, the advection term may be approximated using upstream weighting as an option. In that case the basis functions are asymmetric. For either case the approximations take the general form.

$$\psi(x,y,t) \approx \hat{\psi}(x,y,t) = \sum_{j=1}^{N} \psi_j(t)\phi_j(x,y) \qquad (91)$$

where, ψ represents the dependent variable, N is the number of nodes in the mesh, and ϕ_j is the spatial basis function for node j. Following Galerkin orthogonalization, the resulting integral equations are evaluated in local coordinates using two-by-two Gaussian quadrature.

A matrix equation is assembled in which the advective-dispersive expressions in the form of equation (7) reduce to

$$\underline{\underline{\lambda}}\psi + \underline{\underline{\beta}}\psi + \underline{\underline{\chi}}\,\frac{\partial\psi}{\partial t} = \underline{\omega} \qquad (92)$$

where, $\underline{\underline{\lambda}}$ is a symmetric coefficient matrix containing the dispersive components, $\underline{\underline{\beta}}$ is a coefficient matrix (symmetric in the case of no upstream weighting) that contains the advective contributions, $\underline{\underline{\chi}}$ is the coefficient matrix for the time derivative containing the nonlinear terms in k_1 (equations (39) and (76)), and $\underline{\omega}$ is the vector of nodal source contributions including the nonlinear terms in k_2 (equations

(39) and (77)). A "lumped-mass" approach is used that allows the coefficients in $\underline{\underline{\gamma}}$ to be loaded directly onto the main diagonal. The time derivative is discretized through a backwards finite-difference scheme that yields a fully implicit equation in which the matrix is symmetric (in the case of no upstream weighting) and banded.

The upstream weighting option in SATRA-CHEM is available for stabilizing oscillations in the solution due to highly advective transport. This option, however, only results in an increase in the local longitudinal dispersivity by an amount proportional to the distance between successive nodes along the direction of flow. This weighting factor, therefore, was not used in simulations run for this report.

The model allows for hydraulic conductivities to be anisotropic and variable in both direction and magnitude on an elementwise basis throughout the system. Boundary conditions as well as all sinks and sources are permitted to vary with time.

The current version of the code uses a direct Gaussian solver for banded matricies which, while highly accurate, remains inefficient in terms of time and storage requirements for very large problems. A more efficient solver may be included in future versions of the model. More detailed information on the specifics of the numerical method applied in SATRA-CHEM may be found in Voss [1984].

The computer code is compiled in FORTRAN 77 and has been successfully run on a Prime series 850 and 9950 computers maintained by the U.S. Geological Survey in Reston, Virginia.

MODEL TESTING AND APPLICATION

The results of several examples are presented to demonstrate the ability of SATRA-CHEM to simulate the effects of the two reaction systems on the spatial distribution of dissolved concentration during transport. In particular, the individual and combined influence of the reactions within a given system is discussed. Additionally, where possible, an analytical solution is used to verify that modifications of the original code, SATRA (SUTRA), have not been deleterious to the correctness of the basic transport algorithm. Both the accuracy and precision of the numerical solution generated by SATRA (SUTRA) have been successfully tested and are documented in Voss [1984].

Sorption and Aqueous Complexation

Examples under this heading concern the chemical environment characterized by reactions Rl through R3. All simulations involve steady-state flow for simplicity, and represent a physical system with dimensions typical of field rather than laboratory conditions. Although the model is two dimensional, important trends in the results are best illustrated in one dimension; therefore, most examples are presented in this perspective. In these cases, the region modeled may be viewed as a representative stream tube within a larger physical setting. Except where noted, the system is defined by these physical parameters:

Porosity (ε)	0.20	
Hydraulic Conductivity (K)	3.00×10^{-4}	ft/sec
Longitudinal dispersivity (α_L)	100.0	ft
Hydraulic gradient ($\underline{\nabla}h$)	0.0085	
Average pore velocity ($\underline{v}=(K/\varepsilon)\underline{\nabla}h$)	1.28×10^{-5}	ft/sec
Mesh peclet number ($Pe=\Delta x/\alpha_L$)	2.0	

The finite-element mesh used for all one-dimensional simulations is shown in Figure 3. All chemical parameters are in an arbitrary system of consistent units, and are specified separately for each example.

For the one-dimensional system shown in Figure 3 the porous medium is initially saturated with fluid that contains everywhere zero concentration of all tenads, and hence all reactant species, C(x,t):

$$U(x,0) = 0.0 \Rightarrow C_2(x,0) = 0.0$$

$$V(x,0) = 0.0 \Rightarrow C_4(x,0) = 0.0$$

$$W(x,0) = 0.0 \Rightarrow C_1(x,0) = 0.0$$

The boundaries are defined by constant heads, h(x,t), at both ends of the system:

$$h(0,t) = 34.0 \text{ ft}$$

$$h(4000,t) = 0.0 \text{ ft}$$

The simulation commences as fluid flows continuously into the system across the upstream constant head boundary; upon reaching the downstream Dirichlet boundary, the flow is free to exit the system. The inflow solution contains constant concentrations (≥ 0) of the three tenads alone, or with additional nonreactive ions, in combinations such that an electrostatic balance is maintained. Note that an electrostatic balance is a constraint imposed by natural chemical systems and is independent of the mathematical theory employed by the model. In this study, we have assumed electrical neutrality when arbitrary reactions and concentrations are used for the purpose of demonstration. Under field conditions, however, chemical analyses of the source fluid should yield the appropriate balance of anion and cation concentrations automatically.

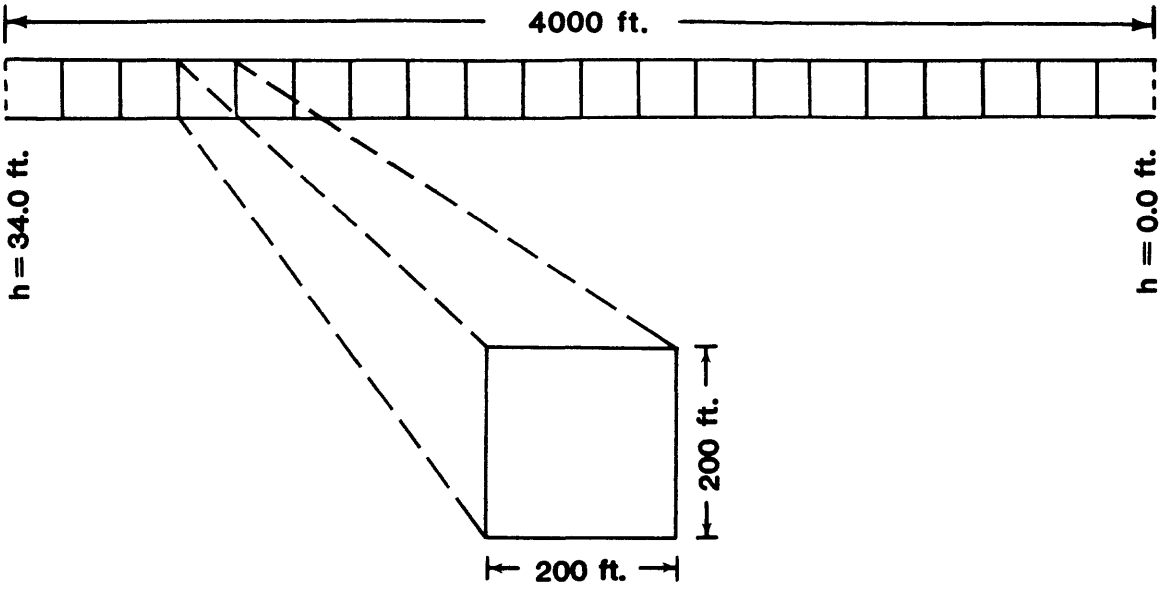

Figure 3. Finite-element mesh and boundary conditions for one-dimensional simulations.

Another important condition that applies to the source fluid is continuous chemical equilibrium among the various components. Aqueous reactions such as R2 and R3 occur in the source fluid if the appropriate components are present and the equilibrium constant is some positive finite value.

The time step interval ($\Delta t = 300$ days) in all one-dimensional examples is selected such that a particle of fluid will progress approximately one element (200 feet) per time step.

Linear Sorption

Single species adsorption has been addressed extensively in the literature (e.g., Lai and Jurinak, 1972; Pickens and Lennox, 1976; Bear, 1979). The well known net effect of this process is to reduce the average pore velocity of the sorbing solute. The amount of this reduction (R) is a function of the porosity, grain density and sorption coefficient, F, and is quantified in the relationship

$$R = \frac{\varepsilon + (1-\varepsilon)\rho_s k_1}{\varepsilon} \qquad (93)$$

This expression can be derived by rewriting equation (11) in terms of a material derivative of concentration with respect to time (Lewis, 1984), and is equivalent to the retardation factors discussed by Lai and Jurinak (1972), and Van Genuchten and Alves (1982) among others. In this case, without additional aqueous reactions, the calculation of R is straight forward because $k_1 = F$. This follows from equation (35) which sets $H = 1.0$ when $k_{12} = k_{14} = 0.0$.

The effect of linear sorption on M_1 without any accompanying aqueous reactions after 14 time steps (approximately 7 years) is shown in Figure 4.

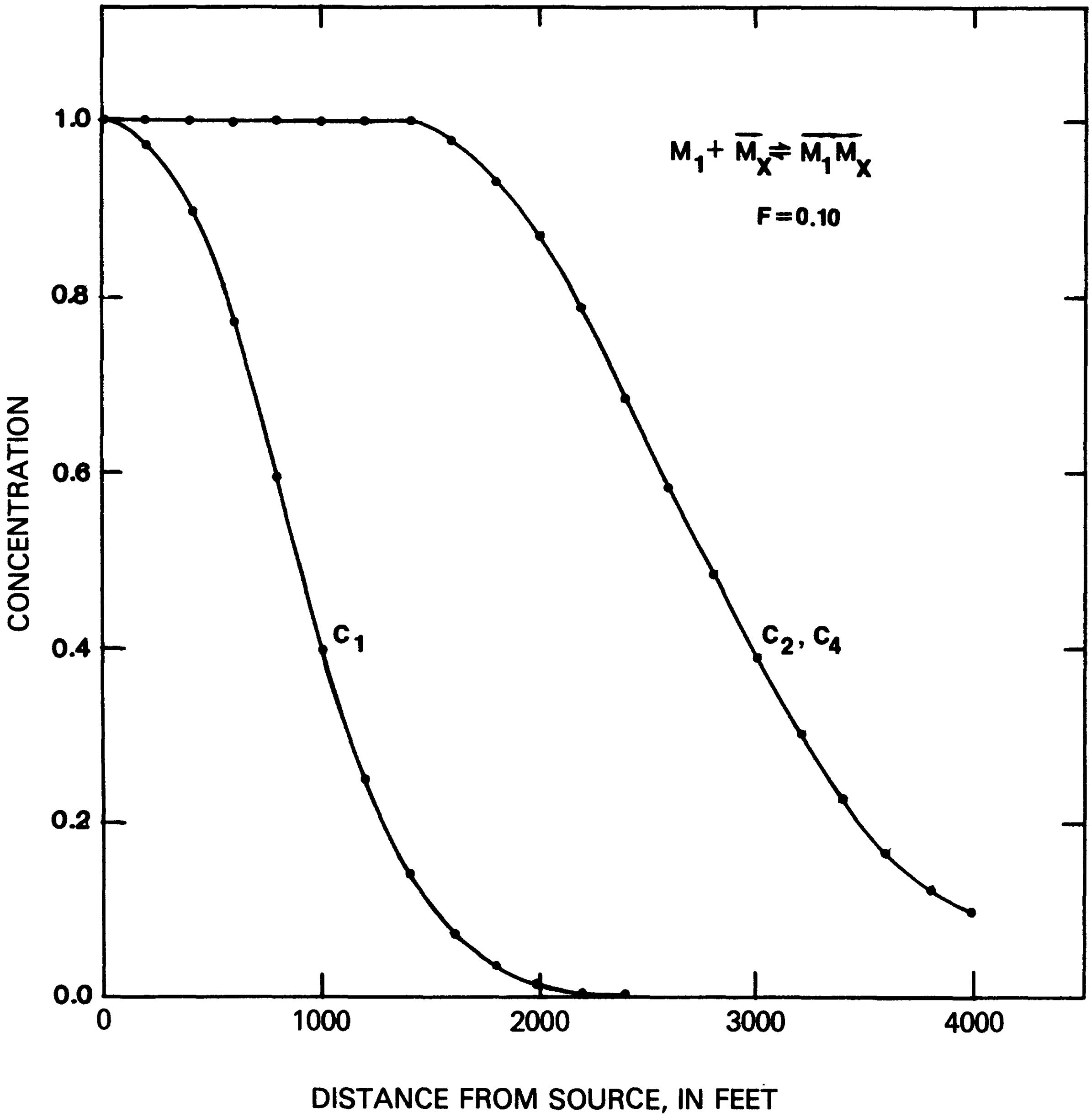

Figure 4. Single species sorption (R1, F = 0.1) and conservative transport (C_2, C_4) after 14 time steps (approximately 7 years).

The inflowing solution contains M_1, M_2 and M_4, each at concentration of 1.0. Because $k_{12} = k_{14} = 0.0$, however, M_2 and M_4 do not react, and are, therefore transported conservatively.

The exact solution for both the sorbing and conservative fronts can be determined by noting that this problem may be described by rewriting equation (11) in one dimension to read

$$R \frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2} - v_x \frac{\partial C}{\partial x} \qquad \text{on } 0 \le x \le L = 4000$$

subject to:

$$C(x,0) = 0$$

$$D_x \frac{\partial C}{\partial x} + v_x C|_{x=0} = v_x C_0 \qquad (C_0 = 1.0) \quad t>0$$

$$\frac{\partial C}{\partial t}(L,t) = \text{finite}$$

The analytical solution, as reported by van Genuchten and Alves (1982), is

$$C(x,t) = \tfrac{1}{2} \operatorname{erfc}\left[\frac{Rx - vt}{2(DRt)^{\frac{1}{2}}}\right] + \left(\frac{v^2 t}{\pi DR}\right)^{\frac{1}{2}} \exp\left[-\frac{(Rx - vt)^2}{4DRt}\right]$$

$$- \frac{1}{2}\left(1 + \frac{vx}{D} + \frac{v^2 t}{DR}\right) \exp(vx/D) \operatorname{erfc}\left[\frac{Rx + vt}{2(DRt)^{\frac{1}{2}}}\right] \qquad (94)$$

where R = 1.0 for conservative transport. Figure 4 contains the analytical solutions for both fronts plotted as solid lines, and the corresponding model results represented by points. Although errors in the results produced by the model are enhanced at the scale used in the figure, the results agree well with the exact solutions for both the non reactive and sorbing distributions.

Aqueous Complexation

Simulating transport influenced by equilibrium-controlled aqueous complexation (R2 and/or R3) provides an opportunity to observe the effect of local chemical equilibrium on reactive transport. The results of simulations involving these reactions are shown after eight time steps (approximately 4 years) in Figures 5 and 6. The chemical system represented in Figure 5 is defined only by a single reaction R2, with K_{12} arbitrarily set at 1.0. Inflow concentrations for the two reaction participants (C_1 and C_2) as well as nonreactive M_4 are 1.0. M_4 is designated nonreactive by assigning $K_{14} = 0.0$. The input of C_4 allows the distribution of a conservative solute to be illustrated simultaneously with reacting solutes.

An important trend that is unique to aqueous equilibrium-controlled reactive transport can be seen in Figure 5: the concentration front for the reactive species (C_1 and C_2) has higher values downstream than that of the conservative equivalent (C_4). Conceptually, a reactive species has these higher concentrations because at every point in the system it is in equilibrium with a dissolved compound that serves as an additional source for that species. Along the solute front, reduction in the concentration of the reactants causes the reaction (R2) to proceed to the left creating more of these reaction participants through dissociation. Without the reactive capability, specifically the presence of the associated complexed species in the inflow solution, a nonreactive species lacks this additional source and its concentration drops off more rapidly.

Mathematically, this situation arises because the actual transported quantities, in terms of the model, are the total dissolved tenad concentrations U, V and W. These distributions are also plotted in the figure.

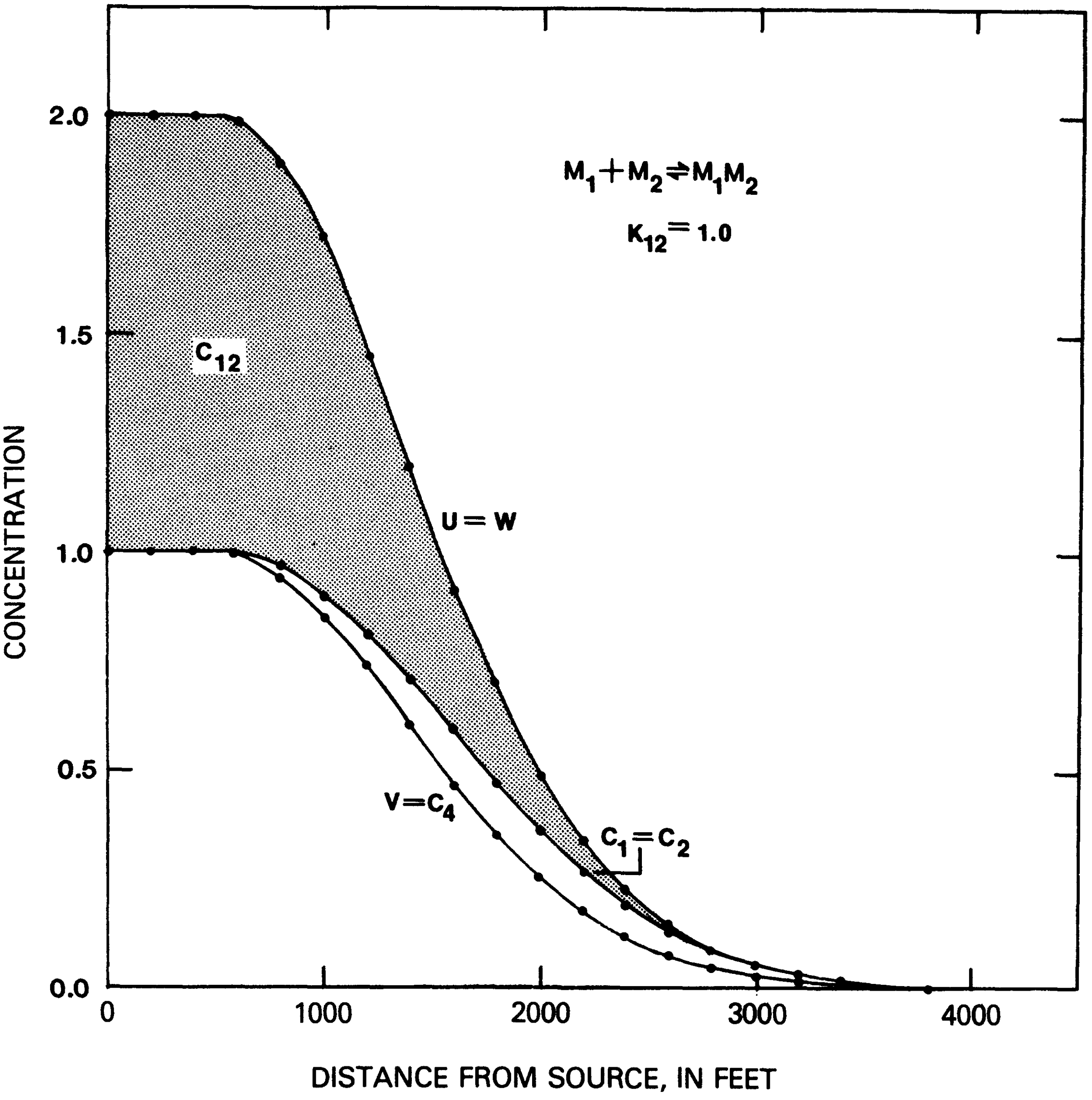

Figure 5. Transport with one aqueous equilibrium reaction (R2, K_{12} = 1.0) and a conservative solute (C_4) after 8 time steps (approximately 4 years).

When M_1 reacts with M_2 the concentration of the dissolved species M_1M_2 (C_{12}) is added to both U and W (equations (16) and (18)); but, without reaction R2, there is no C_{14} available to add to V. Consequently, U and W are greater than V at every point in the system. All three solute concentrations (U, V and W) represent entities that are transported conservatively because no sorption is involved and the total masses of $\{M_1\}$, $\{M_2\}$ and $\{M_4\}$ are reaction-independent. The individual concentrations C_1 and C_2, however, are a function of the reaction, and thus are not transported conservatively. These latter concentrations are determined through equation (83) which reduces to a quadratic equation when $K_{14} = 0$. This nonlinear relationship produces values of C_1 and C_2 that are constantly in equilibrium with C_{12} (shaded are in the figure) given the distributions of U and W.

Inasmuch as the distributions of U, V and W are conservative, their values can be verified through the same analytical solution (94) that was applied in the previous section. Additionally, from the boundary condition in this problem

$$U = W = 2V$$

consequently, the concentration front of $C_1 = C_2$ can also be checked against an exact solution. The only requirement is that W be known or obtained from either U or V. Once W is known, C_1, can be calculated the cubic equation (83) which reduces to the following quadratic when R2 is the only reaction:

$$K_{12}C_1^2 + [1 + K_{12}(W-U)]C_1 - W = 0 \tag{95}$$

In this case, $K_{12} = 1.0$, and $W = U = 1.0$; as a result, equation (95) reduces to

$$C_1^2 + C_1 - 1 = 0 \tag{96}$$

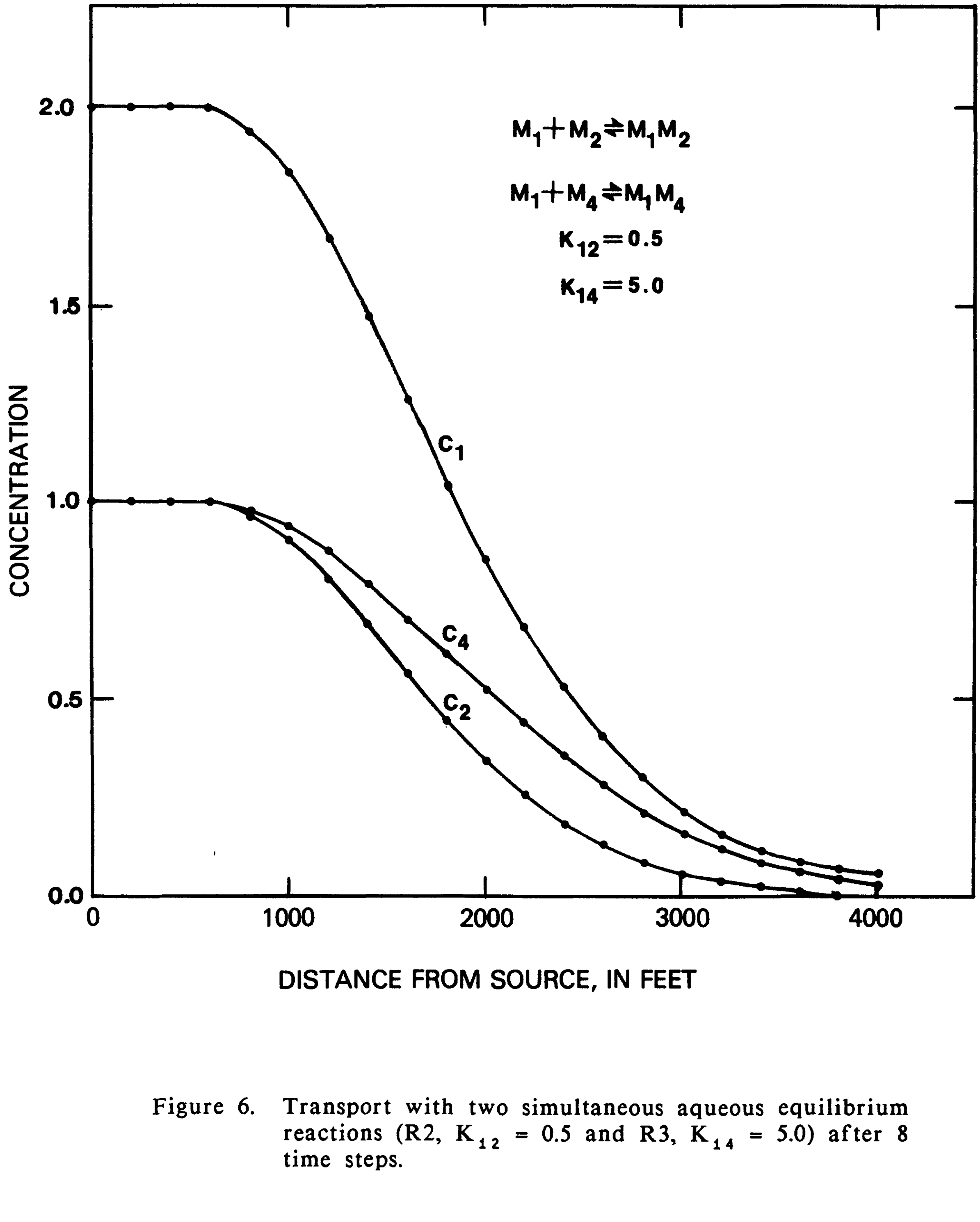

Figure 6. Transport with two simultaneous aqueous equilibrium reactions (R2, K_{12} = 0.5 and R3, K_{14} = 5.0) after 8 time steps.

Because the inflow concentrations of M_1 and M_2 are the same, $C_1=C_2$; consequently, equation (96) also applies to C_2 when substituted for C_1. In either case, (96) serves as a check that the model is obtaining correct values for the individual reactant concentrations at equilibrium.

Simulation results for a chemical transport system defined by both reactions R2 and R3 are shown in Figure 6. Here, the inflow concentrations of M_2 and M_4 are set at 1.0, while C_1 is input at twice this level to maintain an arbitrary electrostatic balance. The equilibrium constants are $K_{12} = 0.5$ and $K_{14} = 5.0$ for R2 and R3 respectively.

The figure shows another important trend resulting from the transport of equilibrium-controlled homogeneously reacting solutes: the participants in the reaction with the higher equilibrium constant has higher values downstream. This result follows from the previous discussion of Figure 5. The higher the equilibrium constant, the the greater the corresponding concentration of C_{ij} ($i=1$, $j=2,4$) for the equivalent input levels of C_i and C_j in both reactions. In this case, the equilibrium conditions imply that C_{14} is everywhere greater than, or equal to, C_{12}. As result, more C_4 is produced (relative to C_2) as both reactions (R2 and R3) proceed to the left along their respective fronts.

Sorption and One Complexation

The next example involves transport where linear sorption (R1) and a homogeneous aqueous reaction (R2) occur simultaneously. The interrelationships among the two reactions and the equilibrium conditions imposed on the system cause the sorption of M_1 to directly affect the resulting distribution of C_2. The nature of the effect depends on the sorption coefficient (F), the equilibrium constant (K_{12}) and the amount of elapsed time.

The individual and combined influence of these parameters will not be demonstrated here. For reference, the effects of varying F and K_{12} are discussed in Lewis (1984). In our simulation of transport with Rl and R2 the inflow concentrations, C_1, C_2 and C_4, are all 1.0. As in the example shown if Figure 5, the transport of C_4 is included in these simulations to represent a conservative solute for comparison.

The results for F = 0.25 and K_{12} = 1.0 are shown in Figure 7. The distribution of C_2 is characterized by a rise and fall in concentration that peaks at almost twice the inflow level. These results are best explained by referring to the reactions (Rl and R2). Because the system is at equilibrium, as M_1 is taken out of solution to form $\bar{M}_1$ the reaction R2 is forced to proceed toward the left. This causes M_1M_2 to dissociate producing both M_1 and M_2. With the sorption of M_1, the system maintains equilibrium through the increased amount of M_2 in solution.

The magnitude and extent of the peak level of C_2 are determined by the equilibrium condition that balances R2 according to the degree of sorption in Rl, and the transport mechanisms of advection and dispersion. These factors combine to cause the level of C_2 to peak at the point of maximum difference between the sorbed (C_1) and the equivalent conservative concentration (C_4). The maximum amount of M_2 produced is limited by the total dissolved $\{M_2\}$ concentration, U, in the source fluid. For reference, the conservative distribution of U is included in Figure 7. The concentrations C_2 and U become the same at the point where C_1 equals zero. Past this point, both reactions Rl and R2 stop because M_1 is no longer available to react. If M_1 has a high tendency to sorb, such that C_1 reaches zero at a point where the conservative solute equals its inflow concentration, then

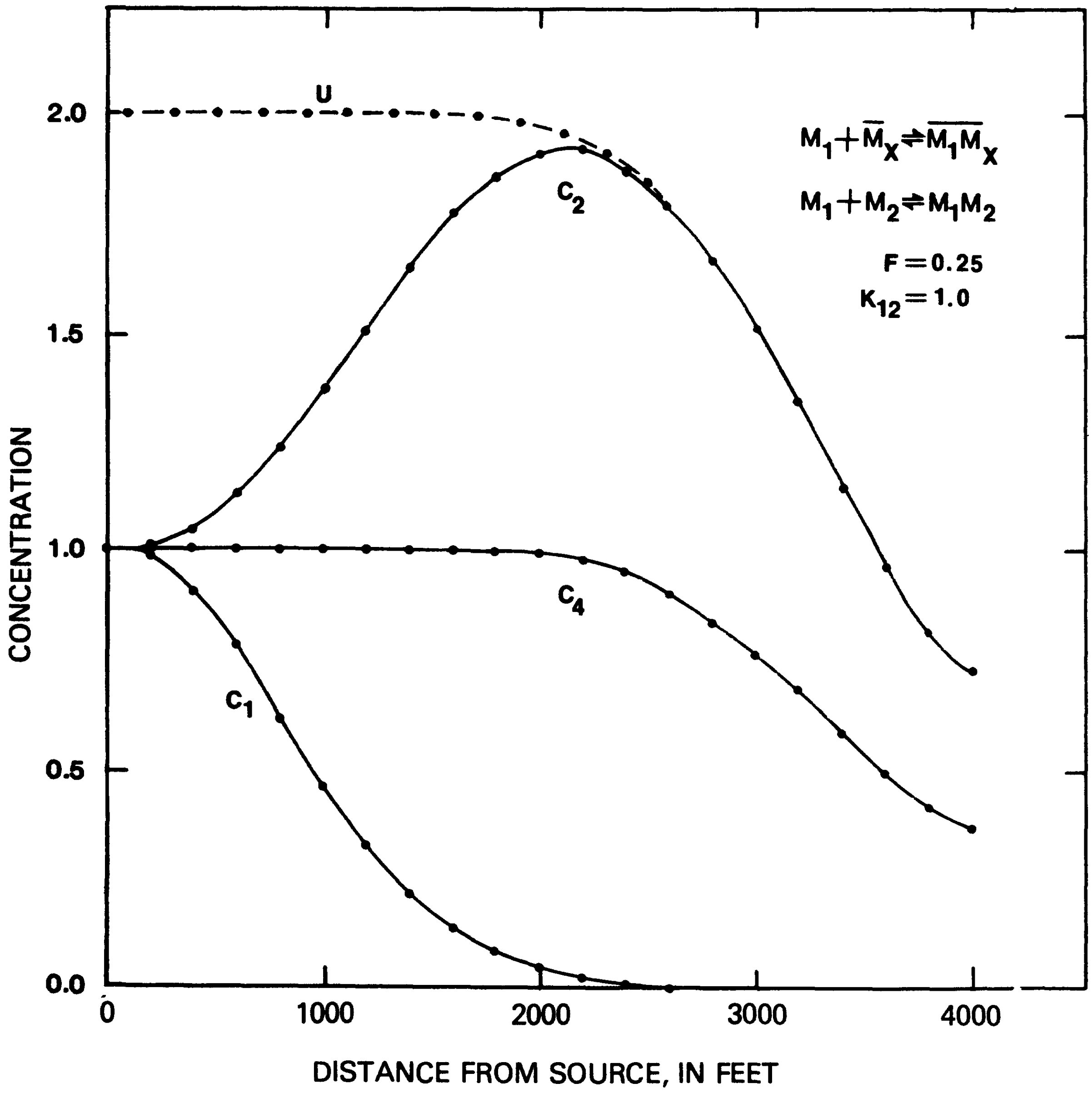

Figure 7. Transport with equilibrium sorption (R1, F = 0.25) and one aqueous complexation (R2, K_{12} = 1.0) after 18 time steps (approximately 9 years).

C_2 will equal U at the maximum possible level. If M_1 is not highly sorbed, however, then C_2 will equal U at some point along the latter's concentration front.

Sorption and Two Complexations

The next example involves transport with all three reactions (R1 through R3) governed by the following chemical parameters: $F = 0.25$; $K_{12} = 0.5$; and $K_{14} = 1.0$. The source fluid contains concentrations of $C_2 = C_4 = 1.0$; and $C_1 = 2.0$.

Figure 8 shows the results generated by the model after 18 time steps. The sorption of M_1 again directly affects the distributions of C_2 and, in this case, C_4 as well. Both concentrations increase above their inflow values, but differ as a function of their respective equilibrium constants. The distribution of C_4 has the higher peak because it is associated with the larger constant. As noted earlier, a higher K_{14} implies a greater ratio of product to reactant concentrations, and thus more M_1M_4 is available to dissociate and form M_1 and M_4. The sorption of M_1 simultaneously decreases C_{12} and C_{14} by equal amounts, but the greater equilibrium constraint imposed on R3 requires C_4 to increase more than C_2.

