Note:

Some of the graphics in this document are not on the actual page. The graphic has been scanned and inserted at the end of the document. The following instructions will link you to the graphics.

Click inside the red square that surrounds the figure number.

Figure "no."

This will link you to the graphic.

To return to the original page, click the << button in the Acrobat Reader tool bar.

EPA/600/R-94/039b
March 1995

The Hydrocarbon Spill Screening Model (HSSM) Volume 2: Theoretical Background and Source Codes

by

Randall J. Charbeneau
Center for Research in Water Resources
The University of Texas at Austin
Austin, Texas 78712

James W. Weaver and Bob K. Lien
Robert S. Kerr Environmental Research Laboratory
United States Environmental Protection Agency
Ada, Oklahoma 74820

Robert S. Kerr Environmental Research Laboratory
Office of Research and Development
U.S. Environmental Protection Agency
Ada, Oklahoma 74820

Disclaimer

The research described in this report has been supported by the United States Environmental Protection Agency under cooperative agreement CR-813080 to The University of Texas at Austin, and by direct support of the EPA authors, it has been subjected to Agency review, and it has been approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

All research projects funded by the U.S. Environmental Protection Agency that make conclusions or recommendations based on environmentally related measurements are required to participate in the Agency Quality Assurance Program. The experimental work described in section 2.7 was conducted under an approved Quality Assurance Project Plan and the procedures specified therein were used. Information on the plan and documentation of the quality assurance activities are available from the second author.

The computer program described within this report simulates the behavior of water-immiscible contaminants (LNAPLs: Non-Aqueous Phase Liquids) in idealized subsurface systems. The approaches described are not suited for application to heterogeneous geological formations nor are they applicable to any other scenario other than that described herein. The model is intended to provide order-of-magnitude estimates of contamination levels only. The full model has not been verified by comparison with either lab or field studies. Therefore the EPA does not endorse the use of this computer program for any specific purpose. As in the case of any subsurface investigation, the scientific and engineering judgement of the model user is of paramount importance. Any model results should be subjected to thorough analysis. In this user's guide, typical values are given for various parameters. These are provided for illustrative purposes only.

When available, the software described in this document is supplied on an "as-is" basis without guarantee or warranty of any kind, expressed or implied. Neither the United States Government (United States Environmental Protection Agency, Robert S. Kerr Environmental Research Laboratory), The University of Texas at Austin, nor any of the authors accept any liability resulting from the use of this software.

Foreword

EPA is charged by Congress to protect the Nation's land, air and water systems. Under a mandate of national environmental laws focused on air and water quality, solid waste management and the control of toxic substances, pesticides, noise and radiation, the Agency strives to formulate and implement actions which lead to a compatible balance between human activities and the ability of natural systems to support and nurture life.

The Robert S. Kerr Environmental Research Laboratory is the Agency's center of expertise for investigation of the soil and subsurface environment. Personnel at the Laboratory are responsible for management of research programs to: (a) determine the fate, transport and transformation rates of pollutants in the soil, the unsaturated and the saturated zones of the subsurface environment; (b) define the processes to be used in characterizing the soil and subsurface environments as a receptor of pollutants; (c) develop techniques for predicting the effect of pollutants on ground water, soil, and indigenous organisms; and (d) define and demonstrate the applicability of using natural processes, indigenous to the soil and subsurface environment, for the protection of this resource.

One of the most common, yet complex, class of subsurface contaminants is the light nonaqueous phase liquids (LNAPLs). Although the LNAPL itself remains distinct from the subsurface water, chemical constituents of the LNAPL can cause serious ground-water contamination. Since a number of phenomena and parameters interact to determine contaminant concentrations at the receptor points, models are needed to estimate the impacts of LNAPL releases on ground water. This volume describes the theoretical basis for the Hydrocarbon Spill Screening Model (HSSM) which is intended to simulate release of an LNAPL. The intent of the model is to provide a practical tool which is easy to apply and runs rapidly on personal computers.

Clinton W. Hall, Director
Robert S. Kerr Environmental
Research Laboratory

Abstract

A screening model for subsurface release of a nonaqueous phase liquid which is less dense than water (LNAPL) is presented. The model conceptualizes the release as consisting of 1) vertical transport from near the surface to the capillary fringe, 2) radial spreading of an LNAPL lens through the capillary fringe and dissolution of LNAPL constituents into a water table aquifer, and 3) transport in the flowing ground water to a potential exposure location. Each component of the conceptual model is treated as a distinct process by separate models. This report describes the modules for the vadose zone, lateral spreading at the water table and dissolution of constituents into the aquifer, and aquifer transport of the dissolved constituents to receptor points. Spreading of the hydrocarbon lens and dissolution of hydrocarbon constituents are transient phenomena, and the aquifer transport model must be capable of addressing a time-variable source term. This is incorporated through application of Duhamel's principle to a gaussian-source plume model. The resulting screening model is computationally efficient and has only moderate parameterization requirements.

CONTENTS

List of Figures

List of Symbols

Latin

A	Area
a_v	Aquifer vertical dispersivity
B	Bulk partition coefficient defined by equation **(25)**
B(ω)	Time varying function of concentration at boundary (equation **(86)**)
b	Aquifer saturated thickness
b_o	Observation well thickness of NAPL
C	Nondimensional concentration (equation **(81)**)
c, c_w	Concentration of the constituent in the water phase
c_m	Maximum concentration
c_o	Concentration of the constituent in the NAPL phase
$c_{o(soil)}$	Constituent concentration in the NAPL in equilibrium with the soil
$c_{o(surf.)}$	Constituent concentration in the released NAPL
c_s	Sorbed phase concentration of the constituent
$c_{w(initial)}$	Initial water phase concentration of the constituent
c_{wo}	Equilibrium constituent concentration for water in contact with the NAPL
D	Nondimensional dispersion coefficient (equation **(81)**)
D_o	Formation free-product thickness
D_L	Longitudinal dispersion coefficient
D_T	Transverse horizontal dispersion coefficient
D_V	Vertical dispersion coefficient
d_{pz}	Depth of the NAPL contaminated zone
E	Constant defined by equation **(93)**
erf()	Error function
erfc()	Complementary error function
F_1	OILENS NAPL source head function
F_2	OILENS lens radius function
f_n	n^{th} function to be solved for a general Runge-Kutta scheme
G	Known constant that relates lens source heads and radii at different times
g	Acceleration of gravity
H	Sum of the NAPL head terms in KOPT; Total penetration depth of leachate in TSGPLUME
H_{adv}	Penetration depth due to vertical advection of water entering the aquifer
H_{dis}	Penetration depth due to vertical dispersion in the aquifer
H_f	NAPL head at the NAPL front
H_s	NAPL head at the surface
h	Pressure head in KOPT (equation **(16)**)
h_{cao}	Air-NAPL capillary pressure head
h_{caw}	Air-water capillary pressure head
h_{cij}	Capillary pressure head or capillary rise for the i-j fluid pair
h_{ceao}	Air-NAPL entry head
h_{ceaw}	Air-water entry head
h_o	NAPL head at a given location
h_{os}	NAPL head in the lens below the source (**Figure 16**)
h_{ow}	NAPL-water capillary pressure head
I_d	Value of the integral in equation **(56)**
I_r	Rate of infiltration outside the facility
J	Advective flux
J_c	Mass flux of the chemical constituent
K	Hydraulic conductivity

K_{ei}	Effective conductivity to fluid i
K_{eo}	Effective conductivity to NAPL
K_{si}	Saturated conductivity to fluid i
K_{so}	Saturated conductivity to NAPL
K_{sw}	Saturated conductivity to water
k	Intrinsic permeability
k_d	Soil/water partition coefficient
k_i	Fluid i/water partition coefficient
k_o	NAPL/water partition coefficient
k_{ri}	Relative permeability to fluid i
k_{rw}	Relative permeability to water
k_{ro}	Relative permeability to NAPL
L	Length (in direction of ground water flow) of a surface facility
L(y)	Length of a chord of a circle
M_t	Total constituent mass in the NAPL lens
m_c	Mass of constituent per unit volume
m	Mass flow rate
m_{diss}	Total constituent flux into the aquifer
m_{infil}	Mass loss rate to the aquifer (equation **(48)**)
m_{source}	Ground water source term from the NAPL lens
n	Time level
p_i	KOPT model parameter
Q_{KOPT}	KOPT-determined NAPL inflow rate (**Figure 16**)
Q_{loss}	Loss rate of NAPL from source cylinder (**Figure 16**)
Q_{out}	NAPL loss rate due to dissolution and trapping (equation **(43)**)
Q_{radial}	Lateral flow from the source cylinder (**Figure 16**)
q	Darcy velocity in aquifer below facility (equation **(67)**)
q_i	Flux of fluid i
q_o	NAPL flux
q_w	Water flux
q_{wi}	Volume flux of infiltrating rainfall
R	NAPL lens radius
R_d	Retardation factor
R_s	NAPL source radius
R_t	Lens radius
r	Radius
S_{ar}	Residual air saturation
S_i	Saturation of fluid i
S_{ir}	Residual saturation of fluid i
S_o	NAPL saturation
$S_{o(max)}$	Maximum NAPL saturation in KOPT solution
S_{or}	Residual NAPL saturation
S_w	Water saturation
$S_{w(avg)}$	Water saturation associated with the average annual recharge rate
S_{wr}	Residual water saturation
SC_i	First order sensitivity coefficient for the KOPT model
T	Nondimensional time (equation **(81)**)
t	Time
t_o	Time origin
V_C	Volume of source cylinder beneath the surface source (equation **(90)**)
V_L	NAPL volume in the spreading lens
V_o	NAPL volume incorporated into the soil in KOPT
V_T	Total lens volume (including LNAPL, water and soil)
V_{vz}	Total lens volume (including LNAPL, water and soil) in the vadose zone

V_{sz}	Total lens volume (including LNAPL, water and soil) in the saturated zone
v	Seepage velocity in the aquifer
W	Width (across the direction of ground water flow) of a surface facility
w	Variable of integration in equations **(55)** and **(95)**
w_R,w_{Rs}	Limits of integration in equation **(95)**
X	Nondimensional x coordinate (equation **(81)**)
x	Distance from upgradient edge of NAPL lens
Y	Nondimensional y coordinate (equation **(81)**)
y_1,y_2,...,y_n	1st through nth independent variable
Z	Nondimensional z coordinate (equation **(81)**)
z	Depth
z_{ao}	Level of the air-NAPL interface
z_{aw}	Level of the air-water interface (water table) in the absence of NAPL
z_{ow}	Level of the NAPL-water interface
z_f	Front depth in KOPT

Greek

β	Density term in equation **(36)**
ΔM_L	Constituent mass loss from NAPL lens during a time step
ΔV_L	Volume of free product (LNAPL) that becomes trapped in a time step
ΔV_R	Change in total lens volume (LNAPL, water and soil)
δ m(y)	Increment of mass flux into the aquifer
ϵ	Brooks and Corey relative permeability exponent
η	Porosity
Λ	Nondimensional effective decay coefficient in TSGPLUME (equation **(81)**)
λ	Brooks and Corey capillary pressure exponent; Decay coefficient in TSGPLUME
λ^*	Effective decay coefficient in TSGPLUME
μ_i	Dynamic viscosity of fluid i
μ_o	Dynamic viscosity of the NAPL
ρ_b	Bulk density
ρ_i	Density of fluid i
ρ_o	Density of NAPL
ρ_w	Density of water
σ	Standard deviation
σ_{aw}	Water surface tension
σ_o,σ_{ao}	NAPL surface tension
σ_{ow}	NAPL-water interfacial tension
θ_o	Volumetric NAPL content
θ_{orv}	Volumetric residual NAPL content in the vadose zone
θ_{ors}	Volumetric residual NAPL content in the aquifer

List of Abbreviations and Acronyms

CSMoS	Center for Subsurface Modeling Support
HSSM	Hydrocarbon Spill Screening Model
HSSM-KO	Computer code that implements KOPT and OILENS
HSSM-T	Computer code that implements TSGPLUME
KOPT	Kinematic Oily Pollutant Transport (vadose zone portion of HSSM)
LNAPL	Lighter-that-water nonaqueous phase liquid
NAPL	Nonaqueous phase liquid
OILENS	HSSM Module for NAPL lens motion and chemical dissolution into the aquifer
RSKERL	Robert S. Kerr Environmental Research Laboratory
TSGPLUME	Transient source gaussian plume model (aquifer module of HSSM)
USEPA	United States Environmental Protection Agency

Acknowledgement

The authors wish to express their appreciation to Julia Mead and Sarah Hendrickson for the simulation runs and graphics presented in Section 5 on parameter variation.

Section 1 Introduction

The subsurface release of a liquid hydrocarbon from a spill or a leaking tank or pipeline is of concern because the hydrocarbon phase or its constituents may migrate through the subsurface environment to contaminate drinking water supplies or fresh water resources. With a near-surface release, the hydrocarbon must migrate vertically through the vadose zone before reaching ground water. Capillary forces play an important role in determining the transport times and rates for constituents reaching the water table. If the nonaqueous phase accumulates at the water table in sufficient quantities, it will build sufficient head to cause radial spreading. At the same time, its constituents will dissolve into ground water flowing beneath the lens and be transported to potential downgradient receptor locations. A detailed analysis of such a release using site-specific models would require a significant computational expense as well as vast resources for characterization of required physical and chemical parameters.

For many applications such resources are not available, especially during the initial phases of a site investigation or in the analysis of impacts from potential releases. In such cases, screening models provide appropriate tools for investigation. These models are based on a simplified interpretation of the hydrogeology, including generally the assumption of uniform aquifer flow in a specified direction and homogeneous conditions for other parameters. These assumptions allow the use of analytic or semi-analytic solutions to the transport problem. Analytic solutions have the advantage of simplicity and ease of computation. Screening models may be developed which capture, in an approximate sense, many of the important factors and processes which control the fate and behavior of subsurface contaminants. Such models may then be used to evaluate the behavior of large numbers of chemicals in the environment.

The Hydrocarbon Spill Screening Model (HSSM) described herein is such a model. The basic scenario of the model is shown in **Figure 1**. A hydrocarbon is released near the ground surface and transported downward through the vadose zone to the water table. At the water table a hydrocarbon lens forms and spreads laterally. Constituents from the hydrocarbon lens dissolve into ground water flowing beneath the lens, creating a plume which may contaminate downgradient wells or other exposure points. HSSM may be used to estimate the effects of LNAPL loading, partition coefficients, ground water flow velocities, etc., on pollutant transport. Since numerous approximations are used for developing the model, the model results must also be viewed as approximations.

1.1 Model Overview

The following paragraphs provide an overview of the HSSM model and discuss some of the model assumptions and limitations. The spill or release of the LNAPL phase may be simulated in three basic ways. First is a release of a known LNAPL flux for a specified duration. The release occurs at the ground surface. Based on an approximate capillary suction relationship, some of the LNAPL may run off at the surface if the flux exceeds the maximum effective LNAPL infiltration capacity. Second, a constant depth of ponded LNAPL, for a known duration, may also be specified. This case represents a slowly leaking tank, or a leaking tank within an embankment. A generalization of this scenario is that a constant depth may by specified for a given duration, with the subsequent period modeled based on continuity between the remaining ponded depth and the cumulative infiltration. This scenario, with a short initial constant ponded depth duration, may be appropriate for modeling an extreme containment failure for a tank within a contained or bermed area. Lastly, a known volume of LNAPL may be placed over a specified depth of the soil. This last scenario represents either a land treatment operation or a landfill containing a known amount of contaminants at the beginning of the simulation.

Transport of the NAPL through the unsaturated zone is assumed to be one-dimensional. Capillary pressure gradients are neglected except as they influence the infiltration of NAPL into the soil. The resulting equations for NAPL flow are hyperbolic and are solved by the generalized method of characteristics. When relatively large amounts of LNAPL are released, downward transport of the LNAPL (say gasoline) is the primary

Figure 1 Schematic view of the Hydrocarbon Spill Screening Model (HSSM) scenario

mechanism for downward transport of hydrophobic chemicals (e.g., benzene, toluene, and xylene). Assumptions concerning aquifer recharge are relatively unimportant in this case. If a large enough volume is supplied, the LNAPL reaches the water table. If sufficient head is available, the water table is displaced downward, lateral spreading begins, and the oil lens part of the model is triggered. Spreading is assumed to be radial, and the thickness of the lens is determined by buoyancy only (Ghyben-Herzberg relations). The shape of the lens is given by the Dupuit assumptions, where the flow is assumed horizontal and the gradient is independent of depth.

The LNAPL is treated as a two-component mixture. The LNAPL itself is assumed to be soluble in water and sorbing. Due to the effects of the recharge water and contact with the ground water, the LNAPL may be dissolved. The LNAPL's transport properties (density, viscosity, capillary pressure, relative permeability),

however, are assumed to be unchanging. The second component is a chemical constituent which can partition between the LNAPL phase, water phase and the soil. This constituent of the LNAPL is considered the primary contaminant of interest. The mass flux of the second constituent into the aquifer comes from recharge water being contaminated by contact with the lens and from dissolution occurring as ground water flows under the lens. The concentration of the chemical in the aquifer is limited by its effective solubility in water.

The aquifer transport of the dissolved contaminant is simulated by using a two-dimensional, vertically averaged analytic solution of the advection-dispersion equation. The vertical extent of the contaminant is estimated from the recharge rate, ground water seepage velocity and vertical dispersivity, rather than assuming the contaminant is distributed over the entire aquifer thickness. The boundary conditions are placed at the downgradient edge of the lens and take the form of a gaussian distribution with the peak directly downgradient of the center of the lens. The peak concentration of the gaussian distribution adjusts through time so that the simulated mass flux from the lens equals that into the aquifer. Although the size of the lens varies with time, a constant representative lens size is used for the aquifer source condition. In many cases, the lens reaches its maximum size rather rapidly compared with the transport in the aquifer, so that the use of the maximum lens size will not introduce large errors. The model contains an option for choosing the effective size of the lens based on its size at the time when the mass flux to the aquifer is greatest. This option may be appropriate for releases of viscous hydrocarbon liquids.

The required input parameters include parameters specifying the type, extent and magnitude of the LNAPL release, the residual oil contents for the unsaturated and saturated zones, the residual water content of the oil lens, the transport properties of the water and LNAPL (density, viscosity, surface tension), the aquifer and soil water retention characteristics (vertical and horizontal hydraulic conductivities, porosity, irreducible water content, pore size distribution index, and air entry head), the dissolved constituent characteristics (initial concentration within the LNAPL, aqueous solubility, and the soil-water and oil-water partition coefficients), and the aquifer transport characteristics (vertical, longitudinal and transverse dispersivities, hydraulic gradient, half-life of the constituent within the aquifer). Other parameters control the simulation and locations where the LNAPL chemical constituent concentrations are calculated. Specific information on running the model is presented in *The Hydrocarbon Spill Screening Model (HSSM) Volume 1: User's Guide* (Weaver et al., 1994).

1.2 Obtaining a Copy of HSSM

HSSM is available from the Center for Subsurface Modeling Support (CSMoS) at the Robert S. Kerr Environmental Research Laboratory (RSKERL) at Ada, Oklahoma. CSMoS distributes software and documentation free-of-charge through a diskette exchange program and provides technical support for the codes they distribute. To obtain the HSSM software and user documentation send a letter of request along with one high density 3.5" formatted diskette to the following address:

Center for Subsurface Modeling Support
Robert S. Kerr Environmental Research Laboratory
United States Environmental Protection Agency
P.O. Box 1198
Ada, Oklahoma 74820
Voice: 405-436-8586
FAX: 405-436-8529

Please indicate if the DOS or Windows version is needed. If both interfaces are needed, enclose two formatted diskettes.

The complete HSSM package consists of the documents

- *The Hydrocarbon Spill Screening Model (HSSM) Volume 1: User's Guide,*
- *The Hydrocarbon Spill Screening Model (HSSM) Volume 2: Theoretical Background and Source*

Codes,

and the two high density 3.5" diskettes. The diskettes contain:

For Windows:

□ diskette HSSM-1-w The Windows Interface, HSSM-WIN

For DOS:

□ diskette HSSM-1-d The DOS Interface, HSSM-DOS

HSSM and the user documentation are in the public domain. They may be freely distributed or copied by anyone.

1.3 Volume 2 Organization

The remainder of the report describes the three *modules* which compose HSSM. These modules each implement one part of the release scenario discussed above. First, in Section 2, the vadose zone module is presented. The approach taken in the vadose zone is to combine Green-Ampt and kinematic wave theory to simulate the flow and transport of the LNAPL. This module is called Kinematic Oily Pollutant Transport (KOPT). The second module, described in Section 3 implements the capillary fringe and dissolution scenarios and is called OILENS. OILENS uses the density difference between water and the LNAPL and the Dupuit assumptions to simulate the growth and decay of the lens at the capillary fringe. The third module is called the Transient Source Gaussian Plume model (TSGPLUME) and is described in Section 4. An example simulation is presented in Section 5 that illustrates the effects of distance to the receptor point and chemical properties on the estimated aquifer concentrations. Section 6 summarizes the document. The complete FORTRAN source codes for HSSM and the HSSM utilities are presented in Appendix 3.

Section 2 Flow and Transport in the Vadose Zone

The method of characteristics has been applied to simplified versions of multiphase, multicomponent transport problems by neglecting diffusion-like terms. By applying fractional flow theory, the problem can be written as a system of hyperbolic conservation laws. Helfferich (1981, 1986) developed a comprehensive theory for problems with uniform initial conditions and constant boundary conditions. The theory includes as special cases the Buckley-Leverett model of two-phase flow (Buckley and Leverett, 1942), and the three-phase models of Dougherty and Sheldon (1964) and Pope et al. (1978). Corapcioglu and Hossain (1990) applied the Buckley-Leverett approach to the flow of DNAPLs in two-dimensional aquifer systems. In mathematical terms, systems of hyperbolic conservation equations with these boundary and initial conditions are called Riemann problems (Smoller, 1983). As noted by Weaver (1991), such solutions potentially have direct application to vadose zone transport only where the release of the fluids is unending.

Several solutions of simplified, multiphase governing equations have been developed for one-dimensional NAPL infiltration, based upon the following restrictive assumptions. Richards' (1931) approach to soil moisture was to formulate only a mass conservation equation for the water phase. For this to be a valid approximation, the flow of the air must not impede the flow of the water. One accounts for the presence of the air phase by the usage of an appropriate relative permeability function. Further, if the saturation of water is uniform and remains so, the continuity equation for the water phase is eliminated. From these assumptions, Mull (1971, 1978), Raisbeck and Mohtadi (1974), Dracos (1978), Reible et al., (1990) and El-Kadi (1992) developed models for NAPL flow assuming that the NAPL fills a fixed portion of the available pore space in a homogeneous medium. The models developed by Raisbeck and Mohtadi (1974) and Dracos (1978) share the limitation that they cannot simulate unsteady NAPL drainage in the soil after the release ends. The models of Mull (1971, 1978) and Reible et al. (1990) simulate drainage with arbitrary assumptions concerning the profile shape. Mull's model uses a series of rectangular profiles while Reible et al. (1990) assume a zone of residual NAPL saturation behind a NAPL body moving within the profile. El-Kadi (1992) extended the approach of Mull to multiple dimensions. During the redistribution period, however, the NAPL saturation may be non-uniform. Also, these models have not included transport of soluble constituents of the NAPL phase. Ryan and Cohen (1991) developed a one-dimensional, finite difference, front tracking model which can simulate non-uniform saturation profiles and chemical transport.

Figure 2 Schematic comparison of sharp and diffuse fronts

The model presented herein extends the simple one-dimensional model approach as it allows nonuniform saturations to be modeled after infiltration ceases and includes advective transport of a partitionable constituent of the NAPL phase. The resulting model is named KOPT for kinematic oily pollutant transport to emphasize its reliance on kinematic wave theory. The objective of this development is to provide a computationally efficient screening model for NAPL infiltration and redistribution. In general, screening models allow comparisons of the effects of various parameters (e.g., hydraulic conductivity, partition coefficients, etc.) on the flow and transport of NAPLs in an idealized setting. By utilizing a semi-analytic method of characteristics approach, a numerical solution technique is avoided. The resulting efficiency in running the model is achieved primarily because the capillary pressure gradient is neglected. This assumption leads to approximate hyperbolic governing equations which can be solved by the generalized method of characteristics (Smoller, 1983). One major effect of this assumption on the simulation results is that the leading edge of the L-NAPL moving into the soil is idealized as a sharp front (**Figure 2**). Method of characteristics solutions have been developed for one-dimensional water flow in the vadose zone by Sisson et al. (1980), Smith (1983), and Charbeneau (1984, 1991). Where the front spreads, Charbeneau (1984), for example, presented a theoretical proof that the mean displacement speed of the sharp and true fronts is the same. Smith (1983) presented a numerical result for water flow showing that a numerical solution of Richards' equation was tracked by a sharp-front solution. The flow visualization experiment presented below demonstrates the ability of the KOPT model to match experimental results to a certain degree of accuracy.

The theory of kinematic waves was presented by Lighthill and Whitham (1955) for flood waves in rivers. They note that whenever an approximate functional relation exists at each point between the flux and concentration (analogously in multiphase flow, flux and saturation), then the wave motion follows directly from the continuity equation. For most cases of interest, the functional relationship is nonlinear, and the kinematic

waves are either regions of constant state, self-spreading, or self-sharpening. In the multiphase flow problem, the region of constant state corresponds to a uniform saturation profile. The self-spreading wave corresponds to a region of internal drainage and redistribution occurring after the end of the NAPL release. The self-sharpening wave corresponds to a wetting front moving into the profile. Each of these features of the solution will be illustrated by the model results presented in **Figure 3** and **Figure 4**. Although the shape of the wetting front is determined by the capillary gradient, the kinematic model is able to move the sharp-front representation of the wetting front downward at the correct speed so that mass is conserved. For these reasons, kinematic models are able to represent the essential features of nonlinear wave phenomena.

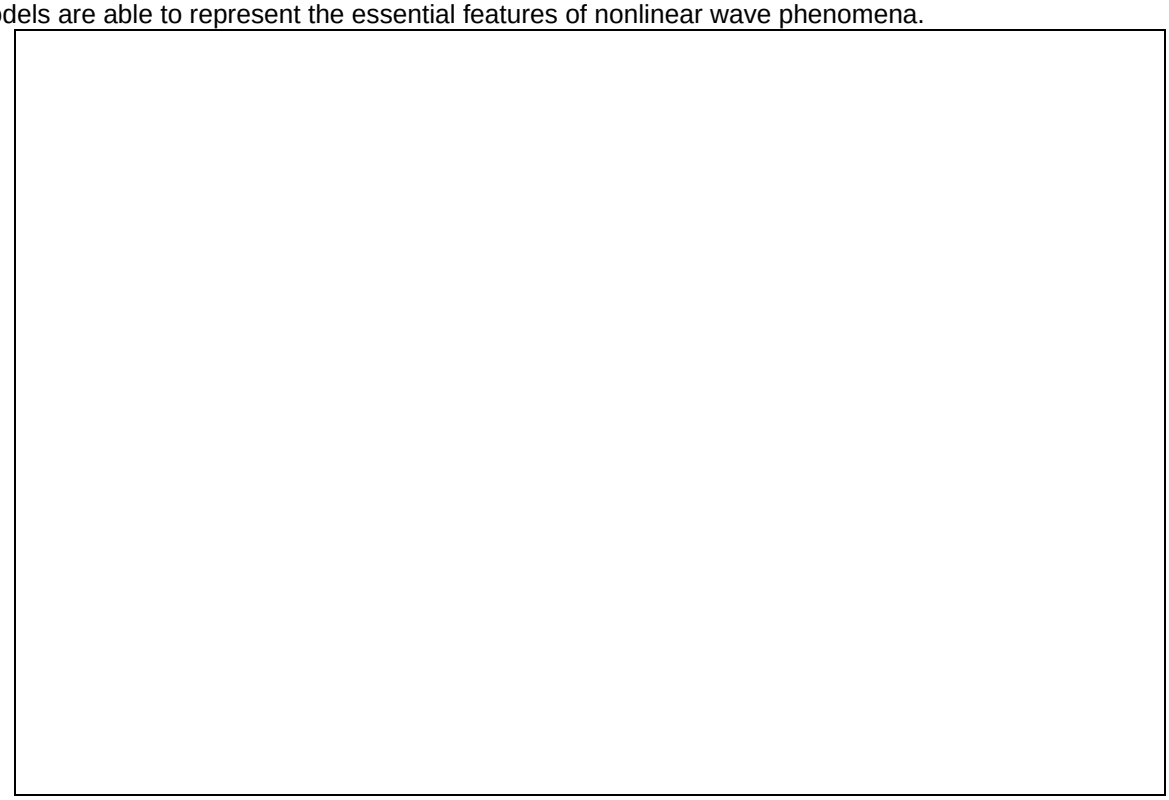

Figure 3 Base characteristic plane

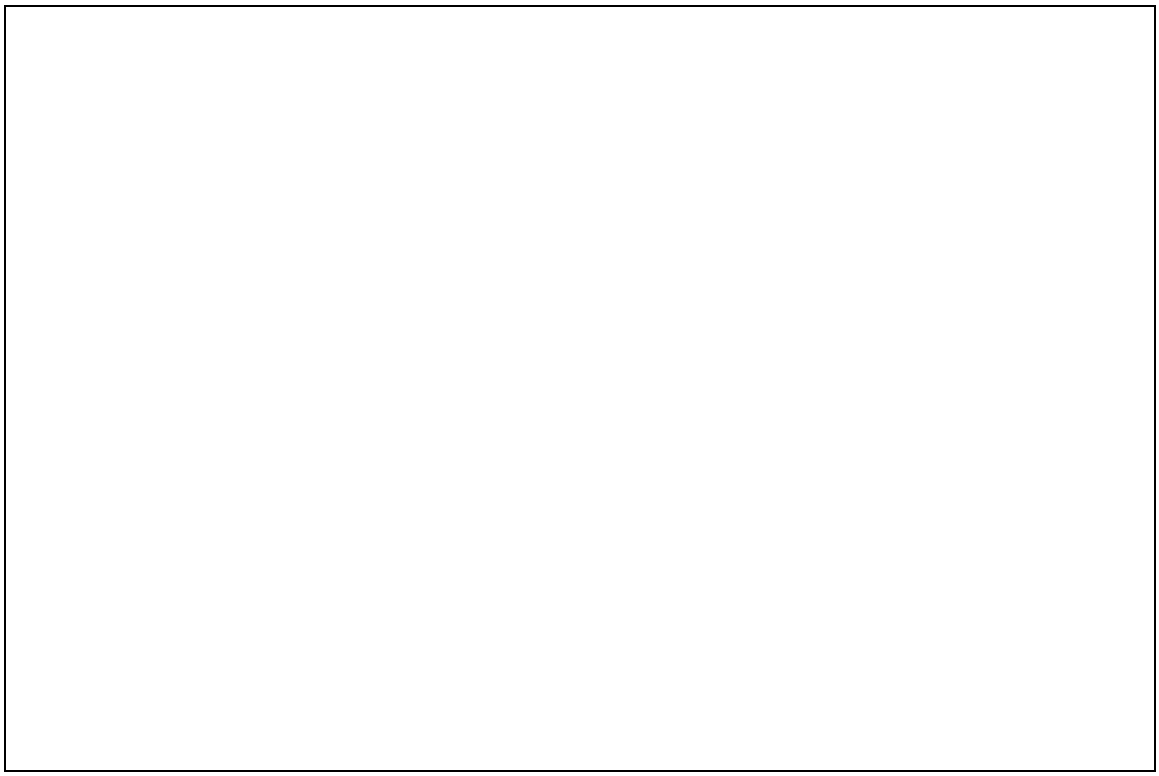

Figure 4 Total liquid profiles

2.1 KOPT Model Framework

Throughout this document, the liquids are assumed incompressible and the medium non-deformable. The NAPL is assumed to be composed of two components. The first is the water immiscible phase which acts as a carrier for the second component. The second component is a chemical of environmental concern. The concentration of dissolved constituents is assumed to have no effect upon the fluid transport properties including densities, dynamic viscosities, and surface tensions.

Although actual flow in the vadose zone is three-dimensional, the model treats transport through the unsaturated zone as being one-dimensional downward. Gravity, which is the only driving force for kinematic model, acts downward, though lateral spreading of the NAPL may occur because of capillary forces. Spreading may also be caused by heterogeneity, as layering may impede flow. For situations where the NAPL is applied over relatively large areas, the flow becomes nearly one-dimensional in the center. For contaminant sources that are of small areal extent, the lateral transport of contaminants may be significant, and the assumption of one-dimensional flow is less applicable. From the point of view of ground-water contamination, however, one-dimensional modeling leads to a conservative model as all of the NAPL is assumed to move downward and potentially reach the water table.

2.2 Derivation of the Vadose Zone NAPL Flow Model

Kinematic models are unit gradient models where drainage occurs due to the force of gravity and capillary pressure gradients are neglected. The downward Darcy volume flux of phase i is

$$q_i = K_{ei} = K_{si}\, k_{ri} \quad \textbf{(1)}$$

where K_{ei} is called effective hydraulic conductivity of the medium to fluid i, K_{si} is the fully saturated conductivity of i, and k_{ri} is the relative permeability of the medium to fluid i. The fully saturated conductivity is related to properties of the fluids and the porous medium through

$$K_{si} = \frac{k \rho_i g}{\mu_i} \quad \textbf{(2)}$$

where k is the intrinsic permeability of the medium, ρ_i is the density of the fluid, g is the acceleration due to gravity, and μ_i is the dynamic viscosity of fluid i.

In order to estimate the Darcy fluxes in equation **(1)** , expressions for the relative permeabilities are needed. One way to develop these relationships is to begin with conceptual models of the porous medium, presumed distributions of fluids within the medium, and a solution of laminar flow through the medium (Bear, 1972). The Burdine (Burdine, 1953, Wylie and Gardner, 1958) equations form one such model. Using the Brooks and Corey (1964) model of the capillary pressure to evaluate the Burdine equations yields the following model of drainage relative permeability for water and NAPL:

$$k_{rw} = \left(\frac{S_w - S_{wr}}{1 - S_{wr}} \right)^{\epsilon} \quad \textbf{(3)}$$

$$k_{ro} = \left(\frac{S_o - S_{or}}{1 - S_{or}} \right)^2 \left[\left(\frac{S_o + S_w - S_{wr}}{1 - S_{wr}} \right)^{\epsilon - 2} - \left(\frac{S_w - S_{wr}}{1 - S_{wr}} \right)^{\epsilon - 2} \right] \qquad S_o > S_{or} \quad \textbf{(4)}$$

$$k_{ro} = 0 \qquad S_o < S_{or} \quad \textbf{(5)}$$

$$\epsilon = \frac{(2 + 3\lambda)}{\lambda} \quad \textbf{(6)}$$

where λ is called the pore size distribution index, S_i is the saturation (saturation is defined as the percent of the pore space filled by a given fluid) of phase i, and S_{ir} is the residual saturation of phase i. The pore size index and residual water saturation are obtained through measurement of the capillary pressure curve. The scaling of the NAPL saturation in equation **(4)** is used so that the NAPL relative permeability is equal to one only when the NAPL entirely fills the pore space (i.e., $S_{wr} = 0$) and so that the NAPL relative permeability is zero when the NAPL saturation is less than or equal to its residual. Implicit in the use of equation **(4)** is the assumption that NAPL has previously displaced air; and that during subsequent drainage, some NAPL is retained as a trapped phase. S_{or} is introduced as an empirical parameter to represent the retention of NAPL in the vadose zone after the passage of a NAPL infiltration event. If no NAPL is retained, then S_{or} can be set equal to zero. Wilson et

al. (1990) present a method for measuring vadose zone residual NAPL saturations in the presence of a residual water saturation. These authors further suggest that residual NAPL saturations are dependent upon the saturation history of the system and would be reduced in the presence of higher water saturations. The usage of equation **(4)** for imbibition at the NAPL front is discussed below. Further refinement of the relative permeability function is not proposed for the KOPT model, because the model is intended for screening calculations where large uncertainty is expected in all the model parameters.

Water is assumed to occupy a fixed, uniform portion of the pore space. This assumption eliminates the mass conservation equation for water from the model. This approach is taken because the temporal distribution of rainfall is required to simulate the time varying moisture profile. This requirement goes beyond the anticipated data available for screening purposes. Richards' assumption is then used to eliminate the mass conservation equation for the air phase in accordance with common soil science practice. This approximation is acceptable as long as pressure does not build up in the air phase (Youngs and Peck, 1964).

The water saturation is calculated from the average annual recharge rate by assuming that maximum effective conductivity of the soil is greater than the recharge rate. A kinematic model is then appropriate; and the recharge rate and water saturation are related through the relative permeability function for the water. If q_{wi} is the average annual recharge flux, then equations **(1)** and **(3)** give the resulting water saturation as

$$S_{w(avg)} = S_{wr} + (1 - S_{wr}) \left(\frac{q_{wi}}{K_{sw}} \right)^{1/\varepsilon} \qquad \textbf{(7)}$$

The calculated $S_{w(avg)}$ is used as the water saturation for the entire profile and is maintained by the assumed recharge through the system. Equation **(7)** replaces the phase conservation equation for water.

The presence of air in the vadose zone is accounted for via the usage of a three-phase relative permeability function and by assuming that there is a trapped air saturation, which limits the maximum effective conductivity, K_{eo}, of the NAPL phase. Bouwer (1966) reported that the maximum effective hydraulic conductivity to water is 40% to 60% of the saturated hydraulic conductivity, K_{sw}. In the present work, 50% K_{sw} is used to determine the trapped air saturation. In equation **(3)** the relative permeability to water, k_{rw}, is set equal to 0.5. S_{ar} is then taken as the air saturation which would occur with that amount of soil water and is calculated by

$$S_{ar} = 1 - S_{wr} - (1 - S_{wr})(0.5)^{1/\varepsilon} \qquad \textbf{(8)}$$

In equation **(8)**, air and water are assumed to fill the pore space; when the NAPL enters the soil, S_{wr} and S_{ar} are assumed to be unchanged.

With equation **(7)** and Richards' assumption, the only phase equation to be solved is that for the NAPL. For gravity driven flow, the NAPL flux q_o from equation (1) becomes

$$q_o = K_{eo}(S_o, S_{w(avg)}) \qquad \textbf{(9)}$$

Since $S_w = S_{w(avg)}$ is constant, K_{eo} is a function of S_o, and $S_{w(avg)}$ serves as a parameter. Using a single-valued, relative permeability function with known $S_{w(avg)}$, the continuity equation for NAPL can be written in terms of its unknown saturation

$$\eta \frac{\partial S_o}{\partial t} + \frac{dK_{eo}(S_o, S_{w(avg)})}{dS_o} \frac{\partial S_o}{\partial z} = 0 \qquad \textbf{(10)}$$

where η is the porosity and z is directed positive downward. The relative permeability function (equation **(4)**) is used in equation **(10)**.

The solution then may proceed as follows. Where the NAPL saturation distribution varies continuously, the classical method of characteristics (MOC) solution of equation **(10)** is

$$\frac{dS_o}{dt} = 0 \qquad \textbf{(11)}$$

along characteristics given by

$$\frac{dz}{dt} = \frac{1}{\eta} \frac{dK_{eo}(S_o, S_{w(avg)})}{dS_o} \qquad \textbf{(12)}$$

The solution is valid only where the derivatives appearing in equation **(10)** exist at each point of the solution domain (e.g., Rozdestvenskii and Janenko, 1980). For most cases, S_o is fixed along the length of the characteristic line, since there is no source term in the governing equation **(10)**. Conservation of mass results in straight characteristics, as the slope dz/dt is constant.

When the saturation derivatives fail to exist in the solution, integral equations are used to find solutions that are called generalized or weak solutions. An integral form of the continuity equation applied to a control volume around the front can be integrated to give (e.g., Charbeneau, 1984),

$$\frac{dz}{dt} = \frac{q_{o(1)} - q_{o(2)}}{\eta\,(S_{o1} - S_{o2})} = \frac{K_{eo(1)} - K_{eo(2)}}{\eta\,(S_{o1} - S_{o2})} \qquad \textbf{(13)}$$

where the subscripts 1 and 2 refer to values at locations on either side of the front as indicated in **Figure 2**. The form of equation **(13)** which uses the effective conductivities (K_{eo}) is applicable for kinematic flows. When the NAPL invades a pristine medium, as assumed in KOPT, S_{o2} and thus $K_{eo(2)}$ are equal to zero.

Equation **(13)** is the solution of equation **(10)** in the weak or integral sense. This is the well-known jump condition. Discontinuities form in the solution domain because either characteristics cross, which is a physically impossible situation, or because the boundary data are discontinuous and the $\partial K/\partial S$ function does not cause smoothing of the front (Weaver, 1991). In KOPT, the latter condition is applicable as the NAPL flux (or ponding depth) is assumed to increase discontinuously from zero to the initial level. $\partial K/\partial S$ is of greater magnitude for the high saturations that occur behind the front, than for those ahead of the front so the front is not smoothed.

So-called entropy conditions are used to pick out a physically realistic solution from a set of multiple possible solutions of equation **(13)**. For fluxes, q, such that $q'' = (\partial^2 z/\partial t^2) > 0$ and saturations $S_1 > \xi > S_2$, then

the appropriate entropy inequality is given by Smoller (1983) as

$$q'(S_1) \;>\; \frac{dz}{dt} \;>\; q'(S_2) \qquad \textbf{(14)}$$

since q'(ξ) = dz/dt which is given by equation **(14)**.

During infiltration, the capillary pressure gradient may play a role in determining the NAPL flux into the soil. If NAPL flux exceeds the kinematic capacity of the media ($q_o > K_{eo}$) or if ponding occurs at the surface, then the kinematic model must be augmented by a dynamic model, because the NAPL flux cannot be solely attributed to gravity. During loadings of either type, the Green and Ampt (1911) model is used as an approximate dynamic supplement to the kinematic model to determine the NAPL flux. With ponding head of H_s, the flux equation is integrated from the surface to the position of the NAPL front, z_f, to give (Neuman, 1976)

$$q_1 \;=\; K_1 \left[\frac{H_s + z_f - H_f}{z_f} \right] \qquad \textbf{(15)}$$

with

$$H_f \;=\; h_1 - \int_{h_2}^{h_1} \frac{K_s k_{ro}}{K_1}\, dh \qquad \textbf{(16)}$$

where q_1 is the flux in the NAPL-filled pore space with NAPL saturation $_1S$, $_1K$ is the corresponding NAPL effective conductivity, h is the pressure head, subscripts 1 and 2 refer to locations behind and ahead of the front, respectively. The head as a function of NAPL saturation is determined by scaling the Brooks and Corey (1964) capillary pressure model from air/water drainage to air/NAPL drainage by the ratio of surface tensions and densities. The pore geometry is assumed to be the same in the NAPL/air system, so that Brooks and Corey's λ is the same for both curves. The air/water entry head, h_{ceaw} is scaled by

$$\frac{h_{ceao}}{h_{ceaw}} \;=\; \frac{\rho_w}{\rho_o}\,\frac{\sigma_{ao}}{\sigma_{aw}} \qquad \textbf{(17)}$$

to estimate the air/NAPL entry head, h_{ceao}. The definition of the effective saturation is modified to reflect that the total liquid saturation S_t, which is the sum of the water and NAPL saturations, is controlled by the air/NAPL capillary pressure (Leverett, 1941). The resulting expression for the air/NAPL capillary head at the front is

$$h_c \;=\; h_{ceao} \left(\frac{S_o + S_w - S_{wr}}{1 - S_{wr}} \right)^{-\frac{1}{\lambda}} \qquad \textbf{(18)}$$

After transforming the independent variable to S_o, the integral appearing in equation **(16)** becomes

$$\frac{-K_{so}\,h_{ceao}}{K_1(1-S_{wr})\lambda}\int_0^{S_1} k_{ro}(S_o,S_w)\left(\frac{S_o+S_w-S_{wr}}{1-S_{wr}}\right)^{-(1+\frac{1}{\lambda})} dS_o \qquad \textbf{(19)}$$

The scaling of the air/NAPL capillary pressure curve in equations **(18)** and **(19)** does not include the residual NAPL saturation, because no NAPL is present in the profile before the passage of the front. In this situation, the relative permeability equation for the NAPL is also assumed to have no residual NAPL saturation. Equation **(19)** is evaluated numerically, because the complexity of the relative permeability function precludes a closed form solution. The capillary head behind the front, h_1, is calculated using equation **(18)**.

The jump condition (Equation **(13)**) is used to determine the speed of the invading front with the effective conductivity, K_{eo}, replaced by the flux determined from equation **(15)**, giving

$$\frac{dz}{dt} = \frac{K_1}{\eta S_1}\left(1+\frac{H}{z_f}\right) \qquad \textbf{(20)}$$

where H is the sum of the head terms in the numerator of equation **(15)**. Although this function could be left as a differential equation and solved numerically, an analytic solution for the position of the front is given by

$$t - t_o = \frac{\eta S_1}{K_1}\left[z_f - H\,ln(z_f+H)\right] \qquad \textbf{(21)}$$

Equation **(21)** is preferred over equation **(20)**, because the high initial pressure gradient causes problems in the numerical solution of equation **(20)**. The contribution of the suction head to the driving force causes increased NAPL flux and is included to assure that the proper amount of fluid is drawn into the soil during the loading period. Even though there is a NAPL head at the surface, the amount of water in the soil is assumed unchanged; thus this boundary condition strictly applies only to situations where the water saturation is residual. The water saturation in the profile, however, remains uniform as it is assumed to result from the continuous supply of recharge at the surface.

When the supply of the NAPL is finite, the kinematic approach is used to determine the flux during redistribution. Like the beginning of the NAPL event, the end of the event is treated as an abrupt change in ponding depth or flux from a specified value to zero. This change in boundary data triggers a wave which displays a smooth transition from low saturation near the surface to higher saturation at depth and the classical method of characteristics solution applies. The behavior is caused by the shape of the relative permeability function for the NAPL phase. Because the derivative of equation **(4)** is a monotonically increasing function of NAPL saturation, the characteristic speeds increase with saturation. Thus the high saturations deeper in the profile move faster than the low saturations near the surface, and the redistribution profile appears smooth.

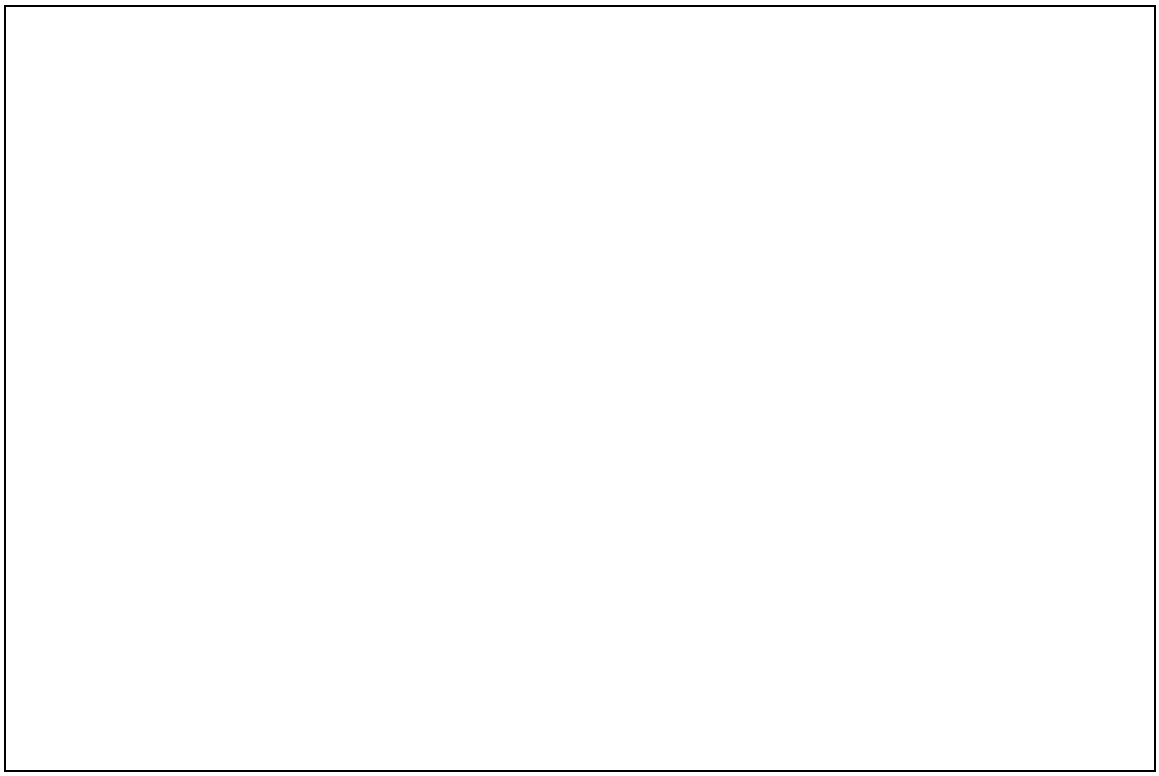

Figure 5 KOPT model release scenarios

Several surface conditions can be included in the model. Four options have been included in KOPT to correspond to spill or release scenarios (**Figure 5**). The first condition is a flux condition. NAPL fluxes less than the maximum NAPL effective conductivity produce no runoff, and all the NAPL enters the soil. Thus, a constant flux for a specified duration is used as a boundary condition. Conditions where the specified flux exceeds the effective conductivity are treated by the Green-Ampt approach discussed above. Excess flux is assumed to run off the surface of the soil. The second condition corresponds to a land treatment scenario where a certain volume of NAPL is incorporated uniformly over a specified depth at time zero. The third option is the constant head ponding scenario. Here, NAPL is ponded at the surface at a constant head. This scenario corresponds to an impoundment which is maintained at a certain depth for a specified duration. A ruptured tank contained within a berm is envisioned here. This boundary condition requires use of the Green-Ampt model. The fourth boundary condition implemented in the model allows constant head ponding for a specified duration, followed by variably decreasing ponding. This boundary condition is primarily useful for laboratory experiments, where ponding depths cannot instantly be reduced to zero. The falling head condition is implemented by considering that H_s is a function of time, and that the reduction in NAPL ponding height is equal to the NAPL infiltrated over a time increment, thus

$$H_s^{n+1} = H_s^n + \eta S_1 (z_f^{n+1} - z_f^n) \qquad \textbf{(22)}$$

where H_s^n is the ponding depth at time n and $_f^n z$ is the front depth at time n. Equation **(22)** is used with equation **(21)** to incorporate the time varying ponding depth into the Green-Ampt model.

Although the primary contaminant of interest is a dissolved constituent of the NAPL, in some cases it is important to consider the dissolution of the NAPL itself. Examples are when NAPL is a pure chemical such as trichloroethene or carbon tetrachloride, or for consistency with models for transport at the water table

(Weaver and Charbeneau, 1990). The water solubility of the NAPL presents a loss of mass of the NAPL phase, thus the loss to the pore water of NAPL reduces NAPL saturations in the profile. The amount of mass lost to the water phase is subtracted uniformly from the NAPL saturation profile at the end of each simulation time step. The result of this operation is that the characteristics curve toward the land surface because of the changing saturations. For this situation, the location of the characteristics must be found by integrating the characteristic speeds (equation **(12)**). Obviously, the additional computational burden for this case is significant.

2.3 Derivation of the Vadose Zone Transport Model

The kinematic model is also applied to the dissolved constituent. This is justified in part from the fact that advection, multiphase partitioning, volatilization, and degradation are the major controls on transport and fate for many organic contaminants. Hydrophobic constituents of NAPLs are largely transported by advection of the NAPL, because of their preferential partitioning into the NAPL phase. Chemicals like xylene dissolved in gasoline move primarily with the NAPL due to this reason. As noted previously, the kinematic models move the concentration fronts at the correct speed for mass conservation.

The dissolved constituent solution proceeds as follows. For the nondispersive migration of a constituent initially dissolved in the NAPL, the mass of constituent per unit volume, m_c, is

$$m_c = \eta (S_o c_o + S_w c_w) + \rho_b c_s \qquad \textbf{(23)}$$

where c_o and c_w are the constituent concentrations in oil and water, ρ_b is the soil's bulk density, and ς is the sorbed concentration expressed as mass of constituent/mass of soil. The mass advection flux is given by

$$J_c = q_o c_o + q_w c_w \qquad \textbf{(24)}$$

substituting these into the general mass conservation equation gives

$$\frac{\partial}{\partial t}\left(\eta B c_w\right) + \frac{\partial}{\partial z}\left(q_o c_o + q_w c_w\right) = 0 \qquad \textbf{(25)}$$

where $B = B(S_o,S_w) = (S_w + S_o k_o) + \rho_b k_d/\eta$, which may be called a bulk water partition coefficient. The individual partition coefficients are given by

$$\begin{aligned} k_o &= c_o/c_w \\ k_d &= c_s/c_w \\ k_w &= 1 \end{aligned} \qquad \textbf{(26)}$$

The partition coefficient, k_d, is the usual soil water distribution coefficient. The NAPL/water partition coefficient, k_o, is estimated from the NAPL composition by using Raoult's law (e.g., Cline et al., 1991). For consistency, with the assumption that the flux of air is neglected, partitioning into the air phase is not used in the mass balance equation for the constituent. Neglect of volatilization and air phase partitioning is conservative since KOPT is one module of a screening model for predicting water phase concentrations in downgradient groundwater wells. In addition, for many constituents, the total percent of mass within the air phase is small compared to that within the NAPL and sorbed to the soil. Expanding the derivatives and using equations **(9)** and **(10)** while applying the linear partitioning relationships gives

$$\eta B \frac{\partial c_w}{\partial t} + \Sigma q_i k_i \frac{\partial c_w}{\partial z} = 0 \qquad \textbf{(27)}$$

After the saturations of water and the NAPL are known, c_w is the only unknown in equation **(27)**.

Applying the MOC results in

$$\frac{dc_w}{dt} = 0 \qquad \textbf{(28)}$$

along

$$\frac{dz}{dt} = \frac{q_w + q_o k_o}{\eta B(S_o, S_w)} \qquad \textbf{(29)}$$

Equation **(27)** is a semi-linear equation, as it is linear in the unknown c_w. For such equations, characteristic speeds match jump speeds as the jumps are co-located with characteristics. The jumps are contact discontinuities by definition (Smoller, 1983). Another feature of both linear and semi-linear equations is that the characteristic speed given by equation **(29)** is independent of the dependent variable, c_w.

The concentration at the beginning of simulation is determined as follows. The constituent concentration in the released NAPL, $c_{o(surf.)}$, changes instantly upon placement in the soil, because of the local equilibrium assumed for the constituent. For situations where the NAPL flux is specified, the constituent concentration in water in the soil can be determined by balancing mass fluxes across the surface of the soil

$$c_{o(surf.)}\, q_o = c_{w(initial)}\, (q_w + q_o k_o) \qquad \textbf{(30)}$$

This relation supplies the initial concentration, $c_{w(initial)}$, for each constituent characteristic. Equation **(30)** shows that only if the water flux is zero will the initial concentration in the NAPL be the same as that in the NAPL phase in the soil; otherwise, $c_{o(soil)} < c_{o(surf.)}$. When a specified volume of NAPL is applied to a zone of thickness, d_{pz} (i.e., the plow zone thickness for a land treatment system), the initial concentration is calculated via equation **(31)** where V_o is the volume of applied oil per unit surface area.

$$c_{o(surf.)}\, V_o = c_{w(initial)}\, B(S_o, S_w)\, d_{pz} \qquad \textbf{(31)}$$

The initial concentration in a land treatment waste, $c_{w(initial)}$ is used to calculate the initial constituent concentration in the oil phase.

2.4 Summary of Approximate Governing Equations for the KOPT Module

The flow of the NAPL is governed by the two parts of the generalized method of characteristics solution. Equations **(11)** and **(12)**, the classical solution, are used where the NAPL saturation varies continuously, in which case both partial derivatives appearing in the continuity equation **(10)** exist in the solution domain. Equation **(13)**, the generalized solution, is used to determine the speed and position of the front at the leading edge of the NAPL. In KOPT, these equations are solved first and determine the distribution of the NAPL in the profile. Equations **(28)** and **(29)** are the approximate governing equations for the dissolved constituent. Since transport of the dissolved constituent is governed by a semi-linear equation, a characteristic is co-located with the leading-edge front and no generalized solution is needed. The solution for the dissolved constituent is determined from the governing equations and the NAPL distribution and flux. By simplifying the governing equations and applying the method of characteristics, the original nonlinear system of coupled partial differential equations has been reduced to a system of nonlinear ordinary differential equations. An analytic solution is not known for this case, so a numerical method is used to solve the ordinary differential equations.

2.5 Model Implementation

The solution of the model equations is obtained by the use of an ordinary differential equation solver. With such a technique, the solution of a system of n equations of the form

$$\frac{d\mathbf{y}}{dt} = f_n(y_1, y_2, y_3, ..., y_n) \qquad \textbf{(32)}$$

may be obtained, if appropriate boundary conditions are specified. The functions, f_n, may be coupled and/or nonlinear functions of the y_i. In this case the y represent the NAPL front position, cumulative NAPL mass applied at the boundary, position of the dissolved chemical characteristics and, if required for the simulation performed, the position and saturation of the NAPL characteristics, the NAPL ponding depth at the surface and cumulative NAPL runoff from the surface. For KOPT, a Runge-Kutta-Fehlberg, RKF, method (Fehlberg, 1969) was selected in order to allow automatic time step variation to control truncation error.

There are several advantages to using a differential equation solver instead of finite difference or finite element techniques for this problem. Use of the differential equation solver leads to simplified programming in the sense that the programmer supplies the functions on the right-hand side of equation **(32)** to the solver in a subroutine. Second, the RKF methods maintain the truncation error below a specified tolerance by reducing the time step. This feature operates automatically during the program execution, resulting in a variable time step routine. The method used here, RKF1(2), uses a first-order scheme to get the solution and an embedded second-order scheme to check the truncation error. Third, under certain conditions, only a few of the equations need to be solved. For example, the characteristics for non-dissolving NAPLs are straight lines; and only the equations for the discontinuities need be solved. A final advantage is that the location of the fronts is determined directly as a function of time.

In addition to the truncation error control of the method, a fairly complicated and specialized system of ad hoc controls is needed to assure the accuracy of the solution. For example, the following features of the solution must occur at times the solver picks for the solution:

-the beginning and end of the NAPL release
-the end of the maximum NAPL saturation region
-the origin of any characteristic

The checks are implemented in a controller routine that is called by the RKF solver. A beneficial side effect of using the controller is that the number of steps rejected for truncation error violations is reduced, since the solver is guided to critical times in the solution by the controller (Charbeneau et al., 1989).

2.6 General Features of the KOPT Solution

Figure 3 shows the base characteristic plane and the projection of a few displacement paths in z-t space. The input parameters for this example are presented in Table 1, Simulation A and are discussed in further detail below. The NAPL leaks into a sand with water at residual saturation. During infiltration, the NAPL fills 76.8% of the pore space and moves at its maximum speed, which is determined by the Green-Ampt model with variable surface ponding. Once the supply of NAPL ends at 6.2 minutes (point B), drainage begins. There is in the profile a region of constant oil saturation (triangle ABC), associated with the NAPL release, that will now be replaced by a region of variable saturation, associated with the MOC solution for NAPL drainage. The displacement path (called a characteristic) of the residual saturation, S_{or}, remains at the surface, because its speed is zero by definition. Bounding the region of maximum saturation, ABC, is the characteristic corresponding to $S_o = S_{o(max)} = 0.768$. In between are characteristics for all intermediate values of S_o, as labeled in **Figure 3**. As noted above, the highest NAPL saturations are found the deepest in the profile, because the derivative of the k_{ro} function is such that the highest NAPL saturations have the highest speeds.

Figure 4 shows oil profiles for 9.0 to 96.0 minutes. In the profiles, water and the total liquid saturations are plotted; the NAPL saturation is read as the difference between the two. By 9 minutes after the beginning of the release, drainage is occurring from the surface down to a depth of 12.4 cm. The region from 12.4 to 24.2 cm still has the original saturation of 0.768. Once the NAPL saturations that are less than $S_{o(max)}$ reach the oil front as has occurred in the profiles for 24, 48 and 96 minutes, then its speed begins to be reduced in accordance with equation **(13)**. As time goes on, the speed of the front slows as it intersects slower and slower characteristics (**Figure 3**). By 96 minutes, which corresponds to the end of the experiment described below, the NAPL front has reached 59.0 cm and its saturation has dropped to 0.367.

The entropy condition is such that the characteristic speeds should decrease across the front, and that the front speed should be intermediate to these two. With a pristine initial condition, the characteristic speed ahead of the front is zero. **Figure 6** shows the characteristic speeds and the front speed, demonstrating compliance with the entropy condition (equation **(14)**). The abrupt drop in both speeds occurs when the surface ponding ends and the Green-Ampt model solution is switched to the kinematic model solution. At this time, the contribution to the speed from the capillary suction at the front is dropped from the model. The subsequent short plateau occurs because the front speed remains constant since the NAPL saturation at the front remains at 0.768 until the drainage wave reaches the front (**Figure 3** and **Figure 4**). Once the latter occurs, the front and characteristic speeds decrease as shown in **Figure 6**.

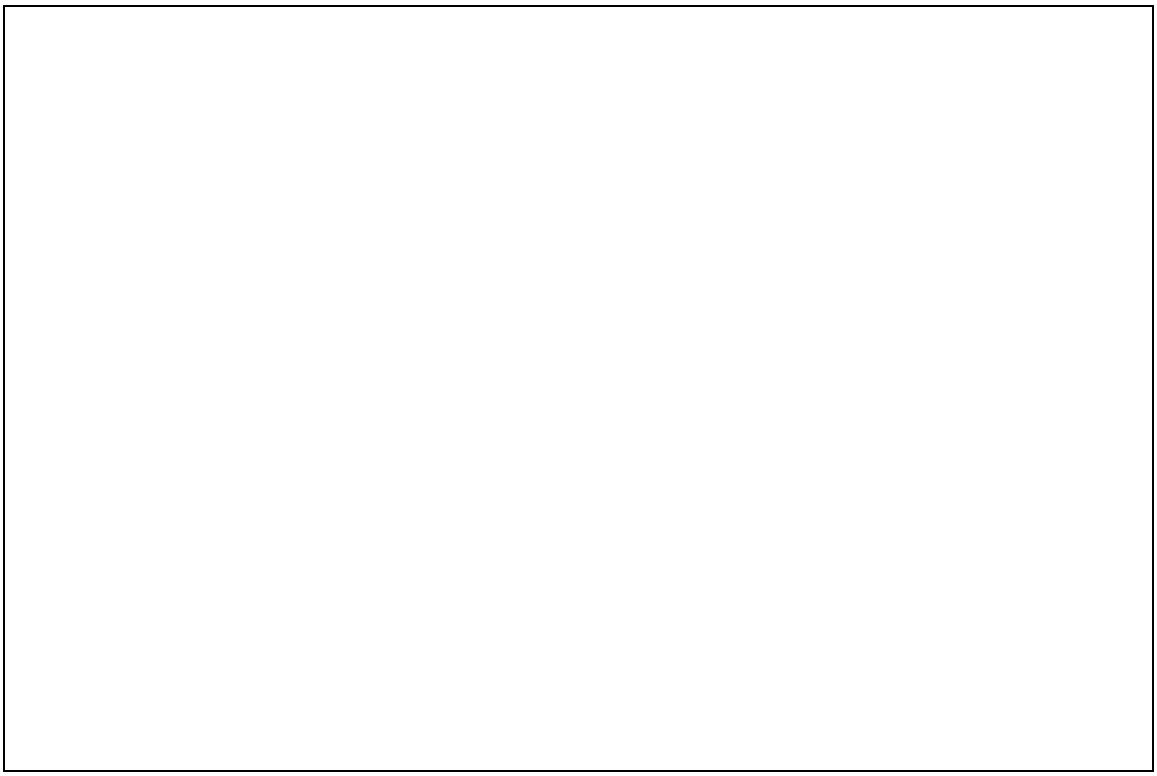

Figure 6 Upstream characteristic and front speeds

2.7 Experimental Results and Simulation

A simple laboratory experiment was conducted in order to evaluate the NAPL phase flow portion of the KOPT model. The ponding depth and front position were observed visually by tracking the flow of a dyed oil in a laboratory column described in the following. A 1.0 m long, 0.05 m diameter glass column with a coarse, porous glass frit at the bottom was modified by adding seven air release vents oppositely (**Figure 7**). The vents were packed with glass wool before packing the column with a uniform sand (Gilson ASTM c109). With the vents closed, the column was purged with CO_2, and followed by an applied vacuum of 20 cm of mercury. De-aired water was allowed to enter from the bottom of the column, gradually saturating the sand pack. The saturated sand pack was used as a permeameter for determining the hydraulic conductivity under steady state flow conditions. Manometers attached to the vents on one side of the column were used to determine the head drop over each 10 cm long section of the column. After measuring hydraulic conductivity, the column was drained by lowering the water table to the top of the frit. The c109 sand is coarse enough to have a low capillary fringe, and the water saturation above the fringe was assumed to be near residual after several hours of drainage.

Approximately 100 g of dyed Soltrol 220 was then released at the surface and allowed to enter the column (with vents opened ahead of the Soltrol front) to simulate a NAPL spill. The ponding depth and location of the NAPL front were recorded with time. The NAPL front was measured at three locations (left edge, center, and right edge) to capture variability in the front; as even in nearly uniform packings, the front does not remain absolutely uniform. In a test case, a core was taken from the column to verify that the NAPL flowed through the sand pack and not preferentially along the walls. The core confirmed that the oil was found at the same depth inside the column and along the walls. **Figure 8** shows the NAPL front position at the left edge, center and right edge of the column. **Figure 9** shows the measured depths of ponded Soltrol at the surface. Both of these figures also show simulation results which are described below.

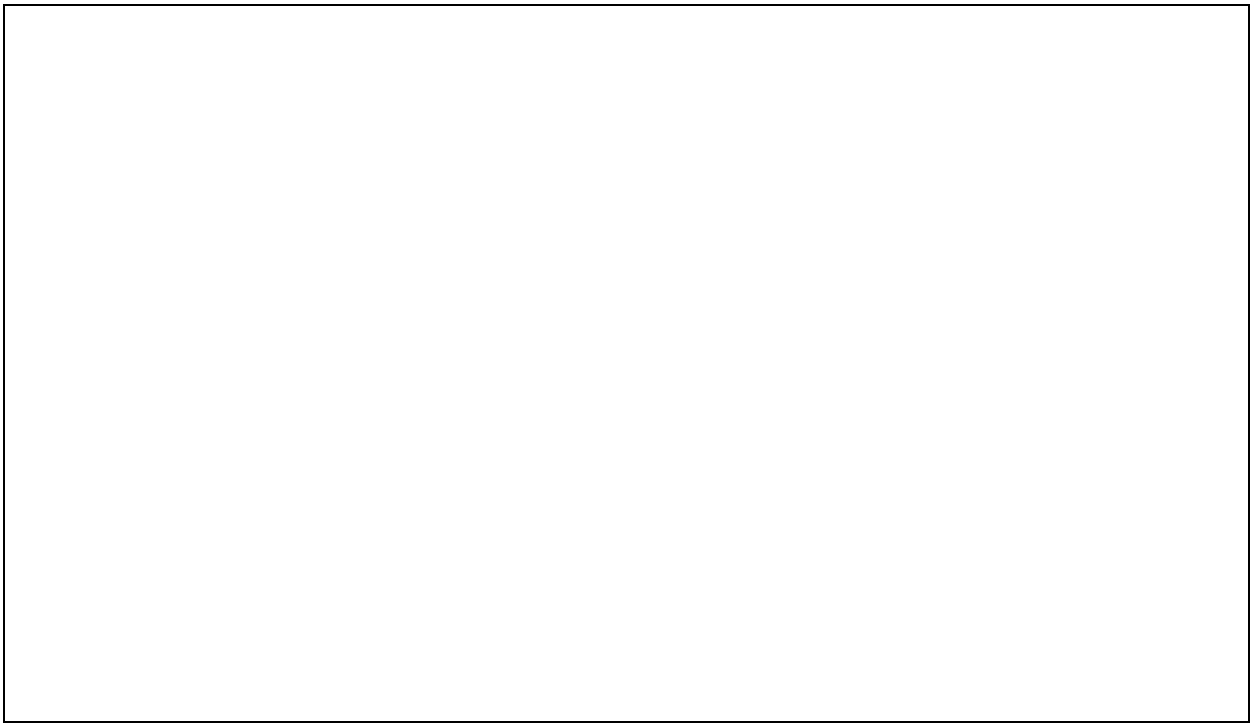

Figure 7 Glass column used for the laboratory evaluation of KOPT

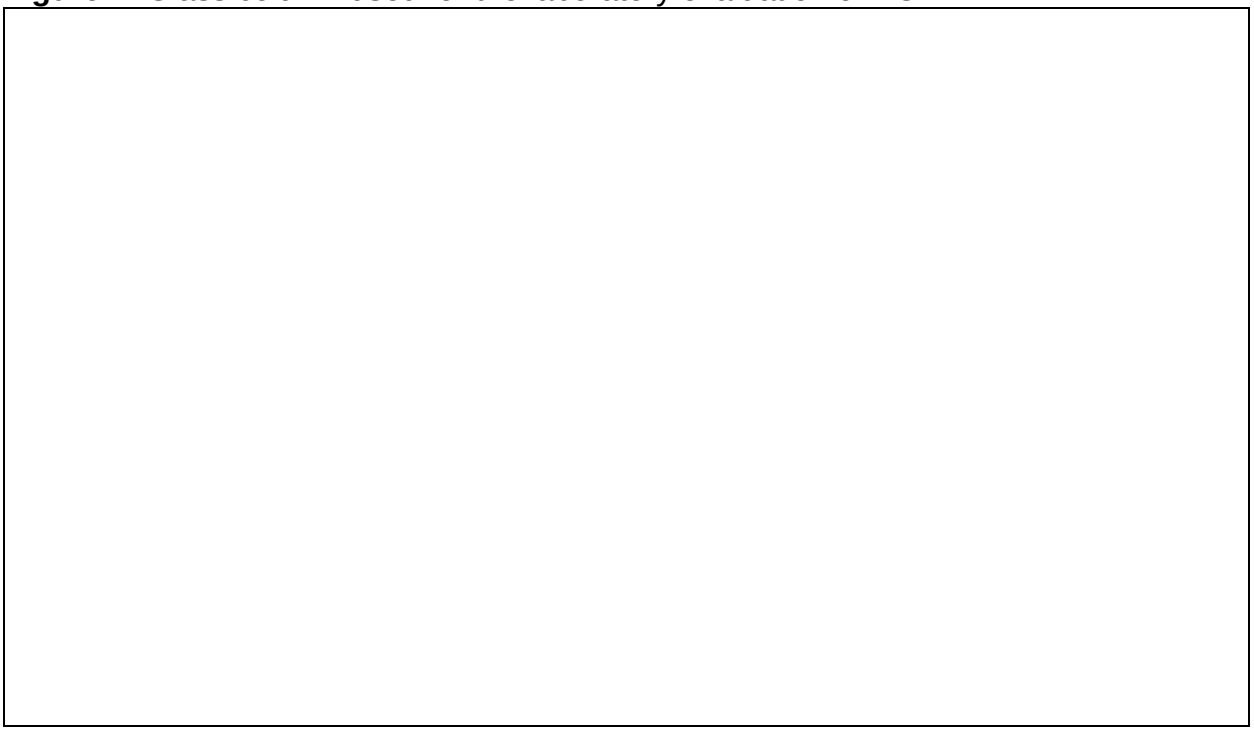

Figure 8 Measured NAPL position at right-hand edge, center and left-hand edge of column.