Sorption and Aqueous Complexation: An Example in Two Dimensions

Inasmuch as SATRA-CHEM can simulate transport in two dimensions, the trends discussed above may be demonstrated in this perspective as well. As an example, the chemical system involving reactions R1 through R3 depicted in Figure 8 ($K_{12} = 0.5$, $K_{14} = 1.0$, $F = 0.25$, with inflow concentraions of $C_1 = 2.0$, $C_2 = 1.0$ and $C_4 = 1.0$) is applied to a two-dimensional, areal, flow field with steady-state hydraulic heads. This requires a new

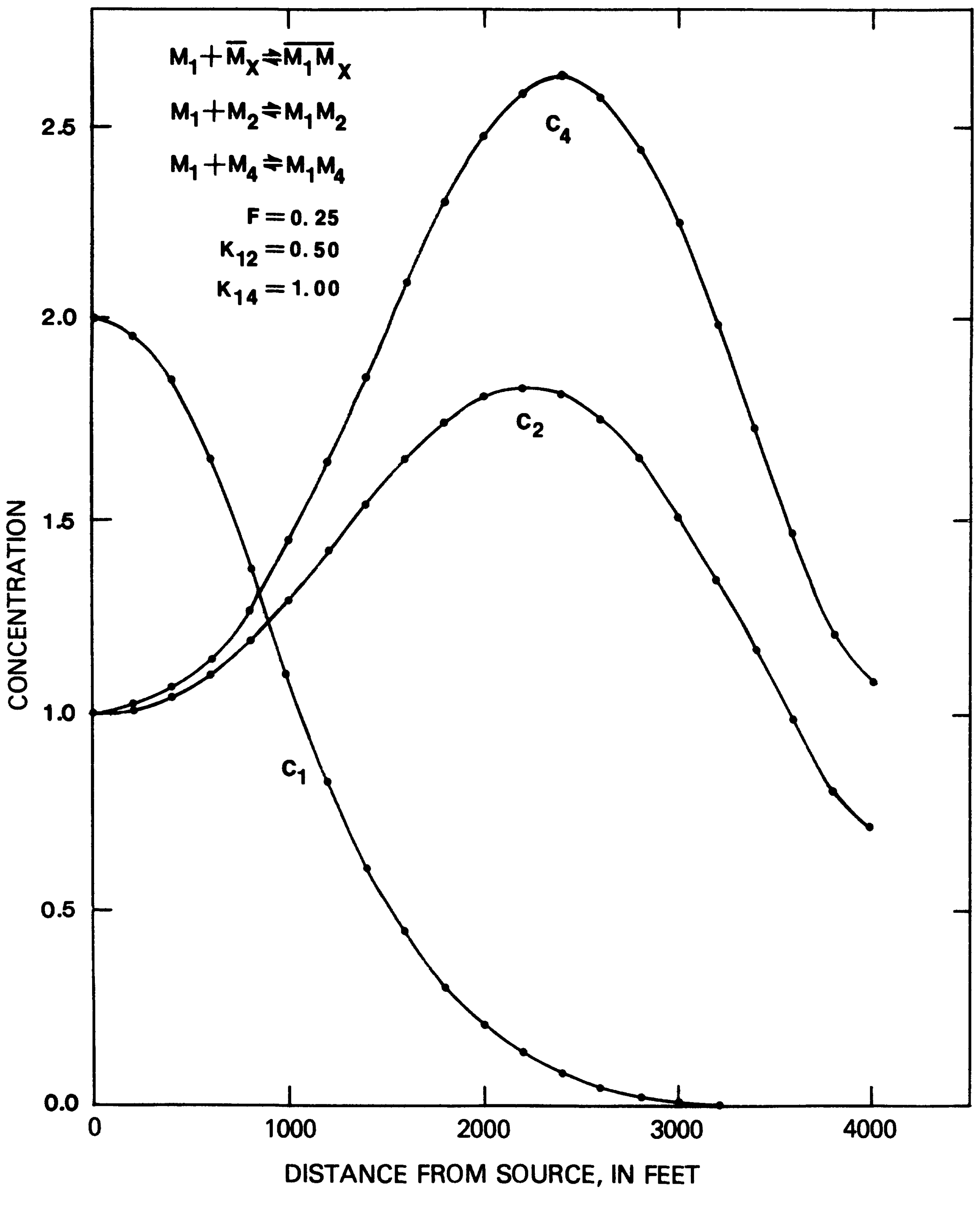

Figure 8. Transport with equilibrium sorption (R1, F = 0.25) and two aqueous complexations (R2, K_{12} = 0.5 and R3, K_{14} = 1.0) after 18 time steps.

finite-element grid and boundary conditions (Figure 9). Most of the physical parameters remain the same; only the values of dispersivity are new: longitudinal (α_L) = 30 feet, and transverse (α_T) = 5 feet.

In this case, flow is initiated at two adjacent nodes along the boundary through a specified volumetric flux. Both the flux and the source concentrations remain constant and continuous throughout the simulation. On the opposite boundary, three nodes have constant heads set at zero. The remainder of the boundary nodes constitute a no-flow boundary. The time-step interval is the same as in the runs above.

Figures 10 through 18 show in plan view the progression of all three reactant fronts through time. The results follow the anticipated pattern of concentration distribution based on the discussion of the system depicted in Figure 8.

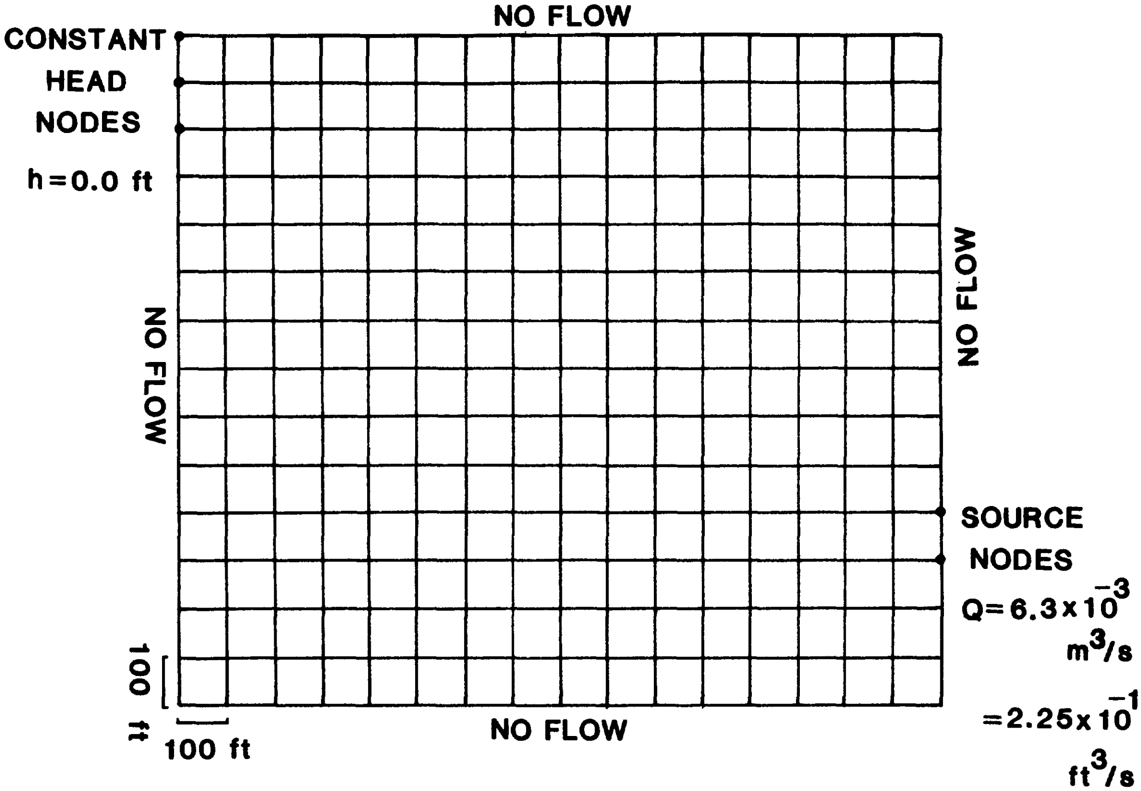

Figure 9. Finite-element grid and boundary conditions for two-dimensional transport simulation.

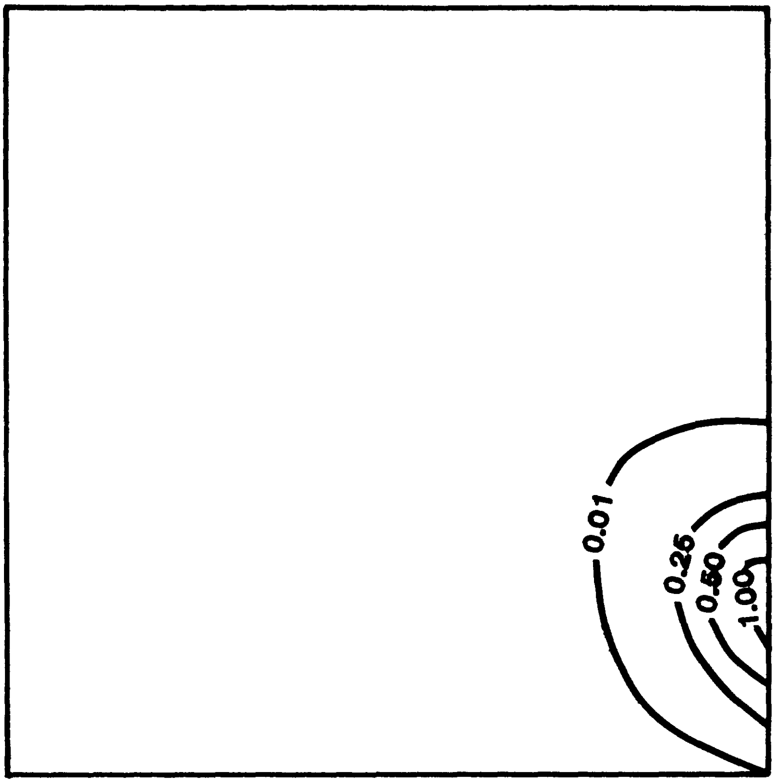

Figure 10. Two-dimensional distribution of C_1 after 1 time step (approximately 90 days). Transport contains sorption (R1, F = 0.25) and two aqueous complexations (R2, K_{12} = 0.5 and R3, K_{14} = 1.0).

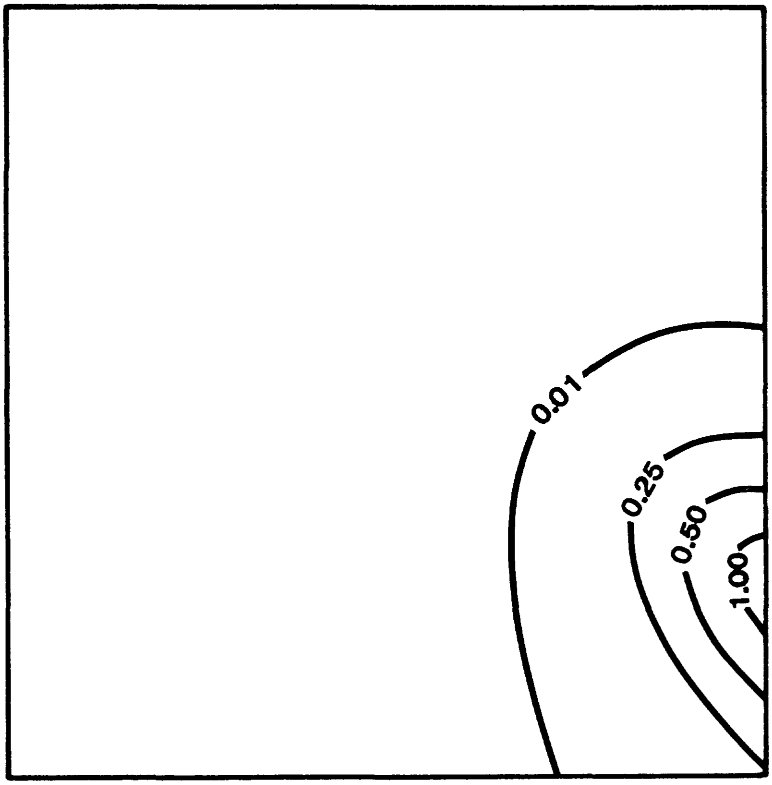

Figure 11. Two-dimensional distribution of C_2 after 1 time step.

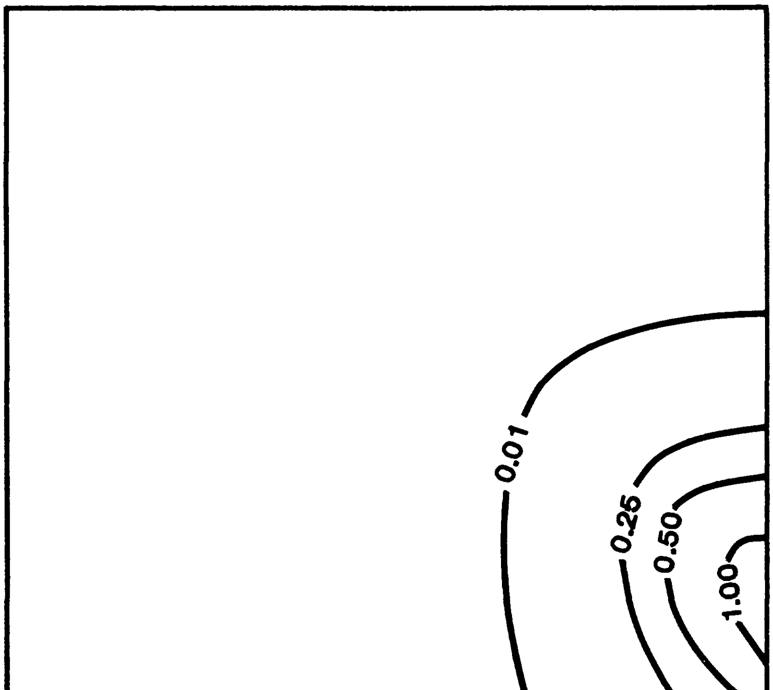

Figure 12. Two-dimensional distribution of C_4 after 1 time step.

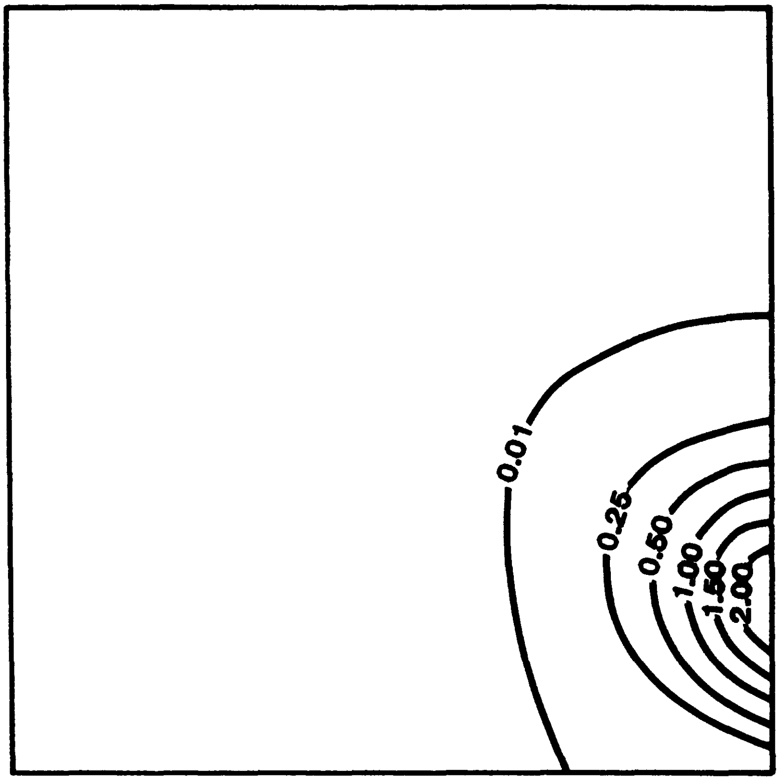

Figure 13. Two-dimensional distribution of C_1 after 5 time steps.

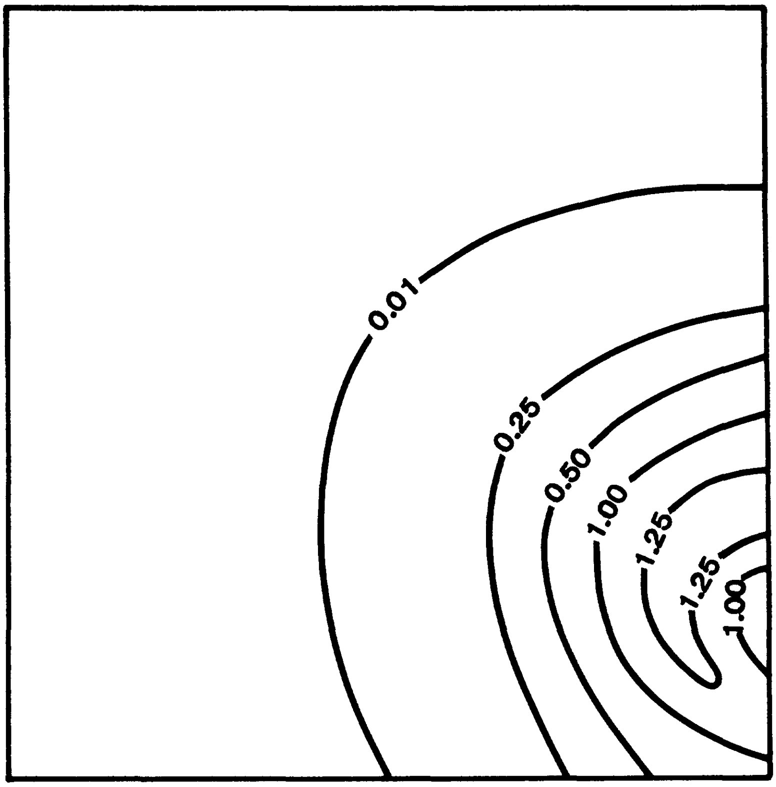

Figure 14. Two-dimensional distribution of C_2 after 5 time steps.

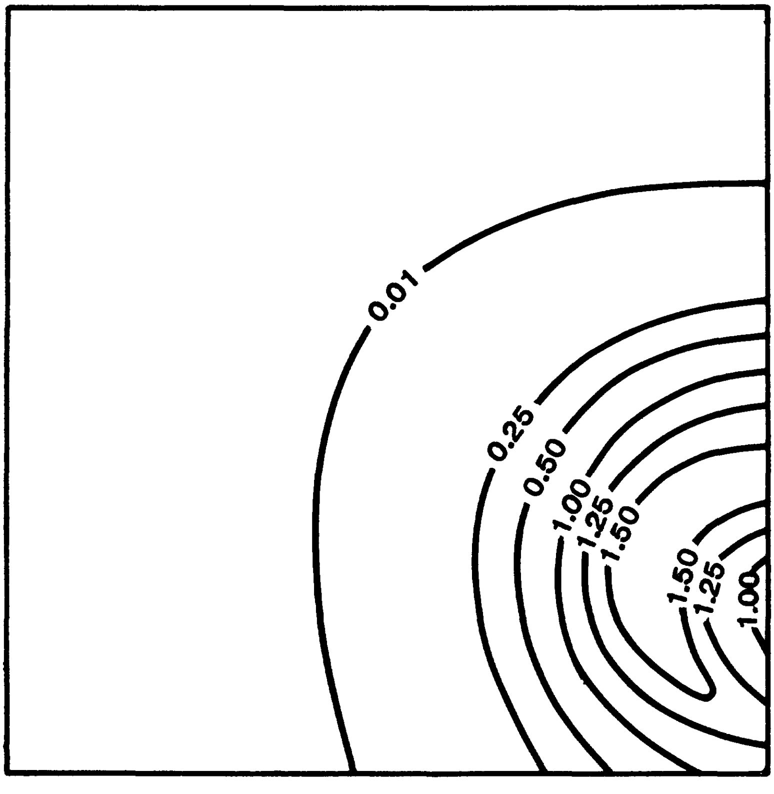

Figure 15. Two-dimensional distribution of C_4 after 5 time steps.

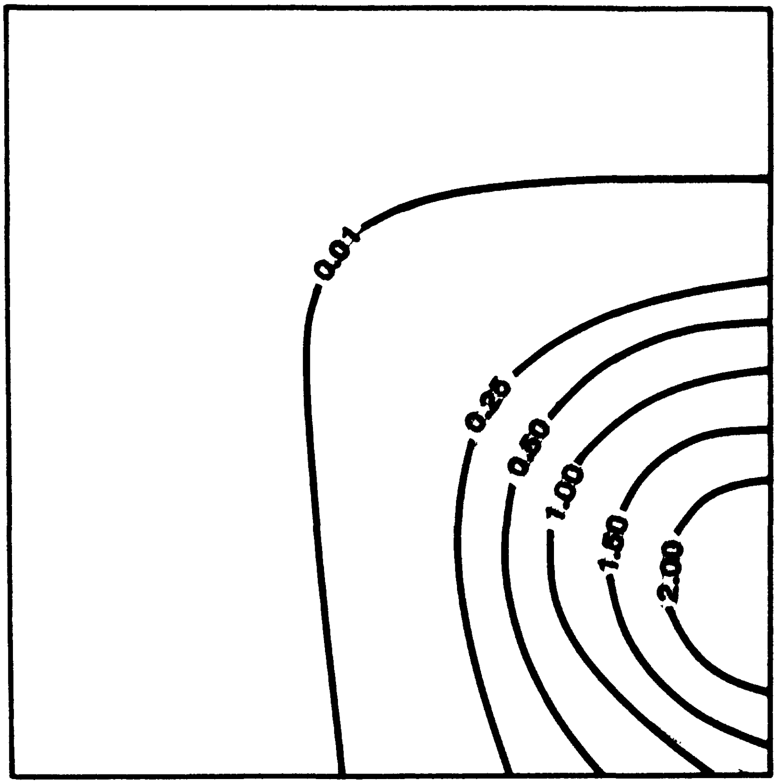

Figure 16. Two-dimensional distribution of C_1 after 20 time steps.

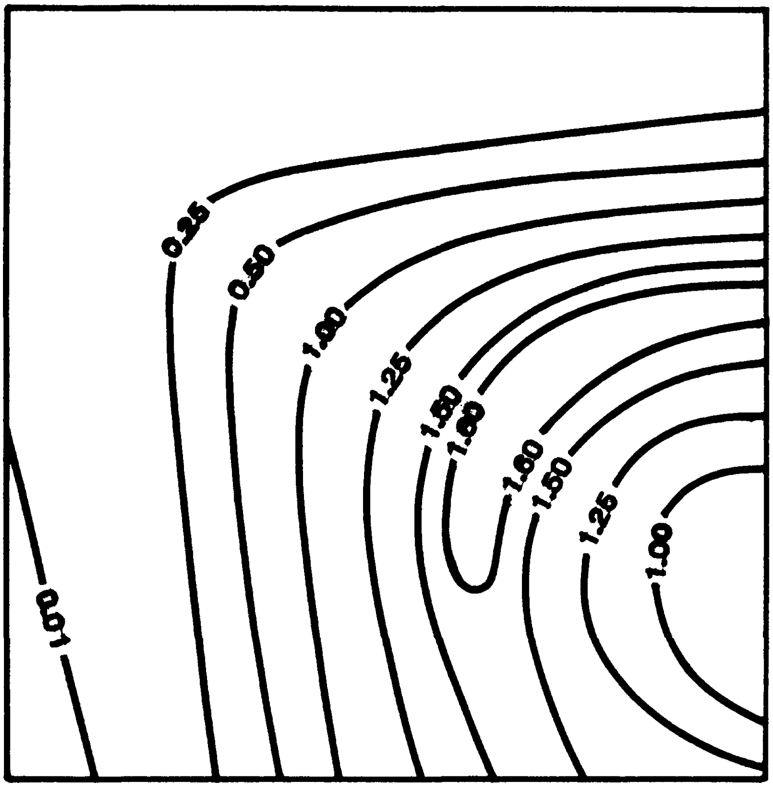

Figure 17. Two-dimensional distribution of C_2 after 20 time steps.

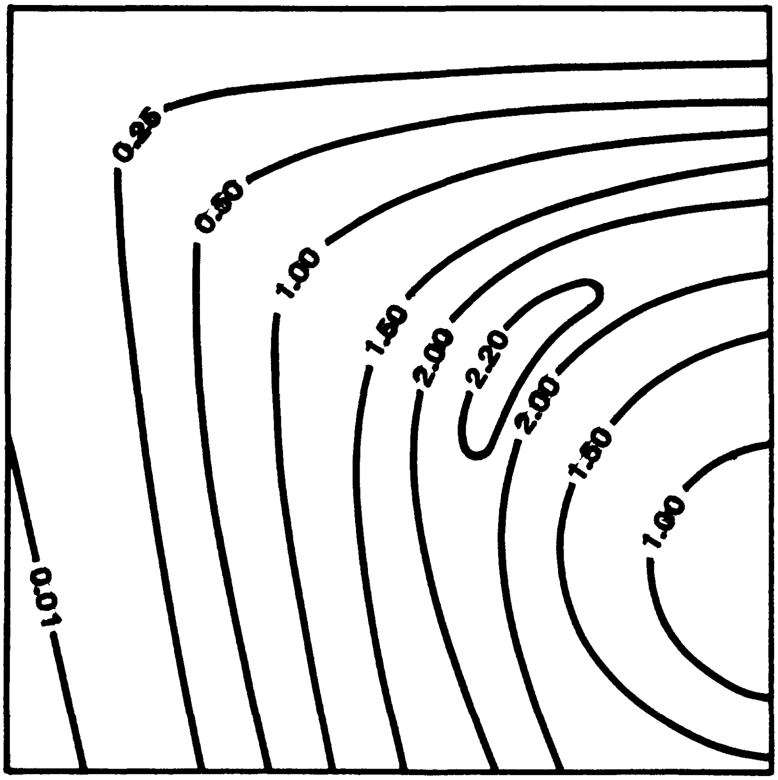

Figure 18. Two-dimensional distribution of C_4 after 20 time steps.

Ion Exchange and Aqueous Complexation

The performance of the model regarding transport with this reaction system is also demonstrated with a few examples. Simulation results are presented for a system in which the physical constraints and boundary conditions are the same as those utilized in the one-dimensional simulations discussed above. Only the reactions and initial conditions differ. Here, the porous medium is initially saturated with fluid containing the following reactant concentrations: $C_1 = 0.0$, $C_2 = 0.0$ and $C_3 = 1.0$. The electrostatic balance is maintained by the presence of an additional nonreactive species. In the source fluid, concentrations are the opposite of the background levels; specifically, $C_1 = 1.0$, $C_2 = 1.0$ and $C_3 = 0.0$.

Ion Exchange

Concentration distributions generated by the model for binary ion exchange (R3) without an accompanying homogeneous reaction are shown in Figure 19. The results are after 15 time steps (approximately 7.5 years) with $K_{13}=1.0$ and $\bar{C}_T=0.02$. The transport of C_2 is included to indicate the distribution of a conservative solute.

Inasmuch as C_1 is constant in the inflow solution and equal to the level of C_3 initially present in the porous media, we may assume that their sum is constant at every point and for all time (i.e., $C_T = C_1+C_3$). This relationship is valid because the total normality of the dissolved ionic species cannot be altered by the exchange process alone (Rubin and James, 1973). The results of the simulation (Figure 19) show that as C_1 increases, C_3 decreases equally such that their sum remains constant at 1.0. Note the addition of a simultaneous aqueous reaction to this chemical system will nullify the condition of constant C_T.

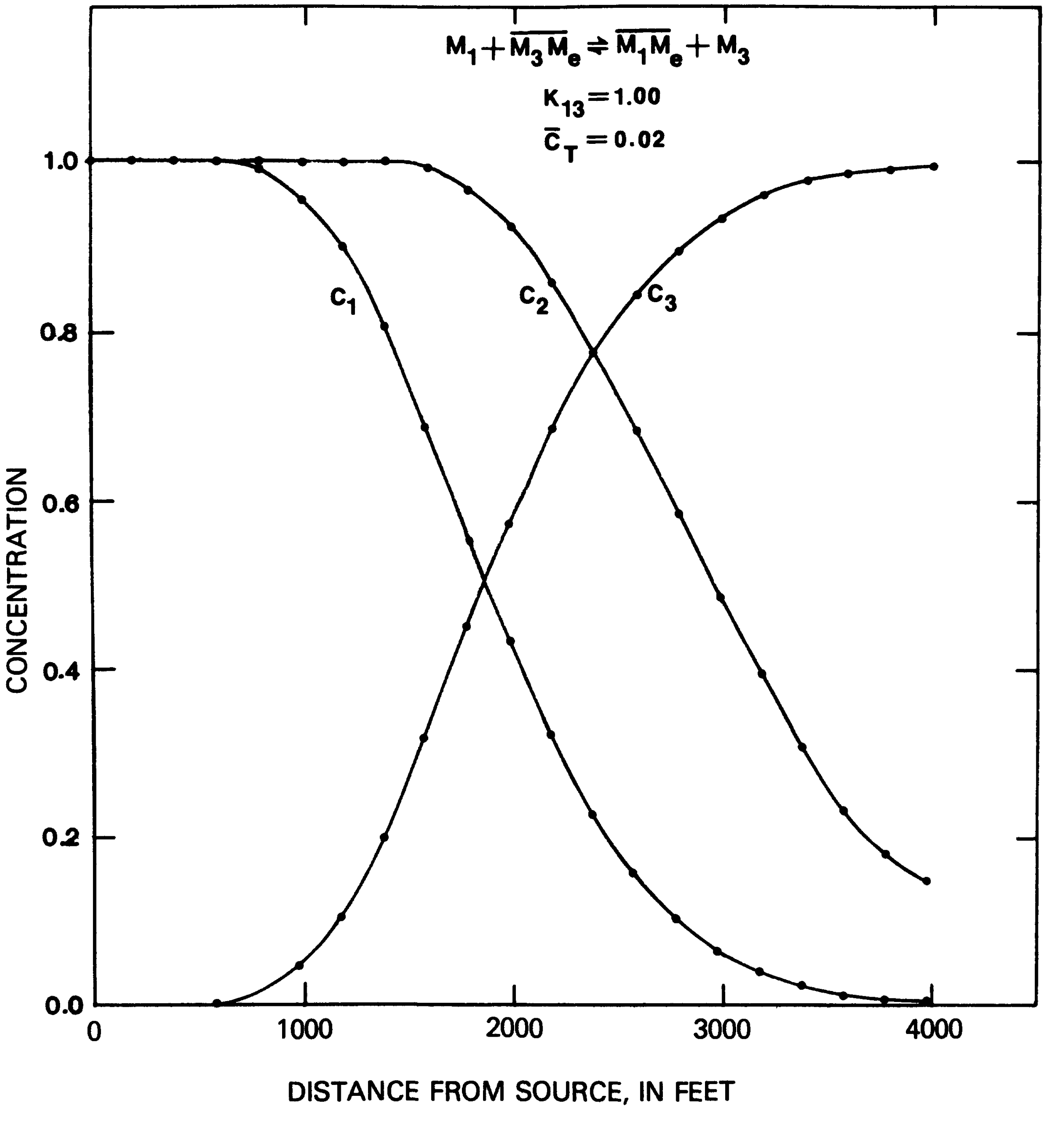

Figure 19. Transport with binary ion exchange (R4, K_{13} = 1.0) and a conservative solute (C_2) after 15 time steps (approximately 7.5 years); $\bar{C}_T$ = 0.02.

Apart from the Peclet number, the geometry of the exchanging C_1 front is a function of both the selectivity coefficient, K_{13}, and the exchange capacity of the media, $\bar{C}_T$. The effect of these two parameters on the spatial distribution of the exchanging front will not be discussed here but is demonstrated in Lewis [1984].

Ion Exchange and One Aqueous Complexation

Results produced by the model when both reactions R3 and R4 occur simultaneously are shown in Figure 20. The distributions represent solute concentrations after 11 time steps when $\bar{C}_T = 0.2$ and $K_{12} = K_{14} = 1.0$. As in the similar case of linear sorption, both C_2 and C_3 increase above their respective boundary conditions.

Figure 20 illustrates the cyclical pattern of sorption and dissociation typical in hybrid chemical systems as defined in this study: the sorption of M_1 through the exchange of M_3 in R3, leads to the production of both M_1 and M_2 through the dissociation of M_1M_2 in R4. Additional M_1 leads to an increase in the proportion of $\bar{C}_T$ comprised of $\bar{C}_1$; consequently, more M_3 is in solution than would be otherwise. Note the rise in C_3 above its background concentration is due to the total dissolved amount of $\{M_1\}$ in the system. In this case, the additional source of $\{M_1\}$ is the M_1M_2 complex. The peak in the M_2 concentration distribution remains at the point of maximum difference between the exchanging (sorbing) front and an equivalent conservative distribution. However, for both M_2 and M_3, the peak concentrations are governed by the equilibrium constraints and the exchange capacity of the media.

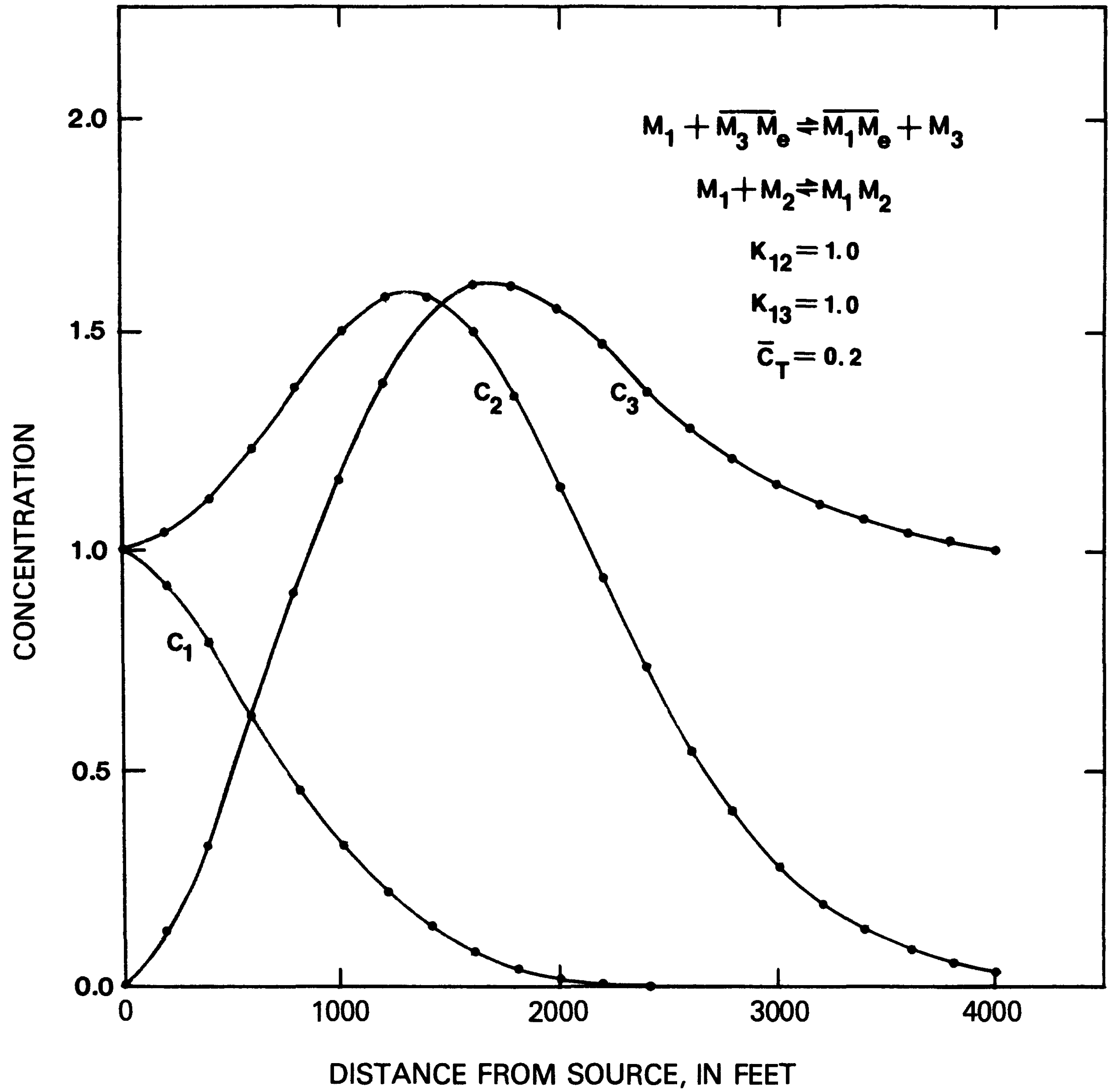

Figure 20. Transport with binary ion exchange (R4, K_{13} = 1.0) and one aqueous equilibrium reaction (R5, K_{12} = 1.0) after 11 time steps (approximately 5.5 years).

REQUIRED INPUT DATA

Inasmuch as the model must first solve for the hydraulic head at each node, data necessary to solve the ground-water flow equation must be supplied. This includes the components of hydraulic conductivity tensor for each element (i.e., the principal conductivities, together with the angle relating the principal directions to the actual coordinate system) as well as the appropriate initial and boundary conditions as they pertain to the head solution. Additionally, for transient flow, specific storage data are required on a nodewise basis.

The basic transport solution requires the input of longitudinal and transverse dispersivities for each element, and porosity values for each node. Moreover, prior to the simulation of either chemical system, up to four additional parameters are needed depending on the number of reactions involved. These parameters include: (1) either F, the linear sorption coefficient or $\bar{C}_T$, the exchange capacity of the media; (2) K_{12}, the equilibrium constant for reaction R2 (or R5); (3) either K_{13}, the selectivity coefficient for reaction R4 or K_{14}, the equilibrium constant for reaction R3; and (4), the solid-grain density (ρ_s).

The parameter F is commonly determined for specific soils through laboratory analyses, and is primarily a function of mineralogy, particle size, temperature, soil moisture, pH, and Eh (Fetter, 1980). Equilibrium constants are principally a function of temperature, and values for specific reactions may be found in selected chemical handbooks (e.g. Weast et al., 1983). For further reference, a complete listing of the input data requirements for SATRA-CHEM is contained in Appendix B.

REFERENCES

Bear, Jacob, 1979, Hydraulics of groundwater: McGraw-Hill, New York, 567 p.

Bredehoeft, John D., and George F. Pinder, 1973, Mass transport in flowing groundwater: Water Resources Research, v. 9, no. 1, p. 194-210.

Cederberg, Gail A., Robert Street, and James O. Leckie, 1985, A groundwater mass transport and equilibrium chemistry model for multi-component systems: Water Resources Research, v. 21, no. 8, p. 1095-1104.

Drever, James I., 1982, The geochemistry of natural waters: Prentice-Hall, Inc., Englewood Cliffs, N.J., 388 p.

Fetter, C. W., 1980, Applied hydrogeology: Merrill Pub. Co., Columbus, Oh., 488 p.

Freeze, Allan R., and John A. Cherry, 1979, Groundwater: Prentice-Hall, Englewood Cliffs, N.J., 604 p.

Grove, David B., and Warren W. Wood, 1979, Prediction and field verification of subsurface-water quality changes during artificial recharge, Lubbock, Texas: Groundwater, v. 17, no. 3, p. 250-257.

Gupta, Surendra P., and Robert Greenkorn, 1973, Dispersion during flow in porous media with bilinear adsorption: Water Resources Research, v. 9, no. 5, p. 1357-1368.

Holly, Donald E., and Paul R. Fenske, 1968, Transport of dissolved chemical contaminants in groundwater systems, in Nevada test site: Geological Society of America Memoir 110, E.B. Eckell Ed., p. 171-183.

Jennings, Aaron A., David J. Kirkner, and Thomas L. Theis, 1982, Multicomponent equilibrium chemistry in groundwater quality models: Water Resources Research, v. 18, no. 4, p. 1089-1096.

Jennings, Aaron A., and David J. Kirkner, 1984, Instantaneous equilibrium approximation analysis: J. Hydraulic Eng., v. 110, no. 12, p.1700-1717.

Konikow, Leonard F., and John D. Bredehoeft, 1974, Modeling flow and chemical quality changes in an irrigated stream-aquifer system: Water Resources Research, v. 10, no. 3, p. 546-562.

Lai, Sung-Ho, and J. J. Jurinak, 1972, Cation adsorption in one-dimensional flow through soils: a numerical solution: Water Resources Research, v. 8, no. 1, p. 99-107.

Lewis, Frank M., 1984, Sorption, ion exchange and equilibrium chemistry in advective-dispersive solute transport: Unpublished Masters Thesis, University of Arizona, 132 p.

Miller C. W., and L. V. Benson, 1983, Simulation of solute transport in a chemically reactive heterogeneous system: model development and application: Water Resources Research, v. 19, no. 2, p. 381-391.

Parkhurst, David L., Donald C. Thorstenson, and L. Niel Plummer, 1982, PHREEQE - A computer program for geochemical calculations: U.S. Geological Survey Water Resource Investigations 80-96, 210 p.

Pickens, John F., and William Lennox, 1976, Numerical simulation of waste movement in steady groundwater flow systems: Water Resources Research, v. 12, no. 2, p. 171-180.

Pinder, George F., 1973, A Galerkin-finite element simulation of groundwater contamination on Long Island, New York: Water Resources Research, v. 9, no. 6, p. 1657-1669.

Plummer, L. Niel, Blair F. Jones, and Alfred H. Truesdell, 1978, WATEQF - A FORTRAN IV version of WATEQ, a computer program for calculating Chemical equilibrium of natural waters: U.S. Geological Survey Water Resource Investigations 76-13, 63 p.

Reardon, E. J., 1981, Kd's - can they be used to describe reversible ion sorption reactions in contaminant migration: Groundwater, v. 19, no. 3, p. 279-286.

Rubin, Jacob, 1983, Transport of reacting solutes in porous media: relation between mathematicl nature of problem formulation and chemical nature of reactions: Water Resources Research, v. 19, no. 5, p. 1231-1252.

Rubin, Jacob and R. V. James, 1973, Dispersion-affected transport of reacting solutes in saturated porous media: Galerkin method applied to equilibrium-controlled exchange in unidirectional steady water flow: Water Resources Research, v. 9, no. 5, p. 1332-1356.

Valocchi, A. J., 1984, Validity of the local chemical equilibrium assumption for describing sorbing solute transport through homogeneous soils: EOS, v. 65, no. 16, p. 208.

Valocchi, Albert J., Robert L. Street, and Paul V. Roberts, 1981, Transport of ion-exchanging solutes in groundwater: chromatographic theory and field simulation: Water Resources Research, v. 17, no. 5, p. 1517-1527.

van Genuchten, M., and W. J. Alves, 1982, Analytical solutions of the one-dimensional convective-dispersive solute transport equation: USDA Technical Bulletin, no. 1661.

Voss, Clifford I., 1984, SUTRA: A finite-element simulation model for saturated-unsaturated, fluid-density-dependent ground-water flow with energy transport or chemically-reactive single-species solute transport: U.S. Geological Survey Water Resource Investigations Report, 84-4369.

Weast, Robert C., Melvin J. Astle, and William H. Beyer, 1983, CRC Handbook of chemistry and physics: Chemical Rubber Publishing Co.

APPENDIX A

NOTATION

A	A function of concentration; defined in equation 68.
B	A function of concentration; defined in equation 69.
C_1,C_2,C_3,C_4	Dissolved concentrations of arbitrary chemical species 1, 2, 3 and 4 in the aquifer $[ML^{-3}]$.
C^*_1,C^*_2,C^*_3,C^*_4	Dissolved concentrations of arbitrary chemical species 1, 2, 3 and 4 in the source fluid $[ML^{-3}]$.
C_{12},C_{14}	Dissolved concentrations of chemical compounds comprised of C_1, C_2 and C_4 in the aquifer $[ML^{-3}]$.
C^*_{12},C^*_{14}	Dissolved concentrations of chemical compounds comprised of C^*_1, C^*_2, and C^*_4 in the aquifer $[ML^{-3}]$.
$\bar{C}_1,\bar{C}_3$	Adsorbed concentrations of arbitrary chemical species 1 and 3 $[MM_s^{-1}]$.
C_T	Total dissolved concentration in the system $[ML^{-3}]$.
$\bar{C}_T$	Total exchange capacity of the medium $[MM_s^{-1}]$; defined in equation (44).
$\underline{\underline{D}}$	Hydrodynamic dispersion tensor $[L^2t^{-1}]$.
f	Adsorbate source in solute mass-balance relationships ships $[ML^{-3}t^{-1}]$; defined in equations (2) and (5).
f_1,f_2	Functions of concentration; defined in equations (61)-(62).
f_s	Solute-mass adsorption rate per unit mass of solid matrix (adsorbate flux) $[M_sM^{-1}t^{-1}]$; defined in equations (3)-(4).
F	Linear sorption coefficient which represents the ratio of solute mass sorbed per unit solid mass to solute mass per unit volume of water $[L^3M_s]$.
g	A function off concentration; defined in equaton (60).
G	A function of concentration; defined in equation (36).
h	Hydraulic head [L].
H	A function of concentration; defined in equation (35).
k	Time step indicator.

k_1, k_2 Coefficients in the nonlinear mass-balance equation that describe the nature of chemical interaction; defined specifically for equilibrium sorption and aqueous complexation in equations (40) and (41), and for binary ion exchange and aqueous complexation in equations (76)-(77) and (80)-(80).

$\underline{K}$ Hydraulic conductivity tensor $[Lt^{-1}]$.

K_{12} Chemical equilibrium constant for aqueous equilibrium reactions R2 and R5 $[L^3M^{-1}]$; defined in equations (12A) and (43).

K_{13} Chemical equilibrium constant (selectivity coefficient) for the binary ion exchange reaction R4 [1]; defined in equation (44).

K_{14} Chemical equilibrium constant for the aqueous equilibrium reaction R3 $[L^3M^{-1}]$; defined in equation (12B).

K_d Equilibrium distribution coefficient; same as the linear sorption coefficient, F, defined above.

L Length

L(C) Linear operator representing the fundamental advective-dispersive components with respect to concentration U; defined in equation (10).

M Mass of fluid or solute.

M_s Mass of solid.

M_1, M_2, M_3, M_4 Arbitrary chemical species 1 through 4 which are comprised of a single tenad and participate in reactions R1-R5.

M_1M_2, M_1M_4 Chemical compounds, defined in reactions R2, R3 and R5, which are comprised of M_1, M_2 and M_4

$\overline{M}_1$, $\overline{M}_3$ Sorbed masses of arbitrary chemical species 1 and 3 which are components of reactions R1 and R4.

$\overline{M}_e$ Arbitrary cation exchanger in reaction R4.

$\overline{M}_x$ Reactive surface of solid grain in reaction R1.

Q Volumetric fluid source (volume of fluid injected per time per volume of aquifer) $[t^{-1}]$.

R Retardation factor with respect to the average linear velocity of a sorbing solute.

Δt Time step increment.

U,V,W Functions of concentration in the aquifer defined in equations (16)-(18) and (53)-(55).

U^*,V^*,W^* Functions of concentration in the source fluid.

α_L Longitudinal dispersivity [L].

α_T Transverse dispersivity [L].

$\underline{\beta}$ Coefficient matrix that contains the advective contributions for the numerical discretization of the basic transport equations.

$\underline{\gamma}$ Coefficient matrix for the time derivative in the in the numerical discretization of the basic transport equations.

ε Porosity [1]

$\underline{\lambda}$ Coefficient matrix that contains the dispersive components in the numerical discretization of the basic transport equations.

ρ Fluid density $[ML^{-3}]$.

ρ_b Bulk density $[ML^{-3}]$.

ρ_s Solid-grain density $[M_s L^{-3}]$.

ϕ_j Spatial basis function for node j.

$\psi(x,y,t)$ Exact value of the dependent variable.

$\psi(x,y,t)$ Approximate value of the dependent variable.

$\underline{\omega}$ Vector of nodal source contributions including the nonlinear terms in k_2.

$\underline{\nabla}$ Two-dimensional vector differential operator $(\frac{\partial}{\partial x},\frac{\partial}{\partial y})$.

APPENDIX B

FORMATED INPUT DATA

This appendix contains the basic information for creating the formated data set necessary to run the model SATRA-CHEM version V05842D. The text below is modified from chapter, 7.7 SUTRA Input Data List found in Voss [1984].

The model generally requires two files of input data. The first, UNIT 5, contains both the aquifer and simulation control parameters, as well as the boundary conditions specified by the user for a given problem. The second file, UNIT 55, contains the initial conditions, and is only required for cold-starts of the model (i.e., for the first time step of the simulation). The option to restart (warm-start) is controlled by the parameters IREAD and ISTORE in DATASET 3 of the UNIT 5 file. ISTORE allows the final results of a simulation to be stored in a form directly usable as UNIT 55 data for future restarts.

List of Input Data for UNIT 5

DATASET 1: Input Data Heading (one card)

Variable	Format	Description
SIMULA	2A6	For areal transport simulation, write "SATRA AREAL SOLUTE TRANSPORT" beginning in the first column. For cross-sectional solute transport simulation, write "SATRA CROSS-SECTIONAL SOLUTE TRANSPORT" beginning in the first column.

DATASET 2: Output Heading (one card)

Variable	Format	Description
TITLE	80A1	First line of a heading for the input data set.

DATASET 3: Simulation Control Numbers (one card)

Variable	Format	Description
NN	I5	Exact number of nodes in finite element mesh.
NE	I5	Exact number of elements in finite element mesh.
NBI	I5	Full bandwidth of global banded matrix. NBI is equal to one plus twice maximum difference in node numbers in the element containing the largest node number difference in the mesh. This number is critical to computational efficiency, and should be minimized by careful numbering of the nodes. Setting NBI too small causes SATRA-CHEM to automatically print out the correct value and stop.
NPINCH	I5	Exact number of pinch nodes in the finite element mesh.
NHBC	I5	Exact number of nodes at which hydraulic head is a specified constant value or function of time.
NC1BC NC2BC NC4BC	I5 I5 I5	Exact number of nodes at which the concentrations (C_1, C_2, C_3 (or C_4) respectively, are. specified constant or a functon of time.
NSO	I5	Exact number of nodes at which a fluid source/ sink is a specified constant value or function of time.
ISSTAT	I5	Set to 0 for simulation with TRANSIENT ground-water flow. Set to +1 for simulation of STEADY-STATE solute transport.
IREAD	I5	To read initial condition data (UNIT 55) for cold start (first time step of a simulation), set to +1. To read initial condition data (UNIT 55) for simulation restart (data which has previously been stored by SATRA on UNIT 66), set to -1.
ISTORE	I5	To store results of final time step on UNIT 66 for later use as initial conditions on a restart, set to +1. To cancel storage of final time step, set to 0.

DATASET 4: Temporary Card (one card)

Variable	Format	Description
GNU	D15.0	Hydraulic head boundary condition 'leakance' factor. A high value causes SATRA simulated head and specified head values at specified head nodes to be equal in all significant figures. A low value causes smulated heads to deviate significantly from specified values. The ideal value of GNU causes simulated and specified heads to match in the largest four or five significant figures only, and deviate in the rest. GNU is entirely a numerical artifact which will eventually be eliminated from input data and calculated automatically. Currently, user trial-and-error is required to determine an ideal GNU value for a given simulation. An initial guess of 0.01 is suggested.

DATASET 5: Spatial Control Parameter (one card}

Variable	Format	Description

UP	F10.0	Fractional upstream weight for stabilization of oscillations in results due to highly advective transport or unsaturated flow. UP may be given any value from 0.0 to +1.0. UP = 0.0 implies no upstream weighting (Galerkin method).
		UP = 0.5 implies 50% upstream weighting.
		UP = 1.0 implies full (100%) upstream weighting.

Warning: upstream weighting increases the local effective longitudinal dispersivity of the simulation by approximately $(UP.(\Delta L)/2)$ where ΔL is the local distance between nodes along the direction of flow. Note that the amount of this increase varies from place to place depending on flow direction and element size.

In order to guarantee a spatially oscillation-free transport simulation, the mesh must be designed fine enough, and/or, the value, UP, must be great enough so that the following condition holds approximately along every streamline:
$\frac{1}{2}(\Delta L_{max}) \leq (UP \cdot (\Delta L)/2) + DSLFAC.DISPL(L)$ where

where ΔL_{max} is the largest distance between nodes along the streamline ΔL is the local distance between nodes along the streamline. The product of DSLFAC and DISPL(L), is the longitudinal dispersivity, specified in DATASET 13 and 14. In other words, the total simulated longitudinal dispersivity (given by the sum of numerical weighting dispersivity and physical dispersivity) must be greater than one-half the maximum element length measured along the flow direction.

DATASET 6: Temporal Control Data (one card)

Variable	Format	Description
ITMAX	I10	Maximum allowed number of time steps in simulation.
DELT	D10.0	Duration of initial time step. ⌊s⌋
TMAX	D10.0	Maximum allowed simulation time. ⌊s⌋ SATRA-CHEM time units are always in seconds. Other time measures are related as follows: ⌊min⌋ = 60. ⌊s⌋ ⌊h⌋ = 60. ⌊min⌋ ⌊d⌋ = 24. ⌊h⌋ ⌊week⌋ = 7. ⌊d⌋ ⌊mo⌋ = 30.4375 ⌊d⌋ ⌊yr⌋ = 365.250 ⌊d⌋
ITCYC	I10	Number of time steps in time step change cycle. A new time step size is begun at time steps numbered: 1+ n (ITCYC).
DTMULT	D10.0	Multiplier for time step change cycle. New time step size is: (DELT)(DTMULT).
NHCYC	I10	Number of time steps in head solution cycle. Hydraulic head is solve on time steps numbered: n(NPCYC), as well as on initial time step.
NUCYC	I10	Number of time step in concentration solution cycle. Transport equation is solved on time steps numbered: n(NUCYC) as well as on initial time step.
NPRINT	I10	Printed output is produced on time steps numbered: n(NPRINT), as well as on first and last time step.

DATASET 7: Printed Output Controls (one card)

Variable	Format	Description
KCOORD	I5	A value of -1 cancels printout of node coordinates, nodewise element thicknesses, and nodewise porosites and nodewise specific storativities. Set to 0 for full printout.
KELINF	I5	A value of -1 cancels printout of elementwise hydraulic conductivities and elementwise dispersivities. Set to 0 for full printout.
KINCID	I5	A value of -1 cancels printout of node incidences and pinch node incidences in elements. Set to 0 for full printout.
KPLOTH	I5	Set to a value of +1 for contourable printer plot of heads at all nodes in mesh. Set to -1 to cancel head plot.
KPLOTU	I5	Set to a value of +1 for contourable printer plot of concentrations or at all nodes in mesh. Set to -1 to cancel plot.
KVEL	I5	Set to a value of +1 to calculate and print fluid velocities at element centroids each time printed output is produced. Note that for non-steady state flow, velocities are based on results and heads of the previous time step and not on the newest values. A value of 0 cancels the option.

DATASET 8: Iteration Controls (one card)

Variable	Format	Description
ITRMAX	I10	Maximum number of iterations allowed per time step to resolve non-linearities. Set to a value of +1 for non-iterative solution.
RHMAX	D10.0	Absolute iteration convergence criterion for head solution. Head solution has converged when largest head change from the previous iteration's solution of any node in mesh is less then RHMAX. May be left blank for non-iterative solution.
RUMAX	D10.0	Absolute iteration convergence criterion for transport solution. Transport solution has converged when largest concentration change from the previous iteration's solution of any node in mesh is less than RUMAX. May be left blank for non-iterative solution.
RTOL	D10.0	Absolute iteration convergence criterion for Newton-Raphson iteration procedure.
MAXITR	D10.0	Maximum number of iterations for Newton-Raphson iteration procedure.

DATASET 9: Simulation Control Identifiers

Variable	Format	Description
NCONT	I2	Number of chemical constituents involved in the simulation (e.g., if C_1, C_2 and C_4 are transported, NCONT = 3).
ISORB	I5	Adsorption indicator. A value of +1 implies the simulation involves linear sorption (i.e., FF>0 in DATASET 10). A value of 0 implies no exchange occurs.
IEXCH	I5	Ion exchange indicator. A value of +1 implies the simulation involves binary ion exchange. A value of 0 implies no exchange occurs.
IEQLIB	I5	Equilibrium chemistry indicator. A value of +1 implies the simulation involves at least one aqueous equilibrium chemical reaction. A value of 0 implies no aqueous reactions take place. If the simulaiton involves just sorption or just ion exchange, IEQLIB = 0 even though the reactions are technically in equilibrium.

DATASET 10: Fluid, Solid Matrix and Solute Properties (one card)

Variable	Format	Description
SIGMAW	D10.0	Fluid diffusivity, σ_w. Represents molecular diffusivity of solute in pure fluid. $[L^2 /s]$.
RHOS	D10.0	Density of a solid grain, ρ_s, for sorption calculations. $[M/L^3]$. (Need be specified only if ISORB≠0)
DECAY	D10.0	Decay rate (first order) of solute, γ $[M_s/s]$. $(\partial C/\partial t) = -\gamma C$. Set to zero for no decay.
FF	D10.0	Linear sorption constant, F. Relates relative concentrations between solid and liquid phase for a particular chemical constituent. Set to positive value when ISORB>0 (DATASET 9).
CBART	D10.0	Total exchange capacity of the porous media, $\bar{C}_T$. Set to positive value when IEXCH>0 (DATASET 9).
EQCSTU	D10.0	Chemical equilibrium constant for reactions R2A and R4. If either of these reactions is simulated, a positive value is required here and for IEQLIB (DATASET 9).
EQUCSTV	D10.0	Chemical equilibrium constant for reactions R2B and R3. If either of these reactions is simulated, a positive value is required here and for IEQLIB (DATASET 9).

DATASET 11: Scale Factor for Nodewise Data (one card)

Variable	Format	Description
SCALX	D10.0	The scaled x-coordinates of nodes in DATASET 12 are multiplied by SCALX in SATRA-CHEM. May be used to change from map to field scales, or from English to SI units. A value of 1.0 gives no scaling.
SCALY	D10.0	The scaled y-coordinates of nodes in DATASET 12 are multiplied by SCALY in SATRA-CHEM. May be used to change from map to field scales, or from English to SI units. A value of 1.0 gives no scaling.
SCALTH	D10.0	The scaled element (mesh) thicknesses at nodes in DATASET 12 are multiplied by SCALTH in SATRA-CHEM. May be used to easily change entire mesh thickness or to convert English to SI units. A value of 1.0 gives no scaling.
PORFAC	D10.0	The scaled nodewise porosities of DATASET 12 are multiplied by PORFAC in SATRA-CHEM. May be used to easily assign a constant porosity value to all nodes by setting PORFAC = porosity and all POR(II) = 1.0 in DATASET 12.
STOFAC	F10.0	The scaled nodewise specific storativities of DATASET 12 are multiplied by STOFAC in SATRA-CHEM. May be used to easily assign a constant storativity value to all nodes.

DATASET 12: Nodewise Data (one card for each of NN nodes)

Variable	Format	Description
II	I5	Number of node to which data on this card refers, i.
X(II)	DIO.0	Scaled x-coordinate of node II, x_i. [L]
Y(II)	D10.0	Scaled y-coordinate of node II, y_i. [L]
THICK(II)	D10.0	Scaled thickness of mesh at node II. [L] In order to simulate radial cross-sections, set THICK(II) = $(2\pi)(\text{radius}_i)$, where radius_i is the radial distance from the vertical center axis to node II.
POR(II)	D10.0	Scaled porosity value at node II, ε_i. [1]
STOR(II)	D10.0	Scaled specific storativity value at node II, S_o, $[L^{-1}]$.

DATASET 13: Scale Factors for Elementwise Data (one card)

Variable	Format	Description
CMAXFA	D10.0	The scaled maximum hydraulic conductivity values of elements in DATASET 14 are multiplied by CMAXFA in SATRA-CHEM. May be used to convert units or to aid in assignment of maximum conductivity values in elements.
CMINFA	D10.0	The scaled minimum hydraulic conductivity values of elements in DATASET 14 are multiplied by CMINFA in SATRA-CHEM. May be used to convert units or to aid in assignment of minimum conductivity values in elements.
ANGFAC	F10.0	The scaled angles between maximum conductivity direction and x-axis of elements in DATASET 14 are multiplied by ANGFAC in SATRA-CHEM. May be used to easily assign a uniform direction of anisotropy by setting ANGFAC = angle, and all ANGLEX(L) = 1.0 in DATASET 14.
DSLFAC	D10.0	The scaled longitudinal dispersivities of elements in DATASET 14 are multiplied by DSLFAC in SATRA-CHEM. May be used to convert units or to aid in assignment of dispersivities.
DSTFAC	D10.0	The scaled transverse dispersivities of elements in DATASET 14 are multiplied by DSTFAC in SATRA-CHEM. May be used to convert units or to aid in assignment of dispersivity.

DATASET 14: Elementwise Data (one card for each of NE elements).

Variable	Format	Description
LN	I10	Number of element to which data on this card refers.
CMAX(LN)	D10.0	Scaled maximum hydraulic conductivity value of element LN, $K_{max}(LN)$ [L/s].
CMIN(LN)	D10.0	Scaled minimum hydraulic conductivity value of element LN, $K_{min}(LN)$. [L/s]. Isotropic conductivity requires: CMIN(LN) = CMAX(LN).
ANGLEX(LN)	F10.0	Angle measured in counterclockwise direction from +x-direction to maximum conductivity direction in element LN, θLN. [°] Arbitrary when CMIN(LN) = CMAX(LN).
DISPL(LN)	F10.0	Scaled longitudinal dispersivity value of element LN, $\alpha_L(LN)$ [L].
DISPT(LN)	FIO.0	Scaled transverse dispersivity value of element LN, $\alpha_T(L)$ [L].

DATASET 15: Optional Data for Printer Plot
(Two or three cards when plot has been requested by DATASET 7)

Card 1: (always required when plot is requested)

Variable	Format	Description
IDIREC	I5	Chooses plot direction: Set to -1 for small plot which fits across the output page. Set to +1 for larger plot which is oriented along the output page.
NLINPI	I5	Number of printer lines per inch.
NCHAPI	I5	Number of printer characters per inch.
NCHAPL	I5	Number of printer characters per output line.

Card 2. (include this card only when pressure plots are requested in DATASET 7)

Variable	Format	Description
HBASE	D13.0	Value for scaling plotted hydraulic heads. (See below.)
HDIGIT	D13.0	Digit output control for head

*The plotting routine prints three digits of the nodal value to be plotted at the (x,y) location of the node on a map of the mesh which the routine constructs. The three digits are not necessarily the first three digits of the value to be plotted, but are always one digit to the left and two digits to the right of the decimal point.

The head value to be plotted, H_{PLOT}, is

calculated by SATRA-CHEM as
H_{PLOT} = (true head h_i/HBASE) (HDIGIT)

For example, HBASE may be used to scale out powers of ten and HDIGIT to shift the scaled digits of interest to the position of the three plotted digits.

Card 3: (include this card only when or concentration plots are requested in DATASET 7)

C1BASE D13.0 Value for scaling plotted concentration C_1 values. (See below.)

C1DGIT D13.0 Digit output control for or concentration. For explanation see (*) above.

The value to be plotted, C_{PLOT}, is calculated by SATRA-CHEM as:
C_{PLOT}= (concentration C_i/C2BASE)(C2DGIT)

For example, CBASE may be set to the highest source concentration in the system, and CDIGIT to a value of ten; then fractional concentrations relative to the highest concentration are plotted with digits ranging from 000 to 999 which represents a relative concentration of 1.000 (~0.999).

C2BASE D13.0 Same principle as above scaling directions.
C2DGIT D13.0
C4BASE D13.0
C4DGIT D13.0

DATASET 16: Optional. Data for Fluid Sources and Sinks
(one card for each of NSO source nodes as specified
in DATASET 3)

Variable	Format	Description
IQC	I5	Number of node to which source/sink data on this card refers. Specifying the node number with a negative sign indicates to SATRA that the source flow rate or concentration or temperature vary in a specified manner with time. All information regarding a time-dependent source node must be programmed by the user in Subroutine BCTIME, and no other data is included on this card.
QINC	D9.0	Fluid source (or sink) which is a specified constant value at node IQC. $[L^3/s]$ A positive value is a source of fluid to the aquifer. Leave blank if IQC is negative.
C1INC	D9.0	Solute concentration of fluid entering the aquifer which is a specified constant value for a fluid source at node IOC. $[M_s/L^3]$ Leave blank if either IQC or QINC is negative.
C2INC	D9.0	
C2INC	D9.0	

DATASET 17: Optional. Data for Specified Hydraulic Head Nodes (one card for each of NHBC specified head nodes as indicated in DATASET 3, plus one blank card)

Variable	Format	Description
Cards 1 to NHBC:		
IHBC	I6	Number of node to which specified head data on this card refers. Specifying the node number with a negative sign indicates to SATRA that the specified hydraulic head value or flow concentration at this node vary in a specified manner with time. All information regarding a time-dependent specified pressure node must be programmed by the user in Subroutine BCTIME, and no other data is included on this card.
HBC	D13.0	Hydraulic head value which is a specified constant at node IHBC [L]. Leave blank if IHBC is negative.
C1BC C2BC C4BC	D13.0 D13.0 D13.0	Solute concentration of any external fluid which enters the aquifer at node IHBC. UBC is a specified constant value [M_s/L^3]. Leave blank if IHBC is negative.

Note, the last card should be blank and placed immediately following all NHBC specified head cards.

DATASET 18: Optional. Data for Specified Concentration Nodes (one card for each of NUBC nodes specified concentration nodes as indicated in DATASET 3, plus one blank card)

Variable	Format	Description
Cards 1 to NUBC.		
IC1BC	I6	Number of node to which specified concentration data on this card refers. Specifying the node number with a negative sign indicates to SATRA that the specified value at this node varies in a specified manner with time. This time-dependence must be programmed by the user in Subroutine BCTIME, and no other data is included on this card.
IC2BC	I6	
IC4BC	I6	
C1BC	D13.0	Concentration value in aquifer which is a specified constant at node IUBC. $[M_s/L^3]$ Leave blank if IUBC is negative.
C2BC	D13.0	
C4BC	D13.0	

Note, the last card should be blank and placed immediately following the specified concentration cards.

DATASET 19: Element Incidence and Pinch Node Data
(one or two cards for each of NE elements)

Variable	Format	Description	
Card A (required for each element)			
LL	I6	Number of element to which data on this card (and the optional next card) refers. If pinch nodes exist in element LL, then the element number must be specified with a minus sign.	
Node Incidence List			
IIN(1)	I6	Number of node 1	List of corner node numbers in element LL, beginning at any node, but taken in an order counterclockwise about the element.
IIN(2)	I6	Number of node 2	
IIN(3)	I6	Number of node 3	
IIN(4)	I6	Number of node 4	

Card B (required immediately following Card A only when LL is negative)

Pinch-Node Incidence List

IEDGE(1)	I6	Node number of	IIN(1) and IIN(2)
IEDGE{2)	I6	pinch node at	IIN(2) and IIN(3}
IEDGE(3)	I6	mid-point of	IIN(3) and IIN(4)
IEDGE(4)	I6	edge between nodes:	IIN(4) and IIN(1)

A blank in the list of pinch node numbers indicates that that no node exists on that particular edge element LL.

End of Input Data List for UNIT 5

List of Input Data for UNIT 55

DATASET 1: Simulation Starting Time (One card)

Variable	Format	Description
TSTART	D20.0	Elapsed time at which the inital conditions for simulation specified in UNIT 55 are given. [s] This sets the simulation clock starting time. Usually set to a value of zero for cold-start.

DATASET 2: Initial Hydraulic Head Values at Nodes

Requires (NN + 3)/4 cards. (Done by integer arithmetic.)

Variable	Format	Description
HVEC(II)	6D13.0	Initial (starting) head values at time, TSTART, at each of NN nodes [L]. Four values per card, in exact order of node numbers. These values are arbitrary and may be left blank if the steady-state flow option in DATASET 3 of UNIT 5 has been chosen. However, the data set is still required. An initial natural head distribution for given hydrologic conditions may be obtained by running a single steady-flow time step with the store option. Then the natural heads just calculated and stored on Unit 55 file without change in format, as initial conditions for a transient run.

DATASET 3: Initial Concentration Values at Nodes

Requires (NN+3)/4 cards. (Done by integer arithmetic.)

Variable	Format	Description
C_1(II)	4D20.0	Initial (or starting) solute concentration values at time, TSTART, at each of NN nodes [M_s/L^3]. Four values per card, in exact order of node numbers.
C_2(II)	4D20.0	Initial (or at time, TSTART) solute concentration, C_2, at each of NN nodes. Four values per card in exact order of node numbers.
C_4(II)	4D20.0	Initial (or at time, TSTART) solute concentration, C_4 (or C_3), at each of NN nodes. Four values per card in exact order of node numbers.