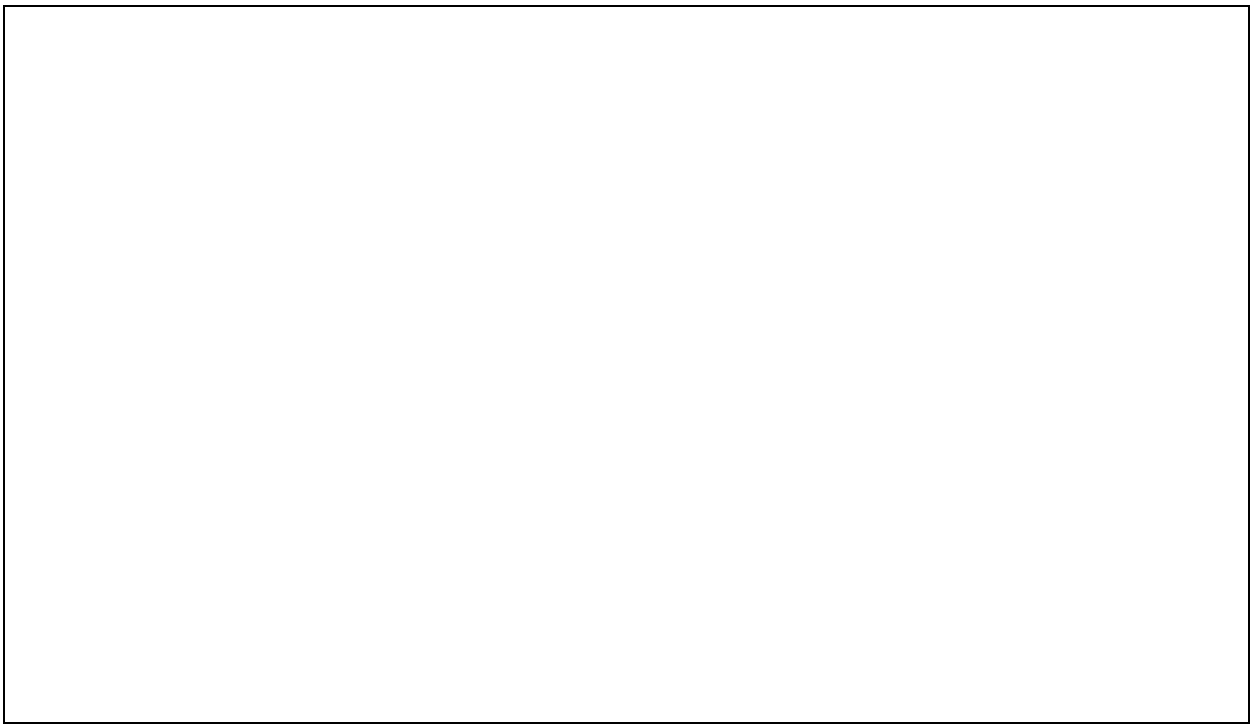

Figure 9 Measured NAPL ponding depth at the surface of the sand

The model parameters, K, η, h_{ceao}, λ, S_{wr}, ρ_o, μ_o, and σ_o were determined independently of the transient flow experiment, so that the ability of the model to match the outcome of the experiment could be seen without fitting or other adjustment of the parameters. The values of the parameters used in simulation A are listed in Table 1, along with the technique used for their measurement. **Figure 10** shows the variation of hydraulic conductivity along the length of the column. The average hydraulic conductivity of the sand is 78.0 m/d. The air/water capillary pressure curve for the c109 sand was measured by the technique developed by Su and Brooks (1980). The measurements were made on a 5.0 cm long, 5.0 cm diameter column, packed using the same procedure as was used for the long column. The Brooks and Corey parameters of the air/water capillary pressure curve were fitted by using the RETC program developed by van Genuchten et al. (1991). RETC returns average values of the parameters along with 95% confidence limits for the values. Two sets of water-air capillary pressure data and the fitted model are shown in **Figure 11**. This figure shows only a few measured points at low water saturations, so the fitted residual water saturation has greater uncertainty than the other parameters. The water saturation in the column, $S_{w(avg)}$, was taken as either the residual, S_{wr}, or as a value determined by the NAPL penetration into the column as discussed below. The porosity was estimated from the bulk density of the sand pack in the long column. The NAPL phase properties, ρ_0, μ_0, and σ_0, were averaged from replicate measurements. The residual NAPL and air saturation were estimated; the trapped NAPL saturation is estimated to be 5% of the pore space, while the trapped air saturation is estimated as the air saturation at k_{rw} = 0.5 (equation **(8)**), which is 0.1757 for this example.

Table 1 Simulation Parameters and the Techniques Used for Their Measurement

Parameter	Simulation				Measurement Technique
	A	B	C	D	
K_s (m/d)	78.0	78.0	83.0	78.0	Column Permeameter
η	0.41[1]	0.41[1]	0.41[1]	0.37[2]	[1] Column Permeameter [2] Su and Brooks Cell
h_{ceaw}(cm)	-24.8	-24.8	-24.8	-24.8	Su and Brooks/RETC
λ	4.84	4.84	4.84	4.84	Su and Brooks/RETC
S_{wr}	0.0588	0.0588	0.0588	0.0588	Su and Brooks/RETC
S_{or}	0.05	0.05	0.05	0.05	Estimated
μ_o (cP)	4.76	4.76	4.76	4.76	Cannon-Fenske Viscometer
ρ_o (g/cc)	0.79	0.79	0.79	0.79	Gravimetric
σ_o (dyne/cm)	25.0	25.0	25.0	25.0	DuNuoy Ring Tensiometer
initial ponding depth (cm)	6.5	6.5	6.5	6.5	Estimated from NAPL Volume
max k_{rw}	0.5	0.5	0.5	0.5	Estimated
$S_{w(avg)}$	0.0588	0.1129	0.1129	0.0588	Estimated as discussed in the text

Figure 8 and **Figure 9** show that the KOPT simulation, using the average parameter values (simulation A, Table 1), captures the qualitative behavior of the front and ponding depth. The simulation matches the initial rapid influx of NAPL during ponded infiltration, followed by a slowing of the front speed as the NAPL redistributes. This simulation, however, fails to match exactly either the ponding depths or the front positions. The model overpredicts the infiltration rate, resulting in a shorter period of ponding than actually observed. This behavior suggests that the model's estimates of effective conductivity or capillary suction may be too high. Also note that in the experiment, the NAPL was poured onto the sand surface; thus the ponding began at zero, increased to a maximum, then declined. In **Figure 9** it can be noted that the simulated ponding depths are achieved instantaneously and do not go through the first experimental point. This behavior is likely to result from the way the NAPL is poured into the column.

Simulation B uses modified water and NAPL saturations in the profile that are determined as follows. In KOPT, the NAPL is assumed to fill a fixed portion of the pore space during infiltration, so at the end of the ponding period, the entire infiltrated volume of NAPL is between the surface and the NAPL front. From the column geometry, porosity, measured depth of the front, and the NAPL volume applied to the surface, the NAPL saturation is estimated by mass balance to be 0.7141, which is lower than the value generated by KOPT for simulation A (0.7682). The water saturation to fill the remaining pore space is 0.1129, if the trapped air saturation is unaltered. KOPT was then rerun with the water saturation set to 0.1129, which forces the model to give precisely the measured front position when the surface ponding first equals zero. **Figure 8** indicates that simulation B lies closer to the experimental data after 25 minutes into the experiment. In effect, the maximum effective NAPL conductivity was reduced by lowering the NAPL saturation (Table 2). Simulation B has a longer ponding period, reflecting the lower average NAPL flux. The results from simulation A suggest that lowering the NAPL flux during infiltration should bring the simulated and measured results closer together.

Figure 10 Measured distribution of hydraulic conductivity in the sand pack

Table 2 Quantities determined from the Simulation Results

Simulation	Approximate Ponding Time (Minutes)	Maximum Effective NAPL Conductivity, K_1 (m/d)	Capillary Suction Contribution to the Green Ampt Model ($-H_f$) (m)
A	6.19	5.55	0.1247
B	7.63	4.64	0.1242
C	7.20	4.93	0.1242
D	6.48	5.53	0.1247

Figure 11 Data from two measured capillary pressure curves for the c109 sand and the fitted Brooks and Corey model (solid line)

KOPT requires single values for each input parameter. As shown in **Figure 10**, the hydraulic conductivity varies along the length of the column. Simulation C uses the value of the hydraulic conductivity found nearest the surface (83.0 m/d, **Figure 10**) and the NAPL and water saturations from simulation B. The NAPL enters sand with higher than average conductivity and initially has generally higher fluxes than given by the average of 78.0 m/d. The duration of ponding is reduced and the simulated front position is closer to the experimental data. Table 2 indicates that the variation in H_f is relatively low so that the Green-Ampt flux is mostly affected by the maximum effective conductivity, rather than variation in H_f.

In order to assess the sensitivity of the model to the parameters, **Figure 12** shows first order, nondimensional, sensitivity coefficients, SC_i, for the first 7.5 minutes of the simulation. Throughout this time, there is some NAPL ponded at the surface; and the Green-Ampt model is used to determine the NAPL flux. The SC_i values are calculated from

$$SC_i = \frac{p_i}{z_f}\frac{\partial z_f}{\partial p_i} \qquad \textbf{(33)}$$

Figure 12 Nondimensional sensitivity coefficients for the Green-Ampt portion of the simulation B parameter set

where the p_i are the model parameters, z_f is the front position at a given time. The derivative in equation **(33)** is calculated by running the model for slightly different values of p_i, and then estimating the derivative by a central difference approximation. Positive SC_i indicate that the front position increases with increasing p_i; while negative SC_i indicate that the front position decreases with increasing p_i. For the parameter set used in simulation B, the Green-Ampt model is most sensitive to the saturated hydraulic conductivity and the porosity (**Figure 12**). The maximum relative permeability to water, which is used to determine the trapped air saturation by equation **(8)**, affects the maximum effective conductivity to the NAPL; so its curve is similar to that for the saturated conductivity. Because of its role in the capillary suction approximation, the air/water entry head is also a fairly sensitive parameter. The model exhibits relatively low sensitivity to the other parameters.

After the ponding ends, the kinematic model takes over. Sensitivity coefficients for the remainder of the simulation are shown in **Figure 13**. The curves are discontinuous across the change from the Green-Ampt model (**Figure 12**) to the kinematic model (**Figure 13**). The kinematic model sensitivity coefficients show increasing sensitivity to saturated hydraulic conductivity, maximum water relative permeability, and porosity; and declining sensitivity to air entry head during an initial period. This effect is evidently related to the time while the NAPL saturation at the front is still a maximum. The increasing sensitivity lasts until after the NAPL saturation begins to be reduced. During the later stages of redistribution, porosity continues to increase in sensitivity and the Brooks and Corey λ shows steadily increasing sensitivity.

Figure 13 Nondimensional sensitivity coefficients for the kinematic portion of the simulation B parameter set

Simulation D is presented to illustrate the sensitivity of the model to the estimate of the porosity. The sensitivity analyses indicate the high sensitivity of the model to the porosity. The value of 0.41 used in simulations A-C is obtained from the 1.0-meter long column. The value used in simulation D is 0.37, which was obtained from the measurement of the air/water capillary pressure curve. The lower value of porosity causes the front to lie deeper in the profile. The density of the sand in the column is likely to vary, as evidenced by the variation in hydraulic conductivity; the porosity is likewise expected to vary. So it is possible that lower porosities are encountered in the column than given by the average of 0.41. This simulation also illustrates that parameters may be adjusted to fit a set of experimental data. In this case, only one parameter of the simulation A data set was adjusted, and the model result closely matches the experimental data. Other sets of parameter values can be found to match the experimental data closely, if fitting the model to data is desired.

Since equation **(10)** expresses mass conservation, a correct solution demonstrates conservation of mass. For simulation A, the model allowed 0.1008 kg of NAPL to infiltrate during ponded infiltration. The NAPL mass in the profile matches nearly exactly the mass applied in the experiment, because the boundary condition allows the amount of mass applied to the column to be precisely specified. The model integrates the NAPL mass flux at the boundary and compares it to the NAPL mass found within the profile. These mass balances performed during simulation A show a maximum error of 0.050%.

2.8 Closure on the KOPT model

Use of Green-Ampt theory for ponded infiltration and kinematic wave theory for redistribution allows the construction of an approximate model of NAPL releases into the vadose zone. The generalized method of characteristics can be applied to both of these approaches; and so allows either model to be used for the NAPL flux as necessary. Use of kinematic wave theory for redistribution of the NAPL yields transient NAPL profiles which are not based on arbitrary assumptions and represent the profiles better than previous simplified models. Transport of a dissolved constituent of the NAPL can also be formulated as kinematic model. The partitioning between the phases can be represented by equilibrium linear partition coefficients, which yields an approximate equation for the water phase concentration of the constituent. The flow and transport model can then be entirely stated in terms of ordinary differential equations, which are solved in the KOPT code by a Runge-Kutta technique. The computational efficiency of the model results from the direct use of the characteristic directions and front speeds, which eliminates the need to discretize the domain and solve the equations by a differencing method.

The laboratory evaluation of the model demonstrates the ability of the model to capture the essential qualitative features of NAPL infiltration and redistribution for the experimental conditions. The quantitative evaluation of the model is limited by the estimation of some of the model parameters in this work, the variability of the sand pack, and the inability to measure the transient saturations and capillary pressure curves in situ. The quantitative agreement of KOPT with the experimental results is still considered acceptable, as in a screening model, exact simulation of heterogeneous conditions is not intended. The simulation of the experiment illustrates that the treatment of the flow processes in KOPT is approximately correct and there are not large errors in the gross behavior of the model.

Section 3 NAPL Lens Formation at the Capillary Fringe and Source Term Characterization

When an LNAPL reaches the water table after a spill or release from a leaking tank or pipeline, it will pond in an oil lens which grows in thickness and spreads. After the source is cut off, the lens will spread until it reaches a thin layer within the capillary fringe. Much of the hydrocarbon will remain isolated both above and below the water table at residual saturations. The constituents from the LNAPL release can dissolve into groundwater which flows beneath the lens, thereby contaminating downgradient drinking water through miscible phase transport. The OILENS model discussed in this section was developed to provide a groundwater transport model source term resulting from dissolution of constituents from a floating free product lens. The OILENS model is based on a number of simplifying assumptions which are listed below:

1) The hydrocarbon and its constituents enter the lens within a circular area of radius R_s centered beneath the surface source area. The hydrocarbon enters at a time-variable rate calculated by the KOPT model.

2) As the oil lens grows and spreads, residual hydrocarbon is trapped within the vadose zone and the saturated zone beneath the lens. Part of this trapping is associated with the dynamics of the source term (from the KOPT model) and lateral spreading capacity of the lens. Release of a more viscous hydrocarbon will result in a lens which achieves a greater thickness before it spreads, and will result in a greater amount of the hydrocarbon being trapped within the porous medium. OILENS calculates dynamic trapping through the simulation model itself, as described below. An additional source of trapping is associated with fluctuations of the water table. Water table fluctuations result in an apparent thickness of hydrocarbon which is independent of that required to drive lateral spreading. Capillary trapping due to fluctuations in the water table is included through a parameter that specifies the thickness of the hydrocarbon layer which must develop before the lens starts to spread. This residual fluctuation thickness is taken into account through the continuity equations.

3) A condition of vertical equilibrium holds for the fluids present at any given location. In particular, for the fluid levels in an observation well, this implies that the levels of the air-oil interface, z_{ao}, the oil-water interface, z_{ow}, the air-water interface (water table) in the absence of hydrocarbon, z_{aw}, and the observation well hydrocarbon thickness, b_o, are related through

$$z_{ao} - z_{aw} = \left(\frac{\rho_w - \rho_o}{\rho_w} \right) b_o \qquad \textbf{(34)}$$

$$z_{ow} - z_{aw} = - \frac{\rho_o}{\rho_w} b_o \qquad \textbf{(35)}$$

$$b_o = z_{ao} - z_{ow} = \left(\frac{\rho_w}{\rho_w - \rho_o} \right) h_o = \beta h_o \qquad \textbf{(36)}$$

where h_o is the head in the hydrocarbon layer at the given location ($h_o = z_{ao}$ if the elevation of the water table, z_{aw}, is chosen as the datum). These are essentially the relationships presented by van Dam (1967) and they are the same as the Ghyben-Herzberg approximations used for modeling fresh water floating on top of saline water in a porous medium (Bear, 1972). It should be noted that equation **(36)** states that the head within the hydrocarbon layer is directly proportional to the oil layer thickness as observed in a well. These relations are helpful in development of a computational model in that they provide the fluid energy distribution in a fashion which is not confounded by capillary pressure effects. The OILENS model is based on the observation well thickness of the lens and an effective volumetric oil content for the lens which comes from mass balance considerations.

4) Spreading of the hydrocarbon is purely radial, which implies that the slope of the regional water table is small enough to be unimportant for the lens motion.

5) In calculating the movement of the lens, both the hydrocarbon and water phase are assumed to be incompressible. Since the flow is assumed to be incompressible, the steady state solution can be applied at each instant in the unsteady motion of the oil lens (Muskat, 1946). The rate of lateral spreading is also assumed to be slow enough to confirm and justify use of a lens shape corresponding to steady-state flow. With the assumption of vertical equilibrium, this implies that a profile based on the Dupuit assumptions is appropriate.

6) An average effective volumetric oil content may be assigned to the lens, θ_o, along with retention oil contents for the vadose zone and the saturated zone beneath the lens, θ_{orv} and θ_{ors}, respectively. The meaning of the term "effective" is that it represents the ratio of the average lens thickness b_o (as seen in an observation well) to the actual free product thickness D_o. That is, $D_o = \theta_o b_o$. The actual distribution of an LNAPL near the water table is a function of the capillary pressure curve for the soil and the fluid densities and interfacial tensions. The capillary pressure curve for the soil (air-water system) may be scaled for the air-LNAPL and LNAPL-water systems following Leverett (1941), Schiegg (1984) and others who suggest that the capillary pressure heads are related by

$$h_{cow} = \frac{\rho_w \sigma_{ow}}{\Delta\rho_{ow} \sigma_{aw}} h_{caw} \qquad \textbf{(37)}$$

$$h_{cao} = \frac{\rho_w \sigma_{ao}}{\rho_o \sigma_{aw}} h_{caw} \qquad \textbf{(38)}$$

where h_{cij} is the capillary pressure head or capillary rise for the ij-fluid pair, h_{caw} is the capillary head for the water-air system, ρ is the fluid density, σ is the interfacial tension, and $\Delta\rho_{ow}$ is the density difference between the hydrocarbon and water. For the oil-water system, its capillary rise is measured from z_{ao}. These allow the LNAPL thickness D_o to be calculated from its thickness as seen in an observation well, as shown in **Figure 14**.

Figure 15 shows representative values of the effective LNAPL saturation as a function of average lens thickness for 35 API petroleum and for a gasoline in a sand soil.

7) The constituent mass is transported from the lens to groundwater by infiltrating water moving through the lens and by groundwater flowing beneath the lens and coming into contact with it. Equilibrium partitioning occurs between the hydrocarbon and water when they are in direct contact.

Figure 14 Calculation of LNAPL thickness in an oil lens

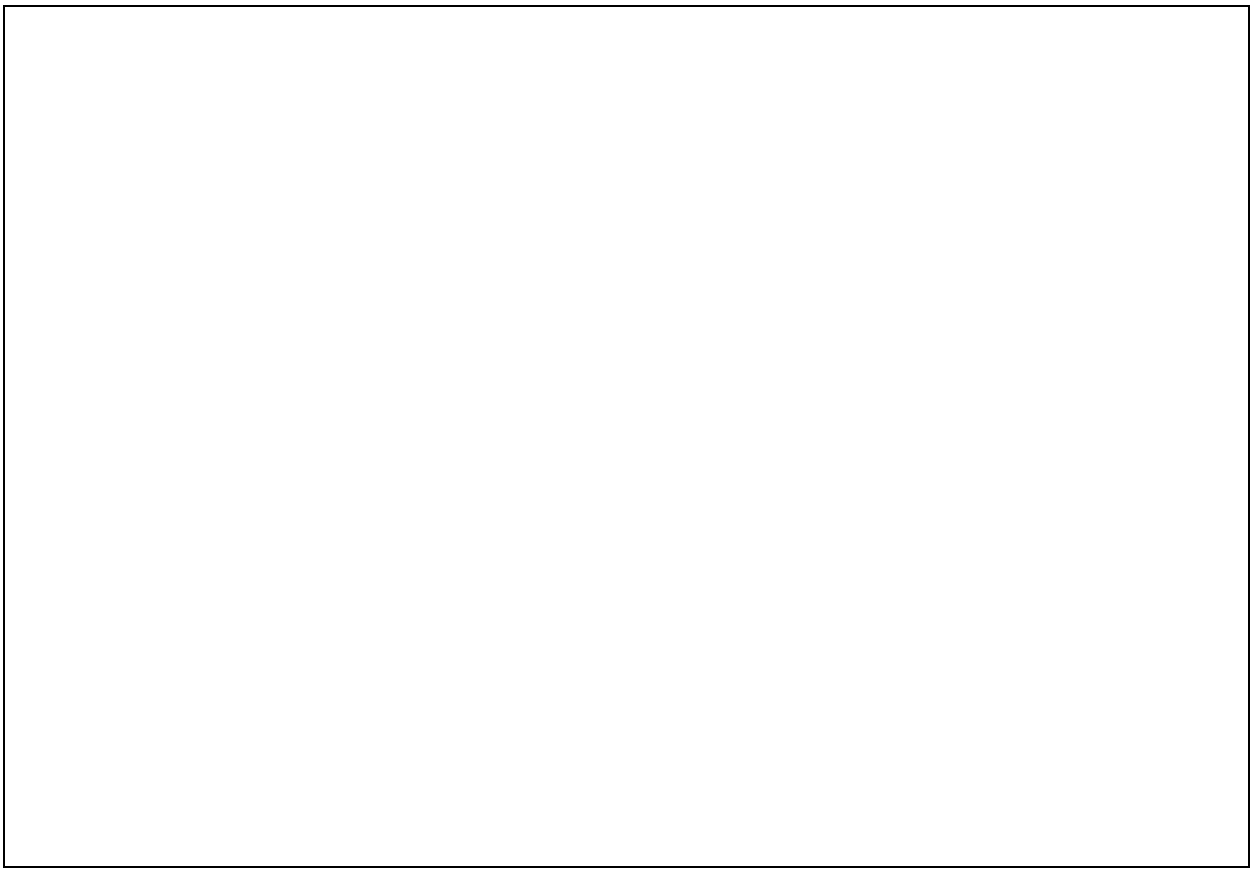

Figure 15 Effective saturation of a hydrocarbon in a sand

3.1 OILENS Model Development

The assumptions of vertical equilibrium, radial flow, and a steady-state hydrocarbon distribution lead to a simplified representation of the lens. At any given time the free product distribution is specified by three variables: the effective lens oil volumetric content, θ_o, the lens head beneath the source, h_{os}, and the radius of the lens, R_t. The lens oil content is specified as a constant input parameter and must be estimated from the conditions of the release. The remaining two variables, h_{os} and R_t, vary with time and must be calculated as part of the model. Their calculation is based on continuity principles, as described below.

From the Dupuit equation, the oil layer head at any radius $r > R_s$ is given by

$$h_o(r) = h_{os}\sqrt{\frac{\ln(R_t/r)^2}{\ln(R_t/R_s)^2}} \qquad \textbf{(39)}$$

In this last equation R_s is the source radius and R_t is the radius of the oil lens. Application of the continuity principle to the vertical circular cylinder of the lens beneath the source zone, as shown in **Figure 16**, gives

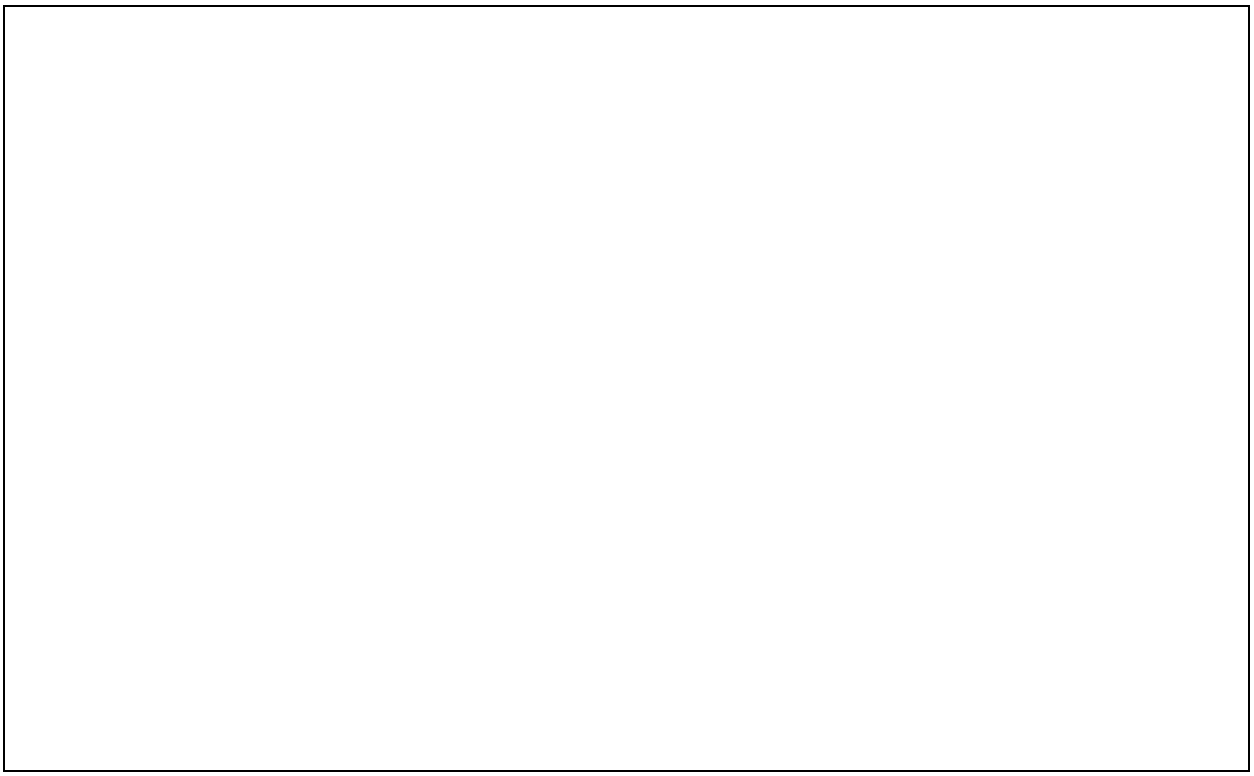

Figure 16 Volume balance for the source cylinder

$$Q_{KOPT} - Q_{radial} - Q_{loss} = \pi R_s^2 \theta_o \beta \frac{dh_{os}}{dt} \quad \textbf{(40)}$$

In equation **(40)**, Q_{KOPT} is the inflow to the lens from the vadose zone as calculated by the KOPT model, Q_{radial} is the lateral flow from the circular cylinder, Q_{loss} includes the volume of oil dissolved from the central cylinder plus the oil which remains trapped at residual saturation above and below the lens as the lens thickness decreases after the source has been cut-off. The right-hand-side in equation **(40)** gives the change in hydrocarbon volume within the cylinder. The radial flow component may be calculated from

$$Q_{radial} = -2\pi R_s b_o K_o \frac{dh_o}{dr}\Big|_{r = R_s} = \frac{\pi \beta K_o h_{os}^2}{\ln \frac{R_t}{R_s}} \quad \textbf{(41)}$$

Equations **(40)** and **(41)**, when combined with the discussion below for calculation of Q_{loss}, provide an ordinary differential equation for solving for the lens source head as a function of time:

$$\frac{dh_{os}}{dt} = F_1\left(h_{os}, R_t; Q_{KOPT}\right) \quad \textbf{(42)}$$

In equation **(42)**, h_{os} and R_t are functions which must be calculated and Q_{KOPT} is a function of time which is provided by the KOPT model.

The second equation for calculation of R_t comes from application of continuity to the lens as a whole. The continuity for the lens volume, V_L, may be written

$$\frac{dV_L}{dt} = Q_{KOPT} - Q_{out} \qquad \textbf{(43)}$$

where Q_{out} represents the hydrocarbon losses from the lens due to dissolution as well as that left as residual during mound decay following source control. V_L includes only the actively spreading LNAPL. Since V_L is a function of h_{os} and R_t, we may use to chain rule to write

$$\frac{dV_L}{dt} = \frac{\partial V_L}{\partial h_{os}} \frac{dh_{os}}{dt} + \frac{\partial V_L}{\partial R_t} \frac{dR_t}{dt} \qquad \textbf{(44)}$$

When combined with equation **(43)**, this equation gives

$$\frac{dR_t}{dt} = \frac{Q_{KOPT} - Q_{out} - \frac{\partial V_L}{\partial h_{os}} \frac{dh_{os}}{dt}}{\frac{\partial V_L}{\partial R_t}} \qquad \textbf{(45)}$$

The lens volume, V_L, is given by (see Appendix 1)

$$V_L = \frac{\pi R_t^2 \theta_o \beta h_{os} \sqrt{\frac{\pi}{4}}\, erf\left(\sqrt{\ln\left(\frac{R_t}{R_s}\right)^2}\right)}{\sqrt{\ln\left(\frac{R_t}{R_s}\right)^2}} \qquad \textbf{(46)}$$

In equation **(46)**, erf( ) is the error function. With this equation the partial derivatives with respect to h_{os} and R_t may be evaluated analytically. The resulting equation is

$$\frac{dR_t}{dt} = F_2\left(h_{os}, R_t, Q_{KOPT}\right) \qquad \textbf{(47)}$$

Thus the lens model (equations **(42)** and **(47)**) gives a system of ordinary differential equations which are integrated with an ordinary differential equation solver. Since the KOPT model is also expressed as a system of ordinary differential equations, the two models are combined together in a single computer code. That code

is called HSSM-KO, and is described in Volume 1 of the User's Guide (Weaver et. al., 1993).

Mass transfer of both the hydrocarbon and the chemical constituent from the oil lens to the aquifer occurs from infiltrating rainfall and dissolution caused by flowing groundwater. As the infiltrating rainfall moves through the lens it comes into chemical equilibrium with both the oil and the constituent, and the mass loss rate to the aquifer is

$$\dot{m}_{infil} = q_{wi} \pi R_t^2 c_{wo} \quad \textbf{(48)}$$

where q_{wi} is the volume flux (Darcy velocity) of infiltrating rainfall and c_{wo} is the equilibrium concentration for water in contact with the hydrocarbon (see discussion below).

For dissolution it is assumed that the concentration of the contaminant at the base of the lens is equal to its equilibrium value in water. As the migrating groundwater within the aquifer approaches the lens it has no contaminant within it, and as the groundwater moves beneath the lens, the contaminant diffuses into the groundwater at a rate determined by continuity and vertical dispersion. This is essentially the model presented by Hunt et al. (1988). Let point x = 0 correspond to the upgradient edge of the lens with z being measured downward from the lens, and consider a column of groundwater which moves with velocity v beneath the lens. Then the continuity equation and boundary conditions for this moving column takes the form

$$v \frac{\partial c_w}{\partial x} = D_v \frac{\partial^2 c_w}{\partial z^2} \quad \textbf{(49)}$$

or with $D_v = a_v v$ where a_v is the vertical dispersivity,

$$\frac{\partial c_w}{\partial x} = a_v \frac{\partial^2 c_w}{\partial z^2} \quad \textbf{(50)}$$

with

$$\begin{aligned} c_w(z,0) &= 0 \\ c_w(0,x) &= c_{wo} \end{aligned} \quad \textbf{(51)}$$

where c_{wo} is the contaminant concentration within the water immediately beneath the lens. The solution is

$$c_w(z,x) = c_{wo}\, erfc\left(\frac{z}{\sqrt{4\, a_v x}}\right) \quad \textbf{(52)}$$

The plan view of the lens is shown in **Figure 17**. The total flux into the aquifer from the strip of width dy and of length L(y) is given by

$$\delta m(y) = \int_0^{L(y)} -\eta D_v \frac{\partial c_w(0,x)}{\partial z}\, dx = 2\, c_{wo}\, \eta\, v \sqrt{\frac{a_v L(y)}{\pi}} \quad \textbf{(53)}$$

The length, L(y), of the chord of the circle is

$$L(y) = 2\sqrt{R_t^2 - y^2} \quad \textbf{(54)}$$

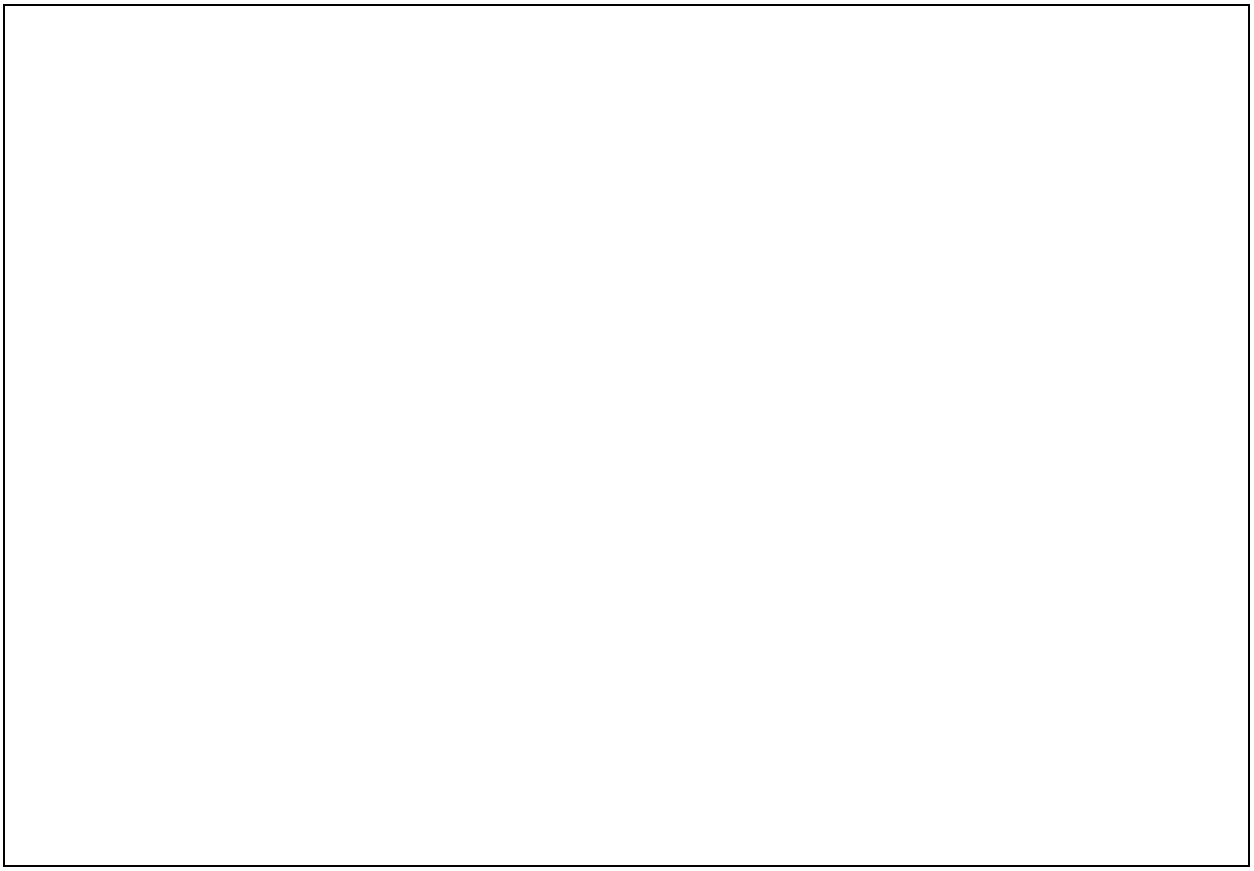

Figure 17 Plan view of the oil lens

so the total flux is given by

$$\begin{aligned} \dot{m}_{diss} &= 2 \int_0^{R_t} \delta\, m(y)\, dy \\ &= 4\, c_{wo}\, \eta\, v\, R_t \sqrt{\frac{2\, R_t\, a_v}{\pi}} \int_0^1 (1 - w^2)^{1/4}\, dw \end{aligned} \qquad \textbf{(55)}$$

The integral in equation **(55)** may be evaluated numerically to give

$$I_d = \int_0^1 (1 - w^2)^{1/4}\, dw \cong 0.87402 \qquad \textbf{(56)}$$

Thus the mass loss due to dissolution within the aquifer is

$$\dot{m}_{diss} = 4\, c_{wo}\, \eta\, v\, R_t\, I_d \sqrt{\frac{2\, R_t\, a_v}{\pi}} \qquad \textbf{(57)}$$

The groundwater source term is given by the sum of the m_{infil} and m_{diss} terms. Thus

$$\dot{m}_{source} = q_{wi}\, \pi\, R_t^2\, c_{wo} + 4\, c_{wo}\, \eta\, v\, R_t\, I_d \sqrt{\frac{2\, R_t\, a_v}{\pi}} \qquad \textbf{(58)}$$

It is apparent that the aquifer source term is dependent on the size of the lens, the infiltration rate and groundwater velocity, the constituent concentration within the lens, and the partitioning characteristics of the constituent between the oil and water.

The groundwater source term given by equation **(58)** requires an estimate of the equilibrium concentration in water in direct contact with the hydrocarbon, c_{wo}. This source term is derived from leaching of trapped hydrocarbon both above and below the lens, and from the spreading lens itself. The constituent mass continuity equations give the total mass, M_t, within the lens plus that trapped within the vadose and saturated zones. This total mass is related to the water equilibrium concentration through the partitioning relationships as follows:

$$M_t = \begin{pmatrix} (\theta_w + \theta_{orv} k_o + \rho_b k_d) V_{vz} + \\ ((\eta - \theta_o) + \theta_o k_o + \rho_b k_d) \dfrac{V_L}{\theta_o} + \\ ((\eta - \theta_{ors}) + \theta_{ors} k_o + \rho_b k_d) V_{sz} \end{pmatrix} c_{wo} \qquad \textbf{(59)}$$

In equation **(59)**, V_{vz} and V_{sz} are the total volumes (including LNAPL, water, and soil) containing residual hydrocarbon in the vadose and saturated zones, and V_L is the hydrocarbon volume in the spreading lens. These volumes are calculated as shown below. With M_t and the volumes known at any time, equation **(59)** provides the effective water phase concentration of the constituent.

It remains to determine the mass which remains behind with the hydrocarbon at residual saturation for a decaying lens after source control. The situation is shown in **Figure 18**. The lens continues to spread even if $dh_{os}/dt < 0$. The hydrocarbon and contaminant within the shaded region of **Figure 18** becomes isolated from the lens with the hydrocarbon at residual saturation and the contaminant dissolved within the hydrocarbon and sorbed on the soil. Since the lens heights are the same at r = R for both times, equation **(39)** gives

$$\frac{\ln \dfrac{R_t(t+\Delta t)}{R}}{\ln \dfrac{R_t(t)}{R}} = \frac{h_{os}^2(t)}{h_{os}^2(t+\Delta t)} \frac{\ln \dfrac{R_t(t+\Delta t)}{R_s}}{\ln \dfrac{R_t(t)}{R_s}} = G \qquad \textbf{(60)}$$

where G is a constant and this equation is written for the lens radius and source height at times t and t+Δt. Since these are calculated from the model and are considered known at the end of time t+Δt, G is a known constant. We then have

$$\frac{R_t(t+\Delta t)}{R} = \left(\frac{R_t(t)}{R} \right)^G \qquad \textbf{(61)}$$

or

$$R = \left(\frac{R_t(t)^G}{R_t(t+\Delta t)} \right)^{\frac{1}{G-1}} \qquad \textbf{(62)}$$

With the radius R known from **(62)**, the change in total volume occupied by residual hydrocarbon (LNAPL, water, and soil) may be found from equation **(100)** of Appendix 1:

$$\Delta V_R = V_T(R; h_{os}(t), R_t(t)) - V_T(R; h_{os}(t+\Delta t), R_t(t+\Delta t)) \qquad \textbf{(63)}$$

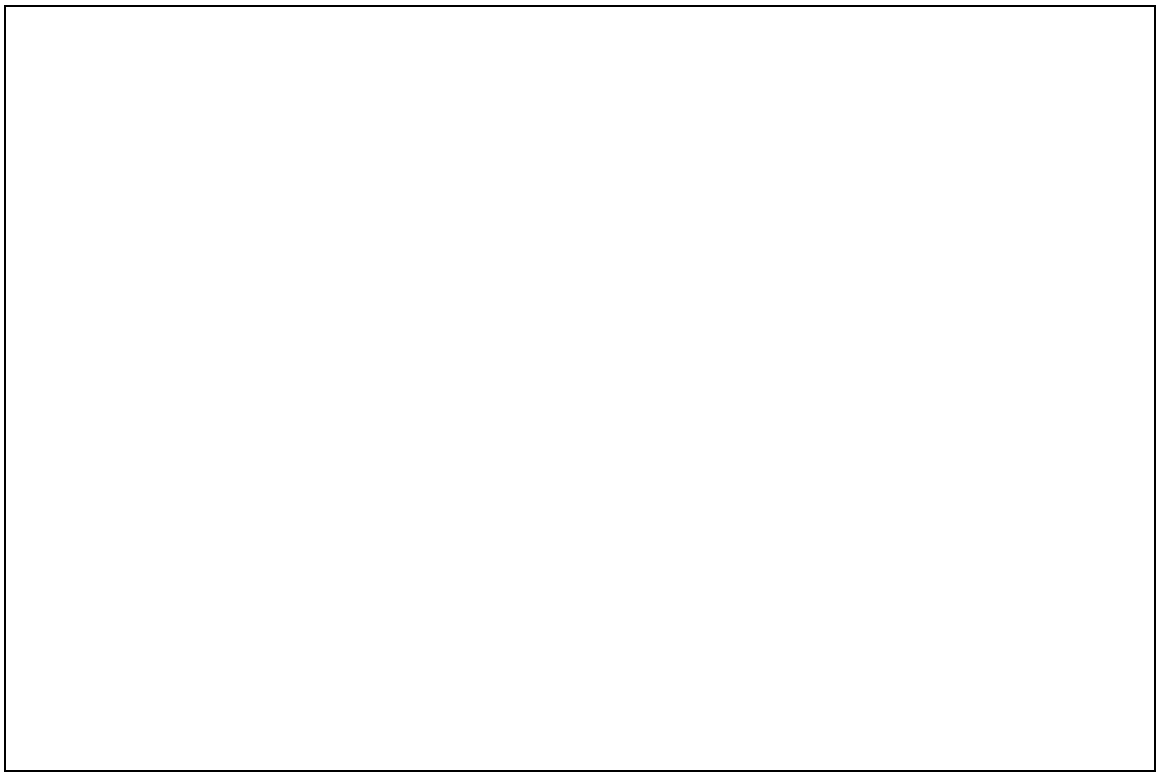

Figure 18 Residual volume for decaying mound

where it is understood that this is used only if $dh_{os}/dt < 0$. The fraction of the residual volume above the lens is $1/\beta$ and the fraction below the lens is $(\beta-1)/\beta$. Thus the volume of free product which becomes trapped during the time step is

$$\Delta V_L = \left(\frac{\theta_{orv}}{\beta} + \frac{\theta_{ors}(\beta - 1)}{\beta} \right) \Delta V_R \qquad \textbf{(64)}$$

The corresponding mass loss is

$$\Delta M_L = c_o(t)\, \Delta V_L \qquad \textbf{(65)}$$

The lens concentration is calculated from the ratio M_L/V_L, where M_L is the total constituent mass within the spreading lens.

Section 4 Gaussian-Source Plume Model

The OILENS model discussed in the last section provides the size of the LNAPL lens and the mass flux to the aquifer as a function of time. Since the contaminant release to the aquifer may occur over a long period of time, aquifer transport leads to the development of a contaminant plume. In order to predict potential exposure concentrations at downgradient receptor locations, a plume model must be coupled with the OILENS model.

The simplest models for predicting plume concentrations from localized sources are point-source plume model. These models consider that the release occurs from a single point, and they are useful for predicting the time development of a plume from a continuous source, though they have the disadvantage of predicting infinite concentrations at the source. The concentrations at the source must be infinite in order to introduce a finite mass flux to the aquifer through a single point. If one is interested in predicting concentrations near the source as well as in the far field, then a source of finite size must be considered. As an alternative, we consider the gaussian-source plume model which provides a useful representation for this purpose. The model presented in this section is very similar to EPACML, the composite landfill model developed for the U. S. Environmental Protection Agency. The basis for EPACML was presented by Huyakorn et al. (1982).

In the gaussian-source plume model, the leachate from a surface facility is assumed to migrate through the unsaturated zone and mix with groundwater flowing beneath the facility. This is shown schematically in **Figure 19**. The groundwater model is set up with a gaussian source placed at the downgradient end of the facility as a boundary condition. Questions of interest concern the depth of penetration of the leachate into the aquifer, and the coupling of the facility release with the aquifer source so that mass balance is achieved.

Figure 20 shows a rectangular facility of length L and width W. The total penetration depth of leachate at the downgradient end of the facility is H. Penetration is caused both by the vertical advection of water as it moves from the vadose zone into the aquifer, and by vertical dispersion:

$$H = H_{adv} + H_{dis} \qquad \textbf{(66)}$$

Both H_{adv} and H_{dis} are estimated using the formulation of EPACML. For H_{dis} it is assumed that the vertical component of the velocity decreases linearly from its inflow value at the water table to zero at the base of the aquifer. Considering the transport of a fluid particle in the resulting flow field leads to

$$H_{adv} = b\left(1 - \exp\left(-\frac{q_{wi} L}{q\, b}\right)\right) \qquad \textbf{(67)}$$

where b is the aquifer saturated thickness, q_{wi} is the infiltration rate through the facility and q is the Darcy velocity in the aquifer beneath the facility.

Figure 19 Basic setup of the gaussian-source plume model

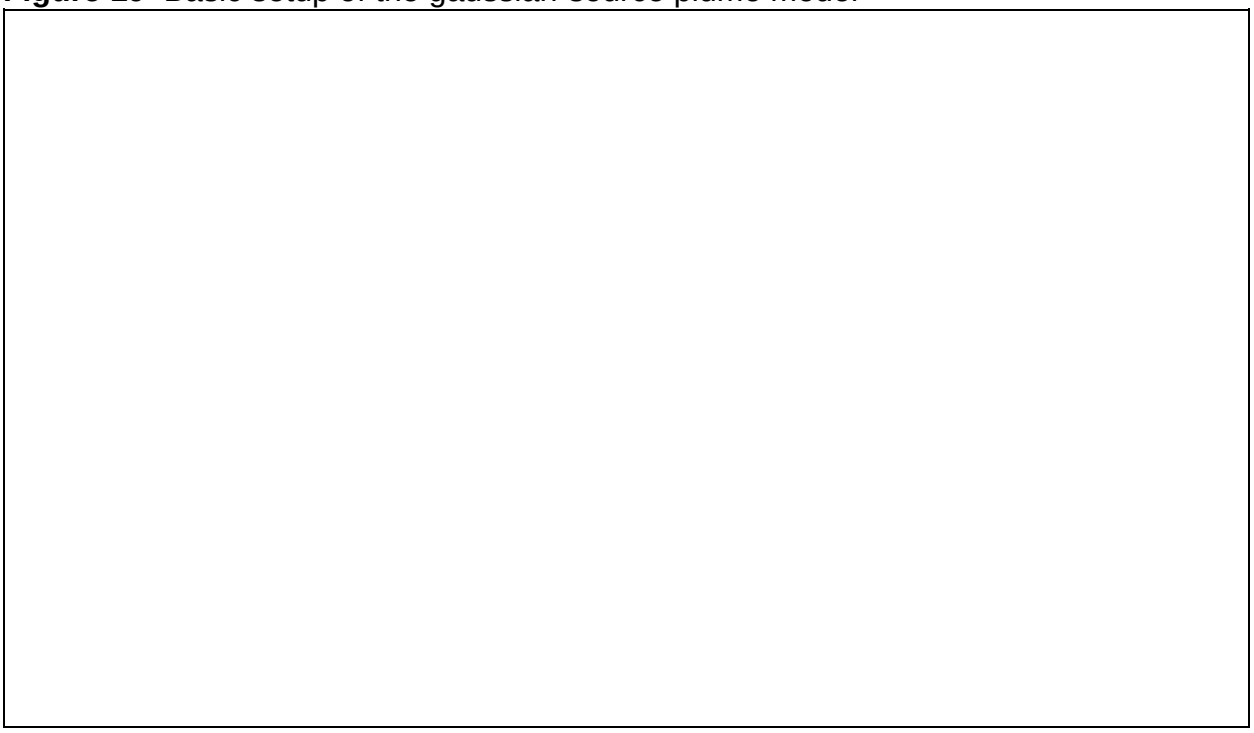

Figure 20 Development of mixing zone beneath the facility

For the dispersive contribution, the width of the dispersive front is proportional to the standard deviation of the concentration distribution. The concentration distribution variance is given by

$$Var = 2\,D\,t \qquad \textbf{(68)}$$

which is Einstein's relation. With $D_v = a_v\,v$ where a_v is the vertical dispersivity, and with $t = L/v$, then

$$H_{dis} = \sqrt{Var} = \sqrt{2\,a_v\,L} \qquad \textbf{(69)}$$

Using these results in equation **(66)** we find

$$H = \sqrt{2\,a_v\,L} + b\left(1 - \exp\left(-\frac{q_{wi}\,L}{q\,b}\right)\right) \qquad \textbf{(70)}$$

Equation **(70)** gives the depth of penetration of the contaminant into the aquifer beneath the facility. If the value of H calculated with equation **(70)** exceeds b, H > b, then in the plume calculations we take H = b.

In the gaussian-source plume model the source is specified by a boundary condition along the x = 0 axis which takes the shape of a gaussian distribution and is specified by

$$c(0, y, t) = c_m \exp\left(\frac{-y^2}{2\,\sigma^2}\right) \qquad \textbf{(71)}$$

where c_m is the maximum concentration and the standard deviation, σ, is a measure of the width of the source. This is shown in **Figure 21**. The boundary condition is coupled with the facility by requiring that the mass flux from the facility equal the advection and dispersion flux from the boundary.

In order to couple the surface release of contaminants with the aquifer boundary condition we use a condition of mass balance. Considering just the advective flux and with reference to **Figure 18** we have

$$\dot{m} = J\,A = q_{wi}\,c_w\,A = q\,H \int_{-\infty}^{\infty} c_m \exp\left(\frac{-y^2}{2\,\sigma^2}\right) dy = \sqrt{2\,\pi}\,q\,H\,c_m\,\sigma \qquad \textbf{(72)}$$

If advection and dispersion are considered, then the background documents for EPACML show that the models are coupled through

$$\dot{m} = \int_{-\infty}^{\infty} \left(q\,c - \eta\, D_L \frac{\partial c}{\partial x} \right) \Big|_{x=0} b\, dy$$

$$= \sqrt{\frac{\pi}{2}}\, H\, q\, \sigma\, c_m \left(1 + \sqrt{1 + \frac{4\lambda^* D_L R_d}{v^2}} \right) \qquad \textbf{(73)}$$

where λ* is the effective decay constant that is defined by

$$\lambda^* = \lambda + \frac{I_r}{\eta\, R_d\, H} \qquad \textbf{(74)}$$

with I_r equal to the diffuse recharge rate outside of the facility.

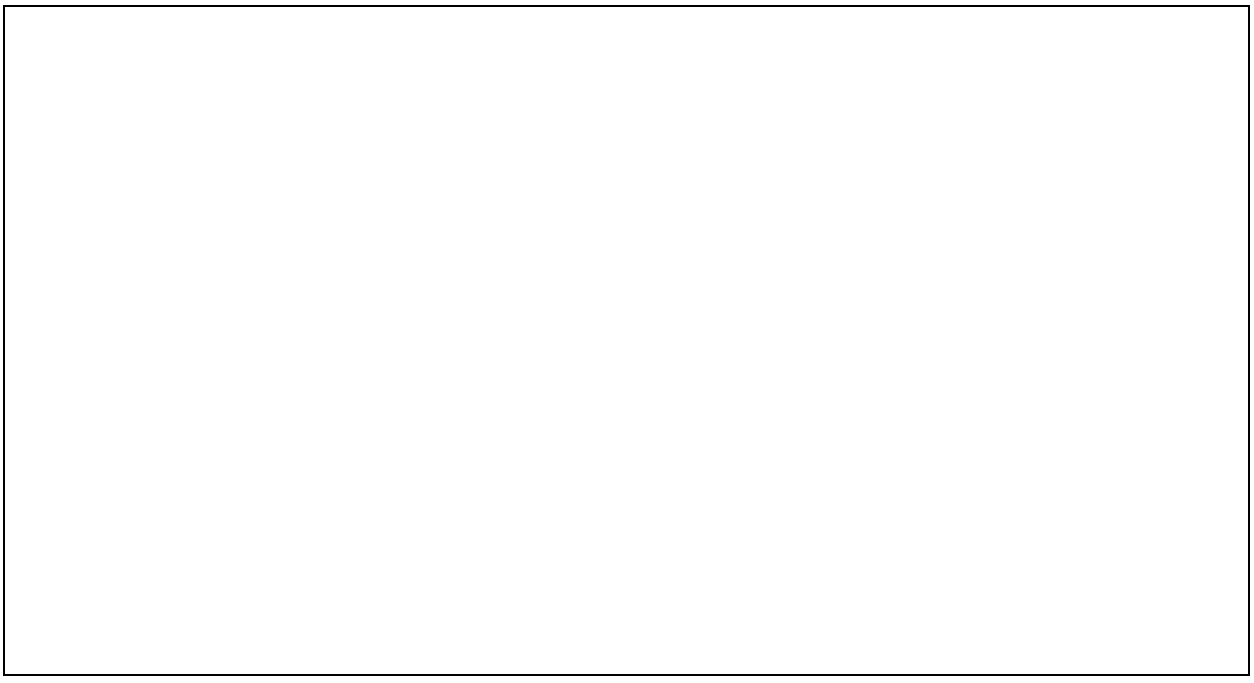

Figure 21 Gaussian distribution which is taken as the boundary condition at the downstream extent of the area beneath the facility

For the implementation of TSGPLUME that is used with HSSM, the diffuse recharge rate outside the facility is taken as being equal to the diffuse recharge rate inside the facility, q_{wi}. The retardation factor, R_d, is defined by

$$R_d = 1 + \frac{\rho_b k_d}{\eta} \qquad \textbf{(75)}$$

where ρ_b is the bulk density, k_d the soil water partition coefficient and η is the porosity. With equation **(73)** we see that the peak concentration beneath the facility is related to the mass rate of flow through

$$c_m = \frac{\dot{m}}{\sqrt{\frac{\pi}{2}} H q \sigma \left(1 + \sqrt{1 + \frac{4\lambda^* D_L R_d}{v^2}} \right)} \qquad \textbf{(76)}$$

A similar relation may be written from equation **(72)**.

Within the aquifer, transport is assumed to occur in two dimensions. In addition, we now want to have the possibility of adding on the effects of dilution from infiltration of surface recharge into the plume, at least in an approximate manner. We assume that recharge serves to dilute the plume and acts as an equivalent decay term. In this case the transport equation is

$$R_d \frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} - D_L \frac{\partial^2 c}{\partial x^2} - D_T \frac{\partial^2 c}{\partial y^2} + \left(R_d \lambda + \frac{I_r}{\eta H} \right) c = 0 \qquad \textbf{(77)}$$

In equation **(77)** the flow is assumed steady and the velocity remains uniform in the x-direction. To simplify notation we use the effective decay coefficient, λ^*, so that the transport equation is

$$R_d \frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} - D_L \frac{\partial^2 c}{\partial x^2} - D_T \frac{\partial^2 c}{\partial y^2} + R_d \lambda^* c = 0 \qquad \textbf{(78)}$$

subject to

$$
\begin{aligned}
c(x,y,0) &= 0 \\
c(0,y,t) &= c_m \exp\left(\frac{-y^2}{2\sigma^2}\right) \\
c(\infty,y,t) &= c(x,-\infty,t) = c(x,\infty,t) = 0
\end{aligned}
\qquad \textbf{(79)}
$$

In order to simplify the development which follows, it is useful to place the problem in dimensionless form. Introduce the following variables:

$$
\begin{aligned}
X &= \frac{v x}{D_L}, \quad Y = \frac{y}{\sigma}, \quad T = \frac{v^2 t}{R_d D_L} \\
\Lambda &= \frac{R_d \lambda^* D_L}{v^2}, \quad D = \frac{D_L D_T}{\sigma^2 v^2}, \quad C = \frac{c}{c_m}
\end{aligned}
\qquad \textbf{(80)}
$$

Then equations **(78)** and **(79)** become

$$
\frac{\partial C}{\partial T} + \frac{\partial C}{\partial X} - \frac{\partial^2 C}{\partial X^2} - D\frac{\partial^2 C}{\partial Y^2} + \Lambda C = 0 \qquad \textbf{(81)}
$$

and

$$
C(0,Y,T) = \exp\left(\frac{-Y^2}{2}\right) \qquad \textbf{(82)}
$$

where the other boundary conditions remain the same. To proceed, it is easiest to first solve the steady-state problem.

For the steady-state problem the transport equation takes the form

$$\frac{\partial C}{\partial X} - \frac{\partial^2 C}{\partial X^2} - D\frac{\partial^2 C}{\partial Y^2} + \Lambda C = 0$$
$$c(0, Y) = \exp\left(\frac{-Y^2}{2}\right) \qquad \textbf{(83)}$$

The solution to this equation follows through application of Fourier transforms, and is found to be

$$C(X, Y) = \sqrt{\frac{2}{\pi}} \int_0^\infty \exp\left(\frac{-w^2 + X - X\sqrt{1 + 4Dw^2 + 4\Lambda}}{2}\right) \cos(wY)\, dw \qquad \textbf{(84)}$$

The mathematical statement of the transient problem is given in equations **(81)** and **(82)**. Application of the Laplace transform reduces the equation to the steady state form whose solution is given above. Using a few well known theorems the general solution is then found to be

$$C(X, Y, T) = X \int_0^T \frac{\exp\left(-\frac{X^2}{4t} - \frac{Y^2}{2 + 4Dt} + \frac{X}{2} - \frac{1 + 4\Lambda}{4} t\right)}{\sqrt{4\pi t^3 (1 + 2Dt)}}\, dt \qquad \textbf{(85)}$$

This is the general solution for a constant boundary condition.

The solution given by equation **(85)** is the transient solution for the case with a fixed boundary condition along X=0. If one wishes to model the case with a time-variable source strength, then one needs the solution for the case with a variable concentration along this boundary. If we assume that the width of the gaussian-source remains constant and that the concentrations change uniformly, then one may find the desired solution directly through use of Duhamel's theorem. Paraphrasing Carslaw and Jaeger (1959, pg. 31), Duhamel's theorem may be stated as follows:

> If C = F(X,Y,T) represents the concentration at point (X,Y) at the time T in an aquifer in which the initial concentration is zero, while its "surface" concentration is the constant function ϕ(X,Y), then the solution of the problem in which the initial concentration is zero, and the surface concentration is B(T) ϕ(X,Y) is given by

$$C(X,Y,T) = \int_0^T B(\omega)\frac{\partial F(X,Y,T-\omega)}{\partial T}\,d\omega = \int_0^T B(T-\omega)\frac{\partial F(X,Y,\omega)}{\partial T}\,d\omega \quad \textbf{(86)}$$

Since the transient solution of equation **(85)** is an integral with T appearing only in the upper limit, the partial derivative with respect to T is simply the integrand. Recognizing this and using equation **(86)**, the solution may be written

$$C(X,Y,T) = \frac{X\exp\left(\frac{X}{2}\right)}{\sqrt{4\pi}} \int_0^T \frac{B(T-\omega)\exp\left(-\frac{X^2}{4\omega}-\frac{Y^2}{2+4D\omega}-\frac{1+4\Lambda}{4}\omega\right)}{\sqrt{\omega^3(1+2D\omega)}}\,d\omega \quad \textbf{(87)}$$

Equation **(87)** forms the basis of the Transient Source Gaussian Plume (TSGPLUME) model. The integral is evaluated using Romberg integration to achieve the desired level of accuracy.

Section 5 The Response of HSSM to Parameter Variation

The HSSM model results are sensitive in varying degrees to all of the input parameters. To give the user a feel for some important parameters of the model, sensitivity analyses are presented in this section. Model parameters are varied individually over arbitrary ranges to demonstrate the behavior of the model. In some cases, the ranges selected are most likely to cover the entire range of variation of the parameter. The results illustrate the variation in chemical concentrations at receptor points that are caused by the variations in input parameter values. This output was chosen to demonstrate the sensitivities of the model because it is a principle output of the model. Other outputs such as the arrival time at the water table (Section 5.1 of Volume 1) or lens radius could also be used to demonstrate the model sensitivities. Depending on the results of interest, the sensitivities may follow different patterns than shown below for the receptor concentration. For example, the lens radius is not greatly affected by variation in the NAPL/water partition coefficient, but the receptor concentration is affected significantly. So for the lens radius, the conclusion is that the NAPL/water partition coefficient is not an important parameter, while it is important for the receptor concentrations. To provide information on other sensitivity measures, Table 10, Table 12 contains the NAPL arrival time at the water table and the lens radius that occurs when the mass flux to the aquifer is at its peak. These tabular results compliment the concentration data and can be used to assess the impact of the parameters on these other outputs.

5.1 Base Scenario

The base scenario for the sensitivity analyses is given in Problem 2, entitled "Transport of Gasoline Constituents in Ground Water to Receptor Locations," presented in Section 5.2 of Volume 1 of the HSSM user's guide (Weaver et al., 1994). The problem statement reads:

> "During a one-day period, 1500 gallons of gasoline leak from a tank surrounded by a circular berm of 2.0 meter radius. Benzene is believed to compose 1.15% by mass of the gasoline. The benzene concentration in the ground water at locations 25, 50, 75, 100, 125 and 150 meters away are needed to assess the impact of the spill. The soil is believed to be predominantly sand in the vicinity of the spill. The aquifer is 10 meters below the ground surface, and its saturated thickness is 15 meters.
>
> "Complete information for the site is not available so many of the HSSM parameters must be estimated. In the absence of better information, parameter values will be estimated from tabulations from the literature. The data set for this example will be organized according to the four dialog boxes for entering data in HSSM-WIN. The parameters for this example are found in the file **X2BT.DAT**, which is found on the HSSM-WIN distribution diskette.

The complete set of results from this problem is found on pages 59 to 62 of Volume 1. Concentration histories at four receptor locations are shown by solid lines in **Figure 22**. Each receptor location lies along the centerline of the flow system, as shown in **Figure 23**. To extend the effects demonstrated by this example, simulations results for the 50 meter receptor are shown for toluene and the xylenes (lumped para-, meta- and ortho-xylene) in **Figure 24**. The peak concentration and arrival time of the peak are influenced by the initial concentration of the constituent in the NAPL and the NAPL/water partition coefficient. These parameter values are shown in Table 3. Comparing the toluene result with the benzene result shows that increasing the initial constituent concentration in the NAPL increases the peak concentration. Comparing the result for xylenes against benzene shows that even with higher initial concentration in the NAPL, the receptor peak concentration is lower for the xylenes because of the increased partition coefficient. Increasing the partition coefficient tends to lower receptor concentrations because the constituent remains in the NAPL thus decreasing the peak mass flux to the aquifer.

Table 3 Constituent Parameters for benzene, toluene and the xylenes		
Constituent	**Initial Concentration in the NAPL (mg/L)**	**NAPL/water partition coefficient**
benzene	8,208	311
toluene	43,600	1,200
xylenes	71,800	4,440

The centerline locations are used below to show the effects of parameter variation. To simplify presentation of the results, only the peak concentration is plotted against the arrival time. The peak concentration for each receptor is indicated by an open square in **Figure 22**.

The response of HSSM to parameter variation can follow nine patterns that are illustrated in **Figure 25**. Squares indicate the peak concentrations and arrival times from the **X2BT.DAT** data set for receptors at 25 m, 50 m, 100 m and 150 m down gradient from the source. By varying input parameters, this curve may shift in various directions as indicated by the labeled arrows. Thus the curve may shift vertically upward if the peak concentration increases and the arrival time remains the same (arrow labeled A), or increased concentrations may occur with earlier (arrow labeled H) or later (arrow labeled B) arrival times. If variation of the parameter has a negligible impact on the peak concentration and the arrival time, then it is classified as "I."

As will be seen to be obvious below, this classification system only approximately captures the variation in the results. Some results would be better described by a rotation of the curve, as the impacts change character with the distance to the receptor (e.g., Figure 49). Also some effects which are dominant near the source tend to die out further away (e.g., **Figure 50**). Deviations such as these are noted in the following text.

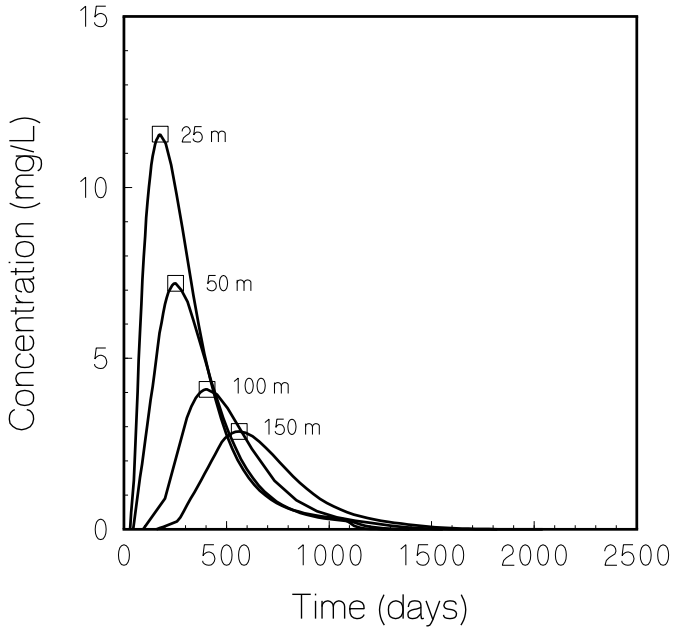

Figure 22 Concentration histories for the `X2BT.DAT` data set

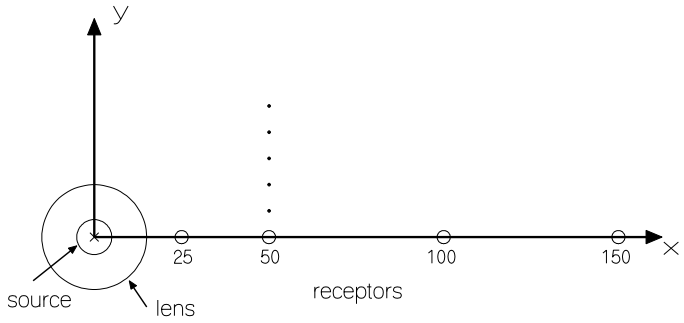

Figure 23 Plan view of HSSM model scenario

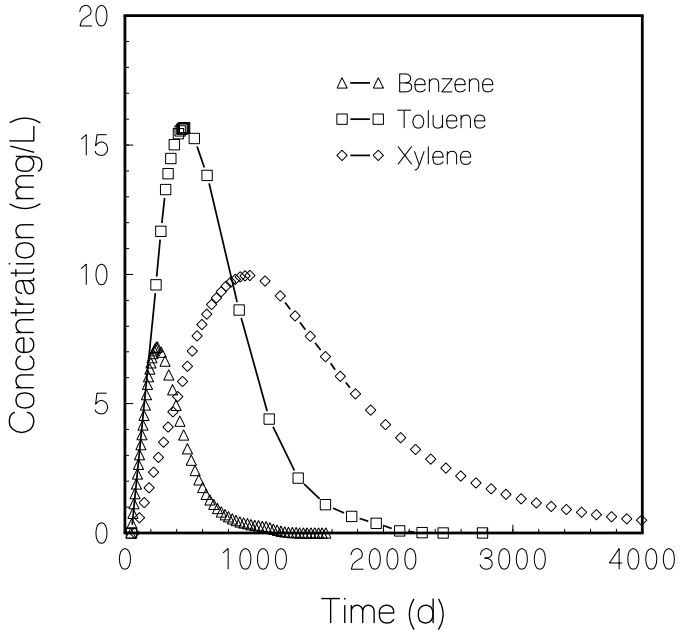

Figure 24 Concentration Histories at 50 meters for benzene, toluene and the xylenes.

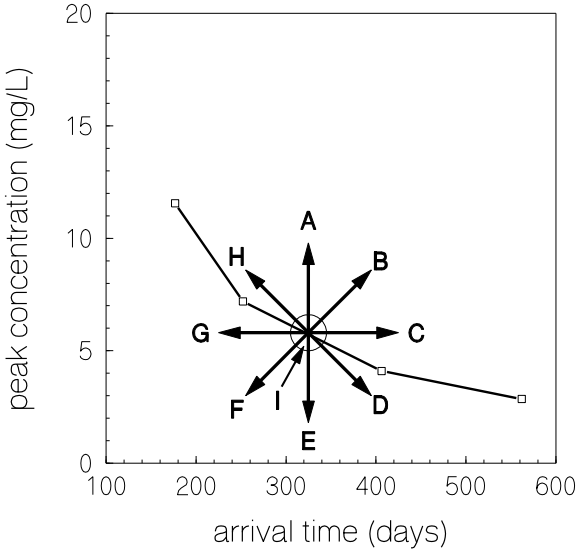

Figure 25 Possible HSSM responses to parameter variation

5.2 Usage of Parameters in HSSM

To determine the concentration history at each receptor point, the aquifer model, TSGPLUME, uses a portion of the HSSM input parameter set, and the mass flux that is determined by HSSM-KO as a boundary condition. Thus, some portion of the TSGPLUME results directly from the effects of HSSM parameter variation on solute transport in the aquifer. A significant portion also depends on the boundary condition. The mass flux to the aquifer is determined by HSSM-KO and depends on the parameters of KOPT and OILENS. Some of these parameters are not used by TSGPLUME, but their effect is felt through the boundary condition.

Generally, the arrival time of the peak concentration at a given receptor depends on the arrival time of the NAPL at the water table, and the time at which the mass flux to the aquifer is a maximum. The arrival time at the water table depends on the KOPT model results and thus represents the effects of vadose zone flow and transport. The time of maximum mass flux depends upon the rate of spreading of the NAPL lens, since the mass flux is a function of the radius of the lens (equation **(58)**). Further, equation **(58)** shows that the mass flux increases with radius because of the terms that include R_t^2 and $R_t^{3/2}$. Also appearing in that equation is the aqueous phase concentration that is in equilibrium with the lens, C_{wo}. As can be noted from equation **(59)**, this concentration depends on the lens volume, which in turn depends upon the lens radius. So for a given set of parameters, the lens radius, R_t, and concentration C_{wo} (plotted for **X2BT.DAT** in **Figure 26**) determine the variation in mass flux. This figure shows that the aqueous concentration reaches a maximum at a relatively low radius, and declines while the radius continues to increase. Notice that because of the contribution of the lens radius, the peak mass flux occurs later than does the peak concentration in this example.

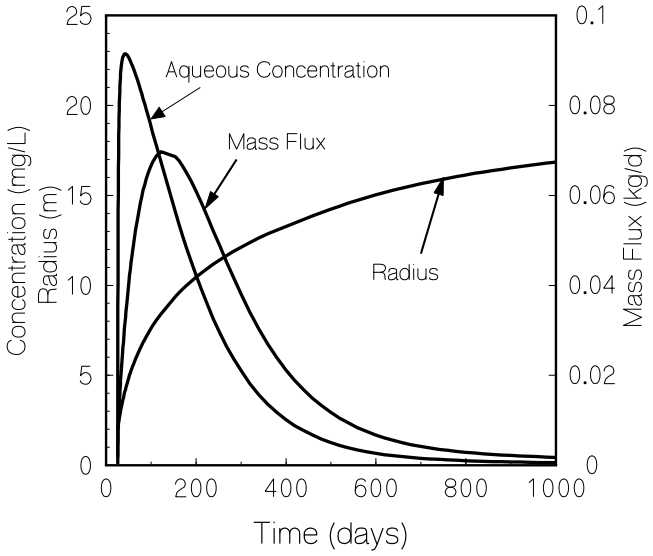

Figure 26 NAPL lens radius and aqueous concentration for the example 2 data set

Equation (76) is used in TSGPLUME to determine the centerline concentration at the boundary (the leading edge of the lens). Combining equations **(58)** and (76), and noting the definition of the penetration thickness (equation **(70)**) gives

$$c_m = \frac{q_{wi}\,\pi\,R_t^2\,c_{wo} \;+\; 4\,c_{wo}\,\eta\,v\,R_t\,I_d\sqrt{\frac{2\,R_t\,a_v}{\pi}}}{\sqrt{\frac{\pi}{2}}\left(\sqrt{2\,a_v\,(2\,R_t)} + b\left(1 - \exp\left(\frac{q_{wi}\,(2\,R_t)}{q\,b}\right)\right)\right) q\,\frac{R_t}{2}\left(1 + \sqrt{1 + \frac{4\lambda^* D_L R_d}{v^2}}\right)} \qquad \textbf{(88)}$$

The length of the facility, L, is taken as twice the lens radius, and the standard deviation of the gaussian distribution, σ, is taken as one quarter of the lens radius. In equation **(88)**, the radius R_t refers to the single value that is used in TSGPLUME. As noted in Volume 1, a value is selected from the radius history (**Figure 26**). Normally the radius that occurs when the mass flux is at its peak is chosen. The concentrations determined from equation **(88)**, thus reflect the balance between mass flux and dilution, both of which increase with the radius of the lens. Peak concentrations from equation (76) are used below to assess the impact of increasing lens radius on the receptor concentrations, because higher receptor concentrations are expected if the concentration at the source increases.

Before beginning the presentation of the results, two features of HSSM must be noted. First, because the procedure used for finding the peak concentration in HSSM-T is an approximation (see discussion on page 14 of Volume 1 of the user's guide), the arrival time is precise only to within a few days (typically 2 to 5). The peak concentrations determined by HSSM-T can vary, depending primarily upon the time step taken in the program. Thus the values reported in this section may not be exactly reproduced if the cases were rerun with a slightly different time interval in HSSM-T. The trends are valid, however, and care was taken to determine that variations in the peak concentration and arrival times were significant, rather than artifacts of the procedures used.

Second, for each parameter that affects the size of the NAPL lens, the appropriate value of the maximum NAPL saturation in the lens, $S_{o(max)}$, must be found. The NTHICK utility (Appendices 3.3 and 7 of Volume 1 of the User's Guide) or an automated version, NTHICK2, was used for these calculations. The $S_{o(max)}$ values can be found in the data sets on the distribution diskette. Generally four or five runs of HSSM-KO were required to converge to a value of $S_{o(max)}$ within a tolerance of 0.0001.

5.3 Sensitivity Results

Each physical and chemical parameter of HSSM was used in the analysis. The parameters were varied over a plausible, but arbitrary, range. The data sets are found in the EXAMPLES\SENS directory on the distribution diskette. Table 12 in Appendix 2 lists the file names for each parameter. The results are classified according to the possible impacts on the peak concentration vs. time-to-peak curve (**Figure 25**). A total of 34 sets of parameter variation trials are reported below. Of these, three are repetitions of some other parameter (smear zone repeats the capillary thickness parameter, "all dispersivities" repeats the individual dispersivities, and aquifer thickness has two response types) and two are not independent physical parameters (percent maximum radius and NAPL saturation in the lens). **Figure 27** shows the frequency of response types from the 29 remaining sets of parameter variation trials. The smallest slices (A, B, and G) each represent one parameter and compose 3.45% of the pie. The largest slice (I) has is the null response and contains eight parameters. Of the remaining slices, the largest are those of the parameters where both the peak concentration and its arrival time are affected (D six parameters, F four parameters, and H six parameter). There are relatively few parameters that impact only the concentration or the arrival time (A one parameter, C no parameters, E two parameters, G one parameter).

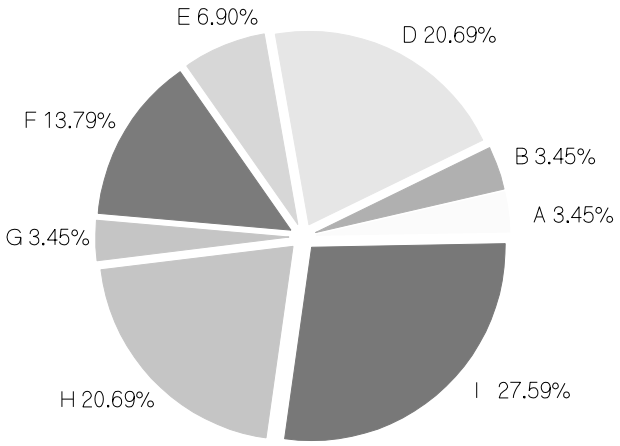

Figure 27 Pie chart showing frequency of parameter variation responses

For each response type, A through I, the parameters are listed in tables followed by a discussion of the results. The following tables (e.g., Table 4) list the parameter names (column 1), the classification or type at the 150 m receptor (column 2), the minimum and maximum parameter values used (columns 3 and 5), the value from the base case (column 4), and the HSSM modules that are affected by the parameter (column 6). The classification at the 150 m receptor is included, because for a number of the parameters the impact changes with distance from the receptor. In cases where the impact is great enough to change the classification, the type listed in column 2 differs from the type of the table. Sometimes a type is indicated with an arrow (→) which means that the results tend toward the indicated type. The 6th column is included because the impacts of certain parameters are similar. Often these parameters are related, both in the HSSM modules that use them and in their physical or chemical significance. Codes are listed for each module of HSSM in the order: KOPT, OILENS, and TSGPLUME. A "Y" or "N" in this column indicates direct use or nonuse of the

parameter, respectively. The "I" indicates an indirect impact, meaning that the parameter only impacts the input to the module and not the equations solved in the module itself. Note, for example, that porosity has a direct impact on KOPT, OILENS and TSGPLUME and the code in column 6 is YYY. NAPL viscosity, however, directly impacts KOPT and OILENS; its impact on TSGPLUME is indirect because the NAPL viscosity is not used in the aquifer model, but affects the model results through the boundary condition. Table 13 in Appendix 2 summarizes the fraction of parameters which impact each module, classified by response type.

Table 4 Parameter with Type A Response (Increasing Peak Concentration, Constant Arrival Time)

Parameter	Type at 150 m	Minimum Value	Value used in Problem 2 Section 5.2 of Volume 1	Maximum Value	Module Impacts (K,0,T)
Initial constituent concentration in the NAPL, C_{oini}	A	820 mg/l (0.0115 %) ρ_o = 0.71 g/cm^3 K_o = 307	8208 mg/l (0.115 %) ρ_o = 0.72 g/cm^3 K_o = 311	12300 mg/l (0.172 %) ρ_o = 0.72 g/cm^3 K_o = 314	YII

5.4 HSSM Response: Increasing Peak Concentration, Constant Arrival Time

5.4.1 Initial Constituent Concentration in the NAPL

The only parameter that causes the peak concentration to increase with constant arrival time is the initial constituent concentration in the NAPL, C_{oini} (Table 4) . When C_{oini} changes so, do the NAPL density, and the NAPL/water partition coefficient. The values shown in Table 4 were determined by using the RAOULT utility (Appendices 3.2 and 6 of Volume 1 of the User's Guide). **Figure 28** shows that the peak concentrations increase with the initial constituent concentrations and that the arrival times are unaffected. C_{oini} is used directly only by KOPT, but determines the magnitude of the mass flux into the NAPL lens and the aquifer. As C_{oini} increases from 820 mg/L to 12300 mg/L the peak mass flux to the aquifer increases from 0.0070 kg/d to 0.10 kd/d. The resulting source concentration (equation **(88)**) in the aquifer increases from 3.905 mg/L to 55.73 mg/L. At the same time the water table arrival times, and time to peak mass flux remain relatively constant (+/- 4 days).

Table 5 Parameter with Type B Response (Increasing Peak Concentration, Increasing Arrival Time)

Parameter	Type at 150 m	Minimum Value	Value used in Problem 2, Section 5.2 of Volume 1	Maximum Value	Module Impacts (K,O,T)
Source radius, R_s	B	0.2 m	2.0 m	4.0 m	YII

5.5 HSSM Response: Increasing Peak Concentration, Increasing Arrival Time

5.5.1 Source Radius

The source radius (defined in **Figure 23**) is the only parameter to cause the both the peak concentration and the arrival time to increase (Table 5). The low peak concentration at low radii are generally related to the reduced amount of contaminant introduced into the aquifer with a low source radius. The arrival time increases with source radius, because the peak mass flux (as input to TSGPLUME) occurs later as the source radius increases. As the radius increases from 0.2 meters to 4.0 meters, the time for the peak mass flux increases from 31.9 days to 142.0 days after the NAPL arrives at the water table. This time lag accounts for the difference in the peak concentration arrival times. The peak mass fluxes to the aquifer and the effective source concentrations also increase with the radius of the source.

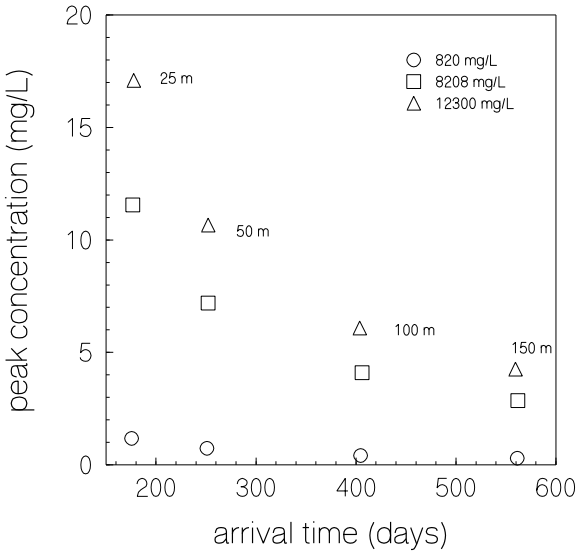

Figure 28 Peak concentration vs arrival time for variation in the initial contaminant concentration

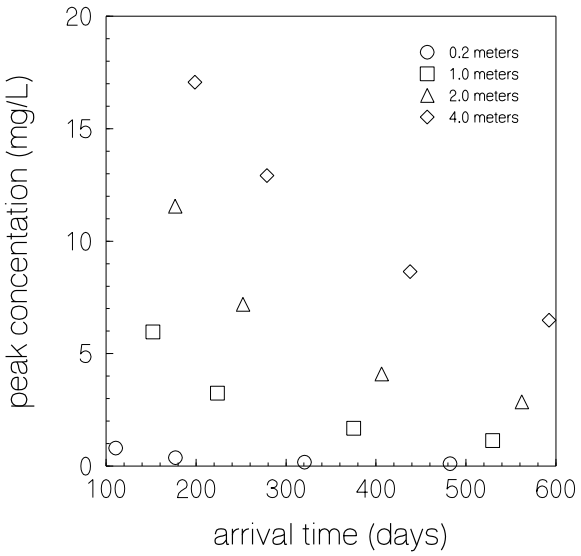

Figure 29 Peak concentration vs arrival time for variation in the source radius

5.6 HSSM Response: Decreasing Peak Concentration, Increasing Arrival Time

5.6.1 Depth to water

In HSSM, the NAPL travels through the vadose zone to reach the aquifer. Thus the depth to water determines the distance from the surface to the water table. This distance affects the travel time through the vadose zone and its ability to retain NAPL. The first effect is that as the depth to the water table increases, so does the arrival time of the NAPL at the water table (Table 12). Thus, the times for the peak mass flux to the aquifer increase with the depth to water. Since some of the NAPL is retained within the vadose zone (because of the residual NAPL saturation of 0.10), the amount of NAPL reaching the water table also decreases with depth to water. This is reflected in reduction in lens radius for the peak mass flux, and resulting reductions in peak mass flux and effective concentration. These lead to reduction in peak receptor concentration with depth to water (**Figure 30**).

Table 6 Parameters with Type D Response (Decreasing Peak Concentration, Increasing Time-to-Peak)					
Parameter	**Type at 150 m**	**Minimum Value**	**Value used in Problem 2 Section 5.2 of Volume 1**	**Maximum Value**	**Module Impacts (K,O,T)**
Depth to water	D	7.5 m	10.0 m	12.5 m	YII
Porosity and bulk density, η and ρ_b	D	0.35 1.72 g/cm^3	0.43 1.51 g/cm^3	0.50 1.32 g/cm^3	YYY
NAPL viscosity, μ_o	→C	0.30 cP	0.45 cP	0.60 cP	YYI
Vadose zone residual NAPL saturation, S_{orv}	D	0.0	0.05	0.075	YYI
Soil/water partition coefficient for the constituent	→C	0.0415 L/kg	0.083 L/kg	0.1660 L/kg	YYY
NAPL/water partition coefficient for the constituent	D	250	311	375	YYI
Smear zone thickness	→C	0.065 m	0.065	1.0 m	NYI

5.6.2 Porosity and bulk density

Porosity, η, and bulk density, ρ_b, both play a role in determining receptor concentrations and are related by

$$\rho_b = \rho_s (1 - \eta) \quad \textbf{(89)}$$

where ρ_s is the solid density. Increasing the porosity delays the NAPL's arrival at the water table, because with higher porosity, more of the NAPL is needed to fill a given volume of the vadose zone. The same effect causes the resulting lenses to be smaller. The lens radius at peak mass flux to the aquifer thus declines, resulting in lowered mass flux and effective concentrations at the source. These effects tend to decrease the receptor concentrations. Sorption of the constituent, however, declines as the porosity increases, because of equation **(89)**. This effect would tend to increase the receptor concentrations, but is not dominant in this example.

5.6.3 NAPL viscosity

The NAPL viscosity in part determines the effective conductivity to the NAPL. Equation **(2)** shows that increasing the NAPL viscosity decreases the effective conductivity. This behavior impacts the receptor concentrations indirectly, as it increases the arrival time at the water table; and slows the rate of expansion of the NAPL lens. With increasing viscosity, NAPL lenses tend to be thicker and thus have lower radii. These characteristics of the lens cause the peak mass flux to the aquifer to occur somewhat later with the range of viscosity used here. Much higher viscosities are encountered with other types of oils, these would accentuate

these effects. As the lens radius at peak mass flux declines, so does the peak mass flux and to some extent the effective source concentration. The result is that increasing the NAPL viscosity delays the arrival and reduces the peak concentration (**Figure 32**).

5.6.4 Vadose zone residual NAPL saturation

In the vadose zone, increasing the residual NAPL saturation increases the amount of NAPL retained per unit volume of soil. The effect is similar to that caused by increasing the depth to water (**Figure 30**), with the exception that the OILENS results are also affected by the vadose zone residual NAPL saturation. As the vadose zone residual NAPL saturation increases, the arrival time at the water table increases. This is due largely to the impact on the NAPL relative permeability function, because as the residual NAPL saturation increases, the amount of NAPL needed to achieve a certain relative permeability also increases (equation **(4)**). The phase flow speeds in vadose zone (equations **(12)** and **(13)**) both depend on the relative permeability and thus both decline with increase vadose zone residual NAPL saturation. In OILENS as the input flux declines, the lens collapses (**Figure 18**) and some NAPL is retained in the vadose zone at the residual vadose zone saturation. The constituent that is held in the residual NAPL is gradually leached into the aquifer by the aquifer recharge. This leaching contributes a relatively small amount to the peak mass flux. In this example, the mass flux to the aquifer is more highly dependent upon the size of the lens at the water table. The result is the type D response shown in **Figure 33**.

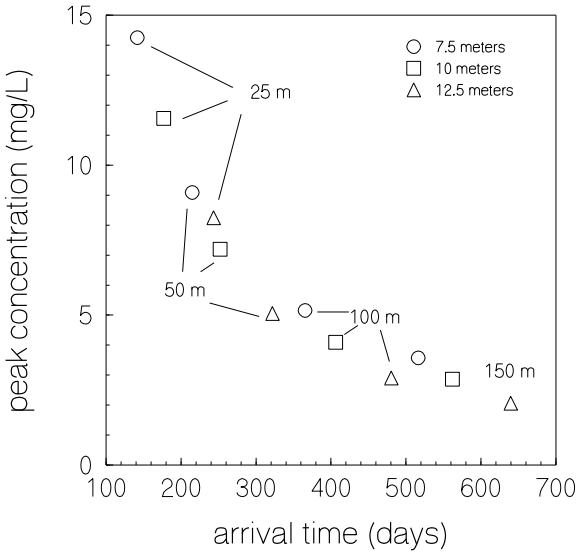

Figure 30 Peak concentration vs arrival time for variation of depth to water

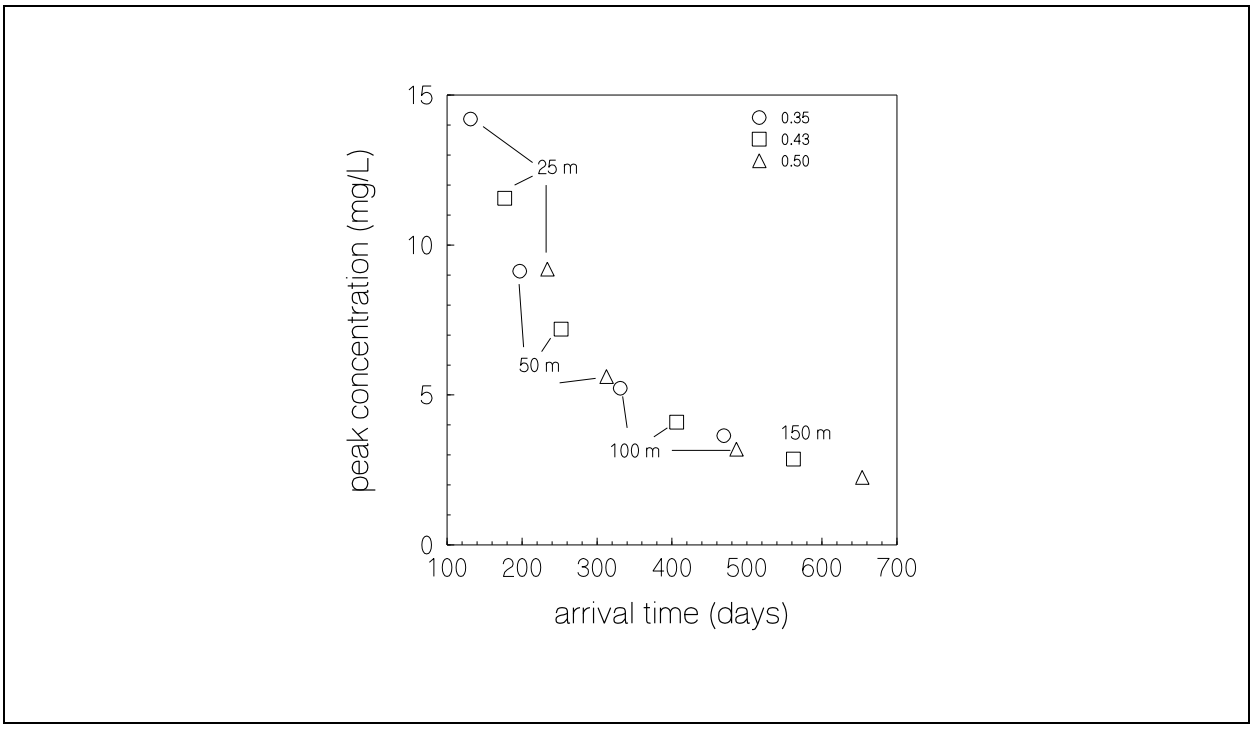

Figure 31 Peak concentration vs arrival time for variation of porosity and bulk density

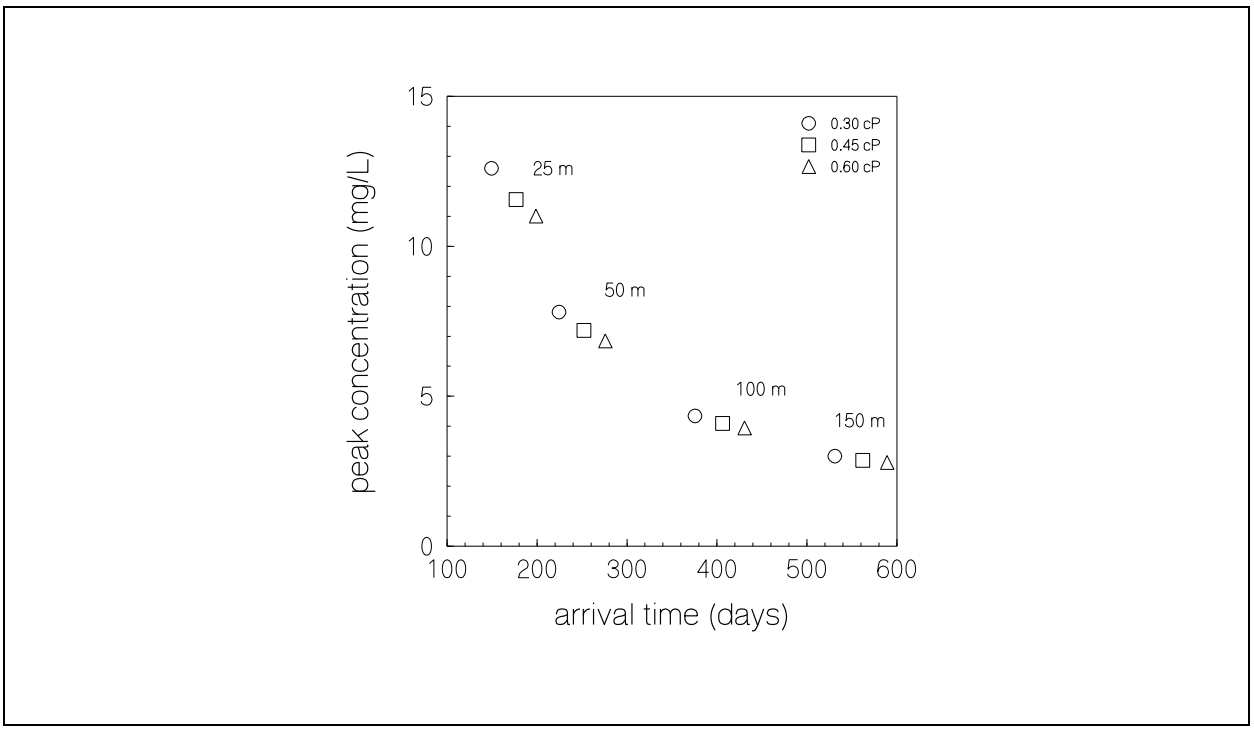

Figure 32 Peak concentration vs arrival time for variation of the NAPL viscosity

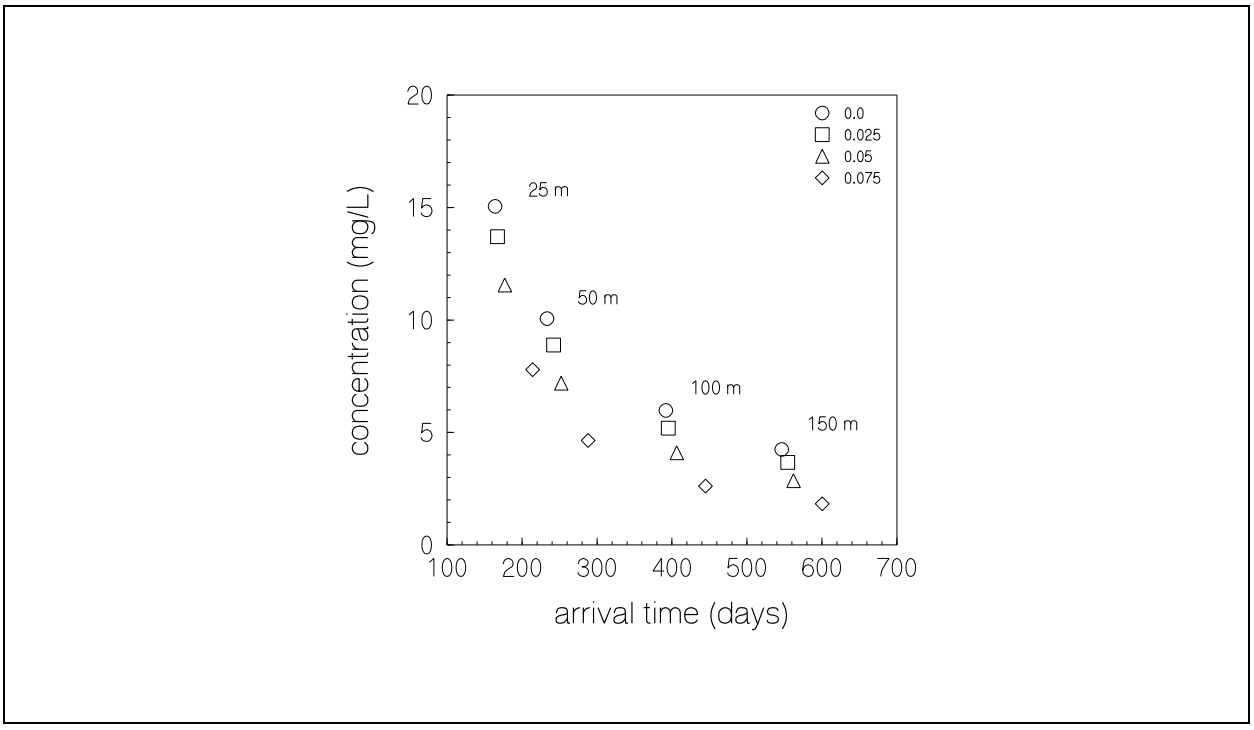

Figure 33 Peak concentration vs arrival time for variation of the vadose zone residual NAPL saturation

5.6.5 Soil/water partition coefficient for the constituent

Although the partition coefficient, k_d, is used by all three modules of HSSM, the KOPT and OILENS results are only slightly affected by changing k_d. This result occurs because the NAPL constituents are usually hydrophobic and tend to remain mostly in the NAPL phase. Flow of the NAPL itself, largely determines the distribution of the constituent within KOPT and OILENS. The main impact of k_d occurs in the aquifer (**Figure 34**) where increased partitioning leads to lower peak concentrations (because sorbed constituent does not add to aqueous concentration) and to later arrival times (because sorbed mass is immobile and the overall rate of transport is reduced). For this example, the partition coefficient was changed by assuming varying amounts of organic carbon in the aquifer. The k_d values of 0.0415 L/kg, 0.083 L/kg, and 0.166 L/kg correspond to organic carbon fractions of 0.0005, 0.001 and 0.002, respectively and an organic carbon/water partition coefficient, K_{oc}, of 83 L/kg for benzene.

5.6.6 NAPL/water partition coefficient for the constituent

The NAPL/water partition coefficient is defined as the ratio between the water phase and the NAPL phase concentrations of the constituent (equation **(26)**). As K_o increases, the time of the peak mass flux increases. This behavior is due to higher retention of the constituent in the NAPL phase, and more time required to remove the constituent from the lens. The peak fluxes to the aquifer decline, because on average the concentrations in the water phase are reduced. In most of the cases examined for this Section increasing lens radius resulted in increased mass flux to the aquifer. The NAPL/water partition coefficient is one example where the radius increases with the parameter value, but the mass flux declines (due to the decreased concentrations).

5.6.7 Smear zone thickness

The smear zone that can be built into HSSM data sets represents water table fluctuation that can spread the NAPL over a certain thickness. When the thickness of the smear zone increases, the size of the lens is reduced. The effect is to reduce the mass flux to the aquifer, thus delaying and reducing the magnitude of the peak (Figure 36). The effective source concentrations are highest, however, for the thicker smear zones.

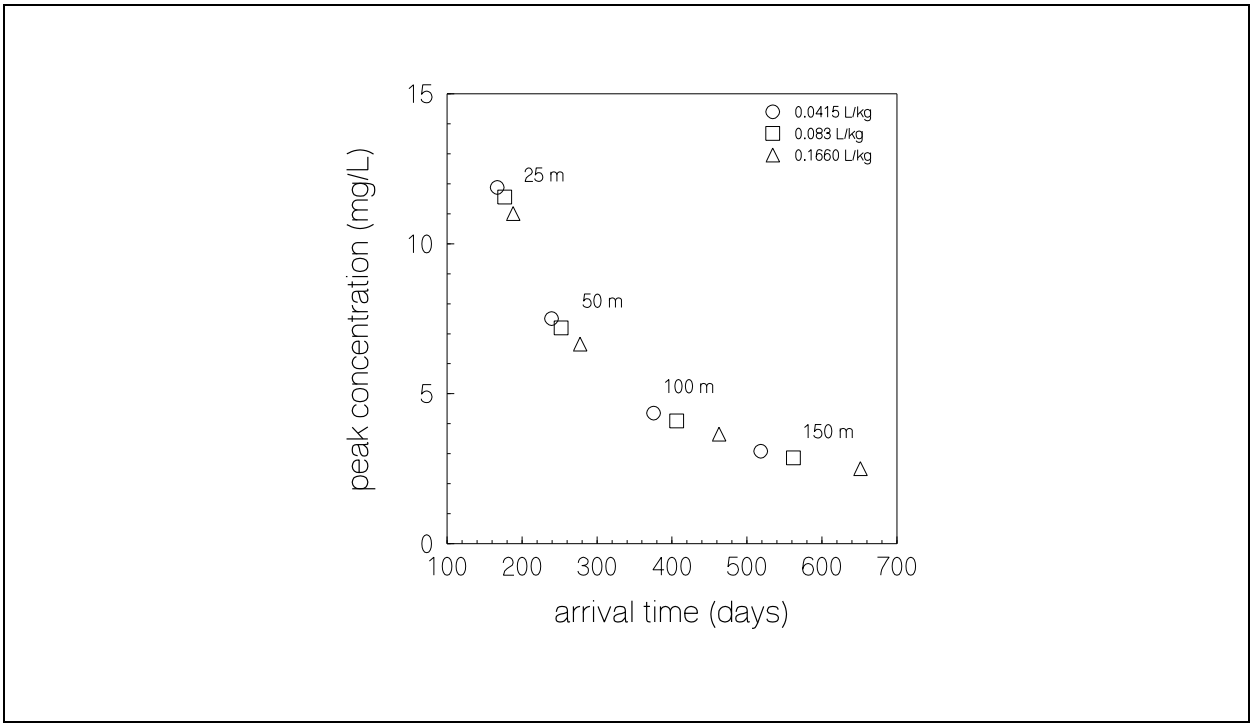

Figure 34 Peak concentration vs arrival time for variation of the soil water partition coefficient for the constituent

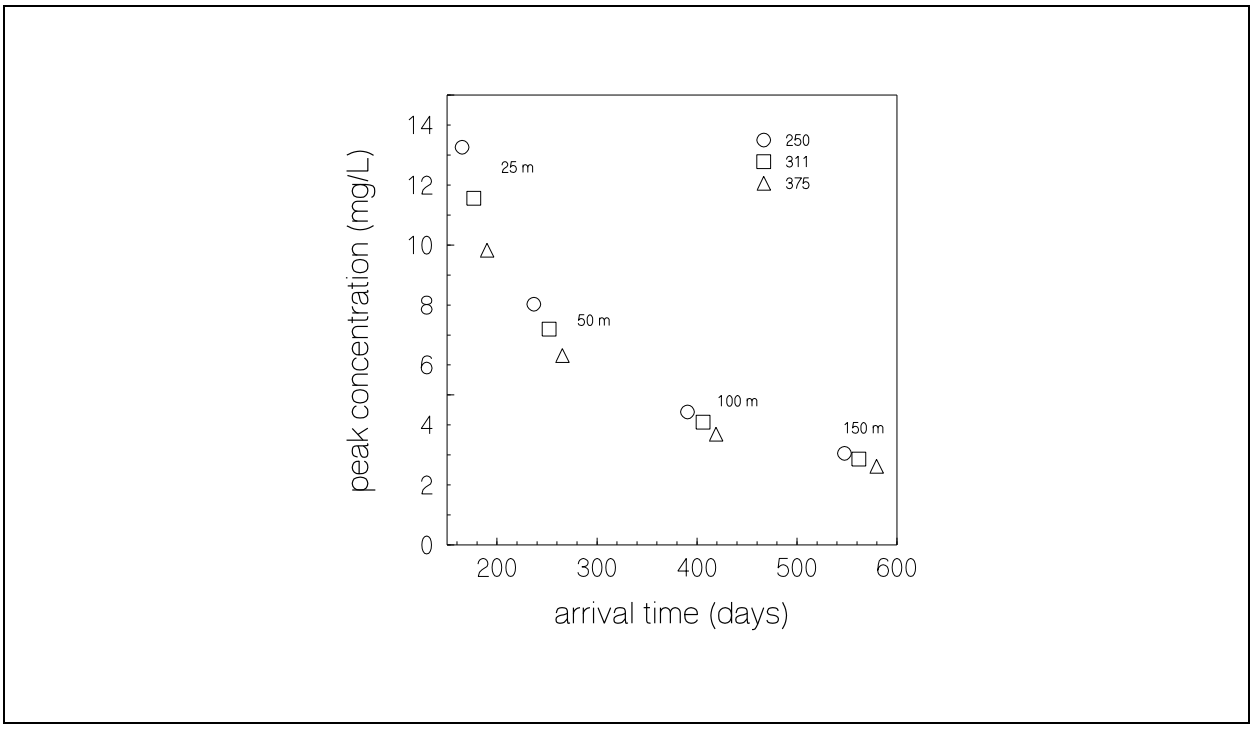

Figure 35 Peak concentration vs arrival time for variation of the NAPL/water partition coefficient

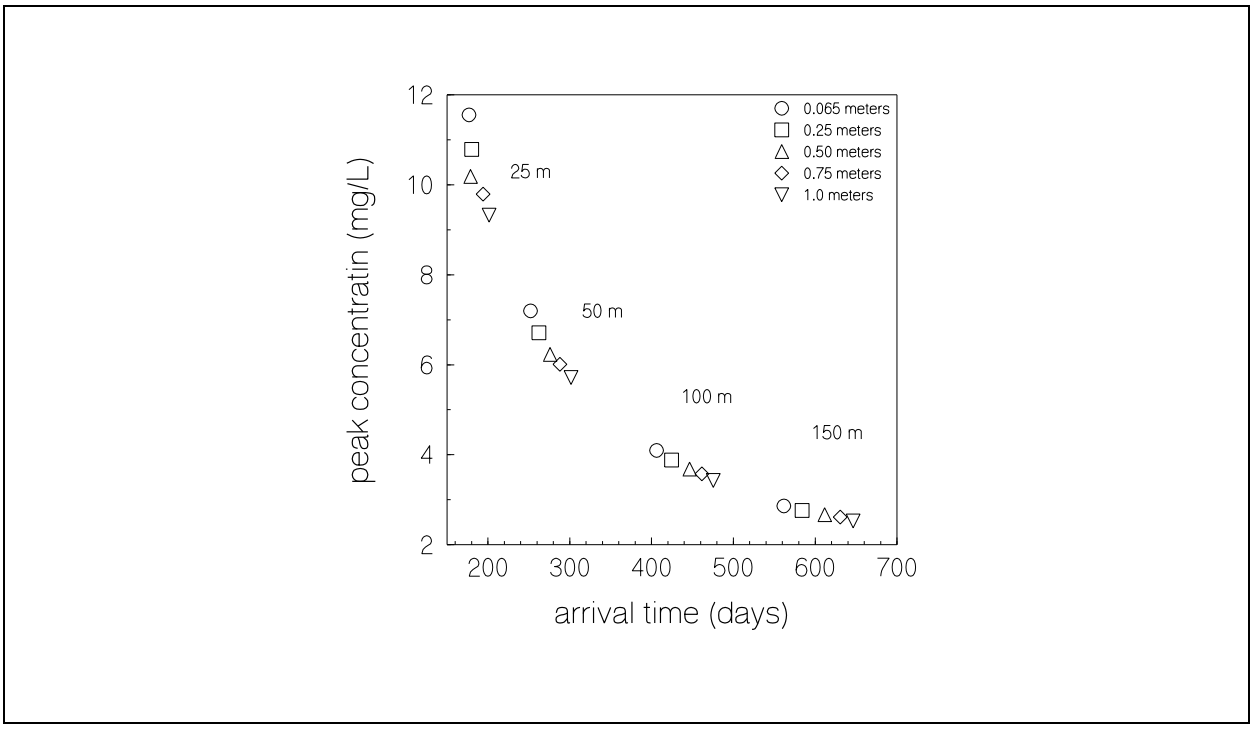

Figure 36 Peak concentration vs arrival time for variation of the smear zone thickness

5.7 HSSM Response: Decreasing Peak Concentration, Constant Arrival Time

Table 7 Parameters with Type E Response (Decreasing Peak Concentration, Constant Time-to-Peak)

Parameter	Type at 150 m	Minimum Value	Value used in Problem 2 Section 5.2 of Volume 1	Maximum Value	Module Impacts (K,O,T)
Aquifer thickness (less than penetration thickness)	E	0.5 m	15 m	20 m	NNY
Transverse horizontal dispersivity, a_T	E	0.5 m	1.0 m	1.5 m	NNY

5.7.1 Aquifer thickness (less than penetration thickness)

In TSGPLUME, the penetration thickness represents the thickness of the aquifer that is contaminated with the NAPL constituent. In this way the model does not assume that the constituent is mixed over the entire aquifer thickness, but is confined to a region near the water table. For this example, the calculated penetration thickness is 1.966 m (equation **(70)**) and the aquifer thickness is 15 m. For the purposes of this study, simulations were run with the aquifer thickness set to values that are less than the calculated penetration thickness (0.5 m, 1.0 m and 1.5 m). In such cases, the concentration increases because the chemical is introduced into a volume that is smaller than that determined from the penetration thickness. The concentrations here indeed increase, with no change in arrival time (**Figure 37**). When the aquifer thickness exceeds the penetration thickness (1.966 m, 15 m, and 20 m), the concentrations become independent of the aquifer thickness.

5.7.2 Transverse horizontal dispersivity

The transverse horizontal dispersivity, a_T, is only used in the aquifer model and determines the amount of lateral spreading of the contaminant plume. The concentrations decline along the centerline of the flow system (Figure 38), because there is increased horizontal spreading of the plume as the transverse dispersivity increases.

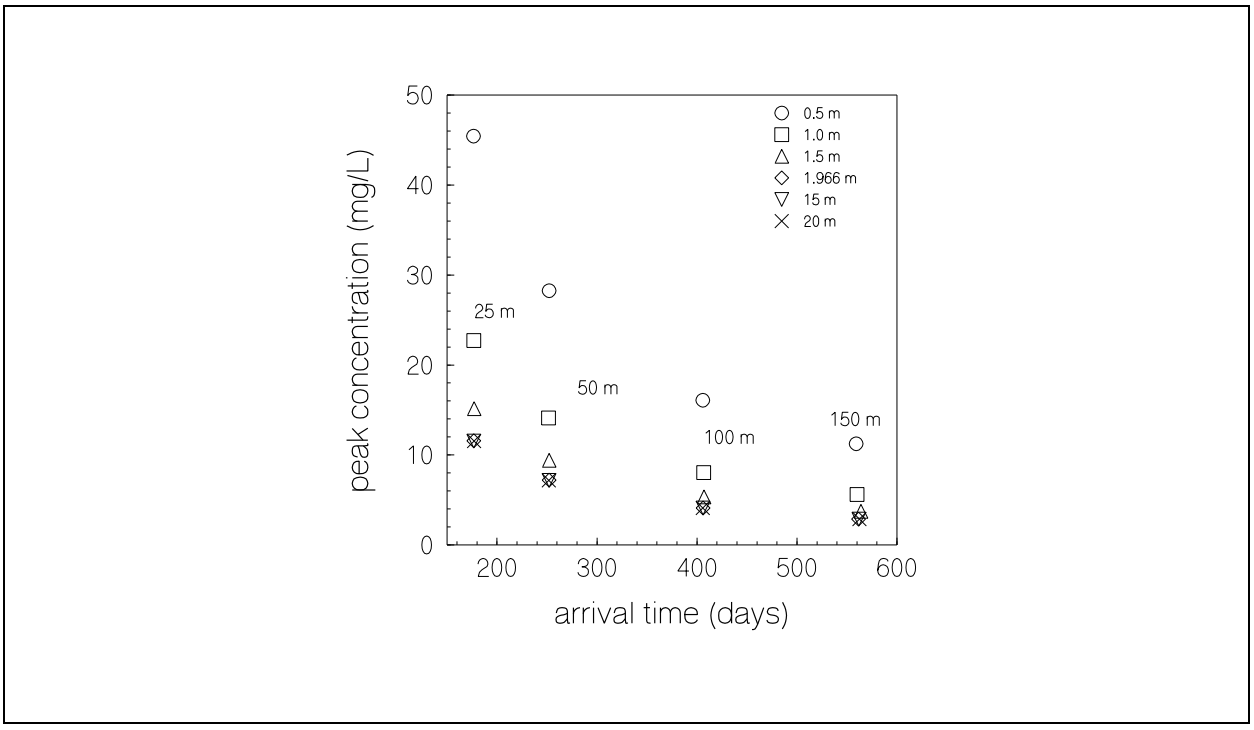

Figure 37 Peak concentration vs arrival time for variation of the aquifer thickness

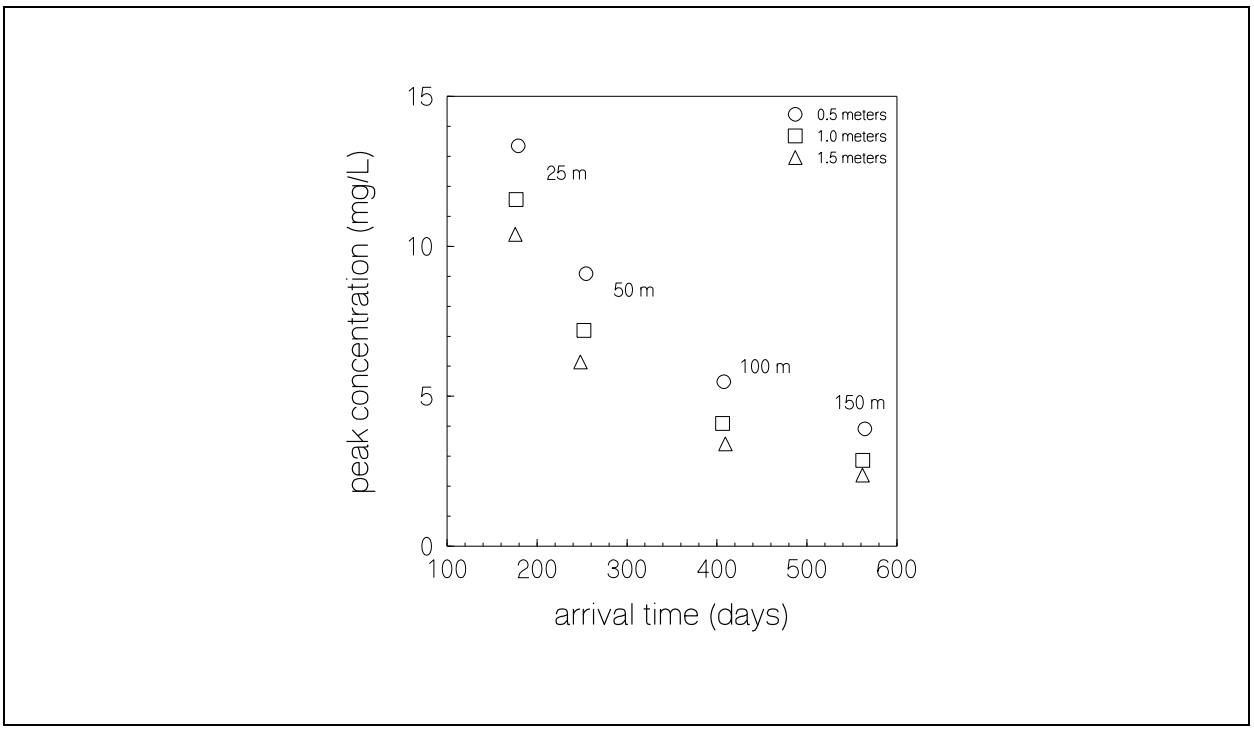

Figure 38 Peak concentration vs arrival time for variation of transverse dispersivity

5.8 HSSM Response: Decreasing Peak Concentration , Decreasing Arrival Time

Table 8 Parameters with Type F Response (Decreasing Peak Concentration, Decreasing Time-to-Peak)

Parameter	Type at 150 m	Minimum Value	Value used in Problem 2 Section 5.2 of Volume 1	Maximum Value	Module Impacts (K,O,T)
Ratio of horizontal to vertical conductivity, RKS	→G	1.0	2.5	10.0	NYY
Gradient	→G	0.005	0.01	0.02	NYY
Transverse vertical dispersivity, a_V	F	0.05 m	0.1 m	0.15 m	NYY
Dispersivities	F	a_L = 5 m a_T = 0.5 m a_V = 0.05 m	a_L = 10 m a_T = 1.0 m a_V = 0.1 m	a_L = 15 m a_T = 1.5 m a_V = 0.15 m	NNY NNY NYY
% maximum radius	F	25 %	49.15 %	100%	NNY
Constituent half-life	F	247.5 d	Infinite	Infinite	NNY

5.8.1 Ratio of horizontal to vertical conductivity

In HSSM, the ratio of the conductivities is used to specify the horizontal conductivity of the aquifer. Thus when the ratio of horizontal to vertical conductivity, RKS, increases, the horizontal hydraulic conductivity increases while the vertical conductivity remains unchanged. KOPT uses only the vertical conductivity, while OILENS and TSGPLUME both use only the horizontal conductivity. Thus the KOPT result, the NAPL arrival at the water table, is independent of horizontal conductivity variation. The extent of the lens, however, is greater with increasing horizontal conductivity, because with higher conductivity the lens can spread more readily. The peak mass flux, though, occurs at a lower radius, leading to a decline in the magnitude of the peak mass flux and the effective concentration. The result is earlier peak arrival times due to the increased conductivity of the aquifer and lower peak concentrations (**Figure 39**) due to the variation in the mass flux distribution.

5.8.2 Gradient

Along with the hydraulic conductivity, the hydraulic gradient in the aquifer determines the ground water velocity. The effect of increasing the gradient is similar to that of increasing the horizontal hydraulic conductivity (**Figure 39**), except that in HSSM the gradient does not affect the NAPL lens. Obviously, the NAPL arrival time at the water table is independent of the ground water gradient. The lens radius at peak mass flux to the aquifer and the effective source concentration decrease with the gradient, so the receptor concentrations decline with increasing gradient (**Figure 40**).

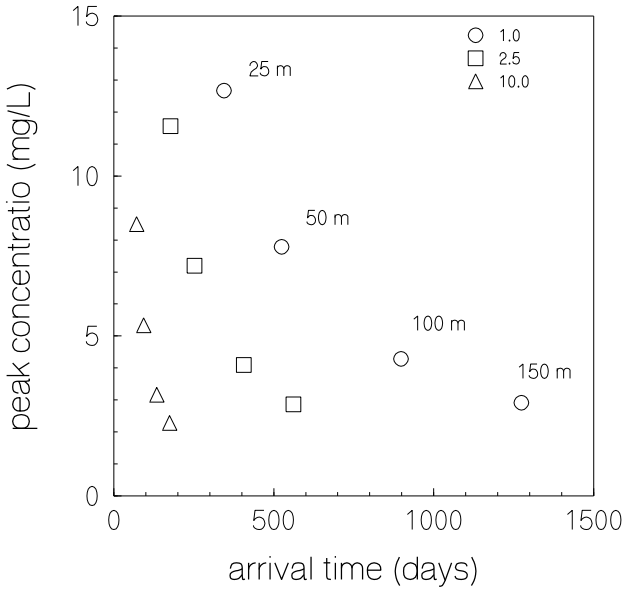

Figure 39 Peak concentration vs arrival time for variation of the ratio of horizontal to vertical conductivity

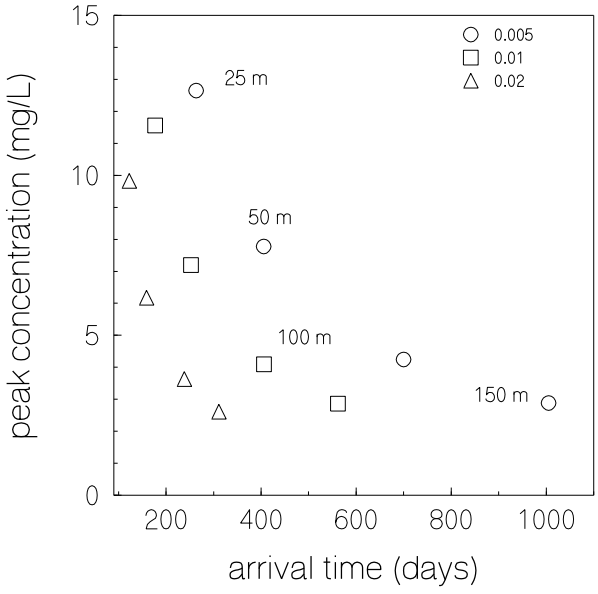

Figure 40 Peak concentration vs arrival time for variation of the hydraulic gradient

5.8.3 Transverse vertical dispersivity

The vertical dispersivity, $a_{V,}$ is used in calculating the mass flux to the aquifer (equation **(58)**), the effective source concentration (equation 76), and the penetration thickness (equation **(70)**). Increasing the vertical dispersivity causes both the mass flux and the penetration thickness to increase. These are competing effects as the first tends to increase peak receptor concentrations and the second to decrease them. Here the net effect is a decrease in the effective source and receptor concentrations (Figure 41). The peak arrival times at the receptors decrease, because the increased mass flux causes the chemical to be leached more rapidly with high a_v than with low.

5.8.4 All dispersivity

When all the dispersivities vary, with the proportions of longitudinal to horizontal transverse and longitudinal to vertical transverse dispersivity maintained at rations of 10:1 and 100:1, respectively; the peak receptor concentrations and their arrival times vary as shown in **Figure 42**. Of the effects of dispersivity seen so far (horizontal transverse, Figure 38; and vertical transverse, Figure 41), the behavior of variation in all the dispersivities is similar to that of the vertical transverse (Figure 41). The pattern also matches the longitudinal dispersivity effect at the more distant receptors (Figure 49). So the combined effects of proportional variation in all of the dispersivities is to decrease both the arrival time and the peak concentration at the receptor points. **Figure 43** shows the horizontal spread of the concentrations at the 50 m receptor (illustrated in Figure 23). These profiles were drawn at the arrival time for each set of dispersivities. With the high-valued set of dispersivities, the center line concentration is reduced because the mass has been transported outward from the centerline of the plume. The opposite effect is evident for the low-value set of dispersivities. This behavior matches that shown on **Figure 42**.

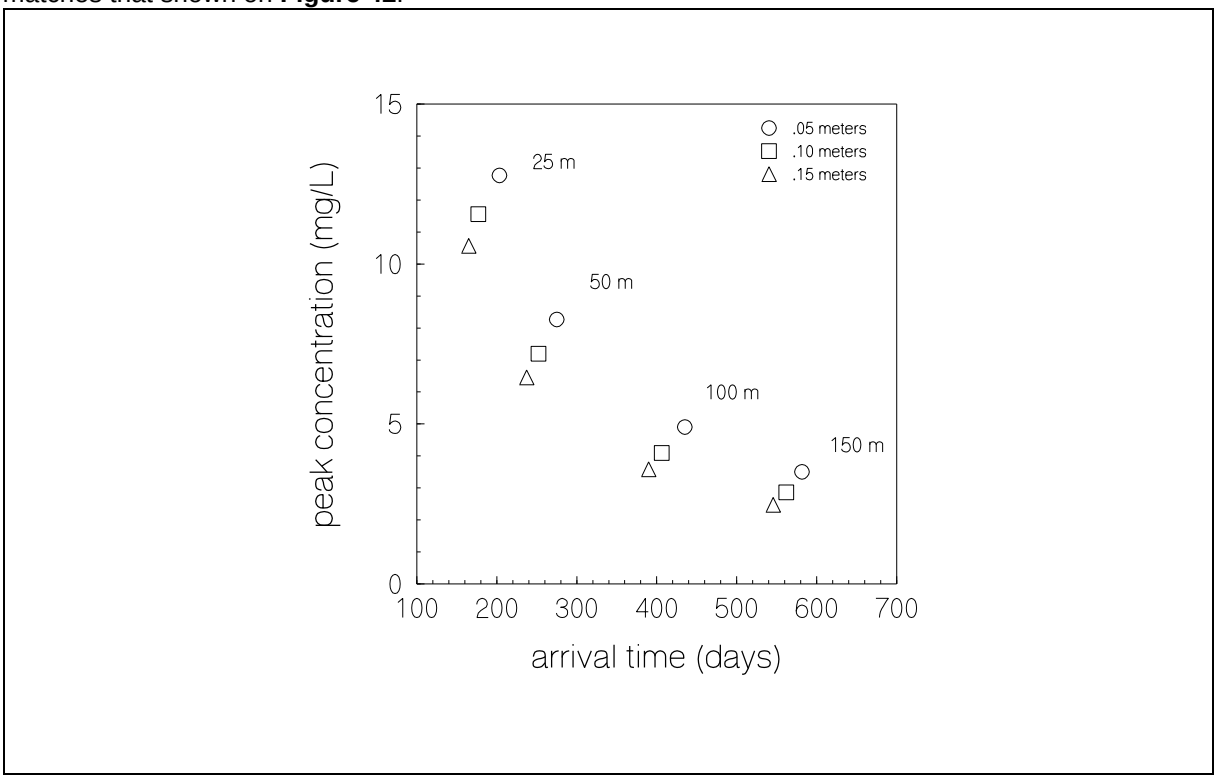

Figure 41 Peak concentration vs arrival time for variation of transverse vertical dispersivity

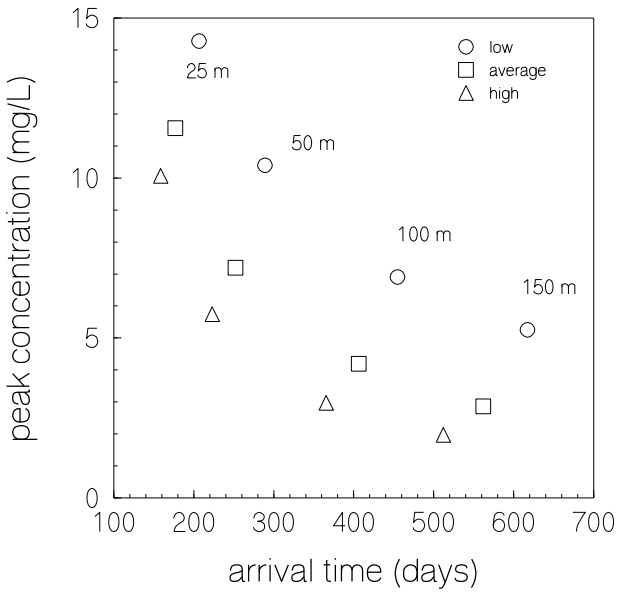

Figure 42 Peak concentration vs arrival time for variation of all dispersivities (longitudinal, transverse and vertical)

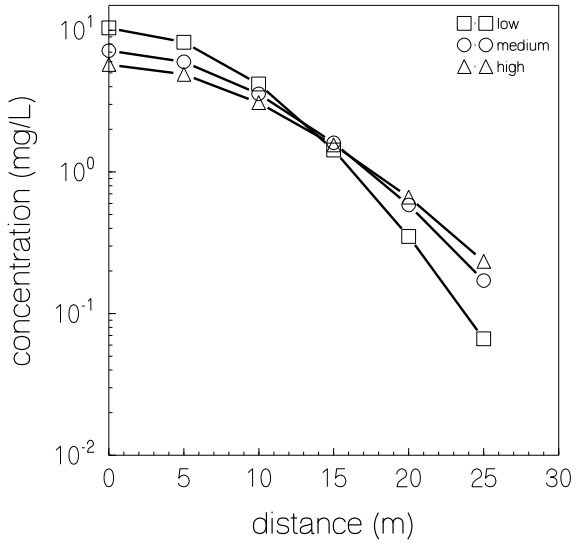

Figure 43 Transverse horizontal concentration profiles for the variation in all the dispersivities

5.8.5 Percent maximum radius

The TSGPLUME input parameter, percent maximum radius, is described in Volume 1 of the Users Guide (page 49 or 142). The recommended value of the parameter is 101. When this value is used, the radius of the boundary condition used in TSGPLUME is taken as the radius that occurs when the mass flux to the aquifer is a maximum. For the base scenario (**X2BT.DAT**), this occurs at a radius of 8.41 m which is 49.15% of the maximum radius. The effect of using other values for the radius is shown in **Figure 44**. The peak concentrations are inversely related to this parameter value, because when the peak mass flux is introduced into a smaller area, the resulting concentrations increase. The arrival times decrease with increasing percent maximum radius used in TSGPLUME, because in any case the initial input of mass to the aquifer occurs at a smaller radius than is used in TSGPLUME. This mass would have to travel all the way to the receptor, but the travel distance is shortened by selecting a radius (which is necessary). The larger the radius that is selected, the greater the shortening of the travel distance and hence the earlier arrival time.

5.8.6 Constituent half-life

The half-life of the constituent affects only the TSGPLUME results. With increasing decay rate, the peak concentrations decline due to loss of the constituent. At the near receptors, the arrival time remains nearly constant. Further away, the arrival time for the peak concentration decreases. The rising limb of the concentration history is truncated before the no-degradation peak is reached, as shown in **Figure 46**. In that figure, the concentration histories for the 100 m receptor begin at the same point in time and begin to rise at the same rate. For the smaller half-life (greater degradation rate) curve, the concentrations increase to a lower peak that occurs slightly earlier, because more time is required for the concentration to reach its maximum in the lower degradation rate case.