End of Input Data List for UNIT 55

APPENDIX C

LISTING OF COMPUTER CODE

```
C   SATRA-CHEM        M  A  I  N     P  R  O G R  A  M     VERSION 0684 AA10.....
                                                                        AA20.....
C                  UNITED STATES GEOLOGICAL SURVEY                      AA30.....
C      TWO-DIMENSIONAL GROUNDWATER FLOW AND MULTISOLUTE TRANSPORT       AA40.....
C SIMULATION MODEL WITH SORPTION, ION EXCHANGE AND EQUILIBRIUM CHEMISTRYAA50.....
                                                                        AA60.....
                                                                        AA70.....
C              --------------------------------------------             AA80.....
C              |                                          |             AA90.....
C              |   S    A    T    R    A   -   C    H    E    M   |     AA100....
C              |                                          |             AA110....
C              --------------------------------------------             AA120....
                                                                        AA130....
                                                                        AA140....
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  AA150....
C    * ->Constant-density ground-water flow and multisolute           *  AA160....
C    *    transport                                                   *  AA170....
C    * ->Options for transport with two different chemical systems:   *  AA180....
C    *    (a) linear sorption and up to two aqueous complexations     *  AA190....
C    *    (b) binary ion exchange and one aqueous complex             *  AA200....
C    * ->Two-dimensional areal or cross-sectional simulation          *  AA210....
C    * ->Either cartesian or radial/cylindrical coordinates           *  AA220....
C    * ->Hybrid Galerkin finite-element method and integrated         *  AA230....
C    *    finite difference method with two-dimensional               *  AA240....
C    *    quadrilateral finite elements                               *  AA250....
C    * ->Finite difference time discretization                        *  AA260....
C    * ->Transient or steady-state solution options                   *  AA270....
C    * ->Optional fluid velocity calculation                          *  AA280....
C    * ->Optional printer plots of output                             *  AA290....
C    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  AA300....
                                                                        AA310....
C    This code is a modified version of the USGS ground-water flow      AA320....
C    and transport program SATRA, which in turn is a simplified         AA330....
C    version of the model SUTRA (Voss, 1984)                            AA340....
                                                                        AA350....
                                                                        AA360....
C    -----------------------------------------------------------------  AA370....
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                AA380....
                                                                        AA390....
     COMMON/LGEM/ RM                                                    AA400....
     COMMON/LGEV/ RV                                                    AA410....
     COMMON/LGEMV/ IMV                                                  AA420....
     COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,       AA430....
    1   NC4BC,NSO,NBCN,NBCSUM,NCONT                                     AA440....
     COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,          AA450....
    1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH                 AA460....
                                                                        AA470....
     CHARACTER*1 TITLE(80)                                              AA480....
     CHARACTER*6 SIMULA(2)                                              AA490....
                                                                        AA500....
     DIMENSION KRV(80)                                                  AA510....
     DIMENSION KRM(5)                                                   AA520....
C    -------------------------------------------------------            AA530....
```

```
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *   The user must dimension three arrays as follows:   *
C     *                                                      *
C     *   DIMENSION  RM( RMDIM), RV( RVDIM), IMV(IMVDIM)     *
C     *                                                      *
C     *   RMDIM  >= 4*NN*NBI                                 *
C     *   RVDIM  >= NNV*NN + NEV*NE + NBCN*13 + 13           *
C     *   IMVDIM >= NE*8 + NN + NPINCH*3 + NSO + NBCN*4 + 8 *
C     *                                                      *
C     *   where,                                             *
C     *                                                      *
C     *   NNV = 38                                           *
C     *   NEV = 8                                            *
C     *   NBCN = NHBC + NC1BC + NC2BC + NC4BC                *
C     *                                                      *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      DIMENSION   RM(80000), RV(25000), IMV(4000)


C     --- READ PROGRAM CONTROL DATA AND WRITE TO THE OUTPUT FILE ---
      READ(5,100) SIMULA
  100 FORMAT(2A6)
      WRITE(6,110)
  110 FORMAT(1H1,4(1X,131(1H*)////)//////
     1 15X,' SSSSSS        A    TTTTTTTTTTT  RRRRRRR          A     ',
     2 '         CCCCCCCC  HH     HH  EEEEEEEE  MM         MM'/
     3 15X,'SSS   SS      AAA   TT  TTT  TT  RR    RR        AAA    ',
     4 '        CCCCCCCCC  HH     HH  EE        MMM       MMM'/
     5 15X,' SSSS        A   A      TTT      RRRRRRR        A   A   ',
     6 '  ----  CC         HH     HH  EEEE      MMMM     MMMM'/
     7 15X,'    SSSS    AAAAAAA     TTT      RR  RR        AAAAAAA  ',
     8 '  ----  CC         HHHHHHHHH  EE        MM MM   MM MM'/
     9 15X,'SS   SSSS  AAA   AAA    TTT      RR   RRR     AAA   AAA ',
     * '        CCCCCCCC   HH     HH  EE        MM  MM MM  MM'/
     1 15X,' SSSSSSS  AAAA   AAAA   TTT      RR    RRRR  AAAA   AAAA',
     2 '         CCCCCCC   HH     HH  EEEEEEEE  MM   MMM   MM'/
     3 6(/),36X,'U N I T E D   S T A T E S   ',
     4 'G E O L O G I C A L   S U R V E Y'////
     5 37X,'GROUNDWATER FLOW AND MULTISOLUTE TRANSPORT',
     6 ' SIMULATION MODEL'//57X,'-VERSION 0584-'///
     7 37X,'   AREAL OR CROSS-SECTIONAL SATURATED FLOW AND',
     8 ' TRANSPORT  '/36X,'WITH SORPTION, ION EXCHANGE AND',
     9 ' AQUEOUS EQUILIBRIUM CHEMISTRY'////4(////1X,131(1H*)))

      IF(SIMULA(1).NE.'SATRA ') GOTO 120
      IF(SIMULA(2).EQ.'AREAL ') GOTO 140
      IF(SIMULA(2).EQ.'CROSS-') GOTO 160
  120 WRITE(6,130)
  130 FORMAT(1H1/////20X,'* * * * *   ERROR IN FIRST DATA CARD--',
     1   '------DATA INPUT HALTED FOR CORRECTIONS   * * * * *')
      STOP
  140 ME=-1                                                             AA1070...
      WRITE(6,150)                                                      AA1080...
  150 FORMAT(1H1//132(1H*)///13X,'* * * * *  S A T R A 2  A R E A L', AA1090...
     1 ' S O L U T E  T R A N S P O R T  S I M U L A T I O N ',       AA1100...
     2 '* * * * *'////132(1H*))                                         AA1110...
      GOTO 180                                                          AA1120...
  160 ME=+1                                                             AA1130...
      WRITE(6,170)                                                      AA1140...
  170 FORMAT(1H1//132(1H*)///10X,'* * * * *  S A T R A 2  C R O S S ',AA1150...
     1 '- S E C T I O N A L  T R A N S P O R T  S I M U L A T I O N'AA1160...
     2 '  * * * * *'////132(1H*))                                       AA1170...
  180 CONTINUE                                                          AA1180...
                                                                        AA1190...
      READ(5,190)TITLE                                                  AA1200...
  190 FORMAT(80A1)                                                      AA1210...
      READ(5,200) NN,NE,NBI,NPINCH,NHBC,NC1BC,NC2BC,NC4BC,NSO,ISSTAT,   AA1220...
     1   IREAD,ISTORE                                                   AA1230...
  200 FORMAT(16I5)                                                      AA1240...
      IF(ISSTAT.EQ.+1) WRITE(6,210)                                     AA1250...
  210 FORMAT(///45X,'-- WITH STEADY-STATE FLOW FIELD --'/               AA1260...
     1       45X,'   ---- ------ ----- ---- -----   ')                  AA1270...
      IF(IREAD.LE.0) WRITE(6,220)                                       AA1280...
  220 FORMAT(///32X,'(START-UP DATA IS TO BE RETRIEVED FROM',           AA1290...
     1   ' STORAGE FILE - UNIT 55.)')                                   AA1300...
      IF(ISTORE.EQ.1) WRITE(6,230)                                      AA1310...
  230 FORMAT(///32X,'(FINAL SOLUTION IS TO BE STORED ON ',              AA1320...
     1   'STORAGE FILE - UNIT 66.)')                                    AA1330...
      WRITE(6,240) TITLE                                                AA1340...
  240 FORMAT(////1X,131(1H-)/26X,80A1/1X,131(1H-))                      AA1350...
                                                                        AA1360...
                                                                        AA1370...
      WRITE(6,250) NN,NE,NBI,NPINCH,NHBC,NC1BC,NC2BC,NC4BC,NSO          AA1380...
  250 FORMAT(////////11X,'S I M U L A T I O N   D I M E N S I O N',     AA1390...
     1   '   D A T A'//11X,I6,5X,'NUMBER OF NODES IN FINITE ',          AA1400...
     2   'ELEMENT MESH'/11X,I6,5X,'NUMBER OF ELEMENTS IN MESH'/         AA1410...
     3   11X,I6,5X,'ESTIMATED MAXIMUM FULL BANDWIDTH FOR MESH'/         AA1420...
     4   11X,I6,5X,'EXACT NUMBER OF PINCH NODES IN MESH'/               AA1430...
     5   11X,I6,5X,'EXACT NUMBER OF NODES IN MESH AT WHICH ',           AA1440...
     6   'HYDRAULIC HEAD IS A SPECIFIED CONSTANT OR FUNCTION OF TIME'/  AA1450...
     7   11X,I6,5X,'EXACT NUMBER OF NODES IN MESH AT WHICH ',           AA1460...
     8   'CONCENTRATION OF SOLUTE C1 IS A SPECIFIED CONSTANT OR ',      AA1470...
     9   'FUNCTION OF TIME'/11X,I6,5X,'EXACT NUMBER OF NODES IN ',      AA1480...
     *   'MESH AT WHICH CONCENTRATION OF SOLUTE C2 IS A SPECIFIED ',    AA1490...
     1   'CONSTANT OR FUNCTION OF TIME'/11X,I6,5X,'EXACT NUMBER ',      AA1500...
     2   'OF NODES IN MESH AT WHICH CONCENTRATION OF SOLUTE C4 IS ',    AA1510...
     3   'A SPECIFIED CONSTANT OR FUNCTION OF TIME'/11X,I6,5X,          AA1520...
     4   'EXACT NUMBER OF NODES AT',                                    AA1530...
     5   ' WHICH FLUID INFLOW OR OUTFLOW IS A SPECIFIED CONSTANT',      AA1540...
     6   ' OR FUNCTION OF TIME')                                        AA1550...
                                                                        AA1560...
                                                                        AA1570...
                                                                        AA1580...
C     --- SET COEFFICIENTS FOR VARIABLE INDENTIFICATION ----            AA1590...
```

```
      NBCSUM=NC1BC+NC2BC+NC4BC                                          AA1600...
      NBCN=NHBC+NBCSUM+1                                                AA1610...
      NSO=NSO+1                                                         AA1620...
      NPINCH=NPINCH+1                                                   AA1630...
      MATDIM=NN*NBI                                                     AA1640...
      NIN=NE*8                                                          AA1650...
                                                                        AA1660...
C     --- REAL MATRIX IDENTIFIERS ---                                   AA1670...
      NNMAT=4                                                           AA1680...
      M2=1                                                              AA1690...
      KRM(1)=1                                                          AA1700...
      M1=M2+1                                                           AA1710...
      M2=M2+(NNMAT)                                                     AA1720...
      DO 260 J=M1,M2                                                    AA1730...
  260 KRM(J)=KRM(J-1)+MATDIM                                            AA1740...
                                                                        AA1750...
C     --- REAL VECTOR IDENTIFIERS ---                                   AA1760...
      NNV=28+10                                                         AA1770...
      NEV=8                                                             AA1780...
      NBCV=9                                                            AA1790...
      M2=1                                                              AA1800...
      KRV(1)=1                                                          AA1810...
      M1=M2+1                                                           AA1820...
      M2=M2+(NNV)                                                       AA1830...
      DO 270 J=M1,M2                                                    AA1840...
  270 KRV(J)=KRV(J-1)+NN                                                AA1850...
      M1=M2+1                                                           AA1860...
      M2=M2+(NEV)                                                       AA1870...
      DO 280 J=M1,M2                                                    AA1880...
  280 KRV(J)=KRV(J-1)+NE                                                AA1890...
      M1=M2+1                                                           AA1900...
      M2=M2+(NBCV)                                                      AA1910...
      DO 290 J=M1,M2                                                    AA1920...
  290 KRV(J)=KRV(J-1)+NBCN                                              AA1930...
                                                                        AA1940...
C     --- INTEGER MATRIX, VECTOR IDENTIFIERS ---                        AA1950...
      KIMV1=1                                                           AA1960...
      KIMV2=KIMV1+NIN                                                   AA1970...
      KIMV3=KIMV2+NPINCH*3                                              AA1980...
      KIMV4=KIMV3+NSO                                                   AA1990...
      KIMV5=KIMV4+NBCN                                                  AA2000...
      KIMV6=KIMV5+NBCN                                                  AA2010...
      KIMV7=KIMV6+NBCN                                                  AA2020...
      KIMV8=KIMV7+NBCN                                                  AA2030...
                                                                        AA2040...
                                                                        AA2050...
C     --- CALL THE PRINCIPAL OPERATIONAL SUBROUTINE ---                 AA2060...
                                                                        AA2070...
      CALL SATCHM (RM(KRM(1)),RM(KRM(2)),RM(KRM(3)),RM(KRM(4)),         AA2080...
     1   RV(KRV(1)),RV(KRV(2)),RV(KRV(3)),RV(KRV(4)),RV(KRV(5)),        AA2090...
     2   RV(KRV(6)),RV(KRV(7)),RV(KRV(8)),RV(KRV(9)),RV(KRV(10)),       AA2100...
     3   RV(KRV(11)),RV(KRV(12)),RV(KRV(13)),RV(KRV(14)),RV(KRV(15)),   AA2110...
     4   RV(KRV(16)),RV(KRV(17)),RV(KRV(18)),RV(KRV(19)),RV(KRV(20)),   AA2120...
5  RV(KRV(21)),RV(KRV(22)),RV(KRV(23)),RV(KRV(24)),RV(KRV(25)),      AA2130...
6  RV(KRV(26)),RV(KRV(27)),RV(KRV(28)),RV(KRV(29)),RV(KRV(30)),      AA2140...
7  RV(KRV(31)),RV(KRV(32)),RV(KRV(33)),RV(KRV(34)),RV(KRV(35)),      AA2150...
8  RV(KRV(36)),RV(KRV(37)),RV(KRV(38)),RV(KRV(39)),RV(KRV(40)),      AA2160...
9  RV(KRV(41)),RV(KRV(42)),RV(KRV(43)),RV(KRV(44)),RV(KRV(45)),      AA2170...
*  RV(KRV(46)),RV(KRV(47)),RV(KRV(48)),RV(KRV(49)),RV(KRV(50)),      AA2180...
1  RV(KRV(51)),RV(KRV(52)),RV(KRV(53)),RV(KRV(54)),RV(KRV(55)),      AA2190...
2  IMV(KIMV1),IMV(KIMV2),IMV(KIMV3),IMV(KIMV4),IMV(KIMV5),           AA2200...
3  IMV(KIMV6),IMV(KIMV7),IMV(KIMV8) )                                AA2210...
                                                                     AA2220...
                                                                     AA2230...
 ENDFILE(6)                                                          AA2240...
 STOP                                                                AA2250...
 END                                                                 AA2260...
```

```
C     SUBROUTINE       S  A  T  C  H  M       SATRA-CHEM - VERSION 0684 A10.....
                                                                        A20.....
C     FUNCTION:                                                         A30.....
C      PRINCIPAL CONTROL ROUTINE FOR SATRA-CHEM SIMULATIONS             A40.....
C      ORGANIZES DATA INPUT, INITIALIZATION, CALCULATIONS FOR           A50.....
C      EACH TIME STEP AND ITERATION, AND VARIOUS OUTPUTS.               A60.....
C      CALL MOST OTHER SUBROUTINES.                                     A70.....
                                                                        A80.....
C     ----------------------------------------------------------------- A90.....
      SUBROUTINE SATCHM (HMAT,UMAT1,UMAT2,UMAT,                         A100....
     1   PPROJ,UPROJ,VPROJ,WPROJ,HM1,UM1,VM1,WM1,                       A110....
     2   X,Y,THICK,VOL,POR,STOR,CS1,CS2,CS3,QIN,QINM1,                  A120....
     3   UIN,VIN,WIN,HVEC,UVEC,VVEC,WVEC,                               A130....
     4   WITER,CC,XX,YY,C1,C1P1,C2,C3,C4,UAVG,VAVG,WAVG,                A140....
     5   PERMXX,PERMXY,PERMYX,PERMYY,DISPL,DISPT,VMAG,VANG,             A150....
     6   HBC,UBC,VBC,WBC,C1BC,C2BC,C4BC,QPL,QPLM1,                      A160....
     7   IN,IPINCH,IQSO,IHBC,IC1BC,IC2BC,IC4BC,INDEX )                  A170....
                                                                        A180....
C     ----------------------------------------------------------------- A190....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               A200....
                                                                        A210....
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,      A220....
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT                                    A230....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        A240....
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTHM1,DLTUM1,DLTPM2,DLTUM2          A250....
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,         A260....
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH                A270....
      COMMON/ITERAT/ITER,ITRMAX,RPM,RPMAX,IPWORS,RWM,RWMAX,IWWORS,      A280....
     1   RTOL,MAXITR                                                    A290....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,RHOS,DECAY,SIGMAW,FF,CBART     A300....
      COMMON/BFACS/ BDELP,BDELU                                         A310....
      COMMON/GNUBC/GNU0,GNU                                             A320....
      COMMON/KPRINT/ KCOORD,KELINF,KINCID,KPLOTP,KPLOTU,KVEL            A330....
      COMMON/CHEM/ EQCSTU,EQCSTV                                        A340....
                                                                        A350....
      DIMENSION HMAT(NN,NBI),UMAT1(NN,NBI),UMAT2(NN,NBI),UMAT(NN,NBI)   A360....
      DIMENSION HVEC(NN),UVEC(NN),VVEC(NN),WVEC(NN)                     A370....
      DIMENSION HM1(NN),UM1(NN),VM1(NN),WM1(NN),WITER(NN)               A380....
      DIMENSION UAVG(NN),VAVG(NN),WAVG(NN)                              A390....
      DIMENSION C1(NN),C1P1(NN),C2(NN),C3(NN),C4(NN)                    A400....
      DIMENSION VOL(NN),POR(NN),STOR(NN),CS1(NN),CS2(NN),CS3(NN)        A410....
      DIMENSION PPROJ(NN),UPROJ(NN),VPROJ(NN),WPROJ(NN)                 A420....
      DIMENSION QIN (NN),UIN(NN),VIN(NN),WIN(NN),IQSO(NSO)              A430....
      DIMENSION IHBC(NBCN),HBC(NBCN),UBC(NBCN),VBC(NBCN),WBC(NBCN)      A440....
      DIMENSION IC1BC(NBCN),IC2BC(NBCN),IC4BC(NBCN),QINM1(NN)           A450....
      DIMENSION C1BC(NBCN),C2BC(NBCN),C4BC(NBCN)                        A460....
      DIMENSION QPL(NBCN),QPLM1(NBCN)                                   A470....
      DIMENSION IN(NIN),IPINCH(NPINCH,3),X(NN),Y(NN),THICK(NN)          A480....
      DIMENSION DISPL(NE),DISPT(NE),VMAG(NE),VANG(NE)                   A490....
      DIMENSION PERMXX(NE),PERMXY(NE),PERMYX(NE),PERMYY(NE)             A500....
      DIMENSION CC(NN),INDEX(NN),XX(NN),YY(NN)                          A510....
                                                                        A520....
C     ----------------------------------------------------------------- A530....
```

```

      READ(5,1) GNU0
    1 FORMAT(D15.0)

C     --- READ IN BASIC NODEWISE AND ELEMENTWISE DATA ---
      CALL INDAT1(X,Y,THICK,POR,STOR,DISPL,DISPT,PERMXX,PERMXY,
     1   PERMYX,PERMYY)


C     --- PLOT NODE NUMBERS IF REQUESTED ---
      IF(KPLOTP+KPLOTU.GE.0) CALL PLOT(0,1,X,Y,CC,INDEX,XX,YY,HVEC)

      IT=0
      IQSOT=1
      IHBCT=1
      IUBCT=1
      MATDIM=NN*NBI
      GNU=GNU0*1.0000D0

      CALL ZERO(QIN,NN,0.0D0)
      CALL ZERO(QINM1,NN,0.0D0)
      CALL ZERO(UIN,NN,0.0D0)
      CALL ZERO(VIN,NN,0.0D0)
      CALL ZERO(WIN,NN,0.0D0)

C     --- READ IN NODAL SOURCE/SINK DATA ---
      IF(NSO-1.GT.0) CALL SOURCE(UIN,VIN,WIN,IQSOT,IQSO,QIN)


C     --- READ IN HEAD, CONCENTRATION BOUNDARY CONDITIONS ---
      IF(NBCN-1.GT.0)
     1 CALL BOUND(IHBC,HBC,IC1BC,IC2BC,IC4BC,UBC,VBC,WBC,C1BC,C2BC,
     2   C4BC,IHBCT,IUBCT)

C     --- SET FLAG FOR TIME DEPENDENT SOURCES OR BOUNDARY CONDITIONS -
      IBCT=IQSOT+IHBCT+IUBCT


C     --- READ IN ELEMENT INCIDENCE AND PINCH NODE DATA ---
      CALL CONNEC(IN,IPINCH)


C     --- CALCULATE TOTAL GLOBAL MATRIX BANDWIDTH ---
      CALL BANWID(IN)


C     --- CHECK CONDITIONS SPECIFIED AT PINCH NODES (IF APPLICABLE) ---
      IF(NPINCH-1.GT.0)
     1 CALL NCHECK(IPINCH,IQSO,IHBC,IC1BC,IC2BC,IC4BC)


C     --- READ IN CURRENT/RECENT VALUES OF HEAD, CONCENTRATION ---
      CALL INDAT2(HVEC,C1,C2,C3,C4,HM1,UM1,VM1,WM1,
     1   UVEC,VVEC,WVEC,CS1,CS3,QPL,QPLM1,UBC,VBC,WBC,C1P1)


C     --- INITIALIZE CHEMICAL CONSTITUENT FLAG ---
      IFLAG=1


C     --- CALCULATE INITIAL TEMPORAL COEFFICIENTS ---
      CALL TIMES(1,ML,TSECP0,TSECU0)


C     --- PRINT OUTPUT HEADINGS ---
      CALL PRISOL(0,0,0,HVEC,C1,C2,C3,C4,VMAG,VANG)


C     --- ADJUST FOR STEADY STATE FLOW ---
      IF(ISSTAT.EQ.1) THEN
      ML=1
      ISSTAT=2
      ITER=0
      GOTO 1095
      END IF


C     ------------------------------------------------------------------
C                     --- BEGIN NEW TIME STEP ---
C     ------------------------------------------------------------------
 1000 IT=IT+1
      ITER=0
      ML=0
      NOMATX=0
      IF(MOD(IT-1,NPCYC).NE.0.AND.IT.GT.2)  NOMATX=1
      IF(ISSTAT.EQ.2.AND.IT.GT.2) NOMATX=1
      IF(IT.EQ.1.AND.ISSTAT.NE.2) GOTO 1005
      IF(MOD(IT,NPCYC).NE.0) ML=2
      IF(MOD(IT,NUCYC).NE.0) ML=1
      IF(MOD(IT,ITCYC).EQ.0.AND.IT.GT.1) DELT=DELT*DTMULT


C     --- CALCULATE TEMPORAL COEFFICIENTS FOR CURRENT TIME STEP ---
 1005 CALL TIMES(2,ML,TSECP0,TSECU0)


C     --- UPDATE VECTORS FOR CURRENT TIME STEP ---
 1095 CALL UPDATE (ML,HM1,UM1,VM1,WM1,QIN,QINM1,HVEC,UVEC,VVEC,WVEC,
     1   QPL,QPLM1)


C     ------------------------------------------------------------------
C                 --- START NEXT GLOBAL ITERATION ---
C     ------------------------------------------------------------------
 1100 ITER=ITER+1

 1111 IF(IFLAG.EQ.3) THEN
      DO 1200 I=1,NN
C     --- SET INITIAL GUESS FOR C1 FOR NEWTON-RAPHSON ITERATION ---
      C 1 P 1 (I) = 1.0D+05
 1200 WITER(I)=WVEC(I)
      END IF


C     --- DETERMINE SORPTION COEFFICIENTS ---
      IF(IFLAG.EQ.3.AND.ISORB.EQ.1) THEN
      CALL ADSORB(CS1,CS2,CS3,C1,C2,C4,UVEC,VVEC,
     1   UVEC,VVEC,UM1,UM2,VM1,VM2)


C     --- DETERMINE ION EXCHANGE COEFFICIENTS ---
      ELSE IF(IFLAG.EQ.3.AND.IEXCH.EQ.1) THEN
      CALL XCHNGE(C1,UVEC,VVEC,UVEC,VVEC,WVEC,UM1,VM1,CS1,CS3)

      ELSE
      CALL ZERO(CS1,NN,0.0D0)
      CALL ZERO(CS2,NN,0.0D0)
      CALL ZERO(CS3,NN,0.0D0)
      END IF


C     --- INITIALIZE SOLUTION VECTORS AND MATRICIES ---
 2000 IF(ML-1) 3000,3000,3300
 3000 CALL ZERO(HMAT,MATDIM,0.0D0)
      CALL ZERO(HVEC,NN,0.0D0)
      CALL ZERO(VOL,NN,0.0D0)
      IF(ML-1) 3300,3400,3300
 3300 IF(NOMATX.EQ.0.AND.IFLAG.EQ.1) CALL ZERO(UMAT,MATDIM,0.0D0)
 3375 IF (IFLAG .EQ. 1) CALL ZERO(UVEC,NN,0.0D0)
      IF (IFLAG .EQ. 2) CALL ZERO(VVEC,NN,0.0D0)
      IF (IFLAG .EQ. 3) CALL ZERO(WVEC,NN,0.0D0)
 3400 CONTINUE


C     - ADJUST FOR TEMPORAL CHANGES IN HEAD, CONC. BOUNDARY CONDITIONS
      IF(ITER.EQ.1.AND.IBCT.NE.3)
     1   CALL BCTIME(IHBC,HBC,IC1BC,IC2BC,IC4BC,UBC,VBC,WBC,
     2   UIN,VIN,WIN,IQSO,IHBCT,IUBCT,IQSOT,QIN)


C     --- ASSEMBLE MATRIX EQUATIONS ---
      IF((NOMATX.EQ.0.AND.IFLAG.EQ.1).OR.ML.LT.2)
     1   CALL ELEMEN(ML,IN,X,Y,THICK,HM1,POR,DISPL,DISPT,
     2   PERMXX,PERMXY,PERMYX,PERMYY,VMAG,VANG,
     3   VOL,HMAT,HVEC,UMAT,UVEC,VVEC,WVEC)

      IF(NOMATX.EQ.0.AND.IFLAG.EQ.1.AND.ML.NE.1) THEN
      DO 3660 J=1,NBI
      DO 3660 I=1,NN
 3660 UMAT1(I,J)=UMAT(I,J)                                              A2130...
      ELSE IF(IFLAG.EQ.3) THEN                                          A2140...
      DO 3690 J=1,NBI                                                   A2150...
      DO 3690 I=1,NN                                                    A2160...
 3690 UMAT2(I,J)=UMAT(I,J)                                              A2170...
      END IF                                                            A2180...
                                                                        A2190...
                                                                        A2200...
C     --- CHANGE MATRIX EQUATIONS TO ADD IN SOURCE/SINK FLOW RATES,     A2210...
C         CONCENTRATIONS, AND REACTIVE COMPONENTS ---                   A2220...
                                                                        A2230...
      CALL NODALB(ML,VOL,HMAT,HVEC,UMAT1,UMAT2,UVEC,VVEC,WVEC,          A2240...
     1   HM1,UM1,VM1,WM1,POR,STOR,CS1,CS2,CS3,UIN,VIN,WIN,QIN)          A2250...
                                                                        A2260...
                                                                        A2270...
C     -CHANGE MATRIX EQUATIONS TO ADD IN DIRICHLET BOUNDARY CONDITIONS  A2280...
      CALL BCB(ML,HMAT,HVEC,UMAT1,UMAT2,UVEC,VVEC,WVEC,IHBC,HBC,IC1BC,  A2290...
     1   IC2BC,IC4BC,UBC,VBC,WBC,QPL,UBCM1,QPLM1,HM1)                   A2300...
                                                                        A2310...
                                                                        A2320...
C     --- ADJUST MATRICES FOR PINCH NODES (IF APPLICABLE) ---           A2330...
      IF(NPINCH-1) 4200,4200,4000                                       A2340...
 4000 CALL PINCHB(ML,IPINCH,HMAT,HVEC,UMAT1,UMAT2,UVEC,VVEC,WVEC)       A2350...
 4200 CONTINUE                                                          A2360...
                                                                        A2370...
                                                                        A2380...
      IHALFB=NBHALF-1                                                   A2390...
      IF(ML-1) 5000,5000,5500                                           A2400...
                                                                        A2410...
C     --- SOLVE FOR UNKNOWN PRESSURE ---                                A2420...
 5000 CALL SOLVEB(000,HMAT,HVEC,NN,IHALFB,NN,NBI)                       A2430...
      DO 5200 IP=1,NHBC                                                 A2440...
      I=IABS(IHBC(IP))                                                  A2450...
 5200 QPL(IP)=GNU*(HBC(IP)-HVEC(I))                                     A2460...
                                                                        A2470...
C     --- SOLVE FOR UNKNOWN CONCENTRATION ---                           A2480...
      IF(ML-1) 5500,6000,5500                                           A2490...
 5500 IF(IFLAG.EQ.1) THEN                                               A2500...
      KKK=000000                                                        A2510...
      IF(NOMATX.EQ.1) KKK=2                                             A2520...
      CALL SOLVEB(KKK,UMAT1,UVEC,NN,IHALFB,NN,NBI)                      A2530...
      DO 5550 I=1,NN                                                    A2540...
 5550 UAVG(I)=(UVEC(I)+UM1(I))*0.50D0                                   A2550...
      ELSE IF (IFLAG.EQ.2) THEN                                         A2560...
      CALL SOLVEB(002,UMAT1,VVEC,NN,IHALFB,NN,NBI)                      A2570...
      DO 5575 I=1,NN                                                    A2580...
 5575 VAVG(I)=(VVEC(I)+VM1(I))*0.50D0                                   A2590...
      ELSE IF (IFLAG.EQ.3) THEN                                         A2600...
      CALL SOLVEB(000,UMAT2,WVEC,NN,IHALFB,NN,NBI)                      A2610...
      CALL PROJEC (UPROJ,VPROJ,WPROJ,WITER,UM1,UM2,                     A2620...
     1   VM1,VM2,WM1,WM2,UVEC,VVEC,WVEC,AU,BU,UAVG,VAVG,WAVG)           A2630...
C     CALL EQCHEM(NN,C1,C1P1,UAVG,VAVG,WAVG,1,C2,C3,C4,1)               A2640...
      CALL EQCHEM(NN,C1,C1P1,UVEC,VVEC,WVEC,1,C2,C3,C4,1)               A2650...
      END IF                                                            A2660...
                                                                        A2670...
 6000 CONTINUE                                                          A2680...
                                                                        A2690...
                                                                        A2700...
C     --- DETERMINE WHEN TO STOP ITERATIONS AND/OR TIME STEP INCREMENTS A2710...
      ISTOP=0                                                           A2720...
      IGOI=0                                                            A2730...
      IF(NCONT.EQ.1) GOTO 7350                                          A2740...
      IF(IFLAG.NE.3) GOTO 7500                                          A2750...
      IF((ITRMAX-1) .GT. 0) THEN                                        A2760...
      RWM=0.D0                                                          A2770...
      IWWORS=0                                                          A2780...
                                                                        A2790...
 7150 IF(ML-1) 7350,7350,7200                                           A2800...
 7200 DO 7300 I=1,NN                                                    A2810...
      RW=ABS(WVEC(I)-WITER(I))                                          A2820...
      IF((RW-RWM).GE.0) THEN                                            A2830...
      RWM=RW                                                            A2840...
      IWWORS=I                                                          A2850...
      END IF                                                            A2860...
 7300 CONTINUE                                                          A2870...
                                                                        A2880...
                                                                        A2890...
C     --- CHECK ERROR TOLERANCE FOR CONCENTRATION ---                   A2900...
      IF(RWM.GT.RWMAX) IGOI=IGOI+2                                      A2910...
                                                                        A2920...
                                                                        A2930...
C     --- SOLUTION NOT CONVERGED WITHIN SPECIFIED NUMBER OF ITERATIONS  A2940...
      IF(IGOI.GT.0.AND.ITER.EQ.ITRMAX) ISTOP=-1                         A2950...
                                                                        A2960...
                                                                        A2970...
C     --- GO THROUGH ANOTHER ITERATION ---                              A2980...
      IF(IGOI.GT.0.AND.ISTOP.EQ.0) GOTO 1100                            A2990...
                                                                        A3000...
      END IF                                                            A3010...
 7350 CONTINUE                                                          A3020...
                                                                        A3030...
                                                                        A3040...
C     --- SIMULATION EXCEEDED SPECIFIED NUMBER OF TIME STEPS ---        A3050...
      IF(ISTOP.NE.-1.AND.IT.EQ.ITMAX) ISTOP=1                           A3060...
                                                                        A3070...
                                                                        A3080...
C     --- SIMULATION EXCEEDED SPECIFIED TIME PERIOD ---                 A3090...
      IF(ISTOP.NE.-1.AND.TSEC.GE.TMAX) ISTOP=2                          A3100...
                                                                        A3110...
 7500 IF(ML.NE.1.AND.ISTOP.NE.-1.AND.IFLAG.LT.NCONT) THEN               A3120...
      IFLAG=IFLAG+1                                                     A3130...
      ITER=1                                                            A3140...
      IF(ML.EQ.0) ML=3                                                  A3150...
      GO TO 1111                                                        A3160...
      ELSE IF(ISTOP.NE.-1.AND.NCONT.EQ.1.AND.ML.NE.1) THEN              A3170...
      CALL EQCHEM(NN,C1,C1P1,UVEC,VVEC,WVEC,2,C2,C3,C4,INR)             A3180...
      END IF                                                            A3190...
      IFLAG=1                                                           A3200...
      IF(ML.EQ.3) ML=0                                                  A3210...
                                                                        A3220...
C     --- STORE RESULTS FOR POSSIBLE RESTART EACH TIME STEP             A3230...
      IF(ISTOP.NE.0.AND.ISTORE.EQ.1)                                    A3240...
     1   CALL STORE(HVEC,C1,C2,C4,HM1,UM1,VM1,WM1,CS1,CS3,QPL,QPLM1,    A3250...
     2   UBC,VBC,WBC)                                                   A3260...
                                                                        A3270...
      IF(IT.GT.1.AND.MOD(IT,NPRINT).NE.0.AND.ISTOP.EQ.0) GOTO 8000      A3280...
                                                                        A3290...
C     --- PLOT RESULTS ON OUTPUT ---                                    A3300...
      CALL PRISOL(ML,ISTOP,IGOI,HVEC,C1,C2,C3,C4,VMAG,VANG)             A3310...
      IF(KPLOTP.NE.1.OR.ML.EQ.2) GOTO 7680                              A3320...
      CALL PLOT(1,2,X,Y,CC,INDEX,XX,YY,HVEC)                            A3330...
 7680 IF(KPLOTU.NE.1.OR.ML.EQ.1) GOTO 8000                              A3340...
      NP=3                                                              A3350...
      IF(ME.EQ.+1) NP=4                                                 A3360...
      IF(NCONT.GT.1) THEN                                               A3370...
      CALL PLOT(1, 3,X,Y,CC,INDEX,XX,YY,C1)                             A3380...
      CALL PLOT(1, 4,X,Y,CC,INDEX,XX,YY,C2)                             A3390...
      IF(IEXCH.EQ.1) THEN                                               A3400...
      CALL PLOT(1, 5,X,Y,CC,INDEX,XX,YY,C3)                             A3410...
      ELSE                                                              A3420...
      CALL PLOT(1, 5,X,Y,CC,INDEX,XX,YY,C4)                             A3430...
      END IF                                                            A3440...
      ELSE                                                              A3450...
      CALL PLOT(1, 4,X,Y,CC,INDEX,XX,YY,C2)                             A3460...
      END IF                                                            A3470...
 8000 CONTINUE                                                          A3480...
                                                                        A3490...
      IF(ISTOP.EQ.0) GO TO 1000                                         A3500...
                                                                        A3510...
      IF(ISTORE.EQ.1) WRITE(6,8200)                                     A3520...
 8200 FORMAT(///11X,'*** FINAL SOLUTION HAS BEEN STORED ON UNIT 66 ***')A3530...
      IF(ISTOP.GT.0) GOTO 8400                                          A3540...
      IF(IGOI-2) 8230,8262,8292                                         A3550...
 8230 WRITE(6,8235)                                                     A3560...
 8235 FORMAT(////////11X,'SIMULATION TERMINATED DUE TO ',               A3570...
     1   'NON-CONVERGENT HYDRAULIC HEAD',                               A3580...
     2                /11X,'********** ********** *** ** ',             A3590...
     3   '************** ********* ****')                               A3600...
      RETURN                                                            A3610...
 8262 WRITE(6,8264)                                                     A3620...
 8264 FORMAT(////////11X,'SIMULATION TERMINATED DUE TO ',               A3630...
     1   'NON-CONVERGENT CONCENTRATION',                                A3640...
     2                /11X,'********** ********** *** ** ',             A3650...
     3   '************** *************')                                A3660...
      RETURN                                                            A3670...
 8292 WRITE(6,8294)                                                     A3680...
 8294 FORMAT(////////11X,'SIMULATION TERMINATED DUE TO ',               A3690...
     1   'NON-CONVERGENT HYDRAULIC HEAD AND CONCENTRATION',             A3700...
     2                /11X,'********** ********** *** ** ',             A3710...
```

```
    3   '************** ********* **** *** *************')
     RETURN

8400 IF(ISTOP.EQ.2) GOTO 8500
     WRITE(6,8450)
8450 FORMAT(////////11X,'SATRA2D SIMULATION TERMINATED AT COMPLETION ',
    1   'OF TIME STEPS'/
    2              11X,'******* ********** ********** ** ********** ',
    3   '** **** *****')
     RETURN
8500 WRITE(6,8550)
8550 FORMAT(////////11X,'SATRA2D SIMULATION TERMINATED AT COMPLETION ',
    1   'OF TIME PERIOD'/
    2              11X,'******* ********** ********** ** ********** ',
    3   '** **** ******')
     RETURN


     END
```

```
C     SUBROUTINE           I  N  D  A  T  1   SATRA-CHEM - VERSION 0684
      SUBROUTINE INDAT1(X,Y,THICK,POR,STOR,DISPL,DISPT,PERMXX,PERMXY,
     1   PERMYX,PERMYY)

C     FUNCTION:
C      TO INPUT, OUTPUT AND ORGANIZE A PORTION OF UNIT-5 INPUT DATA

C     -----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NPBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTPM1,DLTUM1,DLTPM2,DLTUM2
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/ITERAT/ITER,ITRMAX,RPM,RPMAX,IPWORS,RWM,RWMAX,IWWORS,
     1   RTOL,MAXITR
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,RHOS,DECAY,SIGMAW,FF,CBART
      COMMON/KPRINT/ KCOORD,KELINF,KINCID,KPLOTP,KPLOTU,KVEL
      COMMON/GNUBC/GNU0,GNU
      COMMON/CHEM/ EQCSTU,EQCSTV

      DIMENSION X(NN),Y(NN),THICK(NN),POR(NN),STOR(NN)
      DIMENSION DISPL(NE),DISPT(NE)
      DIMENSION PERMXX(NE),PERMXY(NE),PERMYX(NE),PERMYY(NE)

C     -----------------------------------------------------------------
      INSTOP=0

C     --- INPUT DATASET 5 ---
      READ(5,50) UP
   50 FORMAT(1F10.0)
      WRITE(6,70) GNU0,UP
   70 FORMAT(////11X,'S P A T I A L   C O N T R O L   D A T A'//
     1   11X,1PD15.4,5X,'HYDRAULIC HEAD BOUNDARY CONDITION FACTOR - GNU'
     1   /11X,0PF15.5,5X,'"UPSTREAM WEIGHTING" FACTOR')

C     --- INPUT DATASET 6 ---
      READ(5,100) ITMAX,DELT,TMAX,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT
  100 FORMAT(I10,2D10.0,I10,F10.0,3I10)
      WRITE(6,120) ITMAX,DELT,TMAX,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT
  120 FORMAT(////11X,'T E M P O R A L   C O N T R O L   D A T A',
     1   //11X,I15,5X,'MAXIMUM NUMBER OF TIME STEPS'/11X,1PD15.4,5X,
     2   'INITIAL TIME STEP (IN SECONDS)'//11X,1PD15.4,5X,