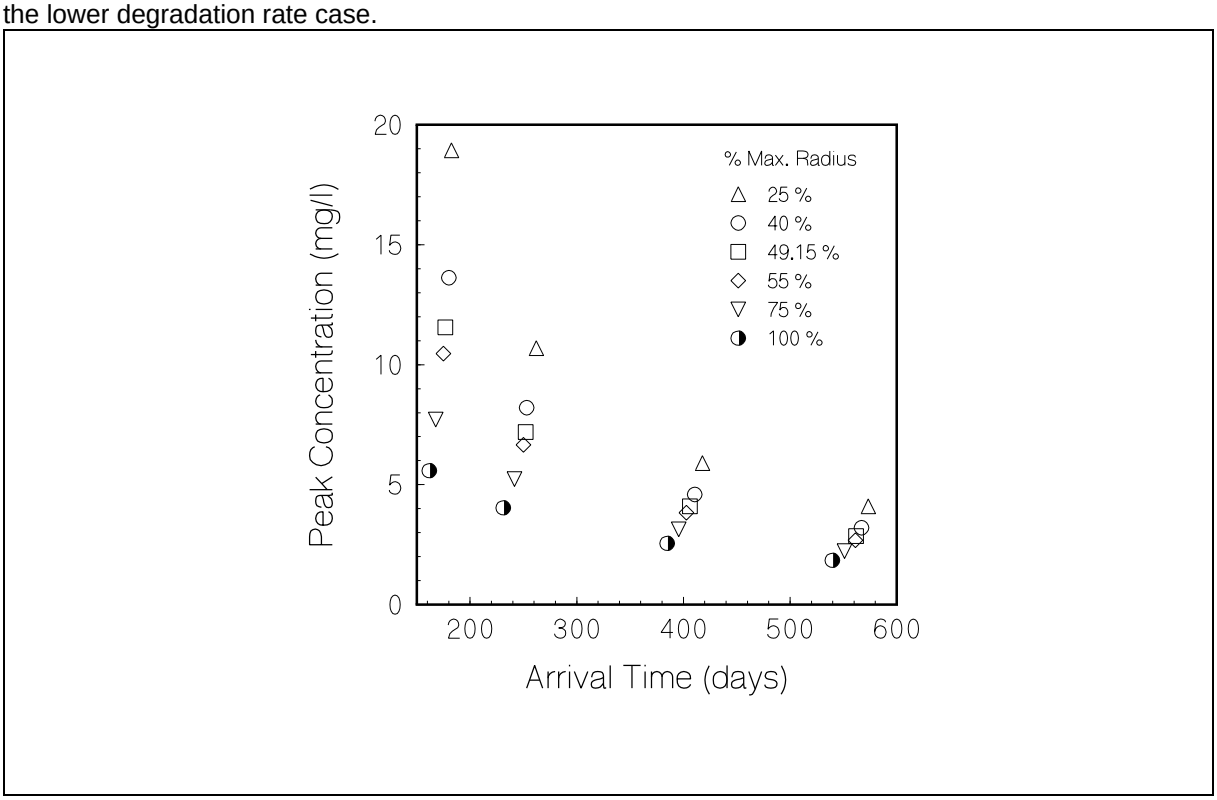

Figure 44 Peak concentration vs arrival time for variation of the percent maximum radius

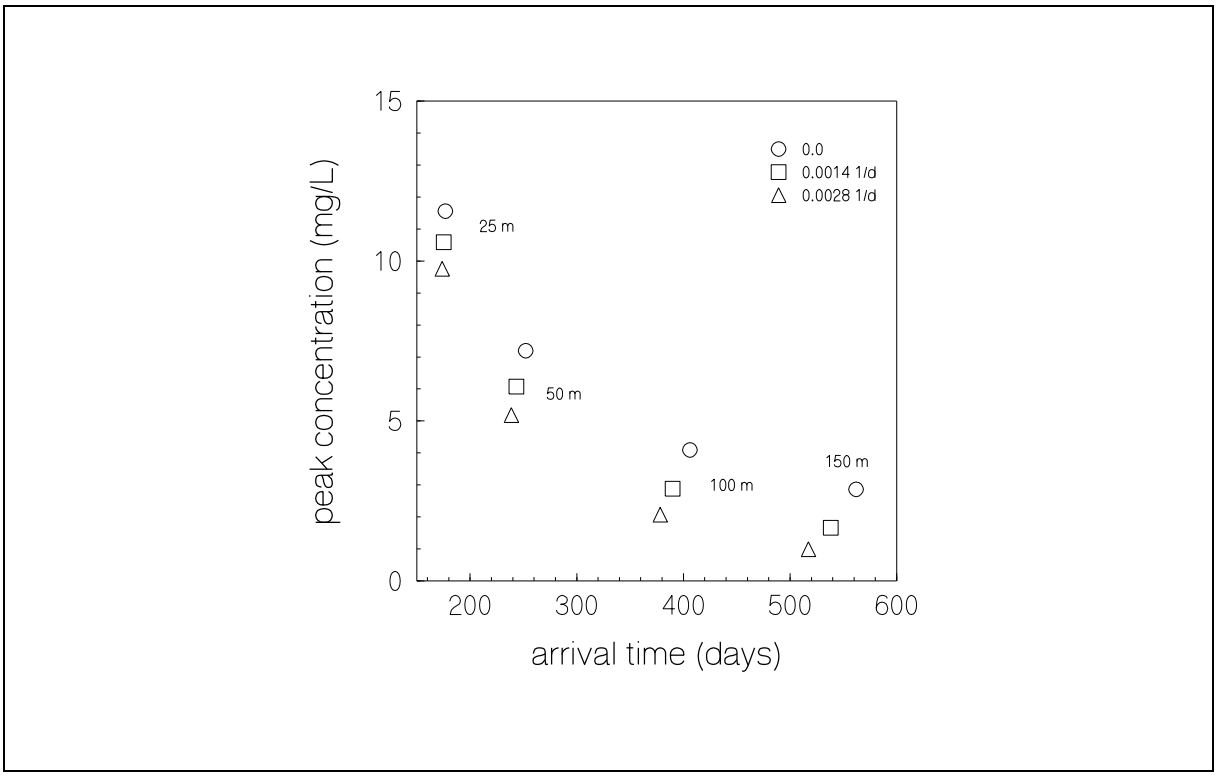

Figure 45 Peak concentration vs arrival time for variation of the constituent decay rate

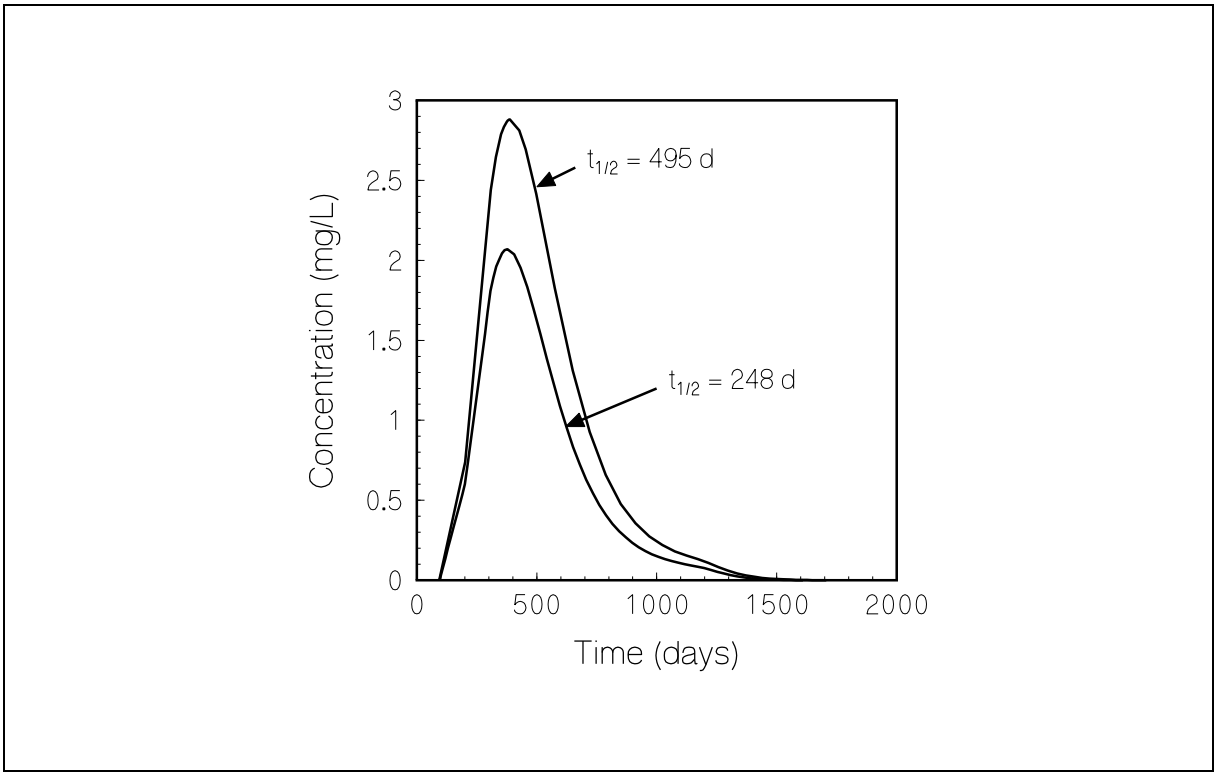

Figure 46 Concentration history for a degrading constituent

5.9 HSSM Response: Constant Peak Concentration, Decreasing Arrival Time

5.9.1 Saturated vertical conductivity

Increasing the saturated vertical conductivity, K_s, also increases the horizontal conductivity, because in HSSM the horizontal conductivity is taken as RKS *x* K_s. As expected, the NAPL arrives sooner at the water table with increasing conductivity. More rapid flow in the ground water causes the time for the peak mass flux to decrease with increasing horizontal conductivity. Lastly, the increased conductivity causes the constituent to be more rapidly transported within the aquifer. These three effects are reflected in the TSGPLUME results which show the arrival time of the peak receptor concentration decreasing with increasing conductivity (**Figure 47**). The peak concentrations remain very similar as the source zone concentrations remain relatively similar (Table 12) despite the variation in conductivity.

Table 9 Parameters with Type G Response (Constant Peak Concentration, Decreasing Arrival Time)

Parameter	Type at 150 m	Minimum Value	Value used in Problem 2 Section 5.2 of Volume 1	Maximum Value	Module Impacts (K,O,T)
Saturated vertical conductivity, K_s	G	1.75 m/d	7.1 m/d	28.4 m/d	YYY

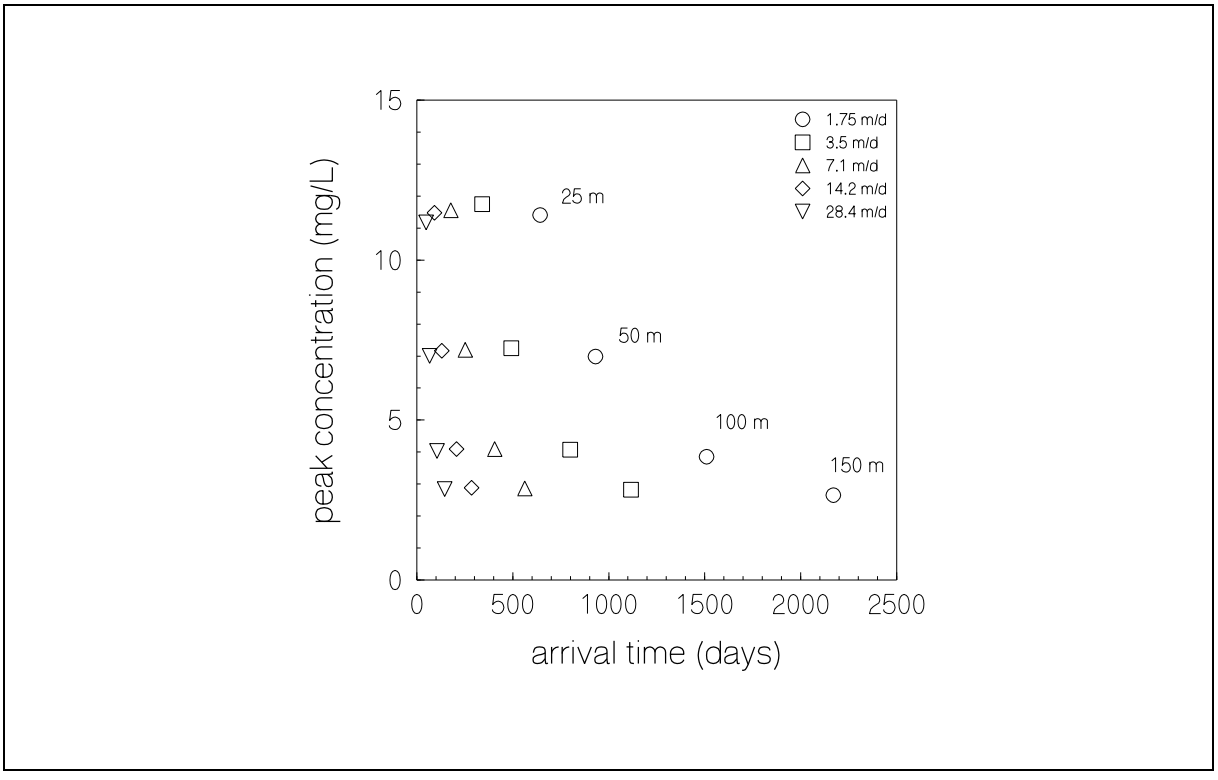

Figure 47 Peak concentration vs arrival time for variation of the saturated hydraulic conductivity

5.10 HSSM Response: Increasing Peak Concentration, Decreasing Arrival Time

Table 10 Parameters with Type H Response (Increasing Peak Concentration, Decreasing Arrival Time)

Parameter	Type at 150 m	Minimum Value	Value used in Problem 2, Section 5.2 of Volume 1	Maximum Value	Module Impacts (K,O,T)
Recharge, q_{wi}	→G	0.0 in/yr	20 in/yr	30 in/yr	YYY
Longitudinal dispersivity, a_L	D	5 m	10 m	30 m	NNY
Residual water saturation, S_{wr}	I	0.05	0.10	0.15	YYI
van Genuchten's n	H	2.0 m^{-1}	2.68 m^{-1}	3.4 m^{-1}	YYI
Source flux, q_o	H	0.3392 m/d	0.4255 m/d	0.9044 m/d	YII
Source duration	H	0.75 d	1.0 d	2.0 d	YII
NAPL Saturation in the Lens	G	0.2000	0.3236	0.4500	NYI

5.10.1 Recharge

The effect of increasing recharge is to increase one component of mass flux to the aquifer (equation **(48)**). With higher recharge the NAPL reaches the water table sooner (Table 12) and the time to the peak mass flux decreases, because the increased mass flux causes the constituent to more rapidly leave the lens. The peak receptor concentrations are higher because of the higher flux to the aquifer throughout the leaching period (**Figure 48**).

5.10.2 Longitudinal dispersivity

The longitudinal dispersivity only affects the aquifer module of HSSM. The NAPL arrival time at the water table and the mass input to the aquifer are all unaffected by this parameter. Thus the results shown in **Figure 40**, **Figure 49** depend only on transport in the aquifer. For all of the receptors, increasing the dispersivity decreases the arrival time. This is caused by the positive contribution of dispersion to the mass flux. Although increased dispersivity is associated with increased apparent dilution of concentration, at the nearest receptors the peak concentrations increase with increasing dispersivity. At the most distant receptors, the peak concentrations decline with increasing longitudinal dispersivity. This example contrasts with that of the horizontal transverse dispersivity where there is increased lateral movement of the constituent. Here, however, the lateral transport remains the same for each value of longitudinal dispersivity and so the width of the plume is constrained. The only effects of dispersion that vary in this example are longitudinal. Apparently there is a trade off between the more rapid arrival time and dilution of the concentrations, which always causes the peak to occur sooner with increasing longitudinal dispersivity. At the near receptors, the more rapid arrival time dominates over the increased dilution effect, which is dominant further downgradient.

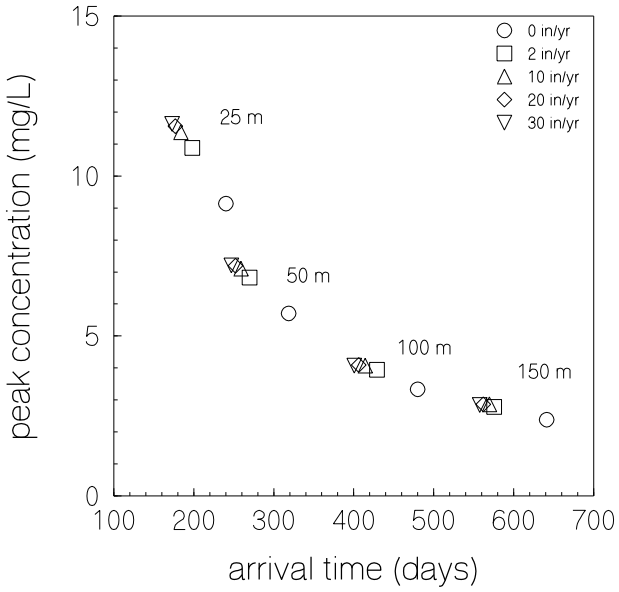

Figure 48 Peak concentration vs arrival time for variation of recharge

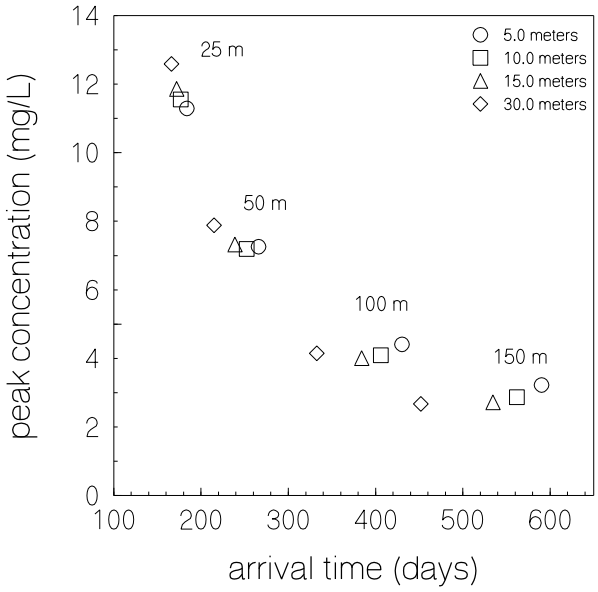

Figure 49 Peak concentration vs arrival time for variation of the longitudinal dispersivity

5.10.3 Residual water saturation

The residual water saturation affects the vadose zone modules (KOPT and OILENS). Variation of the residual water saturation, S_{wr}, has a small effect on the receptor concentrations (**Figure 50**), and this parameter could reasonably be listed in group "I". Increasing the residual water saturation casues the NAPL to arrive at the water table somewhat more rapildy. The peak mass flux to the aquifer occurs more rapildly as S_{wr} increases and the peak mass flux and source concentrations increase. This parameter has no direct effect on the aquifer model, so the impacts are limited to KOPT and OILENS.

5.10.4 van Genuchten's n

van Genuchten's n is a parameter of the capillary pressure curve that describes the steepness of the curve. In **Figure 51** the values range from 2.0 to 4.5; the latter is the value from **X2BT.DAT**. Generally, high values of n indicate uniformity in the pore size distribution. For such media, the effective conductivity to the NAPL stays relatively higher as the NAPL saturation decreases. In the vadose zone, the result is that the NAPL moves more nearly as a pulse if n is high. In HSSM, as the value of n increases, the NAPL arrival time at the water table decreases, as does the time for the peak mass flux. In these cases, the NAPL arrives at the water table in a relatively short pulse, that generates sufficient head to drive the lens radially. Thus the lens radius at peak mass flux increases with n; giving high peak mass fluxes and source concentrations.

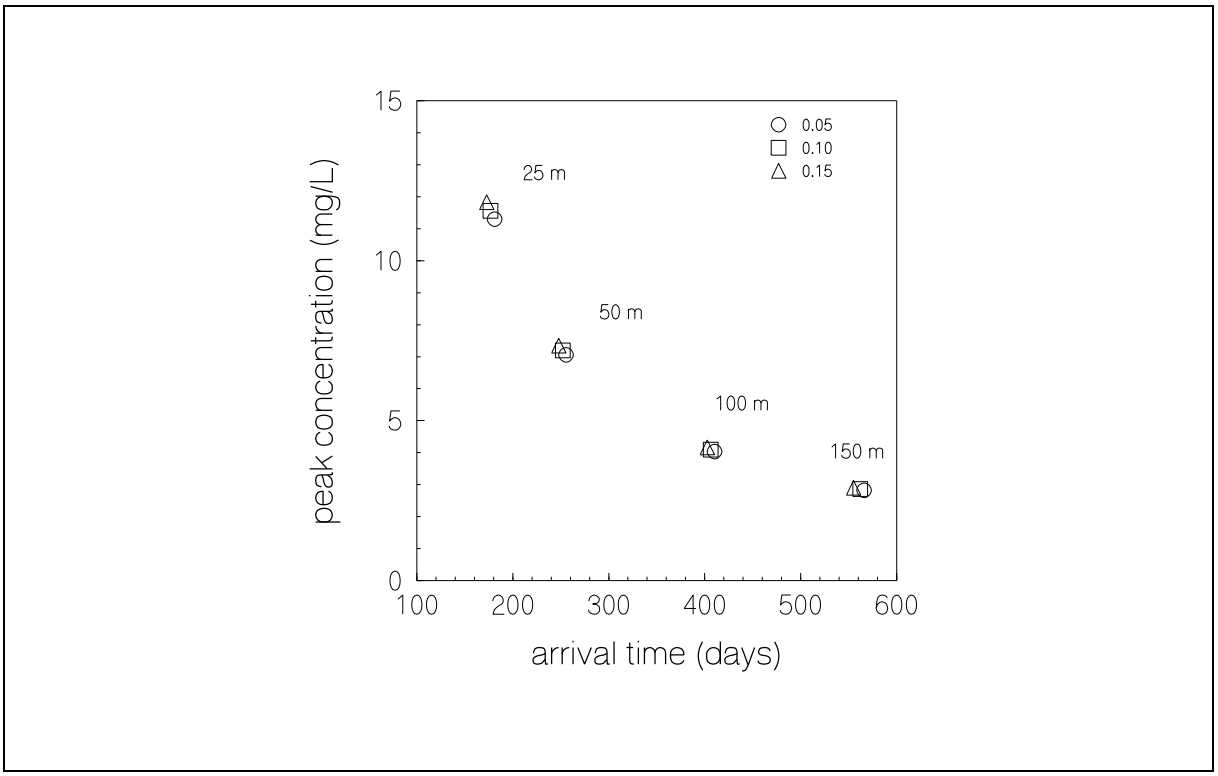

Figure 50 Peak concentration vs arrival time for variation of the residual water saturation

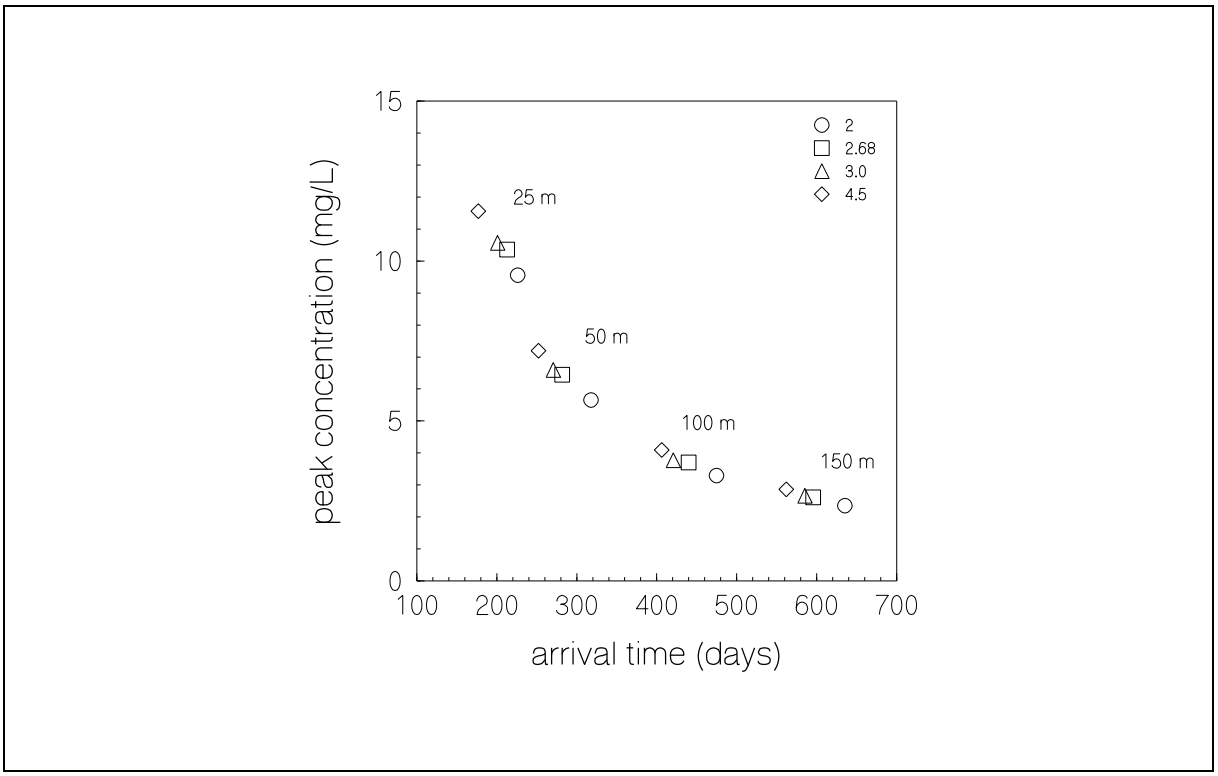

Figure 51 Peak concentration vs arrival time for variation of van Genuchten's n

5.10.5 Source flux

The source flux and duration determine the amount of contaminant mass introduced into the subsurface. With higher flux, the NAPL arrives at the water table sooner (Table 12), and the lenses that form are larger because of the increased mass of NAPL present. Bigger lenses tend to generate higher peak mass fluxes because of their large radii. In this case the larger peak mass flux corresponds to higher peak source concentrations. The result is that the receptor arrival times decrease and the peak concentrations increase with increasing source flux (**Figure 52**).

5.10.6 NAPL saturation in the lens

The NAPL saturation in the lens, S_{omax}, is an integrated measure of the vertical distribution of NAPL through the lens. Each HSSM input data set has an appropriate value of S_{omax} that can be determined by using the NTHICK or NTHICK2 utilities. The **X2BT.DAT** data set contains the correct value of 0.3236; the two other values used here were selected to illustrate the sensitivity of the results to this parameter. Generally, lenses with higher NAPL saturations are smaller because more NAPL is contained in a unit volume of the lens. The time to the peak mass flux decreases with S_{omax}; the radius increases and the peak mass flux increases. The reduction in time to peak mass flux occurs because higher NAPL saturations give higher NAPL lens effective conductivities and thus lenses which develop quicker. Hence the peak mass flux and transport to the receptor occur more rapidly (**Figure 53**).

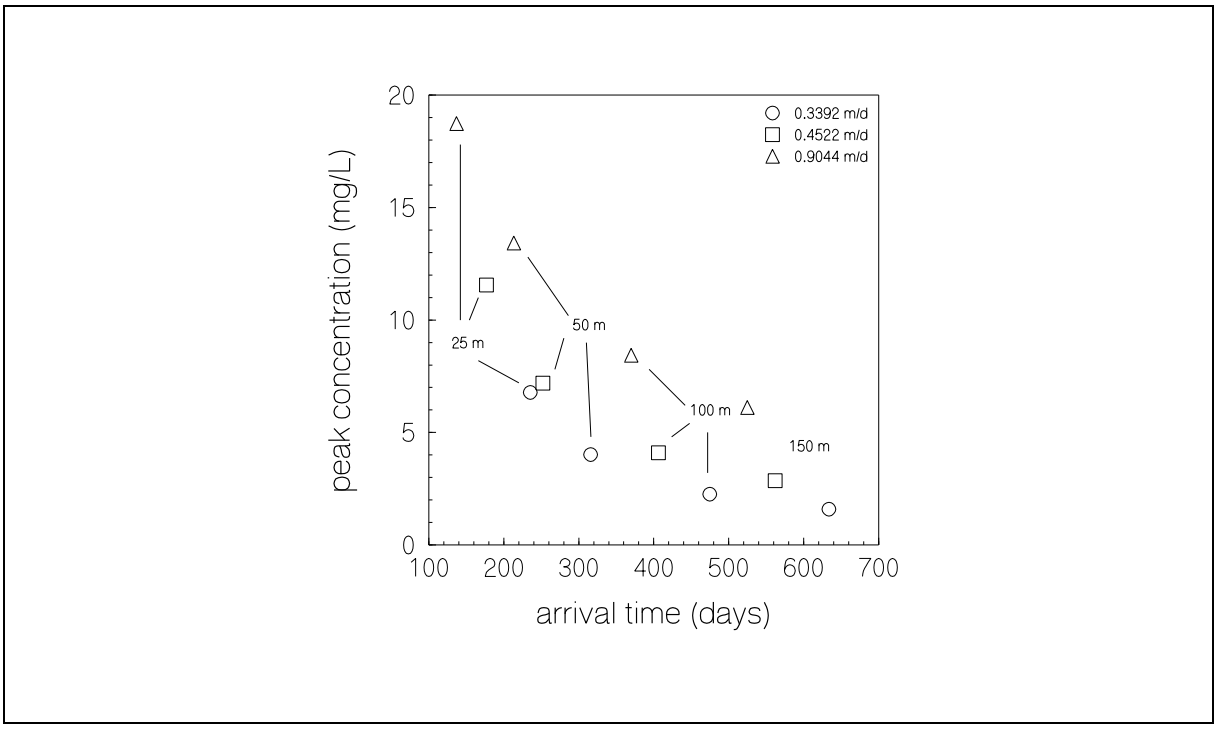

Figure 52 Peak concentration vs arrival time for variation of source flux

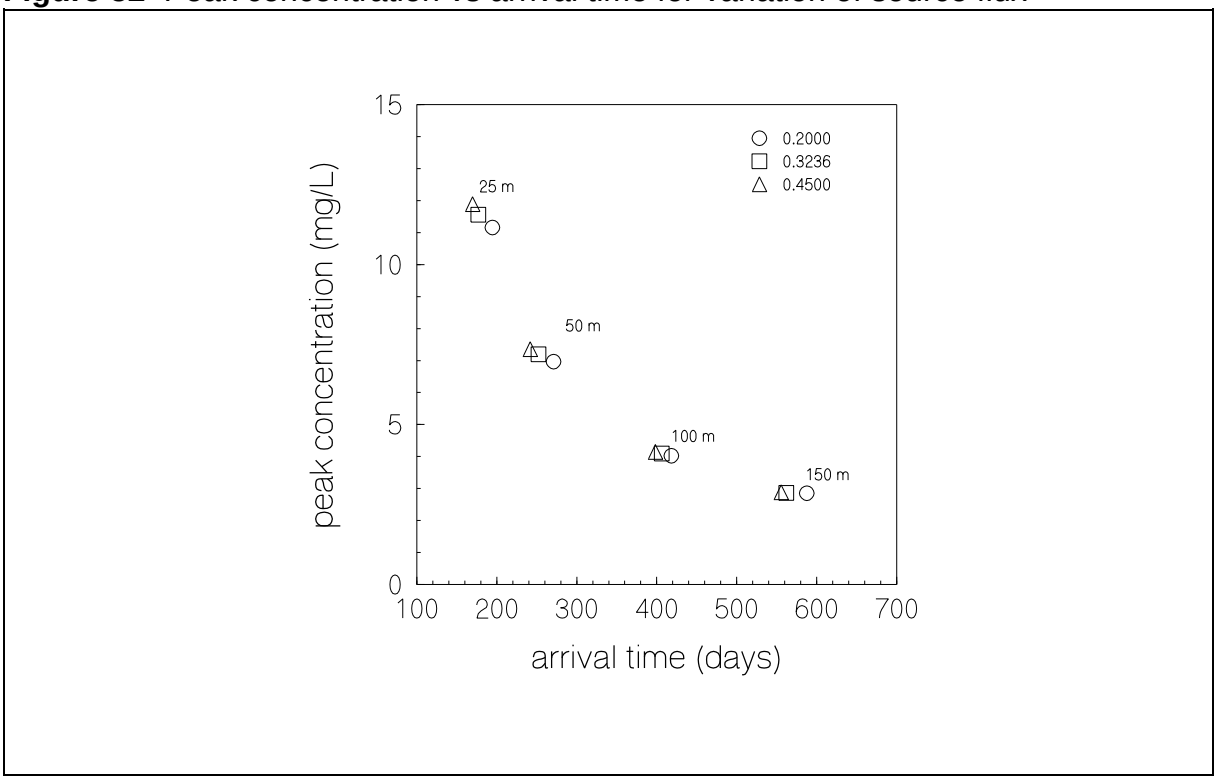

Figure 53 Peak concentration vs arrival time for variation of the NAPL saturation in the lens

5.11 HSSM Response: Constant Peak Concentration, Constant Arrival Time

Table 11 Parameters with Type I Response (Constant Peak Concentration, Constant Arrival Time)

Parameter	Type at 150 m	Minimum Value	Value used in Problem 2 Section 5.2 of Volume 1	Maximum Value	Module Impacts (K,O,T)
Aquifer thickness (greater than the penetration thickness)	I	1.966	15.0	20.0	NNY
van Genuchten's α	I	2.0	4.5	4.5	YYI
Water surface tension, σ_{aw}	I	58 dyne/cm	65 dyne/cm	72 dyne/cm	YYI
Max. water phase relative permeability during infiltration, k_{rwmax}	I	0.4	0.5	0.6	YII
NAPL surface tension, σ_{ao}	I	25 dyne/cm	35 dyne/cm	45 dyne/cm	YYI
NAPL density, ρ_o	I	0.64 g/cm^3	0.72 g/cm^3	0.80 g/cm^3	YYI
Aquifer residual NAPL saturation, S_{ors}	I	0.075	0.15	0.50	NYI
NAPL/water interfacial tension	I	30 dyne/cm	45 dyne/cm	60 dyne/cm	NYI
Capillary thickness parameter	→C	0.001 m	0.01 m	0.02 m	NYI

5.11.1 van Genuchten's α

van Genuchten's α is one of the parameters that describes the capillary pressure curve. The values used here are: 2.0 m^{-1}, 2.68 m^{-1}, and 3.4 m^{-1}. These values correspond to entry pressures of 37 cm, 28 cm and 22 cm respectively. These are the heights for fully saturated capillary rise of water. **Figure 54** shows that this range of values has a negligible impact on the receptor concentrations.

5.11.2 Water surface tension

The surface tension of water is used by HSSM only in the KOPT module and for determining the NAPL saturation in the lens by NTHICK. **Figure 55** shows its minimal impact on the receptors.

5.11.3 Maximum water phase relative permeability during infiltration

During infiltration, a certain amount of the pore space is occupied by trapped air. The trapped air saturation is included in HSSM by setting a maximum water phase relative permeability during infiltration k_{rmax}. This parameter has little impact on receptor concentration histories when the vadose zone is relatively permeable as in this example (**Figure 56**).

5.11.4 NAPL surface tension

The NAPL surface tension impacts vadose zone flow and transport and the NAPL saturation in the lens in much the same was as does the water surface tension. The NAPL surface tension has little impact on the receptor concentration histories (**Figure 57**).

5.11.5 NAPL density

The NAPL density affects vadose zone flow and the development of the lens. For LNAPLs, the typical range of density variation is limited; the maximum is 1.0 g/cm^3, and 0.50 g/cm^3 or 0.06 g/cm^3 would be the lower bound. As seen in **Figure 58**, this parameter has little impact on the receptors.

5.11.6 Aquifer residual NAPL saturation

In HSSM, the residual NAPL saturation in the aquifer, S_{ors}, is used in the development of the NAPL lens. Over the range used in the analysis, 0.075 to 0.50, there is essentially no impact on the receptor concentrations (**Figure 59**).

5.11.7 NAPL/water interfacial tension

The NAPL/water interfacial tension, σ_{ow}, is used only in the NTHICK utility. σ_{ow} plays a role in determining the maximum NAPL saturation in the lens, S_{omax}. By increasing σ_{ow} from 30 dyne/cm to 60 dyne/cm, there is negligible impact on the peak concentrations and arrival times (**Figure 60**). The range of σ_{ow} used in this example covers the expected range.

5.11.8 Capillary thickness parameter

The capillary thickness parameter is used to establish a smear zone. The values for the capillary thickness parameter (0.001 m, 0.01 m and 0.02 m) used in **Figure 61** represent nominal smear zone thicknesses of 0.64 cm, 6.5 cm, and 13 cm respectively. This range of variation causes little change in the receptor concentration history, in contrast to the larger smear zones used in **Figure 36**, **Figure 53**.

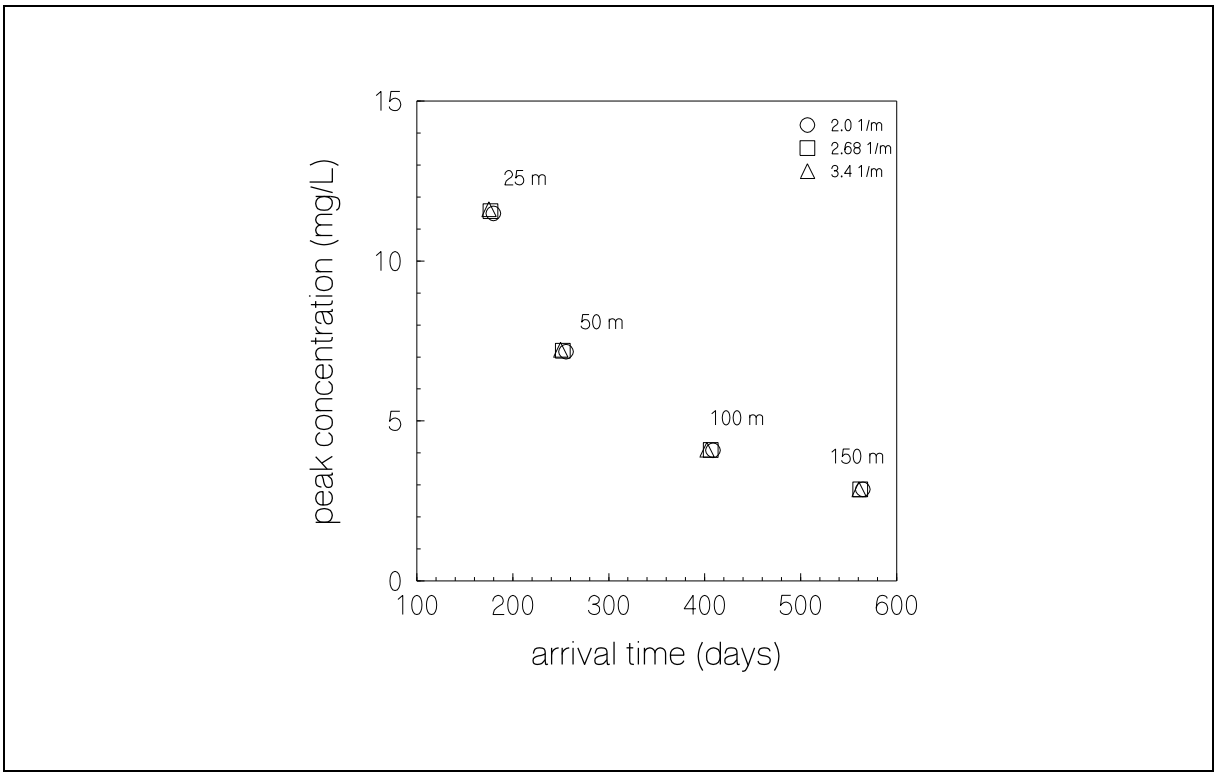

Figure 54 Peak concentration vs arrival time for variation of van Genuchten's α

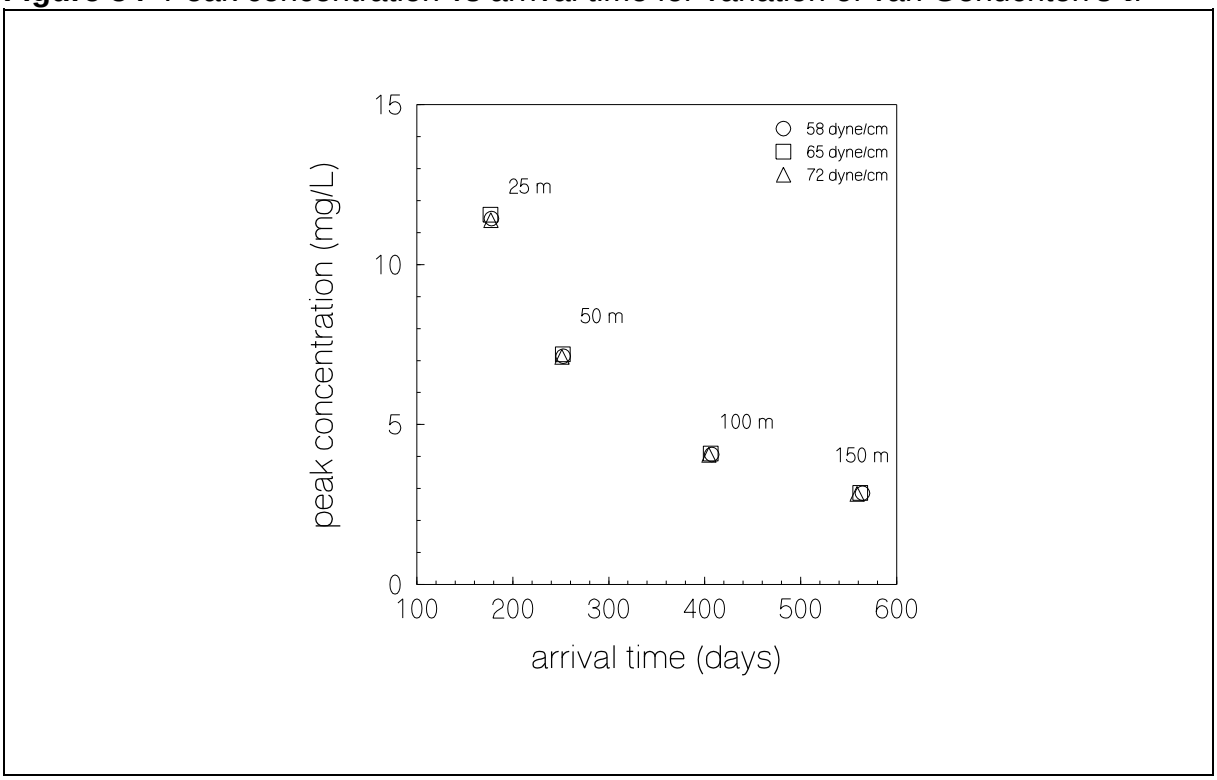

Figure 55 Peak concentration vs arrival time for variation of the water surface tension

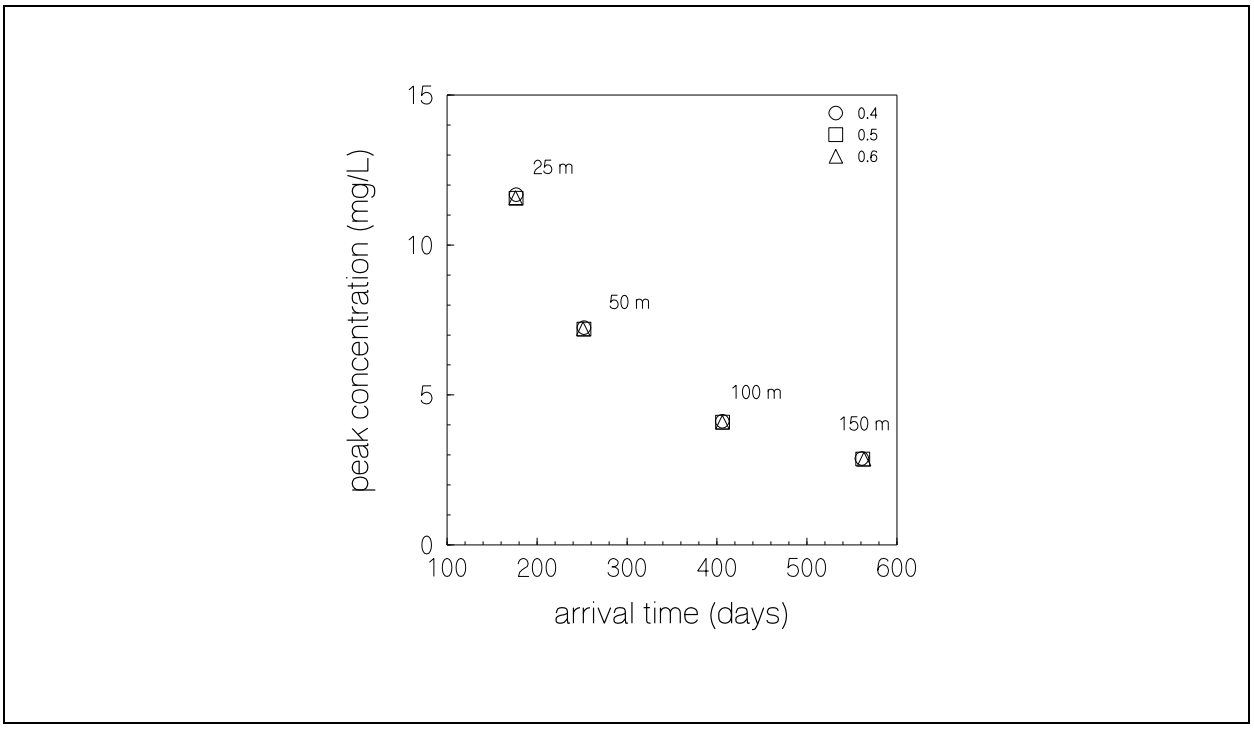

Figure 56 Peak concentration vs arrival time for variation of the maximum water phase relative permeability during infiltration

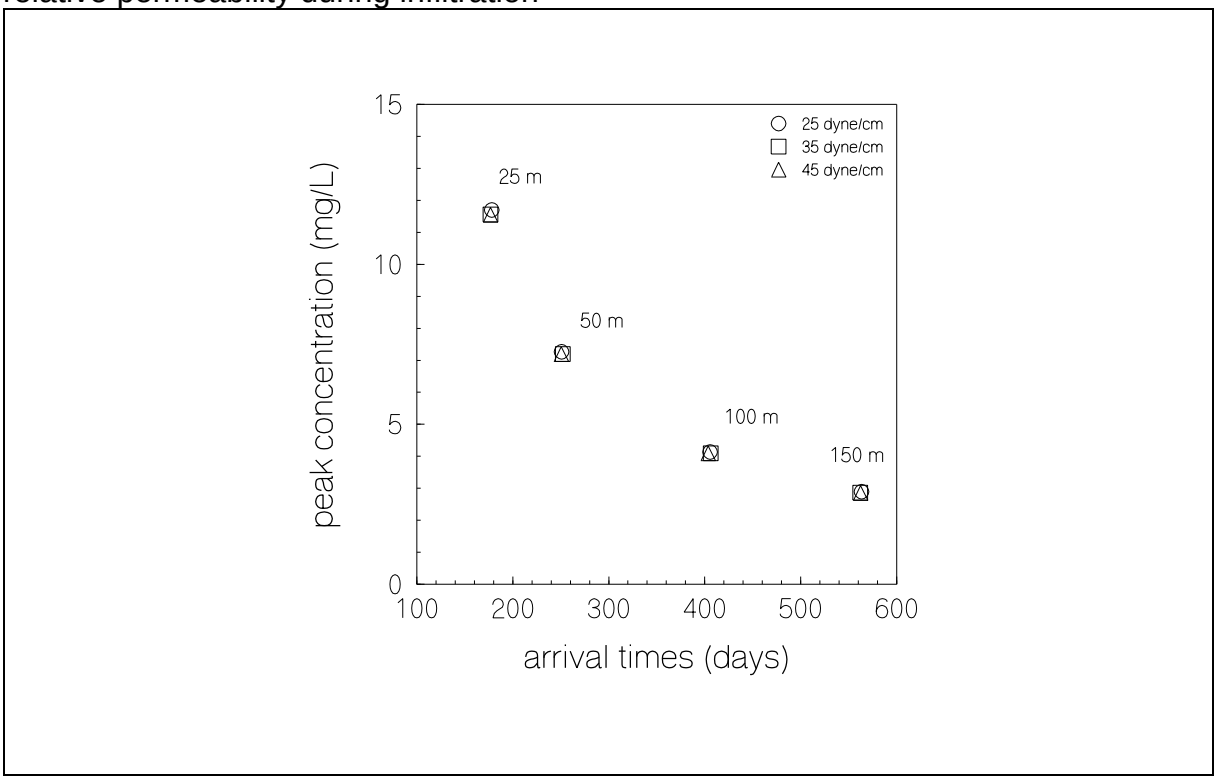

Figure 57 Peak concentration vs arrival time for variation of the NAPL surface tension

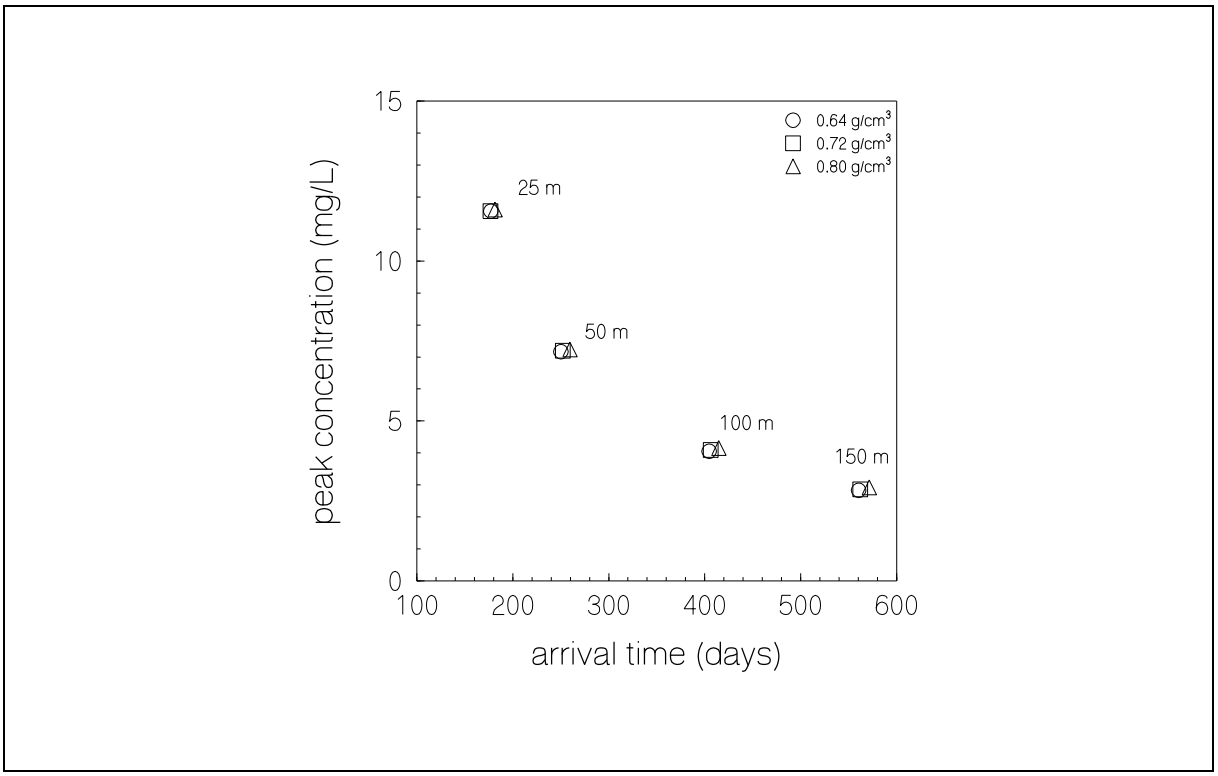

Figure 58 Peak concentration vs arrival time for variation of the NAPL density

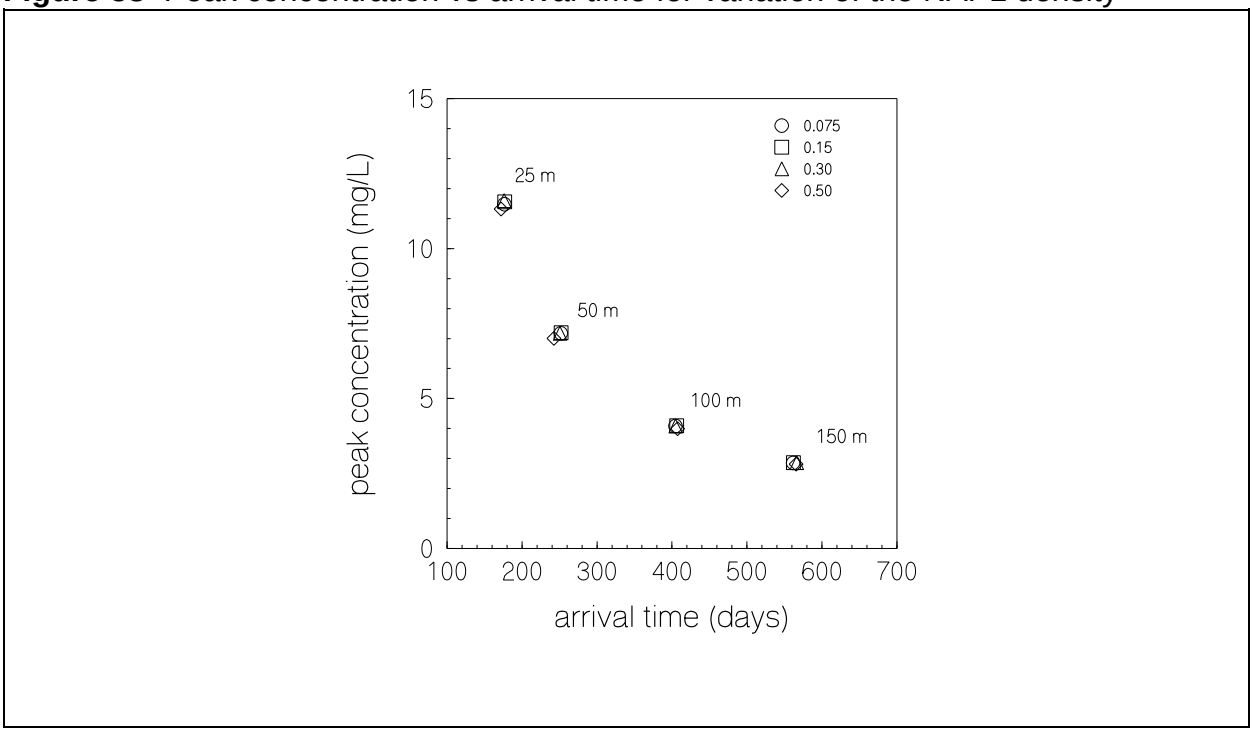

Figure 59 Peak concentration vs arrival time for variation of the aquifer residual NAPL saturation

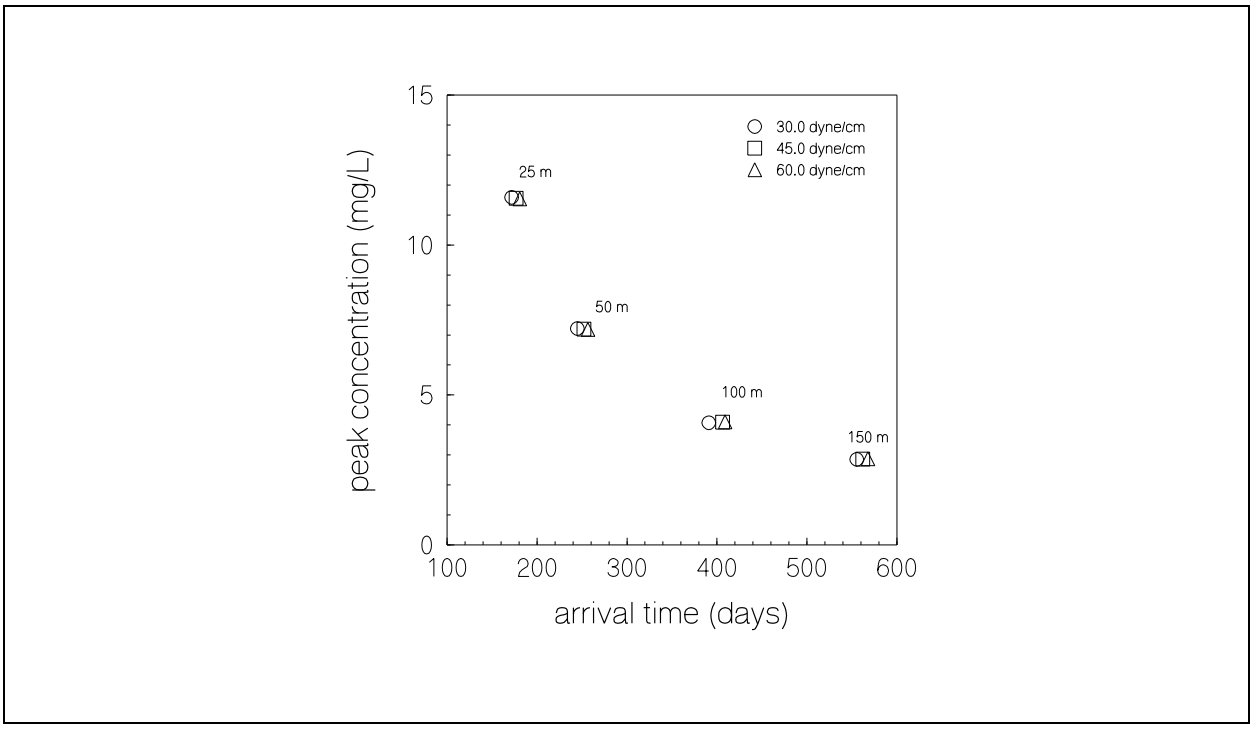

Figure 60 Peak concentration vs arrival time for variation of the NAPL/water interfacial tension

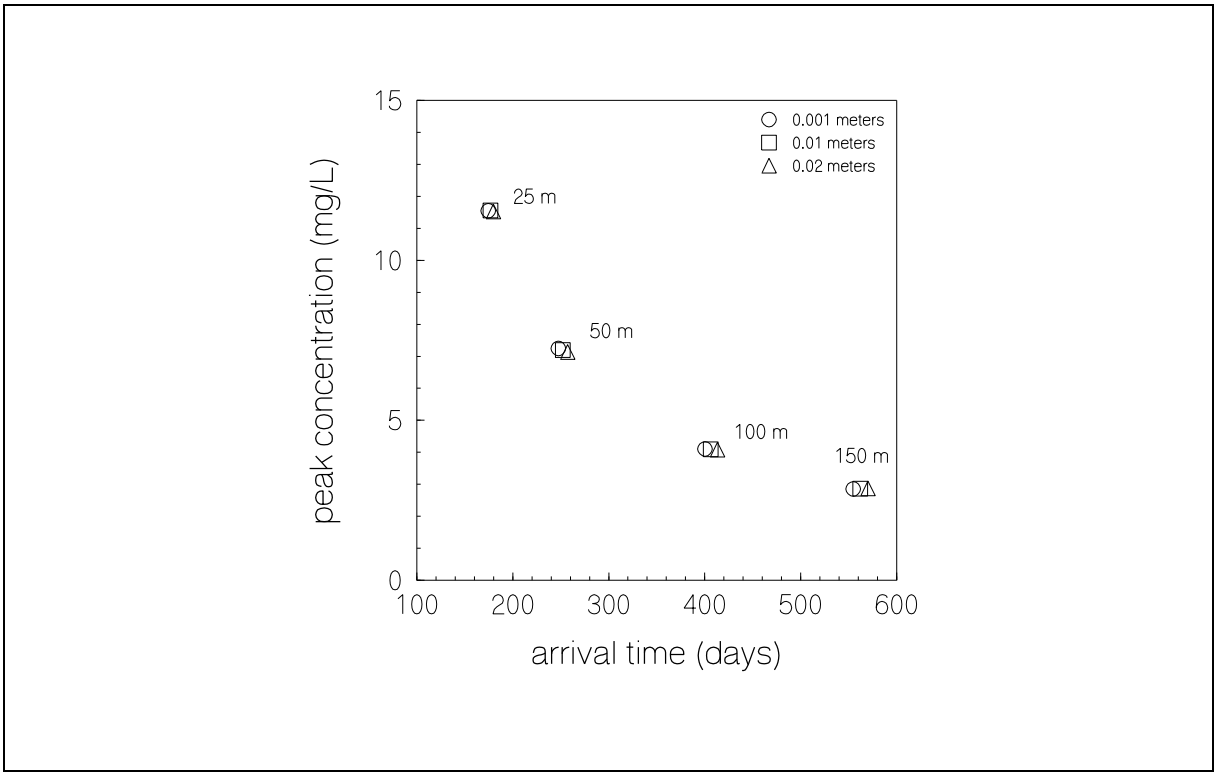

Figure 61 Peak concentration vs arrival time for variation of the capillary thickness parameter

Section 6 Discussion

The HSSM model is a screening model for exposure assessment from spills or other releases of LNAPLs to the subsurface. The model decomposes the transport problem to three independent components which are simulated by three separate models. KOPT models the transport of a NAPL from the release location to the water table. This model incorporates the effects of capillarity and multiphase partitioning, though volatilization and degradation processes are neglected. KOPT provides the LNAPL and constituent flux to the water table. OILENS models the spreading of an LNAPL along the capillary fringe taking into account the time variable source strength and buoyancy. This model computes the size of the lens as a function of time and the constituent mass transfer to the water table aquifer due to infiltration through the lens and groundwater flowing beneath the lens. Transport in the water table aquifer is modeled using TSGPLUME. This model uses the time variable source strength provided by OILENS to calculate downgradient concentrations at potential exposure locations. TSGPLUME estimates the depth of penetration of the contaminant into the aquifer and uses this depth in a two-dimensional model. The processes of advection, dispersion, sorption, and degradation (including dilution) are included in the model.

HSSM is a model which uses many approximations, and at the present time it is not possible to evaluate the adequacy of each of them. Nevertheless, it is hoped that the model has captured the essential behavior of the underlying processes. The model is computationally efficient and may be used to estimate the potential impacts of a large number of chemicals in an economical fashion.

References

Abriola, L. M. and G. F. Pinder, A multiphase approach to the modeling of porous media contamination by organic compounds, 1 Equation development, Water Resources Research, 21, 11-18, 1985a.

Abriola, L. M. and G. F. Pinder, A multiphase approach to the modeling of porous media contamination by organic compounds, 2 Numerical simulation, Water Resources Research, 21, 19-26, 1985b.

Allen, M.B., Numerical modelling of multiphase flow in porous media, Advances in Water Resources, 8, 162-187, 1985.

Bear, J., Dynamics of Fluids in Porous Media, American Elsevier, New York, 1972.

Bouwer, H., Rapid field measurements of air entry value and hydraulic conductivity of soil as significant parameter in flow system analysis, Water Resources Research, 2, 729-738, 1966.

Brooks, R. H. and A. T. Corey, Hydraulic Properties of Porous Media, Colorado State University Hydrology Paper No. 3, Ft. Collins, Colorado, 1964.

Buckley, S. E. and M. C. Leverett, Mechanism of fluid displacement in sands, Transactions American Institute of Mining Engineers, 146, 107-116, 1942.

Burdine, N. T., Relative permeability calculations from pore size distribution data, Transactions American Institute of Mining Engineers, 198, 71-78, 1953.

Carslaw, H.W. and J.C. Jaeger, Conduction of Heat in Solids, 2nd ed., Oxford Univ. Press (Clarendon), London, 1959.

Charbeneau, R. J., Kinematic models for soil moisture and solute transport, Water Resources Research, 20, 699-706, 1984.

Charbeneau, R. J. , Simulation of the transient soil water content profile for a homogeneous bare soil, Water Resources Research, 17, 1271-1279, 1991.

Charbeneau, R. J., J. W. Weaver, and V. J. Smith, Kinematic Modeling of Multiphase Solute Transport in the Vadose Zone, U.S. EPA/600/2-89/045, 1989.

Cline, P.V., J. J Delfino, and P. S. C. Rao, Partitioning of aromatic constituents into water from gasoline and other complex solvent mixtures, Environmental Science and Technology, 25, 914-920, 1991.

Corapcioglu, M. Y., and A. Baehr, A compositional multiphase model for groundwater contamination by petroleum products, I. Theoretical considerations, Water Resources Research, 23, 201-243, 1987

Corapcioglu, M. Y., and M. A. Hossain, Ground-water contamination by high-density immiscible hydrocarbon slugs in gravity-driven gravel aquifers, Ground Water, 28, 403-412, 1990.

Dougherty, E. L. and J. W. Sheldon, The use of fluid interfaces to predict the behavior of oil recovery process, Society of Petroleum Engineers Journal, 4, 171-182, 1964.

Dracos, T., Theoretical considerations and practical implications on the infiltration of hydrocarbons in aquifers, Proceedings International Symposium on Groundwater Pollution by Oil Hydrocarbons, International

Association of Hydrologists, Prague, June 5-9, 1978.

El-Kadi, A. I., Applicability of sharp-interface models for NAPL transport: 1. Infiltration, Ground Water, 30, 849-856, 1992.

Faust, C. R., Transport of immiscible fluids within and below the unsaturated zone--a numerical model, Water Resources Research, 21, 587-596, 1985.

Faust, C. R., J. H. Guswa, and J. W. Mercer, Simulation of three-dimensional flow of immiscible fluids within and below the unsaturated zone, Water Resources Research, 25(12), 2449-2464, 1989.

Fehlberg, E., Low-order Classical Runge-Kutta Formulas With Stepsize Control and Their Application to Some Heat Transfer Problems, NASA TR R-315, 1969.

Green, W. H. and G. A. Ampt, Studies on Soil Physics, Journal of Agricultural Science, 4, 1-24, 1911.

Helfferich, F. G., Theory of multicomponent, multiphase displacement in porous media, Society of Petroleum Engineers Journal, 21, 51-62, 1981.

Helfferich, F.G., Multicomponent wave propagation: Attainment of coherence from arbitrary starting conditions, Chemical Engineering Communications, 44, 275-285, 1986.

Hochmuth, D. P. and D. K. Sunada, Ground-water model of two-phase immiscible flow in coarse materials, Groundwater, 23, 617-626, 1985.

Hunt, J.R., N. Sitar and K.S. Udell, "Nonaqueous phase liquid transport and cleanup 1. Analysis of Mechanisms," Water Resour. Res., 24(8), 1247-1258, 1988.

Huyakorn, P.S., M.J. Ungs, L.A. Mulkey and E.A. Sudicky, "A three-dimensional analytical model for predicting leachate migration," Ground Water, 25(5), 588-598, 1982.

Kessler, A. and H. Rubin, Relationships between water infiltration and oil spill migration in sandy soils, Journal of Hydrology, 91, 187-204, 1987.

Kueper, B. H. and E. O. Frind, Two-phase flow in heterogeneous porous media, 1. Model development, Water Resources Research, 27(6), 1049-1057, 1991.

Leverett, M.C., Capillary behavior in porous solids, Transactions American Institute of Mining Engineers, 142, 152-169, 1941

Lighthill, M.J. and G.B. Whitham, On kinematic waves I. flood movement in long rivers, Proceedings of the Royal Society A, 229, 281-316, 1955.

Mull, R., The migration of oil-products in the subsoil with regard to ground-water pollution by oil, Proceedings of the Fifth International Conference on Advances in Water Pollution Research, San Francisco and Hawaii, S. A. Jenkins, ed., Pergammon, Oxford, Vol. 2, HA 7(A)/1-8, 1971.

Mull, R., Calculations and experimental investigations of the migration of oil products in natural soils, Proceedings International Symposium on Groundwater Pollution by Oil Hydrocarbons, International Association of Hydrologists, 167-181, Prague, June 5-9, 1978.

Muskat, M., The Flow of Homogeneous Fluids Through Porous Media, J.W. Edwards, Inc., Ann Arbor, 1946.

Neuman, S.P., Wetting front pressure head in the infiltration model of Green and Ampt, Water Resources Research, 12, 564-566, 1976.

Osborne, M. and J. Sykes, Numerical modelling of immiscible organic transport at the Hyde Park landfill, Water Resources Research, 22, 25-33, 1986.

Pinder, G. F. and L. M. Abriola, On the simulation of nonaqueous phase organic compounds in the subsurface, Water Resources Research, 22, 109s-119s, 1986.

Pope, G.A., L.W. Lake, and F.G. Helfferich, Cation exchange in chemical flooding: Part 1- Basic theory without dispersion, Society of Petroleum Engineers Journal, 18, 418-434, 1978.

Raisbeck, J. M. and M. F. Mohtadi, The environmental impacts of oil spills on land in the arctic regions, Water, Air and Soil Pollution, 3, 195-208, 1974.

Reible, D.D., T.H. Illangasekare, D.V. Doshi, and M.E. Malhiet, Infiltration of immiscible contaminants in the unsaturated zone, Groundwater, 28(5), 685-692, 1990.

Richards, L. A., Capillary conduction of liquids through porous mediums, Physics, 1, 318-333, 1931.

Rozdestvenskii, B. L. and N. N. Janenko, Systems of Quasilinear Equations and Their Applications to Gas Dynamics, American Mathematical Society, Providence, 1980.

Ryan, P. A., and Y. Cohen, One-dimensional subsurface transport of a nonaqueous phase liquid containing sparingly water soluble organics: A front tracking model, Water Resources Research, 27(7), 1487-1500, 1991.

Schiegg, H.O., "Consideration on water, oil and air in porous media," Water Science and Technology, 17, 467-476, 1984.

Sisson, J. B., A. H. Ferguson, and M. Th. van Genuchten, Simple method for predicting drainage from field plots, Soil Science Society of America Journal, 44, 1147-1152, 1980.

Smith, R. E., Approximate soil water movement by kinematic characteristics, Soil Science Society of America Journal, 47, 3-8, 1983.

Smoller, J., Shock Waves and Reaction-Diffusion Equations, Springer, New York, 1983.

Su, C. and R. H. Brooks, Water retention measurement for soils, Journal of the Irrigation and Drainage Division of the American Society of Civil Engineers, 106(IR2), 105-112, 1980.

van Dam, J., "The migration of hydrocarbons in a water-bearing stratum." In The Joint Problems of the Oil and Water Industries, P. Hepple (Ed.), The Institute of Petroleum, London, 1967.

van Genuchten, M. Th., F. J. Leij and S. R. Yates, The RETC Code for Quantifying the Hydraulic Functions of Unsaturated Soils, U.S. EPA, EPA/600/2-91/065, December, 1991.

Weaver, J.W., R.J. Charbeneau, and B.K. Lien, A screening model for nonaqueous phase liquid transport in the vadose zone using Green-Ampt and kinematic wave theory, Water Resources Research, 30(1), 93-105, 1994.

Weaver, J. W., R.J. Charbeneau, J.K. Tauxe, B.K. Lien, and J.B. Provost, The Hydrocarbon Spill Screening Model (HSSM) Volume 1: User's Guide, US EPA, EPA/600/R-94/039a, 1994.

Weaver, J. W., Approximate Multiphase Flow Modeling By Characteristic Methods, U.S. Environmental Protection Agency, EPA/600/2-91/015, NTIS PB91-190959, May, 1991.

Weaver, J.W., and R. J. Charbeneau, Hydrocarbon spill exposure assessment modeling, Proceedings of the NWWA/API Conference on Petroleum Hydrocarbons and Organic Chemicals, Oct. 31 - Nov. 2, Houston, Texas, pp.233-247, 1990.

Wilson, J.L., S. H. Conrad, W.R. Mason, W. Peplinski and E. Hagan, 1990, Laboratory Investigation of Residual Liquid Organics from Spills, Leaks and the Disposal of Hazardous Wastes in Groundwater, U.S. Environmental Protection Agency, EPA/600/6-90/004, April, 1990.

Wylie, M. R. J. and G. H. F. Gardner, The generalized Kozeny-Carman equation: Its application to problems of multiphase flow in porous media, World Oil, 146, 210-227, 1958.

Youngs, E. G. and A. J. Peck, Moisture profile development and air compression during water uptake by bounded porous bodies: 1. Theoretical introduction, Soil Science, 98(5), 290-294, 1964.

Appendix 1 Evaluation of the Volume Integral

The objective of this appendix is to evaluate the total volume contained within the lens out to a radius R, as shown in **Figure 62**. The volume within the source cylinder beneath the surface source is

$$V_C = \pi R_s^2 \beta h_{os} \quad \textbf{(90)}$$

Figure 62 Representation for the lens volume

The volume contained within the outer part of the lens is given by

$$V_O = \beta \int_{R_s}^{R} 2\pi h_o(r)\, dr \quad \textbf{(91)}$$

With equation **(38)** this gives

$$V_O = E \int_{R_s}^{R} r \sqrt{\ln\left(\frac{R_t}{r}\right)^2}\, dr \quad \textbf{(92)}$$

where

$$E = \frac{2\pi\beta h_{os}}{\sqrt{\ln\left(\frac{R_t}{R_s}\right)^2}} \quad \textbf{(93)}$$

To evaluate **(92)** substitute

$$w^2 = \ln\left(\frac{R_t}{r}\right)^2 \quad \textbf{(94)}$$

to find

$$V_O = E R_t^2 \int_{W_R}^{w_{R_s}} w^2 e^{-w^2} dw \quad \textbf{(95)}$$

Integrating by parts $\int udv = uv - \int vdu$ with

$$\begin{aligned} u &= -\frac{w}{2}; \quad du = -\frac{dw}{2} \\ v &= e^{-w^2}; \quad dv = -2 w e^{-w^2} dw \end{aligned} \quad \textbf{(96)}$$

equation (**(95)**) becomes,

$$V_O = E R_t^2 \left(\frac{W_R}{2}\left(\frac{R}{R_t}\right)^2 - \frac{W_{R_S}}{2}\left(\frac{R_s}{R_t}\right)^2 + \frac{1}{2}\int_{W_R}^{W_{R_s}} e^{-w^2} dw \right) \quad \textbf{(97)}$$

noting that

$$e^{-w^2} = \left(\frac{r}{R_t}\right)^2 \quad \textbf{(98)}$$

With the definition of the error function and equation **(93)**, equation **(97)** becomes

$$V_O = \pi\beta h_{os}\left(R^2 \sqrt{\frac{\ln\left(\frac{R_t}{R}\right)^2}{\ln\left(\frac{R_t}{R_s}\right)^2}} - R_s^2 \right) + \frac{\pi\beta h_{os} R_t^2 \sqrt{\frac{\pi}{4}}}{\sqrt{\ln\left(\frac{R_t}{R_s}\right)^2}} \left(erf\left(\sqrt{\ln\left(\frac{R_t}{R_s}\right)^2} \right) - erf\left(\sqrt{\ln\left(\frac{R_t}{R}\right)^2} \right) \right) \quad \textbf{(99)}$$

The total volume with $R > R_s$ is given by $V_T = V_C + V_O$, or

$$V_T(R) = \pi\beta h_{os} R^2 \sqrt{\frac{\ln\left(\frac{R_t}{R}\right)^2}{\ln\left(\frac{R_t}{R_s}\right)^2}} + \frac{\pi\beta h_{os} R_t^2 \sqrt{\frac{\pi}{4}}}{\sqrt{\ln\left(\frac{R_t}{R_s}\right)^2}} \left(erf\left(\sqrt{\ln\left(\frac{R_t}{R_s}\right)^2} \right) - erf\left(\sqrt{\ln\left(\frac{R_t}{R}\right)^2} \right) \right) \quad \textbf{(100)}$$

As expected, equation **(100)** reduces to equation **(46)** when $R = R_t$.

Appendix 2 Summary of KOPT and OILENS Sensitivity Results

Table 12 contains a summary of results from the HSSM sensitivity analysis that is presented in Section 5. Further information is provided in that Section.

Table 12 Summary of Sensitivity Results

Parameter Value	Water table Arrival Time (d)	Mass Flux			Effective Source Concentration (equation 88) (mg/L)	Data File Name
		Time (d)	Radius (m)	Peak Value (kg/d)		
A1. Initial Contaminant Concentration						
820 mg/L	25.0	116.7	8.27	0.0070	3.9	XIC1.DAT
8208 mg/L	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
12300 mg/L	24.6	124.1	8.49	0.1040	55.7	XIC2.DAT
B1. Source Radius						
0.2 m	24.6	56.5	0.80	0.0166	32.3	XSOURCE1.DAT
1.0 m	24.6	92.9	4.16	0.0230	36.7	XSOURCE3.DAT
2.0 m	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
4.0 m	24.6	166.6	17.0	0.2080	38.2	XSOURCE2.DAT
D1. Depth to Water						
7.5 m	7.74	89.0	10.23	0.1040	41.9	DEPTH75.DAT
10.0 m	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
12.5 m	70.5	186.4	6.60	0.0411	32.4	DEPTH125.DAT
D2. Porosity and Bulk Density						
0.35 1.72 g/cm^3	10.5	84.8	10.43	0.1050	41.0	SP1.DAT
0.43 1.51 g/cm^3	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
0.50 1.32 g/cm^3	49.2	168.0	6.94	0.0475	34.66	SP2.DAT

Table 12 (Continued) Summary of Sensitivity Results						
Parameter Value	**Water table Arrival Time (d)**	**Mass Flux**			**Effective Source Concentration (equation 88) (mg/L)**	**Data File Name**
		Time (d)	**Radius (m)**	**Peak Value (kg/d)**		
D3. NAPL viscosity						
0.3 cP	16.8	97.3	9.14	0.0828	39.6	XVIS1.DAT
0.45 cP	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.60 cP	32.5	135.9	7.69	0.0608	37.9	XVIS2.DAT
D4. Vadose Zone Residual NAPL Saturation						
0.0	11.9	108.0	11.5	.121	40.7	SORV0.DAT
0.025	15.7	113.8	10.1	.0976	39.9	SORV025.DAT
0.05	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.075	47.5	148.4	6.06	0.0363	32.6	SORV075.DAT
D5. Soil/water Partition Coefficient for the Constituent						
0.0415 L/kg	24.6	118.7	8.37	0.0705	38.6	FOC0005.DAT
0.083 L/kg	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.166 L/kg	24.6	125.3	8.29	0.0678	37.0	FOC002.DAT
D6. NAPL/water Partition Coefficient for the Constituent						
250	24.6	110.3	7.95	0.0775	45.9	XPC1.DAT
311	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
375	24.6	145.9	9.18	0.0635	30.2	XPC2.DAT
D7. Smear Zone Thickness						
0.065 m	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.25 m	24.3	129.7	7.48	0.0594	38.7	SZ25.DAT
0.50 m	24.0	130.4	6.41	0.0499	41.1	SZ50.DAT
0.75 m	23.6	124.9	5.60	0.0431	43.7	SZ75.DAT
1.00 m	23.3	126.6	5.07	0.0382	45.0	SZ100.DAT

Table 12 (Continued) Summary of Sensitivity Results						
Parameter Value	**Water table Arrival Time (d)**	**Mass Flux**			**Effective Source Concentration (equation 88) (mg/L)**	**Data File Name**
		Time (d)	**Radius (m)**	**Peak Value (kg/d)**		
E1. Aquifer Thickness						
0.5 m	24.6	121.3	8.39	0.0696	149.2	AQU05.DAT
1.0 m	24.6	121.3	8.39	0.0696	74.6	AQU10.DAT
1.5 m	24.6	121.3	8.39	0.0696	49.7	AQU15.DAT
1.966 m	24.6	121.9	8.39	0.0697	38.0	AQU19.DAT
15.0 m	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
20.0 m	24.6	121.3	8.39	0.0696	38.0	AQU20.DAT
E2. Transverse Horizontal Dispersivity						
0.5 m	24.6	121.3	8.39	0.0696	38.0	DIST05.DAT
1.0 m	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
1.5 m	24.6	121.3	8.39	0.0696	38.0	DIST15.DAT
E3. Constituent Half-Life						
Infinite	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
495.1 d	24.6	121.9	8.41	0.0697	26.7	XHL1.DAT
247.5 d	24.6	121.9	8.41	0.0697	23.8	XHL2.DAT
F1. Ratio of horizontal to vertical conductivity						
1.0	24.6	226.8	9.28	0.0377	39.9	RKS1.DAT
2.5	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
10.0	24.6	56.4	6.81	0.164	32.3	RKS10.DAT
F2. Gradient						
0.005	24.6	177.8	9.97	0.0494	38.6	GRAD005.DAT
0.01	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
0.02	24.6	89.1	7.14	0.0985	35.5	GRAD02.DAT

Table 12 (Continued) Summary of Sensitivity Results

Parameter Value	Water table Arrival Time (d)	Mass Flux			Effective Source Concentration (equation 88) (mg/L)	Data File Name
		Time (d)	Radius (m)	Peak Value (kg/d)		
F3. Transverse vertical dispersivity						
0.05 m	24.6	142.1	9.04	0.0584	39.1	DIS5.DAT
0.1 m	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
0.15 m	24.6	117.7	8.27	0.0775	35.7	DIS30.DAT
F4. All Dispersivities						
a_L = 5.0 m a_T = 0.5 m a_V = 0.05 m	24.6	146.9	9.18	0.0586	38.3	DVL.DAT
a_L = 10.0 m a_T = 0.1 m a_V = 0.01 m	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
a_L = 15.0 m a_T = 1.5 m a_V = 0.15 m	24.6	111.3	8.04	0.0772	37.2	DVH.DAT
F5. Percent maximum radius						
25	24.6	121.3	4.28	0.0696	106.4	MN25.DAT
40	24.6	121.9	6.84	0.0697	51.96	MN50.DAT
49.15	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
55	24.6	121.9	9.41	0.0697	31.9	MN55.DAT
75	24.6	121.3	12.8	0.0696	19.8	MN75.DAT
100	24.6	121.3	17.1	0.0696	12.7	MN100.DAT
G1. Hydraulic conductivity						
1.75 m/d	77.9	429.2	8.45	0.0211	38.5	HC0175.DAT
3.5 m/d	44.0	225.3	8.34	0.0377	38.9	HC035.DAT
7.1 m/d	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
14.2 m/d	14.1	63.4	8.27	0.131	37.8	HC142.DAT
28.4 m/d	8.2	334.2	8.26	0.251	36.9	HC284.DAT

Table 12 (Continued) Summary of Sensitivity Results						
Parameter Value	**Water table Arrival Time (d)**	**Mass Flux**			**Effective Source Concentration (equation 88) (mg/L)**	**Data File Name**
		Time (d)	**Radius (m)**	**Peak Value (kg/d)**		
H1. Recharge						
0.0 in/yr	60.7	175.0	6.98	0.0442	34.1	RECH0.DAT
2.0 in/yr	33.0	141.6	8.17	0.0598	36.1	RECH.DAT
10.0 in/yr	27.1	128.6	8.36	0.0657	37.3	RECH10.DAT
20.0 in/yr	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
30.0 in/yr	23.2	117.3	8.42	0.0728	38.2	RECH30.DAT
H2. Longitudinal Dispersivity						
5.0 m	24.6	121.3	8.39	0.0696	38.0	DIS5.DAT
10.0 m	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
15.0 m	24.6	121.3	8.39	0.0696	38.0	DIS15.DAT
30.0 m	24.6	121.3	8.39	0.0696	38.0	DIS30.DAT
H3. Residual water saturation						
0.05	26.5	129.1	8.44	0.0684	37.0	XWAT1.DAT
0.10	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.15	22.8	114.3	8.37	0.0709	38.8	XWAT2.DAT
H4. van Genuchten's n						
2.0	63.7	174.5	7.28	0.0492	33.4	VGN20.DAT
2.68	42.1	143.3	7.72	0.0582	36.1	VGN26.DAT
3.0	36.7	140.6	8.02	0.0614	35.9	VGN30.DAT
4.5	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
H5. Source flux						
0.3392 m/d	74.2	172.0	5.40	0.0299	32.0	XSF1.DAT
0.4522 m/d	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
0.9044 m/d	3.6	99.6	15.87	0.213	43.6	XSF2.DAT

Table 12 (Continued) Summary of Sensitivity Results						
Parameter Value	**Water table Arrival Time (d)**	**Mass Flux**			**Effective Source Concentration (equation 88) (mg/L)**	**Data File Name**
		Time (d)	**Radius (m)**	**Peak Value (kg/d)**		
H5. NAPL saturation in the lens						
0.2000	24.6	133.1	7.65	0.0612	38.5	XSAT25.DAT
0.3236	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
0.4500	24.6	111.4	8.47	0.0724	38.9	XSAT45.DAT
I1. Aquifer Thickness (see E1)						
I2. van Genuchten's α						
2.0 m^{-1}	24.6	123.9	8.30	0.0684	38.0	VGA20.DAT
2.68 m^{-1}	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
3.4 m^{-1}	24.6	119.6	8.46	0.0706	38.0	VGA34.DAT
I3. Water surface tension						
58 dyne/cm	24.6	124.8	8.44	0.0693	37.5	WST1.DAT
65 dyne/cm	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
72 dyne/cm	24.6	125.9	8.60	0.0703	36.9	WST2.DAT
I4. Maximum water phase relative permeability during infiltration						
0.4	24.6	118.5	8.30	0.0694	38.5	MKR1.DAT
0.5	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.6	24.6	121.0	8.38	0.0595	38.1	MKR2.DAT
I5. NAPL surface tension						
25 dyne/cm	24.6	118.2	8.21	0.0687	38.8	NST1.DAT
35 dyne/cm	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
45 dyne/cm	24.6	121.3	8.48	0.0703	37.7	NST2.DAT

Table 12 (Continued) Summary of Sensitivity Results						
Parameter Value	**Water table Arrival Time (d)**	**Mass Flux**			**Effective Source Concentration (equation 88) (mg/L)**	**Data File Name**
		Time (d)	**Radius (m)**	**Peak Value (kg/d)**		
I6. NAPL density						
0.64 g/cm^3	27.6	119.2	8.48	0.0705	37.8	XDEN1.DAT
0.72 g/cm^3	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.80 g/cm^3	22.3	121.3	7.95	0.0661	39.2	XDEN2.DAT
I7. Aquifer residual NAPL saturation						
0.075	24.6	125.0	8.58	0.0705	37.2	SORS075.DAT
0.15	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.30	24.6	116.7	8.12	0.0682	39.1	SORS30.DAT
0.50	24.6	118.0	8.01	0.0666	39.0	SORS50.DAT
I8. NAPL/water interfacial tension						
30 dyne/cm	24.6	119.3	8.65	0.0721	37.5	X2BT1.DAT
45 dyne/cm	24.6	121.9	8.41	0.0697	37.9	X2BT.DAT
60 dyne/cm	24.6	122.5	8.18	0.0676	38.4	X2BT2.DAT
I9. Capillary thickness parameter						
0.001	24.7	124.6	9.02	0.0742	36.2	CAP1.DAT
0.01	24.6	121.9	8.41	0.0696	37.9	X2BT.DAT
0.02	24.5	117.5	7.82	0.0652	39.6	CAP2.DAT

Table 13 Fractions of Y, N, and I Responses for Each Type (A-I)			
Type (Number of Parameters)	**Fraction of Parameters for (KOPT, OILENS, and TSGPLUME)**		
	Y	**N**	**I**
A (1)	(1.000, 0.000, 0.000)	(0.000, 0.000, 0.000)	(0.000, 1.000, 1.000)
B (1)	(1.000, 0.000, 0.000)	(0.000, 0.000, 0.000)	(0.000, 1.000, 1.000)
C (0)	(0.000, 0.000, 0.000)	(0.000, 0.000, 0.000)	(0.000, 0.000, 0.000)
D (7)	(0.857, 0.857, 0.286)	(0.143, 0.000, 0.000)	(0.000, 0.143, 0.714)
E (2)	(0.000, 0.000, 1.000)	(1.000, 1.000, 0.000)	(0.000, 0.000, 0.000)
F (6)*	(0.000, 0.500, 1.000)	(1.000, 0.500, 0.000)	(0.000, 0.000, 0.000)
G (1)	(1.000, 1.000, 1.000)	(0.000, 0.000, 0.000)	(0.000, 0.000, 0.000)
H (7)	(0.714, 0.571, 0.286)	(0.286, 0.143, 0.000)	(0.000, 0.286, 0.714)
I (9)	(0.566, 0.778, 0.111)	(0.444, 0.111, 0.000)	(0.000, 0.111, 0.889)

* For Type F, eight is used in the denominator for calculating the fractions, because of the three parameters used for "all dispersivities"

Appendix 3 FORTRAN Source Codes for HSSM and the Utility Programs

3.1 Source Code for HSSM-KO

The source code for HSSM-KO consists of a driver (RNHSSM), 80 subroutines, one function and one block data subprogram. The following listing is organized roughly according to function as follows. The heart of the program consists of the four subroutines HSSM, RKF12, EQS, and CHK. These routines are listed first. The main program RNHSSM and the subroutine HSSM "manage" the simulation calling the input, initialization, computation, post processing and output subroutines. The computations are performed in the Runge-Kutta solver (subroutine RKF12) as all of the KOPT and OILENS equations are in the form of ordinary differential equations. The right hand sides of those equations are contained in the subroutine EQS and the subroutines which it calls: OEQS, CEQS, OILENS, and CLENS. It is throught EQS that the equations to be solved are entered into the solver. The solution of the equations is controlled by the subroutine CHK and the subroutines which is calls. These guide the solver to critical points in the solution and turn equations on and off as their solutions are needed.

The remaining routines are grouped into the following categories and are listed in sequence.