     3   'MAXIMUM SIMULATION TIME (IN SECONDS)'/11X,I15,5X,
     4   'TIME STEP CHANGE CYCLE (IN TIME STEPS)'/11X,0PF15.5,5X,
     5   'MULTIPLIER FOR TIME STEP CHANGE'//11X,I15,5X,
     6   'FLOW SOLUTION CYCLE (IN TIME STEPS)'/11X,I15,5X,
     7   'TRANSPORT SOLUTION CYCLE (IN TIME STEPS)'/11X,I15,5X,
     8   'PRINTED OUTPUT CYCLE (IN TIME STEPS)')
      IF(NPCYC.GE.1.AND.NUCYC.GE.1) GOTO 140
      WRITE(6,130)
  130 FORMAT(//11X,'* * * * ERROR DETECTED :  BOTH NPCYC AND ',
     1   'NUCYC MUST BE AT LEAST =1 .')                                 B540....
      INSTOP=INSTOP-1                                                   B550....
  140 IF(NPCYC.EQ.1.OR.NUCYC.EQ.1) GOTO 160                             B560....
      WRITE(6,150)                                                      B570....
  150 FORMAT(//11X,'* * * * ERROR DETECTED :  EITHER NPCYC OR ',        B580....
     1   'NUCYC MUST BE SET TO 1.')                                     B590....
      INSTOP=INSTOP-1                                                   B600....
  160 CONTINUE                                                          B610....
      IF(ISSTAT.EQ.0) GOTO 164                                          B620....
      NPCYC=ITMAX+1                                                     B630....
  164 CONTINUE                                                          B640....
                                                                        B650....
C     --- INPUT DATASET 7 ---                                           B660....
      READ(5,170) KCOORD,KELINF,KINCID,KPLOTP,KPLOTU,KVEL               B670....
  170 FORMAT(16I5)                                                      B680....
      IF(KVEL.EQ.1) WRITE(6,172)                                        B690....
  172 FORMAT(/31X,'FLUID VELOCITIES (AT ELEMENT CENTROIDS) ARE PRINTED'/B700....
     1   31X,'EACH TIME HYDRAULIC HEADS ARE PRINTED.')                  B710....
      IF(KVEL.NE.1) WRITE(6,173)                                        B720....
  173 FORMAT(/31X,'FLUID VELOCITIES (AT ELEMENT CENTROIDS) ARE ',       B730....
     1   'NOT OUTPUT.')                                                 B740....
                                                                        B750....
C     --- INPUT DATASET 8 ---                                           B760....
      READ(5,175) ITRMAX,RPMAX,RWMAX,RTOL,MAXITR                        B770....
  175 FORMAT(I10,3D10.0,I10)                                            B780....
                                                                        B790....
C     --- INPUT DATASET 9 ---                                           B800....
      READ(5,200) NCONT,ISORB,IEXCH,IEQLIB                              B810....
                                                                        B820....
C     --- INPUT DATASET 10 ---                                          B830....
      READ(5,210) SIGMAW,RHOS,DECAY,FF,CBART,EQCSTU,EQCSTV              B840....
  200 FORMAT(I2,3I5)                                                    B850....
  210 FORMAT(7D10.0)                                                    B860....
                                                                        B870....
C      --- IN ION EXCHANGE, IEQLIB REFERS ONLY TO EQCSTU ---            B880....
      IF((IEQLIB.EQ.0.OR.NCONT.EQ.1).AND.IEXCH.EQ.0) THEN               B890....
      EQCSTU=0.0D0                                                      B900....
      EQCSTV=0.0D0                                                      B910....
      IEQLIB=0                                                          B920....
      END IF                                                            B930....
      IF(ISORB.EQ.0) FF=0.0D0                                           B940....
      IF(IEXCH.EQ.0) CBART=0.0D0                                        B950....
                                                                        B960....
C      --- ALLOW ITERATIONS FOR WVEC ONLY IF THERE ARE                  B970....
C      --- EQUILIBRIUM REACTIONS AND SORPTION OR ION EXCHANGE ---       B980....
      IF((IEQLIB.LT.1.OR.ISORB.NE.1).AND.IEXCH.EQ.0) ITRMAX=1           B990....
      WRITE(6,180) ITRMAX,RPMAX,RWMAX,MAXITR,RTOL                       B1000...
  180 FORMAT(////11X,'I T E R A T I O N   C O N T R O L   D A T A',     B1010...
     1   //11X,I15,5X,'MAXIMUM NUMBER OF ITERATIONS PER TIME STEP',     B1020...
     2   /11X,1PD15.4,5X,'ABSOLUTE CONVERGENCE CRITERION FOR FLOW',     B1030...
     3   ' SOLUTION'/11X,1PD15.4,5X,'ABSOLUTE CONVERGENCE CRITERION',   B1040...
     4   ' FOR TRANSPORT SOLUTION'/11X,I15,5X,'MAXIMUM NUMBER ',        B1050...
     5   'OF ITERATIONS FOR NEWTON-RAPHSON SOLUTION'/11X,1PD15.4,5X,    B1060...
     6   'ABSOLUTE CONVERGENCE CRITERION FOR NEWTON-RAPHSON SOLUTION')  B1070...
                                                                        B1080...
      WRITE(6,220) NCONT,SIGMAW,RHOS,DECAY,FF,CBART,EQCSTU,EQCSTV       B1090...
  220 FORMAT(////11X,'C O N S T A N T   P R O P E R T I E S   O F',     B1100...
     1   '   F L U I D , M E D I U M   A N D   S O L U T E'//           B1110...
     2   11X,I15,5X,'NUMBER OF CHEMICAL CONSTITUENTS'/                  B1120...
     2   11X,1PD15.4,5X,'SOLUTE DIFFUSIVITY IN FLUID'/11X,1PD15.4,5X,   B1130...
     3   'NORMALIZED SOLID GRAIN DENSITY'/11X,1PD15.4,5X,               B1140...
     4   'SOLUTE DECAY RATE'/11X,1PD15.4,5X,'LINEAR SORPTION',          B1150...
     5   ' CONSTANT'/11X,1PD15.4,5X,'EXCHANGE CAPACITY OF THE',         B1160...
     6   ' THE MEDIUM'/11X,1PD15.4,5X,'EQUILIBRIUM CONSTANT FOR',       B1170...
     7   ' C1 AND C2'/11X,1PD15.4,5X,'EQUILIBRIUM CONSTANT FOR',        B1180...
     8   ' C1 AND (C3 OR C4)')                                          B1190...
                                                                        B1200...
C     --- INPUT DATASET 11 ---                                          B1210...
      READ(5,300) SCALX,SCALY,SCALTH,PORFAC,STOFAC                      B1220...
  300 FORMAT(8F10.0)                                                    B1230...
C     --- SET  SPECIFIC STORATIVITY=0  FOR STEADY FLOW CASE ---         B1240...
      IF(ISSTAT.EQ.1) STOFAC=0.0D0                                      B1250...
      DO 450 I=1,NN                                                     B1260...
                                                                        B1270...
C     --- INPUT DATASET 12 ---                                          B1280...
      READ(5,400) II,X(II),Y(II),THICK(II),POR(II),STOR(II)             B1290...
  400 FORMAT(I5,5F10.0)                                                 B1300...
      X(II)=X(II)*SCALX                                                 B1310...
      Y(II)=Y(II)*SCALY                                                 B1320...
      THICK(II)=THICK(II)*SCALTH                                        B1330...
      POR(II)=POR(II)*PORFAC                                            B1340...
  450 STOR(II)=STOR(II)*STOFAC                                          B1350...
      IF(KCOORD.EQ.-1) WRITE(6,469) SCALX,SCALY,SCALTH,PORFAC,STOFAC    B1360...
  469 FORMAT(////11X,'N O D E   I N F O R M A T I O N'//16X,            B1370...
     1   'PRINTOUT OF NODE COORDINATES, THICKNESSES, POROSITIES AND ',  B1380...
     2   'STORATIVITIES CANCELLED.'//16X,'SCALE FACTORS :'/33X,1PD15.4, B1390...
     3   5X,'X-SCALE'/33X,1PD15.4,5X,'Y-SCALE'/33X,1PD15.4,5X,          B1400...
     4   'THICKNESS FACTOR'/33X,1PD15.4,5X,'POROSITY FACTOR'/           B1410...
     5   33X,1PD15.4,5X,'SPECIFIC STORATIVITY FACTOR')                  B1420...
      IF(KCOORD.NE.-1)                                                  B1430...
     1   WRITE(6,470) (I,X(I),Y(I),THICK(I),POR(I),STOR(I),I=1,NN)      B1440...
  470 FORMAT(1H1//11X,'N O D E   I N F O R M A T I O N'//13X,           B1450...
     1   'NODE',7X,'X',16X,'Y',17X,'THICKNESS',6X,'POROSITY',4X,        B1460...
     2   'SPECIFIC STORATIVITY',                                        B1470...
     3   //(11X,I6,3(3X,1PD14.5),6X,0PF8.5,10X,1PD14.5))                B1480...
                                                                        B1490...
C     --- INPUT DATASET 13 ---                                          B1500...
      READ(5,490) PMAXFA,PMINFA,ANGFAC,DSLFAC,DSTFAC                    B1510...
  490 FORMAT(2D10.0,3F10.0)                                             B1520...
      IF(KELINF.NE.-1) WRITE(6,500)                                     B1530...
  500 FORMAT(1H1//11X,'E L E M E N T   I N F O R M A T I O N'//         B1540...
     1   11X,'ELEMENT',4X,'MAXIMUM',9X,'MINIMUM',12X,                   B1550...
     2   'ANGLE BETWEEN',3X,'LONGITUDINAL',5X,'TRANSVERSE'/             B1560...
     3   22X,'HYDRAULIC',7X,'HYDRAULIC',11X,'+X-DIRECTION',             B1570...
     4   3X,'DISPERSIVITY',3X,'DISPERSIVITY'/                           B1580...
     5   22X,'CONDUCTIVITY',4X,'CONDUCTIVITY',                          B1590...
     6   9X,'AND MAXIMUM'/58X,'CONDUCTIVITY'/58X,'(IN DEGREES)'//)
      DO 550 LL=1,NE

C     --- INPUT DATASET 14 ---
      READ(5,510) L,PMAX,PMIN,ANGLEX,DISPL(L),DISPT(L)
  510 FORMAT(I10,3D10.0,2F10.0)
      PMAX=PMAX*PMAXFA
      PMIN=PMIN*PMINFA
      ANGLEX=ANGLEX*ANGFAC
      DISPL(L)=DISPL(L)*DSLFAC
      DISPT(L)=DISPT(L)*DSTFAC
      IF(KELINF.NE.-1) WRITE(6,520) L,PMAX,PMIN,ANGLEX,
     1   DISPL(L),DISPT(L)
  520 FORMAT(11X,I7,2X,2(1PD14.5,2X),8X,3(0PF10.3,5X))

C     --- ROTATION FROM MAXIMUM/MINIMUM TO X/Y DIRECTIONS ---
      RADIAX=1.745329D-2*ANGLEX
      SINA=DSIN(RADIAX)
      COSA=DCOS(RADIAX)
      SINA2=SINA*SINA
      COSA2=COSA*COSA
      PERMXX(L)=PMAX*COSA2+PMIN*SINA2
      PERMYY(L)=PMAX*SINA2+PMIN*COSA2
      PERMXY(L)=(PMAX-PMIN)*SINA*COSA
      PERMYX(L)=PERMXY(L)
  550 CONTINUE
      IF(KELINF.EQ.-1) WRITE(6,569) PMAXFA,PMINFA,ANGFAC,DSLFAC,DSTFAC
  569 FORMAT(////11X,'E L E M E N T   I N F O R M A T I O N'//
     1   16X,'PRINTOUT OF ELEMENT HYDRAULIC CONDUCTIVITIES ',
     2   'AND DISPERSIVITIES CANCELLED.'//16X,'SCALE FACTORS :'/33X,
     3   1PD15.4,5X,'MAXIMUM CONDUCTIVITY FACTOR'/33X,1PD15.4,5X,
     4   'MINIMUM CONDUCTIVITY FACTOR'/33X,1PD15.4,5X,
     5   'ANGLE FROM +X TO MAXIMUM DIRECTION FACTOR'/33X,1PD15.4,5X,
     6   'LONGITUDINAL DISPERSIVITY FACTOR'/
     7   33X,1PD15.4,5X,'TRANSVERSE DISPERSIVITY FACTOR')


      IF(INSTOP.EQ.0) GOTO 1000
      WRITE(6,999)
  999 FORMAT(////////11X,'PLEASE CORRECT INPUT DATA AND RERUN.',11X,
     1   11X,'S I M U L A T I O N   H A L T E D'/
     2   11X,'*******************   ***********')
      STOP


 1000 RETURN
      END
```

```
C     SUBROUTINE           P  L  O  T          SATRA-CHEM - VERSION 0684 C10.....
      SUBROUTINE PLOT (ICALL,NP,X,Y,CC,INDEX,XX,YY,CVEC)                 C20.....
                                                                         C30.....
C     FUNCTION:                                                          C40.....
C      TO READ PLOT SET-UP DATA, AND TO PLOT THE FINITE ELEMENT          C50.....
C      MESH, THE HEAD AND/OR CONCENTRATION SOLUTION ON THE PRINTED       C60.....
C      OUTPUT PAGE.                                                      C70.....
                                                                         C80.....
C     ------------------------------------------------------------------ C90.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                C100....
      COMMON/KPRINT/ KCOORD,KELINF,KINCID,KPLOTP,KPLOTU,KVEL             C110....
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,          C120....
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH                 C130....
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NPBC,NC1BC,NC2BC,       C140....
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT                                     C150....
                                                                         C160....
      CHARACTER*1 PRNT(122),SYM(17),BLANK(60)                            C170....
      DOUBLE PRECISION NX(500),NY(140)                                   C180....
      CHARACTER*4 DIGIT(82),VF1(6),VF2(6),VF3(7)                         C190....
      CHARACTER*30 TITLE(1,5)                                            C200....
      DIMENSION K(10),N(10)                                              C210....
      DIMENSION X(NN),Y(NN),CC(NN),XX(NN),YY(NN),INDEX(NN),CVEC(NN)      C220....
      DATA SYM/'1','2','3','4','5','6','7','8','9','0',' ','.','Y','*', C230....
     1'[','-','+'/,PRNT/122*' '/,BLANK/60*' '/,NDS/1/                    C240....
      DATA DIGIT/'1','2','3','4','5','6','7','8','9','10','11','12','13'C250....
     1,'14','15','16','17','18','19','20','21','22','23','24','25','26' C260....
     2,'27','28','29','30','31','32','33','34','35','36','37','38','39',C270....
     2'40',      '41','82','83','84','85','86','87','88','89','90','91',C280....
     1'92','93','94','95','96','97','98','99','100','101','102','103', 'C290....
     2104','105','106','107','108','109','110','111','112','113','114','C300....
     3115','116','117','118','119','120','121','122'/                    C310....
      DATA TITLE/' * * * *  N O D E S  * * * *  ',                       C320....
     1   ' * * * *  HEAD/HBASE  * * * * ',                               C330....
     2   '* * C1 CONCENTRATION/CBASE * *',                               C340....
     3   '* * C2 CONCENTRATION/CBASE * *',                               C350....
     4   '* * C4 CONCENTRATION/CBASE * *'/                               C360....
      DATA VF1/'(1H ',',',','    ','A1,F','10.2',')'/                    C370....
      DATA VF2/'(1H ',',',','    ','A1,1','X,A8',')'/                    C380....
      DATA VF3/'(1H0',',',','    ','A1,F','3.0,','12F1','0.2)'/          C390....
C                                                                        C400....
C                                                                        C410....
      IF(ICALL) 1100,1100,1                                              C420....
C     ---READ PLOT SETUP DATA---                                         C430....
 1100 READ(5,1200) IDIREC,NLINPI,NCHAPI,NCHAPL                           C440....
 1200 FORMAT(4I5)                                                        C450....
      PLTWID=(DBLE(NCHAPL)-13.0D0)/DBLE(NCHAPI)                          C460....
      N1=NLINPI                                                          C470....
      N2=NCHAPI                                                          C480....
      N3=NCHAPL                                                          C490....
      XN1=1.D0/(2.D0*N1)                                                 C500....
      NXS=1                                                              C510....
      NYS=1                                                              C520....
      NINY=PLTWID                                                        C530....
      K(1)=NN                                                           C540....
C                                                                       C550....
      IF(KPLOTP.NE.1) GOTO 1400                                         C560....
      READ(5,1300) PBASE,PDIGIT                                         C570....
 1300 FORMAT(6D13.0)                                                    C580....
 1400 IF(KPLOTU.NE.1) GOTO 1500                                         C590....
      READ(5,1300) C1BASE,C1DGIT,C2BASE,C2DGIT,C4BASE,C4DGIT            C600....
 1500 CONTINUE                                                          C610....
      WRITE(6,1520) IDIREC,NLINPI,NCHAPI,NCHAPL                         C620....
 1520 FORMAT(////11X,'P L O T   I N F O R M A T I O N'//                C630....
     1   11X,'PLOT ORIENTATION'/                                        C640....
     2   I15,5X,'IDIREC....=-1 SMALL PLOT ACROSS PAPER, =+1 ',          C650....
     3   'LARGE PLOT ALONG PAPER'//11X,'LINE PRINTER CHARACTERISTICS'/  C660....
     4   11X,I15,5X,'NUMBER OF OUTPUT ',                                C670....
     5   'LINES PER INCH'/11X,I15,5X,'NUMBER OF OUTPUT CHARACTERS',     C680....
     6   ' PER INCH'/11X,I15,5X,'MAXIMUM NUMBER OF OUTPUT ',            C690....
     7   'CHARACTERS PER LINE')                                         C700....
      IF(KPLOTP.NE.1) GOTO 1540                                         C710....
      WRITE(6,1530) PBASE,PDIGIT                                        C720....
 1530 FORMAT(//11X,'HYDRAULIC HEAD PLOT DATA'/11X,1PD15.4,5X,           C730....
     1   'HBASE....PLOTTED HEAD VALUE IS HEAD/HBASE'/                   C740....
     2   11X,1PD15.4,5X,'HDIGIT....SIGNIFICANT DIGIT MULTIPLIER',       C750....
     3   ' FOR PLOTTED HEAD VALUES')                                    C760....
 1540 IF(KPLOTU.NE.1) GOTO 1580                                         C770....
      WRITE(6,1555) C1BASE,C1DGIT,C2BASE,C2DGIT,C4BASE,C4DGIT           C780....
 1555 FORMAT(//11X,'CONCENTRATION PLOT DATA'/11X,1PD15.4,5X,            C790....
     1   'C1BASE....PLOTTED CONCENTRATION VALUE IS CONCENTRATION/',     C800....
     2   'C1BASE'/                                                      C810....
     3   11X,1PD15.4,5X,'C1DGIT....SIGNIFICANT DIGIT MULTIPLIER',       C820....
     4   ' FOR PLOTTED CONCENTRATION VALUES'/11X,1PD15.4,5X,            C830....
     5   'C2BASE....PLOTTED CONCENTRATION VALUE IS CONCENTRATION/',     C840....
     6   'C2BASE'/11X,1PD15.4,5X,'C2DGIT....SIGNIFICANT DIGIT',         C850....
     7   ' MULTIPLIER FOR PLOTTED CONCENTRATION VALUES'/11X,1PD15.4,    C860....
     8   5X,'C4BASE....PLOTTED CONCENTRATION VALUE IS CONCENTRATION/',  C870....
     9   'C4BASE'/11X,1PD15.4,5X,'C4DGIT....SIGNIFICANT DIGIT',         C880....
     *   ' MULTIPLIER FOR PLOTTED CONCENTRATION VALUES')                C890....
 1580 WRITE(6,1590)                                                     C900....
 1590 FORMAT(//31X,'THE THREE DIGITS PLOTTED ARE THE ONE TO THE LEFT,', C910....
     1   /31X,'AND THE TWO TO THE RIGHT OF THE DECIMAL POINT',          C920....
     2   /31X,'AFTER THE VALUE TO BE PLOTTED HAS BEEN MULTIPLIED BY',   C930....
     3   /31X,'THE -SIGNIFICANT DIGIT MULTIPLIER-.')                    C940....
                                                                        C950....
C --- SET LONGER PLOT AXIS DOWN (IDIREC=+1) OR ACROSS PAPER (IDIREC=-1) C960....
      SMALLX=0.0D0                                                      C970....
      SMALLY=0.0D0                                                      C980....
      BIGX=0.0D0                                                        C990....
      BIGY=0.0D0                                                        C1000...
      DO 1600 I=1,NN                                                    C1010...
      IF(X(I).GT.BIGX) BIGX=X(I)                                        C1020...
      IF(X(I).LT.SMALLX) SMALLX=X(I)                                    C1030...
      IF(Y(I).GT.BIGY) BIGY=Y(I)                                        C1040...
 1600 IF(Y(I).LT.SMALLY) SMALLY=Y(I)                                    C1050...
      XRANGE=BIGX-SMALLX                                                C1060...
      YRANGE=BIGY-SMALLY                                                C1070...
      TENTHX=XRANGE/10.0D0                                              C1080...
      TENTHY=YRANGE/10.0D0                                              C1090...
      IF(XRANGE.GE.YRANGE.AND.IDIREC.NE.-1) KKKKK=+1                    C1100...
      IF(XRANGE.GE.YRANGE.AND.IDIREC.EQ.-1) KKKKK=-1                    C1110...
      IF(XRANGE.LT.YRANGE.AND.IDIREC.NE.-1) KKKKK=-1                    C1120...
      IF(XRANGE.LT.YRANGE.AND.IDIREC.EQ.-1) KKKKK=+1                    C1130...
      IF(KKKKK.EQ.-1) GOTO 344                                          C1140...
      XMIN=SMALLX-TENTHX                                                C1150...
      XMAX=BIGX+TENTHX                                                  C1160...
      YMIN=SMALLY-TENTHY                                                C1170...
C     YMAX=BIGY+TENTHY                                                  C1180...
      GOTO 345                                                          C1190...
  344 XMIN=SMALLY-TENTHY                                                C1200...
      XMAX=BIGY+TENTHY                                                  C1210...
      YMIN=SMALLX-TENTHX                                                C1220...
C     YMAX=BIGX+TENTHX                                                  C1230...
  345 CONTINUE                                                          C1240...
      XRANGE=XRANGE*1.20D0                                              C1250...
      YRANGE=YRANGE*1.20D0                                              C1260...
      IF(KKKKK.EQ.+1) NINX=(NINY/YRANGE)*XRANGE+0.50D0                  C1270...
      IF(KKKKK.EQ.-1) NINX=(NINY/XRANGE)*YRANGE+0.50D0                  C1280...
C                                                                       C1290...
C     INITIALIZE PLOT COORDINATES...ROTATE IF REQUIRED (WHEN KKKKK=-1)  C1300...
C     (NOTE: YY PLOTS ACROSS PAGE, XX PLOTS ALONG PAGE)                 C1310...
      IF (KKKKK.EQ.-1) GOTO 361                                         C1320...
      DO 362 I=1,NN                                                     C1330...
      XX(I)= X(I)                                                       C1340...
      YY(I)= Y(I)                                                       C1350...
  362 INDEX(I)= I                                                       C1360...
      GOTO 368                                                          C1370...
  361 DO 363 I=1,NN                                                     C1380...
      XX(I)=+Y(I)                                                       C1390...
      YY(I)=+X(I)                                                       C1400...
C     NOTE THAT THE SIGN OF YY IS REVERSED LATER                        C1410...
C     IN ORDER TO COMPLETE THE ROTATION                                 C1420...
  363 INDEX(I)= I                                                       C1430...
  368 CONTINUE                                                          C1440...
C                                                                       C1450...
C     ---INITIALIZE VARIABLES---                                        C1460...
      NXD=NXS*NINX                                                      C1470...
      NYD=NYS*NINY                                                      C1480...
      IF(NXD.GE.((NYD+1)/2)) GOTO 11                                    C1490...
      NINX=1+((NYD-1)/(2*NXS))                                          C1500...
      NXD=NXS*NINX                                                      C1510...
   11 XSF=XRANGE/NXD                                                    C1520...
      YSF=YRANGE/NYD                                                    C1530...
      IF(KKKKK.EQ.+1) GOTO 12                                           C1540...
      XSF=YRANGE/NXD                                                    C1550...
      YSF=XRANGE/NYD                                                    C1560...
   12 CONTINUE                                                          C1570...
      N4=NXD*N1+1                                                       C1580...
      N5=NXD+1                                                          C1590...
      N6=NYD+1                                                          C1600...
      N7=N1*NINX                                                        C1610...
      N8=N2*NYD+1                                                       C1620...
      N9=N2*NINY                                                        C1630...
      NR=N8-1                                                           C1640...
      NA=N4/2-2                                                         C1650...
      NBB=N4/2+4                                                        C1660...
      NC=(N3-N8-10)/2                                                   C1670...
      ND=NC+N8                                                          C1680...
      NEE=MAX0(N5,N6)                                                   C1690...
      VF1(3)=DIGIT(ND-40)                                               C1700...
      VF2(3)=DIGIT(ND-40)                                               C1710...
      VF3(3)=DIGIT(NC)                                                  C1720...
C     ---ARRANGE EACH DATA SET IN DESCENDING VALUES OF X---             C1730...
      DO 90 L=1,NDS                                                     C1740...
      NNN=K(L)                                                          C1750...
      DO 30 I=1,NNN                                                     C1760...
      BIG=XX(I)                                                         C1770...
      KK=I                                                              C1780...
      DO 20 J=I,NNN                                                     C1790...
      IF(XX(J).GT.BIG) GO TO 15                                         C1800...
      GO TO 20                                                          C1810...
   15 BIG=XX(J)                                                         C1820...
      KK=J                                                              C1830...
   20 CONTINUE                                                          C1840...
      TEMPI=YY(I)                                                       C1850...
      TEMPII=XX(I)                                                      C1860...
      TEMP3=INDEX(I)                                                    C1870...
      YY(I)=YY(KK)                                                      C1880...
      XX(I)=XX(KK)                                                      C1890...
      INDEX(I)=INDEX(KK)                                                C1900...
      INDEX(KK)=TEMP3                                                   C1910...
      YY(KK)=TEMPI                                                      C1920...
   30 XX(KK)=TEMPII                                                     C1930...
   90 CONTINUE                                                          C1940...
C                                                                       C1950...
C     ---COMPUTE NUMBERS FOR X AND Y AXES---                            C1960...
      DO 100 I=1,NEE                                                    C1970...
      NNX=N5-I                                                          C1980...
      NNY=N6-I                                                          C1990...
      IF(NNY.LT.0) GO TO 95                                             C2000...
      NY(I)=YSF*NNY+YMIN                                                C2010...
      IF(KKKKK.EQ.-1)   NY(I)=YMIN+(I-1)*YSF                            C2020...
   95 IF(NNX.LT.0) GO TO 100                                            C2030...
      NX(I)=XSF*NNX+XMIN                                                C2040...
  100 CONTINUE                                                          C2050...
C                                                                       C2060...
C     ---SET UP PLOT OF MESH---                                         C2070...
      DO 105 I=1,NN                                                     C2080...
  105 CVEC(I)=I*00.0100D0                                               C2090...
C                                                                       C2100...
C                                                                       C2110...
C                                                                       C2120...
```

```
C
C      ---ENTRY FOR HEAD AND CONCENTRATION PLOTS---
C      -----------
    1 CONTINUE
C      -----------
C       ---NORMALIZE VARIABLE TO BE PLOTTED---
      CCNORM=1.0D0
      IF(NP.EQ.2) CCNORM=PBASE/PDIGIT
      IF(NP.EQ.3) CCNORM=C1BASE/C1DGIT
      IF(NP.EQ.4) CCNORM=C2BASE/C2DGIT
      IF(NP.EQ.5) CCNORM=C4BASE/C4DGIT
      DO 2 I=1,NN
    2 CC(I)=CVEC(INDEX(I))/CCNORM
C
C      ---INITIALIZE VARIABLES---
      Z=XMAX
      IF(NP.NE.4.AND.NP.NE.5) WRITE (6,40)
      DO 10 I=1,NDS
   10 N(I)=1
      DO 210 I=1,N4
C
C      ---LOCATE X AXES---
      IF (I.EQ.1.OR.I.EQ.N4) GO TO 110
      DO 114 J=1,N8,N9
  114 PRNT(J)=SYM(15)
C
C      ---LOCATE Y AXES---
      IF ((I-1)/N1*N1.NE.I-1) GO TO 117
  115 PRNT(1)=SYM(14)
      PRNT( N8)=SYM(14)
  117 IF((I-1)/N7*N7.NE.I-1) GO TO 130
      DO 118 J=2,NR
      IF((J-1)/N9*N9.EQ.J-1)PRNT(J)=SYM(17)
  118 IF((J-1)/N9*N9.NE.J-1)PRNT(J)=SYM(16)
      GO TO 130
  110 DO 120 J=1,N8
      IF ((J-1)/N2*N2.EQ.J-1) PRNT(J)=SYM(14)
  120 IF ((J-1)/N2*N2.NE.J-1) PRNT(J)=SYM(16)
C
C      ---COMPUTE LOCATION OF POINTS---
  130 DO 150 J=1,NDS
  135 IF (N(J).EQ.K(J)+1) GO TO 150
      IF(I.GT.1) GO TO 137
      IF(XX(N(J)).LE.Z+XN1*XSF) GO TO 137
      N(J)=N(J)+1
      GO TO 135
  137 IF (XX(N(J)).LE.Z+XN1*XSF.AND.XX(N(J)).GE.Z-XN1*XSF) GO TO 140
      GO TO 150
C 140 M=NR+0.5D0- ((YY(N(J))-YMIN)*N2)/YSF
  140 DELYC=       ((YY(N(J))-YMIN)*N2)/YSF
      M=NR+0.5D0 - DELYC
C
C      REVERSE SIGN OF YY (I.E. REVERSE PLOTTING DIRECTION) IF
```

```
C     GRAPH IS TO BE TRANSPOSED....
      IF(KKKKK.EQ.-1)  M=0.5D0  + DELYC
C
      IF(M.LT.0.OR.M.GT.NR) GO TO 145
      IF(CC(N(J)))142,146,147
  142 IF(M.NE.0) PRNT(M)=SYM(16)
      NUM=(-CC(N(J))+.005D0)*10.D0
      GO TO 141
  147 NUM=(CC(N(J))+0.005D0)*100.D0
      IF (NUM.GT.999) NUM=MOD(NUM,1000)
  141 IF(NUM.LT.100) GO TO 143
      INDX3=NUM/100
      IF (M.NE.0.AND.CC(N(J)).GT.0.) PRNT(M)=SYM(INDX3)
      NUM=NUM-INDX3*100
  143 INDX1=MOD(NUM,10)
      IF(INDX1.EQ.0) INDX1=10
      INDX2=NUM/10
      IF(INDX2.EQ.0) INDX2=10
      GO TO 144
  146 INDX1=14
      INDX2=14
  144 PRNT(M+1)=SYM(INDX2)
      PRNT(M+2)=SYM(INDX1)
  145 N(J)=N(J)+1
      IF (N(J).EQ.K(J)+1) GO TO 150
      IF (XX(N(J)).LE.Z+XN1*XSF.AND.XX(N(J)).GE.Z-XN1*XSF) GO TO 140
  150 CONTINUE
C
C     ---PRINT AXES,LABELS, AND POINTS---
C     IF (I-NA.EQ.0) GO TO 170
C     IF (I-NBB.EQ.0) GO TO 180
      IF ((I-1)/N1*N1-(I-1)) 190,160,190
  160 WRITE (6,VF1)(BLANK(J),J=1,NC),(PRNT(J),J=1,N8),NX(1+(I-1)/N1)
      GO TO 200
C 170 WRITE (6,VF2)(BLANK(J),J=1,NC),(PRNT(J),J=1,N8)
C     GO TO 200
C 180 WRITE (6,VF2)(BLANK(J),J=1,NC),(PRNT(J),J=1,N8)
C     GO TO 200
  190 WRITE (6,VF2)(BLANK(J),J=1,NC),(PRNT(J),J=1,N8)
C
C     ---COMPUTE NEW VALUE FOR Z AND INITIALIZE PRNT---
  200 Z=Z-2.D0*XN1*XSF
      DO 210 J=1,N8
  210 PRNT(J)=SYM(11)
C
C     ---NUMBER AND LABEL Y AXIS AND PRINT TITLE---
      WRITE (6,VF3)(BLANK(J),J=1,NC),(NY(I),I=1,N6)
      WRITE (6,80)  (TITLE(1,NP))
C
      RETURN
C
C     ---FORMATS---
   40 FORMAT ('1')
   80 FORMAT ('0',41X,1A30)
      END
```

```
C     SUBROUTINE        S O U R C E     SATRA-CHEM - VERSION 0684
      SUBROUTINE SOURCE(UIN,VIN,WIN,IQSOT,IQSO,QIN)

C     FUNCTION:
C      READ AND ORGANIZE SOURCE DATA FOR FLUID AND SOLUTE MASSES

C     ----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NPBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/CHEM/ EQCSTU,EQCSTV

      DIMENSION UIN(NN),VIN(NN),WIN(NN),QIN(NN),IQSO(NSO)

C     ----------------------------------------------------------------
      IQSOT=1
      NSOI=NSO-1
      IF (IEXCH) 50,50,200
   50 WRITE(6,100)
  100 FORMAT(////////11X,'S O U R C E   D A T A'////11X,'**** ',
     1   'NODES AT WHICH INFLOWS OR OUTFLOWS ARE SPECIFIED ****'//15X,
     2   'NODE NUMBER',10X,'FLUID INFLOW(+)/OUTFLOW(-)',22X,
     3   'CONCENTRATION OF'/12X,'(MINUS INDICATES',10X,
     4   '(VOLUME/SECOND)',31X,'INFLOWING FLUID'/14X,'TIME-VARYING',
     5   /17X,'FLOW OR',54X,'C1',12X,'C2',12X,'C4'/14X,
     6   'CONCENTRATION)'/)
      GO TO 300
  200 WRITE(6,250)
  250 FORMAT(////////11X,'S O U R C E   D A T A'////11X,'**** ',
     1   'NODES AT WHICH INFLOWS OR OUTFLOWS ARE SPECIFIED ****'//15X,
     2   'NODE NUMBER',10X,'FLUID INFLOW(+)/OUTFLOW(-)',22X,
     3   'CONCENTRATION OF'/12X,'(MINUS INDICATES',10X,
     4   '(VOLUME/SECOND)',31X,'INFLOWING FLUID'/14X,'TIME-VARYING',
     5   /17X,'FLOW OR',54X,'C1',12X,'C2',12X,'C3'/14X,
     6   'CONCENTRATION)'/)
  300 CONTINUE

      DO 500 I=1,NSOI
C     --- INPUT DATASET 16 ---
      READ(5,400)IQC,QINC,C1INC,C2INC,C4INC
  400 FORMAT(I5,4G9.0)
      IQSO(I)=IQC
      IF(IQSO(I).NE.0) NC=NC+1
      IF (IQC.LT.0) IQSOT=-1
      IC=IABS(IQC)
      QIN(IC)=QINC

C     --- CALCULATE TOTAL DISSOLVED SPECIES CONCENTRATIONS ---
      IF (IEXCH) 420,420,410
  410 UIN(IC)=C2INC+EQCSTU*C1INC*C2INC
      WIN(IC)=C1INC+EQCSTU*C1INC*C2INC
      VIN(IC)=C4INC+WIN(IC)
      GO TO 430
  420 UIN(IC)=C2INC+EQCSTU*C1INC*C2INC
      VIN(IC)=C4INC+EQCSTV*C1INC*C4INC
      WIN(IC)=C1INC+(EQCSTU*C1INC*C2INC)+(EQCSTV*C1INC*C4INC)
  430 CONTINUE


      WRITE(6,450) IQC,QINC,C1INC,C2INC,C4INC
  450 FORMAT( 11X,I10,15X,1PD15.4,15X,1PD15.4,1PD15.4,1PD15.4)
  500 CONTINUE

      IF(IQSOT.EQ.-1) WRITE(6,7000)
 7000 FORMAT(////11X,'THE SPECIFIED TIME VARIATIONS ARE PROGRAMMED ',
     1  'IN SUBROUTINE  B C T I M E .')


      RETURN
      END
```

```
C     SUBROUTINE          B O U N D       SATRA-CHEM - VERSION 0684
      SUBROUTINE BOUND(IHBC,HBC,IC1BC,IC2BC,IC4BC,UBC,VBC,WBC,C1BC,C2BC,
     1   C4BC,IHBCT,IUBCT)

C     FUNCTION:
C      READ AND ORGANIZE SPECIFIED HEAD AND CONCENTRATION DATA

C     -----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/CHEM/ EQCSTU,EQCSTV

      DIMENSION IHBC(NBCN),HBC(NBCN),IC1BC(NBCN),IC2BC(NBCN),IC4BC(NBCN)
      DIMENSION UBC(NBCN),VBC(NBCN),WBC(NBCN)
      DIMENSION C1BC(NBCN),C2BC(NBCN),C4BC(NBCN)
C     -----------------------------------------------------------------
      IHBCT=1
      IUBCT=1
      ISTOPP=0
      ISTOPU=0
      IPU=0
      WRITE(6,50)
   50 FORMAT(////////11X,'B O U N D A R Y   C O N D I T I O N S')
      IF(NHBC.EQ.0) GOTO 400
      WRITE(6,100)
  100 FORMAT(//11X,'**** NODES AT WHICH HYDRAULIC HEADS ARE',
     1   ' SPECIFIED ****'/)
      IF(IEXCH) 101,101,106
  101 WRITE(6,105)
  105 FORMAT(11X,'     AS WELL AS INFLOWING FLUID SOLUTE ',
     1   'CONCENTRATIONS'/16X,'WHEN FLUID INFLOW OCCURS      '//12X,
     2   'NODE',4X,'HYDRAULIC HEAD',5X,'C1 CONCENTRATION',5X,
     3   'C2 CONCENTRATION',5X,'C4 CONCENTRATION'//)
      GO TO 110
  106 WRITE(6,108)
  108 FORMAT(11X,'     AS WELL AS INFLOWING FLUID SOLUTE ',
     1   'CONCENTRATIONS'/16X,'WHEN FLUID INFLOW OCCURS      '//12X,
     2   'NODE',4X,'HYDRAULIC HEAD',5X,'C1 CONCENTRATION',5X,
     3   'C2 CONCENTRATION',5X,'C3 CONCENTRATION'//)
  110 CONTINUE

  120 IPU=IPU+1

C     --- INPUT DATASET 17: DATA FOR SPECIFIED HYDRAULIC HEADS ---
      READ(5,150) IHBC(IPU),HBC(IPU),C1BC(IPU),C2BC(IPU),C4BC(IPU)
  150 FORMAT(I6,4D13.0)
      IF(IHBC(IPU).LT.0) IHBCT=-1
      IF(IHBC(IPU).EQ.0) GOTO 180
      WRITE(6,160) IHBC(IPU),HBC(IPU),C1BC(IPU),C2BC(IPU),C4BC(IPU)
  160 FORMAT(10X,I5,6X,1PD12.5,7X,1PD12.5,10X,1PD12.5,9X,1PD12.5)
```

```
      IF(IEXCH) 170,170,165                                             E540....
  165 UBC(IPU)=C2BC(IPU)+EQCSTU*C1BC(IPU)*C2BC(IPU)                     E550....
      WBC(IPU)=C1BC(IPU)+EQCSTU*C1BC(IPU)*C2BC(IPU)                     E560....
      VBC(IPU)=C4BC(IPU)+WBC(IPU)                                       E570....
      GOTO 120                                                          E580....
  170 UBC(IPU)=C2BC(IPU)+EQCSTU*C1BC(IPU)*C2BC(IPU)                     E590....
      VBC(IPU)=C4BC(IPU)+EQCSTV*C1BC(IPU)*C4BC(IPU)                     E600....
      WBC(IPU)=C1BC(IPU)+EQCSTU*C1BC(IPU)*C2BC(IPU)+EQCSTV*C1BC(IPU)*   E610....
     1         C4BC(IPU)                                                E620....
                                                                        E630....
      GOTO 120                                                          E640....
                                                                        E650....
  180 IPU=IPU-1                                                         E660....
      IP=IPU                                                            E670....
      IF(IP.EQ.NHBC) GOTO 200                                           E680....
      ISTOPP=1                                                          E690....
  200 IF(IHBCT.NE.-1) GOTO 400                                          E700....
      WRITE(6,206)                                                      E710....
  206 FORMAT(//11X,'TIME-DEPENDENT SPECIFIED HEAD OR INFLOW ',          E720....
     1   'CONCENTRATION INDICATED BY NEGATIVE NODE NUMBER')             E730....
  400 IF(NC1BC.EQ.0.AND.NC2BC.EQ.0.AND.NC4BC.EQ.0) GOTO 2000            E740....
                                                                        E750....
      IF(IEXCH) 500,500,1005                                            E760....
  500 WRITE(6,1000)                                                     E770....
 1000 FORMAT(////11X,'**** NODES AT WHICH SOLUTE CONCENTRATIONS ARE ',  E780....
     1   'SPECIFIED TO BE INDEPENDENT OF LOCAL FLOWS AND FLUID SOURCES',E790....
     2   ' ****'//12X,'NODE',5X,'C1 CONCENTRATION',10X,'NODE',5X,       E800....
     3   'C2 CONCENTRATION',10X,'NODE',5X,'C4 CONCENTRATION'//)         E810....
      GOTO 1100                                                         E820....
 1005 WRITE(6,1010)                                                     E830....
 1010 FORMAT(////11X,'**** NODES AT WHICH SOLUTE CONCENTRATIONS ARE ',  E840....
     1   'SPECIFIED TO BE INDEPENDENT OF LOCAL FLOWS AND FLUID SOURCES',E850....
     2   ' ****'//12X,'NODE',5X,'C1 CONCENTRATION',10X,'NODE',5X,       E860....
     3   'C2 CONCENTRATION',10X,'NODE',5X,'C3 CONCENTRATION'//)         E870....
 1100 CONTINUE                                                          E880....
                                                                        E890....
 1120 IPU=IPU+1                                                         E900....
                                                                        E910....
C     --- INPUT DATASET 18: DATA FOR SPECIFIED CONCENTRATION NODES ---  E920....
      READ(5,1125) IC1BC(IPU),IC2BC(IPU),IC4BC(IPU),C1BC(IPU),C2BC(IPU),E930....
     1   C4BC(IPU)                                                      E940....
 1125 FORMAT(3I6,3E13.0)                                                E950....
                                                                        E960....
      IF(IC1BC(IPU).LT.0.OR.IC2BC(IPU).LT.0.OR.IC4BC(IPU).LT.0) IUBCT=-1E970....
      IF(IC1BC(IPU).EQ.0.AND.IC2BC(IPU).EQ.0.AND.IC4BC(IPU).EQ.0)       E980....
     1   GO TO 1180                                                     E990....
      WRITE(6,1150) IC1BC(IPU),C1BC(IPU),IC2BC(IPU),C2BC(IPU),          E1000...
     1   IC4BC(IPU),C4BC(IPU)                                           E1010...
 1150 FORMAT(11X,I5,6X,1PD12.5,12X,I5,6X,1PD12.5,12X,I5,6X,1PD12.5)     E1020...
      IF(IEXCH) 1160,1160,1155                                          E1030...
 1155 UBC(IPU)=C2BC(IPU)+EQCSTU*C1BC(IPU)*C2BC(IPU)                     E1040...
      WBC(IPU)=C1BC(IPU)+EQCSTU*C1BC(IPU)*C2BC(IPU)                     E1050...
      VBC(IPU)=C4BC(IPU)+WBC(IPU)                                       E1060...
      GOTO 1120                                                         E1070...
 1160 UBC(IPU)=C2BC(IPU)+EQCSTU*C1BC(IPU)*C2BC(IPU)                     E1080...
      VBC(IPU)=C4BC(IPU)+EQCSTV*C1BC(IPU)*C4BC(IPU)                     E1090...
      WBC(IPU)=C1BC(IPU)+EQCSTU*C1BC(IPU)*C2BC(IPU)+EQCSTV*C1BC(IPU)*   E1100...
     1         C4BC(IPU)                                                E1110...
      GOTO 1120                                                         E1120...
                                                                        E1130...
 1180 BIGGER=NC1BC                                                      E1140...
      IF(NC1BC .LT. NC2BC) THEN                                         E1150...
      BIGGER=NC2BC                                                      E1160...
      IF(NC2BC .LT. NC4BC) BIGGER=NC4BC                                 E1170...
      ELSE                                                              E1180...
      IF(NC1BC .LT. NC4BC) BIGGER=NC4BC                                 E1190...
      END IF                                                            E1200...
      IPU=IPU-1                                                         E1210...
      IU=IPU-IP                                                         E1220...
      IF((NC1BC.EQ.NC2BC.AND.NC1BC.NE.0).OR.(NC1BC.EQ.NC4BC.AND.        E1230...