- KOPT equations (to complete OEQS and CEQS)
- OILENS equations (to complete OILENS and CLENS)
- General parameter calculation used by KOPT and OILENS routines
- Simulation control incorporated into the CHK subroutines
- Mass balance calculation routines
- Numerical methods
- Input routines
- Initialization routines
- Post processing
- Output
- File manipulation

```
      SUBROUTINE HSSM
C     ************************************************************
C     ************************************************************
C     *
C     *   HSSM-KO    KOPT and OILENS Modules of
C     *              the  HYDROCARBON SPILL SCREENING MODEL (HSSM)
C     *
C     *
C     *
C     ************************************************************
C     *
C     *   WARNING:
C     *   THE RESULTS FROM THIS CODE HAVE NOT BEEN VERIFIED BY
C     *   EITHER LABORATORY OR FIELD STUDIES.
C     *   THIS PROGRAM SIMULATES IDEALIZED BEHAVIOR OF OILY-PHASE
C     *   CONTAMINANTS IN IDEALIZED POROUS MEDIA, AND IS NOT INTENDED
C     *   FOR APPLICATION TO HETEROGENEOUS SITES.
C     *   THE SUITIBILITY OF THIS PROGRAM FOR ANY PURPOSE IS NOT IM-
C     *   PLIED.  NEITHER THE AUTHOR, THE UNIVERSITY OF TEXAS, NOR
C     *   THE UNITED STATES GOVERNMENT ACCEPTS ANY LIABILITY RESULTING
C     *   FROM ITS USAGE.
C     *
C     ************************************************************
C     *
C     *   AUTHOR  JIM WEAVER
C     *           U.S. ENVIRONMENTAL PROTECTION AGENCY
C     *           R.S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *           ADA, OKLAHOMA  74820
C     *
C     *   REFERENCES:
C     *   Weaver, J.W., R.J. Charbeneau, and B.K. Lien, A Screening
C     *    Model for Vadose Zone Nonaqueous Phase Liquid Transport
C     *    Using Green-Ampt and Kinematic Wave Theory, Water Resources
C     *    Research, 30(1), 1994
C     *   Weaver, J.W., R.J. Charbeneau, J.D. Tauxe, B.K. Lien, and
C     *    J.B. Provost, The Hydrocarbon Spill Screening Model (HSSM)
C     *    Volume 1:  User's Guide, US EPA, 1994, EPA/600/R-94/39a
C     *  Charbeneau, RJ and JW Weaver, The Hydrocarbon Spill Screening
C     *    Model Volume 2: Theoretical Background and Source Codes,
C     *    US EPA, 1994, EPA/600/R-94/039b
C     *
C     *  ANSI STD. X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES: HSSM,DATA1,BLOCK DATA THREE,EQS,OEQS,ENTROP,
C     *                    CEQS,INCAP,PFILE,RP18,RP19,DMONTE,MOUT,MESSAG,
C     *                    CHK,YWRITE,SCHK,SCHK2,OCHK,OCHK2,GACHK,GACHK2,
C     *                    CCHK,CCHK2,RECHAR,DCHK2,GAHS,RD,CVEL,CONCE,
C     *                    CAPSUC,CAFN,GLQ,EDITIN,ZMASS,ZLOC,GLOBAL,OCP,
C     *                    DSO,OIL,SFLUX,OFLUX,DATA,SOPEN,INPUT,RFLAGS,
C     *                    WPC,DPF,WTHICK,UNITS,VG2BC,INITIA,MEPS,TSGP1,
C     *                    TSGP3,MBISEC,WFFS,INTCHR,HEADS,POST,COUNT,WKE,
C     *                   KRCHK,MRKF12,INQUAD,INQSET,BINARY,BISEC,OILENS,
C     *                    CLENS,HMASS,CMASS,LCHK,LCHK2,INTLNS,PROLNS,
C     *                    DUPUIT,MAXSAV,REPACK,VOLT,ERF,LFLUX,PREKOP,
C     *                    NSOPEN,PKCON,DELF,IOPOST,TSGP2,OPNFLE,DIR
C     *  NON ANSI STANDARD ROUTINE
C     *                      CMD
C     *
C     *  REVISION HISTORY:
C     *  OCTOBER 21, 1987    KOPT COMPLETED
C     *  APRIL 15, 1989      FIXED SYSTEM OF UNITS
C     *  APRIL 15, 1989      OILENS INPUT ADDED
C     *  JUNE 28, 1989       DOCUMENTATION REVISED
C     *  JULY 6, 1989        VERTICAL AND HORIZONTAL CONDUCTIVITY
C     *  JULY 31, 1989       VARIOUS STOPPING CRITERIA ADDED
C     *  AUGUST 30, 1989     SELECTIVE READ/WRITE OPTIONS ADDED
C     *  AUGUST 31, 1989     BURDINE-VAN GENUCHTEN RELATIVE PERMEABILITY
C     *  OCTOBER 10, 1989    CAPILLARY SUCTION APPROXIMATION AT FRONT
C     *  JANUARY 11, 1990    DISSOLUTION OF HYDROCARBON PHASE IN KOPT
C     *  JANUARY 25, 1990    TWO DIFFERENT OIL RESIDUALS
C     *  SEPTEMBER 4, 1990   CREATION OF TSGPLUME INPUT DATA FILES
C     *  MARCH 22, 1991      MODIFIED CONSTANT HEAD PONDING CONDITION
C     *                      TO INCLUDE IAT=4
C     *  SEPTEMBER 30, 1991  MONTE CARLO SIMULATION FOR KOPT
C     *  DECEMBER 22, 1991   MULL'S VADOSE ZONE RECTANGULAR PROFILE MODEL
C     *  DECEMBER 27, 1991   ENTROPY INEQUALITY CALCULATION FOR KOPT
C     *  JANUARY 5, 1992     SENSITIVITY ANALYSIS FOR KOPT
C     *
C     ************************************************************
C     ************************************************************
      CHARACTER*40 MFILE(3),OFILE(7),ANS*1
      DIMENSION NEQ(6),YY(101,6)
      COMMON /CPSA/ HOE,HWE,OSIG,PHIF,WSIG,XLAMBO
      COMMON /COUN/ NUM(100)
      COMMON /ENTR/ CHU,CHD,FSPD
      COMMON /FLAG/ ICONC,IKOPT,ILENS,ITSGP
      COMMON /FILE/ MFILE,OFILE
      COMMON /FIL2/ NOF
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /MONT/ IMBAA,NMONT,MNO,NOT,OILM,ZMO(10)
      COMMON /OUTP/ ERCMX,EROMX,NPT,NREG
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RLTV/ KRF,KRFO,XLAMB,A2A,A3A,A4A,A5A,A6A
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /TRAN/ DWV,EVAP,HLC,HLO,PMU,RH,RHOS,TEMP,WMU,WRHO,XXKS,
     *              XXKSH,XXKV
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      EXTERNAL CHK,EQS
      DATA ND /101/
C
```

```
C
C     ***WRITE MESSAGE TO SCREEN
      CALL MESSAG (1)
      IMCS = 0
      NSIM = 0
      NMONT = 0
C
C
 5    CONTINUE
       NUM(1) = 1
       MNO = 0
C
C
C       ***INITIALIZATION
C       ***SET CONSTANTS
       CALL DATA1
C
C
C       ***INPUT DATA
       CALL DATA (IREO,IEND)
       IF (IEND.EQ.1) STOP 'HSSM--END'
       CALL MESSAG (2)
C
C
C       ***COUNT THE MONTE CARLO RUNS
       IF (IWR.EQ.0) THEN
         IF (IMCS.EQ.0) THEN
           NSIM = IMBAA-1
           IMCS = 1
         END IF
         NSIM = NSIM + 1
       END IF
C
C
       CALL MESS4 (30,NSIM)
C
C
C       ***FURTHER INITIALIZATION
       CALL INITIA (NEQ,YY,ND)
C
C       ***ADD TO THE TSGPLUME DATA FILE
       IF (ITSGP.EQ.1) CALL TSGP3 (QW)
C
C
C
C       ***OIL AND DISSOLVED CONSTITUENT COMPUTATIONS
C
C       ***SOLUTION VIA RUNGE-KUTTA-FEHLBERG 1(2) METHOD IMPLEMENTED IN
C       ***ROUTINE MRKF12
C       ***RIGHT HAND SIDES OF O.D.E.S ARE CALCULATED IN SUBROUTINE EQS
C       ***AD HOC CONTROLS OF SOLUTION PROGRESS ARE MADE IN ROUTINE CHK
C       ***SOLUTION BEGINS AT TIME TPB AND ENDS AT TIME TM
C       ***MAXIMUM TIME STEP = DM
C       ***MRKF12 TRUNCTION ERROR CONTROL IS INVOKED FOR ERRORS OF 1.E-5
C       ***ACTIVE DIFFERENTIAL EQUATIONS ARE INDICATED BY
C       ***THE INTEGER ARRAY NEQ,
C       ***SEE ROUTINE EQS FOR MORE INFORMATION
C       ***ON RETURN:
C       ***NPT IS THE NUMBER OF STEPS TAKEN
C       ***NREG IS THE NUMBER OF REJECTED STEPS
C       ***YY CONTAINS THE SOLUTION OF THE EQUATIONS AT TIME TM
C       ***ND IS THE ROW DIMENSION OF YY
C       ***ET IS THE TRUE ENDING TIME OF THE SOLUTION
C
C
       CALL MESSAG (3)
       CALL MRKF12 (EQS,CHK,TPB,TM,DM,1.E-5,NEQ,NPT,NREG,YY,ND,ET)
C
C
C
       CALL MESSAG (12)
C       ***POST PROCESSING
C
C
C       ***CALCULATIONS FOR THE OILENS MODEL
       CALL MESSAG (4)
       CALL POST (ET,CLTIME,HLTIME,YY,ND)
C
C
C       ***RUN INFORMATION
       CALL COUNT
C
C
       IF (IWR.EQ.1) THEN
C         ***CREATE KOPT/OILENS OUTPUT FILE
         CALL IOPOST (OFILE(1),IWR,IREO)
C
C         ***CREATE KOPT/OILENS PLOT FILE
         CALL PFILE (OFILE(2),OFILE(3),OFILE(4))
C
C         ***CREATE TSGPLUME DATA FILE
         IF (ITSGP.EQ.1) CALL TSGP2 (OFILE(5),IWR,IREO)
C
C         ***DELETE TEMPORARY FILES
         CALL DELF
         CALL MESSAG (5)
C
       ELSE IF (IWR.EQ.0) THEN
C         ***OUTPUT MONTE CARLO OUTPUT
         CALL MOUT (NSIM)
       END IF
C
C       ***REPEAT SIMULATION
       ANS = 'N'
       IF (NSIM.LT.NMONT) ANS = 'y'
C
     IF (ANS.EQ.'y'.OR.ANS.EQ.'Y') GO TO 5
C
     RETURN
     END
```

```
      SUBROUTINE MRKF12 (FNC,CHK,TX,TF,DM,ER,NEQ,NS,NR,YY,ND,ET)
C     ************************************************************
C     *                                                          *
C     *   MRKF12  (MULTIPLE) RUNGE KUTTA FELHBERG FIRST-SECOND   *
C     *           ORDER ORDINARY DIFFERENTIAL EQUATION SOLVER     *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE SOLVES A SYSTEM OF ORDINARY *
C     *   DIFFERENTIAL EQUATIONS OF THE FORM                     *
C     *                                                          *
C     *   DY/DT = F(Y,T)                                         *
C     *                                                          *
C     *   THE FUNCTIONS F(Z,T) ARE CONTAINED IN THE ROUTINE FNC. *
C     *   THE EQUATIONS ARE IN 6 GROUPS, SOME OF WHICH MAY NOT BE *
C     *   SOLVED AT ALL TIMES DURING THE SOLUTION PROCESS.       *
C     *   ROUTINE CHK CONTAINS CHECKS ON THE SOLUTION AND SWITCHES *
C     *   TO CONTROL THE EXECUTION.  THE FELHBERG FEATURE ALLOWS *
C     *   THE TIME STEP TO VARY.                                 *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................  *
C     *   FNC   = SUBROUTINE WHICH CONTAINS FUNCTIONS F(Y,T)     *
C     *   CHK   = SUBROUTINE WHICH CONTAINS CHECKS AND CONTROLS  *
C     *   TX    = STARTING TIME                                  *
C     *   TF    = ENDING TIME                                    *
C     *   DM    = MAXIMUM TIME STEP                              *
C     *   ER    = ERROR TOLERANCE PER STEP                       *
C     *   NEQ   = ARRAY CONTAINING NUMBER OF EQUATIONS PER GROUP *
C     *   NS    = STEP NUMBER (STARTING)                         *
C     *   YY    = INITIAL CONDITION                              *
C     *   ND    = ROW DIMENSION OF YY                            *
C     *                                                          *
C     *   OUTPUT ARGUMENTS.....................................  *
C     *   DM    = MAXIMUM TIME STEP                              *
C     *   NEQ   = NUMBER OF EQUATIONS PER GROUP                  *
C     *   NS    = STEP NUMBER (ENDING)                           *
C     *   NR    = NUMBER OF REJECTED STEPS                       *
C     *   YY    = SOLUTION AT ENDING TIME                        *
C     *   ET    = ENDING TIME OF SOLUTION                        *
C     *                                                          *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   REFERENCE:  FEHLBERG, 1969, NASA TR R-315              *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...SEE INPUT ARGUMENTS                *
C     *   X-21-87                                                *
C     *                                                          *
C     ************************************************************
      COMMON /COUN/ NUM(100)
C     ***CAUTION: THE FIRST DIMENSION OF XK0,XK1,XK2 & YI MUST MATCH ND
      DIMENSION DTR(20),IRR(20),NEQ(*),XK0(101,6),XK1(101,6),XK2(101,6),
     1YI(101,6),YY(ND,*)
      PARAMETER (A1=0.5,B10=A1,B20=1./256.,B21=255./256.,C3=1./512.)
C     DATA IRR,NDT,NGR /20*0,20,6/
C
C
      DO 11 I=1,20
        IRR(I)=0
11    CONTINUE
      NDT=20
      NGR=6
C
      NUM(17) = NUM(17) + 1
      IF (ND.NE.101) STOP 'RUNGE-KUTTA'
C
C     ***INITIAL VALUES
      IC   = 0
      DT   = 0.0001
      TN   = TX
      IU   = 0
C
 1    CONTINUE
C     ***SAVE INITIAL CONDITION IN CASE PROPOSED TIME STEP IS
C     ***TOO LARGE
      DO 25  J=1,NGR
        NEND = NEQ(J)
        IF (J.EQ.1) NEND = MAX(NEQ(J),1)
        nend = nd
        DO 20  I=1,NEND
          YI(I,J) = YY(I,J)
 20     CONTINUE
 25   CONTINUE
 30   CONTINUE
      DO 35 I=1,NDT
        DTR(I) = 1.E20
 35   CONTINUE
C
C     ***INTEGRATE ONE STEP
C     ***SET SOME CONSTANTS FOR THIS TIME STEP
      A1D  = DT * A1
      B10D = DT * B10
      B20D = DT * B20
      B21D = DT * B21
      C3D  = DT * C3
      XT = TN
C
C     ***CHECK FOR A CHANGE IN THE NUMBER OF EQUATIONS BEING SOLVED
c     NTEQ = 0
      DO 37 J=1,NGR
        NTEQ = NTEQ + NEQ(J)
 37   CONTINUE
C     ***IF THERE IS A CHANGE IN THE NUMBER OF EQUATIONS
C     ***THEN EVALUATE AT XT
```

```
      IF (NTEQ.NE.NTEQO) IU = 0
      NTEQO = NTEQ
C
C     ***BEGIN SERIES OF FUNCTION EVALUATIONS FOR CERTAIN
C     ***VALUES OF TN AND YY
      IF (IU.EQ.0) CALL FNC (XT,YI,XK0,ND,NEQ)
C
C     ***THE SECOND EVALUATION
      TN = XT + A1D
      DO 45  J=1,NGR
        DO 40  I=1,NEQ(J)
          IF (XK0(I,J).EQ.0.) GO TO 40
          YY(I,J) = YI(I,J) + B10D*XK0(I,J)
 40     CONTINUE
 45   CONTINUE
      CALL FNC (TN,YY,XK1,ND,NEQ)
C
C     ***THE THIRD EVALUATION
      TN = XT + DT
      DO 55  J=1,NGR
        DO 50  I=1,NEQ(J)
          YY(I,J) = YI(I,J) + B20D*XK0(I,J) + B21D*XK1(I,J)
 50     CONTINUE
 55   CONTINUE
      CALL FNC (TN,YY,XK2,ND,NEQ)
C
C     ***DETERMINE NEW TIME STEP FROM THE FUNCTION EVALUATIONS
      IE = 0
      IE2 = 0
      TR = 1000.
      DO 75  J=1,NGR
        DO 70  I=1,NEQ(J)
          IF (XK0(I,J).EQ.XK2(I,J)) GO TO 70
          IF (XK0(I,J).EQ.0..AND.XK2(I,J).GT.0.) GO TO 70
          TE = ABS(  C3D*(XK0(I,J) - XK2(I,J))  )
          BB = TE / (AMAX1 (1., ABS(  YY(I,J)     ) ) )
          IF (BB.GT.ER) THEN
C           ***STEP REJECTED
            IE = 1
            DTR(1) = DT*.5
            IRJ = J
            IRI = I
            GO TO 76
          END IF
          TS = DT * (  (ER + ER*ABS(YY(I,J)   ) )/TE  ) ** 0.25
          IF (TS.LT.TR) TR = TS
 70     CONTINUE
 75   CONTINUE
 76   CONTINUE
C
      IE2 = 0
      DT1 = DT
C
C     ***CHECK CONDITIONS FOR STEP REJECTION
      CALL CHK (TN,YY,YI,XK2,ND,NEQ,DM,DT1,DTR,NS,IC,IRR,IE2,IE)
      ET = TN
      IF (IC.EQ.1) RETURN
      IF (IE2.EQ.1) IU = 0
      IF (IE.EQ.1.OR.IE2.EQ.1) THEN
C       ***REJECT STEP
        NR = NR + 1
        TN = XT
C
C       ***SELECT SMALLEST STEP SIZE LIMITATION
        DT = 1.E19
        DO 80 II=1,NDT
          IF (DTR(II).LT.DT.AND.DTR(II).GT.0.) THEN
            DT = DTR(II)
            IN = II
          ELSE IF (DTR(II).LE.0.) THEN
            WRITE (*,*) 'STEP REJECTION ERROR ',II
            STOP
          END IF
 80     CONTINUE
        IRR(IN) = 1
        GO TO 30
      END IF
      NS = NS + 1
C
C     ***SET NEW TIME STEP
      IU = 1
      IF (DT1.LT.DT) TR = DT1*1.25
      DT = TR * 0.80
      IF (DT.GT.DM) DT = DM
C
C     ***THIRD EVALUATION BECOMES FIRST EVALUATION FOR NEW STEP
      DO 95 J=1,NGR
        DO 90 I=1,NEQ(J)
          XK0(I,J) = XK2(I,J)
 90     CONTINUE
 95   CONTINUE
      GO TO 1
      END
```

```
      SUBROUTINE EQS (TT,ZZ,F,ND,NEQ)
C     ************************************************************
C     *                                                          *
C     *   EQS  EQUATIONS FOR HSSM-KO                             *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE CONTAINS THE FUNCTIONS F(Y,T) *
C     *   FOR THE OIL AND CONSTITUENT EQUATIONS THAT ARE USED BY  *
C     *   MRKF12.                                                *
C     *   THE EQUATIONS ARE DIVIDED INTO 6 GROUPS AS FOLLOWS     *
C     *   GROUP 1.  #1 OIL FRONT                                 *
C     *             #2 VOLATILE FRONT                            *
C     *             #3 OIL HEAD AT SURFACE (IAT = 4 ONLY)        *
C     *   GROUP 2.  OIL CHARACTERISTICS                          *
C     *             FROM THE SURFACE (#1) TO THE OIL FRONT (UP   *
C     *             TO #101)                                     *
C     *   GROUP 3.  CONSTITUENT CHARACTERISTICS                  *
C     *             FROM THE FRONT (#1) TO THE TOP (#11 OR #15)  *
C     *   GROUP 4.  OILENS SUBMODEL                              *
C     *             #1 CUMULATIVE OIL VOLUME ENTERING LENS       *
C     *             #2 LENS HEIGHT                               *
C     *             #3 RADIUS                                    *
C     *   GROUP 5.  CONSTITUENT CONCENTRATION IN OILENS          *
C     *             #1 CUMULATIVE CONSTITUENT MASS ENTERING LENS *
C     *             OR CROSSING WATER TABLE                      *
C     *             #2 CONTAMINANT SPEED IN LENS                 *
C     *   GROUP 6.  MISC.                                        *
C     *             #1 CUMULATIVE OIL PHASE RUNOFF VOLUME/AREA   *
C     *             #2 CUMULATIVE OIL PHASE APPLIED AT BOUNDARY  *
C     *   THIS ARRANGEMENT ALLOWS THE MRKF12 ROUTINE TO VARY     *
C     *   EASILY THE NUMBER OF EQUATIONS SOLVED                  *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................  *
C     *   TT      = TIME                                         *
C     *   ZZ(I,J) = DEPTH FOR EQUATION I OF GROUP J              *
C     *   ND      = ROW DIMENSION OF ZZ                          *
C     *   NEQ(J)  = ARRAY OF NUMBER OF EQUATIONS FOR EACH GROUP  *
C     *                                                          *
C     *   OUTPUT ARGUMENTS....................................  *
C     *   F(I,J) = FUNCTION VALUE FOR EQUATION I OF GROUP J      *
C     *   NEQ(J) = NUMBER OF EQUATIONS PER GROUP                 *
C     *                                                          *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...OEQS,OILENS,CEQS,CLENS             *
C     *   X-21-87                                                *
C     *                                                          *
C     ************************************************************
      DIMENSION F(ND,*),NEQ(*),ZZ(ND,*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /COUN/ NUM(100)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /MATR/ ETA,OMSPR,OMSWR,SAR,SR,SWR
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(2) = NUM(2) + 1
C
C
      PT(2) = TT
      ISE = ISE + NEQ(1)+NEQ(2)+NEQ(3)+NEQ(4)+NEQ(5)+NEQ(6)
C
C
C     ***OIL COMPUTATIONS
      CALL OEQS (TT,ZZ,F,ND,NEQ)
C     ***CONSTITUENT EQUATIONS
      IF (COINI.GT.0.) CALL CEQS (TT,ZZ,F,ND,NEQ)
C     ***OILENS SUBMODEL
      CALL OILENS (TT,ZZ,F,ND,NEQ)
      CALL CLENS (TT,ZZ,F,ND,NEQ)
C
C
      RETURN
      END
```

```
      SUBROUTINE CHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ***************************************************************
C     *                                                             *
C     *   CHK                                                       *
C     *                                                             *
C     *                                                             *
C     *   PURPOSE....THIS SUBROUTINE CONTAINS A SERIES OF CHECKS    *
C     *   AND CONTROLS FOR THE SOLUTION OF THE KINEMATIC MODEL.     *
C     *   IF A STEP IS NOT REJECTED BY CHK OR MRKF12, THEN RESULTS  *
C     *   FOR THE CURRENT STEP ARE OUTPUT AND VARIOUS VALUES ARE    *
C     *   ROLLED FOR THE NEXT EVALUATION.                           *
C     *                                                             *
C     *   INPUT ARGUMENTS.....................................     *
C     *   TT      = SOLUTION TIME JUST OBTAINED                     *
C     *   ZZ(I,J) = DEPTH FOR EQUATION I GROUP J                    *
C     *   Z1(I,J) = PREVIOUS DEPTH FOR EQUATION I GROUP J           *
C     *   F(I,J)  = CURRENT FUNCTION VALUES                         *
C     *   ND      = ROW DIMENSION OF ZZ(),Z1(), & F()               *
C     *   NEQ(J)  = NUMBER OF EQUATION IN EQCH GROUP                *
C     *   DMAX    = MAXIMUM TIME STEP                               *
C     *   DT      = CURRENT TIME STEP                               *
C     *   NS      = NUMBER OF STEPS                                 *
C     *   IRP     = 1 IF MRKF12 ALREADY HAS REJECTED THE STEP       *
C     *                                                             *
C     *   OUTPUT ARGUMENTS.....................................     *
C     *   NEQ(J)  = ADJUSTED NUMBER OF EQUATIONS                    *
C     *   DMAX    = ADJUSTED MAX. TIME STEP                         *
C     *   DT      = NEW TIME STEP FOR ACCEPTED STEPS                *
C     *   DTR()   = SERIES OF PROPOSED REDUCED TIME STEPS           *
C     *   IC      = 1 TO END SIMULATION                             *
C     *   IRR()   = CODES CORESPONDING TO VARIOUS REDUCED STEPS     *
C     *   IR      = 1 TO REJECT STEP                                *
C     *                                                             *
C     *                                                             *
C     *   AUTHOR  JIM WEAVER                                        *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN                 *
C     *                                                             *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77          *
C     *   REQUIRED ROUTINES...ZMASS,COUNT,HIST,OCHK,OCHK2,GACHK,    *
C     *                       GACHK2,CCHK,CCHK2,LCHK,LCHK2,SCHK,    *
C     *                       SCHK2,DHCHK2,YWRITE,MESS2             *
C     *   THIS ROUTINE IS CALLED BY:  MRKF12                        *
C     *   X-21-87                                                   *
C     *                                                             *
C     ***************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /ENTR/ CHU,CHD,FSPD
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /HIPR/ IHIST,NTIMES,PR(10),NZS,HI(5)
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /MONT/ IMBAA,NMONT,MNO,NOT,OILM,ZMO(10)
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /OUTP/ ERCMX,EROMX,NPT,NREG
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /TRIA/ DTOLD,DTOLD1,I1,I2,I5,ICC,ICHAR,TEND
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(3) = NUM(3) + 1
C
C
      IF (NUM(3).EQ.1) THEN
        T2 = 0.0
        OZZ1 = DPA
      END IF
C
C
C
C     ***1A.) TENTATIVE CHECK CONDITIONS
C     ***OIL CHECKS
      CALL OCHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***VARIABLE HEAD PONDING CHECKS
      CALL GACHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***CONSTITUENT CHECKS
      CALL CCHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***OILENS SUBMODEL CHECKS
      IF (WTABLE.GT.0.)
     1  CALL LCHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***SIMULATION CHECKS
      CALL SCHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C
C
C
C     ***RETURN ANY REJECTED STEP
      IF (IR.EQ.1.OR.IRP.EQ.1) RETURN
C
C
C
C
C     ***SR CHK PART 1B.) CORRECTIONS FOR CONDITIONS DETECTED ABOVE
C     ***OIL CHECKS
      CALL OCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
```

```
C     ***VARIABLE HEAD PONDING
      CALL GACHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***CONSTITUENT CHECKS
      CALL CCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***DISSOLVED HYDROCARBON CHECKS
      CALL DHCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***OILENS SUBMODEL CHECKS
      IF (WTABLE.GT.0.)
     1  CALL LCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***SIMULATION CHECKS
      CALL SCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C
C
C     ***SR CHK PART 2.)  DATA OUTPUT FOR ACCEPTED STEPS
C     ***PROFILES
      IF (IWR.EQ.0) THEN
C       ***MONTE CARLO DATA  COLLECTION
        IF (ABS(TT-PR(1)).LT.1.E-6) THEN
          DO 23 J=1,NTIMES-1
            PR(J) = PR(J+1)
 23       CONTINUE
          NTIMES = NTIMES - 1
C         ***COLLECT DATA FOR MONTE CARLO SIMULATION
          MNO = MNO + 1
          ZMO(MNO) = ZZ(1,1)
          CALL ZMASS (TT,ZZ(1,1),PS(2),NEQ,0,ZZ,ND,OMASS,CMASS)
          OILM = OMASS*AC
        END IF
      ELSE IF (IWR.EQ.1) THEN
        IF (NTIMES.GT.0.AND.ABS(TT-PR(1)).LT.8.E-5) THEN
          DO 24 J=1,NTIMES-1
            PR(J) = PR(J+1)
 24       CONTINUE
          NTIMES = NTIMES - 1
          CALL ZMASS (TT,ZZ(1,1),PS(2),NEQ,1,ZZ,ND,OMASS,CMASS)
        END IF
C
C
C       ***GLOBAL CONSTITUENT MASS CONSERVATION
C
C       ***WRITE OUT OIL FRONT LOCATIONS UNTIL OIL LENS FORMS
        OMASS = 0.
        CMASS = 0.
C
C
C
        IF (VIN2.EQ.0.) THEN
C
C         ***PRINT ONLY WHEN THE INTERVAL BETWEEN PRINTED STEPS
C         ***IS AT LEAST DTPR
        IF (T2+DTPR.LT.PT(2).OR.ABS(PT(2)-TM).LE.1.E-5) THEN
C
C           ***MASS BALANCE
            T2 = PT(2)
            CALL ZMASS (TT,ZZ(1,1),PS(2),NEQ,0,ZZ,ND,OMASS,CMASS)
            IF (CMASS.LT.1.E-10.AND.OMASS.LE.1.E-10) CALL COUNT
C
C           ***ONLY PRINT IF 0.1% CHANGE IN FRONT POSITION OR MORE
            ZZ1 = PZ(2)
            ISATP = 1
            DNM = 1.
            IF (OZZ1.GT.0.) DNM = OZZ1
            IF (ABS((ZZ1-OZZ1)/DNM).LT.0.001) THEN
C             ***DO NOT PRINT
              ISATP = 0
            END IF
C
C
            ZZ2 = ZC(1,2)
            ZZ3 = DPA
            IF (NEQ(3).GT.0) ZZ3 = ZC(NEQ(3),2)
            CCC = CC(1,2)*XXK1
            IF (OMASS.LE.1.E-10) ZZ1 = 0.
            IF (XXK3.GT.0.AND.ZV(2).GT.ZZ3) ZZ3 = ZV(2)
            IF (CMASS.LT.1.E-10) THEN
              ZZ2 = 0.
              ZZ3 = 0.
              CCC = 0.
            END IF
C
C
C
            ICONP = 1
            DNM = 1.
            IF (OZZ2.GT.0.0) DNM = OZZ2
C           ***ONLY PRINT IF 0.1% CHANGE IN FRONT POSITION OR MORE
            IF (ABS((ZZ2-OZZ2)/DNM).LT.0.001) THEN
C             ***DO NOT PRINT
              ICONP = 0
            END IF
C
C           ***DO NOT PRINT DEPTHS GREATER THAN THE TOP
C           ***OF THE OILENS OR THE AQUIFER
            ZLENS = WTABLE-ZZ(2,4)-FRING3
            IF (ZC(1,2).GT.ZLENS) THEN
              ICONP = 0
            END IF
C
```

```
C           ***CONVERT RUNOFF VOLUME/AREA TO TOTAL MASS
            RO = ZZ(1,6)*AC*PRHO*DMKG/DVSV
C
            IF (ISATP.EQ.1) THEN
              WRITE (15,9300) NS,TT,ZZ1,PS(2),PF(2),RO,OMASS*AC,CHS(2)
C    *         CHU,FSPD,CHD
            END IF
C
C
            IF (COINI.GT.0..AND.ICONP.EQ.1) THEN
              WRITE (11,9300) NS,TT,ZZ2,ZZ3,CCC,CMASS*AC
            END IF
C
            OZZ1 = ZZ1
            OZZ2 = ZZ2
C
          END IF
        END IF
      END IF
C
C
C     ***HISTORIES
      IF (IHIST.NE.0.AND.IWR.EQ.1) CALL HIST (ZZ,Z1,ND,NEQ,TT)
C
C
C     ***SR CHK PART 3.) ROLL VALUES FOR NEXT EVALUATION
      PT(1) = PT(2)
      PS(1) = PS(2)
      PZ(1) = PZ(2)
      PF(1) = PF(2)
      PRO(1) = PRO(2)
      ZLN(1) = ZLN(2)
      VIN1  = VIN2
      CIN1  = CIN2
      DO 2 I=1,NEQ(2)
        SI(I,1) = SI(I,2)
        ZI(I,1) = ZI(I,2)
 2    CONTINUE
      DO 4 I=1,NEQ(3)
        CC(I,1) = CC(I,2)
        ZC(I,1) = ZC(I,2)
 4    CONTINUE
      CHS(1) = CHS(2)
      IF (XXK3.GT.0..AND.TT.GT.TPE) THEN
        ZV(1) = ZV(2)
        CV(1) = CV(2)
        XJV(1) = XJV(2)
        CVF(1) = CVF(2)
      END IF
C
C
C     ***WRITE SCREEN MESSAG AT THE END OF EACH YEARS SIMULATION
      CALL YWRITE (TT)
C
C     ***ADJUST THE NUMBER OF OIL CHARACTERISTICS
      IF (NEQ(2).EQ.0) RETURN
      IF (ZZ(NEQ(2)-1,2).GT.ZZ(1,1)) NEQ(2) = NEQ(2) - 1
C
C
C
      RETURN
 9300 FORMAT (1X,I5,4F10.4,2G10.4,f10.4,3F10.4)
 9310 FORMAT (1X,I5,60X,5F10.4)
      END
```

```
      SUBROUTINE OEQS (TT,ZZ,F,ND,NEQ)
C     *******************************************************
C     *
C     *  OEQS FORMS THE RIGHT HAND SIDES OF THE ORDINARY
C     *  DIFFERENTIALEQATIONS FOR THE OIL PHASE IN KOPT
C     *
C     *  INPUT ARGUMENTS.....................................
C     *  TT      = TIME
C     *  ZZ(I,J) = DEPTH FOR EQUATION I OF GROUP J
C     *  ND      = ROW DIMENSION OF ZZ
C     *  NEQ(J)  = ARRAY OF NUMBER OF EQUATIONS FOR EACH GROUP
C     *
C     *  OUTPUT ARGUMENTS....................................
C     *  F(I,J) = FUNCTION VALUE FOR EQUATION I OF GROUP J
C     *  NEQ(J) = NUMBER OF EQUATIONS PER GROUP
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES:  BISEC(GAHS),ENTROP,INCAP,INQUAD,OIL,
C     *    PKE,SLOPE
C     *  THIS ROUTINE IS CALLED BY:  EQS
C     *******************************************************
      DIMENSION F(ND,*),NEQ(*),ZZ(ND,*)
      COMMON /COUN/ NUM(100)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /CPSA/ HOE,HWE,OSIG,PHIF,WSIG,XLAMBO
      COMMON /ENTR/ CHU,CHD,FSPD
      COMMON /FLAG/ ICONC,IKOPT,ILENS,ITSGP
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /HYDI/ HDC1,HDC2,HDC3,HDC4
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /MATR/ ETA,OMSPR,OMSWR,SAR,SR,SWR
      COMMON /MULL/ VPA
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      EXTERNAL GAHS
      NUM(29) = NUM(29) + 1
      SPF = 0.
C
C
      IF (NUM(29).EQ.1) THEN
        Z2 = 1.0 + DPA
      END IF
C
C     ***CHARACTERISTICS FOR DEGRADING OILS
      DO 10 I=1,NEQ(2)
        F(I,2) = 0.
        SI(I,2) = SPR
        ZI(I,2) = DPA
        IF (TT.LT.T0(I)) GO TO 10
        XXX = 1.
        IF (TT.GT.OLAG) THEN
          TAU = AMAX1(OLAG,T0(I))
          XXX = EXP(-DLO*(TT-TAU))
        END IF
        SP  = S0(I)*XXX
        SI(I,2) = SP
        ZI(I,2) = ZZ(I,2)
        CALL SLOPE (SP,SWMAX,SL)
        F(I,2) = SL/ETA
 10   CONTINUE
C
C     ***FOR REFERENCE, DETERMINE THE LOCATION OT THE DRAINAGE
C     ***WAVE BEGINNING AT TIME = TPE, FOR NONDEGRADING KINEMATIC
C     ***FLOWS  AT EARLY (LESS THAN TPE) TIMES ZS IS NEGATIVE
      ZS = DPA + (TT-TPE)*DZPDDT
C
C     ***THE OIL FRONT
      F(1,1) = 0.
      PRO(2) = 0.
      IQF = 1
      IF (NEQ(2).EQ.0) THEN
C       ***NON-DEGRADING AND NON-DISSOLVING OILS
C       ***AND LOADING FOR ALL CASES
        IF (IGA.EQ.1) THEN
          CHS(2) = HS
          IF (IAT.EQ.4) CHS(2) = ZZ(3,1)
          IF (TT.GT.TPE) CHS(2) = 0.
          IF (CHS(2).LE.0..AND.NEQ(1).EQ.3) CHS(2) = 1.E-6
        END IF
C
        IKENE= 0
C       ***IF TEST FOR THE KINEMATIC MODEL
        IF (IKOPT.EQ.1.AND.ZS.LT.ZZ(1,1).OR.CHS(2).GT.0.) IKENE=1
C       ***IF TEST FOR MULL'S RECTANGULAR PROFILE MODEL
        IF (IKOPT.EQ.2.AND.VPA.LE.0.) IKENE=1
C
        IF (IKENE.EQ.1) THEN
C          ***CONSTANT MAXIMUM OIL SATURATION
          SPF = SPMAX
          IF (IGA.EQ.1.AND.CHS(2).GT.0.) THEN
C            ***PRESSURIZED CONDITIONS
            IF (IAT.EQ.3.OR.(IAT.EQ.4.AND.PT(2).LE.TPEM)) THEN
C
C              ***ANALYTIC SOLUTION FOR FRONT WITH CONSTANT DEPTH
C              ***PONDING
              ISE = ISE - 1
              Z1 = DPA
```

```
C
            CALL BISEC(GAHS,Z1,Z2,1.E-6,175,TT-TPB,XX,ZZ(1,1),DT,IE)
C
            IF (TT-TPB.LE.0.) ZZ(1,1) = DPA
            IF (ZZ(1,1).GE.0.15*Z2) Z2 = 5.*Z2
C
            IF (IE.EQ.-1.AND.PT(2).GT.TPB)
     *       STOP 'GREEN-AMPT MODEL FAILED'
C
            IQF = 0
C
          ELSE IF (IAT.EQ.4) THEN
C
C            ***NUMERICAL SOLUTION FOR VARIABLE HEAD PONDING
            CONTINUE
C
          END IF
C
C          ***FLUX AT OIL FRONT
          IF (ZZ(1,1).GT.DPA) THEN
C          IF (ZZ(1,1).GT.0.0001) THEN
C
            QPF = XKPMX*(1.+(CHS(2)+PHIF)/(ZZ(1,1)-DPA))
C
          ELSE
C
C            ***FIRST STEP
            if (zz(1,1)-dpa.gt.0.) then
              qpf = xkpmx*(1.+(chs(2)+phif)/(zz(1,1)-dpa))
            else
              QPF = XKPMX*(1.+(CHS(2)+PHIF))/0.0001
              QPF = 100.*XKPMX
            end if
          END IF
C
C
        ELSE
C
C           ***DETERMINE THE INFILTRATION CAPACITY (OIL FLUX--QPF)
C           ***AND OIL RUNOFF--PRO(2)
          CALL INCAP (QP,XKPMX,ZZ(1,1),DPA,PHIF,TPE,PT(2),QGA,QPF,
     1      PRO(2))
        END IF
C
C
      ELSE IF (IKENE.EQ.0) THEN
C
        IF (IKOPT.EQ.1) THEN
C          ***GRAVITY DRAINAGE (KINEMATIC MODEL)
          SL = (ZZ(1,1)-DPA)/(TT-TPE)
          CALL INQUAD (WP,50,SPR,1.-SWMAX,SL*ETA,SPF)
        ELSE IF (IKOPT.EQ.2) THEN
C          ***MULL'S RECTANGULAR PROFILE MODEL
          CALL OIL (ZZ(1,1),TT,SWMAX,NEQ(2),SPF)
          SPF = VPA/ZZ(1,1)/ETA
          END IF
C
C
          CALL PKE (SPF,SWMAX,QPF)
C
        END IF
C
      ELSE IF (NEQ(2).NE.0) THEN
C
C       ***DEGRADING OR DISSOLVING OILS
        IF (IKOPT.EQ.2) STOP 'MULL MODEL ONLY FOR NON-DISSOLVING NAPL'
        CHS(2) = 0.
        SL = 0.
        S1 = SPMAX
        LZ = NEQ(2)
C
        DO 20 I=NEQ(2),2,-1
          IF (ZI(I,2).GE.ZZ(1,1).AND.ZZ(1,1).GT.ZI(I-1,2)) THEN
            LZ = I
            GO TO 21
          END IF
 20     CONTINUE
 21     CONTINUE
C
        IF (TT.GT.T0(LZ-1)) S1 = SI(LZ-1,2)
        IF (ZI(LZ,2).NE.ZI(LZ-1,2))
     1  SL = (S1-SI(LZ,2))/(ZI(LZ-1,2)-ZI(LZ,2))
        SPF = SI(LZ,2) + (ZZ(1,1)-ZI(LZ,2))*SL
        QPF = 0.
        IF (SPF.GT.SPR) CALL PKE (SPF,SWMAX,QPF)
C
      END IF
C
C
C
        IF (IAT.EQ.2.AND.QP.EQ.0.) THEN
          XXX = 1.
          IF (TT.GT.OLAG) XXX = EXP(-DLO*(TT-OLAG))
          SPF = SPMAX*XXX
        END IF
C
C
C       ***OIL PHASE PROPERTIES AT THE OIL FRONT
        PS(2) = SPF
        PZ(2) = ZZ(1,1)
        PF(2) = QPF
        FSPD  = QPF/SPF/ETA
C
C       ***CHARACTERISTIC SPEEDS  FOR ENTROPY CONDITION EVAULATION
C       ***(DOWNSTREAM OIL SATURATION = SPI = 0.0)
        SPI = 0.0
        CALL ENTROP (SPF,SPI,IKOPT,ETA,SWMAX,IGA,CHS(2),GA3,ZZ(1,1),
     *CHU,CHD)
C
C       ***INTEGRATION OF THE VOLUME FLUX APPLIED AT BOUNDARY DURING OIL
```

```
C     ***EVENT FOR LAND APPLICATION SIMULATION, IAT = 2, F(2,6) WILL
C     ***EQUAL ZERO
      F(2,6) = 0.0
      IF (PT(2).LE.TPE) F(2,6) = QPF
C
C     ***RATE OF CHANGE OF OIL PONDING AT SURFACE (IAT=4)
      F(3,1) = 0.
      IF (PT(2).GE.TPEM.AND.IAT.EQ.4.AND.IQF.EQ.1) F(3,1) = -QPF
C
C     ***THE SPEED OF THE OIL FRONT
      IF (IQF.NE.1.OR.QPF.LE.0.) RETURN
C
C     ***OIL FRONT SPEED AS MODIFIED BY OIL PHASE DISSOLUTION (I-3-90)
      IKENE = 0
C     ***KINEMATIC MODEL FOR NON-DISSOLVING NAPL
      ZADH = DPL + HDC1*(TT-TPB)
      IF (IKOPT.EQ.1.AND.ZZ(1,1).LT.ZADH.OR.TT.GT.TPE) IKENE = 1
C     ***MULL'S RECTANGULAR PROFILE MODEL
      IF (IKOPT.EQ.2) IKENE = 1
C
C     ***NAPL FRONT SPEED
      IF (IKENE.EQ.1) THEN
C       ***OIL FRONT SHALLOWER THAN THE AQUEOUS DISSOLVED PHASE
        F(1,1) = QPF/SPF/ETA
      ELSE
C       ***OIL FRONT AT SAME DEPTH AS AQUEOUS DISSOLVED PHASE
        F(1,1) = (QPF*PRHO + HDC2)/(ETA*SPF*PRHO + HDC3)
      END IF
C
C     ***INTEGRATION OF THE RUNOFF VOLUME FLUX PER UNIT AREA
C     ***TO DETERMINE CUMULATIVE RUNOFF
      F(1,6) = PRO(2)
C
      RETURN
      END
```

```
      SUBROUTINE CEQS (TT,ZZ,F,ND,NEQ)
C     ************************************************************
C     *
C     *  CEQS FORMS THE RIGHT HAND SIDES OF THE ORDINARY DIFFERENTIAL
C     *  EQATIONS FOR THE CONSTITUENT EQUATIONS
C     *
C     *  INPUT ARGUMENTS.....................................
C     *  TT      = TIME
C     *  ZZ(I,J) = DEPTH FOR EQUATION I OF GROUP J
C     *  ND      = ROW DIMENSION OF ZZ
C     *  NEQ(J)  = ARRAY OF NUMBER OF EQUATIONS FOR EACH GROUP
C     *
C     *  OUTPUT ARGUMENTS.....................................
C     *  F(I,J) = FUNCTION VALUE FOR EQUATION I OF GROUP J
C     *  NEQ(J) = NUMBER OF EQUATIONS PER GROUP
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES:  CVEL,OIL,RD
C     *  THIS ROUTINE IS CALLED BY:  EQS
C     *
C     *  REVISIONS:
C     *  7-28-94   REMOVAL OF VOLATILIZATION MODEL
C     *
C     ************************************************************
      DIMENSION F(ND,*),NEQ(*),ZZ(ND,*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /COUN/ NUM(100)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /MATR/ ETA,OMSPR,OMSWR,SAR,SR,SWR
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(30) = NUM(30) + 1
C
C     ***DISSOLVED CONSTITUENTS
C     ***CHARACTERISTICS
      DO 50 I=1,NEQ(3)
        F(I,3) = 0.
        ZC(I,2) = DPA
        IF (TT.LT.TC(I)) GO TO 49
        CALL OIL (ZZ(I,3),TT,SWMAX,NEQ(2),SO)
        IF (ABS(TT-TPE).LE.0.0001.AND.I.EQ.NEQ(3)) SO = SPR
        CALL CVEL (TT,SO,SWMAX,F(I,3))
        XXX = 1.
        YYY = 1.
        IF (DLC.GT.0..AND.TT.GE.CLAG) XXX = EXP(-DLC*(PT(2)-PT(1)))
        IF (DLO.GT.0..AND.SO.GT.0..AND.TT.GT.OLAG) THEN
C         ***CORRECTION FOR OIL DECAY
          SO2 = SO/EXP(  -DLO*(PT(2)-PT(1))  )
          YYY = RD(SO2,SWMAX)/RD(SO,SWMAX)
        END IF
        CC(I,2) = CC(I,1)*XXX*YYY
 49     CONTINUE
        ZC(I,2) = ZZ(I,3)
 50   CONTINUE
      F(2,1) = 0.
C
C     ***VOLATILIZATION
C     ***VOLATILIZATION MODEL REMOVED FROM KOPT BECAUSE VOLATILIZATION
C     ***IS NOT INCLUDED IN THE CURRENT VERSION OF OILENS
C
C
      RETURN
      END
```

```
      SUBROUTINE GAHS (ZZZ,XY,DT)
C     ************************************************************
C     *
C     * GAHS   DETERMINES THE GREEN-AMPT MODEL SOLUTION FOR A GIVEN
C     *        DEPTH.  THIS ROUTINE IS USED WITH A BISECTION SEARCH
C     *
C     *
C     *
C     * INPUT ARGUMENTS.........................................
C     *   ZZZ    DEPTH (METERS)
C     * OUTPUT ARGUMENTS........................................
C     *   XY     DUMMY ARGUMENT REQUIRED BY BINARY
C     *   DT     TIME INCREMENT ASSOCIATED WITH ZZZ (DAYS)
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE/DOUBLE  PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  BISEC(OEQS)
C     *
C     ************************************************************

      DOUBLE PRECISION ZZ,GGA1,GGA2,GGA3,GGA4
      DOUBLE PRECISION XT1,XT2,XT3
      COMMON /COUN/ NUM(100)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /GAE2/ GGA1,GGA2,GGA3,GGA4
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      NUM(5) = NUM(5) + 1
C
C
C     ***ANALYTIC SOLUTION OF THE GREEN AMPT MODEL WITH CONSTANT
C     ***PONDING DEPTH
      XY = 1.
      ZZ = DBLE(ZZZ-DPA)
      DT = GA1*SNGL(ZZ - DBLE(GA3)*DLOG(DBLE(GA3)+ZZ) + DBLE(GA4))
      XT1 = GGA3*DLOG(GGA3+ZZ)
      XT2 = -XT1 + GGA4
      XT3 = ZZ + XT2
      XT4 = GA1*SNGL(XT3)
      DT = GA1*SNGL(ZZ - GGA3*DLOG(GGA3+ZZ) + GGA4)
C
C
      RETURN
      END
```

```
      SUBROUTINE OILENS (TT,ZZ,F,ND,NEQ)
C     ************************************************************
C     THE PROGRAM OILENS MODELS THE MIGRATION OF A CONTAMINANT AND AN
C     IMMISCIBLE HYDROCARBON WHICH IS LESS DENSE THAN WATER AS THEY
C     SPREAD ALONG THE CAPILLARY FRINGE.  THE PROCESSES OF DISSOLUTION,
C     BIODEGRADATION, AND ISOLATION AT RESIDUAL SATURATIONS ARE
C     INCLUDED.  THE ALGORITHM USES THE DUPUIT ASSUMPTIONS FOR RADIAL
C     FLOW AND ASSUMES VERTICAL EQUILIBRIUM (GHYBEN-HERZBERG) PRESSURES.
C     THE VOLUME BALANCE EQUATIONS ARE SOLVED USING A FOURTH-ORDER
C     RUNGE-KUTTA SCHEME.
C
C                    THE PROGRAM WAS WRITTEN BY
C             RANDALL J. CHARBENEAU AND SUSAN R. SHULTZ
C           ENVIRONMENTAL AND WATER RESOURCES ENGINEERING
C
C     REFERENCE: CHARBENEAU AND SHULTZ, HYDROCARBON SOURCE TERM
C                CHARACTERIZATION FOR GROUNDWATER MODELS,
C                SUBMITTED TO WATER RESOURCES RESEARCH, APRIL 1989
C
C     MODIFIED FOR INCLUSION IN KOPT BY
C     JIM WEAVER
C     ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     ADA, OKLAHOMA 74820
C
C     ANSI STANDARD X3.9-1978 FORTRAN 77
C
C     XII-8-1988   REWRITTEN FOR INCLUSION IN KOPT
C     V-25-1989    LENS PROFILING ADDED BY JIM WEAVER
C     IIX-15-1989  CONTAMINANT MOVING INDEPENDENTLY IN OIL LENS
C
C     (EQUATION NUMBERS IN COMMENT LINES REFER TO ABOVE REFERENCE)
C
C     ************************************************************
      DIMENSION F(ND,*),NEQ(*),ZZ(ND,*)
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /COUN/ NUM(100)
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LNSS/ ILSTOP
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
      NUM(38) = NUM(38) + 1
C
C
C
      F(1,4) = 0.
      TRAV = (TRA + ALPHA*TRB)/AP1
      ZLENS = WTABLE-ZZ(2,4)-FRING3
      ZLN(2) = ZZ(2,4)
C
C     ***FLUX INTO LENS OR CAPILLARY FRINGE
      IF (ZZ(1,1).LT.ZLENS) RETURN
      CALL OFLUX (TT,ZLENS,NEQ,QINL,CINL,CWINL)
C
C     ***INTEGRATE FLUX INTO LENS
      F(1,4) = QINL
      IF (NEQ(4).LE.1) RETURN
C
C
C
C
C     ***SET UP RIGHT HAND SIDE OF O.D.E.S
C     ***F(2,4) IS THE CHANGE OF THE MOUND HEIGHT WITH TIME
C     ***F(3,4) IS THE CHANGE OF THE MOUND RADIUS WITH TIME
C
C
      IF (ZZ(2,4).LT.0.) ZZ(2,4) = 0.
C
C
C
C
C     ***TEMPORARY DEBUG TRAP
      IF (PT(2).GE.50.) THEN
        CONTINUE
      END IF
C
C     ***RADIAL FLUX OF OIL OUT FROM CENTRAL CYLINDER (EQ. 6)
C     QRAD = PI*PERMO*ZZ(2,4)*(AP1*ZZ(2,4)+FRING3)/ALOG(ZZ(3,4)/RADI)
C
C
C
C     ***CORRECTION FOR WHEN THE RADIAL FLUX IS GREATER THAN
C     ***THE INFLUX, AND THE SOLUTION IS JUST STARTING
      IF (ZZ(3,4).GT.RMF*RADI) THEN
C       ***AFTER LENS IS ESTABLISHED GRADIENT IS COMPUTED
C       ***AT THE SOURCE RADIUS (RADI)
        DHDR = -0.5*ZZ(2,4)/RADI/ALOG(ZZ(3,4)/RADI)
      ELSE
C       ***THE GRADIENT IS APPROXIMATED AT 0.5*(RADI+RMF*RADI)
C       ***FOR THE FIRST STEP
        R1 = RADI
        R  = 0.5*(RADI + ZZ(3,4))
        RT = ZZ(3,4)
        DHDR = -0.5*ZZ(2,4)/R/ALOG(RT/R1)
        DHDR = DHDR*(   (ALOG(RT/R)/ALOG(RT/R1))**(-0.5)   )
      END IF
C
C     ***RADIAL FLUX OF OIL OUT FROM CENTRAL CYLINDER (EQ. 6)
      QRAD = - PERMO*AC*(ZZ(2,4)*AP1+FRING3)*DHDR
C
C
C     ***CONSTANTS FOR FURTHER CALCULATION
      EB  = ALOG((ZZ(3,4)/RADI)**2)
```

```
      IF (EB.LE.0.) STOP 'OILENS SINGULARITY ENCOUNTERED, INCREASE RMF'
      ARG = SQRT(EB)
      A   = ERF(ARG)
      B   = (1.+ALPHA)/ARG
C
C
C
C     ***PARTIAL OF LENS VOLUME WRT HEIGHT (EQ. 12)
      DVDH = PI*THETO*A*B*SPI4*ZZ(3,4)*ZZ(3,4)
C     ***PARTIAL OF LENS VOLUME WRT RADIUS (EQ. 13)
      DVDR = 2.*PI*THETO*FRING3*ZZ(3,4) + PI*THETO*B*ZZ(2,4)*ZZ(3,4)*
     1       (SQRTPI*A*(1.-0.5/EB) + (RADI/ZZ(3,4))**2/ARG)
C
C
C
C
C     ***FLUXES TO CENTRAL CYLINDER OF LENS
C     ***TOTAL QLOSSH IS APPORTIONED TO CENTRAL CYLINDER
C     ***BY AREA WEIGHTING BASED ON THE CYLINDER RADIUS (RADI)
C     ***AND THE OUTER RADIUS OF THE LENS (ZZ(3,4))
      TERM1 = QRAD + QLOSSH*(RADI/ZZ(3,4))**2
      DTHETO = THETO
C
C
C     ***ADDITIONAL FLUX COMPONENT DUE TO CHANGE IN HEIGHT OF LENS
      IF (ZLN(1).LE.ZLN(2)) THEN
C       ***AVERAGE OIL SATURATION FOR GROWING LENS
c       CALL ZMASS (PT(2),ZLENS,PS(2),NEQ,0,ZZ,ND,OMASS,CMASS)
c       OVOL = OMASS*DVSV/PRHO/DMKG/ETA
c       SBAR = OVOL/ZLENS
C       ***TEMPORARY SBAR CALCULATION
        CALL OIL (ZLENS,PT(2),PS(2),NEQ(2),SO)
        SBAR = SO
      ELSE
C       ***AVERAGE OIL SATURATION FOR DECAYING LENS
        CALL OIL (ZLENS,PT(2),PS(2),NEQ(2),SO)
        SBAR = SO
      END IF
C     IF (QINL.LE.TERM1) THEN
C       DTHETO = THETO-Trav
C     END IF
C
C
C
C     ***CHANGE IN LENS HEIGHT WITH TIME
      F1 = 1.-ETA*SO/AP1/DTHETO
      F1 = 1.
      F(2,4) = (QINL-TERM1)/AC/AP1/DTHETO/F1
C
C
C
C     ***CHANGE IN LENS RADIUS WITH TIME
      TERM2 = QLOSSH*FRACT + QLOSSR + DVDH*F(2,4)
      F(3,4) = (QINL-TERM2)/DVDR
      IF (TERM2.LT.0.) THEN
        CONTINUE
      END IF
      IF (F(3,4).LT.0.) THEN
        CONTINUE
      END IF
C
C
      RETURN
      END
```

```
      SUBROUTINE CLENS (TT,ZZ,F,ND,NEQ)
C     ******************************************************************
C     *
C     * CLENS--CONSTITUENT LENS EQUATIONS
C     *
C     * CLENS DETERMINES THE MASS FLUX INTO THE LENS OR AQUIFER
C     *
C     * INPUT ARGUMENTS.............................................
C     *   TT      CURRENT SOLUTION TIME
C     *   ZZ(,)   CURRENT VALUES OF SOLUTION VARIABLES
C     *   F(,)    RATES OF CHANGE OF SOLUTION VARIABLES
C     *   ND      ROW DIMENSION OF F(,) AND ZZ(,)
C     *   NEQ()   NUMBER OF EQUATIONS IN EACH GROUP
C     *            (COLUMN OF F(,) AND ZZ(,))
C     * OUTPUT ARGUMENTS............................................
C     *   F(1,5)  MASS FLOW RATE INTO LENS OR AQUIFER
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES: OFLUX,LFLUX
C     *
C     ******************************************************************
      DIMENSION F(ND,*),NEQ(*),ZZ(ND,*)
      COMMON /COUN/ NUM(100)
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(44) = NUM(44) + 1
C
C
C
C     ***SET UP RIGHT HAND SIDE OF O.D.E.S
C     ***F(1,5) = MASS FLUX TO OIL LENS OR WATER TABLE
C
      ZLENS = WTABLE-ZZ(2,4)-FRING3
C
C
C
C
C     ***RETURN IF CONSTITUENT FRONT HASN'T REACHED LENS
      IF (ZZ(1,3).LE.ZLENS) RETURN
C
C
C
C     ***VOLUME FLUX TO LENS AND CONTAMINANT CONCENTRATION IN OIL PHASE
      CALL OFLUX (TT,ZLENS,NEQ,QINL,CINL,CWINL)
C
C     ***CHANGE IN CONTAMINANT MASS FLUX WITH TIME
C     ***(INTEGRATED TO GET CUMULATIVE MASS OF CONTAMINANT IN LENS)
C     ***CONTRIBUTION ONLY WHEN CONSTITUENT HAS REACHED TOP OF OIL LENS
C     ***OR THE WATER TABLE
C     ***THE MASS FLUX IS THE SUM OF THE FLUX DUE TO THE NAPL
C     ***AND THE WATER
      F(1,5) = QINL*CINL + QW*AC*CWINL
C
C
C     ***SPEED OF THE CONTAMINANT FRONT IN THE OIL LENS
C     ***THE SPEED IS LIMITED TO THE MAXIMUM SPEED OF THE RADIUS
      IF (ABS(ZZ(2,5)).GT.ABS(ZZ(3,4))) ZZ(2,5) = ZZ(3,4)
C
      QR = 0.
      IF (ZZ(2,5).GT.0.) THEN
C       ***DETERMINE THE FLUX IN THE LENS IF THE CONTAMINANT RADIUS
C       ***IS GREATER THAN ZERO
        CALL LFLUX (ZZ(2,5),ZZ(3,4),ZZ(2,4),DHDR,QR,Q)
      END IF
C
C     ***BULK PARTITION COEFFICIENT WHICH ACCOUNTS FOR THE
C     ***ACTIVELY SPREADING LENS, AND THE TRAPPING ABOVE AND
C     ***BELOW THE WATER TABLE
C     BPTC = VTRESA*(ETA*(XXK1*SWMAX + SPR) + XXK2)
C     BPTC = BPTC + VCLEN*(ETA*(XXK1*(1.-SOLENS) + SOLENS) + XXK2)
C     BPTC = BPTC + VTRESB*(ETA*(XXK1*(1.-SPRB) + SPRB) + XXK2)

      F(2,5) = QR/BW
      F(2,5) = 0.
      IF (F(3,4).GT.0.0) THEN
        F(2,5) = F(3,4)
      END IF
C
C     ***LIMIT THE SPEED OF THE CONTAMINANT TO THE MAX SPEED OF THE LENS
      IF (ABS(F(2,5)).GT.ABS(F(3,4))) F(2,5) = F(3,4)
C
C
      RETURN
      END
```

```
      SUBROUTINE HMASS (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ****************************************************************
C     *
C     * HMASS--HYDROCARBON MASS BALANCE CALCULATIONS
C     *
C     * HMASS CALCULATES THE HYDROCARBONE PHASE MASS BALANCES
C     *
C     * INPUT ARGUMENTS.....................................
C     *   TT      = SOLUTION TIME JUST OBTAINED
C     *   ZZ(I,J) = DEPTH FOR EQUATION I GROUP J
C     *   Z1(I,J) = PREVIOUS DEPTH FOR EQUATION I GROUP J
C     *   F(I,J)  = CURRENT FUNCTION VALUES
C     *   ND      = ROW DIMENSION OF ZZ(),Z1(), & F()
C     *   NEQ(J)  = NUMBER OF EQUATION IN EQCH GROUP
C     *   DMAX    = MAXIMUM TIME STEP
C     *   DT      = CURRENT TIME STEP
C     *   NS      = NUMBER OF STEPS
C     *   IRP     = 1 IF MRKF12 ALREADY HAS REJECTED THE STEP
C     *
C     * OUTPUT ARGUMENTS....................................
C     *   NEQ(J)  = ADJUSTED NUMBER OF EQUATIONS
C     *   DMAX    = ADJUSTED MAX. TIME STEP
C     *   DT      = NEW TIME STEP FOR ACCEPTED STEPS
C     *   DTR()   = SERIES OF PROPOSED REDUCED TIME STEPS
C     *   IC      = 1 TO END SIMULATION
C     *   IRR()   = CODES CORESPONDING TO VARIOUS REDUCED STEPS
C     *   IR      = 1 TO REJECT STEP
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES: VOLT
C     * CALLED BY LCHK2
C     *
C     ****************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      CHARACTER NT*15
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GLMB/ CCAL,CCMD,CCRA,CCRB,CHLENS,CHLOSS,CHRES
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /NAME/ NT(15)
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RESV/ VTRESA,VTRESB
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /STOI/ BDFC,BDFH,SOLC,SOLH,STOICC,STOICH
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      DATA IWRITE,VLOSSH,RINT1 /0,0.,.8740192/

      NUM(45) = NUM(45) + 1
      IF (NUM(45).EQ.1) THEN
        IWRITE = 0
        VTLEN  = 0.
        VHLOSS = 0.
        VLOSSH = 0.
        VTRES  = 0.
        RINT1  = 0.8740192
        RTOLD  = RADI
        IRTT = 0
        RT1 = 0.
        HLR = 0.
        RLR = 0.
        T2 = 0.
      END IF

      ***DEPTH WEIGHTED AVERAGE RESIDUAL OIL CONTENT
      TRAV = (TRA + ALPHA*TRB)/AP1

      ***VOLUME BALANCE COMPUTATIONS FOR THE LENS
      HO = ZZ(2,4)
      RT = ZZ(3,4)

      ***CALCULATE THE RESIDUAL HYDROCARBON LEFT AS THE MOUND DECAYS
      ***AFTER THE SOURCE HAS STOPPED OR SLOWED DOWN
      DELTV = 0.
      VSRCE = 0.
      VA1 = 0.
      VB1 = 0.
      IF (F(2,4).LT.0.AND.RHMAX.GT.0..AND.HLR.GT.HO) THEN
        G = (HLR*HLR/HO/HO)*ALOG(RT/RADI)/ALOG(RLR/RADI)
        RST = ((RLR**G)/RT)**(1./(G-1.))
        CALL VOLT (RST,RLR,HLR,VA1,VB1,VT1)
        CALL VOLT (RST,RT,HO,VA2,VB2,VT2)
        DELTV = VT1 - VT2
        DELVO = (VA1-VA2)*TRA + (VB1-VB2)*TRB
        ***VOLUME WEIGHTED AVERAGE RESIDUAL OIL CONTENT
        IF (DELVO.GT.0..AND.DELTV.GT.0.) TRAV = DELVO/DELTV
        ***FIND RESIDUAL OIL VOLUME ABOVE CENTRAL CYLINDAR OF LENS
        VSRCE = (HLR-HO)*AC*TRA
        VSRCE = 0.
        ***SUBTRACT OFF RESIDUAL VOLUME IN SOURCE AREA ABOVE CYLINDAR
        DELVO = DELVO - VSRCE
        QLOSSR = DELVO/DT
      END IF
```

```
      HLR = HO
      RLR = RT
C
C     ***VOLUME BALANCE COMPUTATIONS
C     ***TOTAL VOLUME OF LENS
      CALL VOLT (RT,RT,HO,VAA1,VBB1,VTLEN)
C     ***CUMULATIVE OIL VOLUME ADDED TO LENS
      VIN2  = ZZ(1,4)
C
C     ***HYDROCARBON DISSOLUTION INTO GROUNDWATER
      X = SQRT(2.*RT*VDISP/PI)
C     ***FLXGWC,FLXGWH
C     ***ARE EXPRESSED IN THE MIXED UNITS OF:
C     ***CONCENTRATION MASS*SIMULATION VOLUME/CONCENTRATION VOLUME/TIME
C     ***ON V-24-89 THIS IS MILIGRAM*CUBIC METER/LITER/DAY
C     ***CONVERTED ON OUTPUT TO KILGRAMS/DAY OR KILOGRAMS VIA UC2
      VF1 = 4.*RT*VEL*X*RINT1
      VF2 = QW*PI*RT*RT
C
C     ***HYDROCARBOM DISSOLUTION FROM THE LENS
      FLXGWH = (VF1 + VF2)*SOLH
      RMDISH = DT*(FLXGWH+FLXH1)*0.5
      T1 = TT-DT
      DF = FLXGWH-FLXH1
      RMDISH = FLXH1*DT + 0.5*DF*(TT+T1) - DF*T1
      CMDH   = CMDH + RMDISH
C
C     CALCULATE BIODEGRADATION FROM OXYGEN DIFFUSION INTO THE LENS
      FLXO2 = COXY*VF1
      RMOXY = DT*(FLXO2+FLXO1)*0.5
C
C     ***HYDROCARBON DEGRADATION
      RMDEGH = RMOXY*STOICH*BDFH
C     ***TOTAL HYDROCARBON LOSS FROM THE LENS
      RLOSSH = RMDEGH + RMDISH
C
C     ***VOLUME OF HYDROCARBON LOST AND RATE OVER THIS TIME STEP
      VLOSSH = RLOSSH*UC1/PRHO
      QLOSSH = VLOSSH/DT
C     ***TOTAL LENS VOLUME ISOLATED AT RESIDUAL
C     ***LOSSES ARE APPORTIONED TO BOTH THE MOBILE AND THE IMMOBILE OIL
      VTRES  = VTRES + DELTV - (1.-FRACT)*VLOSSH/TRAV
C     VTRES  = VTRES + DELTV
      VTLENS = VTLENS - FRACT*VLOSSH/THETO
      qlossh = qlossh*fract
C
      IF (VA1.GT.0.) THEN
C       ***VADOSE ZONE IMMOBILE BULK VOLUME
        IF (TRA.GT.0.) THEN
          VTRESA = VTRESA + VA1-VA2 - (1.-FRACT)*VLOSSH/TRA
        END IF
C       ***AQUIFER IMMOBILE BULK VOLUME
        IF (TRB.GT.0.) THEN
          VTRESB = VTRESB + VB1-VB2 - (1.-FRACT)*VLOSSH/TRB
        END IF
      END IF
C     ***VOLUME OF HYDROCARBON LOST
      VHLOSS = VHLOSS + VLOSSH
C     ***FRACTION OF HYDRACARBON AT SATURATION ABOVE RESIDUAL
      FRACT = THETO*VTLEN/(THETO*VTLEN + TRAV*VTRES)
C     ***CUMULATIVE MASSES FOR GLOBAL MASS CONSERVATION CALCULATION
      CHLENS = VTLEN*THETO*PRHO*DMKG/DVSV
      CHRES  = (VTRES*TRAV - VSRCE)*PRHO*DMKG/DVSV
      CHRES  = (VTRES*TRAV)*PRHO*DMKG/DVSV
      CHLOSS = VHLOSS*PRHO*DMKG/DVSV
C
C     ***DATA OUTPUT
      IF ((T2+DTPR.LT.TT.OR.ABS(TT-TM).LE.1.E-5)
     *    .AND.ABS(RT-RTOLD)/RTOLD.GT.1.E-5) THEN
        T2 = TT
        IF (IWR.EQ.1.AND.IWRITE.EQ.0) THEN
          IWRITE = 1
        END IF
C       ***E = PERCENT ERROR IN CUMULATIVE OIL VOLUME
        E = 100.*(THETO*VTLEN+VTRES*TRAV+VHLOSS-VIN2)/VIN2
        IF (IWR.EQ.1) THEN
          WRITE (12,9100) NS,T2,HO,RT,THETO*VTLEN,
     *    VTRES*TRAV,VHLOSS,VIN2,E
C         ***DATA OUTPUT FOR PLOT FILE 3
          WRITE (18,9110) T2,RT
          RTOLD = RT
        END IF
      END IF
C
C     ***ROLL VALUES FOR NEXT EVALUATION
      FLXH1 = FLXGWH
      FLXO1 = FLXO2
C
      RETURN
 9100 FORMAT (1X,I5,1X,F8.2,1X,F8.4,1X,6(F8.2,1X))
 9110 FORMAT (4X,2(G10.4,1X))
      END
```

```
      SUBROUTINE CMASS (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ****************************************************************
C     *
C     * CMASS--CONSTITUENT MASS BALANCE CALCULATIONS
C     *
C     * CMASS PERFORMS THE MASS BALANCE CALCULATION FOR THE CONSITUENT
C     *
C     * INPUT ARGUMENTS.....................................
C     *   TT      = SOLUTION TIME JUST OBTAINED
C     *   ZZ(I,J) = DEPTH FOR EQUATION I GROUP J
C     *   Z1(I,J) = PREVIOUS DEPTH FOR EQUATION I GROUP J
C     *   F(I,J)  = CURRENT FUNCTION VALUES
C     *   ND      = ROW DIMENSION OF ZZ(),Z1(), & F()
C     *   NEQ(J)  = NUMBER OF EQUATION IN EQCH GROUP
C     *   DMAX    = MAXIMUM TIME STEP
C     *   DT      = CURRENT TIME STEP
C     *   NS      = NUMBER OF STEPS
C     *   IRP     = 1 IF MRKF12 ALREADY HAS REJECTED THE STEP
C     *
C     * OUTPUT ARGUMENTS.....................................
C     *   NEQ(J)  = ADJUSTED NUMBER OF EQUATIONS
C     *   DMAX    = ADJUSTED MAX. TIME STEP
C     *   DT      = NEW TIME STEP FOR ACCEPTED STEPS
C     *   DTR()   = SERIES OF PROPOSED REDUCED TIME STEPS
C     *   IC      = 1 TO END SIMULATION
C     *   IRR()   = CODES CORESPONDING TO VARIOUS REDUCED STEPS
C     *   IR      = 1 TO REJECT STEP
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES: VOLT
C     * CALLED BY LCHK2
C     *
C     ****************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      CHARACTER NT*15
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GLMB/ CCAL,CCMD,CCRA,CCRB,CHLENS,CHLOSS,CHRES
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /NAME/ NT(15)
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RESV/ VTRESA,VTRESB
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /STOI/ BDFC,BDFH,SOLC,SOLH,STOICC,STOICH
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C     DATA CT1,CWC,DEGMAS,RINT1 /3*0.,.8740192/
C     DATA IWRITE,IMSSC,CMB,CFOL /2*0,2*0./

      NUM(46) = NUM(46) + 1

C
      IF (NUM(46).EQ.1) THEN
        T2 = 0.
        RTOLD = 0.
        IRTT = 0
        RT1 = 0.
        CT1=0.
        CWC=0.
        DEGMAS=0.
        RINT1=.8740192
        IWRITE=0
        IMSSC=0
        CMB=0.
        CFOL=0.
        RMCMAX = 0.
        RMC = 0.
        RMC2 = 0.
      END IF


C     ***CHECK FOR ARRIVAL OF THE CONSTITUENT AT THE WATERTABLE
      ZLENS = WTABLE-ZZ(2,4)-FRING3
      IF (ZZ(1,3).LT.ZLENS) RETURN

C     ***RETURN IF NO DISSOLVED CONSTITUENT
      IF (COINI.LE.0.) RETURN

C     ***CONTAMINANT MASS BALANCES
      HO = ZZ(2,4)
      RT = ZZ(3,4)
      IF (ZZ(2,4).LE.0.) RT = RADI
      RC = ZZ(2,5)

C     ***CUMULATIVE CONTAMINANT MASS ADDED TO LENS
      CMSS = ZZ(1,5)



C     ***EXIT IF NO MASS HASS BEEN ADDED TO THE LENS OR AQUIFER
      IF (CMSS.LE.0.) RETURN
```

```
C                                                                       C
C                                                                       C
C                                                                               RMDISC = DT*(FLXGWC+FLXC1)*0.5
C                                                                       C
C     ***CONSTITUENT MASS ADDED TO LENS ONLY IF IT HAS REACHED          C
C     ***THE TOP OF THE LENS OR THE WATER TABLE                         C
      DELM = 0.                                                         C       ***FIRST ORDER LOSS (HYDROLYSIS) OF CONTAMINANT
      IF (NEQ(5).GE.1) THEN                                                     XXX = 1.
        DELM = ZZ(1,5)-Z1(1,5)                                                  IF (DLC.GT.0..AND.PT(2).GT.CLAG) THEN
C       ***IMSSC USED LATER TO CALCULATE LOSSES ONLY AFTER CONTAMINANT            XXX = EXP(-DLC*(PT(2)-PT(1)))
C       ***HAS REACHED THE LENS                                                 END IF
        IMSSC = 1                                                               FOLOSS = RMC*(1.-XXX)
      END IF                                                            C
C                                                                       C
C                                                                       C
C                                                                       C       ***OXYGEN LIMITED BIODEGRADATION OF CONTAMINANT
C     CONTAMINANT DISSOLUTION INTO GROUNDWATER                          C       ****FROM OXYGEN DIFFUSION INTO THE LENS
      X = SQRT(2.*RC*VDISP/PI)                                                  FLXO2 = COXY*VF1
C     ***FLXGWC,FLXGWH                                                          RMOXY = DT*(FLXO2+FLXO1)*0.5
C     ***ARE EXPRESSED IN THE MIXED UNITS OF:                                   RMDEGC = RMOXY*STOICC*BDFC
C     ***CONCENTRATION MASS*SIMULATION VOLUME/CONCENTRATION VOLUME/TIME         DEGMAS = DEGMAS + RMDEGC
C     ***ON V-24-89 THIS IS MILIGRAM*CUBIC METER/LITER/DAY              C
C     ***CONVERTED ON OUTPUT TO KILGRAMS/DAY OR KILOGRAMS VIA UC2       C
C     ***VF1 = VOLUME FLUX FROM OIL LENS/WATER CONTACT                  C
C     ***VF1 = 0. IF LENS HAS NOT FORMED                                C       ***TOTAL CONTAMINANT LOSSES FROM LENS
C     ***VF2 = VOLUME FLUX FROM INFILTRATING WATER                              RLOSSC = RMDEGC + RMDISC + FOLOSS
      VF1 = 0.                                                                  RMC2 = RMC + DELM - RLOSSC
      IF (ZZ(2,4).GT.0.) VF1 = 4.*RC*VEL*X*RINT1                                IF (RMC.LT.0.0)  RMC = 0.0
      VF2 = QW*PI*RC*RC                                                 C
C                                                                       C
C                                                                       C
C                                                                       C       ***WATER PHASE CONCENTRATION
C                                                                               IF (ZZ(2,4).GT.0..AND.RMC2.GT.0.) THEN
C     ***CONTAMINANT DISSOLUTION FROM THE LENS                          C
      IF (IMSSC.EQ.1) THEN                                              C         ***VOLUME OF ACTIVELY MOVING LENS
C                                                                               CALL VOLT (RC,RT,HO,VA,VB,VCLEN)
C                                                                       C
 10     CONTINUE                                                        C         ***BULK PARTITION COEFFICIENT WHICH ACCOUNTS FOR THE
C                                                                       C         ***ACTIVELY SPREADING LENS, AND THE TRAPPING ABOVE AND
C                                                                       C         ***BELOW THE WATER TABLE
C                                                                               BPTC1A = VTRESA*(ETA*(XXK1*SWMAX + SPR) + XXK2)
C       ***MASS FLUX OF CONTAMINANT TO AQUIFER (FLXGWC)                         BPTC2L = VCLEN*(ETA*(XXK1*(1.-SOLENS) + SOLENS) + XXK2)
C       ***CWC IS THE WATER PHASE CONCENTRATION SO THAT THE                     BPTC3B = VTRESB*(ETA*(XXK1*(1.-SPRB) + SPRB) + XXK2)
C       ***FLUXES ARE THOSE CARRIED BY THE WATER PHASE                          BPTC = BPTC1A + BPTC2L + BPTC3B
        FLXGWC = (VF1 + VF2)*CWC                                        C
C                                                                       C         ***WATER PHASE CONCENTRATION:
C                                                                       C         ***NAPL PHASE CONCENTRATION = MASS/BULK PARTITION COEFFICIENT
C       ***MAXIMUM FLUX TO GROUNDWATER                                  C         ***CO/XXKO = WATER PHASE CONCENTRATION
        IF (FLXGWC.GT.BFLXC) THEN                                               CO = RMC2/BPTC
          BFLXC = FLXGWC                                                        CWC = CO/XXKO
        ELSE IF (FLXGWC.LT.BFLXC.AND.BFLXC.GT.0..AND.KSTOP.EQ.3) THEN       ELSE
C       ***STOPPING CRITERIA FOR FLUX REDUCED FROM MAXIMUM              C         ***NO LENS PRESENT
          IR = 1                                                                CALL OFLUX (TT,ZLENS,NEQ,QINL,CINL,CWC)
          IC = 1                                                            END IF
        END IF                                                                  IF (CWC.GT.SOLC) CWC = SOLC
```

```
C
C
C
C       ***RECALCULATION OF THE CONSTITUENT FLUX (FLXGWC)
C       ***BASED ON CURRENT CWC
C       ***(WITHOUT THIS RECALCULATION THERE WILL BE AN ERROR IN
C       ***FLUX WHENEVER A STEP IS REJECTED)
        FLXGWC = (VF1 + VF2)*CWC
C
C
        RMDIS2 = DT*(FLXGWC+FLXC1)*0.5
C
C
C
C       ***CHECK FOR CONVERGENCE OF DISSOLUTION TERMS
C       ***WITH RELATIVE ERROR CRITERIA
        DNM = 1.
        IF (RMDIS2.GT.0.) DNM = RMDIS2
        IF (ABS((RMDIS2-RMDISC)/DNM).GT.1.E-5) THEN
          GO TO 10
        END IF
        IF (RMC2.GT.0.) THEN
C         ***RECALCULATE THE LOSS TERMS
          RLOSSC = RMDEGC + RMDIS2 + FOLOSS
          RMC2 = RMC + DELM - RLOSSC
          RMC = RMC2
        END IF
C
C
C
C       ***MAXIMUM CONTAMINANT MASS IN THE LENS
        IF (ZZ(1,4).GT.0.) THEN
C         ***A LENS HAS FORMED (I.E., THE CUMULATIVE FLUX INTO THE LENS IS
C         ***GREATER THAN ZERO)
          IF (RMC.GT.RMCMAX) THEN
C           ***DETERMINE THE MAXIMUM MASS IN THE LENS
            RMCMAX = RMC
          ELSE IF (RMC.LT.OPERC*RMCMAX.AND.RMCMAX.GT.0..AND.
     *    KSTOP.EQ.4) THEN
C           ***STOP THE SIMULATION DUE TO THE MASS BEING REDUCED BELOW THE
C           ***MINIMUM TOLERANCE
            IR = 1
            IC = 1
          END IF
        ELSE
C         ***NO LENS
          IF (KSTOP.EQ.4) THEN
            IF (CMSS.GT.0..AND.DELM.LE.0.) THEN
C             ***STOP THE SIMULATION IF MASS HAS ENTERED THE AQUIFER
C             ***IN THE PAST, BUT THAT THE INCREMENT OF MASS ADDED
C             ***DURING THE CURRENT STEP IS ZERO
              IR = 1
              IC = 1
            END IF
          END IF
        END IF
C
C       ***CUMULATIVE MASS BIODEGRADED
        CMB = CMB + RMDEGC
C       ***CUMULATIVE MASS DISSOLVED
        CMDC = CMDC + RMDISC
C       ***CUMULATIVE FIRST ORDER LOSS
        CFOL = CFOL + FOLOSS
C
C       ***GLOBAL MASS BALANCE QUANTITIES USED IN S.R. ZMASS
C       ***CONSTITUENT IN THE ACTIVE LENS
        CCAL = BPTC2L*CWC*CMKG/CVSV
C       ***CONSTITUENT IN THE RESIDUAL OIL ABOVE THE WATER TABLE
        CCRA = BPTC1A*CWC*CMKG/CVSV
C       ***CONSTITUENT IN THE RESIDUAL OIL BELOW THE WATER TABLE
        CCRB = BPTC3B*CWC*CMKG/CVSV
C       ***CUMULATIVE MASS DISSOLVED
        CCMD = CMDC*CMKG/CVSV
C
      END IF
C
C
C
C     ***DATA OUTPUT WHEN STEP IS GREATER THAN THE MINIMUM
C     ***PRINTING INTERVAL
C     ***AND THE CONTAMINANT FLUX HAS CHANGED SINCE THE LAST OUTPUT
      IF (T2+DTPR.LT.TT.OR.ABS(TT-TM).LE.1.E-5) THEN
        T2 = TT
        DNM = 1.
        IF (FLXGWC.GT.0.) THEN
          DNM = FLXGWC
        END IF
        IF (FLXGWC.LT.BFLXC.
     *  AND.(ABS(FLXGWC-FLAST)/DNM).LT.0.01) THEN
          CONTINUE
        ELSE
          FLAST = FLXGWC
          IF (IWR.EQ.1.AND.IWRITE.EQ.0) THEN
C           ***HEADINGS FOR OILENS CHEMICAL OUTPUT
            WRITE (13,9600) (NT(I),I=1,15)
            WRITE (13,9610)
C           WRITE (19,429)
            IWRITE = 1
          END IF
          E = 0.
          IF (IMSSC.EQ.1.AND.CMSS.NE.0.)
```

```
     1    E = 100.*(RMC+CMB+CMDC+CFOL-CMSS)/CMSS
          IF (IWR.EQ.1) THEN
            WRITE (7,430) T2,RC,FLXGWH*UC2,FLXGWC*UC2,
     1      CMDC*UC2,DEGMAS*UC2,RMC*UC2,CWC,E
C           ***DATA OUTPUT FOR PLOT FILE 3
            WRITE (19,431) T2,RC,FLXGWC*UC2,CMDC*UC2,RMC*UC2
            RTOLD = RC
          END IF
        END IF
       END IF
C
C
C
C
C     ***ROLL VALUES USED FOR INTEGRATION IN THE NEXT TIME STEP
      FLXC1 = FLXGWC
      FLXO1 = FLXO2
C
C
C
C
      RETURN
 429  FORMAT ('BEGIN')
 430  FORMAT (G9.4,(F8.2),2g9.3,4(1X,F8.2),1X,F7.2)
 431  FORMAT (4X,5(G10.4,1X))
 9100 FORMAT (/10X,'SIMULATION ENDS--',
     1'ALL CONTAMINANT LEACHED FROM LENS'/)
 9600 FORMAT (1H1,/15X,50(1H*)/
     1        15X,'*    OILENS MODEL OUTPUT--AQUEOUS CONTAMINANTS   *'/
     2        15X,'************************************************'/
     3        15X,5A10/15X,5A10/15X,5A10/)
 9610 FORMAT (' ',//,10X,'SPECIES','   OIL','      SPECIES',
     1 '  SPECIES', ' MASS  ','   MASS  ','    WATER',
     2 '    P.C.'/
     3 '  TIME   ',' RADIUS   ',' DISSOL. ',' DISSOL. ',
     4 ' DISSOL. ','DEGRADED ','REMAINING',' CONC.  ',
     5 '  MASS'/
     6 ' (DAYS)  ','   (M)   ',' (KG/D)  ',' (KG/D)  ',
     7 '  (KG)   ','  (KG)   ','  (KG)   ',' (MG/L)',
     8 '   ERROR'/
     * 1X,79(1H=)/)
      END
```

```
      SUBROUTINE DUPUIT (RADI,RT,R,HO,HR)
C     ******************************************************************
C     *
C     * DUPUIT
C     *
C     * DUPUIT DETERMINES THE HEIGHT OF THE LENS AS A FUNCTION OF
C     * RADIUS BASED ON THE DUPUIT ASSUMPTIONS OF THE LENS SHAPE
C     *
C     * INPUT ARGUMENTS.............................................
C     *   RADI   RADIUS OF CENTRAL CYLINDER
C     *   RT     RADIUS OF LENS
C     *   R      RADIUS FOR CACULATION
C     *   HO     HEIGHT AT CENTRAL CYLINDER
C     * OUTPUT ARGUMENTS............................................
C     *   HR     HEIGHT AT RADIUS R
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY PROLNS,MAXSAV
C     *
C     ******************************************************************
      COMMON /COUN/ NUM(100)
      NUM(48) = NUM(48) + 1
C
C
      HR   = 0.
      IF (R.LT.0.) STOP 'DUPUIT RADIUS'
      IF (R.LE.RADI) THEN
C       ***INSIDE CYLINDER, H = HO
        FACT = 1.
      ELSE IF (RADI.LT.R.AND.R.LT.RT) THEN
C       ***HEIGHT OF LENS GIVEN BY EQ 2 OF CHARBENEAU AND SHULTZ
        ARG1 = ALOG(RT*RT/R/R)
        ARG2 = ALOG(RT*RT/RADI/RADI)
        FACT = SQRT(ARG1/ARG2)
      ELSE IF (R.GE.RT) THEN
C       ***OUTSIDE LENS, H = 0.
        FACT = 0.
      END IF
      HR = HO*FACT
C
      RETURN
      END
```

```
      SUBROUTINE MAXSAV (RADI,RT,HO,RS,HHS,NR)
C     ****************************************************************
C     *
C     * MAXSAV
C     *
C     * MAXSAV SAVES THE MAXIMUM LENS HEIGHT FOR ALL RADII.  THE DATA
C     * IS USED TO DEFINE THE RESIDUALLY SATURATED ZONES AS THE LENS
C     * DECAYS
C     *
C     * INPUT ARGUMENTS.............................................
C     *    RADI   RADIUS OF CENTRAL CYLINDER
C     *    RT     LENS RADIUS
C     *    HO     LENS HEIGHT AT CENTRAL CYLINDER
C     * OUTPUT ARGUMENTS............................................
C     *    RS()   ARRAY OF RADII
C     *    HHS()  ARRAY OF SAVED LENS HEIGHTS
C     *    NR     NUMBER OF SAVED POINTS
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  DUPUIT,BINARY,REPACK
C     * CALLED BY LCHK2
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      REAL HHS(*),RS(*)
C     DATA IS /0/
      NUM(49) = NUM(49) + 1
      IF (NUM(49).EQ.1) IS = 0
C
C
      IF (RT.LT.RADI) RETURN
      IF (IS.EQ.0) THEN
        IS = 1
        NR = 2
        RS(1) = 0.
        RS(2) = RADI
        HHS(1) = 0.
      END IF
C
      IF (HO.GT.HHS(1)) THEN
        HHS(1) = HO
        HHS(2) = HO
      END IF
C
      DR = (RT-RADI)*0.05
      RBEG = RADI+DR
      REND = RT+0.5*DR
      DO 20 R=RBEG,REND,DR
C       ***DETERMINE THE HEIGHT AT THIS RADIUS
        CALL DUPUIT (RADI,RT,R,HO,HR)
        IF (R.LT.RS(NR)) THEN
C         ***ALL RADII LESS THAN THE PREVIOUS MAXIMUM RADIUS
C         ***INTERPLOLATE THE RADIUS TO APPROXIMATE THE HEIGHT
          CALL BINARY (R,RS(2),NR-1,1,IN)
          IN = IN + 1
          SL = (HHS(IN+1)-HHS(IN))/(RS(IN+1)-RS(IN))
          HN = HHS(IN) + (R-RS(IN))*SL
          IF (HR.GT.HN) THEN
C           ***THE FIRST NEW RADIUS WHERE THE HEIGHT EXCEEDS
C           ***THE PREVIOUS MAXIMUM
            DO 10 I=IN,1,-1
              IF (HHS(I).GT.HR) THEN
C               ***THE PLACE WHERE THE NEW HEIGHT(HR) BELONGS IN THE ARRAY
                NR = I + 1
                HHS(NR) = HR
                RS(NR) = R
                GO TO 15
              END IF
 10         CONTINUE
            STOP 'ERROR IN MAXSAV'
 15         CONTINUE
          END IF
        ELSE
C         ***RADII GREATER THAN PREVIOUS MAXIMUM RADIUS