     1 NC1BC.NE.0).OR.(NC2BC.EQ.NC4BC.AND.NC2BC.NE.0))                  E1240...
     2 IU=IPU-IP+BIGGER                                                 E1250...
      IF((NC1BC.EQ.NC2BC.AND.NC1BC.NE.0).AND.(NC1BC.EQ.NC4BC))          E1260...
     1 IU=IPU-IP+2*BIGGER                                               E1270...
                                                                        E1280...
      IF(IU .EQ. NBCSUM) GO TO 1200                                     E1290...
      ISTOPU=1                                                          E1300...
 1200 IF(IUBCT.NE.-1) GOTO 2000                                         E1310...
      WRITE(6,1206)                                                     E1320...
 1206 FORMAT(//11X,'TIME-DEPENDENT SPECIFIED CONCENTRATION IS ',        E1330...
     1   'INDICATED BY NEGATIVE NODE NUMBER')                           E1340...
                                                                        E1350...
 2000 IF(ISTOPP.EQ.0.AND.ISTOPU.EQ.0) GOTO 6000                         E1360...
      IF(ISTOPP.EQ.1) WRITE(6,3000) IP,NHBC                             E1370...
 3000 FORMAT(////11X,'ACTUAL NUMBER OF SPECIFIED HEAD NODES',           E1380...
     1   ' READ, ',I5,', IS NOT EQUAL TO NUMBER SPECIFIED IN',          E1390...
     2   ' INPUT, ',I5)                                                 E1400...
      IF(ISTOPU.EQ.1) WRITE(6,4000) IU,NC1BC+NC2BC+NC4BC                E1410...
 4000 FORMAT(////11X,'ACTUAL NUMBER OF SPECIFIED CONCENTRATION NODES',  E1420...
     1   ' READ, ',I5,', IS NOT EQUAL TO NUMBER SPECIFIED IN',          E1430...
     2   ' INPUT, ',I5)                                                 E1440...
      WRITE(6,5000)                                                     E1450...
 5000 FORMAT(////11X,'PLEASE CORRECT DATA AND RERUN.'////////           E1460...
     1   22X,'S I M U L A T I O N   H A L T E D'/                       E1470...
     2   22X,'___________________________________')                     E1480...
      STOP                                                              E1490...
                                                                        E1500...
 6000 IF(IHBCT.EQ.-1.OR.IUBCT.EQ.-1) WRITE(6,7000)                      E1510...
 7000 FORMAT(////11X,'THE SPECIFIED TIME VARIATIONS ARE PROGRAMMED ',   E1520...
     1   'IN SUBROUTINE  B C T I M E .')                                E1530...
                                                                        E1540...
                                                                        E1550...
      RETURN                                                            E1560...
      END                                                               E1570...
```

```
C     SUBROUTINE           C  O  N  N  E  C   SATRA-CHEM - VERSION 0684
      SUBROUTINE CONNEC(IN,IPINCH)

C     FUNCTION:
C      READ, ORGANIZE AND CHECK DATA ON NODE INCIDENCES AND
C      PINCH NODE INCIDENCES.

C     -----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/ NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NPBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/KPRINT/ KCOORD,KELINF,KINCID,KPLOTP,KPLOTU,KVEL

      DIMENSION IN(NIN),IPINCH(NPINCH,3)
      DIMENSION IIN(4),IEDGE(4),IK(8)
      DATA IK/1,2,2,3,3,4,4,1/
C     -----------------------------------------------------------------

      ISTOP=0
      IPIN=0
      IF(KINCID.EQ.-1) WRITE(6,1)
    1 FORMAT(////11X,'M E S H   C O N N E C T I O N   D A T A'//
     1   16X,'PRINTOUT OF NODAL INCIDENCES AND PINCH NODE ',
     2   'CONNECTIONS CANCELLED.')
      IF(KINCID.NE.-1) WRITE(6,2)
    2 FORMAT(1H1//11X,'M E S H   C O N N E C T I O N   D A T A',
     1   ///11X,'**** NODAL INCIDENCES ****'///)

      DO 1000 L=1,NE
      DO 4 I=1,4
    4 IEDGE(I)=0
      READ(5,10) LL,(IIN(II),II=1,4)
   10 FORMAT(5I6)
      DO 5 II=1,4
      III=II+(L-1)*4
    5 IN(III)=IIN(II)
      IF(IABS(LL).EQ.L) GOTO 25
      WRITE(6,20) LL
   20 FORMAT(11X,'ELEMENT ',I6,'INCIDENCE DATA IS NOT IN NUMERICAL',
     1   ' ORDER IN THE DATA SET')
      ISTOP=ISTOP+1
   25 IF(LL.GE.0) GOTO 500

C     --- INPUT DATASET 19 AND CHECK FOR ERRORS ---
      READ(5,30) (IEDGE(I),I=1,4)
   30 FORMAT(4I6)
      DO 200 K=1,4
      I=IEDGE(K)
      IF(I) 200,200,100
  100 IPIN=IPIN+1
      IPINCH(IPIN,1)=I
      KK1=2*K-1
      KK2=KK1+1
      KKK1=IK(KK1)
      KKK2=IK(KK2)
      IPINCH(IPIN,2)=IIN(KKK1)
      IPINCH(IPIN,3)=IIN(KKK2)
  200 CONTINUE

  500 M1=(L-1)*4+1
      M4=M1+3
      IF(KINCID.EQ.-1) GOTO 1000
      WRITE(6,650) L,(IN(M),M=M1,M4)
  650 FORMAT(11X,'ELEMENT',I6,5X,' NODES AT : ',6X,'CORNERS ',
     1   5(1H*),4I6,1X,5(1H*))
      IF(LL.LT.0) WRITE(6,700)(IEDGE(M),M=1,4)
  700 FORMAT(11X,'EDGES',4I6)

 1000 CONTINUE
      IF(IPIN.EQ.0) GOTO 5000
      IF(IPIN.EQ.NPINCH-1) GOTO 1500
      WRITE(6,1450) IPIN,NPINCH
 1450 FORMAT(//////11X,'ACTUAL NUMBER OF PINCH NODES,',I4,
     1   ', DIFFERS FROM NUMBER ALLOWED AS SPECIFIED IN INPUT, ',I4//
     2   11X,'PLEASE CORRECT INPUT DATA AND/OR DIMENSIONS AND RERUN.'
     3   ////////22X,'S I M U L A T I O N   H A L T E D'/
     4          22X,'_________________________________')
      STOP

 1500 CONTINUE
      IF(KINCID.EQ.-1) GOTO 5000
      WRITE(6,3000)
 3000 FORMAT(//////11X,'**** PINCH NODE CONNECTIONS ****'//7X,
     1   'PINCH NODE',17X,'CONNECTED NODES'///)
      DO 4000 I=1,IPIN
 4000 WRITE(6,4500) (IPINCH(I,NP),NP=1,3)
 4500 FORMAT(11X,I6,10X,2I6)


 5000 RETURN
      END
```

```
C     SUBROUTINE          B  A  N  W  I  D   SATRA-CHEM - VERSION 0684 G10.....
      SUBROUTINE BANWID(IN)                                             G20.....
                                                                        G30.....
C     FUNCTION:                                                         G40.....
C      CALCULATION AND CHECKING OF THE BAND WIDTH OF THE FINITE         G50.....
C      ELEMENT MESH                                                     G60.....
                                                                        G70.....
C     ----------------------------------------------------------------- G80.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               G90.....
      COMMON/DIMS/ NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NPBC,NC1BC,NC2BC,     G100....
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT                                    G110....
      DIMENSION IN(NIN)                                                 G120....
C     ----------------------------------------------------------------- G130....
                                                                        G140....
      NBTEST=0                                                          G150....
      NDIF=0                                                            G160....
      II=0                                                              G170....
      WRITE(6,100)                                                      G180....
  100 FORMAT(////////11X,'**** MESH ANALYSIS ****'//)                   G190....
                                                                        G200....
C     --- FIND ELEMENT WITH MAXIMUM DIFFERENCE IN NODE NUMBERS ---      G210....
      DO 2000 L=1,NE                                                    G220....
      II=II+1                                                           G230....
      IELO=IN(II)                                                       G240....
      IEHI=IN(II)                                                       G250....
      DO 1000 I=2,4                                                     G260....
      II=II+1                                                           G270....
      IF(IN(II).LT.IELO) IELO=IN(II)                                    G280....
 1000 IF(IN(II).GT.IEHI) IEHI=IN(II)                                    G290....
      NDIFF=IEHI-IELO                                                   G300....
      IF(NDIFF.GT.NDIF) NDIF=NDIFF                                      G310....
      LEM=L                                                             G320....
      NB=2*NDIF+1                                                       G330....
      NBHALF=NDIF+1                                                     G340....
      IF(NB.GT.NBI) WRITE(6,1500) L,NB,NBI                              G350....
 1500 FORMAT(/11X,'ELEMENT ',I4,' HAS BANDWIDTH ',I5,                   G360....
     1   ' WHICH EXCEEDS INPUT BANDWIDTH ',I3)                          G370....
      IF(NB.GT.NBI) NBTEST=NBTEST+1                                     G380....
 2000 CONTINUE                                                          G390....
                                                                        G400....
      WRITE(6,2500) NB,LEM,NBI                                          G410....
 2500 FORMAT(//11X,'ACTUAL MAXIMUM BANDWIDTH, ',I3,                     G420....
     1   ', WAS CALCULATED IN ELEMENT ',I4/11X,7(1H-),                  G430....
     2   'INPUT BANDWIDTH IS ',I3)                                      G440....
      IF(NBTEST.EQ.0) GOTO 3000                                         G450....
                                                                        G460....
      WRITE(6,2800) NBTEST                                              G470....
 2800 FORMAT(//////11X,'INPUT BANDWIDTH IS EXCEEDED IN ',I3,' ELEMENTS',G480....
     1   /11X,'PLEASE CORRECT INPUT DATA AND RERUN.',                   G490....
     2   ////////22X,'S I M U L A T I O N   H A L T E D'/,              G500....
     3           22X,'_________________________________')               G510....
      STOP                                                              G520....
                                                                        G530....
 3000 RETURN                                                            G540....
      END                                                               G550....
```

```
C     SUBROUTINE           N  C  H  E  C  K    SATRA-CHEM - VERSION 0684
      SUBROUTINE NCHECK(IPINCH,IQSO,IHBC,IC1BC,IC2BC,IC4BC)

C     FUNCTION:
C      TO CHECK THAT PINCH NODES ARE NOT ASSIGNED SPECIFIED
C      HEADS, CONCENTRATIONS OR SOURCES.

C     ----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      DIMENSION JQX(30),JPX(30),JUX(30)
      DIMENSION IPINCH(NPINCH,5),IQSO(NSO)
      DIMENSION IC1BC(NBCN),IC2BC(NBCN),IC4BC(NBCN),IHBC(NBCN)
C     ----------------------------------------------------------------

      IQX=0
      IPX=0
      IUX=0
      NPIN=NPINCH-1

C     --- MATCH PINCH NODES WITH FLUID SOURCE NODES ---
      NSOI=NSO-1
      DO 1000 I=1,NPIN
      IPIN=IPINCH(I,1)
      IF(NSOI.EQ.0) GOTO 300
      DO 200 IQ=1,NSO
      IF((IPIN-IABS(IQSO(IQ))).EQ.0) THEN
      IQX=IQX+1
      JQX(IQX)=IPI
      END IF
  200 CONTINUE

C     --- MATCH PINCH NODES WITH SPECIFIED HEAD NODES ---
  300 IF(NHBC.EQ.0) GOTO 500
      DO 400 IP=1,NHBC
      IF(IPIN-IABS(IHBC(IP))) 400,350,400
  350 IPX=IPX+1
      JPX(IPX)=IPIN
  400 CONTINUE

C     --- MATCH PINCH NODES WITH SPECIFIED CONCENTRATION NODES ---
  500 IF(NBCSUM.EQ.0) GOTO 1000
      DO 600 IU=1,NBCSUM
      IUP=IU+NHBC
      IF((IPIN-IABS(IC1BC(IUP))).EQ.0.OR.(IPIN-IABS(IC2BC(IUP)))
     1 .EQ.0.OR.(IPIN-IABS(IC4BC(IUP))).EQ.0) THEN
       IUX=IUX+1
       JUX(IUX)=IPIN
      END IF
  600 CONTINUE
```

```
1000 CONTINUE                                                          H540....
                                                                       H550....
     IF(IQX.EQ.0) GOTO 1400                                            H560....
     WRITE(6,1250) (JQX(I),I=1,IQX)                                    H570....
1250 FORMAT(/////11X,'THE FOLLOWING NODES MAY NOT BE SPECIFIED AS',    H580....
    1   ' SOURCE NODES : '/15X,2(20I6/))                               H590....
     WRITE(6,1251)                                                     H600....
1251 FORMAT(/11X,'PLEASE REDISTRIBUTE SOURCES OR CHANGE THESE PINCH',  H610....
    1   ' NODES TO NORMAL CORNER MESH NODES AND THEN RERUN.')          H620....
1400 IF(IPX.EQ.0) GOTO 1500                                            H630....
     WRITE(6,1450) (JPX(I),I=1,IPX)                                    H640....
1450 FORMAT(/////11X,'THE FOLLOWING NODES MAY NOT BE INPUT AS',        H650....
    1   ' SPECIFIED HEAD NODES : ',/15X,2(20I6/))                      H660....
     WRITE(6,1451)                                                     H670....
1451 FORMAT(/11X,'PLEASE REMOVE SPECIFIED HEAD RESTRICTION OR',        H680....
    1   ' CHANGE THESE PINCH NODES TO NORMAL CORNER MESH NODES AND',   H690....
    2   ' THEN RERUN.')                                                H700....
1500 IF(IUX.EQ.0) GOTO 1680                                            H710....
     WRITE(6,1650) (JUX(I),I=1,IUX)                                    H720....
1650 FORMAT(/////11X,'THE FOLLOWING NODES MAY NOT BE INPUT AS',        H730....
    1   ' SPECIFIED CONCENTRATION NODES : ',/15X,2(20I6/))             H740....
     WRITE(6,1651)                                                     H750....
1651 FORMAT(/11X,'PLEASE REMOVE SPECIFIED CONCENTRATION RESTRICTION ', H760....
    1   'OR CHANGE THESE PINCH NODES TO NORMAL CORNER NODES AND',      H770....
    2   ' THEN RERUN.')                                                H780....
                                                                       H790....
1680 IF(IQX+IPX+IUX) 1800,1800,1700                                    H800....
1700 WRITE(6,1750)                                                     H810....
1750 FORMAT(////////11X,'S I M U L A T I O N   H A L T E D'/,          H820....
    1               11X,'_________________________________')           H830....
     STOP                                                              H840....
                                                                       H850....
1800 RETURN                                                            H860....
     END                                                               H870....
```

```
C     SUBROUTINE          I  N  D  A  T  2    SATRA-CHEM - VERSION 0684
      SUBROUTINE INDAT2(HVEC,C1,C2,C3,C4,HM1,UM1,VM1,WM1,
     1   UVEC,VVEC,WVEC,CS1,CS3,QPL,QPLM1,UBC,VBC,WBC,C1P1)

C     FUNCTION:
C      TO READ INITIAL CONDITIONS FROM UNIT-55, AND TO
C      INITIALIZE DATA FOR EITHER WARM OR COLD START OF
C      THE SIMULATION.

C     -----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTHM1,DLTUM1,DLTPM2,DLTUM2

      DIMENSION HVEC(NN),C1(NN),C2(NN),C3(NN),C4(NN),HM1(NN),UM1(NN)
      DIMENSION VM1(NN),WM1(NN),C1P1(NN),QPL(NBCN),QPLM1(NBCN)
      DIMENSION UBC(NBCN),VBC(NBCN),WBC(NBCN),CS1(NN),CS3(NN)
      DIMENSION UVEC(NN),VVEC(NN),WVEC(NN)
C     -----------------------------------------------------------------

      IF(IREAD) 500,500,520
C     --- INPUT INITIAL CONDITIONS FOR WARM START (UNIT-55 DATA).
  500 READ(55,510) TSTART,DELTP,DELTU,DLTUM1
  510 FORMAT(4D20.10)
      READ(55,515) (HVEC(I),I=1,NN)
      READ(55,515) (C1(I),I=1,NN)
      READ(55,515) (C2(I),I=1,NN)
      READ(55,515) (C4(I),I=1,NN)
      READ(55,515) (HM1(I),I=1,NN)
      READ(55,515) (UM1(I),I=1,NN)
      READ(55,515) (VM1(I),I=1,NN)
      READ(55,515) (WM1(I),I=1,NN)
      READ(55,515) (CS1(I),I=1,NN)
      READ(55,515) (CS3(I),I=1,NN)
      READ(55,515) (QPL(IPU),IPU=1,NBCN)
      READ(55,515) (QPLM1(IPU),IPU=1,NBCN)
      READ(55,515) (UBC(IPU),IPU=1,NBCN)
      READ(55,515) (VBC(IPU),IPU=1,NBCN)
      READ(55,515) (WBC(IPU),IPU=1,NBCN)
  515 FORMAT(4D20.0)
C 515 FORMAT(6D13.0)
      IF(IEXCH.EQ.1) THEN
      DO 517 I=1,NN
  517 C3(I)=C4(I)
      END IF

C     --- DETERMINE THE TOTAL DISSOLVED CONCENTRATIONS ---
      CALL EQCHEM(NN,C1,C1P1,UVEC,VVEC,WVEC,0,C2,C3,C4,0)
      GOTO 530
```

```
C      --- INPUT INITIAL CONDITIONS FOR A COLD START (UNIT-55 DATA) ---
  520 READ(55,510) TSTART
      READ(55,515) (HVEC(I),I=1,NN)
      READ(55,515) (C1(I),I=1,NN)
      READ(55,515) (C2(I),I=1,NN)
      READ(55,515) (C4(I),I=1,NN)
      IF(IEXCH.EQ.1) THEN
      DO 525 I=1,NN
  525 C3(I)=C4(I)
      END IF

C     START-UP WITH O(DELT) TIME DERIVATIVE BY SETTING  BETA=1.0D-16
C     IN O(DELT**2) TIME DERIVATIVE FORMULA.
C     SET DLT&M1=DELT*1E6 TO GET INITIAL TIME STEP FORMS OF QUAD PROJ
      DELTP=DELT*1.D16
      DLTHM1=DELT*1.D16
      DELTU=DELT*1.D16
      DLTUM1=DELT*1.D16

C     DLTUM1 IS ARBITRARY AS U=UM1=UM2.

C     --- DETERMINE TOTAL DISSOLVED CONCENTRATIONS ---
      CALL EQCHEM(NN,C1,C1P1,UVEC,VVEC,WVEC,0,C2,C3,C4,0)
      DO 527 I=1,NN
      HM1(I)=HVEC(I)
      UM1(I)=UVEC(I)
      VM1(I)=VVEC(I)
  527 WM1(I)=WVEC(I)

      CALL ZERO(QPL,NBCN,0.0D0)
      CALL ZERO(QPLM1,NBCN,0.0D0)
  530 CONTINUE

      TSEC=TSTART


      RETURN
      END
```

```
C     SUBROUTINE           E  Q  C  H  E  M    SATRA-CHEM - VERSION 0684
      SUBROUTINE EQCHEM(NJ,C1,C1P1,U,V,W,IEQ,C2,C3,C4,INR)

C     FUNCTION:
C      TO DETERMINE THE TOTAL DISSOLVED CONCENTRATIONS FROM THE
C      INDIVIDUAL COMPONENT CONCENTRATIONS AND VICE VERSA.  THE
C      SUBROUTINE NRITER (NEWTON-RAPHSON) IS CALLED FROM THIS
C      ROUTINE.

C     -----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/CHEM/ EQCSTU,EQCSTV

      DIMENSION C1(NJ),C1P1(NJ),U(NJ),V(NJ),W(NJ),C2(NJ),C3(NJ),C4(NJ)
C     -----------------------------------------------------------------


      IF(IEQ-1) 10,30,70

C     --- CALCULATION OF U, V AND W FROM INPUT CONCENTRATIONS ---
   10 DO 20 I=1,NJ
      IF (NCONT.GT.1.AND.IEXCH.EQ.0) THEN
      U(I)=C2(I)+EQCSTU*C1(I)*C2(I)
      V(I)=C4(I)+EQCSTV*C1(I)*C4(I)
      W(I)=C1(I)+EQCSTU*C1(I)*C2(I)+EQCSTV*C1(I)*C4(I)
      ELSE IF (NCONT.GT.1.AND.IEXCH.EQ.1) THEN
      W(I)=C1(I)+EQCSTU*C1(I)*C2(I)
      U(I)=C2(I)+EQCSTU*C1(I)*C2(I)
      V(I)=C3(I)+W(I)
      ELSE
      U(I)=C2(I)
      V(I)=0.0D0
      W(I)=0.0D0
      END IF
   20 CONTINUE
      RETURN


C     --- CALCULATION OF C'S FROM U, V AND W ---
   30 IF (IEQLIB.GE.1.AND.IEXCH.EQ.0) THEN
      CALL NRITER(NJ,C1,C1P1,U,V,W,INR)
      ELSE IF (IEXCH.EQ.1.AND.IEQLIB.NE.0) THEN
      DO 35 I=1,NJ
      A=W(I)-U(I)-(1/EQCSTU)
      B=DSQRT(A*A+(4.0D0*(1/EQCSTU)*W(I)))
   35 C1(I)=.5D0*(A+B)
      ELSE
      DO 40 I=1,NJ
   40 C1(I)=W(I)
```

```
      END IF
      DO 50 I=1,NJ
      C2(I)=U(I)/(1.D0+EQCSTU*C1(I))
      IF (IEXCH.EQ.1) THEN
      C3(I)=V(I)-W(I)
      C4(I)=0.0D0
      ELSE
      C4(I)=V(I)/(1.D0+EQCSTV*C1(I))
      END IF
   50 CONTINUE

      RETURN


C     --- FOR SINGLE CONSTITUENT C2=U ---
C  70 ICOUNT=ICOUNT+1
   70 DO 80 I=1,NJ
   80  C2(I)=U(I)


      RETURN
      END
```

```
C     SUBROUTINE        N  R  I  T  E  R    SATRA-CHEM - VERSION 0684
      SUBROUTINE NRITER (NJ,C1,C1P1,U,V,W,INR)

C     FUNCTION:
C      TO SOLVE THE POLYNOMIAL FOR C1 USING NEWTON-RAPHSON ITERATIONS

C     --------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/ITERAT/ITER,ITRMAX,RPM,RPMAX,IPWORS,RWM,RWMAX,IWWORS,
     1   RTOL,MAXITR
      COMMON/CHEM/ EQCSTU,EQCSTV

      DIMENSION C1(NJ),C1P1(NJ),U(NJ),V(NJ),W(NJ)
C     --------------------------------------------------------------

      ICOUNT=1
   10 RMAX=0.0D0
      DO 20 I=1,NJ
      EQPROD=EQCSTU*EQCSTV
      EQSUM=EQCSTU+EQCSTV
      CRESID=U(I)+V(I)-W(I)
      C1SQ=C1P1(I)*C1P1(I)
      C1CUB=C1SQ*C1P1(I)
      RNUM=EQPROD*C1CUB+(EQSUM+EQPROD*CRESID)*C1SQ
     1  +(1.D0+EQCSTU*(U(I)-W(I))+EQCSTV*(V(I)-W(I)))
     2  *C1P1(I)-W(I)
      RDEN=3.*EQPROD*C1SQ+2.D0*(EQSUM+EQPROD*CRESID)*C1P1(I)
     1  +1.D0+EQCSTU*(U(I)-W(I))+EQCSTV*(V(I)-W(I))
      C1(I)=C1P1(I)-RNUM/RDEN
      ERR=ABS(C1(I)-C1P1(I))
      IF (ERR .LT. RMAX) GOTO 15
      RMAX=ERR
      IRWORS=I
   15 C1P1(I)=C1(I)
   20 CONTINUE

      IF (RMAX.GT.RTOL.AND.ICOUNT.LE.MAXITR) THEN
      ICOUNT=ICOUNT+1
      GO TO 10
      ELSE IF (ICOUNT .GT. MAXITR) THEN
      IF(INR.EQ.0) THEN
      WRITE(6,40) IRWORS,RMAX
   40 FORMAT(/////11X,'NEWTON-RAPHSON ITERATION PRIOR TO SOLVE DOES',
     1  ' NOT CONVERGE WITHIN LIMIT.'/15X,'AT NODE ',I3,', CHANGE IS ',
     2  1PD15.7////)
      ELSE
      WRITE(6,50) IRWORS,RMAX
   50 FORMAT(/////11X,'NEWTON-RAPHSON ITERATION AFTER SOLVE DOES',
     1  ' NOT CONVERGE WITHIN LIMIT.'/15X,'AT NODE ',I3,', CHANGE IS ',
     2  1PD15.7////)
      END IF
      ELSE IF (RMAX .LE. RTOL) THEN
      GO TO 70
      END IF                                                           K540....
                                                                       K550....
   70 CONTINUE                                                         K560....
                                                                       K570....
                                                                       K580....
      RETURN                                                           K590....
      END                                                              K600....
```

```
C     SUBROUTINE          P  R  I  S  O  L        SATRA2 - VERSION 0583
      SUBROUTINE PRISOL(ML,ISTOP,IGOI,HVEC,C1,C2,C3,C4,VMAG,VANG)

C     FUNCTION:
C      TO PRINT HEAD OR CONCENTRATION SOLUTIONS AND TO OUTPUT
C      INFORMATION ON TIME STEP, ITERATIONS, AND FLUID VELOCITIES.

C     -----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTPM1,DLTUM1,DLTPM2,DLTUM2
      COMMON/ITERAT/ITER,ITRMAX,RPM,RPMAX,IPWORS,RWM,RWMAX,IWWORS,
     1   RTOL,MAXITR
      COMMON/KPRINT/ KCOORD,KELINF,KINCID,KPLOTP,KPLOTU,KVEL

      DIMENSION HVEC(NN),C1(NN),C2(NN),C3(NN),C4(NN),VMAG(NE),VANG(NE)
C     -----------------------------------------------------------------

      IF(IT.GT.0.OR.ISSTAT.EQ.2) GOTO 100
      WRITE(6,60)
   60 FORMAT(1H1////11X,'I N I T I A L   C O N D I T I O N S',
     1              /11X,'___________________________________')
      IF(IREAD.EQ.-1) WRITE(6,65)
   65 FORMAT(//11X,'INITIAL CONDITIONS RETRIEVED FROM STORAGE ',
     1   'ON UNIT 55.')
      GOTO 500

C     --- OUTPUT MAJOR HEADINGS FOR CURRENT TIME STEP ---
  100 IF(IGOI.NE.0.AND.ISTOP.EQ.0) WRITE(6,150) ITER,IT
  150 FORMAT(////////11X,'ITERATION ',I3,' SOLUTION FOR TIME STEP ',I4)

      IF(ISTOP.EQ.-1) WRITE(6,250) IT,ITER
  250 FORMAT(1H1//11X,'SOLUTION FOR TIME STEP ',I4,
     1   ' NOT CONVERGED AFTER ',I3,' ITERATIONS.')

      IF(ISTOP.GE.0) WRITE(6,350) IT
  350 FORMAT(1H1//11X,'RESULTS FOR TIME STEP ',I4/
     1   11X,'_______ ___ ____ ____ ____')
      IF(ITRMAX.EQ.1) GOTO 500
      IF(ISTOP.GE.0.AND.IT.GT.0) WRITE(6,355) ITER
      IF(IT.EQ.0.AND.ISTOP.GE.0.AND.ISSTAT.EQ.2) WRITE(6,355) ITER
  355 FORMAT(11X,'(AFTER ',I3,' ITERATIONS) :')
      WRITE(6,450) RPM,IPWORS,RWM,IWWORS
  450 FORMAT(//11X,'MAXIMUM H CHANGE FROM PREVIOUS ITERATION ',
     1   1PD14.5,' AT NODE ',I5/11X,'MAXIMUM W CHANGE FROM PREVIOUS ',
     2   'ITERATION ',1PD14.5,' AT NODE ',I5)

  500 IF(IT.EQ.0.AND.ISSTAT.EQ.2) GOTO 680
      WRITE(6,550) DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,
   1   TMONTH,TYEAR
550 FORMAT(///11X,'TIME INCREMENT :',T27,1PE15.4,' SECONDS'//11X,
   1   'ELAPSED TIME :',T27,1PD15.4,' SECONDS',/T27,1PD15.4,' MINUTES'
   2   /T27,1PD15.4,' HOURS'/T27,1PD15.4,' DAYS'/T27,1PD15.4,' WEEKS'/
   3   T27,1PD15.4,' MONTHS'/T27,1PD15.4,' YEARS')

    IF(ML.EQ.2.AND.ISTOP.GE.0) GOTO 700
    IF(ISSTAT.EQ.1) GOTO 700
    WRITE(6,650) (I,HVEC(I),I=1,NN)
650 FORMAT(///11X,'H  Y  D  R  A  U  L  I  C    H  E  A  D'//
   1   7X,7('NODE',14X)/(7X,7(I4,1X,1PD12.5,1X)))
    IF(KVEL.EQ.1.AND.IT.GT.0) WRITE( 6,655) (L,VMAG(L),L=1,NE)
    IF(KVEL.EQ.1.AND.IT.GT.0) WRITE( 6,656) (L,VANG(L),L=1,NE)
655 FORMAT(///11X,'F  L  U  I  D    V  E  L  O  C  I  T  Y'//
   1   11X,'M A G N I T U D E   AT CENTROID OF ELEMENT'//
   2   4X,7('ELEMENT',11X)/(7X,7(I4,1X,1PD12.5,1X)))
656 FORMAT(///11X,'F  L  U  I  D    V  E  L  O  C  I  T  Y'//
   1   11X,'A N G L E   IN DEGREES FROM +X-AXIS TO FLOW DIRECTION ',
   2   'AT CENTROID OF ELEMENT'//
   3   4X,7('ELEMENT',11X)/(7X,7(I4,1X,1PD12.5,1X)))
    GOTO 700
680 WRITE(6,690) (I,HVEC(I),I=1,NN)
690 FORMAT(///11X,'S  T  E  A  D  Y  -  S  T  A  T  E    H  Y  D  R',
   1   '  A  U  L  I  C    H  E  A  D'//7X,7('NODE',14X)/
   2   (7X,7(I4,1X,1PD12.5,1X)))
    GOTO 800

700 IF(ML.EQ.1.AND.ISTOP.GE.0) GOTO 800
    IF(NCONT .GT. 1) THEN
720 WRITE(6,725) (I,C1(I),I=1,NN)
725 FORMAT(///11X,'C1    C  O  N  C  E  N  T  R  A  T  I  O  N'//7X,
   1   7('NODE',14X)/(7X,7(I4,1X,1PD12.5,1X)))
    WRITE(6,750) (J,C2(J),J=1,NN)
750 FORMAT(///11X,'C2    C  O  N  C  E  N  T  R  A  T  I  O  N'//7X,
   1   7('NODE',14X)/(7X,7(I4,1X,1PD12.5,1X)))
    IF(IEXCH) 755,755,765
755 WRITE(6,760) (K,C4(K),K=1,NN)
760 FORMAT(///11X,'C4    C  O  N  C  E  N  T  R  A  T  I  O  N'//7X,
   1   7('NODE',14X)/(7X,7(I4,1X,1PD12.5,1X)))
    GO TO 775
765 WRITE(6,770) (K,C3(K),K=1,NN)
770 FORMAT(///11X,'C3    C  O  N  C  E  N  T  R  A  T  I  O  N'//7X,
   1   7('NODE',14X)/(7X,7(I4,1X,1PD12.5,1X)))
775 CONTINUE
    ELSE
    WRITE(6,780) (J,C2(J),J=1,NN)
780 FORMAT(///11X,'C2    C  O  N  C  E  N  T  R  A  T  I  O  N'//7X,
   1   7('NODE',14X)/(7X,7(I4,1X,1PD12.5,1X)))
    END IF
    IF(ISSTAT.NE.2.OR.IT.NE.1.OR.KVEL.NE.1) GOTO 800
    WRITE( 6,695) (L,VMAG(L),L=1,NE)
    WRITE( 6,696) (L,VANG(L),L=1,NE)
695 FORMAT(///11X,'S  T  E  A  D  Y  -  S  T  A  T  E     ',
     1   'F L U I D     V E L O C I T Y'//
     2   11X,'M A G N I T U D E   AT CENTROID OF ELEMENT'//
     3   4X,7('ELEMENT',11X)/(7X,7(I4,1X,1PD12.5,1X)))
  696 FORMAT(///11X,'S  T  E  A  D  Y  -  S  T  A  T  E     ',
     1   'F L U I D     V E L O C I T Y'//
     2   11X,'A N G L E   IN DEGREES FROM +X-AXIS TO FLOW DIRECTION ',
     3   'AT CENTROID OF ELEMENT'//
     4   4X,7('ELEMENT',11X)/(7X,7(I4,1X,1PD12.5,1X)))
C
C
  800 RETURN
      END
```

```
C     SUBROUTINE          T  I  M  E  S      SATRA-CHEM - VERSION 0684 M10.....
      SUBROUTINE TIMES (ITM,ML,TSECP0,TSECU0)                           M20.....
                                                                        M30.....
C     FUNCTION:                                                         M40.....
C      TO INITIALIZE TIME DEPENDENT COEFFICIENTS                        M50.....
                                                                        M60.....
C     ----------------------------------------------------------------- M70.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               M80.....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        M90.....
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTPM1,DLTUM1,DLTPM2,DLTUM2          M100....
      COMMON/BFACS/ BDELP,BDELU                                         M110....
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,         M120....
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH                M130....
C     ----------------------------------------------------------------- M140....
                                                                        M150....
      IF(ITM.EQ.1) THEN                                                 M160....
      TSECP0=TSEC                                                       M170....
      TSECU0=TSEC                                                       M180....
      ELSE                                                              M190....
      TSEC=TSEC+DELT                                                    M200....
      END IF                                                            M210....
      TMIN=TSEC/60.D0                                                   M220....
      THOUR=TMIN/60.D0                                                  M230....
      TDAY=THOUR/24.D0                                                  M240....
      TWEEK=TDAY/7.D0                                                   M250....
      TMONTH=TDAY/30.4375D0                                             M260....
      TYEAR=TDAY/365.25D0                                               M270....
      IF(ISSTAT.EQ.1) THEN                                              M280....
      DLTPM1=DELTP                                                      M290....
      DLTPM2=DELTP                                                      M300....
      DLTUM1=DELTU                                                      M310....
      DLTUM2=DELTU                                                      M320....
      BDELP=0.0D0                                                       M330....
      BDELU=0.0D0                                                       M340....
      END IF                                                            M350....
      IF(ITM.EQ.1) RETURN                                               M360....
                                                                        M370....
C     --- UPDATE HEAD AND CONCENTRATION TIME INCREMENTS ---             M380....
      IF(ML-1) 1010,1020,1030                                           M390....
 1010 DLTUM2=DLTUM1                                                     M400....
      DLTUM1=DELTU                                                      M410....
      DLTPM2=DLTPM1                                                     M420....
      DLTPM1=DELTP                                                      M430....
      GOTO 1040                                                         M440....
 1020 DLTPM2=DLTPM1                                                     M450....
      DLTPM1=DELTP                                                      M460....
      GOTO 1040                                                         M470....
 1030 DLTUM2=DLTUM1                                                     M480....
      DLTUM1=DELTU                                                      M490....
 1040 CONTINUE                                                          M500....
                                                                        M510....
      DELTP=TSEC-TSECP0                                                 M520....
      DELTU=TSEC-TSECU0                                                 M530....
      BDELP=0.0D0                                                       M540....
      BDELU=0.0D0                                                       M550....
                                                                        M560....
      IF(ML-1) 1060,1070,1080                                           M570....
 1060 TSECP0=TSEC                                                       M580....
      TSECU0=TSEC                                                       M590....
      GOTO 1090                                                         M600....
 1070 TSECP0=TSEC                                                       M610....
      GOTO 1090                                                         M620....
 1080 TSECU0=TSEC                                                       M630....
 1090 CONTINUE                                                          M640....
                                                                        M650....
      RETURN                                                            M660....
      END                                                               M670....
```

```
C     SUBROUTINE        U P D A T E      SATRA-CHEM - VERSION 0684
      SUBROUTINE UPDATE (ML,HM1,UM1,VM1,WM1,QIN,QINM1,
     1   HVEC,UVEC,VVEC,WVEC,QPL,QPLM1)

C     FUNCTION:
C      TO UPDATE TIME-DEPENDENT VECTORS FOLLOWING A GIVEN TIME STEP

C     ------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT

      DIMENSION HM1(NN),UM1(NN),VM1(NN),WM1(NN)
      DIMENSION HVEC(NN),UVEC(NN),VVEC(NN),WVEC(NN)
      DIMENSION QIN(NN),QINM1(NN),QPL(NBCN),QPLM1(NBCN)
C     ------------------------------------------------------------------

      IF (ML-1) 10,70,120

C     --- UPDATING WHEN SOLVING FOR BOTH HEAD AND CONCENTRATION ---

   10 DO 20, I=1,NN
      HM1(I)=HVEC(I)
      QINM1(I)=QIN(I)
      UM1(I)=UVEC(I)
      VM1(I)=VVEC(I)
      WM1(I)=WVEC(I)
   20 CONTINUE

      DO 30, IP=1,NHBC
   30 QPLM1(IP)=QPL(IP)

      RETURN

C     --- UPDATING WHEN SOLVING FOR HEAD ONLY ---
   70 DO 80, I=1,NN
      HM1(I)=HVEC(I)
   80 QINM1(I)=QIN(I)

      DO 90, IP=1,NHBC
   90 QPLM1(IP)=QPL(IP)

      RETURN

C --- UPDATING WHEN SOLVING FOR CONCENTRATION ONLY ---
  120 DO 130, I=1,NN
      UM1(I)=UVEC(I)
      HM1(I)=HVEC(I)
      QINM1(I)=QIN(I)
      VM1(I)=VVEC(I)
      WM1(I)=WVEC(I)
  130 CONTINUE
```

```
      DO 140, IP=1,NHBC                                                 N540....
  140 QPLM1(IP)=QPL(IP)                                                 N550....
                                                                        N560....
      RETURN                                                            N570....
      END                                                               N580....
```

```
C     SUBROUTINE        A  D  S  O  R  B     SATRA-CHEM - VERSION 0684
      SUBROUTINE ADSORB(CS1,CS2,CS3,C1,C2,C4,UVEC,VVEC,
     1   UAVG,VAVG,UM1,UM2,VM1,VM2)

C     FUNCTION:
C      TO DETERMINE VALUES FOR THE APPROPRIATE COEFFICIENTS WHEN
C      SIMULATING EQUILIBRIUM SORPTION

C     ----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTPM1,DLTUM1,DLTPM2,DLTUM2
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,RHOS,DECAY,SIGMAW,FF,CBART
      COMMON/ITERAT/ITER,ITRMAX,RPM,RPMAX,IPWORS,RWM,RWMAX,IWWORS,
     1   RTOL,MAXITR
      COMMON/CHEM/ EQCSTU,EQCSTV

      DIMENSION C1(NN),C2(NN),C4(NN),UVEC(NN),VVEC(NN)
      DIMENSION CS1(NN),CS2(NN),CS3(NN),UM1(NN),UM2(NN),VM1(NN),VM2(NN)
      DIMENSION UAVG(NN),VAVG(NN)

C     ----------------------------------------------------------------

C     --- SET H AND G WHEN EQUILIBRIUM REACTIONS DO NOT OCCUR ---
      IF(IEQLIB.GE.1) GOTO 500
      DO 100 I=1,NN
      CS1(I)=FF
  100 CS3(I)=0.0D0
      RETURN

C     --- CALCULATIONS WHEN EQUILIBRIUM REACTIONS OCCUR ---
  500 DO 700 I=1,NN
      DUDT=(UVEC(I)-UM1(I))/DELTU
      DVDT=(VVEC(I)-VM1(I))/DELTU

C     --- CALCULATE ADSORPTION VARIABLES ---
      H=1.D0+EQCSTU*C2(I)+EQCSTV*C4(I)
     1  -(((EQCSTU/(1.D0+EQCSTU*C1(I)))**2)*UAVG(I)*C1(I))
     2  -(((EQCSTV/(1.D0+EQCSTV*C1(I)))**2)*VAVG(I)*C1(I))

      G=DUDT*(C1(I)*EQCSTU)/(1.D0+EQCSTU*C1(I))
     1  +DVDT*(C1(I)*EQCSTV)/(1.D0+EQCSTV*C1(I))

      CS1(I)=FF/H
      CS3(I)=-CS1(I)*G

  700 CONTINUE
      RETURN

      END
```

```
C     SUBROUTINE         X  C  H  N  G  E     SATRA-CHEM - VERSION 0684 P10.....
      SUBROUTINE XCHNGE(C1,UVEC,VVEC,UAVG,VAVG,WAVG,UM1,VM1,CS1,CS3)    P20.....
                                                                        P30.....
C     FUNCTION:                                                         P40.....
C      TO DETERMINE THE COEFFICIENTS NECESSARY FOR BINARY ION           P50.....
C      EXCHANGE SIMULATIONS                                             P60.....
                                                                        P70.....
C     ----------------------------------------------------------------- P80.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               P90.....
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,      P100....
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT                                    P110....
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,         P120....
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH                P130....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        P140....
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTPM1,DLTUM1,DLTPM2,DLTUM2          P150....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,RHOS,DECAY,SIGMAW,FF,CBART     P160....
      COMMON/CHEM/ EQCSTU,EQCSTV                                        P170....
                                                                        P180....
      DIMENSION C1(NN),UVEC(NN),VVEC(NN),CS1(NN),CS3(NN)                P190....
      DIMENSION UM1(NN),VM1(NN),UAVG(NN),VAVG(NN),WAVG(NN)              P200....
C     ----------------------------------------------------------------- P210....
                                                                        P220....
      DO 1000 I=1,NN                                                    P230....
      DUDT=(UVEC(I)-UM1(I))/DELTU                                       P240....
      DVDT=(VVEC(I)-VM1(I))/DELTU                                       P250....
      G=EQCSTV*C1(I)+(VAVG(I)-WAVG(I))                                  P260....
      F3=(EQCSTV*C1(I)*CBART)/G                                         P270....
      F1=EQCSTV*(CBART-F3)                                              P280....
                                                                        P290....
      IF (IEQLIB.GT.0) THEN                                             P300....
      A=WAVG(I)-UAVG(I)-(1.0D0/EQCSTU)                                  P310....
      B=DSQRT(A*A+(4.0D0*(1.0D0/EQCSTU)*WAVG(I)))                       P320....
      CS1(I)=(F3+(F1*((1.0D0/EQCSTU)+C1(I))/B))/G                       P330....
      CS3(I)=-((((F1*C1(I)*DUDT)/B)+(F3*DVDT))/G)                       P340....
      ELSE                                                              P350....
      CS1(I)=(F1+F3)/G                                                  P360....
      CS3(I)=-(F3*DVDT)/G                                               P370....
      END IF                                                            P380....
                                                                        P390....
 1000 CONTINUE                                                          P400....
      RETURN                                                            P410....
                                                                        P420....
      END                                                               P430....
```

```
C     SUBROUTINE          Z  E  R  O          SATRA-CHEM - VERSION 0684
      SUBROUTINE ZERO(A,IADIM,FILL)

C     FUNCTION:
C      TO INITIALIZE ANY GIVEN VECTOR OR MATRIX WITH ZEROS

C     ----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(IADIM)
C     ----------------------------------------------------------------

      DO 10 I=1,IADIM
   10 A(I)=FILL

      RETURN
      END
```

```
C     SUBROUTINE           B  C  T  I  M  E     SATRA-CHEM - VERSION 0684
      SUBROUTINE BCTIME(IHBC,HBC,IC1BC,IC2BC,IC4BC,UBC,VBC,WBC,
     1   UIN,VIN,WIN,IQSO,IHBCT,IUBCT,IQSOT,QIN)

C     FUNCTION:
C      USER-PROGRAMMED SUBROUTINE WHICH ALLOWS THE USER TO SPECIFY:
C       (1) TIME-DEPENDENT SPECIFIED PRESSURES AND TIME-DEPENDENT
C           CONCENTRATIONS OR TEMPERATURES OF INFLOWS AT THESE POINTS
C       (2) TIME-DEPENDENT SPECIFIED CONCENTRATIONS OR TEMPERATURES
C       (3) TIME-DEPENDENT FLUID SOURCES AND CONCENTRATIONS
C           OR TEMPERATURES OF INFLOWS AT THESE POINTS
C       (4) TIME-DEPENDENT ENERGY OR SOLUTE MASS SOURCES


C     ----------------------------------------------------------------
C                --- DEFINITION OF REQUIRED VARIABLES ---
C     ----------------------------------------------------------------
C     NN   = EXACT NUMBER OF NODES IN MESH
C     NHBC = EXACT NUMBER OF SPECIFIED HEAD NODES
C     NUBC = EXACT NUMBER OF SPECIFIED CONCENTRATION

C     IT = NUMBER OF CURRENT TIME STEP

C     TSEC   = TIME AT END OF CURRENT TIME STEP IN SECONDS
C     TMIN   =  "    "   "   "       "      "    "  "  MINUTES
C     THOUR  =  "    "   "   "       "      "    "  "  HOURS
C     TDAY   =  "    "   "   "       "      "    "  "  DAYS
C     TWEEK  =  "    "   "   "       "      "    "  "  WEEKS
C     TMONTH =  "    "   "   "       "      "    "  "  MONTHS
C     TYEAR  =  "    "   "   "       "      "    "  "  YEARS

C     HBC(IP)     = SPECIFIED HEAD VALUE AT IP(TH) SPECIFIED
C                   HEAD NODE
C     UBC(IP)     = SPECIFIED CONCENTRATION VALUE OF ANY
C                   INFLOW OCCURRING AT IP(TH) SPECIFIED HEAD NODE
C     IHBC(IP)    = ACTUAL NODE NUMBER OF IP(TH) SPECIFIED HEAD NODE
C                   [WHEN NODE NUMBER I=IHBC(IP) IS NEGATIVE (I<0),
C                   VALUES MUST BE SPECIFIED FOR HBC AND UBC.]

C     UBC(IUP)    = SPECIFIED CONCENTRATION VALUE AT IU(TH) SPECIFIED
C                   CONCENTRATION NODE (WHERE IUP=IU+NHBC)
C     IUBC(IUP)   = ACTUAL NODE NUMBER OF IU(TH) SPECIFIED CONCENTRATION
C                   NODE (WHERE IUP=IU+NHBC)
C                   [WHEN NODE NUMBER I=IUBC(IU) IS NEGATIVE (I<0),
C                   A VALUE MUST BE SPECIFIED FOR UBC.]

C     IQSOP(IQP) = NODE NUMBER OF IQP(TH) FLUID SOURCE NODE.
C                   [WHEN NODE NUMBER I=IQSOP(IQP) IS NEGATIVE (I<0),
C                   VALUES MUST BE SPECIFIED FOR QIN AND UIN.]
C     QIN(-I)     = SPECIFIED FLUID SOURCE VALUE AT NODE (-I)
C     UIN(-I)     = SPECIFIED CONCENTRATION VALUE OF ANY INFLOW OCCURRING
C                   AT FLUID SOURCE NODE (-I)
C     ------------------------------------------------------------------
```

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTPM1,DLTUM1,DLTPM2,DLTUM2
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/CHEM/ EQCSTU,EQCSTV

      DIMENSION IHBC(NBCN),HBC(NBCN),UBC(NBCN),VBC(NBCN),WBC(NBCN)
      DIMENSION IC1BC(NBCN),IC2BC(NBCN),IC4BC(NBCN),QIN(NN)
      DIMENSION UIN(NN),VIN(NN),WIN(NN),IQSO(NSO)
C     -----------------------------------------------------------------

      NSOI=NSO-1
      IF(IHBCT) 50,240,240
   50 DO 200 IP=1,NHBC
      I=IHBC(IP)
      IF(I) 100,200,200
  100 CONTINUE

C     NOTE : A FLOW AND TRANSPORT SOLUTION MUST OCCUR FOR ANY
C            TIME STEP IN WHICH HBC( ) CHANGES.

C      HBC(-I) =
C      UBC(IP) =
C      ENDIF
  200 CONTINUE

  240 IF(IUBCT) 250,440,440
  250 DO 400 IU=1,NBCSUM
      IUP=IU+NHBC
      IF(IFLAG.EQ.1) I=IC2BC(IUP)
      IF(IFLAG.EQ.2) I=IC4BC(IUP)
      IF(IFLAG.EQ.3) I=IC1BC(IUP)
      IF(I) 300,400,400
  300 CONTINUE
C     NOTE : A TRANSPORT SOLUTION MUST OCCUR FOR ANY TIME STEP
C            IN WHICH UBC( ) CHANGES.  IN ADDITION, IF FLUID PROPERTIES
C            ARE SENSITIVE TO 'U' THEN A FLOW SOLUTION MUST OCCUR AS WEL
C     UBC(IUP) =   ((          ))
  400 CONTINUE

C      WRITE(6,10)
C   10 FORMAT(/2X,'TIME',2X,'NODE',15X,'FLUX')

  440 IF(IQSOT) 450,640,640
  450 DO 600 IQ=1,NSOI
      I=IQSO(IQ)
      IF(I) 500,600,600
  500 CONTINUE

```

```
C     NOTE : A FLOW AND TRANSPORT SOLUTION MUST OCCUR FOR ANY
C            TIME STEP IN WHICH QIN( ) CHANGES.

C  502 QIN(-I) =
  503 CONTINUE

C     NOTE : A TRANSPORT SOLUTION MUST OCCUR FOR ANY
C            TIME STEP IN WHICH UIN( ) CHANGES.

C      IF(IEXCH) 590,590,570
C  570 IF(IFLAG.EQ.1) UIN(-I) =
C      IF(IFLAG.EQ.2) VIN(-I) =
C      IF(IFLAG.EQ.3) WIN(-I) =
C      GO TO 600
C  590 IF(IFLAG.EQ.1) UIN(-I) =
C      IF(IFLAG.EQ.2) VIN(-I) =
C      IF(IFLAG.EQ.3) WIN(-I) =
C      ELSE
C      QIN(-I)=
C      UIN(-I)=
C      VIN(-I)=
C      WIN(-I)=
C      ENDIF

  600 CONTINUE

      RETURN
      END
```

```
C     SUBROUTINE          E L E M E N         SATRA-CHEM - VERSION 0684 S10.....
      SUBROUTINE ELEMEN(ML,IN,X,Y,THICK,HM1,POR,DISPL,DISPT,            S20.....
     1   PERMXX,PERMXY,PERMYX,PERMYY,VMAG,VANG,VOL,HMAT,HVEC,           S30.....
     2   UMAT,UVEC,VVEC,WVEC)                                           S40.....
                                                                        S50.....
C     FUNCTION:                                                         S60.....
C      TO CONTROL AND CARRY OUT ALL CALCULATIONS FOR EACH ELEMENT BY    S70.....
C      OBTAINING ELEMENT INFORMATION FROM THE BASIS FUNCTION ROUTINE,   S80.....
C      CARRYING OUT GAUSSIAN INTEGRATION OF FINITE ELEMENT INTEGRALS,   S90.....
C      AND SENDING RESULTS OF ELEMENT INTEGRATIONS TO A GLOBAL ASSEMBLY S100....
C      ROUTINE.  ADDITIONALLY, VELOCITY AND ITS ANGLE FROM +X-AXIS IS   S110....
C      CALCULATED AT EACH ELEMENT CENTROID PER PRINTED OUTPUT.          S120....
                                                                        S130....
C     ----------------------------------------------------------------- S140....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               S150....
      COMMON/DIMS/ NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,     S160....
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT                                    S170....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,RHOS,DECAY,SIGMAW,FF,CBART     S180....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        S190....
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTHM1,DLTUM1,DLTPM2,DLTUM2          S200....
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,         S210....
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH                S220....
      COMMON/KPRINT/ KCOORD,KELINF,KINCID,KPLOTP,KPLOTU,KVEL            S230....
                                                                        S240....
      DIMENSION IN(NIN),X(NN),Y(NN),THICK(NN),HM1(NN)                   S250....
      DIMENSION POR(NN),DISPL(NE),DISPT(NE),VMAG(NE),VANG(NE)           S260....
      DIMENSION PERMXX(NE),PERMXY(NE),PERMYX(NE),PERMYY(NE)             S270....
      DIMENSION VOL(NN),HMAT(NN,NBI),HVEC(NN),UMAT(NN,NBI),UVEC(NN)     S280....
      DIMENSION VVEC(NN),WVEC(NN)                                       S290....
      DIMENSION BFLOWE(4,4),DFLOWE(4),BTRANE(4,4),DTRANE(4,4),VOLE(4)   S300....
      DIMENSION F(4,4),W(4,4),DET(4),DFDXG(4,4),DFDYG(4,4)              S310....
      DIMENSION DWDXG(4,4),DWDYG(4,4)                                   S320....
      DIMENSION PORG(4),VXG(4),VYG(4),VGMAG(4),THICKG(4)                S330....
      DIMENSION BXXG(4),BXYG(4),BYXG(4),BYYG(4),EXG(4),EYG(4)           S340....
C     ----------------------------------------------------------------- S350....
                                                                        S360....
      DATA GLOC/0.577350269189626D0/                                    S370....
C     DATA GLOC/0.57735026919D0/                                        S380....
                                                                        S390....
C     --- DECIDE WHETHER TO CALCULATE CENTROID VELOCITIES ON THIS CALL  S400....
      IVPRNT=0                                                          S410....
      IF(MOD(IT,NPRINT).EQ.0.AND.ML.LT.2.AND.IT.NE.0) IVPRNT=1          S420....
C     IF(IT.EQ.1.AND.ISSTAT.EQ.2) IVPRNT=1                              S430....
      IF(IT.EQ.1) IVPRNT=1                                              S440....
      KVPRNT=IVPRNT+KVEL                                                S450....
                                                                        S460....
      DO 5000 L=1,NE                                                    S470....
      IX=-1                                                             S480....
      IY=-1                                                             S490....
      KG=0                                                              S500....
      DO 20 IYL=1,2                                                     S510....
      DO 10 IXL=1,2                                                     S520....
      KG=KG+1                                                           S530....
      XLOC=IX*GLOC                                                      S540....
      YLOC=IY*GLOC                                                      S550....
                                                                        S560....
C     ---- CALCULATE BASIS FUNCTIONS AND ASSOCIATED PARAMETERS ---      S570....
      CALL BASIS2(ML,L,XLOC,YLOC,IN,X,Y,F(1,KG),W(1,KG),DET(KG),        S580....
     1   DFDXG(1,KG),DFDYG(1,KG),DWDXG(1,KG),DWDYG(1,KG),               S590....
     2   HM1,POR,THICK,THICKG(KG),VXG(KG),VYG(KG),                      S600....
     3   PORG(KG),VGMAG(KG),PERMXX,PERMXY,PERMYX,PERMYY)                S610....
   10 IX=-IX                                                            S620....
   20 IY=-IY                                                            S630....
                                                                        S640....
C     --- CALCULATE VELOCITY AT ELEMENT CENTROID WHEN REQUIRED ---      S650....
      IF(KVPRNT-2) 29,22,29                                             S660....
   22 AXSUM=0.0D0                                                       S670....
      AYSUM=0.0D0                                                       S680....
      DO 24 KG=1,4                                                      S690....
      AXSUM=AXSUM+VXG(KG)                                               S700....
   24 AYSUM=AYSUM+VYG(KG)                                               S710....
      VMAG(L)=DSQRT(AXSUM*AXSUM+AYSUM*AYSUM)/4.0D0                      S720....
      IF(AXSUM) 25,27,28                                                S730....
   25 AYX=AYSUM/AXSUM                                                   S740....
      VANG(L)=DATAN(AYX)/1.745329D-2                                    S750....
      IF(AYSUM.LT.0.0) GOTO 26                                          S760....
      VANG(L)=VANG(L)+180.0D0                                           S770....
      GOTO 29                                                           S780....
   26 VANG(L)=VANG(L)-180.0D0                                           S790....
      GOTO 29                                                           S800....
   27 VANG(L)=90.0D0                                                    S810....
      IF(AYSUM.LT.0.0) VANG(L)=-90.0D0                                  S820....
      GOTO 29                                                           S830....
   28 AYX=AYSUM/AXSUM                                                   S840....
      VANG(L)=DATAN(AYX)/1.745329D-2                                    S850....
                                                                        S860....
C     --- INCLUDE MESH THICKNESS IN NUMERICAL INTEGRATION ---           S870....
   29 DO 30 KG=1,4                                                      S880....
   30 DET(KG)=THICKG(KG)*DET(KG)                                        S890....
                                                                        S900....
      IF(ML-1) 40,40,900                                                S910....
                                                                        S920....
C     --- INTEGRATE FLUID MASS BALANCE ---                              S930....
   40 DO 700 I=1,4                                                      S940....
C     DF = 0.0D0                                                        S950....
      VO=0.0D0                                                          S960....
      DO 500 KG=1,4                                                     S970....
C     DF = DF + 0.0D0                                                   S980....
  500 VO=VO+F(I,KG)*DET(KG)                                             S990....
      DO 600 J=1,4                                                      S1000...
      BF=0.0D0                                                          S1010...
      DO 550 KG=1,4                                                     S1020...
  550 BF=BF+((PERMXX(L)*DFDXG(J,KG)+PERMXY(L)*DFDYG(J,KG))*DFDXG(I,KG)  S1030...
     1    + (PERMYX(L)*DFDXG(J,KG)+PERMYY(L)*DFDYG(J,KG))*DFDYG(I,KG))  S1040...
     2   *DET(KG)                                                       S1050...
  600 BFLOWE(I,J)=BF                                                    S1060...
```