          NR = NR + 1
          NMAX = 50
          IF (NR+1.GT.100) THEN
            CALL REPACK (RS,HHS,NR-1,NMAX,NP)
            NR = NP
          END IF
          HHS(NR) = HR
          RS(NR) = R
        END IF
 20   CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE REPACK (RS,HHS,NR,NMAX,NP)
C     ****************************************************************
C     *
C     *  REPACK                                                      C
C     *
C     *  REPACK ARRAYS HS AND RS WHEN NR HAS EXCEEDED
C     *  THE MAXIMUM DIMENSIONING
C     *  NEW VALUES ARE SET ON AN EVEN SPACING BY LINEAR INTERPOLATION  50
C     *                                                              C
C     *
C     * INPUT ARGUMENTS ..............................................
C     *  RS   = INDEPENDENT REAL ARRAY TO BE REPACKED (NR ENTRIES)   C
C     *  HS   = DEPENDENT   REAL ARRAY TO BE REPACKED (NR ENTRIES)   C
C     *  NR   = NUMBER OF POINTS (MAXIMUM NUMBER ALLOWED OR DIMENSIONED)
C     *  NMAX = TARGET MAXIMUM NUMBER OF POINTS DESIRED IN ARRAY
C     *       MUST BE LESS THAN THE DIMENSIONING
C     * OUTPUT ARGUMENTS..............................................
C     *  RS   = ADJUSTED INDEPENDENT REAL ARRAY (NP ENTRIES)
C     *  HS   = ADJUSTED DEPENDENT   REAL ARRAY (NP ENTRIES)
C     *  NP   = ACTUAL NUMBER OF POINTS RETURNED
C     *
C     ***WORK
C     *  DR   = NEW SPACING INCREMENT FOR RS
C     *  R    = NEW VALUE FOR RS ARRAY
C     *  IOLD = POINTER TO POSITION IN RS WHOSE VALUE IS JUST BELOW R
C     *  INEW = POINTER TO POSITION IN RS WHERE R WILL BE STORED
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES
C     * CALLED BY
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      REAL HHS(*),RS(*)
      NUM(50) = NUM(50) + 1
C
C
C
      DR = (RS(NR)-RS(2))/FLOAT(NMAX)
      NP = NMAX + 2
      IOLD = 2
C
C     ***LOOP TO REPACK RS AND HHS ARRAYS
      DO 50 INEW = 3,NP-1
        R = RS(INEW-1) + DR
C     ***WHILE LOOP TO LOCATE PLACE IN RS JUST BELOW R
        II = IOLD + 1
30      CONTINUE
        IF(RS(II).LE.R .AND. II.LT.NR) THEN
          II = II + 1
          GO TO 30
        ENDIF
        IOLD = II - 1
C     ***INTERPOLATE FOR HHS(INEW)
        SL = (HHS(IOLD+1) - HHS(IOLD))/(RS(IOLD+1) - RS(IOLD))
        HHS(INEW)=HHS(IOLD) + SL * (R - RS(IOLD))
        RS(INEW) = R
50    CONTINUE
C     ***COPY THE LAST POINT
      RS(NP)=RS(NR)
      HHS(NP)=HHS(NR)
C
C
      RETURN
      END
```

```
      SUBROUTINE VOLT (R,RT,HO,VA,VB,VT)                                C
C     ***************************************************************   C
C     *                                                                 C
C     *  VOLT   CALCULATES THE BULK VOLUME OF THE LENS OUT              C
C     *         TO A RADIUS R                                           C
C     *                                                                 C
C     * INPUT ARGUMENTS.............................................    C
C     *   R      CALCULATE VOLUME TO RADIUS R                           C
C     *   RT     LENS RADIUS                                            C
C     *   HO     OIL HEIGHT ABOVE CAPILLARY FRINGE AT CENTRAL CYLINDAR  C
C     * OUTPUT ARGUMENTS............................................    C
C     *   VA     LENS BULK VOLUME ABOVE WATER TABLE                     C
C     *          (INC. CAPILLARY FRINGE)                                C
C     *   VB     LENS BULK VOLUME BELOW WATER TABLE                     C
C     *   VT     TOTAL VOLUME OCCUPIED BY LENS (BULK VOLUME)            C
C     *                                                                 C
C     *                                                                 C
C     * JIM WEAVER                                                      C
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY                C
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY                   C
C     * ADA, OKLAHOMA  74820                                            C
C     *                                                                 C
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77                  C
C     * REQUIRED ROUTINES  ERF                                          C
C     * CALLED BY  CMASS,HMASS                                          C
C     *                                                                 C
C     ***************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
C
C
      NUM(41) = NUM(41) + 1
C
      ARGR = 0.0
      IF (RT.GE.R) ARGR = SQRT(ALOG((RT/R)**2))
      AR = ERF(ARGR)
C
      ARG = SQRT(ALOG((RT/RADI)**2))
      A = ERF(ARG)
      B1 = PI*HO
      B2 = PI*ALPHA*HO
      T  = R*R*ARGR/ARG + RT*RT*SPI4*(A-AR)/ARG
C
C     ***VOLUME ABOVE THE WATER TABLE (INCLUDES CAPILLARY FRINGE VOLUME)
      VA = PI*(R**2)*FRING3
      VA = VA + B1*T
C
C     ***VOLUME BELOW THE WATER TABLE
      VB = B2*T
C
C     ***TOTAL VOLUME
      VT = VA + VB
C
      RETURN
      END
```

```
      SUBROUTINE LFLUX (R,RT,HO,DHDR,QD,Q)
C     ****************************************************************
C     *
C     *  LFLUX   DETERMINES THE OIL FLUX IN THE LENS
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   R      A GIVEN RADIUS
C     *   RT     RADIUS OF LENS
C     *   HO     HEIGHT OF OIL AT SOURCE
C     * OUTPUT ARGUMENTS............................................
C     *   DHDR   HEAD GRADIENT IN LENS
C     *   QD     DARCY FLUX OF OIL IN LENS
C     *   Q      VOLUME FLUX OF OIL
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  DUPUIT
C     * CALLED BY OILENS
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      NUM(65) = NUM(65) + 1
C
      DHDR = 0.
      QD = 0.
      Q = 0.
C
C     ***HEAD GRADIENT
C     ***#12 PG. 7
      IF (ABS(RT-R).LE.1.E-4.OR.ABS(RADI-RT).LE.1.E-4) RETURN
      T1   = ALOG(RT/R)
      T2   = ALOG(RT/RADI)
      DHDR = -(0.5*HO/R)/SQRT(T1/T2)/T2
C
C     ***DARCY FLUX
      QD = -PERMO*DHDR
C
C     ***VOLUME FLUX
      CALL DUPUIT (RADI,RT,R,HO,HR)
      AREA = 2.*PI*R*(FRING3 + AP1*HR)
      Q = QD*AREA
C
      RETURN
      END
```

```
      SUBROUTINE OIL (ZZ,TT,SW,NEQ2,SO)
C     ************************************************************
C     *
C     *  OIL
C     *
C     *  FOR TIME TT, (BETWEEN PT(1) AND PT(2)), OIL FINDS THE OIL
C     *  SATURATION AT DEPTH ZZ GIVEN THE WATER SATURATION SW
C     *
C     *                                                           
C     * INPUT ARGUMENTS.....................................     
C     *   ZZ    DEPTH IN METERS                                 
C     *   TT    TIME IN DAYS                                    
C     *   SW    WATER SATURATION                                
C     *   NEQ2  NUMBER OF OIL CHARACTERISTICS                   
C     * OUTPUT ARGUMENTS.....................................     
C     *   SO    OIL SATURATION                                  
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES   BINARY,INQUAD
C     * CALLED BY   CEQS,DSO,HIST,OFLUX,OCP,OILENS,ZMASS
C     *
C     ************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /FLAG/ ICONC,IKOPT,ILENS,ITSGP
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /MULL/ VPA
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
C
      NUM(11) = NUM(11) + 1
      SO = 0.
C
C
C     ***IMMOBILIZED OIL LAYE1S
      IF (IAT.EQ.2.AND.QP.LE.0.) THEN
C
        IF (DPA.LE.ZZ.AND.ZZ.LE.DPL) THEN
          XXX = 1.
          IF (TT.GT.OLAG) XXX = EXP(-DLO*(TT-OLAG))
          SO = SPMAX*XXX
        END IF
        RETURN
C
      END IF
C
      IF (ZZ.LT.DPA) RETURN
C
C
C     ***OILENS
      IF (VIN2.GT.0..AND.WTABLE.GT.0.) THEN
      END IF
      DT = PT(2) - PT(1)
      ZF = PZ(2)
      IF (TT.LT.PT(2).AND.PT(2).NE.PT(1)) THEN
        IF (TT.LT.PT(1)) STOP 'BAD-OIL'
        ZF = PZ(1)+(TT-PT(1))*(PZ(2)-PZ(1))/DT
      END IF
      IF (ZZ.GT.ZF) RETURN
C
C
C     ***MOBILE OILS
      IF (DLO.EQ.0.) THEN
        Z22 = DPA + (TT - TPE)*DZPDDT
        IHS = 0
        IF (IKOPT.EQ.1) THEN
C
          IF (ZZ.GE.Z22.AND.(ZZ.LT.ZF.OR.
     *    ABS(ZZ-ZF).LE.1.E-5)) THEN
            SO = SPMAX
          ELSE IF (ZZ.LE.ZF) THEN
C           ***KINEMATIC MODEL
            SL = (ZZ-DPA)/(TT-TPE)
            CALL INQUAD (WP,50,SPR,1.-SWMAX,SL*ETA,SO)
          END IF
C
        END IF
        IF (IKOPT.EQ.2) THEN
C
C         ***MULL'S RECTANGULAR PROFILE MODEL
          IF (VPA.LE.0.) THEN
            SO = SPMAX
          ELSE
            SO = VPA/ETA/ZF
          END IF
C
        END IF
      ELSE
C
C
C       ***DEGRADING OR DISSOLVING NAPL
```

```
        IF (IKOPT.EQ.2) STOP 'MULL MODEL FOR NON-DISSOLVING NAPL ONLY'
        ZF = PZ(2)
        IF (PT(1).LE.TPE) THEN
          LZ = NEQ2
          DO 20 I=NEQ2-1,1,-1
            IF (ZI(I,2).LE.DPA) THEN
              LZ = I + 1
              GO TO 21
            END IF
 20       CONTINUE
 21       CONTINUE
        ELSE
          LZ = 1
        END IF
C
C
        IF (ZZ.LE.ZI(LZ,2).AND.TT.LE.TPE) THEN
          SL = 0.
          IF (ZI(LZ,2).GT.DPA) SL = (SI(LZ,2)-SPMAX)/
     1    (ZI(LZ,2)-DPA)
          SO = SPMAX + SL*(ZZ-DPA)
        ELSE
          IX = 0
          CALL BINARY (ZZ,ZI(LZ,2),NEQ2-LZ+1,1,IX)
          IF (LZ.GT.1) IX = LZ + IX - 1
          SL = (SI(IX+1,2)-SI(IX,2))/(ZI(IX+1,2)-ZI(IX,2))
          SO = SI(IX,2) + (ZZ-ZI(IX,2))*SL
        END IF
C
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE CONCE (TT,ZZ,SW,SO,NEQ3,CO,CMSS,IV)
C     ************************************************************
C     *
C     *   CONCE
C     *
C     *   CONCE DETERMINES THE OIL PHASE CONCENTRATION AT TIME TT
C     *   AND DEPTH ZZ.  OTHER NEEDED INFORMATION ARE THE WATER
C     *   AND OIL SATURATIONS (SW,SO) AND THE NUMBER OF CONC.
C     *   CHARACTERISTICS (NEQ2).  CMSS IS THE MASS IN ALL PHASES
C     *   IF IV=1 VOLATILIZATION IS CONSIDERED.
C     *
C     * INPUT ARGUMENTS.........................................
C     *   TT   TIME IN DAYS
C     *   ZZ   DEPTH IN METERS
C     *   SW   WATER SATURATION
C     *   SO   OIL SATURATION
C     *   NEQ3 (INTEGER) NUMBER OF EQUATIONS IN GROUP 3
C     *   IV   VOLATILIZATION ARGUMENT (NOT USED)
C     * OUTPUT ARGUMENTS........................................
C     *   CO   CONCENTRATION IN THE OIL PHASE
C     *   CMSS CONSITUENT MASS
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES:  BINARY,SFLUX,RD
C     *  CALLED BY: LCHK2,CEQS,ZMASS,OCP,HIST,OFLUX
C     ************************************************************
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(8) = NUM(8) + 1
C
C
      CO = 0.
      CMSS = 0.
      IF (COINI.LE.0.) RETURN
      IF (IV.EQ.1.AND.ZZ.LT.ZV(2)) RETURN
      DT = PT(2) - PT(1)
C
C     ***FIND NUMBER OF ACTIVE CHARACTERISTICS (LZ)
C     ***DEFAULT IS TOTAL NUMBER (NEQ3)
      LZ = NEQ3
      IF (TT.LT.TPE*1.15) THEN
        DO 20 I=2,NEQ3
          IF (ZC(I,2).EQ.DPA) THEN
            LZ = I
            IF (TT.LT.TPE) LZ = LZ - 1
            GO TO 21
          END IF
 20     CONTINUE
 21     CONTINUE
      END IF
C
C     ***FIND DEPTH OF DEEPEST & SHALLOWEST CHARACTERISTICS
      ZLOW  = ZC(LZ,2)
      CLOW  = CC(LZ,2)
      ZHIGH = ZC(1,2)
C     ***DETERMINE THE CONCENTRATION
      IF (ZZ.GT.ZHIGH) RETURN
      IF (ZZ.LE.ZLOW) THEN
        IF (TT.GT.TPE) THEN
          RETURN
        ELSE IF (TT.LE.TPE) THEN
C         ***DURING LOADING
          CC1 = CPART
          IF (IGA.EQ.1) THEN
            CALL SFLUX (TT,QO)
            IF (QO.GT.0..AND.QW.GT.0.) CC1 = QO*COINI/(QO + XXK1*QW)
          END IF
          CO = CC1
          IF (ZLOW.GT.DPA) CO = CC1 + (ZZ-DPA)*(CLOW - CC1)/(ZLOW-DPA)
        END IF
      ELSE
C       ***LINEAR INTERPOLATION
        CALL BINARY (ZZ,ZC(1,2),LZ,1,IN2)
        C1 = CC(IN2,2)
        C2 = CC(IN2+1,2)
        Z1 = ZC(IN2,2)
        Z2 = ZC(IN2+1,2)
        IF (ZC(IN2+1,2).LT.PZ(2).AND.PZ(2).LT.ZC(IN2,2).
     1  AND.DLO.GT.0.) THEN
C         ***APPROXIMATE CONCENTRATION NEAR THE OIL FRONT
          IF (ZZ.LT.PZ(2)) THEN
            C1 = CC(IN2+1,2)
            Z1 = PZ(2)
          ELSE
            Z2 = PZ(2)
          END IF
        END IF
        SL = 0.
```

```
         IF (ABS(Z2-Z1).GE.1.E-5.AND.ABS(C2-C1).GE.1.E-5)
     1   SL = (C2-C1)/(Z2-Z1)
         CO = C1 + (ZZ-Z1)*SL
       END IF
C
       CMSS = RD(SO,SW)*CO
C
       RETURN
       END
```

```
      SUBROUTINE CVEL (TT,SO,SW,DZCDT)                                  C
C     ****************************************************************
C     *                                                                C
C     *  CVEL   THE CONCENTRATINO VELOCITY--I.E., THE CONCENTRATION    C
C     *         VELOCITY ASSOCIATED WITH A GIVEN OIL AND WATER SATURATION
C     *         AT A GIVEN TIME                                        C
C     *                                                                C
C     *                                                                C
C     *                                                                C
C     * INPUT ARGUMENTS.....................................           C
C     *   TT    TIME IN DAYS                                           C
C     *   SO    OIL SATURATION                                         C
C     *   SW    WATER SATURATION                                       C
C     * OUTPUT ARGUMENTS....................................           C
C     *   DZCDT CONCENTRATION VELOCITY
C     *
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  PKE,RD
C     * CALLED BY  CEQS
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
      NUM(7) = NUM(7) + 1
C
C
      QWC = QW
      IF (IGA.EQ.1.AND.CHS(2).GE.0.) THEN
C
C       ***PRESSURE FLOW
        QOC = 0.
        IF (SO.GT.SPR) THEN
          QOC = PF(2)
          IF (TT.LT.PT(2)) THEN
            IF (TT.LT.PT(1)) STOP 'CVEL-TIME'
            SL = (PF(2)-PF(1))/(PT(2)-PT(1))
            QOC = PF(1) + (TT-PT(1))*SL
          END IF
        END IF
      ELSE
C
C       ***KINEMATIC FLOW
        CALL PKE (SO,SW,QOC)
C
      END IF
C
C
C
      DZCDT = (QWC*XXK1 + QOC)/RD(SO,SW)
C
      RETURN
      END
```

```
      FUNCTION RD(SO,SW)
C     ****************************************************************
C     *
C     * THE FUNCTION RD   IS USED TO DETERMINE THE NAPL BULK PHASE
C     *                   PARTITION COEFFICIENT
C     *
C     *
C     *
C     * INPUT ARGUMENTS.........................................
C     *   SO    OIL SATURATION
C     *   SW    WATER SATURATION
C     * OUTPUT ARGUMENTS........................................
C     *   RD    THE RETARDATION COEFFICIENT
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES
C     * CALLED BY  CVEL,CONCE,LCHK2,INTLNS
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
C
      NUM(6) = NUM(6) + 1
C
C
C     ***BULK NAPL PHASE PARTITION COEFFICIENT
      RD = ETA*(SW*XXK1+SO) + XXK2
C
      RETURN
      END
```

```
      SUBROUTINE DSO (TT,ZZ,NEQ2,COW,CH)
C     ****************************************************************
C     *
C     *  DSO DETERMINES THE MASS OF OIL DISSOLVED IN THE WATER PHASE
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   TT     TIME IN DAYS
C     *   ZZ     DEPTH IN METERS
C     *   NEQ2   NUMBER OF OIL CHARACTERISTICS
C     * OUTPUT ARGUMENTS............................................
C     *   COW    CONCENTRATION OF OIL DISSLOVED IN WATER
C     *   CH     DISSOLVED OIL MASS PER UNIT VOLUME
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES
C     * CALLED BY
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /HYDI/ HDC1,HDC2,HDC3,HDC4
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /STOI/ BDFC,BDFH,SOLC,SOLH,STOICC,STOICH
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(64) = NUM(64) + 1
C     ***DEFAULTS
      COW = 0.
      CH = 0.
C
C     ***RETURN IF OIL IMMISCIBLE
      IF (SOLH.LE.0.) RETURN
C     ***RETURN IF OIL NEAR SURFACE HAS DISSOLVED AWAY
      CALL OIL (ZZ,TT,SWMAX,NEQ2,SO)
      IF (ZZ.LE.PZ(2).AND.SO.LE.0.) RETURN
C     ***RETURN IF DEPTH GREATER THAN MAXIMUM DEPTH OF DISS. OIL
      ZT = MAX(PZ(2) , (PT(2)-TPB)*HDC1)
      IF (ZZ.GT.ZT) RETURN
C
C     ***DISSOLVED OIL MASS PER UNIT VOLUME
      COW = SOLH
      CH = HDC4
C
      RETURN
      END
```

```
      SUBROUTINE SFLUX (TT,QO)
C     ****************************************************************
C     *
C     *  SFLUX   DETERMINES THE OIL FLUX AT THE SURFACE
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   TT   TIME IN DAYS
C     * OUTPUT ARGUMENTS............................................
C     *   QO   OIL FLUX
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  CONCE,OCHK2
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
C
C
      NUM(13) = NUM(13) + 1
      QO = 0.
C
C
      IF (IGA.EQ.0) THEN
        IF (TT.LE.TPE) QO = QP
      ELSE
C
        IF (CHS(1).GT.0..OR.(TT.LE.TPE)) THEN
          QO = PF(2)
          IF (PT(1).LE.TT.AND.TT.LT.PT(2)) THEN
            SL = (PF(2)-PF(1))/(PT(2)-PT(1))
            QO = PF(1) + (TT-PT(1))*SL
          END IF
        END IF
C
      END IF
C
C
      RETURN
      END
```

```
      SUBROUTINE OFLUX (TT,ZH,NEQ,QINL,CINL,CWINL)
C     ************************************************************
C     *                                                          *
C     * OFLUX                                                    *
C     *                                                          *
C     * OFLUX DETERMINES THE NAPL VOLUME FLUX (M^3/T) AT A SPECIFIED
C     * DEPTH AND TIME.  THE ROUTINE ALSO RETURNS THE NAPL AND WATER
C     * PHASE CONSITUTENT CONCENTRATIONS (MG/L).
C     *
C     * INPUT ARGUMENTS
C     * TT = TIME
C     * ZH = DEPTH
C     *
C     * OUTPUT ARGUMENTS
C     * QINL  = OIL FLUX (VOLUME/TIME)
C     * CINL  = CONSTITUENT CONCENTRATION IN THE NAPL PHASE
C     * CWINL = CONSTITUENT CONCENTRATION IN THE WATER PHASE
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION, ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES: OIL,PKE,CONCE
C     * CALLED BY : OILENS,CLENS
C     ************************************************************
      DIMENSION NEQ(*)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(37) = NUM(37) + 1
C
C
C
      IF (IGA.EQ.1.AND.TT.LE.TPE) THEN
C       ***PRESSURE AND GRAVITY DRIVEN NAPL FLUX DETERMINED BY OEQS
        QINL = PF(2)
        SO = SPMAX
      ELSE
C       ***KINEMATIC NAPL FLUX
        CALL OIL (ZH,TT,SWMAX,NEQ(2),SO)
        CALL PKE (SO,SWMAX,QINL)
      END IF
C
C
C     ***DETERMINE THE CHEMICAL CONCENTRATION IN THE NAPL PHASE
      CALL CONCE (TT,ZH,SWMAX,SO,NEQ(3),CINL,XX,IX)
C
C     ***DETERMINE THE WATER PHASE CONCENTRATION
      CWINL = XXK1*CINL
C
C     ***NAPL VOLUME FLUX=M**3/M**2/T * AREA OF SOURCE
      QINL = QINL*AC
C
C
      RETURN
      END
```

```
      SUBROUTINE WKE (SW,XKS,XKR)
C     ****************************************************************
C     ***
C     ***WATER EFFECTIVE/RELATIVE PERMEABILITY
C     ***
C     ***INPUT ARGUMENTS
C     ***SW     WATER SATURATION
C     ***XKS    SATURATED CONDUCTIVITY
C     ***       ENTER 1.0 FOR RELATIVE PERMEABILITY
C     ***OUTPUT ARGUMENTS
C     ***XKR    RELATIVE OR EFFECTIVE PERMEABILITY
C     ***
C     ***X-21-87
C     ***IIX-31-89 BURDINE VAN GENUCHTEN MODEL ADDED
C
C     * REQUIRED ROUTINES:  NONE
C     * ENTRY POINTS: PKE,SLOPE,KRSET
C     * THIS ROUTINE IS CALLED BY:  WFFS, INITIA
C     ****************************************************************
      COMMON /BVGE/ XM,XMMO,XMR,XMRMO,XN
      COMMON /COUN/ NUM(100)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RLTV/ KRF,KRFO,XLAMB,EPS,EM1,EM2,CC1,REM1
      NUM(22) = NUM(22) + 1
C
      XKR = 0.
      IF (SW.LE.SWR) RETURN
      IF (KRF.EQ.1) THEN
C       ***BURDINE--BROOKS & COREY
        XKR = XKS*((SW - SWR)/OMSWR)**EPS
      ELSE IF (KRF.EQ.2) THEN
C       ***BURDINE--VAN GENUCHTEN
        SE = (SW-SWR)/OMSWR
        XKR = 1. - (1.-SE**XMR)**XM
        XKR = XKS*SE*SE*XKR
      END IF
      RETURN
      END
```

```
      SUBROUTINE SCHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ****************************************************************
C     *
C     * SCHK    PERFORMS CHECKS ON THE SIMULATION PARAMETERS
C     *
C     *
C     *
C     * INPUT ARGUMENTS.....................................
C     *     SAME AS SUBROUTINE CHK
C     * OUTPUT ARGUMENTS....................................
C     *     SAME AS SUBROUTINE CHK
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  CHK
C     *
C     ****************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /HIPR/ IHIST,NTIMES,PR(10),NZS,HI(5)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /TRIA/ DTOLD,DTOLD1,I1,I2,I5,ICC,ICHAR,TEND
      NUM(32) = NUM(32) + 1
C
C
C     ***MAKE LARGE STEP OUTPUT LOOK NICE
      IF (IRR(11).EQ.0.AND.PT(2)-PT(1).GT.DTOLD.AND.TT.GT.TPE) THEN
        DTR(11) = FLOAT(IFIX(PT(2)*10000.)/10000) - PT(1)
        IF (DTR(11).GT.0.) THEN
          IR = 1
        ELSE
          DTR(11) = DTOLD
        END IF
      END IF
C
C
C     ***SIMULATION ENDING TIME
      IF (KSTOP.EQ.1.AND.IRR(12).EQ.0.AND.PT(2).GT.TM) THEN
        IR = 1
        DTR(12) = TM - PT(1)
        IF (DTR(12).EQ.0.) IC = 1
      END IF
C
C
```

```
C     ***PROFILE TIMES
      IF (IRR(13).EQ.0.AND.NTIMES.GT.0) THEN
        IF (TT.GE.PR(1)) THEN
          DTR(13) = PR(1) - PT(1)
          IF (ABS(TT-PR(1)).LT.1.E-6) THEN
            DTR(13) = 1.E19
            IRR(13) = 1
          ELSE IF (DTR(13).GT.0.) THEN
            IR = 1
          END IF
        END IF
      END IF
C
C
C     *** HISTORY DEPTHS
      IF (IRR(17).EQ.0.AND.NZS.GT.0) THEN
        IF (ZZ(1,1).GT.HI(I5)) THEN
          IR = 1
          SL = (ZZ(1,1)-Z1(1,1))/(PT(2)-PT(1))
          DTR(17) = (HI(I5)-Z1(1,1))/SL
        END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE SCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)          RETURN
C     ***************************************************************          END
C     *
C     * SCHK2   ADJUSTS SIMULATION PARAMETERS ON ACCEPTED STEPS
C     *
C     *
C     * INPUT ARGUMENTS.....................................
C     *     SAME AS SUBROUTINE CHK
C     * OUTPUT ARGUMENTS....................................
C     *     SAME AS SUBROUTINE CHK
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  CHK
C     *
C     ***************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /TRIA/ DTOLD,DTOLD1,I1,I2,I5,ICC,ICHAR,TEND
      NUM(33) = NUM(33) + 1
C
C
C     ***MAKE LARGE STEP OUTPUT LOOK NICE
      IF (IRR(11).EQ.1) THEN
        IRR(11) = 2
      ELSE IF (IRR(11).EQ.2) THEN
        IF (PT(2)-PT(1).LT.DTOLD) THEN
          IRR(11) = 0
        END IF
      END IF
C
C
C     ***SIMULATION ENDING TIME
      IF (IRR(12).EQ.1) IC = 1
C
C
C     ***PROFILE TIMES
      IF (IRR(13).EQ.1) IRR(13) = 0
C
C
C     ***HISTORY DEPTHS
      IF (IRR(17).EQ.1) THEN
        I5 = I5 + 1
        IF (I5.EQ.4) IRR(17) = 2
      END IF
C
```

```
      SUBROUTINE OCHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ************************************************************
C     *
C     *  OCHK CHECKS GENERAL CONDITIONS IN THE OIL(NAPL) PHASE
C     *  THE STEP IS REJECTED IF A CONDITION IS VIOLATED
C     *  THE CONDITIONS CHECKED:
C     *      IRR(6)     MINIMUM OF 5 STEPS DURING THE LOADING
C     *      IRR(4)     ENDING TIME FOR POLLUTANT LOADING (T=TPE)
C     *      IRR(5)     START OF EACH CONSTITUENT CHARACTERISTIC
C     *      IRR(2)     END OF PLATEAU OIL(NAPL) SATURATION
C     *
C     *  INPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *  OUTPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES: NONE
C     *  THIS ROUTINE IS CALLED BY: CHK
C     ************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /TRIA/ DTOLD,DTOLD1,I1,I2,I5,ICC,ICHAR,TEND
C     DATA NCALL /0/
C
C
      NUM(31) = NUM(31) + 1
      IF (NUM(31).EQ.1) THEN
        NCALL = 0
      END IF
      IF (NCALL.EQ.0) DTOLD = DMAX
      NCALL = 1
C
C
C
C     ***ASSURE AT LEAST 5 STEPS THROUGH OIL APPLICATION
      IF (IRR(6).EQ.0.AND.DMAX.GT.0.75*DTP
C     ***EXLUDE CASES OF ZERO DURATION (IAT=2 AND SOME IAT=4)
     *.AND.DTP.GT.0..AND.IAT.NE.2) THEN
        DTOLD = DMAX
        DMAX = DTP*.2
        DTR(6) = DT
        IRR(6) = 1
        IR = 1
      END IF
C
C
C     ***CHECK FOR POLLUTANT ENDING TIME IF NOT A VOLUME LOADING
      IC4 = 0
      IF (IRR(4).EQ.0.AND.IAT.NE.2.AND.IAT.NE.4) IC4 = 1
C     ***ONLY CHECK IAR=4 CASES WHEN PAST TPE (IRR(20)=1)
      IF (IRR(4).EQ.0.AND.IAT.EQ.4.AND.IRR(20).EQ.1) IC4 = 1
      IF (IC4.EQ.1) THEN
        IF (TT.GT.TPE) THEN
          IR = 1
          DTR(4) = TPE - PT(1)
        END IF
      END IF
C
C
      IF (((COINI.GT.0..AND.I1.LE.NEQ(3))
C     ***CASES WITH DISSOLVED CONSTITUENT
     *.OR.(DLO.GT.0..AND.I1.LE.10.AND.IGA.EQ.0))
C     ***BIODEGRADATION CASES WITHOUT GREEN-AMPT FLOW
     *.AND.PT(1).LE.TPE) THEN
C       ***STEP TO THE TIME EACH NEW CHARACTERISTIC STARTS
        IF (TC(2).NE.T0(100).AND.IGA.NE.1) STOP 'OCHK-100'
        IF (IRR(5).EQ.0.) THEN
          IF (TC(I1).LE.TT) THEN
C           ***I1 WILL BE INCREMENTED ONLY IF THIS NEW STEP SIZE DOMINATES
            IF (TC(I1).EQ.PT(1)) GO TO 300
            IR = 1
            DTR(5) = TC(I1)-PT(1)
          END IF
 300      CONTINUE
        END IF
      END IF
C
C
C     ***END OF MAXIMUM OIL SATURATION
      IF (IRR(2).EQ.0.AND.((TT.GT.TPE.AND.IGA.EQ.0).OR.
     1(CHS(1).LE.0.AND.IGA.EQ.1)).AND.DLO.EQ.0.) THEN
        IF (PS(1).LT.SPMAX) IRR(2) = 2
        ZS = DPA + (TT-TPE)*DZPDDT
        IF (ZZ(1,1).LT.ZS) THEN
          SL = DZPMDT
          T11 = TPB
          Z11 = DPA
          IF (IGA.EQ.1) THEN
            SL = (PZ(2)-PZ(1))/(PT(2)-PT(1))
            T11 = PT(1)
            Z11 = PZ(1)
          ELSE IF (IAT.EQ.2) THEN
            T11 = TPE
            Z11 = DPL
          END IF
          TNEW = (Z11 - DPA - T11*SL + TPE*DZPDDT)/(DZPDDT - SL)
          DTX = TNEW - PT(1)
          IF (DTX.GT.0.) THEN
            DTR(2) = DTX
            IR = 1
          END IF
```

```
          END IF
        END IF
C
        RETURN
        END
```

```
      SUBROUTINE OCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ************************************************************
C     *
C     *  GACHK2 SETS GENERAL OIL(NAPL) PHASE CONDITIONS
C     *  THE CONDITIONS SET:
C     *      IRR(6)     MINIMUM OF 5 STEPS DURING THE LOADING
C     *      IRR(4)     ENDING TIME FOR POLLUTANT LOADING (T=TPE)
C     *      IRR(5)     START OF EACH CONSTITUENT CHARACTERISTIC
C     *      IRR(2)     END OF PLATEAU OIL(NAPL) SATURATION
C     *      IRR(8)     BIODEGRADATION IN GREEN-AMPT CASES
C     *
C     *  INPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *  OUTPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES:  SFLUX, MESSAG
C     *  THIS ROUTINE IS CALLED BY: CHK
C     ************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /STOI/ BDFC,BDFH,SOLC,SOLH,STOICC,STOICH
      COMMON /TRIA/ DTOLD,DTOLD1,I1,I2,I5,ICC,ICHAR,TEND
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
C
      NUM(34) = NUM(34) + 1
C     DATA IMES /0/
      IF (NUM(34).EQ.1) THEN
        IMES = 0
      END IF
      IF (IMES.EQ.0) THEN
        CALL MESSAG (10)
        IMES = 1
      END IF
C
C
C
C     ***ASSURE AT LEAST 5 STEPS THROUGH OIL APPLICATION
      IF (IRR(6).EQ.1) THEN
        IF (TT.GE.TPE) THEN
          DMAX = DTOLD
          IRR(6) = 2
        END IF
      END IF
C
C
      IF (TT.EQ.TPE) IRR(4) = 1
C     ***CHECK FOR POLLUTANT ENDING TIME
      IF (IRR(4).EQ.1.OR.(IRR(20).EQ.1.AND.IAT.EQ.4)) THEN
        IRR(4) = 2
        IRR(5) = 1
C       ***TRIGGER GREEN-AMPT SOLUTION CHANGES
        IF (IGA.EQ.1) IRR(8) = 1
        IF (NEQ(1).EQ.0) NEQ(1) = 1
C       ***RESET TIME STEP AND MAXIMUM STEP
        DT = 0.0001
        DTOLD1 = DMAX
C       ***VOLATILE FRONT SOLUTION
        IF (XXK3.GT.0..AND.COINI.GT.0.) NEQ(1) = 2
C       ***TRIGGER SOLUTION FOR DISSOLVING OIL PHASE CHARACTERISTICS
        IF (SOLH.GT.0.) IRR(18) = 2
        CALL MESSAG (11)
      END IF
C
C
C     ***STEP TO THE TIME EACH NEW CHARACTERISTIC STARTS
      IF (IRR(5).EQ.1) THEN
C       ***CORRECT CONCENTRATIONS FOR GREEN AMPT CASES
        IF (IGA.EQ.1.AND.QW.GT.0..AND.XXK1.GT.0.) THEN
          CALL SFLUX (TC(I1),QO)
          CPART = QO*COINI/(QO + XXK1*QW)
          C0(I1) = CPART
          CC(I1,2) = C0(I1)
        END IF
        I1 = I1 + 1
        IRR(5) = 0
      END IF
C
C
C     ***END OF MAXIMUM OIL SATURATION
      IF (IRR(2).EQ.1) THEN
        IRR(2) = 0
        IF (PS(1).LT.SPMAX) IRR(2) = 2
        DT = 0.0001
C       DMAX = DTOLD1
      END IF
C
C
C     ***BIODEGRADATION DURING GREEN AMPT CASES
      IF (IRR(8).EQ.1) THEN
        IRR(2) = 0
        IRR(8) = 3
        DT = 0.0001
        IF (ICHAR.EQ.1) THEN
          IF (TPE.GT.TC(NEQ(3))) THEN
            NEQ(3) = NEQ(3) + 1
          ELSE
            N1 = NEQ(3)
            DO 20  I=N1,1,-1
              IF (TPE.GT.TC(I)) THEN
```

```
                NEQ(3) = I + 1
                GO TO 21
              END IF
 20         CONTINUE
            STOP 'ICHAR-FAIL'
 21         CONTINUE
          END IF
          TC(NEQ(3)) = TPE
          C0(NEQ(3)) = CPART
          CC(NEQ(3),2) = CPART
          ZC(NEQ(3),2) = DPA
        END IF
        IF (DO.GT.0.) THEN
C         ***TURN ON BIODEGRADATION AFTER PRESSURE FLOW STOPS
          DLO = DO
          NEQ(2) = 101
C         ***DEFAULTS FOR CHARACTERISTICS
          DZ = (PZ(2)-DPA)*0.1
          Z  = PZ(2)
          DO 50 I=101,91,-1
            T0(I) = TPE
            ZI(I,1) = Z
            ZI(I,2) = Z
            ZZ(I,2) = Z
            Z = Z - DZ
 50       CONTINUE
        END IF
      END IF
C
C
      RETURN
      END
```

```
      SUBROUTINE GACHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ************************************************************
C     *
C     *  GACHK CHECKS CONDITIONS FOR VARIABLE HEAD PONDING.
C     *  THE STEP IS REJECTED IF A CONDITION IS VIOLATED
C     *  THE CONDITIONS CHECKED:
C     *      IRR(20)    ZERO PONDING AT SURFACE--END OF GREEN-AMPT
SOLUTION
C     *
C     *  INPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *  OUTPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES: NONE
C     *  THIS ROUTINE IS CALLED BY: CHK
C     ************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
C
C
C
C     ***EXECUTE CHECKS ONLY IF IAT = 4
      IF (IAT.NE.4) RETURN
C
C
C
C     ***KEEP TPE GREATER THAN CURRENT TIME (ACTUAL TPE STORED IN TPEM)
      IF (CHS(2).GT.0..AND.PT(2).GE.TPE-1.) TPE = TPE + 3.0
C
C
C     ***SAVE OIL EVENT ENDING TIME IN TEMPORARY VARIABLE
C     ***(DONE IN SUBROUTINE INITIA)
C
C
C     ***ZERO HEAD PONDING CONDITION AT SURFACE
      IF (IRR(20).EQ.0.AND.ZZ(3,1).LE.0.AND.IAT.EQ.4) THEN
        SL = (ZZ(3,1)-Z1(3,1))/(PT(2)-PT(1))
        IF (SL.NE.0.) THEN
          IR = 1
          DTR(20) = -Z1(3,1)/SL
        END IF
      END IF
C
C
      RETURN
      END
```

```
      SUBROUTINE GACHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ************************************************************
C     *
C     *  GACHK2 SETS CONDITIONS FOR VARIABLE HEAD PONDING (
C     *  BOUNDARY CONDITION
C     *  IAT = 4) AFTER  ACCEPTED STEPS
C     *  THE CONDITIONS SET:
C     *      IRR(20)    ZERO HEAD PONDING AT SURFACE
C     *
C     *  INPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *  OUTPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES:  RECHAR,SFLUX,ZMASS
C     *  THIS ROUTINE IS CALLED BY: CHK
C     ************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /MULL/ VPA
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
C
C     ***EXECUTE CHECKS ONLY IF IAT = 4
      IF (IAT.NE.4) RETURN
C
C
C     ***ZERO HEAD PONDING AT SURFACE
      IF (IRR(20).EQ.1) THEN
        CHS(2) = 0.
        ZZ(3,1) = 0.
        NEQ(1) = 2
        TPE = TT
        IRR(20) = 2
C       ***NAPL VOLUME/UNIT AREA AT THE END OF THE PONDING PERIOD
        CALL ZMASS (TT,ZZ(1,1),SPMAX,NEQ,0,ZZ,ND,OMASS,CMASS)
        VPA = OMASS*DVSV/DMKG/PRHO
        IF (COINI.GT.0.)
     *  CALL RECHAR (30,45,PT(1),ZZ,ND,TC,C0,ZC,CC,NEQ)
        NEQ(3) = NEQ(3)+1
        TC(NEQ(3)) = TT
        CALL SFLUX (TT,QO)
        CPART = QO*COINI/(QO + XXK1*QW)
        C0(NEQ(3)) = CPART
        CC(NEQ(3),2) = CPART
        CC(NEQ(3),1) = CPART
        I1 = NEQ(3)
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE CCHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ************************************************************
C     *
C     *  CCHK CHECKS CONDITIONS FOR DISSOLVED CONSTITUENTS.
C     *  THE STEP IS REJECTED IF A CONDITION IS VIOLATED
C     *  THE CONDITIONS CHECKED:
C     *      IRR(10)    CHARACTERISTICS LEAVE OIL PHASE
C     *      IRR(3)     VOLATILIZATION OF CONSITITUENT
C     *
C     *  INPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *  OUTPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES: NONE
C     *  THIS ROUTINE IS CALLED BY: CHK
C     ************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /TRIA/ DTOLD,DTOLD1,I1,I2,I5,ICC,ICHAR,TEND
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(35) = NUM(35) + 1
C
C     ***ONLY CHECK IF CONSTITUENT HAS BEEN APPLIED
      IF (COINI.LE.0.) RETURN
C
C
C     ***IMPROVE ACCURACY IF CONSTITUENT CHARACTERISTICS LEAVE OIL
 5    CONTINUE
      IF (IRR(10).EQ.0.AND.QW.GT.0.) THEN
        IF (ZC(I2,2).GE.PZ(2)) THEN
          IF (ZC(I2,2).EQ.PZ(2)) THEN
            IRR(10) = 1
            GO TO 5
          END IF
          SL1 = (ZC(I2,2)-ZC(I2,1))/DT
          SL2 = (PZ(2)-PZ(1))/DT
          TNEW = (PZ(1)-ZC(I2,1)+PT(1)*(SL1-SL2))/(SL1-SL2)
          DTT = TNEW-PT(1)
          IF (DTT.GT.0.) THEN
            IR = 1
            DTR(10) = DTT
          END IF
        END IF
      END IF
C
C
C     ***VOLATILIZATION OF ENTIRE MASS OF CONSTITUENT
      IF (XXK3.GT.0..AND.IRR(3).EQ.0) THEN
        IF (ZV(2).GT.ZC(1,2)) THEN
          IR = 1
          DD = PT(2) - PT(1)
          SL1 = (ZV(2)-ZV(1))/DD
          SL2 = (ZC(1,2)-ZC(1,1))/DD
          TNEW = (ZV(1)-ZC(1,1)+PT(1)*(SL2-SL1))/(SL2-SL1)
          DTR(3) = TNEW - PT(1)
        END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE CCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ************************************************************
C     *
C     *  CCHK2 SETS CONDITIONS FOR DISSOLVED CONSTITUENTS AFTER
C     *  ACCEPTED STEPS
C     *  THE CONDITIONS SET:
C     *      IRR(10)    CHARACTERISTICS LEAVE OIL PHASE
C     *      (*)        GENERATION OF CHARACTERISTICS FOR IAT=4
C     *                 SOURCE CONDITION--PONDING DECAY
C     *      IRR(3)     VOLATILIZATION OF CONSITITUENT
C     *
C     *  INPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *  OUTPUT ARGUMENTS:  SAME AS SUBROUTINE CHK
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES: RECHAR,SFLUX
C     *  THIS ROUTINE IS CALLED BY: CHK
C     ************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /TRIA/ DTOLD,DTOLD1,I1,I2,I5,ICC,ICHAR,TEND
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
C
      NUM(36) = NUM(36) + 1
C
C
C     ***CHECK ONLY IF DISSOLVED CONSTITUENT IS PRESENT
      IF (COINI.LE.0.) RETURN
C
C
C     ***IMPROVE ACCURACY IF CONSTITUENT CHARACTERISTICS LEAVE OIL
      IF (IRR(10).EQ.1) THEN
        IRR(10) = 0
        IF (ABS(ZC(I2,2)-PZ(2)).LT.0.005.OR.DTR(10).LT.0.) I2=I2+1
      END IF
C
C     ***GENERATE CONSTITUENT CHARACTERISTICS FOR GREEN-AMPT
C     ***CASES WITH DECAYING PONDING (IAT.EQ.4)
      IF (IAT.EQ.4.AND.TT.GT.TC(NEQ(3)).AND.CHS(2).GT.0.) THEN
        CALL RECHAR (35,70,PT(1),ZZ,ND,TC,C0,ZC,CC,NEQ)
        NEQ(3) = NEQ(3)+1
        TC(NEQ(3)) = TT
        CALL SFLUX (TT,QO)
        CPART = QO*COINI/(QO + XXK1*QW)
        C0(NEQ(3)) = CPART
        CC(NEQ(3),2) = CPART
        CC(NEQ(3),1) = CPART
        I1 = NEQ(3)
      END IF
C
C
C     ***VOLATILIZATION OF ENTIRE MASS OF CONSTITUENT
      IF (IRR(3).EQ.1) THEN
        IF (ZV(2).GE.ZC(1,2)) THEN
          IRR(3) = 2
          NEQ(3) = 0
          NEQ(1) = 1
        ELSE
          IRR(3) = 0
        END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE DHCHK2 (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ****************************************************************
C     *
C     * DCHK2  IMPLEMENTS CHANGES IN DISSOLVED HYDROCARBON PROPERTIES
C     *        FOR ACCEPTED STEPS
C     *
C     *
C     *
C     * INPUT ARGUMENTS.....................................
C     *     SAME AS SUBROUTINE CHK
C     * OUTPUT ARGUMENTS....................................
C     *     SAME AS SUBROUTINE CHK
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  CHK
C     *
C     ****************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      DOUBLE PRECISION FL,XML,XMO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /HYDI/ HDC1,HDC2,HDC3,HDC4
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /STOI/ BDFC,BDFH,SOLC,SOLH,STOICC,STOICH
      NUM(63) = NUM(63) + 1
C
C
C     ***DISSOLVED HYDROCARBON CHECKS
C     ****START THE CURVED CHARACTERISTICS
      IF (IRR(18).EQ.2) THEN
        IRR(18) = 3
        NEQ(2) = 101
        Z = PZ(2)
        DZ = (PZ(2)-DPA)/100.
        DS = (PS(2)-SPR)/89.
        DO 1 I=101,91,-1
C         ***SET UP CHARACTERISTICS
          T0(I) = TPE
          S0(I) = PS(2)
          SI(I,2) = PS(2)
          ZI(I,2) = Z
          ZZ(I,2) = Z
          Z = Z - DZ
 1      CONTINUE
        S = PS(2)
        DO 2 I=90,1,-1
C         ***ADJUST SATURATION DISTRIBUTION--DRAINAGE WAVE
          T0(I) = TPE
          S0(I) = S
          SI(I,2) = S
          ZI(I,2) = DPA
          ZZ(I,2) = DPA
          S = S - DS
 2      CONTINUE
        RETURN
      END IF
C
C
C     ****DETERMINE THE NEW OIL SATURATIONS FOR ACCEPTED STEPS
C
C
C     ****CONSTANT SATURATION FOR INSOLUBLE OIL PHASES
      IF (SOLH.LE.0.) RETURN
C     ****INACTIVATE ROUTINE AT TIMES BEFORE ENDING OF OIL EVENT
      IF (TT.LE.TPE) RETURN
C
C
C     ****INTEGRAL MASS BALANCE RELATION DETERMINES AVERAGE MASS LOSS
C     *****COMPONENTS OF THE INTEGRAL MASS BALANCE
C     *****OIL PHASE MASS ADDED ACROSS BOUNDARY
      T1 = TT
      IF (T1.GT.TPE) T1 = TPE
      IF (IAT.EQ.1) THEN
        XMO = DBLE(QP*(T1-TPB)*PRHO)
      ELSE IF (IAT.EQ.2) THEN
        XMO = DBLE(PVOL*PRHO)
      ELSE IF (IAT.EQ.3.OR.IAT.EQ.4) THEN
        XMO = DBLE(ZZ(2,6)*PRHO)
      END IF
c     ***subtract mass not in oil phase
      zdh = max(pz(2), dpl + HDC1*(PT(2)-TPB) )
      xmo = xmo - dble(zdh*hdc3)
C
C     *****WATER AND SORBEB MASS
C     ***SELECT MAXIMUM DEPTH OF THE OIL FRONT, THE DEPTH WATER
C     ***HAS CARRIED THE DISOLVED HYROCARBON
      DZ = MAX(  PZ(2)-PZ(1) ,  (PT(2)-PT(1))*HDC1 )
C     ***HYDROCARBON DISSOLUTION WITHIN LENS IS HANDLED BY OILENS
C     ***THE DISSOLUTION ABOVE THE WATER TABLE IS PROPORTIONAL TO
C     ***THE DISSOLUTION OCCURING THROUGHOUT THE PROFILE (IF NO
C     ***OIL LENS HAD FORMED),
C     *** BUT ONLY THE WATER FLUX CONTRIBUTES TO DISSOLUTION
      IF (NEQ(4).GT.0.) DZ = (PT(2)-PT(1))*HDC1
      XML = DBLE(DZ*HDC3)
C
C
C     *****FRACTION LOST
      FL = XML/XMO
C
C
```

```
C     *****CORRECTION OF SATURATIONS USING FRACTION REMAINING (1.-FL)
      DO 5 I=1,NEQ(2)
        SI(I,2) = SNGL(DBLE(SI(I,2))*(1.-FL))
        S0(I)   = SI(I,2)
 5    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE LCHK (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ****************************************************************
C     *                                                              *
C     * LCHK                                                         *
C     *                                                              *
C     * LCHK PERFORMS THE CHECKS FOR THE LENS EQUATIONS AND COMPUTATIONS
C     *
C     * INPUT ARGUMENTS.....................................
C     *   TT      = SOLUTION TIME JUST OBTAINED
C     *   ZZ(I,J) = DEPTH FOR EQUATION I GROUP J
C     *   Z1(I,J) = PREVIOUS DEPTH FOR EQUATION I GROUP J
C     *   F(I,J)  = CURRENT FUNCTION VALUES
C     *   ND      = ROW DIMENSION OF ZZ(),Z1(), & F()
C     *   NEQ(J)  = NUMBER OF EQUATION IN EQCH GROUP
C     *   DMAX    = MAXIMUM TIME STEP
C     *   DT      = CURRENT TIME STEP
C     *   NS      = NUMBER OF STEPS
C     *   IRP     = 1 IF MRKF12 ALREADY HAS REJECTED THE STEP
C     *
C     * OUTPUT ARGUMENTS.....................................
C     *   NEQ(J)  = ADJUSTED NUMBER OF EQUATIONS
C     *   DMAX    = ADJUSTED MAX. TIME STEP
C     *   DT      = NEW TIME STEP FOR ACCEPTED STEPS
C     *   DTR()   = SERIES OF PROPOSED REDUCED TIME STEPS
C     *   IC      = 1 TO END SIMULATION
C     *   IRR()   = CODES CORESPONDING TO VARIOUS REDUCED STEPS
C     *   IR      = 1 TO REJECT STEP
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES: NONE
C     * CALLED BY CHK
C     *
C     ****************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      DOUBLE PRECISION GGA1,GGA2,GGA3,GGA4
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /GAE2/ GGA1,GGA2,GGA3,GGA4
      COMMON /GLMB/ CCAL,CCMD,CCRA,CCRB,CHLENS,CHLOSS,CHRES
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      NUM(39) = NUM(39) + 1


      ***TRIGGERING LENS
      DT = PT(2)-PT(1)
      IF (IRR(14).EQ.0) THEN
        IF (ZZ(1,1).GT.WTABLE-FRING3) THEN
          SL = (ZZ(1,1)-Z1(1,1))/DT
          IF (SL.GT.0.) THEN
            DTR(14) = (WTABLE-FRING3-Z1(1,1))/SL
            IR = 1
          END IF
        END IF
      ELSE IF (IRR(14).EQ.2) THEN
C       ***OIL FILLING CAPILLARY FRINGE BEFORE MOVING OUTWARD
        VF = ZZ(1,4)
        CF = ZZ(1,5)
        CHLENS = VF*PRHO*DMKG/DVSV
        IF (IRR(15).EQ.0) THEN
C         IF (IAT.EQ.3) THEN
C           ***REMOVE CAPILLARY SUCTION TERM FROM GREEN-AMPT FLUX c  C   C
C              ***DETERMINATION
C           ***SET CONSTANTS FOR USE IN SUBROUTINE GAHS
C           GA1 = SPMAX*ETA/XKPMX
C           GA2 = ALOG( HS )
C           GA3 = HS
C           GA4 = GA3*GA2
C           GGA1 = DBLE(SPMAX*ETA/XKPMX)
C           GGA3 = DBLE(HS)
C           GGA2 = DLOG(GGA3)
C           GGA4 = GGA3*GGA2
C         END IF
          IF (VFRING.GT.VF) THEN
            VIN2 = VF
            CIN2 = CF
          ELSE
            IR = 1
            SL = (VF-VIN1)/DT
            DTR(15) = (VFRING-VIN1)/SL
          END IF
        ELSE IF (IRR(15).EQ.1) THEN
C         ***DOUBLE CHECK TO SEE IF THE FRINGE VOLUME IS CORRECT
C         ***AFTER REJECTION OF A STEP (TO COMPLETE THE FILLING OF
C         ***THE FRINGE
          IF (VFRING.GT.5.) THEN
            TOL15 = 1.E-5
          ELSE
            TOL15 = 1.E-4
          END IF
          IF (ABS((VFRING-VF)/VFRING).GT.TOL15) THEN
C           ***REJECT THE STEP AGAIN (THE LINEAR INTERPOLATION WAS
C           ***NOT ACCEPTABLE)
            IR = 1
            SL = (VF - VIN1)/DT
```

```
              DTR(15) = (VFRING - VIN1)/SL
              IRR(15) = 0
            END IF
          END IF
        END IF
C
C
C
C     ***CONTAMINANT REACHES LENS OR WATER TABLE
      WW = WTABLE-ZZ(2,4)-FRING3
      IF (IRR(9).EQ.0.AND.COINI.GT.0.0) THEN
        IF (ZZ(1,3).GT.WW) THEN
          SL1 = (ZZ(1,3)-Z1(1,3))/DT
          W1  = WTABLE-Z1(2,4)-FRING3
          SL2 = (WW-W1)/DT
          DTR(9) = (W1-Z1(1,3))/(SL1-SL2)
          IR = 1
        ELSE IF (ABS(ZZ(1,3)-WW)/WW.LE.1.E-5) THEN
C         ***FORTUITOUS SATSIFACTION OF THE CONDITION
          IRR(9) = 1
        END IF
      ELSE IF (IRR(9).EQ.1.AND.
     *         ZZ(1,3).GT.WW.AND.
c    *         ABS((ZZ(1,3)-WW)/WW).GE.1.E-5.AND.
     *         COINI.GT.0.0) THEN
C       ***THE LINEAR INTERPOLATION HAS MISSED
        IRR(9) = 0
        SL1 = (ZZ(1,3)-Z1(1,3))/DT
        W1  = WTABLE-Z1(2,4)-FRING3
        SL2 = (WW-W1)/DT
        DTR(9) = 0.25*(W1-Z1(1,3))/(SL1-SL2)
        IR = 1
      END IF
C
      IF (NEQ(4).EQ.0) RETURN
C
C     ***OIL BACKS UP TO SURFACE
      IF (IRR(16).EQ.0.AND.IRR(14).EQ.3) THEN
        IF (ZZ(2,4).GT.WTABLE-FRING3) THEN
          SL = (ZZ(2,4)-Z1(2,4))/DT
          DTR(16) = (WTABLE-FRING3-Z1(2,4))/DT
          IR = 1
        END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE LCHK2(TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C     ****************************************************************
C     *                                                              *
C     * LCHK2                                                        *
C     *                                                              *
C     * LCHK2 MAKES ADJUSTMENTS TO THE LENS EQUATIONS AFTER A STEP HAS *
C     * BEEN REJECTED BY A CHECK IN LCHK                             *
C     *                                                              *
C     * INPUT ARGUMENTS.....................................         *
C     *   TT      = SOLUTION TIME JUST OBTAINED                      *
C     *   ZZ(I,J) = DEPTH FOR EQUATION I GROUP J                     *
C     *   Z1(I,J) = PREVIOUS DEPTH FOR EQUATION I GROUP J            *
C     *   F(I,J)  = CURRENT FUNCTION VALUES                          *
C     *   ND      = ROW DIMENSION OF ZZ(),Z1(), & F()                *
C     *   NEQ(J)  = NUMBER OF EQUATION IN EQCH GROUP                 *
C     *   DMAX    = MAXIMUM TIME STEP                                *
C     *   DT      = CURRENT TIME STEP                                *
C     *   NS      = NUMBER OF STEPS                                  *
C     *   IRP     = 1 IF MRKF12 ALREADY HAS REJECTED THE STEP        *
C     *                                                              *
C     * OUTPUT ARGUMENTS.....................................        *
C     *   NEQ(J)  = ADJUSTED NUMBER OF EQUATIONS                     *
C     *   DMAX    = ADJUSTED MAX. TIME STEP                          *
C     *   DT      = NEW TIME STEP FOR ACCEPTED STEPS                 *
C     *   DTR()   = SERIES OF PROPOSED REDUCED TIME STEPS            *
C     *   IC      = 1 TO END SIMULATION                              *
C     *   IRR()   = CODES CORESPONDING TO VARIOUS REDUCED STEPS      *
C     *   IR      = 1 TO REJECT STEP                                 *
C     *                                                              *
C     * JIM WEAVER                                                   *
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY             *
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY                *
C     * ADA, OKLAHOMA  74820                                         *
C     *                                                              *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77               *
C     * REQUIRED ROUTINES: HMASS,CMASS,MESSAG,MAXSAV                 *
C     * CALLED BY CHK                                                *
C     *                                                              *
C     ****************************************************************
      DIMENSION DTR(*),F(ND,*),IRR(*),NEQ(*),ZZ(ND,*),Z1(ND,*)
      CHARACTER NT*15
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LNSS/ ILSTOP
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /NAME/ NT(15)
      COMMON /RLNS/ HHS(100),NR,RS(100)
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
C     DATA ISTOP /0/
C
      NUM(40) = NUM(40) + 1
      IF (NUM(40).EQ.1) ISTOP = 0
C
C     ***FOR ACCEPTED STEPS
C
C     ***RECORD MAXIMUM RATE OF LENS HEIGHT CHANGE FOR USAGE IN HMASS
      IF (F(2,4).GT.RHMAX) RHMAX = F(2,4)
      DT = PT(2)-PT(1)
C
C     ***LENS TRIGGERED
      IF (IRR(14).EQ.1) THEN
        IF (WTABLE-ZZ(2,4)-FRING3.GT.ZZ(1,1)) THEN
C         ***LENS HAS REALLY NOT FORMED YET, TRY AGAIN
C         ***THE FRONT, ZZ(1,1), IS NOT DEEP ENOUGH
          IRR(14) = 0
        ELSE
C         ***LENS HAS BEEN TRIGGERED
          HLTIME = TT
C         ***HEADINGS FOR THE OILENS OUTPUT
          IF (IWR.EQ.1) THEN
            WRITE (12,9600) (NT(I),I=1,15)
            WRITE (12,9620)
            WRITE (12,9630) HLTIME,SOLENS,FRING3
          END IF
          IRR(14) = 2
          CALL MESSAG (7)
        END IF
      END IF
C
      IF (IRR(15).EQ.1) THEN
C       ***CHECK FOR INITIAL CONTAMINANT MASS NEAR THE WATERTABLE
c       CALL CONCE (PT(2),WTABLE-FRING3,SWMAX,0.,NEQ(3),C2,XX,0)
c       CALL CONCE (PT(2),WTABLE,SWMAX,0.,NEQ(3),C1,XX,0)
c       CM = 0.5*(C2+C1)*RD(SO,SWMAX)*PI*RADI*RADI*FRING3
C       ***INITIAL VOLUME OF OIL IN THE CAPILLARY FRINGE
        VIN2 = VFRING
        NEQ(4) = 3
        IRR(15) = 3
        IRR(14) = 3
        DT = 0.001
        T2 = PT(2)-DTPR
        IF (KKSTOP.EQ.2) KSTOP = KKSTOP
      END IF
C
C
C
C     ***CONTAMINANT REACHES OIL LENS OR WATER TABLE
      IF (IRR(9).EQ.1) THEN
C       ***CHECK TO MAKE SURE THAT THE CONTAMINANT HAS
C       ***ACTUALLY REACHED THE TOP OF THE OIL LENS
C       ***IN SOME CASES THE CONTAMINANT APPEARS TO REACH
C       ***THE TOP OF THE LENS BUT THE LINEAR INTERPOLATION
C       ***OF THE NEW TIME STEP IS NOT ACCURATE ENOUGH
        ZLENS = WTABLE-ZZ(2,4)-FRING3
C       IF (ZLENS.GT.ZZ(1,3)) THEN
```

```
        IF (ABS((ZLENS-ZZ(1,3))/ZLENS).GT.1.E-5) THEN
C         ***THE CONTAMINANT HAS NOT REALLY REACHED THE TOP OF THE LENS
          IRR(9) = 0
        ELSE
C         ***THE CONTAMINANT HAS REALLY REACHED THE TOP OF THE LENS
          NEQ(5) = 2
          IRR(9) = 2
          rmc = 0.
          zz(1,5) = 0.
          IF (KKSTOP.EQ.3.OR.KKSTOP.EQ.4) KSTOP = KKSTOP
          CALL MESSAG (9)
          CLTIME = TT
        END IF
      END IF
C
C
C
C     ***CONTAMINANT MASS BALANCE
      CALL CMASS (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C
      IF (IRR(15).NE.3.OR.ZZ(2,4).LE.0.) RETURN
C
C     ***OIL BACKS UP TO SURFACE
      IF (IRR(16).EQ.1) THEN
        IC = 1
        IRR(16) = 2
      END IF
C
C
C     ***** LAST CONDITION ADDED ON VII-13-89...DWC *****
C     ***STOPPING TIME OF THE LENS SPREAD
      IF (ZZ(3,4).LE.Z1(3,4)
     *.AND.ISTOP.EQ.0
     *.AND.ZZ(2,4).LT.Z1(2,4)
     *.AND.RHMAX.GT.F(2,4)) ILSTOP = 1
C     ***ILSTOP CAN ALSO BE SET TO 1 IN OILENS IF F(3,4) GOES NEGATIVE
      IF (ILSTOP.EQ.1) THEN
C       ***OIL LENS SPREAD HAS CEASED
        IF (KSTOP.EQ.2) THEN
          IR = 1
          IC = 1
        END IF
        DO 50 I=1,4
          NEQ(4) = 0
 50     CONTINUE
C       IF (IWR.EQ.1) WRITE (12,9100) PT(2)
        CALL MESSAG (8)
        ISTOP = 1
        ILSTOP = 2
        RETURN
      END IF
C
C
C
C     ***HYDROCARBON MASS BALANCE CHECKS
      IF (ISTOP.EQ.0)
     1CALL HMASS (TT,ZZ,Z1,F,ND,NEQ,DMAX,DT,DTR,NS,IC,IRR,IR,IRP)
C
C     ***SAVE MAXIMUM HEIGHT OF THE LENS
      IF (ISTOP.EQ.0) CALL MAXSAV (RADI,ZZ(3,4),ZZ(2,4),RS,HHS,NR)
C
C     ***OIL BACKS UP TO THE SURFACE
      IF (IWR.EQ.1.AND.IRR(16).EQ.2) WRITE (12,9110) PT(2)
C
C     ***CONTAMINANT REACHES OIL LENS OR WATER TABLE
      IF (IRR(9).EQ.2) THEN
        NEQ(5) = 2
        IRR(9) = 3
      END IF
C
      RETURN
 9100 FORMAT (/10X,'OIL LENS MOTION STOPS AT TIME ',F10.4/)
 9600 FORMAT (1H1,/15X,50(1H*)/
     1        15X,'*    OILENS MODEL OUTPUT--OIL LENS DESCRIPTION   *'/
     2        15X,'**********************************************'/
     3        15X,5A10/15X,5A10/15X,5A10/)
 9620 FORMAT (//, 17X,' LENS    ',' LENS    ',' LENS    ',' RESIDUAL',
     * ' VOLUME','   CUM.   ','  PER CENT'/,
     * 8X, '  TIME   ',' HEIGHT ','  RADIUS ','  VOLUME ',
     * '  VOLUME ', '  LOSSES ', '  INFLOW ','  VOLUME'/
     * '   STEP ',' (DAYS)  ','(METERS)',' (METERS)',
     * '  (CU.M.)','  (CU.M.)','  (CU.M.)','  (CU.M.)','  ERROR'/
     * 1X,79(1H=)/)
 9630 FORMAT (/10X,'*** OIL FILLING CAPILLARY FRINGE'/
     *         10X,'*** TIME                          = ',F10.4/
     *         10x,'*** OIL SATURATION IN LENS        = ',F10.4/
     *         10X,'*** CAPILLARY FRINGE OIL THICKNESS = ',F10.4/)
 9110 FORMAT (/10X,'OIL BACKS UP TO THE SURFACE AT TIME ',F10.4/)
      END
```

```
      SUBROUTINE RECHAR (NADD,LVALUE,TT,ZZ,ND,TC,C0,ZC,CC,NEQ)
C     ************************************************************
C     *
C     *  RECHAR REDUCES THE NUMBER OF CHARACTERISTICS USED FOR
C     *  THE DISSOLVED CONSTITUENT, IF THE NUMBER EXCEEDS THE
C     *  ALLOWABLE STORAGE
C     *
C     *  INPUT ARGUMENTS.....................................
C     *  NADD    = NUMBER OF CHARACTERISTICS ABOVE THE STANDARD 15
C     *            TO HAVE IN THE COMPLETED ARRAY
C     *  LVALUE  = RETURN IF NUMBER OF CHARACTERISTICS LESS THAN LVALUE
C     *  TT      = TIME
C     *  ZZ(I,J) = DEPTH FOR EQUATION I OF GROUP J
C     *  ND      = ROW DIMENSION OF ZZ
C     *  TC()    = TIME ORIGINS OF THE CHARACTERISTICS
C    *  C0()    = CONCENTRATIONS AT ORIGINS OF THE CHARACTERISTICS
C     *  ZC(I,J) = DEPTHS OF TWO ACTIVE POINTS ON CHARACTERISTICS
C     *  CC(I,J) = CONCENTRATIONS OF TWO ACTIVE POINTS ON CHARACTERISTIC
C     *  NEQ(J)  = ARRAY OF NUMBER OF EQUATIONS FOR EACH GROUP
C     *
C     *  OUTPUT ARGUMENTS.....................................
C     *  ZZ(I,J) = REVISED DEPTH FOR EQUATION I OF GROUP J
C     *  TC()    = REVISED TIME ORIGINS OF THE CHARACTERISTICS
C     *  C0()    = CONCENTRATIONS AT ORIGINS OF THE CHARACTERISTICS

C     *  CC(I,J) = CONCENTRATIONS OF TWO ACTIVE POINTS ON CHARACTERISTIC
C     *  ZC(I,J) = DEPTHS OF TWO ACTIVE POINTS ON CHARACTERISTICS
C     *  NEQ(J)  = REVISED ARRAY OF NUMBER OF EQUATIONS FOR EACH GROUP
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *  IIX-19-92
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES: BINARY
C     *  THIS ROUTINE IS CALLED BY: CCHK2,GACHK2
C     ************************************************************
      DIMENSION C0(*),CC(ND,*),TC(*),ZC(ND,*),ZZ(ND,*)
      INTEGER NEQ(*)
      DIMENSION CN(16:101),CNN(16:101),CNN2(16:101),TN(16:101),
     *ZA(16:101),ZB(16:101)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
C
      IF (NEQ(3).LT.LVALUE) RETURN
C
C     ***INTERPOLATE NEW VALUES BASED ON EXISTING ARRAY
C     ***(TPEM IS USED INSTEAD OF TPE, BECAUSE TPE IS SET TO AN ARBITRARY
C     ***VALUE WHEN IAT = 4)
      DT = (TT-TPEM)/FLOAT(NADD)
      J = 15
      DO 10  TIME = TPEM,TT,DT
        J = J+1
        CALL BINARY (TIME,TC,NEQ(3),1,IN)
        DT1 = TC(IN+1)-TC(IN)
        DT2 = TIME-TC(IN)
C       ***THE SURFACE CONCENTRATION
        SL = (C0(IN+1)-C0(IN))/DT1
        CN(J) = C0(IN) + SL*DT2
        TN(J) = TIME
C       ***ONLY CURRENT VALUES ARE INTERPOLATED SINCE
C       ***THESE ARE ABOUT TO BE SAVED BECAUSE RECHAR
C       ***IS ONLY USED FOR ACCEPTED STEPS
C       ***THE CHARACTERISTIC CONCENTRATION
        SL = (CC(IN+1,2)-CC(IN,2))/DT1
        CNN(J) = CC(IN,2) + SL*DT2
C       ***THE PREVIOUS CHARACTERISTIC CONCENTRATION
        SL = (CC(IN+1,1)-CC(IN,1))/DT1
        CNN2(J) = CC(IN,1) + SL*DT2
C       ***THE ACTIVE CHARACTERISTIC DEPTH
        SL = (ZZ(IN+1,3)-ZZ(IN,3))/DT1
        ZA(J) = ZZ(IN,3) + SL*DT2
C       ***THE ACTIVE CHARACTERISTIC DEPTH NICKNAME
        ZB(J) = ZA(J)
 10   CONTINUE
C
C     ***SAVE RESULTS IN ARRAYS
      DO 20 J=16,15+NADD
        TC(J) = TN(J)
        C0(J) = CN(J)
        CC(J,2) = CNN(J)
        CC(J,1) = CNN2(J)
        ZZ(J,3) = ZA(J)
        ZC(J,2) = ZB(J)
 20   CONTINUE
C
      NEQ(3) = 15+NADD
C
      RETURN
      END
```

```
      SUBROUTINE ZMASS (TH,ZH,SH,NEQ,IW,ZZ,ND,VVM,CVL)
C     ************************************************************
C     *                                                          *
C     *   ZMASS                                                  *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE CREATES SATURATION AND CON- *
C     *   CENTRATION PROFILES AT GIVEN TIMES.  THE MASS OF OIL AND *
C     *   CONSTITUENT MASS IN THE PROFILE IS OBTAINED BY GAUSS   *
C     *   QUADRATURE OF THE PROFILE (ROUTINE OCP).               *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................   *
C     *   TH      = TIME                                         *
C     *   ZH      = MAXIMUM DEPTH OF OIL                         *
C     *   SH      = OIL SATURATION AT ZH                         *
C     *   NEQ     = ARRAY CONTAINING NUMBER OF EQUATIONS         *
C     *   ZZ(,)   = MATRIX OF DEPENDENT VARIABLES                *
C     *   ND      = ROW DIMENSION OF ZZ                          *
C     *   IW      = WRITE RESULTS INTO FILE PROF IF IW=1         *
C     *                                                          *
C     *   OUTPUT ARGUMENTS.....................................  *
C     *   VVM     = VOLUME/AREA OF OIL                           *
C     *   CVL     = MASS/AREA OF CONSTITUENT                     *
C     *                                                          *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...CONCE,OIL,OCP,DSO,GLOBAL,ZLOC,MESS2 *
C     *   THIS ROUTINE IS CALLED BY:  OILENS, CHK                *
C     *                                                          *
C     *   X-21-87, VII-18-88                                     *
C     *   10-17-93  BROKEN INTO SUBROUTINES (ZLOC,GLOBAL)        *
C     *                                                          *
C     ************************************************************
      DIMENSION NEQ(*),ZZ(ND,*),ZX(10)
      CHARACTER NT*15
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /NAME/ NT(15)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(10) = NUM(10) + 1
C
C
C     ***SELECT DEPTHS WITHIN THE KOPT PROFILE
      CALL ZLOC (TH,ZH,SH,NEQ,ZZ,ND,ZX,NN,SS)
C
C
C     ***BEGIN PROFILING
C     ***SCREEN MESSAGE IF PRINTED PROFILE
      IF (IWR.EQ.1.AND.IW.EQ.1) CALL MESS2 (15,TH)
      VVM = 0.
      CVL = 0.
      DHM = 0.
      CALL CONCE (TH,DPA,SWMAX,SS,NEQ(3),CCC,CXX,1)
      CALL DSO (TH,DPA,NEQ(2),COW,CH)
C
      IF (IWR.EQ.1.AND.IW.EQ.1) THEN
        WRITE (9,9100) TH,(NT(II),II=1,15)
        WRITE (9,9106)
        WRITE (9,9107) DPA,SS,CCC*XXK1,COW
        WRITE (16,9301) TH
        WRITE (16,9302) DPA,SS
      END IF
C
C
C
C     ***KOPT OIL AND CONSTITUENT PROFILE
      DO 100 I=1,NN-1
C
        OVOL = 0.
        CMASS = 0.
        DHMASS = 0.
        ZPE = ZX(I)*1.000001
        ZME = ZX(I+1)*0.999999
C
C
C       ***ADD POINT JUST DEEPER THAN BEGINNING POINT TO PRINTED PROFILE
        IF (IWR.EQ.1.AND.IW.EQ.1) THEN
          CALL OIL (ZPE,TH,SWMAX,NEQ(2),SSS)
          CALL CONCE (TH,ZPE,SWMAX,SSS,NEQ(3),CCC,CXX,1)
          CALL DSO (TH,ZPE,NEQ(2),COW,CH)
          WRITE (9,9107) ZX(I),SSS,CCC*XXK1,COW
          WRITE (16,9302) ZX(I),SSS
        END IF
C
C
C       ***DETERMINE POINTS ON PROFILE
C       ***CALCULATE INCREMENTS OF VOLUME AND MASS
        CALL OCP (TH,NEQ,ZX(I),ZX(I+1),IW,OVOL,CMASS,DHMASS)
C
```

```
C
C       ***ADD POINT JUST SHALLOWER THAN ENDING POINT TO PRINTED PROFILE
        IF (IWR.EQ.1.AND.IW.EQ.1) THEN
          CALL OIL (ZME,TH,SWMAX,NEQ(2),SSS)
          CALL CONCE (TH,ZME,SWMAX,SSS,NEQ(3),CCC,CXX,1)
          CALL DSO (TH,ZME,NEQ(2),COW,CH)
          WRITE (9,9107) ZX(I+1),SSS,CCC*XXK1,COW
          WRITE (16,9302) ZX(I+1),SSS
        END IF
C
C
C       ***CUMULATE OIL AND CONSTITUENT MASS PER UNIT AREA
        DHM = DHM + DHMASS
        VVM = VVM + OVOL*ETA*PRHO*DMKG/DVSV
        CVL = CVL + CMASS*CMKG/CVSV
C
 100  CONTINUE
C
      IF (IWR.EQ.1.AND.IW.EQ.1) THEN
        WRITE (16,9303)
      END IF
C
C     ***GLOBAL MASS BALANCE REPORTING
      IF (IWR.EQ.1.AND.IW.EQ.1) THEN
        CALL GLOBAL (TH,ZZ,ND,NEQ,VVM,DHM,CVL)
      END IF
C
C
      RETURN
 9100 FORMAT (1H1/10X,'SATURATION AND CONCENTRATION PROFILE AT ',F10.4/
     2        10X,50(1H*)/
     3        3(10X,5A10/)/)
 9106 FORMAT (15X,3X,'DEPTH',2X,5X,4X,'SAT.',2X,'CONC.(WATER)',
     *            ' DISSOL. NAPL CONC'/
     *        15X,53(1H=))
 9107 FORMAT (15X,F10.4,5X,3F10.4)
 9301 FORMAT (F10.4)
 9302 FORMAT (2F10.4)
 9303 FORMAT ('STOP')
      END
```

```
      SUBROUTINE ZLOC (TH,ZH,SH,NEQ,ZZ,ND,ZX,NN,SS)
C     ************************************************************
C     *
C     *   ZLOC
C     *
C     *   ZLOC DETERMINES THE DEPTHS FOR KOPT PROFILE POINTS.
C     *   THE ROUTINE DETERMINES POINTS THAT ARE NECESSARY
C     *   FOR KOPT PROFILES TO GIVE ACCURATE MASS BALANCES.
C     *   THE POINTS ARE GENERATED IN ARBITRARY ORDER, ZEROS
C     *   AND DUPLICATES ARE ELIMINATED AND THE POINTS ARE
C     *   SORTED.
C     *
C     *
C     *   INPUT ARGUMENTS.....................................
C     *   TH      = TIME
C     *   ZH      = MAXIMUM DEPTH OF OIL
C     *   SH      = OIL SATURATION AT ZH
C     *   NEQ     = ARRAY CONTAINING NUMBER OF EQUATIONS
C     *   ZZ(,)   = MATRIX OF DEPENDENT VARIABLES
C     *   ND      = ROW DIMENSION OF ZZ
C     *
C     *   OUTPUT ARGUMENTS.....................................
C     *   ZX()    = ARRAY OF KOPT PROFILING LOCATIONS
C     *   NN      = NUMBER OF LOCATIONS SELECTED
C     *   SS      = OIL SATURATION AT THE SURFACE
C     *
C     *   JIM WEAVER
C     *   ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *   UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     *   ADA, OKLAHOMA  74820
C     *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *   REQUIRED ROUTINES: NONE
C     *   CALLED BY:  ZMASS
C     *
C     *   10-17-93
C     *
C     ************************************************************
      DIMENSION NEQ(*),ZZ(ND,*),ZX(*)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FLAG/ ICONC,IKOPT,ILENS,ITSGP
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /GLMB/ CCAL,CCMD,CCRA,CCRB,CHLENS,CHLOSS,CHRES
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /HYDI/ HDC1,HDC2,HDC3,HDC4
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /STOI/ BDFC,BDFH,SOLC,SOLH,STOICC,STOICH
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
C
      ZADH = 0.0
C
C
C     ***ZERO ARRAY OF PROFILE DIVISIONS, ZX(10)
      NN = 1
      DO 5 II=1,10
        ZX(II) = 0.
 5    CONTINUE
C
C
      IF (TH.GT.TPE) THEN
        SS = SPR
        IF (DLO.GT.0..AND.TH.GT.OLAG)
     1  SS = SPR*EXP( - DLO*(TH-OLAG))
      ELSE
        SS = SPMAX
      END IF
C
C
C     ***DIVIDE THE PROFILE INTO SUBSECTIONS FOR INTEGRATION
C     ***OIL LOCATIONS
      NN = NN + 1
      ZX(NN) = ZH
      IF (SH.GE.SPMAX.AND.DLO.LE.0.) THEN
C
C       ***STRAIGHT CHARACTERISTICS BEFORE END OF SPMAX
        T1 = TPE
        IF (CHS(2).GT.0..AND.IGA.EQ.1) T1 = 2.*TH
C
        IF (TH.GT.T1) THEN
          NN = NN + 1
          ZD = DPA + (TH-TPE)*DZPDDT
          ZX(NN) = ZD
          NN = NN + 1
          ZX(NN) = DPA + 0.3333*(ZD-DPA)
          NN = NN + 1
          ZX(NN) = DPA + 0.6666*(ZD-DPA)
        END IF
C
      ELSE IF (DLO.GT.0.) THEN
C
```

```
      IF (TH.GT.TPE.AND.NEQ(2).GT.91.AND.ZI(91,2).GT.DPA
     1  .AND.ZH.GT.ZI(91,2))THEN
C       ***CURVED CHARACTERISTICS BEFORE APPROX. END OF SPMAX
        NN = NN + 1
        ZX(NN) = ZI(91,2)
        NN = NN + 1
        ZX(NN) = DPA + (ZI(91,2)-DPA)*0.5
      END IF
C
     ELSE
C
C       ***OTHERWISE
      DZ = (ZX(NN)-DPA)/3.
      NN = NN + 1
      ZX(NN) = DPA + DZ
      NN = NN + 1
      ZX(NN) = DPA + DZ + DZ
C
     END IF
C
C
C
     IF (COINI.GT.0.) THEN
C       ***CONSTITUENT PROFILE LOCATIONS
      NN = NN + 1
      ZX(NN) = ZC(1,2)
      ZCT = ZC(NEQ(3),2)
      ZCV = ZV(2)
      IF (ZCV.GT.ZCT) ZCT = ZCV
      NN = NN + 1
      ZX(NN) = ZCT
     END IF
C
C    ***EXTRA LOCATION FOR DISSOLVED OIL, IF NEEDED
     IF (SOLH.GT.0.) THEN
      ZADH = DPL + HDC1*(PT(2)-TPB)
      IF (ZADH.GT.ZH) THEN
        NN = NN + 1
        ZX(NN) = ZADH
      END IF
     END IF
C
C    ***SORT THE DEPTHS
     NLAST = NN - 1
     DO 30 J=1,NN
      IH = 0
      DO 25  I=1,NLAST
        IF (ZX(I).GT.ZX(I+1)) THEN
          IH = 1
          ZTEMP = ZX(I)
          ZX(I) = ZX(I+1)
          ZX(I+1) = ZTEMP
        END IF
 25   CONTINUE
      IF (IH.EQ.0) GO TO 31
        NLAST = NLAST - 1
 30   CONTINUE
 31   CONTINUE
C
C    *** ELIMINATE THE EXTRA ZERO'S
     DO 35 I = 1,NN
       IF(ZX(I).GT.0.) GO TO 36
35   CONTINUE
36   IF(I.GT.1) THEN
       II = 2
       ZX(1) = 0.
       DO 38 J = I,NN
         ZX(II) = ZX(J)
         II = II + 1
38     CONTINUE
       NN = NN - I + 1
       NN = NN + 1
     ENDIF
C
C    ***STOP VERITCAL PROFILE WHEN OIL LENS PRESENT
     IF (ILENS.EQ.1.AND.NEQ(4).GT.0.OR.HLTIME.GT.0.) THEN
       DLENS = WTABLE-FRING3-ZZ(2,4)
       DO 40 I=1,NN
         IF (DLENS.LT.ZX(I)) THEN
           ZX(I) = DLENS
           NN = I
           GO TO 41
         END IF
 40    CONTINUE
 41    CONTINUE
     END IF
C
      RETURN
      END
```

```
      SUBROUTINE GLOBAL (TH,ZZ,ND,NEQ,VVM,DHM,CVL)
C     ************************************************************
C     *                                                          *
C     *  GLOBAL                                                  *
C     *                                                          *
C     *  GLOBAL PRINTS OUT A GLOBAL MASS BALANCE BASED ON INFORMATION
C     *  DEVELOPED IN S.R.S ZMASS, HMASS AND CMASS               *
C     *                                                          *
C     * INPUT ARGUMENTS.....................................     *
C     * TH       CURRENT SIMULATION TIME                        *
C     * ZZ(,)    ARRAY OF DEPENDENT VARIABLES                   *
C     * ND       ROW DIMENSION OF ZZ(,)                         *
C     * NEQ()    NUMBER OF EQUATIONS IN EACH COLUMN OF ZZ(,)    *
C     * VVM      OIL VOLUME FROM ZMASS                          *
C     * DHM      DISSOLVED OIL MASS FROM ZMASS                  *
C     * CVL      DISSOLVED CONSITUTENT MASS FROM ZMASS          *
C     * OUTPUT ARGUMENTS.....................(NONE).............*
C     *                                                          *
C     *                                                          *
C     * JIM WEAVER                                               *
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY         *
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY            *
C     * ADA, OKLAHOMA  74820                                     *
C     *                                                          *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77           *
C     * REQUIRED ROUTINES:  PROLNS                               *
C     * CALLED BY:  ZMASS                                        *
C     *                                                          *
C     *  10-17-93                                                *
C     ************************************************************
      DIMENSION NEQ(*),ZZ(ND,*)
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GLMB/ CCAL,CCMD,CCRA,CCRB,CHLENS,CHLOSS,CHRES
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
C
C
C     ***OUTPUT MASS BALANCE INFORMATION
C
C
C     ***WRITE MASSES FOR KOPT PORTION OF PROFILE
      WRITE (9,9115) VVM,DHM,CVL,CVL*AC
      VVM = VVM + DHM
      WRITE (9,9116) VVM*AC
C
C
C
C     ***OILENS PROFILING
      IF (NEQ(4).GT.1.OR.HLTIME.GT.0.) THEN
        CALL PROLNS (TH,ZZ(2,4),ZZ(3,4),1)
      END IF
C
C
C
C     ***GLOBAL OIL MASS CONSERVATION (I.E., KOPT AND OILENS)
C     ***CONVERT RUNOFF VOLUME/AREA TO MASS/AREA
      RO = ZZ(1,6)*PRHO*DMKG/DVSV
C     ***GLOBAL MASS
      GOMASS = AC*(RO+VVM)+CHLENS+CHRES+CHLOSS
C
C
C
C     ***TOTAL MASS ADDED AT BOUNDARY
      IF (IAT.EQ.1) THEN
        T1 = PT(2)
        IF (T1.GT.TPE) T1 = TPE
        TMASS = AC*(T1-TPB)*QP*PRHO*DMKG/DVSV
      ELSE IF (IAT.EQ.2) THEN
        TMASS = AC*PVOL*PRHO*DMKG/DVSV
      ELSE IF (IAT.EQ.3.OR.IAT.EQ.4) THEN
        TMASS = AC*ZZ(2,6)*PRHO*DMKG/DVSV
      END IF
C
C
C
      WRITE (9,9198) ZZ(1,4)*PRHO*DMKG/DVSV
C     WRITE (9,9199) CHLENS,CHRES,CHLOSS
      WRITE (9,9200) TMASS,GOMASS,100.*(GOMASS-TMASS)/TMASS
C
C
C
C     ***GLOBAL CONSTITENT MASS BALANCE
      IF (COINI.GT.0.) THEN
C       CALL CGMB ()
      END IF
C
C
      RETURN
 9115 FORMAT (//10X,'KOPT PROFILE MASS PER UNIT AREA:'/
     *          10X,'NAPL                          (KG/M/M) ',G10.4/