```
      VOLE(I)=VO
  700 DFLOWE(I) = 0.0D0

      IF(ML-1) 900,3000,900
  900 IF(NOMATX.NE.0.OR.IFLAG.NE.1) GOTO 3000

C     --- CALCULATE PARAMETERS FOR SOLUTE MASS BALANCE AT GAUSS POINTS
      DO 1000 KG=1,4
      IF(VGMAG(KG)) 930,930,960
  930 EXG(KG)=0.0D0
      EYG(KG)=0.0D0
      DXXG=0.0D0
      DXYG=0.0D0
      DYXG=0.0D0
      DYYG=0.0D0
      GOTO 990
  960 EXG(KG)=PORG(KG)*VXG(KG)
      EYG(KG)=PORG(KG)*VYG(KG)

C     --- CALCULATE PARAMETERS FOR DISPERSION TENSOR ---
      DLG=DISPL(L)*VGMAG(KG)
      DTG=DISPT(L)*VGMAG(KG)
      V2GMI=1.D0/(VGMAG(KG)*VGMAG(KG))
      V2ILTG=V2GMI*(DLG-DTG)
      VX2G=VXG(KG)*VXG(KG)
      VY2G=VYG(KG)*VYG(KG)

C     --- DISPERSION TENSOR TERMS ---
      DXXG=V2GMI*(DLG*VX2G+DTG*VY2G)
      DYYG=V2GMI*(DTG*VX2G+DLG*VY2G)
      DXYG=V2ILTG*VXG(KG)*VYG(KG)
      DYXG=DXYG

C     --- ADD DIFFUSION AND DISPERSION TERMS TO TOTAL DISPERSION TENSOR
  990 BXXG(KG)=PORG(KG)*(DXXG+SIGMAW)
      BXYG(KG)=PORG(KG)* DXYG
      BYXG(KG)=PORG(KG)* DYXG
 1000 BYYG(KG)=PORG(KG)*(DYYG+SIGMAW)

C     --- INTEGRATE SOLUTE MASS BALANCE ---
      DO 2000 I=1,4
      DO 2000 J=1,4
      BT=0.
      DT=0.
      DO 1500 KG=1,4
      BT=BT+((BXXG(KG)*DFDXG(J,KG)+BXYG(KG)*DFDYG(J,KG))*DFDXG(I,KG)
     1      +(BYXG(KG)*DFDXG(J,KG)+BYYG(KG)*DFDYG(J,KG))*DFDYG(I,KG))
     2   *DET(KG)
 1500 DT=DT+(EXG(KG)*DFDXG(J,KG)+EYG(KG)*DFDYG(J,KG))*W(I,KG)*DET(KG)
      BTRANE(I,J)=BT
 2000 DTRANE(I,J)=DT
 3000 CONTINUE
```

```
C     --- SEND RESULTS OF INTEGRATIONS FOR ELEMENT TO GLOBAL ASSEMBLY   S1600...
 5000 CALL GLOBAN(L,ML,VOLE,BFLOWE,DFLOWE,BTRANE,DTRANE,                S1610...
     1   IN,VOL,HMAT,HVEC,UMAT,UVEC,VVEC,WVEC)                          S1620...
                                                                        S1630...
                                                                        S1640...
      RETURN                                                            S1650...
      END                                                               S1660...
```

```
C     SUBROUTINE           B  A  S  I  S  2     SATRA-CHEM - VERSION 0684 T10.....
      SUBROUTINE BASIS2(ML,L,XLOC,YLOC,IN,X,Y,F,W,DET,                    T20.....
     1   DFDXG,DFDYG,DWDXG,DWDYG,PM1,POR,THICK,THICKG,VXG,VYG,            T30.....
     2   PORG,VGMAG,PERMXX,PERMXY,PERMYX,PERMYY)                          T40.....
                                                                          T50.....
C     FUNCTION:                                                           T60.....
C      TO CALCULATE VALUES OF BASIS AND WEIGHTING FUNCTIONS AND THEIR     T70.....
C      DERIVATIVES, TRANSFORMATION MATRICES BETWEEN LOCAL AND GLOBAL      T80.....
C      COORDINATES AND PARAMETER VALUES AT A SPECIFIED POINT IN A         T90.....
C      QUADRILATERAL FINITE ELEMENT                                       T100....
                                                                          T110....
C     ------------------------------------------------------------------- T120....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                 T130....
      COMMON/DIMS/ NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,       T140....
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT                                      T150....
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,           T160....
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH                  T170....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,RHOS,DECAY,SIGMAW,FF,CBART        T180....
                                                                          T190....
                                                                          T200....
      DIMENSION IN(NIN),X(NN),Y(NN),PM1(NN),                              T210....
     1   POR(NN),PERMXX(NE),PERMXY(NE),PERMYX(NE),PERMYY(NE),THICK(NN)    T220....
      DIMENSION F(4),W(4),DFDXG(4),DFDYG(4),                              T230....
     1   DWDXG(4),DWDYG(4)                                                T240....
      DIMENSION FX(4),FY(4),AFX(4),AFY(4),                                T250....
     1   DFDXL(4),DFDYL(4),DWDXL(4),DWDYL(4),                             T260....
     2   XDW(4),YDW(4),IIX(4),IIY(4)                                      T270....
      DATA IIX/-1,+1,+1,-1/,IIY/-1,-1,+1,+1/                              T280....
C     ------------------------------------------------------------------- T290....
                                                                          T300....
C     --- AT THIS LOCATION IN LOCAL COORDINATES, (XLOC,YLOC),             T310....
          CALCULATE SYMMETRIC WEIGHTING FUNCTIONS, F(I), SPACE            T320....
          DERIVATIVES, DFDXG(I) AND DFDYG(I), AND THE DETERMINANT         T330....
          OF THE JACOBIAN, DET.                                           T340....
                                                                          T350....
      XF1=1.D0-XLOC                                                       T360....
      XF2=1.D0+XLOC                                                       T370....
      YF1=1.D0-YLOC                                                       T380....
      YF2=1.D0+YLOC                                                       T390....
                                                                          T400....
C     --- CALCULATE BASIS FUNCTION, F ---                                 T410....
      FX(1)=XF1                                                           T420....
      FX(2)=XF2                                                           T430....
      FX(3)=XF2                                                           T440....
      FX(4)=XF1                                                           T450....
      FY(1)=YF1                                                           T460....
      FY(2)=YF1                                                           T470....
      FY(3)=YF2                                                           T480....
      FY(4)=YF2                                                           T490....
      DO 10 I=1,4                                                         T500....
   10 F(I)=0.250D0*FX(I)*FY(I)                                            T510....
                                                                          T520....
C     --- CALCULATE DERIVATIVES WITH RESPECT TO LOCAL COORDINATES ---     T530....
```

```
      DO 20 I=1,4                                                       T540....
      DFDXL(I)=IIX(I)*0.250D0*FY(I)                                     T550....
   20 DFDYL(I)=IIY(I)*0.250D0*FX(I)                                     T560....
                                                                        T570....
C     --- CALCULATE JACOBIAN, CJ(), AT CURRENT LOCAL COORDINATES ---    T580....
      CJ11=0.0D0                                                        T590....
      CJ12=0.0D0                                                        T600....
      CJ21=0.0D0                                                        T610....
      CJ22=0.0D0                                                        T620....
      DO 100 IL=1,4                                                     T630....
      II=(L-1)*4+IL                                                     T640....
      I=IN(II)                                                          T650....
      CJ11=CJ11+DFDXL(IL)*X(I)                                          T660....
      CJ12=CJ12+DFDXL(IL)*Y(I)                                          T670....
      CJ21=CJ21+DFDYL(IL)*X(I)                                          T680....
  100 CJ22=CJ22+DFDYL(IL)*Y(I)                                          T690....
                                                                        T700....
C     --- CALCULATE DETERMINANT OF JACOBIAN, DET ---                    T710....
      DET=CJ11*CJ22-CJ21*CJ12                                           T720....
      ODET=1.D0/DET                                                     T730....
                                                                        T740....
C     --- CALCULATE COMPONENTS OF INVERSE JACOBIAN MATRIX, CIJ ---      T750....
      CIJ11=+ODET*CJ22                                                  T760....
      CIJ12=-ODET*CJ12                                                  T770....
      CIJ21=-ODET*CJ21                                                  T780....
      CIJ22=+ODET*CJ11                                                  T790....
                                                                        T800....
C     --- CALCULATE DERIVATIVES WITH RESPECT TO GLOBAL COORDINATES ---  T810....
      DO 200 I=1,4                                                      T820....
      DFDXG(I)=CIJ11*DFDXL(I)+CIJ12*DFDYL(I)                            T830....
  200 DFDYG(I)=CIJ21*DFDXL(I)+CIJ22*DFDYL(I)                            T840....
                                                                        T850....
C     --- CALCULATE PARAMETER VALUES AT CURRENT LOCAL COORDINATES ---   T860....
      DPDXG=0.0D0                                                       T870....
      DPDYG=0.0D0                                                       T880....
      PORG=0.0D0                                                        T890....
      THICKG=0.0D0                                                      T900....
      DO 1000 IL=1,4                                                    T910....
      II=(L-1)*4 +IL                                                    T920....
      I=IN(II)                                                          T930....
      DPDXG=DPDXG+PM1(I)*DFDXG(IL)                                      T940....
      DPDYG=DPDYG+PM1(I)*DFDYG(IL)                                      T950....
      PORG=PORG+POR(I)*F(IL)                                            T960....
 1000 THICKG=THICKG+THICK(I)*F(IL)                                      T970....
                                                                        T980....
C     --- CALCULATE GLOBAL FLUID VELOCITY AT CURRENT LOCAL COORDINATES  T990....
      DENOM=1.D0/PORG                                                   T1000...
      VXG=-DENOM*(PERMXX(L)*DPDXG+PERMXY(L)*DPDYG)                      T1010...
      VYG=-DENOM*(PERMYX(L)*DPDXG+PERMYY(L)*DPDYG)                      T1020...
      VGMAG=DSQRT(VXG*VXG+VYG*VYG)                                      T1030...
                                                                        T1040...
C     --- RETURN IF THIS IS A FLOW-ONLY TIME-STEP ---                   T1050...
      IF(ML-1) 1020,1010,1020                                           T1060...
 1010 RETURN                                                              T1070...
                                                                          T1080...
C     --- RETURN IF UP=0, I.E.: IF NO UPSTREAM WEIGHTING ---              T1090...
 1020 IF(UP.GT.0.0) GOTO 1030                                             T1100...
      DO 1025 I=1,4                                                       T1110...
 1025 W(I)=F(I)                                                           T1120...
      RETURN                                                              T1130...
                                                                          T1140...
                                                                          T1150...
C     ---- CALCULATE ASYMMETRIC WEIGHTING FUNCTIONS ---                   T1160...
 1030 VXL=CIJ11*VXG+CIJ21*VYG                                             T1170...
      VYL=CIJ12*VXG+CIJ22*VYG                                             T1180...
      VLMAG=DSQRT(VXL*VXL+VYL*VYL)                                        T1190...
      AA=0.0D0                                                            T1200...
      BB=0.0D0                                                            T1210...
      IF(VLMAG) 1900,1900,1800                                            T1220...
 1800 AA=UP*VXL/VLMAG                                                     T1230...
      BB=UP*VYL/VLMAG                                                     T1240...
                                                                          T1250...
 1900 XIXI=.750D0*AA*XF1*XF2                                              T1260...
      YIYI=.750D0*BB*YF1*YF2                                              T1270...
      DO 2000 I=1,4                                                       T1280...
      AFX(I)=.50D0*FX(I)+IIX(I)*XIXI                                      T1290...
 2000 AFY(I)=.50D0*FY(I)+IIY(I)*YIYI                                      T1300...
                                                                          T1310...
      DO 3000 I=1,4                                                       T1320...
 3000 W(I)=AFX(I)*AFY(I)                                                  T1330...
                                                                          T1340...
                                                                          T1350...
      RETURN                                                              T1360...
      END                                                                 T1370...
```

```
C     SUBROUTINE        G L O B A N     SATRA-CHEM - VERSION 0684
      SUBROUTINE GLOBAN(L,ML,VOLE,BFLOWE,DFLOWE,BTRANE,DTRANE,
     1   IN,VOL,HMAT,HVEC,UMAT,UVEC,VVEC,WVEC)

C     FUNCTION:
C      TO  ASSEMBLE RESULTS OF ELEMENTWISE INTEGRATIONS INTO
C      A GLOBAL BANDED MATRIX AND GLOBAL VECTOR FOR BOTH
C      FLOW AND TRANSPORT.

C     ----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/ NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH

      DIMENSION BFLOWE(4,4),DFLOWE(4),BTRANE(4,4),DTRANE(4,4),VOLE(4)
      DIMENSION VOL(NN),HMAT(NN,NBI),HVEC(NN),UMAT(NN,NBI),UVEC(NN)
      DIMENSION IN(NIN),VVEC(NN),WVEC(NN)
C     ----------------------------------------------------------------

      N1=(L-1)*4+1
      N4=N1+3

C     --- ADD RESULTS OF INTEGRATIONS OVER ELEMENT L TO GLOBAL
          H-MATRIX AND H-VECTOR.

      IF(ML-1) 50,50,150
   50 IE=0
      DO 100 II=N1,N4
      IE=IE+1
      IB=IN(II)
      VOL(IB)=VOL(IB)+VOLE(IE)
      HVEC(IB)=HVEC(IB)+DFLOWE(IE)
      JE=0
      DO 100 JJ=N1,N4
      JE=JE+1
      JB=IN(JJ)-IB+NBHALF
  100 HMAT(IB,JB)=HMAT(IB,JB)+BFLOWE(IE,JE)

C     --- ADD RESULTS OF INTEGRATIONS OVER ELEMENT L TO GLOBAL U-MATRIX
      IF(ML-1) 150,300,150
  150 IF(NOMATX.NE.0.OR.IFLAG.NE.1) GOTO 300
      IE=0
      DO 200 II=N1,N4
      IE=IE+1
      IB=IN(II)

C     --- POSITION FOR ADDITION TO U-VECTOR ---
C     UVEC(IB)=UVEC(IB)+ ((    ))

      JE=0
      DO 200 JJ=N1,N4
```

```
    JE=JE+1                                             U540....
    JB=IN(JJ)-IB+NBHALF                                 U550....
200 UMAT(IB,JB)=UMAT(IB,JB)+DTRANE(IE,JE)+BTRANE(IE,JE) U560....
300 CONTINUE                                            U580....
                                                        U600....
    RETURN                                              U610....
    END                                                 U620....
```

```
C     SUBROUTINE        N  O  D  A  L  B        SATRA2 - VERSION 0684
      SUBROUTINE NODALB(ML,VOL,HMAT,HVEC,UMAT1,UMAT2,UVEC,VVEC,WVEC,
     1   HM1,UM1,VM1,WM1,POR,STOR,CS1,CS2,CS3,UIN,VIN,WIN,QIN)

C     FUNCTION:
C      (1)  TO CARRY OUT ALL CELLWISE CALCULATIONS AND TO ADD CELLWISE
C             TERMS TO THE GLOBAL BANDED MATRIX AND GLOBALVECTOR FOR
C             BOTH FLOW AND TRANSPORT EQUATIONS.
C      (2)  TO ADD FLUID SOURCE TERMS TO THE MATRIX EQUATIONS.

C -----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/ NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTHM1,DLTUM1,DLTPM2,DLTUM2
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,RHOS,DECAY,SIGMAW,FF,CBART
      COMMON/BFACS/ BDELP,BDELU
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH

      DIMENSION VOL(NN),HMAT(NN,NBI),HVEC(NN),UMAT1(NN,NBI),UVEC(NN)
      DIMENSION UMAT2(NN,NBI),VVEC(NN),WVEC(NN)
      DIMENSION HM1(NN),UM1(NN),VM1(NN),WM1(NN)
      DIMENSION POR(NN),STOR(NN),UIN(NN),VIN(NN),WIN(NN)
      DIMENSION CS1(NN),CS2(NN),CS3(NN),QIN(NN)
C --------------------------------------------------------------------

      DO 1000 I=1,NN
      IF(ML-1) 20,20,30

C     --- CELLWISE TERM FOR H EQUATION ---
   20 AFLN=STOR(I)*VOL(I)/DELTP

C     --- ADD CELLWISE TERM AND FLUID SOURCES OR FLUXES TO H EQUATION
      HMAT(I,NBHALF)=HMAT(I,NBHALF)+AFLN
      HVEC(I)=HVEC(I)+AFLN*HM1(I)+QIN(I)

      IF(ML-1) 30,1000,30

C     --- CALCULATE CELLWISE TERMS FOR U EQUATION ---
   30 EPRS=(1.D0-POR(I))*RHOS
      ATRN=(POR(I)+EPRS*CS1(I))*VOL(I)/DELTU
      ETRN=EPRS*CS3(I)*VOL(I)
      GTRN=(EPRS*CS2(I)+POR(I)*DECAY)*VOL(I)
      IF(IFLAG.EQ.1) ATRNR=ATRN*UM1(I)
      IF(IFLAG.EQ.2) ATRNR=ATRN*VM1(I)
      IF(IFLAG.EQ.3) ATRNR=ATRN*WM1(I)

C     --- CALCULATE SOURCES OF SOLUTE CONTAINED IN SOURCES OF FLUID ---
      QR=0.0D0
      QL=0.0D0
      IF(QIN(I)) 100,100,50
   50 QL=-QIN(I)
      IF(IFLAG.EQ.1) QR=-QL*UIN(I)
      IF(IFLAG.EQ.2) QR=-QL*VIN(I)
      IF(IFLAG.EQ.3) QR=-QL*WIN(I)

C     --- ADD CELLWISE TERMS, SOURCES, FLUXES TO U-EQUATION ---
  100 IF (IFLAG .LT. 3) THEN
      IF (IFLAG .EQ. 2) THEN
      VVEC(I)=VVEC(I)+ATRNR-ETRN+QR
      ELSE
      UVEC(I)=UVEC(I)+ATRNR-ETRN+QR
      IF(NOMATX.EQ.0) UMAT1(I,NBHALF)=UMAT1(I,NBHALF)+ATRN+GTRN-QL
      END IF
      ELSE
      UMAT2(I,NBHALF)=UMAT2(I,NBHALF)+ATRN+GTRN-QL
      WVEC(I)=WVEC(I)+ATRNR-ETRN+QR
      END IF

 1000 CONTINUE

      RETURN
      END
```

```
C     SUBROUTINE           B  C  B              SATRA-CHEM - VERSION 0684
      SUBROUTINE BCB(ML,HMAT,HVEC,UMAT1,UMAT2,UVEC,VVEC,WVEC,IHBC,HBC,
     1   IC1BC,IC2BC,IC4BC,UBC,VBC,WBC,QPL,QPLM1,PM1)

C     FUNCTION:
C      TO IMPLEMENT SPECIFIED HEAD AND SPECIFIED CONCENTRATION
C      CONDITIONS BY MODIFYING THE GLOBAL FLOW AND TRANSPORT MATRIX
C      EQUATIONS.

C     ---------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/ NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTPM1,DLTUM1,DLTPM2,DLTUM2
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,RHOS,DECAY,SIGMAW,FF,CBART
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/BFACS/ BDELP,BDELU
      COMMON/GNUBC/ GNU0,GNU

      DIMENSION HMAT(NN,NBI),HVEC(NN),UMAT1(NN,NBI),UVEC(NN)
      DIMENSION IHBC(NBCN),HBC(NBCN),IC1BC(NBCN),UBC(NBCN)
      DIMENSION QPL(NBCN),QPLM1(NBCN),UMAT2(NN,NBI),VVEC(NN)
      DIMENSION WVEC(NN),IC2BC(NBCN),IC4BC(NBCN)
      DIMENSION VBC(NBCN),WBC(NBCN),PM1(NN)
C     ---------------------------------------------------------------

      IF(NHBC.EQ.0) GOTO 1050

C     --- SPECIFIED HEAD BOUNDARY CONDITIONS ---
      DO 1000 IP=1,NHBC
      I=IABS(IHBC(IP))

      IF(ML-1) 100,100,200

C     --- MODIFY H-EQUATION ADDING FLUID SOURCE AT SPECIFIED HEAD NODE
  100 HMAT(I,NBHALF)=HMAT(I,NBHALF)+GNU
      HVEC(I)=HVEC(I)+GNU*HBC(IP)

      IF(ML-1) 200,1000,200

C     --- MODIFY U-EQUATION BY ADDING U SOURCE WHEN FLUID FLOWS IN
C         AT SPECIFIED HEAD NODES

  200 GUR=0.0D0
      GUL=0.0D0
      IF(QPLM1(IP).GT.0) THEN
      GUL=-QPLM1(IP)
      IF(IFLAG .EQ. 1) GUR=-GUL*UBC(IP)
      IF(IFLAG .EQ. 2) GUR=-GUL*VBC(IP)
      IF(IFLAG .EQ. 3) GUR=-GUL*WBC(IP)
      END IF

      IF (IFLAG .LT. 3) THEN
      IF (IFLAG .EQ. 2) THEN
      VVEC(I)=VVEC(I)+GUR
      ELSE
      UVEC(I)=UVEC(I)+GUR
      IF(NOMATX .EQ. 0) UMAT1(I,NBHALF)=UMAT1(I,NBHALF)-GUL
      END IF
      ELSE
      WVEC(I)=WVEC(I)+GUR
      UMAT2(I,NBHALF)=UMAT2(I,NBHALF)-GUL
      END IF
 1000 CONTINUE

C     --- SPECIFIED CONCENTRATION (U) BOUNDARY CONDITIONS ---

C     --- MODIFY U-EQUATION AT SPECIFIED U NODE TO READ U=UBC ---
 1050 IF(ML-1) 1100,3000,1100
 1100 IF (IFLAG .EQ. 1) THEN
      IF(NC2BC.EQ.0) GOTO 3000
      DO 1700 IU=1,NC2BC
      IUP=IU+NHBC
      I=IABS(IC2BC(IUP))
      IF(NOMATX) 1200,1200,1700
 1200 DO 1500 JB=1,NB
 1500 UMAT1(I,JB)=0.0D0
      UMAT1(I,NBHALF)=1.0D0
 1700 UVEC(I)=UBC(IUP)

      ELSE IF (IFLAG .EQ. 2) THEN
      IF(NC4BC .EQ. 0) GO TO 3000
      DO 2150 IV=1,NC4BC
      IVP=IV+NHBC
      I=IABS(IC4BC(IVP))
 2150 VVEC(I)=VBC(IVP)

      ELSE
      IF(NC1BC .EQ. 0) GO TO 3000
      DO 2500 IW=1,NC1BC
      IWP=IW+NHBC
      I=IABS(IC1BC(IWP))
      DO 2350 JB=1,NB
 2350 UMAT2(I,JB)=0.0D0
      UMAT2(I,NBHALF)=1.0D0
 2500 WVEC(I)=WBC(IWP)
      END IF

 3000 CONTINUE


      RETURN
      END
```

```
C     SUBROUTINE         P I N C H B          SATRA-CHEM - VERSION 0684 X10.....
      SUBROUTINE PINCHB(ML,IPINCH,HMAT,HVEC,UMAT1,UMAT2,UVEC,VVEC,WVEC) X20.....
                                                                        X30.....
C     FUNCTION:                                                         X40.....
C      TO IMPLEMENT PINCH NODE CONDITIONS BY MODIFYING THE GLOBAL FLOW  X50.....
C      AND TRANSPORT MATRIX EQUATIONS                                   X60.....
                                                                        X70.....
C     ----------------------------------------------------------------- X80.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               X90.....
      COMMON/DIMS/ NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,     X100....
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT                                    X110....
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,         X120....
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH                X130....
                                                                        X140....
      DIMENSION IPINCH(NPINCH,3),HMAT(NN,NBI),HVEC(NN),UVEC(NN)         X150....
      DIMENSION UMAT1(NN,NBI),UMAT2(NN,NBI),VVEC(NN),WVEC(NN)           X160....
C     ----------------------------------------------------------------- X170....
C     --- NPIN IS ACTUAL NUMBER OF PINCH NODES IN THE MESH ---          X190....
      NPIN=NPINCH-1                                                     X200....
      DO 1000 IPIN=1,NPIN                                               X210....
                                                                        X220....
C     --- SET NUMBERING OF PINCH NODES AND NEIGHBORING NODES ---        X230....
      I=IABS(IPINCH(IPIN,1))                                            X240....
      ICOR1=IPINCH(IPIN,2)                                              X250....
      ICOR2=IPINCH(IPIN,3)                                              X260....
      JC1=ICOR1-I+NBHALF                                                X270....
      JC2=ICOR2-I+NBHALF                                                X280....
                                                                        X290....
      IF(ML-1) 50,50,250                                                X300....
C     --- ADJUST H-EQUATION FOR PINCH NODE CONDITIONS ---               X320....
   50 DO 100 JB=1,NB                                                    X330....
  100 HMAT(I,JB)=0.0D0                                                  X340....
      HVEC(I)=0.0D0                                                     X350....
      HMAT(I,NBHALF)=1.00D0                                             X360....
      HMAT(I,JC1)=-0.50D0                                               X370....
      HMAT(I,JC2)=-0.50D0                                               X380....
      IF(ML-1) 250,1000,250                                             X390....
                                                                        X400....
C     --- ADJUST U-EQUATION FOR PINCH NODE CONDITIONS ---               X410....
  250 IF (IFLAG .LT. 3) THEN                                            X420....
      DO 300 JB=1,NB                                                    X430....
  300 UMAT1(I,JB)=0.0D0                                                 X440....
      UMAT1(I,NBHALF)=1.00D0                                            X450....
      UMAT1(I,JC1)=-0.50D0                                              X460....
      UMAT1(I,JC2)=-0.50D0                                              X470....
      IF(IFLAG .EQ. 1) THEN                                             X480....
      UVEC(I)=0.0D0                                                     X490....
      ELSE                                                              X500....
      VVEC(I)=0.0D0                                                     X510....
      END IF                                                            X520....
      ELSE                                                              X530....
      DO 310 JB=1,NB                                                    X540....
  310 UMAT2(I,JB)=0.0D0                                                 X550....
```

```
      UMAT2(I,NBHALF)=1.00D0
      UMAT2(I,JC1)=-0.50D0
      UMAT2(I,JC2)=-0.50D0
      WVEC(I)=0.0D0
      END IF
 1000 CONTINUE

      RETURN
      END
```

```
C     SUBROUTINE        S O L V E B        SATRA2 - VERSION 0684
      SUBROUTINE SOLVEB(KKK,C,R,NNP,IHALFB,MAXNP,MAXBW)

C     FUNCTION:
C      TO SOLVE THE MATRIX EQUATION BY:
C       (1) DECOMPOSING THE MATRIX
C       (2) MODIFYING THE RIGHT-HAND SIDE
C       (3) BACK-SUBSTITUTING FOR THE SOLUTION

C     ------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION C(MAXNP,MAXBW),R(MAXNP)

C     ------------------------------------------------------------------

      IHBP=IHALFB+1

C     --- DECOMPOSE MATRIX C BY BANDED GAUSSIAN ELIMINATION FOR A
C         NON-SYMMETRIC MATRIX

      IF(KKK-1) 5,5,50
    5 NU=NNP-IHALFB
      DO 20 NI=1,NU
      PIVOTI=1./C(NI,IHBP)
      NJ=NI+1
      IB=IHBP
      NK=NI+IHALFB
      DO 10 NL=NJ,NK
      IB=IB-1
      A=-C(NL,IB)*PIVOTI
      C(NL,IB)=A
      JB=IB+1
      KB=IB+IHALFB
      LB=IHBP-IB
      DO10 MB=JB,KB
      NB=LB+MB
   10 C(NL,MB)=C(NL,MB)+A*C(NI,NB)
   20 CONTINUE
      NR=NU+1
      NU=NNP-1
      NK=NNP
      DO 40 NI=NR,NU
      PIVOTI=1.D0/(C(NI,IHBP))
      NJ=NI+1
      IB=IHBP
      DO 30 NL=NJ,NK
      IB=IB-1
      A=-C(NL,IB)*PIVOTI
      C(NL,IB)=A
      JB=IB+1
      KB=IB+IHALFB
      LB=IHBP-IB
      DO 30 MB=JB,KB
      NB=LB+MB
   30 C(NL,MB)=C(NL,MB)+A*C(NI,NB)
   40 CONTINUE
      IF(KKK-1) 50,44,50
   44 RETURN

C     --- UPDATE RIGHT-HAND SIDE VECTOR, R ---
   50 NU=NNP+1
      IBAND=2*IHALFB+1
      DO 70 NI=2,IHBP
      IB=IHBP-NI+1
      NJ=1
      SUM=0.0D0
      DO 60 JB=IB,IHALFB
      SUM=SUM+C(NI,JB)*R(NJ)
   60 NJ=NJ+1
   70 R(NI)=R(NI)+SUM
      IB=1
      NL=IHBP+1
      DO 90 NI=NL,NNP
      NJ=NI-IHBP+1
      SUM=0.0D0
      DO 80 JB=IB,IHALFB
      SUM=SUM+C(NI,JB)*R(NJ)
   80 NJ=NJ+1
   90 R(NI)=R(NI)+SUM

C     --- BACK SOLVE ---
      R(NNP)=R(NNP)/C(NNP,IHBP)
      DO 110 IB=2,IHBP
      NI=NU-IB
      NJ=NI
      MB=IHALFB+IB
      SUM=0.0D0
      DO 100 JB=NL,MB
      NJ=NJ+1
  100 SUM=SUM+C(NI,JB)*R(NJ)
  110 R(NI)=(R(NI)-SUM)/C(NI,IHBP)
      MB=IBAND
      DO 130 IB=NL,NNP
      NI=NU-IB
      NJ=NI
      SUM=0.0D0
      DO 120 JB=NL,MB
      NJ=NJ+1
  120 SUM=SUM+C(NI,JB)*R(NJ)
  130 R(NI)=(R(NI)-SUM)/C(NI,IHBP)


      RETURN
      END
```

```
C     SUBROUTINE          P  R  O  J  E  C           SATRA2 - VERSION 0684
      SUBROUTINE PROJEC (UPROJ,VPROJ,WPROJ,WITER,UM1,UM2,
     1   VM1,VM2,WM1,WM2,UVEC,VVEC,WVEC,AU,BU,UAVG,VAVG,WAVG)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/CONTRL/ ME,ISSTAT,ITCYC,DTMULT,NPCYC,NUCYC,NPRINT,
     1   IREAD,ISTORE,UP,NOMATX,IFLAG,ISORB,IEQLIB,IEXCH
      COMMON/ITERAT/ITER,ITRMAX,RPM,RPMAX,IPWORS,RWM,RWMAX,IWWORS,
     1   RTOL,MAXITR

      DIMENSION UPROJ(NN),UM1(NN),UM2(NN),UVEC(NN),VVEC(NN),WVEC(NN)
      DIMENSION VPROJ(NN),WPROJ(NN),VM1(NN),VM2(NN),WM1(NN),WM2(NN)
      DIMENSION WITER(NN),UAVG(NN),VAVG(NN),WAVG(NN)

      IF (ITER .EQ. 1) THEN
      DO 160 I=1,NN
      WAVG(I)=(WVEC(I)+WM1(I))*0.5D0
  160 WPROJ(I)=WVEC(I)

      ELSE
C     --- EXTRAPOLATION/DAMPING FORMULA FOR PROJECTED W. ---
      ALPHA=0.0000000000D0
      IF(IEXCH.EQ.1) ALPHA=.80000000000D0
      DO 170 I=1,NN
      WPROJ(I)=(1.D0+ALPHA)*WVEC(I)-ALPHA*WITER(I)
  170 WAVG(I)=(WPROJ(I)+WM1(I))*0.50D0

      END IF
      RETURN

      END
```

```
C     SUBROUTINE          S T O R E         SATRA-CHEM - VERSION 0684
      SUBROUTINE STORE(HVEC,C1,C2,C4,HM1,UM1,VM1,WM1,
     1   CS1,CS3,QPL,QPLM1,UBC,VBC,WBC)

C     FUNCTION:
C      TO STORE RESULTS THAT MAY BE USED LATER TO RE-START THE
C      SIMULATION.

C     -----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/NN,NE,NIN,NBI,NB,NBHALF,NPINCH,NHBC,NC1BC,NC2BC,
     1   NC4BC,NSO,NBCN,NBCSUM,NCONT
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,
     1   IT,ITMAX,TMAX,DELTP,DELTU,DLTHM1,DLTUM1,DLTPM2,DLTUM2

      DIMENSION HVEC(NN),C1(NN),C2(NN),C4(NN),HM1(NN),UM1(NN)
      DIMENSION CS1(NN),VM1(NN),WM1(NN),CS3(NN)
      DIMENSION QPL(NBCN),QPLM1(NBCN),UBC(NBCN),VBC(NBCN),WBC(NBCN)
C     -----------------------------------------------------------------

C     --- STORE TIME INFORMATION ---
      WRITE(66,100) TSEC,DELTP,DELTU,DLTUM1
  100 FORMAT(4D20.10)

C     --- STORE SOLUTION ---
      WRITE(66,110) (HVEC(I),I=1,NN)
      WRITE(66,110) (C1(I),I=1,NN)
      WRITE(66,110) (C2(I),I=1,NN)
      WRITE(66,110) (C4(I),I=1,NN)
      WRITE(66,110) (HM1(I),I=1,NN)
      WRITE(66,110) (UM1(I),I=1,NN)
      WRITE(66,110) (VM1(I),I=1,NN)
      WRITE(66,110) (WM1(I),I=1,NN)
      WRITE(66,110) (CS1(I),I=1,NN)
      WRITE(66,110) (CS3(I),I=1,NN)
      WRITE(66,110) (QPL(IPU),IPU=1,NBCN)
      WRITE(66,110) (QPLM1(IPU),IPU=1,NBCN)
      WRITE(66,110) (UBC(IPU),IPU=1,NBCN)
      WRITE(66,110) (VBC(IPU),IPU=1,NBCN)
      WRITE(66,110) (WBC(IPU),IPU=1,NBCN)
  110 FORMAT(4(1PD20.13))


      ENDFILE(66)
      RETURN
      END
```

☆U.S. GOVERNMENT PRINTING OFFICE: 1986- 491-405:40002