     *          10X,'DISSOLVED NAPL                (KG/M/M) ',G10.4/
     *          10X,'CONSTITUENT                   (KG/M/M) ',G10.4//
     *          10X,'KOPT PROFILE TOTAL MASS:'/
     *          10X,'CONSTITUENT                      (KG) ',G10.4)
 9116 FORMAT (10X,'NAPL                              (KG) ',G10.4)
```

```
9198 FORMAT (/10X,'CUMULATIVE INFLUX TO LENS ',G10.4/)
9199 FORMAT (/10X,'CHLENS ',G10.4/
    *         10X,'CHRES  ',G10.4/
    *         10X,'CHLOSS ',G10.4/)
9200 FORMAT (/10X,'KOPT AND OILENS GLOBAL MASS BALANCES'/
    *         10X,'TOTAL NAPL MASS ADDED AT BOUNDARY  (KG) ',G10.4/
    *         10X,'NAPL MASS RECOVERED BY MASS BALANCE (KG)',G10.4/
    *         10X,'PER CENT ERROR                          ',G10.4////)
     END
```

```
      SUBROUTINE OCP (TT,NEQ,A,B,IW,SUM1,SUM2,SUM3)
C     ************************************************************
C     *
C     * OCP
C     *
C     * MODIFIED GAUSS-LAGRANGE QUADRATURE FOR OIL AND CONSTITUENT
C     * II-17-87, METHOD FROM ABRAMOWITZ AND STEGUN P. 916
C     * CREATES AND PRINTS THE OIL AND CONSTITUENT PROFILE
C     * INTEGRATES MASS UNDER CURVES
C     *
C     *
C     * INPUT ARGUMENTS.....................................
C     *  TT     TIME IN DAYS
C     *  NEQ    ARRAY OF NUMBER OF EQUATION IN EACH GROUP
C     *  A      LOWER LIMIT OF INTEGRATION (SHALLOWEST DEPTH)
C     *  B      UPPER LIMIT OF INTEGRATION (DEEPEST DEPTH)
C     *  IW     WRITE FLAG (WRITE IF IW = 1)
C     * OUTPUT ARGUMENTS.....................................
C     *  SUM1   OIL PHASE VOLUME/UNIT AREA
C     *  SUM2   CONSTITUENT MASS/UNIT AREA
C     *  SUM3   DISSOLVED OIL MASS/UNIT AREA
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  CONCE,DSO,OIL
C     * CALLED BY  ZMASS
C     *
C     ************************************************************
      DIMENSION NEQ(*),CS(7),CY(7)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /COUN/ NUM(100)
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      DATA CY /-.949108,-.741531,-.405845,0.,.405845,.741531,.949108/
      DATA CS /.129485,.279705,.381830,.417959,.381830,.279705,.129485/
C
C
      NUM(9) = NUM(9) + 1
      XC1 = (B - A)*0.5
      XC2 = (B + A)*0.5
C
C
C     ***FUNCTION EVALUATIONS AT GAUSS POINTS
C     ***SUM1 = OIL PHASE VOLUME/UNIT AREA
C     ***SUM2 = CONSTITUENT MASS/UNIT AREA
C     ***SUM3 = DISSOLVED OIL MASS/UNIT AREA
      SUM1 = 0.
      SUM2 = 0.
      SUM3 = 0.
      DO 10 I=1,7
        Y = CY(I)*XC1 + XC2
        CALL OIL (Y,TT,SWMAX,NEQ(2),SO)
        CALL CONCE (TT,Y,SWMAX,SO,NEQ(3),CXO,CM,1)
        CALL DSO (TT,Y,NEQ(2),COW,CH)
        IF (IWR.EQ.1.AND.IW.EQ.1) WRITE (9,9101) Y,SO,CXO*XXK1,COW
        IF (IWR.EQ.1.AND.IW.EQ.1) WRITE (16,9302) Y,SO
        SUM1 = SUM1 + SO*CS(I)
        SUM2 = SUM2 + CM*CS(I)
        SUM3 = SUM3 + CH*CS(I)
 10   CONTINUE
      SUM1 = SUM1*XC1
      SUM2 = SUM2*XC1
      SUM3 = SUM3*XC1
C
 9101 FORMAT (15X,F10.4,5X,3F10.4)
 9302 FORMAT (2F10.4)
      RETURN
      END
```

```
      SUBROUTINE PROLNS (TIME,HO,RT,IW)
C     ****************************************************************
C     *                                                              *
C     * PROLNS                                                       *
C     *                                                              *
C     * PROLNS PERFORMS THE LENS PROFILING                           *
C     *                                                              *
C     * INPUT ARGUMENTS.....................................         *
C     *   TIME    CURRENT SIMULATION TIME                            *
C     *   HO      LENS HEIGHT                                        *
C     *   RT      LENS RADIUS                                        *
C     *   IW      WRITE FLAG (WRITE IF IW = 1)                       *
C     *                                                              *
C     * OUTPUT ARGUMENTS.................(NONE).............         *
C     *                                                              *
C     * JIM WEAVER                                                   *
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY             *
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY                *
C     * ADA, OKLAHOMA  74820                                         *
C     *                                                              *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77               *
C     * REQUIRED ROUTINES: DUPUIT, BINARY                            *
C     * CALLED BY ZMASS                                              *
C     *                                                              *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /RLNS/ HHS(100),NR,RS(100)
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      NUM(47) = NUM(47) + 1
C
C     ***PROFILE THROUGH THE OILENS
C     ***TIME  = SIMULATION TIME
C     ***HO    = HEIGHT OF OIL LENS ABOVE CAPILLARY FRINGE
C     ***RT    = RADIUS OF OIL LENS
C
      IF (IWR.EQ.1.AND.IW.EQ.1) THEN
        WRITE (9,9200) TIME,RT,WTABLE
        WRITE (9,9208)
        WRITE (9,9210)
        WRITE (9,9220) 0.0,WTABLE-HO-FRING3,WTABLE+ALPHA*HO,
     1  WTABLE-HHS(1)-FRING3,WTABLE+ALPHA*HHS(1)
        WRITE (9,9220) RADI,WTABLE-HO-FRING3,WTABLE+ALPHA*HO,
     1  WTABLE-HHS(2)-FRING3,WTABLE+ALPHA*HHS(2)
        IF (NR.GT.1) THEN
C         ***WRITE THE PLOT FILE IF LENS HAS FORMED
          WRITE (17,9301) TIME
          WRITE (17,9302) 0.0,WTABLE-HO-FRING3,WTABLE+ALPHA*HO,
     1    WTABLE-HHS(1)-FRING3,WTABLE+ALPHA*HHS(1)
          WRITE (17,9302) RADI,WTABLE-HO-FRING3,WTABLE+ALPHA*HO,
     1    WTABLE-HHS(2)-FRING3,WTABLE+ALPHA*HHS(2)
        END IF
      END IF
C
      DR = (RT-RADI)*0.05
      RBEG = RADI+DR
      REND = RT+0.5*DR
      IF (NR.GT.1) THEN
        DO 10 R=RBEG,REND,DR
C         ***CURRENT SHAPE OF LENS
          CALL DUPUIT (RADI,RT,R,HO,HR)
C         ***EXTENT OF MAXIMUM EXTENT OF LENS
          CALL BINARY (R,RS(2),NR-1,1,IN)
          IN = IN + 1
          SL = (HHS(IN+1)-HHS(IN))/(RS(IN+1)-RS(IN))
          HM = HHS(IN) + SL*(R-RS(IN))
          E1 = WTABLE-HR-FRING3
          E2 = WTABLE+ALPHA*HR
          E3 = WTABLE-HM-FRING3
          E4 = WTABLE+ALPHA*HM
          IF (IWR.EQ.1.AND.IW.EQ.1) THEN
            WRITE (9,9220) R,E1,E2,E3,E4
            WRITE (17,9302) R,E1,E2,E3,E4
          END IF
 10     CONTINUE
      END IF
C
C     ***WRITE STOP STATEMENT IN THE PLOT FILE
      IF (NR.GT.1.AND.IWR.EQ.1.AND.IW.EQ.1) WRITE (17,9303)
C
      RETURN
 9200 FORMAT (1H1/5X,50(1H*)/
     1        5X,'* RADIAL PROFILE THROUGH OIL LENS'/
     2        5X,50(1H*)/
     3        5X,'TIME                = ',F10.4/
     4        5X,'LENS RADIUS         = ',F10.4/
     5        5X,'DEPTH TO WATER TABLE = ',F10.4/)
 9208 FORMAT (25X,'CURRENT OIL LENS',10X,'MAXIMUM EXTENT OF OIL LENS')
 9210 FORMAT (5X,'  RADIUS  ',2(5X,'  DEPTH OF ',5X,'  DEPTH OF ')/
     1        15X,2(5X,'TOP OF LENS',5X,'LENS BOTTOM')/
     1        5X,10(1H=),4(5X,11(1H=))/)
 9220 FORMAT (5X,F10.4,5X,4(F11.4,5X))
 9301 FORMAT (F10.4)
 9302 FORMAT (F10.4,1X,4(F10.4,1X))
 9303 FORMAT ('STOP')
      END
```

```
      SUBROUTINE INQUAD (X,NX,AMIN,AMAX,VA,RE)
C     ************************************************************
C     *                                                          *
C     *   INQUAD                                                 *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE PERFORMS INVERSE QUADRATIC  *
C     *   INTERPOLATION ON A SET OF PREDETERMINED NUMBERS TO     *
C     *   INVERT A NONLINEAR FUNCTION.  INQSET IS CALLED FIRST   *
C     *   TO SET UP THE ARRAY OF FUNCTION VALUES.  SUBSEQUENT    *
C     *   CALLS TO INQUAD GIVE THE INVERSE FOR SPECIFIED VALUES  *
C     *   OF THE DEPENDENT VARIABLE                              *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................   *
C     *   X      = ARRAY OF FUNCTION VALUES FROM INQSET          *
C     *   NX     = NO. OF X'S                                    *
C     *   AMIN   = MINIMUM INDEPENDENT VARIABLE VALUE            *
C     *   AMAX   = MAXIMUM INDEPENDENT VARIABLE VALUE            *
C     *   VA     = VALUE OF DEPENDENT VARIABLE                   *
C     *                                                          *
C     *   OUTPUT ARGUMENTS.....................................  *
C     *   RE     = VALUE OF INDEPENDENT VARIABLE:  VA = F(RE)    *
C     *                                                          *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   REFERENCES: ABRAMOWITZ AND STEGUN, PG. 882.            *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...INQSET                             *
C     *                                                          *
C     *                                                          *
C     ************************************************************
      COMMON /COUN/ NUM(100)
      REAL X(*)
C     DATA I /1/
C
      NUM(19) = NUM(19) + 1
      IF (NUM(19).EQ.1) THEN
        I = 1
      END IF
C
      DX = (AMAX - AMIN)/FLOAT(  NX - 1  )
      CALL BINARY (VA,X,NX,1,I)
      IF (I.EQ.1.OR.I.EQ.NX-1) THEN
C       ***LINEAR INTERPOLATION FOR END INTERVALS
        P = (VA - X(I))/(X(I+1) - X(I))
      ELSE IF (I.LT.NX) THEN
        A = X(I+1) - 2.*X(I) + X(I-1)
        B = X(I+1) - X(I-1)
        C = 2.*(X(I) - VA)
        IF (A.GT.0.) THEN
C         ***QUADRATIC X() DATA
          Q = B*B - 4.*A*C
          IF (Q.LT.0.) STOP 240
          Q = SQRT(  Q  )
          P = (Q - B)*0.5/A
        ELSE
C         ***LINEAR X() DATA
C         ***SOME PORTIONS OF QUADRATIC DATA MAY BE NEARLY LINEAR
          P = C/B
        END IF
      ELSE
        STOP 241
      END IF
      RE = AMIN + (FLOAT(I-1) + P)*DX
      RETURN
C
```

```
      ENTRY INQSET (FNC,AMIN,AMAX,PARA,NX,X)
C     ************************************************************
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE SETS UP ARRAYS FOR INQUAD   *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................   *
C     *   FNC    = SUBROUTINE NAME OF A NONLINEAR FUNCTION       *
C     *   AMIN   = MINIMUM VALUE OF INDEPENDENT VARIABLE         *
C     *   AMAX   = MAXIMUM VALUE OF INDEPENDENT VARIABLE         *
C     *   PARA   = REAL PARAMETER PASSED TO FNC                  *
C     *   NX     = NUMBER OF DEPENDENT VALUES TO PRODUCE         *
C     *                                                          *
C     *   OUTPUT ARGUMENTS....................................   *
C     *   X      = ARRAY OF DEPENDENT VARIABLES                  *
C     *                                                          *
C     ************************************************************
      NUM(28) = NUM(28) + 1
      SS = AMIN
      DX = (AMAX - AMIN)/FLOAT(  NX - 1  )
      DO 1  I=1,NX
        CALL FNC (SS,PARA,X(I))
        SS = SS + DX
 1    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BINARY (U,X,NX,INCX,IN)
C     ************************************************************
C     *                                                          *
C     *   BINARY                                                 *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE PERFORMS A BINARY SEARCH ON *
C     *   THE ARRAY X TO FIND THE INDEX OF THE X VALUE WHICH LIES *
C     *   JUST BELOW U                                           *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................   *
C     *   U      = POINT SEARCHED FOR                            *
C     *   X      = ARRAY OF VALUES                               *
C     *   NX     = NUMBER OF X VALUES                            *
C     *   INCX   = INCREMENT (MUST = 1)                          *
C     *                                                          *
C     *   OUTPUT ARGUMENTS.....................................  *
C     *   IN     = INDEX OF X VALUE JUST BELOW U                 *
C     *                                                          *
C     *   INTERNAL VARIABLES                                     *
C     *   ILOW   = INDEX OF LOW END OF THE RANGE                 *
C     *   IHIGH  = INDEX OF HIGH END OF THE RANGE                *
C     *   IMID   = INDEX OF MIDPOINT OF THE RANGE                *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   REFERENCE:  FORSYTHE, MALCOLM AND MOLER, 1977, PG. 76  *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...NONE                               *
C     *                                                          *
C     ************************************************************
      COMMON /COUN/ NUM(100)
      REAL X(*)
C
      NUM(20) = NUM(20) + 1
C
      IF (INCX.NE.1) STOP 'BINARY-INDEX'
      IF (X(1).LT.X(NX)) THEN
C     *** X'S ORDERED BY INCREASING VALUE
C     *** BINARY SEARCH
        ILOW = 1
        IHIGH = NX + 1
2       CONTINUE
          IMID = (ILOW+IHIGH)/2
          IF(U.LT.X(IMID)) IHIGH = IMID
          IF(U.GE.X(IMID)) ILOW = IMID
        IF(IHIGH.GT.ILOW+1) GO TO 2
      ELSE
C     *** X'S ORDERED BY DECREASING VALUE
C     *** BINARY SEARCH
        ILOW = 1
        IHIGH = NX + 1
5       CONTINUE
          IMID = (ILOW+IHIGH)/2
          IF(U.LT.X(IMID)) ILOW = IMID
          IF(U.GE.X(IMID)) IHIGH = IMID
        IF(IHIGH.GT.ILOW+1) GO TO 5
      ENDIF
9     IN = ILOW
      RETURN
      END
```

```
      SUBROUTINE BISEC (FNC,ZB,ZE,CONV,IS,SV,TM,ZM,VM,IE)
C     ************************************************************
C     *                                                          *
C     *   BISEC                                                  *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE USES THE BISECTION METHOD TO  *
C     *   FIND THE ZERO OF A FUNCTION                            *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................   *
C     *   FNC    = SUBROUTINE NAME CONTAINING FUNCTION           *
C     *   ZB     = LOWER LIMIT OF SEARCH                         *
C     *   ZE     = UPPER LIMIT OF SEARCH                         *
C     *   CONV   = CONVERGENCE TOLERANCE                         *
C     *   IS     = MAXIMUM NUMBER OF ITERATIONS                  *
C     *   SV     = FUNCTION VALUE TO BE MATCHED BY BISEC ITERATION *
C     *   TM     = REAL PARAMETER PASSED TO FNC                  *
C     *                                                          *
C     *   OUTPUT ARGUMENTS....................................   *
C     *   ZM     = BISEC RESULT (ZB.LE.ZM.LE.ZE)                 *
C     *   VM     = FUNCTION VALUE AT ZM (SHOULD MATCH SV)        *
C     *   IE     = ERROR CODE IE=-1: BISEC FAILED                *
C     *                                                          *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   REFERENCE: FORSYTHE, MALCOLM & MOLER, 1977, PG. 157    *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...SEE INPUT ARGUMENTS                *
C     *   THANKS TO SHERYL FRANKLIN                              *
C     ************************************************************
      COMMON /COUN/ NUM(100)
      NUM(21) = NUM(21) + 1
      IE = -1
      XF = 1.
      ZM = 0.
      Z0 = ZB
      ZF = ZE
C
C     ***ENDPOINT VALUES
      CALL FNC (Z0,TM,V1)
      IF (V1.GT.SV) XF = -1.
      CALL FNC (ZF,TM,V2)
      IF ((V1-SV)*(V2-SV).GT.0.) RETURN
      IE = 0
C     ***ITERATE FOR A MAXIMUM NUMBER OF ITERATIONS
      DO 1  I=1,IS
        ZM = Z0 + (ZF - Z0)*0.5
        CALL FNC (ZM,TM,VM)
        XM = XF*(VM - SV)
        XD = 1.
        IF (SV.GT.0.) XD = SV
        IF (ABS(XM/XD).LE.CONV) RETURN
        IF (XM.GE.0.) THEN
          ZF = ZM
        ELSE
          Z0 = ZM
        END IF
    1   CONTINUE
        STOP 'BISEC: FAIL'
        END
```

```
      SUBROUTINE MBISEC (FNC,ZB,ZE,CONV,EPS,IS,SV,TM,ZM,VM,IE)
C     ************************************************************
C     *                                                          *
C     *   BISEC                                                  *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE USES THE BISECTION METHOD TO  *
C     *   FIND THE ZERO OF A FUNCTION                            *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................   *
C     *   FNC    = SUBROUTINE NAME CONTAINING FUNCTION           *
C     *   ZB     = LOWER LIMIT OF SEARCH                         *
C     *   ZE     = UPPER LIMIT OF SEARCH                         *
C     *   CONV   = CONVERGENCE TOLERANCE                         *
C     *   EPS    = MACHINE FLOATING POINT PRECISION (FROM ROUTINE MEPS)
C     *   IS     = MAXIMUM NUMBER OF ITERATIONS                  *
C     *   SV     = FUNCTION VALUE TO BE MATCHED BY BISEC ITERATION *
C     *   TM     = REAL PARAMETER ARRAY PASSED TO FNC
C     *                                                          *
C     *   OUTPUT ARGUMENTS.....................................   *
C     *   ZM     = BISEC RESULT (ZB.LE.ZM.LE.ZE)                 *
C     *   VM     = FUNCTION VALUE AT ZM (SHOULD MATCH SV)        *
C     *   IE     = ERROR CODE IE=-1: BISEC FAILED                *
C     *                                                          *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   REFERENCE: FORSYTHE, MALCOLM & MOLER, 1977, PG. 157    *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...SEE INPUT ARGUMENTS                *
C     *   THANKS TO SHERYL FRANKLIN                              *
C     ************************************************************
c     IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
C
      IE = -1
      XF = 1.
      C = 0.
      A = ZB
      B = ZE
C
      IF (IS.GT.250) STOP 'MAXIMUM BISEC ITERATIONS 250'
C
C     ***ENDPOINT VALUES
      CALL FNC (A,TM,V1)
      FA = V1-SV
      CALL FNC (B,TM,V2)
      FB = V2-SV
      IF (FA*FB.GT.0.) RETURN
      IE = 0
      FC = FB
C
C     ***ITERATE FOR A MAXIMUM NUMBER OF ITERATIONS
      DO 1  I=1,IS
        IF (FB*FC.GT.0.) THEN
          C = A
          FC = FA
          D = B - A
          E = D
        END IF
        IF (ABS(FC).LT.ABS(FB)) THEN
          A = B
          FA = FB
          B = C
          FB = FC
          C = A
          FC = FA
        END IF
        TL1 = 2.*EPS*ABS(B) + 0.5*CONV
        XM = 0.5*(C-B)
        IF (ABS(XM).LE.TL1.OR.FB.EQ.0.) THEN
          ZM = B
          VM = V2
          IE = 0
          RETURN
        END IF
        IF (ABS(E).GE.TL1.AND.ABS(FA).GT.ABS(FB)) THEN
          S = FB/FA
          IF (A.EQ.C) THEN
            P = 2.*XM*S
            Q = 1.-S
          ELSE
            T = FA/FC
            R = FB/FC
            P = S*((C-B)*T*(T-R) - (B-A)*(R-1.))
            Q = (R-1.)*(S-1.)*(T-1.)
          END IF
          IF (P.GT.0.) Q = -Q
          P = ABS(P)
          IF (2.*P.LT.MIN(3.*XM*Q-ABS(TL1*Q),ABS(E+Q))) THEN
            E = D
            D = P/Q
          ELSE
            D = XM
            E = D
          END IF
        ELSE
          D = XM
          E = D
        END IF
        A = B
        FA = FB
        IF (ABS(D).GT.TL1) THEN
          B = B+D
```

```
        ELSE                                                                  END
          B = B+SIGN(TL1,XM)
        END IF
        CALL FNC (B,TM,V2)
        FB = V2-SV
 1    CONTINUE
      STOP 'MBISEC: FAIL'
      END
      SUBROUTINE GLQ (FNC,A,B,SUM)
C     ************************************************************
C     *                                                          *
C     *   GLQ    SEVEN POINT GAUSS LAGRANGE QUADRATURE           *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE EVALUATES INTEGRALS BY SEVEN  *
C     *   POINT GAUSS-LAGRANGE QUADRATURE                        *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................   *
C     *   FNC    = NAME OF ROUTINE CONTAINING INTEGRAND          *
C     *   A      = LOWER LIMIT OF INTEGRATION                    *
C     *   B      = UPPER LIMIT OF INTEGRATION                    *
C     *                                                          *
C     *   OUTPUT ARGUMENTS.....................................   *
C     *   SUM    = VALUE OF INTEGRAL                             *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   REFERENCES: ABRAMOWITZ AND STEGUN, P. 916              *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...SEE INPUT ARGUMENTS                *
C     *   THIS ROUTINE IS CALLED BY:  CAPSUC                     *
C     *   II-17-87                                               *
C     *                                                          *
C     ************************************************************
      REAL CS(7),CY(7)
      COMMON /COUN/ NUM(100)
      DATA CY /-.949108,-.741531,-.405845,0.,.405845,.741531,.949108/
      DATA CS /.129485,.279705,.381830,.417959,.381830,.279705,.129485/
C
      NUM(61) = NUM(61) + 1
C
      C1 = (B - A)*0.5
      C2 = (B + A)*0.5
C
C     ***FUNCTION EVALUATIONS AT GAUSS POINTS
      SUM = 0.
      DO 10 I=1,7
        Y = CY(I)*C1 + C2
        CALL FNC (Y,VAL)
        SUM = SUM + VAL*CS(I)
 10   CONTINUE
      SUM = SUM*C1
C
      RETURN
```

```
      SUBROUTINE DATA (IREO,IEND)
C     ************************************************************
C     *
C     *  DATA PROMTS THE USER FOR THE DESIRED INPUT DATA FILE
C     *  THE PRE-KOPT PREPROCESSOR CAN BE INVOKED FROM SUBROUTINE DATA
C     *  TO CREATE THE INPUT DATA SET
C     *  VII-31-1989
C     *
C     *
C     *
C     * INPUT ARGUMENTS.........................................
C     *   IREO
C     *   IEND
C     * OUTPUT ARGUMENTS.........(NONE)..........................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  INPUT,NSOPEN,SOPEN
C     * CALLED BY  HSSM
C     *
C     ************************************************************
      CHARACTER*40 IFILE
      COMMON /COUN/ NUM(100)
      NUM(52) = NUM(52) + 1
      DATA IFILE /' '/
C
C
C
C     ISTD = 0
 5    CONTINUE
C
C
C     ***CURRENT SETTING OPENS INPUT FILE FROM DOS PROMPT ON FIRST RUN
C     ***STANDARD FORTRAN ON SUBSEQUENT RUNS
      IF (ISTD.EQ.1) THEN
C       ***STANDARD FORTRAN 77 ROUTINE TO OPEN THE INPUT FILE
INTERACTIVELY
       IF (IFILE.EQ.' ') CALL SOPEN (IFILE)
      ELSE IF (ISTD.EQ.0) THEN
C       ***NON-STANDARD ROUTINE TO OPEN THE INPUT FILE FROM THE DOS PROMPT
C       ***I.E., (HSSM INPUT_FILE.DAT)
       CALL NSOPEN (IFILE)
       ISTD = 1
      END IF
C
C
C     ***READ HSSM INPUT DATA FROM FILE (IFILE) SELECTED ABOVE
      CALL INPUT(IFILE,IREO,IEND,IPRE,IINP)
C
C     ***IF PRE-KOPT WAS RUN DURING EXECUTION OF SUBROUTINE INPUT
C     ***THE INPUT DATA FILE MUST BE RE-OPENED BEFORE RUNNING HSSM
      IF (IPRE.EQ.1) GO TO 5
C     ***RE-OPEN INPUT FILE IF NAME HAS BEEN CHANGED
      IF (IINP.EQ.1) GO TO 5
C
      RETURN
      END
```

```
      SUBROUTINE INPUT (IFILE,IREO,IEND,IPRE,IINP)
C     ****************************************************************
C     *
C     *  KOPT REQUIRED INPUT DATA
C     *  DATA FILES MAY BE PREPARED OR EDITED BY USING THE PRE_KOPT
C     *  PREPROCESSOR
C     *
C     *  INPUT ARGUMENTS:
C     *  IFILE  INPUT DATA FILE NAME (CHARACTER*40)
C     *
C     *  OUTPUT ARGUMENTS: NONE
C     *
C     *  IREO   ECHO PRINT FLAG (EQUIVALENT TO IKOPT)
C     *         0 = ECHO PRINT INPUT DATA ONLY
C     *         1 = RUN KOPT
C     *  IEND   END SIMULATION IF IEND = 1
C     *  IPRE   PRE-KOPT RUN FLAG
C     *         0 = PRE-KOPT NOT RUN FROM SUBROUTINE DPF
C     *         1 = PRE-KOPT RUN FROM SUBROUTEIN DPF
C     *             (MUST RE-OPEN INPUT FILE TO EXECUTE HSSM)
C     *  IINP   INPUT FILE NAME CHANGE FLAG
C     *         0 = NO CHANGE
C     *         1 = NEW INPUT DATA FILE HAS BEEN ENTERED
C     *
C     *  REQUIRED ROUTINES: DPF,IOPOST,UNITS,KRSET,KRCHK,TSGP1,MESSAG
C     *  THIS ROUTINE IS CALLED BY:  DATA
C     *
C     *  NOTES:
C     *  1.   ALL VARIABLE NAMES ARE IN ACCORDANCE WITH FORTRAN NAM-
C     *       ING CONVENTIONS--NAMES BEGINNING WITH I THROUGH M ARE
C     *       INTEGERS, ALL OTHERS ARE REALS.
C     *  2.   ALL INPUT IS FREE-FORMAT
C     *  3.   ZEROS SHOULD BE READ IN FIELDS PERTAINING TO UNUSED VALUES
C     *  4.   INPUT DATA UNITS ARE SPECIFIED AS FOLLOWS
C     *       (*)       DIMENSIONLESS OR NOT APPLICABLE
C     *       (M)       METERS
C     *       (D)       DAYS
C     *       (C)       DEGREES C
C     *       (CP)      CENTIPOISE  1.0 CP = 0.01 GR/CM/SEC
C     *       (M/D)     METERS PER DAY
C     *       (M2/D)    METERS SQUARED PER DAY
C     *       (MG/L)    MILLIGRAMS PER LITER
C     *       (L/KG)    LITERS PER KILOGRAM SOIL
C     *       (GR/CC)   GRAMS PER CUBIC CENTIMETER
C     *
C     *  LINE 1.....PRINT OUTPUT FLAG.................................
C     *  IFACE      INTERFACE FLAG (CHARACTER*1)
C     *             'W' HSSM-KO INPUT DATA SET PREPARED BY WINDOWS
INTERFACE
C     *             'D' HSSM-KO INPUT DATA SET PREPARED BY MSDOS INTERFACE
C     *             ' ' NO INTERFACE USED
C     *  IWR        OUTPUT WRITING FACTOR                          (*)
C     *             0  SUPPRESS ALL OUTPUT AND RUNS KOPT WITH MONTE CARLO
C     *             1  PRODUCE OUTPUT
C     *  IKOPT      KOPT MODEL FLAG
C     *             0  READ AND ECHO PRINT INPUT DATA ONLY
C     *             1  RUN KOPT MODEL
C     *             2  USE MULL'S MODEL (VADOSE ZONE OIL PHASE SIM. ONLY)
C     *  ICONC      DISSOLVED CONSTITUENT FLAG
C     *             0  NO DISSOLVED CONSTITUENT
C     *             1  DISSOLVED CONSTITUENT
C     *  ILENS      OILENS MODEL FLAG
C     *             0  DO NOT RUN OILENS
C     *             1  RUN OILENS
C     *  ITSGP      TSGPLUME FILE FLAG
C     *             0  DO NOT CREATE TSGPLUME INPUT FILE
C     *             1  CREATE TSGPLUME FILE
C     *
C     *  LINE 2-6...OUTPUT AND PLOT FILE NAMES.........................
C     *  OFILE(1)   KOPT/OILENS OUTPUT FILE
C     *  OFILE(2)   KOPT/OILENS PLOT FILE 1
C     *  OFILE(3)   KOPT/OILENS PLOT FILE 2
C     *  OFILE(4)   KOPT/OILENS PLOT FILE 3
C     *  OFILE(5)   TSGPLUME INPUT DATA FILE
C     *  OFILE(6)   TSGPLUME OUTPUT FILE
C     *  OFILE(7)   TSGPLUME PLOT FILE
C     *
C     *  LINE 7-9...RUN TITLE...(5A10/5A10/5A10).......................
C     *  NT(15)     RUN TITLE  3 LINES OF 50 CHARACTERS EACH         (*)
C     *
C     *  LINE 10....MATRIX PROPERTIES..................................
C     *  WKS        SATURATED VERITCAL HYDRAULIC CONDUCTIVITY (WATER)
(M/D)
C     *  RKS        RATIO OF HORIZONTAL TO VERTICAL CONDUCTIVITY    (*)
C     *  KRF        RELATIVE PERMEABILITY MODEL SELECTION INDEX     (*)
C     *             1 BURDINE--BROOKS & COREY MODEL
C     *             2 BURDINE--VAN GENUCHTEN MODEL
C     *  XLAMB      PORE SIZE DISTRIBUTION INDEX                    (*)
C     *             FOR KRF = 1, ENTER LAMBDA
C     *             FOR KRF = 2, ENTER N
C     *  ETA        POROSITY                                        (*)
C     *  SWR        RESIDUAL WATER SATURATION                       (*)
C     *
C     *  LINE 11....WATER PROPERTIES...................................
C     *  WMU        DYNAMIC VISCOSITY OF WATER                       (CP)
C     *  WRHO       DENSITY OF WATER                               (GR/CC)
C     *  IRT        RAINFALL INPUT TYPE:  1=FLUX SPECIFIED           (*)
C     *                                   2=SATURATION SPECIFIED     (*)
C     *  QW/SWMAX   CONSTANT WATER FLUX OR SATURATION           (M/D) OR
(*)
C     *  WTABLE     DEPTH TO WATER TABLE                             (M)
C     *
C     *  LINE 12....OIL CHARACTERISTICS................................
C     *  PMU        DYNAMIC VISCOSITY OF OIL                         (CP)
C     *  PRHO       OIL DENSITY                                    (GR/CC)
C     *  SPR        RESIDUAL (TRAPPED) OIL SATURATION                (*)
C     *  IAT        OIL INPUT TYPE  1=FLUX SPECIFIED                 (*)
C     *                             2=VOLUME/AREA SPECIFIED
C     *                             3=CONSTANT PONDING DEPTH
C     *
```

```
C     *  LINE 13....CAPILLARY SUCTION APPROXIMATION.(ADDITIONAL
PARAMETERS)
C     *  HWE         AIR ENTRY HEAD, IF KRF = 1                          (M)
C     *              ALPHA, IF KRF = 2                                 (1/M)
C     *  WSIG        WATER SURFACE TENSION
(DYNE/CM)
C     *  OSIG        OIL SURFACE TENSION
(DYNE/CM)
C     *
C     *  LINE 14....(FOR IAT=1 AND IAT=3)...OIL FLUX.....................
C     *  QP          OIL FLUX FOR IAT = 1 CASES                        (M/D)
C     *  TPB         OIL EVENT BEGINNING TIME                           (D)
C     *  TPE         OIL EVENT ENDING TIME                              (D)
C     *  HS          CONSTANT HEAD FOR IAT=3 CASES                      (M)
C     *
C     *  LINE 15....(FOR IAT = 2)...OIL VOLUME...........................
C     *  PVOL        OIL VOLUME/AREA INCORPORATED INTO THE SOIL          (M)
C     *  DPL         LOWER DEPTH OF INITIALLY POLLUTED ZONE              (M)
C     *
C     *  LINE 16....DISSOLVED CONSTITUENT................................
C     *  COINI       INITIAL CONCENTRATION IN OIL (SEE NOTE 5.)       (MG/L)
C     *
C     *  LINE 17....DISSOLVED CONSTITUENT................................
C     *  PARTITIONING COEFFICIENTS:
C     *  XXKO        OIL/WATER        (C0 = XXKW*CW)                     (*)
C     *  XXKS        SOLID/WATER (CONSTITUENT)                       (L/KG)
C     *  XXKSH       SOLID/WATER (HYDROCARBON)                       (L/KG)
C     *  RHOS        BULK DENSITY OF MATRIX                         (GR/CC)
C     *
C     *  LINE 18....OILENS SUB-MODEL PARAMETERS (1)......................
C     *  RADI        SOURCE RADIUS                                       (M)
C     *  RMF         RADIUS MULTIPLYING FACTOR                           (*)
C     *  FRING       HEIGHT OF CAPILLARY FRINGE                          (M)
C     *  VDISP       VERTICAL DISPERSIVITY OF AQUIFER                 (M2/D)
C     *  GRAD        GROUNDWATER GRADIENT                                (*)
C     *  SPRB        TRAPPED OIL SATURATION BELOW THE WATER TABLE        (*)
C     *
C     *  LINE 19....OILENS SUB-MODEL PARAMETERS (2)......................
C     *  XMSOL       MAX OIL SATURATION IN THE OIL LENS                  (*)
C     *  SOLC        WATER SOLUBILITY OF CONSTITUENT                  (MG/L)
C     *  SOLH        WATER SOLUBILITY OF HYDROCARBON (OIL)            (MG/L)
C     *
C     *  LINE 20....SIMULATION PARAMETERS................................
C     *  TM          SIMULATION ENDING TIME   (SEE KSTOP)                (D)
C     *  DM          MAXIMUM SOLUTION TIME STEP                          (D)
C     *  DTPR        MINIMUM TIME BETWEEN PRINTED TIME STEPS AND         (D)
C     *              MASS BALANCE CHECKS
C     *  KSTOP       ENDING CRITERIA                                     (*)
C     *              1  USER SPECIFIED ENDING TIME (TM)
C     *              2  OIL LENS MOTION STOPS
C     *              3  CONSTITUENT MASS FLUX TO AQUIFER LESS THAN MAXIMUM
C     *              4  CONSTITUENT MASS IN OIL LENS LESS THAN OPERC*
C     *                 MAXIMUM CUMULATIVE INFLUX TO LENS
C     *              (1 IS DEFAULT FOR NO OILENS SIMULATION OR WHEN OIL
C     *              DOES NOT REACH THE WATER TABLE BEFORE TIME = TM)
C     *  OPERC       FACTOR USED WITH KSTOP = 4  (0.0 < OPERC < 1.0)     (*)
C     *
C     *  LINE 21....PROFILES AND HISTORIES...............................
C     *  NTIMES      NUMBER OF PROFILES (UP TO 10)                       (*)
C     *
C     *  LINE 22.........................................................
C     *  PR(NTIMES)  PROFILE TIMES                                       (D)
C     *              OMIT LINE 17 IF NTIMES = 0
C     *
C     *  LINE 23....TSGPLUME INPUT PARAMETERS............................
C     *  DLONG       AQUIFER LONGITUDINAL DISPERSIVITY                   (M)
C     *  DTRAN       AQUIFER TRANSVERSE DISPERSIVITY                     (M)
C     *  PMAX        PERCENT OF MAXIMUM CONSTITUENT RADIUS               (*)
C     *  CMINW       MIMINUM RECEPTOR WELL CONCENTRATION OF INTEREST (MG/L)
C     *  ZLAM        CONSTITUENT HALF LIFE IN AQUIFER                    (D)
C     *  NWELL       NUMBER OF RECEPTOR WELLS (UP TO 6)                  (*)
C     *
C     *  LINE 24....TSGPLUME INPUT PARAMETES 2...........................
C     *  BEGT        BEGINNING TIME                                      (D)
C     *  ENDT        ENDING TIME                                         (D)
C     *  TINC        TIME INCREMENT                                      (D)
C     *  TAQU        AQUIFER THICKNESS                                   (M)
C     *
C     *  LINE 25.........................................................
C     *  XWELL(I)    X-COORDINATE OF RECEPTOR WELL                       (M)
C     *  YWELL(I)    Y-COORDINATE OF RECEPTOR WELL                       (M)
C     *
C     ****************************************************************
      CHARACTER IFACE*1,NT*15
      CHARACTER*40 IFILE,OFILE(7),MFILE(3)
      DIMENSION XWELL(6),YWELL(6)
      COMMON /BVGE/ XM,XMMO,XMR,XMRMO,XN
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /COUN/ NUM(100)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /CPSA/ HOE,HWE,OSIG,PHIF,WSIG,XLAMBO
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /FLAG/ ICONC,IKOPT,ILENS,ITSGP
      COMMON /FILE/ MFILE,OFILE
      COMMON /FIL2/ NOF
      COMMON /HIPR/ IHIST,NTIMES,PR(10),NZS,HI(5)
      COMMON /HYDI/ HDC1,HDC2,HDC3,HDC4
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /LMTS/ RLOW(6,13),HIGH(6,13),MXROW(12)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /NAME/ NT(15)
```

```
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RLTV/ KRF,KRFO,XLAMB,A2A,A3A,A4A,A5A,A6A
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /STOI/ BDFC,BDFH,SOLC,SOLH,STOICC,STOICH
      COMMON /TRAN/ DWV,EVAP,HLC,HLO,PMU,RH,RHOS,TEMP,WMU,WRHO,XXKS,
     *              XXKSH,XXKV
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      EXTERNAL SLOPE
      NUM(14) = NUM(14) + 1
C
C
C     ***READ THE INPUT DATA
      REWIND 4
      IERR = 0
C
C
C     ***HSSM CONTROL FLAGS
      CALL RFLAGS (IFACE,IWR,IKOPT,ICONC,ILENS,ITSGP)
      IREO = IKOPT
      IF (IKOPT.EQ.2) THEN
        ILENS = 0
        ITSGP = 0
        ICONC = 0
      END IF
C
C
C     ***NUMBER OF OUTPUT AND PLOT FILES (NOF)
      NOF = 7
C     ***READ OUTPUT AND PLOT FILE NAMES
      READ (4,8900,IOSTAT=IERR) (OFILE(II),II=1,NOF)
C
C
C     ***DISPLAY AND PROMPT FILE NAMES
      IF (IWR.EQ.1)  CALL DPF (IFILE,OFILE,NOF,IFACE,IEND,IPRE,IINP)
C
C
      IF (IEND.EQ.1) RETURN
      IF (IPRE.EQ.1) RETURN
C
C
C     ***OPEN OUTPUT FILES
      IF (IWR.EQ.1) CALL OPNFLE
C
C
      CALL MESSAG (6)
C
C     ***RUN TITLE
      READ (4,9000,IOSTAT=IERR) (NT(I),I=1,15)
      IF (IWR.EQ.1) WRITE (15,9100) (NT(I),I=1,15)
      IF (IWR.EQ.1) WRITE (15,9098) IFILE,(OFILE(II),II=1,NOF)
      IF (IWR.EQ.1) WRITE (15,9099) IFACE,IWR,IKOPT,ICONC,ILENS,ITSGP
C
C     ***CONSTANTS AND MATRIX PROPERTIES
      READ (4,*,IOSTAT=IERR) WKS,RKS,KRF,XLAMB,ETA,SWR
      KRFO = KRF
      IF (IWR.EQ.1) WRITE (15,9101) WKS,RKS,KRF,ETA,SWR
      IF (KRF.EQ.1) THEN
        IF (IWR.EQ.1) WRITE (15,9120) XLAMB
      ELSE IF (KRF.EQ.2) THEN
        XN = XLAMB
        IF (IWR.EQ.1) WRITE (15,9121) XN
      END IF
C
C     ***WATER EVENT CHARACTERISTICS
      READ (4,*,IOSTAT=IERR) WMU,WRHO,IRT,QSIN,XMKRW,
     *WTABLE
      IF (IRT.EQ.1) THEN
        QW = QSIN
      ELSE IF (IRT.EQ.2) THEN
        SWMAX = QSIN
      END IF
      IF (IWR.EQ.1) WRITE (15,9102) WMU,WRHO,IRT,QSIN,XMKRW,WTABLE
      WFLUC = 0.0
C
C     ***USE ILENS FLAG TO CONTROL OILENS SIMULATION
C     ***NO OILENS SIMULATION IF ILENS = 0 OR WTABLE < 0
      IF (ILENS.EQ.0) WTABLE = -ABS(WTABLE)
C
C     ***POLLUTANT CHARACTERISTICS
      READ (4,*,IOSTAT=IERR) PMU,PRHO,SPR,IAT
      IF (IWR.EQ.1) WRITE (15,9103) PMU,PRHO,SPR,IAT
      HLO = 0.0
      OLAG = 0.0
C
C     ***CAPILLARY SUCTION PARAMETERS (ADDITIONAL)
      READ (4,*,IOSTAT=IERR) HWE,WSIG,OSIG
C     IF (KRF.EQ.2.AND.IAT.EQ.3) THEN
C       WRITE (*,9900)
C       STOP
C     END IF
      IF (KRF.EQ.2) VGALPA=HWE
      IF (IWR.EQ.1) THEN
        IF (KRF.EQ.1) THEN
          WRITE (15,9104) HWE,WSIG,OSIG
        ELSE IF (KRF.EQ.2) THEN
          WRITE (15,9204) VGALPA,WSIG,OSIG
        END IF
      END IF
C
C     ***APPROXIMATE CONVERSION BETWEEN VAN GENUCHTEN AND BROOKS AND COREY
      IVG2BC = 0
      IF (KRF.EQ.2) THEN
        CALL VG2BC (VGALPA,XN,XLAMB,HWE)
        IVG2BC = 1
        KRF = 1
```

```
      END IF
      XLAMBO = XLAMB
C
      QP = 0.
      HS = 0.
      TPB = 0.
      TPE = 0.
      DPL = 0.
      DPA = 0.
      PVOL = 0.
      IF (IAT.EQ.1) THEN
C       ***OIL FLUX EVENT
        READ (4,*,IOSTAT=IERR) QP,TPB,TPE
        IF (IWR.EQ.1) WRITE (15,9105) QP,TPB,TPE
      ELSE IF (IAT.EQ.2) THEN
C       ***POLLUTANT VOLUME EVENT CHARACTERISTICS
        READ (4,*,IOSTAT=IERR) PVOL,DPL
        IF (IWR.EQ.1) WRITE (15,9106) PVOL,DPL
      ELSE IF (IAT.EQ.3) THEN
C       ***CONSTANT OIL PONDING
        READ (4,*,IOSTAT=IERR) TPB,TPE,HS
        IF (IWR.EQ.1) WRITE (15,9205) TPB,TPE,HS
      ELSE IF (IAT.EQ.4) THEN
C       ***CONSTANT OIL PONDING FOLLOWED BY VARIABLE PONDING
        READ (4,*,IOSTAT=IERR) TPB,TPE,HS
        IF (IWR.EQ.1) WRITE (15,9205) TPB,TPE,HS
      END IF
C
C     ***DISSOLVED CONSTITUENT PARAMETERS
      READ (4,*,IOSTAT=IERR) COINI
      READ (4,*,IOSTAT=IERR) XXKO,XXKS,XXKSH,RHOS
      IF (IWR.EQ.1) WRITE (15,9107) COINI
      IF (IWR.EQ.1) WRITE (15,9108) XXKO,XXKS,XXKSH,RHOS
      IF (COINI.LE.0..AND.XXKO.LE.0.) XXKO = 1.0
      HLC = 0.0
      CLAG = 0.0
      XXKV = 0.0
C
C     ***CONTROL DISSOLVED CONSTITUENT SIMULATION WITH ICONC FLAG
C     ***NO DISSOLVED CONSTITUENT IF ICONC = 0 OR COINI = 0.
      IF (ICONC.EQ.0) COINI = 0.0

C     ***REMNANT OF THE VOLATILIZATION MODEL
      DAIR = 0.
      TEMP = 0.
      RH   = 0.
      EVAP = 0.
      DWV  = 0.
C
C     ***OILENS SUBMODEL PARAMETERS
      READ (4,*,IOSTAT=IERR) RADI,RMF,FRING,VDISP,GRAD,SPRB
      IF (IWR.EQ.1) WRITE (15,9110) RADI,RMF,FRING,VDISP,GRAD,SPRB
      COXY = 0.
      READ (4,*,IOSTAT=IERR) XMSOL,SOLC,SOLH
      IF (IWR.EQ.1) WRITE (15,9111) XMSOL,SOLC,SOLH
C     ***ARCHAIC VARIABLES
C     ***BDFC=CONSTITUENT BIODEGRADATION FACTOR 0.0 TO 1.0
C     ***BDFH=HYDROCARBON BIODEGRADATION FACTOR 0.0 TO 1.0
C     ***STOICC=CONSTITUENT STOIC. RATIO
C     ***STOICH=HYDROCARON STOIC. RATIO
      BDFC = 1.0
      BDFH = 0.0
      STOICC = 0.0
      STOICH = 0.0
      IF (BDFC+BDFH.GT.1.0)
     *STOP 'CHECK BIODEGRADATION FACTORS FOR OILENS SIMULATION'
C     ***STORE MAX LENS OIL SATURATION IN SOLENS FOR PASSAGE OUT
      SOLENS = XMSOL
C
C     ***SIMULATION PARAMETERS
      READ (4,*,IOSTAT=IERR) TM,DM,DTPR,KKSTOP,OPERC
      IF (IWR.EQ.1) WRITE (15,9112) TM,DM,DTPR,KKSTOP,OPERC
C
C     ***OUTPUT PROFILE TIMES AND HISTORY DEPTHS
      READ (4,*,IOSTAT=IERR) NTIMES
      NZS = 0
C
C     ***PROFILE TIMES
      IF (NTIMES.GT.0) THEN
        READ (4,*,IOSTAT=IERR) (PR(I),I=1,NTIMES)
        IF (IWR.EQ.1) WRITE (15,9113) NTIMES,(PR(I),I=1,NTIMES)
      END IF
C
C     ***HISTORY DEPTHS
      IF (NZS.GT.0) THEN
        READ (4,*,IOSTAT=IERR) (HI(I),I=1,NZS)
        IF (IWR.EQ.1) WRITE (15,9114) NZS,(HI(I),I=1,NZS)
      END IF
C
C     ***TSGPLUME INPUT DATA FILE PARAMETERS
      READ (4,*,IOSTAT=IERR) DLONG,DTRAN,PMAX,CMINW,ZLAM,NWELL
      IF (IWR.EQ.1) WRITE (15,9115) DLONG,DTRAN,PMAX,CMINW,ZLAM,NWELL
      READ (4,*,IOSTAT=IERR) BEGT,ENDT,TINC,TAQU
      IF (IWR.EQ.1) WRITE (15,9116) BEGT,ENDT,TINC,TAQU
C
C     ***READ NUMBER OF WELLS
      IF (NWELL.GT.0) THEN
        DO 11 I=1,NWELL
          READ (4,*,IOSTAT=IERR) XWELL(I),YWELL(I)
 11     CONTINUE
      END IF
      IF (IWR.EQ.1.AND.NWELL.GT.0) THEN
        WRITE (15,9117)
        DO 12 I=1,NWELL
```

```
          WRITE (15,9118) XWELL(I),YWELL(I)
 12     CONTINUE
      END IF
C
C     ***END OF INPUT DATA
      IF (IWR.EQ.1) THEN
        WRITE (15,9189)
        WRITE (15,9190)
      END IF
C
C     ***CHECK FOR MISSING DATA ITEM
      IF (IERR.NE.0) THEN
        CALL MESSAG (14)
        CALL IOPOST (OFILE,IWR,IREO)
        CALL MESSAG (5)
        STOP
      END IF
C
C
C     ***CHECK LIMITS ON DATA
      CALL EDITIN
C
C
C     ***STOP IF READ AND ECHO PRINT DATA ONLY
      IF (IREO.EQ.0.AND.IWR.EQ.1) THEN
        CALL MESSAG (4)
        CALL IOPOST (OFILE,IWR,IREO)
        CALL MESSAG (5)
        STOP 'KOPT--ECHO PRINT ONLY'
      END IF
C
C     ***MODIFY THE INPUT IF MONTE CARLO SIMULATION IS BEING USED
      IF (IWR.EQ.0) THEN
C       ***READ THE LAST THREE LINES OF THE MAIN INPUT FILE
        READ (4,8900,IOSTAT=IERR) MFILE(1),MFILE(2),MFILE(3)
C       ***READ THE PARAMETERS VALUES FOR THE CURRENT MONTE CARLO RUN
        CALL DMONTE (WKS,HWE,XLAMB,SWR,SPR,PMU,DUR,ETA,
     *  SWMAX,PRHO,XMKRW,OSIG,NTIMES,PR)
        TPE = TPB + DUR
        IRT = 2
      END IF
C
C     ***END OF INPUT DATA
C
C
C     ***WRITE DATA SET FOR THE NTHICK UTILITY
      CALL WTHICK
C
C
C
C
C     ***SET CONSTANTS
C
C     ***SET UP BUILT-IN UNIT CONVERSIONS
      CALL UNITS
C
      OLAG = TPB+OLAG
      CLAG = TPB+CLAG
      KSTOP = 1
C
C     ***SORT THE PROFILE TIMES
      NLAST = NTIMES - 1
      DO 15 J=1,NTIMES
        IH = 0
        DO 14 I=1,NLAST
          IF (PR(I).GT.PR(I+1)) THEN
            IH = 1
            PTEM = PR(I)
            PR(I) = PR(I+1)
            PR(I+1) = PTEM
          END IF
 14     CONTINUE
        IF (IH.EQ.0) GO TO 16
        NLAST = NLAST - 1
 15   CONTINUE
 16   CONTINUE
C
C     ***ELIMINATE ZEROS
      DO 17 I=1,NTIMES
        IF (PR(I).GT.0.0) GO TO 18
17    CONTINUE
18    IF(I.GT.1) THEN
        II = 1
        DO 19 J = I,NTIMES
          PR(II) = PR(J)
          II = II + 1
19      CONTINUE
        NTIMES = NTIMES - I + 1
      ENDIF
C
C     ***GROUNDWATER FLOW VELOCITY
      VEL = WKS*RKS*ABS(GRAD)
C     ***REDUCED SATURATIONS
      SR    = SPR + SWR
      OMSWR = 1. - SWR
      OMSOR = 1. - SPR
C     ***OIL SATURATED CONDUCTIVITY
      PKS   = WKS*PRHO*WMU/WRHO/PMU
C     ***OIL ENTRY HEAD
      HOE = HWE*OSIG*WRHO/WSIG/PRHO
      XXK1  = 1./XXKO
      XXK2  = XXKS*UC3/XXKO
      XXK2  = RHOS*XXKS*UC3/XXKO
      XXK3  = XXKV/XXKO
      IHIST = 0
      DO 20 I=1,NZS
        IF (HI(I).GE.DPA) IHIST = 1
 20   CONTINUE
      IF (IWR.EQ.1.AND.IHIST.EQ.0.AND.NZS.GT.0) WRITE (15,9170)
C
```

```
C     ***INITIALIZE RELATIVE PERMEABILITY FUNCTION
      CALL KRSET
C     ***CHECK RELATIVE PERMEABILITY FUNCTION FOR REAR SHOCK FORMATION
C     ***NO REAR SHOCK IF DERIVATIVE OF KRO IS MONOTONICALLY INCREASING
C     ***REAR SHOCK IF OTHERWISE (IRS = 1)
      CALL KRCHK (IRS)
C
      IF (IAT.NE.2) THEN
        DPL = DPA
        DTP   = TPE - TPB
        PVOL  = DTP*QP
      END IF
      AC = RADI*RADI*PI
      IF (IWR.EQ.1) THEN
        WRITE (15,9180) PKS,AC
        IF (IVG2BC.EQ.1) WRITE (15,9185) XLAMB,HWE
      END IF
C
      IF (XXK3.GT.0.) THEN
C       ***CALCULATION OF ATMOSPHERIC BOUNDARY LAYER
C       ***LINSLEY, KOHLER & PAULHUS, 1975, PG. 35
C       ***SATURATION VAPOR PRESSURE FORMULA PG. 35
        ES = 33.8639*((0.00738*TEMP + 0.8072)**8 - .000019*
     1  ABS(1.8*TEMP+ 48.) + 0.001316)
C       ***WATER VAPOR DENSITY, PG. 34
        RWV = 0.622*UC4*ES/(TEMP+273.)/2870.
C       ***ATMOSPHERIC BOUNDARY LAYER THICKNESS
        ABL = DWV*RWV*(1.-RH)/2./EVAP/WRHO
        IF (IWR.EQ.1) WRITE (15,9184) ABL
      END IF
      ALPHA  = PRHO/(WRHO-PRHO)
      AP1    = 1. + ALPHA
C
C     ***DECAY RATES FROM HALF LIVES
      IF (HLO.NE.0.) THEN
        DLO  = - ALOG(0.5)/HLO
        IF (IWR.EQ.1) WRITE (15,9181) DLO
      END IF
      IF (HLC.NE.0.) THEN
        DLC = - ALOG(0.5)/HLC
        IF (IWR.EQ.1) WRITE (15,9182) DLC
      END IF
      IF (ZLAM.NE.0.) THEN
        ZLAM  = - ALOG(0.5)/ZLAM
        IF (IWR.EQ.1) WRITE (15,9181) ZLAM
      END IF
C
C     ***CREATE FIRST PART OF TSGPLUME INPUT DATA FILE
      IF (ITSGP.EQ.1) CALL TSGP1 (IFILE,OFILE,KKSTOP,IFACE,
     *DLONG,DTRAN,VDISP,TAQU,PMAX,CMINW,BEGT,ENDT,TINC,
     *NWELL,XWELL,YWELL,VEL,ETA,XXKS,RHOS,ZLAM)
      RETURN
 8800 FORMAT (A1,I5,4(1X,I5))
 8900 FORMAT (A40/A40/A40/A40/A40/A40/A40)
 9000 FORMAT (5A10/5A10/5A10)
 9098 FORMAT (15X,'DATA FILES:'/
     *        15X,' HSSM-KO INPUT:     ',A40/
     *        15X,' HSSM-KO OUTPUT:    ',A40/
     *        15X,' HSSM-KO PLOT 1:    ',A40/
     *        15X,' HSSM-KO PLOT 2:    ',A40/
     *        15X,' HSSM-KO PLOT 3:    ',A40/
     *        15X,' HSSM-T INPUT:      ',A40/
     *        15X,' HSSM-T OUTPUT:     ',A40/
     *        15X,' HSSM-T PLOT:       ',A40)
 9099 FORMAT (15X,30HINTERFACE FLAG                = ,A1/
     *        15X,30HWRITING CRITERIA              = ,I10/
     *        15X,30HKOPT RUN FLAG                 = ,I10/
     *        15X,30HDISSOLVED CONSTITUENT FLAG    = ,I10/
     *        15X,30HOILENS RUN FLAG               = ,I10/
     *        15X,30HTSGPLUME RUN FLAG             = ,I10/)
 9100 FORMAT (1H1/10X,50(1H*)/
     *        10X,'HSSM            HYDROCARBON SPILL SIMULATION MODEL'/
     *        10X,50(1H*)/
     *        10X,'KOPT            KINEMATIC OILY POLLUTANT TRANSPORT'/
     *        10X,'OILENS          RADIAL OIL LENS MOTION'/
     *        10X,'TSGPLUME        TRANSIENT SOURCE GAUSSIAN PLUME'/
     *        10X,50(1H*)/
     *        3(10X,5A10/)//
     *        15X,10HINPUT DATA/
     *        15X,10(1H=)/)
 9101 FORMAT (15X,30HCONSTANTS & MATRIX PROPERTIES.,10(1H.)/
     *        15X,30HSAT. VERT. HYD.CONDUCTIVITY = ,G10.4,' (M/D)'/
     *        15X,30HRATIO OF HORZIZONTAL TO       ,/
     *        15X,30HVERTICAL CONDUCTIVITY       = ,G10.4,' (*)'/
     *        15X,30HRELATIVE PERMEABILITY INDEX = ,I10,' (*)'/
     *        15X,30HPOROSITY                    = ,G10.4,' (*)'/
     *        15X,30HRESIDUAL WATER SATURATION   = ,G10.4,' (*)')
 9120 FORMAT (15X,30HBROOKS AND COREY'S LAMBDA   = ,G10.4,' (*)'/)
 9121 FORMAT (15X,30HVAN GENUCHTEN'S N           = ,G10.4,' (*)'/)
 9102 FORMAT (15X,30HWATER EVENT CHARACTERISTICS...,10(1H.)/
     *        15X,30HDYNAMIC VISCOSITY           = ,G10.4,' (CP)'/
     *        15X,30HDENSITY                     = ,G10.4,' (G/CC)'/
     *        15X,30HRAIN TYPE : 1-FLUX 2-SAT.   = ,I10,' (*)'/
     *        15X,30HWATER FLUX OR SATURATION    = ,G10.4,' (M/D OR *)'/
     *        15X,30HMAX KRW DURING INFILTRATION = ,G10.4,' (*)'/
     *        15X,30HDEPTH TO WATER TABLE        = ,G10.4,' (M)'/)
 9103 FORMAT (15X,35HPOLLUTANT EVENT CHARACTERISTICS....,5(1H.)/
     *        15X,30HDYNAMIC VISCOSITY           = ,G10.4,' (CP)'/
     *        15X,30HDENSITY                     = ,G10.4,' (G/CC)'/
     *        15X,30HRESIDUAL NAPL SATURATION    = ,G10.4,' (*)'/
     *        15X,30HOIL LOADING TYPE            = ,I10,' (*)'/)
 9104 FORMAT (15X,30HCAPILLARY SUCTION PARAMETERS..,10(1H.)/
     *        15X,30HAIR ENTRY HEAD              = ,G10.4,' (M)'/
```

```
     *        15X,30HWATER SURFACE TENSION       = ,G10.4,' (DYNE/CM)'/
     *        15X,30HOIL SURFACE TENSION         = ,G10.4,' (DYNE/CM)'/)
 9204 FORMAT (15X,30HCAPILLARY SUCTION PARAMETERS..,10(1H.)/
     *        15X,30HVAN GENUCHTENS ALPHA        = ,G10.4,' (1/M)'/
     *        15X,30HWATER SURFACE TENSION       = ,G10.4,' (DYNE/CM)'/
     *        15X,30HOIL SURFACE TENSION         = ,G10.4,' (DYNE/CM)'/)
 9105 FORMAT (15X,30HFLUX LOADING RATE           = ,G10.4,' (M/D)'/
     *        15X,30HBEGINNING TIME              = ,G10.4,' (D)'/
     *        15X,30HENDING TIME                 = ,G10.4,' (D)'/)
 9205 FORMAT (15X,30HBEGINNING TIME              = ,G10.4,' (D)'/
     *        15X,30HENDING TIME                 = ,G10.4,' (D)'/
     *        15X,30HDEPTH OF PONDING            = ,G10.4,' (M)'/)
 9106 FORMAT (15X,30HOIL VOLUME/AREA             = ,G10.4,' (M)'/
     *        15X,30HDEPTH: BOTTOM OF NAPL LAYER = ,G10.4,' (M)'/)
 9107 FORMAT (15X,32HDISSOLVED CONSTITUENT PARAMETERS,8(1H.)/
     *        15X,30HINITIAL CONC. IN NAPL       = ,G10.4,' (MG/L)')
 9108 FORMAT (15X,30HNAPL/WATER PARTITION COEF.  = ,G10.4,' (*)'/
     *        15X,30HSOIL/WATER PARTITION COEF.  = ,G10.4,' (L/KG)'/
     *        15X,30HSOIL/WATER (HYDROCARBON)    = ,G10.4,' (L/KG)'/
     *        15X,30HBULK DENSITY                = ,G10.4,' (G/CC)'/)
 9109 FORMAT (15X,31HVOLATILIZATION MODEL PARAMETERS,9(1H.)/
     *        15X,30HAIR-GAS  DIFFUSION COEFF.   = ,G10.4/
     *        15X,30HAIR-WATER VAPOR DIFF. COEFF.= ,G10.4/
     *        15X,30HEVAPORATION RATE            = ,G10.4/
     *        15X,30HTEMPERATURE DEG. C          = ,G10.4/
     *        15X,30HRELATIVE HUMIDITY           = ,G10.4/)
 9110 FORMAT (15X,30HOILENS SUBMODEL PARAMETERS....,10(1H.)/
     *        15X,30HRADIUS OF POLLUTANT SOURCE  = ,G10.4,' (M)'/
     *        15X,30HRADIUS MULTIPLYING FACTOR   = ,G10.4,' (*)'/
     *        15X,30HTHICKNESS OF CAP. FRINGE    = ,G10.4,' (M)'/
     *        15X,30HAQUIFER'S VERT DISPERSIVITY = ,G10.4,' (M)'/
     *        15X,30HGROUNDWATER GRADIENT        = ,G10.4,' (*)'/
     *        15X,30HNAPL RESIDUAL IN AQUIFER    = ,G10.4,' (*)')
 9111 FORMAT (15X,30HMAX NAPL SATURATION IN LENS = ,G10.4,' (*)'/
     *        15X,30HWATER SOLUBILITY CONTAMINANT= ,G10.4,' (MG/L)'/
     *        15X,30HWATER SOLUBILITY OF OIL     = ,G10.4,' (MG/L)'/)
 9112 FORMAT (15X,30HSIMULATION PARAMETERS.........,10(1H.)/
     *        15X,30HSIMULATION ENDING TIME      = ,G10.4,' (D)'/
     *        15X,30HMAXIMUM RKF TIME STEP       = ,G10.4,' (D)'/
     *        15X,30HMIN. TIME BETWEEN PRINTING  = ,G10.4,' (D)'/
     *        15X,30HENDING CRITERIA             = ,I10,' (*)'/
     *        15X,30HFACTOR FOR ENDING CRITERIA 4= ,G10.4,' (*)'/)
 9113 FORMAT (15X,30HPROFILES......................,10(1H.)/
     *        15X,30HNUMBER OF PROFILES          = ,I10,' (*)'/
     *        15X,30HAT TIMES:                     ,10X,' (D)'/
     *        4(15X,3(F12.4,1X)/))
 9114 FORMAT (15X,30HNUMBER OF HISTORIES         = ,I10/
     *        15X,30HAT DEPTHS:                    ,/
     *        15X,3(F10.2,1X)/)
 9115 FORMAT (/15X,30HTSGPLUME MODEL PARAMETERS.....,10(1H.)/
     *        15X,30HLONGITUDINAL DISPERSIVITY     ,G10.4,' (M)'/
     *        15X,30HTRANSVERSE DISPERSIVITY       ,G10.4,' (M)'/
     *        15X,30HPERCENT MAX. RADIUS           ,G10.4,' (M)'/
     *        15X,30HMINIMUM OUTPUT CONC.          ,G10.4,' (MG/L)'/
     *        15X,30HCONSTITUENT HALF LIFE         ,G10.4,' (D)'/
     *        15X,30HNUMBER OF RECEPTOR LOCATIONS  ,I10,' (*)')
 9116 FORMAT (15X,30HBEGINNING TIME (D)           ,G10.4,' (D)'/
     *        15X,30HENDING TIME (D)             ,G10.4,' (D)'/
     *        15X,30HTIME INCREMENT (D)          ,G10.4,' (D)'/
     *        15X,30HAQUIFER THICKNESS (M)       ,G10.4,' (M)'/)
 9117 FORMAT (15X,'RECEPTOR LOCATIONS      ' /
     *        20X,9X,'X',10X,'Y')
 9118 FORMAT (20X,G10.4,1X,G10.4)
 9160 FORMAT (15X,30HRELATIVE VAPOR DENSITY      = ,G10.4)
 9170 FORMAT (15X,'***HISTORY DEPTHS ABOVE APPLICATION DEPTH***')
 9180 FORMAT (15X,21HCALCULATED PARAMETERS,19(1H.)/
     *        15X,30HSAT VERT NAPL  CONDUCTIVITY = ,G10.4,' (M/D)'/
     *        15X,30HAREA OF THE SOURCE          = ,G10.4,' (M^2)')
 9181 FORMAT (15X,30HOIL DECAY RATE              = ,G10.4,' (1/D)')
 9182 FORMAT (15X,30HCONSTITUENT DECAY RATE      = ,G10.4,' (1/D)')
 9183 FORMAT (15X,30HCONST. AQUIFER DECAY RATE   = ,G10.4,' (1/D)')
 9184 FORMAT (15X,30HATM. BNDRY. LAYER THICKNESS = ,G10.4,' (M)')
 9185 FORMAT (15X,30HAPPROX. BROOKS AND COREY      ,/
     *        15X,30HLAMBDA                      = ,G10.4,' (*)'/
     *        15X,30HAIR ENTRY HEAD              = ,G10.4,' (M)')
 9189 FORMAT (//15X,'LEGEND'/15X,'======'/
     *       15X,'(*)       DIMENSIONLESS OR NOT APPLICABLE'/
     *       15X,'(M)       METERS'/
     *       15X,'(D)       DAYS'/
     *       15X,'(CP)      CENTIPOISE  1.0 CP = 0.01 GR/CM/SEC'/
     *       15X,'(M/D)     METERS PER DAY'/
     *       15X,'(DYNE/CM) DYNE PER CENTIMETER'/
     *       15X,'(MG/L)    MILLIGRAMS PER LITER'/
     *       15X,'(L/KG)    LITERS PER KILOGRAM SOIL'/
     *       15X,'(G/CC)    GRAMS PER CUBIC CENTIMETER')
 9190 FORMAT (///15X,'***END OF INPUT DATA***'/)
 9200 FORMAT (/10X,'*** DATA INPUT          ***')
 9900 FORMAT (/5X,'VAN GENUCHTEN MODEL IS INCOMPATIBLE WITH THE'/
     *         5X,'CAPILLARY SUCTION APPROXIMATION IN THIS'/
     *         5X,'VERSION OF KOPT/OILENS'/)
      END
```

```
      subroutine rflags (iface,iwr,ikopt,iconc,ilens,itsgp)
c     ************************************************************
c     *
c     *  rflays reads the input flags as a large character string
c     *         it then extracts the integer flags
c     *         (this is to avoid formatted input statements for
c     *         the mixed character and integer line of data)
c     *
c     *
c     * input arguments.............(none).........................
c     * output arguments.....................................
c     *   iface     character*1  interface flag indicates which interface
c     *              was used to develop the input data set
c     *   iwr       write flag (write output files if iwr=1)
c     *   ikopt     kopt run flag
c     *   iconc     dissolved constituent flag
c     *   ilens     oilens run flag
c     *   itsgp     create tsgplume input data flag
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * single precision ansi std x3.9-1978 fortran 77
c     * required routines  none
c     * called by  input
c     *
c     *
c     *
c     ************************************************************
      character buffer*80,iface*1
      dimension ivars(5)
c
      read (4,'(a)') buffer
c
c     ***interface flag (should be W,D or space)
      iface = buffer(1:1)
c
      iv = 0
c     ***HSSM execution flags
c     ***these are extracted from the string,rather than
c     ***being read directly due to problems in reading
c     ***free format input from lines with both character and
c     ***integer data
      do 10 i=2,80
c
        if (buffer(i:i).eq.'0') then
          iv = iv + 1
          ivars(iv) = 0
        else if (buffer(i:i).eq.'1') then
          iv = iv + 1
          ivars(iv) = 1
        else if (buffer(i:i).eq.'2') then
          iv = iv + 1
          ivars(iv) = 2
        end if
c
        if (iv.eq.5) go to 11
c
 10   continue
      stop 'all input flags not found'
 11   continue
c
      iwr   = ivars(1)
      ikopt = ivars(2)
      iconc = ivars(3)
      ilens = ivars(4)
      itsgp = ivars(5)
c
      return
      end
```

```
      SUBROUTINE DPF (IFILE,OFILE,NOF,IFACE,IEND,IPRE,IINP)
C     ***************************************************************
C     *
C     *  DPF    DISPLAY THE FILE NAMES AND PROMPT FOR NEW FILES
C     *
C     *
C     *
C     * INPUT ARGUMENTS.........................................
C     *   IFILE   INPUT FILE NAME
C     *   OFILE   OUTPUT FILE NAMES
C     *   NOF     NUMBER OF OUTPUT FILE NAMES
C     *   IFACE   CHARACTER*1 INTERFACE FLAG
C     *   IEND    EXIT FLAG (DO NOT RUN HSSM IF IEND = 1)
C     *   IPRE    PREKOPT RUN FLAG (=1 IF PREKOPT HAS RUN)
C     *   IINP    FILE NAME HAS BEEN CHANGED IF IINP = 1
C     * OUTPUT ARGUMENTS........................................
C     *   IFILE   INPUT FILE NAME
C     *   OFILE   OUTPUT FILE NAMES
C     *   IEND    EXIT FLAG (DO NOT RUN HSSM IF IEND = 1)
C     *   IPRE    PREKOPT RUN FLAG (=1 IF PREKOPT HAS RUN)
C     *   IINP    FILE NAME HAS BEEN CHANGED IF IINP = 1
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  DIR,MESSAG,PREKOP
C     * CALLED BY   INPUT
C     *
C     ***************************************************************
C
C
      CHARACTER*40 ANS*1,IFACE*1,IFILE,OFILE(*)
      CHARACTER*60 OTEXT(8)
      LOGICAL EXIST
C
      IEND = 0
      IPRE = 0
      IINP = 0
C
C     ***SET TEXT STRINGS
      OTEXT(1) = 'ENTER THE NAME OF THE KOPT/OILENS OUTPUT FILE'
      OTEXT(2) = 'ENTER THE NAME OF THE KOPT/OILENS PLOT FILE 1'
      OTEXT(3) = 'ENTER THE NAME OF THE KOPT/OILENS PLOT FILE 2'
      OTEXT(4) = 'ENTER THE NAME OF THE KOPT/OILENS PLOT FILE 3'
      OTEXT(5) = 'ENTER THE NAME OF THE TSGPLUME INPUT DATA FILE'
      OTEXT(6) = 'ENTER THE NAME OF THE TSGPLUME OUTPUT FILE'
      OTEXT(7) = 'ENTER THE NAME OF THE TSGPLUME PLOT FILE'
C
 5    CONTINUE
C
      DO 40 I=1,NOF
C       ***IF ANY FILE NAME IS BLANK PROMPT FOR THE NAME
        IF (OFILE(I).EQ.' ') THEN
 10       CONTINUE
          WRITE (*,*) ' '
          WRITE (*,*) OTEXT(I)
          WRITE (*,*) '----+----*-40-CHARACTER-LIMIT*----+----*'
          READ (*,'(A)') OFILE(I)
C         ***CHECK TO SEE IF FILE ALREADY EXISTS
          INQUIRE (FILE=OFILE(I),EXIST=EXIST)
          IF (EXIST.EQV..TRUE.) THEN
C           ***ASK IF EXISTING FILE SHOULD BE OVERWRITTEN
            WRITE (*,*)
     *       'OUTPUT FILE ALREADY EXISTS--OVERWRITE? (Y OR N)'
            READ (*,'(A)') ANS
            IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
C             ***OVERWRITE EXISTING FILE (DELETE THE OLD ONE)
              OPEN (30,FILE=OFILE(I),STATUS='OLD')
              CLOSE (30,STATUS='DELETE')
            ELSE
C             ***GO BACK AND ASK FOR A NEW NAME
              GO TO 10
            END IF
          END IF
        END IF
C
 40   CONTINUE
C
 60   CONTINUE
C     ***PROMPT FOR NEW FILE NAMES
      WRITE (*,9000) IFILE,(OFILE(II),II=1,NOF)
      WRITE (*,*) ' '
C
      WRITE (*,*) 'TO RUN HSSM             ENTER <RETURN>'
      IX = 0
      IF (IFACE.EQ.'W'.OR.IFACE.EQ.'w') IX = 1
      IF (IX.EQ.0) THEN
        WRITE (*,*) 'TO CHANGE INPUT FILE    ENTER F'
        WRITE (*,*) 'TO VIEW DIRECTORY       ENTER D'
      END IF
      WRITE (*,*) 'TO EXIT                 ENTER 1'
      WRITE (*,*) ' '
      READ (*,'(A)') ANS
C
C
C
      IF (IX.EQ.0.AND.(ANS.EQ.'f'.OR.ANS.EQ.'F')) THEN
        IINP = 1
        IFILE = ' '
      ELSE IF (ANS.EQ.'1') THEN
C       ***EXIT WITHOUT RUNNING KOPT
        WRITE (*,*) ' '
        IEND = 1
        RETURN
      ELSE IF (IX.EQ.0.AND.(ANS.EQ.'D'.OR.ANS.EQ.'d')) THEN
        CALL DIR
        GO TO 5
```

```
      END IF
C
      RETURN
 9000 FORMAT (/1X,'OUTPUT AND PLOT FILE NAMES:'//
     *         1X,'HSSM INPUT DATA FILE  ',A40/
     *         1X,'HSSM-KO OUTPUT        ',A40/
     *         1X,'HSSM-KO PLOT 1        ',A40/
     *         1X,'HSSM-KO PLOT 2        ',A40/
     *         1X,'HSSM-KO PLOT 3        ',A40/
     *         1X,'HSSM-T INPUT          ',A40/
     *         1X,'HSSM-T OUTPUT         ',A40/
     *         1X,'HSSM-T PLOT           ',A40/)
      END
```

```
      SUBROUTINE DATA1
C     ************************************************************
C     *  DATA1 INITIALIZES VARIABLES IN KOPT AND OILENS
C     *
C     *  INPUT ARGUMENTS:  NONE
C     *  OUTPUT ARGUMENTS: NONE
C     *
C     *  JIM WEAVER                                                    C
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY              C
C     *                                                                
C     *  ANSI STD. X3.9-1978 FORTRAN 77                                C
C     *  REQUIRED ROUTINES:  NONE
C     *  THIS ROUTINE IS CALLED BY:  HSSM
C     ************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /HIPR/ IHIST,NTIMES,PR(10),NZS,HI(5)
      COMMON /TRIA/ DTOLD,DTOLD1,I1,I2,I5,ICC,ICHAR,TEND
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /OUTP/ ERCMX,EROMX,NPT,NREG
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
C
C
C     ***INITIALIZE GENERAL HSSM PARAMETERS AND KOPT PARAMTERS
      DO 10 I=1,100
        NUM(I) = 0
10    CONTINUE
C
C
      NTIMES = 0
      DO 11 I=1,10
        PR(I) = 0.
 11   CONTINUE
C
C
      DZPMDT = 0.
      NPT = 2
      ISE = 0
      DPA=0.
      DPL=0.
      HS=0.
      THPZ=0.
      TPB=0.
      TPE=0.
      QP=0.
      IP=0
      DO 15 I=1,2
        CVF(I)=0.
        XJV(I)=0.
15    CONTINUE
      ERCMX=0.
      EROMX=0.
      DTOLD1=0.
      I1=2
      I2=1
      I5=1
      ICC=0
      ICHAR=0
C
C
      pi=3.141592654
C
      RETURN
      END
```

```
      BLOCK DATA THREE
C     ****************************************************************
C     *
C     * BLOCK DATA THREE   ANSI STANDARD INITIALIZATION OF VARIABLES IN
C     *                    COMMON STATEMENTS
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............(NONE).........................
C     * OUTPUT ARGUMENTS............(NONE).........................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     *
C     ****************************************************************
      CHARACTER NT*15
      COMMON /HIPR/ IHIST,NTIMES,PR(10),NZS,HI(5)
      COMMON /LIST/ VALUES(6,13),IWR
      COMMON /LMTS/ RLOW(6,13),HIGH(6,13),MXROW(12)
      COMMON /NAME/ NT(15)
      COMMON /TPMO/ XWELL(8),YWELL(8)
C
C    ***** DATA ITEM LIMITS GROUPED BY COLUMNS OF 6 *****
C    ***COLUMNS CORRESPOND TO LINES 3 TO 13 OF THE DISPLAY MENU
C    ***I.E., COLUMN 1 CORRESPONDS TO LINE 4 MATRIX PROPERTIES
C    ***I.E., COLUMN 2 CORRESPONDS TO LINE 5 WATER PROPERTIES
C    ***I.E., COLUMN 3 CORRESPONDS TO LINE 6 OIL PROPERTIES
C    ***I.E., COLUMN 4 CORRESPONDS TO LINE 7 CAPILLARY PROPERTIES
C    ***I.E., COLUMN 5 CORRESPONDS TO LINE 8 OIL FLUX
C    ***I.E., COLUMN 6 CORRESPONDS TO LINE 9 DISSOLVED CONSTIT.
C    ***I.E., COLUMN 7 CORRESPONDS TO LINE 10 PARTITION COEFF.
C    ***I.E., COLUMN 8 CORRESPONDS TO LINE 11 OILENS-1
C    ***I.E., COLUMN 9 CORRESPONDS TO LINE 12 OILENS-2
C    ***I.E., COLUMN 10 CORRESPONDS TO LINE 13 SIMULATION
C    ***I.E., COLUMN 11 CORRESPONDS TO LINE 14 PROFILES AND HIST.
C    ***I.E., COLUMN 12 CORRESPONDS TO LINE 17 TSGPLUME DATA
      DATA RLOW /
     *           2*0.,1.,3*0.,
     *           2*0.,1.,0.,0.4,-99999.,
     *           3*0.,1.,2*0.,
     *           6*0.,6*0.,6*0.,6*0.,
     *           0.,1.0001,4*0.,6*0.,
     *           0.,0.1,0.,1.,2*0.,
     *           6*0.,6*0.,6*0./
      DATA HIGH /
     *           864.,99999.,1.,3*99999.,
     *           2*99999.,2.,99999.,0.6,99999.,
     *           3*99999.,4.,2*99999.,
     *           6*99999.,6*99999.,6*99999.,6*99999.,
     *           99999.,1.1,3*99999.,1.,
     *           6*99999.,
     *           99999.,365.,99999.,4.,2*99999.,
     *           10.,2.,4*99999.,
     *           2*99999.,1000.,2*99999.,8.,
     *           6*99999./
      DATA MXROW/6,6,4,3,6,1,4,6,3,5,6,4/
      DATA NTIMES,PR,NZS,HI /0,10*0.0,0,5*0.0/
      DATA NT /15*' '/
      DATA XWELL,YWELL /16*0./
C
      END
```

```
      SUBROUTINE MEPS (EPS)
C     ****************************************************************
C     *
C     * MEPS   MACHINE FLOATING POINT PRECISION
C     *        SEE  FORSYTHE, MALCOLM AND MOLER, PG.14
C     *
C     * INPUT ARGUEMENTS......(NONE)..............................
C     * OUTPUT ARGUMENT..........................................
C     *    EPS   ESTIMATE OF MACHINE PRECISION
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  INITIA
C     *
C     ****************************************************************
C
C
      EPS = 1.
 10   CONTINUE
        EPS = 0.5*EPS
        EP1 = EPS + 1.
      IF (EP1.GT.1) GO TO 10
C
      RETURN
      END
```

```
      SUBROUTINE INITIA (NEQ,ZZ,ND)
C     ***************************************************************
C     *
C     *  INITIA INITIALIZES VARIOUS PARAMETERS OF THE MODEL
C     *
C     *
C     *
C     * INPUT ARGUMENTS.....................................      C
C     *   ND     ROW DIMENSION OF ZZ ARRAY                         C
C     * OUTPUT ARGUMENTS.....................................      C
C     *   NEQ    INTEGER ARRAY OF NUMBER OF EQUATIONS PER GROUP
C     *   ZZ     INITIAL VALUES OF DEPENDENT VARIABLES
C     *
C     * JIM WEAVER                                                 1
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES BISEC,WFFS,PKE,CAPSUC,SLOPE,INTLNS,INTCHR,
C     *                   MBISEC,HEADS,MEPS,WKE (ARGUMENT)
C     * CALLED BY  INPUT
C     *                                                            2
C     ***************************************************************  C
      DIMENSION NEQ(*),ZZ(ND,*)                                        C
      DOUBLE PRECISION GGA1,GGA2,GGA3,GGA4                             C     ***WATER COMPUTATIONS
      CHARACTER NT*15                                                  C     ***CONSTANT WATER SATURATION ABOVE THE WATERTABLE
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)          C     ***ASSUMED REPRESENTATIVE OF CLIMATIC CONDITIONS
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,       call meps (epps)
     *              UC1,UC2,UC3,UC4,UC5                                     CALL MBISEC (WKE,XMKRW,1.,1.E-5,epps,250,XMKRW,1.,SLIM,XXX,IE)
      COMMON /COUN/ NUM(100)                                           c
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO                   SAR = 1.-SLIM
      COMMON /CPSA/ HOE,HWE,OSIG,PHIF,WSIG,XLAMBO                            IF (IWR.EQ.1) WRITE (15,9400) SAR
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)                       IF (IRT.EQ.1) THEN
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)          C       ***WATER FLUX SPECIFIED
      COMMON /HIPR/ IHIST,NTIMES,PR(10),NZS,HI(5)                      C       ***DETERMINE THE WATER SATURATION CORRESPONDING TO THE GIVEN FLUX
      COMMON /HYDI/ HDC1,HDC2,HDC3,HDC4                                        CALL WFFS
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA       ELSE
      COMMON /GAE2/ GGA1,GGA2,GGA3,GGA4                                C       ***WATER SATURATION SPECIFIED
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE                              IF (SWMAX.GT.1.-SAR) SWMAX = 1.-SAR
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)     CALL WKE (SWMAX,WKS,QW)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL      END IF
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR                               CALL PKE (1.-SWMAX-SAR,SWMAX,OKEM)
      COMMON /MULL/ VPA                                                      IF (IWR.EQ.1) WRITE (15,9500) SWMAX,QW,OKEM
      COMMON /NAME/ NT(15)                                                   IF (QW.GE.XMKRW*WKS) STOP 'WATER FILLS PORE SPACE'
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)       C
      COMMON /OUTP/ ERCMX,EROMX,NPT,NREG                               C     ***HYDROCARBON DISSOLUTION CONSTANTS
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,     HDC1 = QW/(SWMAX*ETA + RHOS*XXKSH)
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22                       HDC2 = QW*SOLH*UC5
      COMMON /RLTV/ KRF,KRFO,XLAMB,EPS,EM1,EM2,CC1,REM1                      HDC3 = UC5*SOLH*(ETA*SWMAX + RHOS*XXKSH)
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM            HDC4 = SOLH*(ETA*SWMAX + RHOS*XXKSH)*CMKG/CVSV
      COMMON /STOI/ BDFC,BDFH,SOLC,SOLH,STOICC,STOICH                  C
      COMMON /TRAN/ DWV,EVAP,HLC,HLO,PMU,RH,RHOS,TEMP,WMU,WRHO,XXKS,
     *              XXKSH,XXKV
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      EXTERNAL SLOPE,PKE,WKE
      NUM(15) = NUM(15) + 1



      DO 1 I=1,101
        ZZ(I,2) = DPA
        ZZ(I,3) = DPA
    1 CONTINUE
      ZZ(1,1) = DPL
      ZZ(1,6) = 0.0
      ZZ(2,6) = 0.0
      DO 2 I=1,2
        PT(I) = TPB
        PZ(I) = DPL
        PS(I) = SPR
        PF(I) = 0.
    2 CONTINUE
```

```
C     ***PRELIMINARY OIL COMPUTATIONS
      NEQ(1) = 1
      NEQ(6) = 2
      IGA = 0
      IF (IAT.NE.2) THEN
        SPMAX = 1.-SWMAX-SAR
        IF (SPMAX.LT.SPR) THEN
C         ***THE WATER AND AIR DO NOT LEAVE ENOUGH PORE SPACE
C         ***FOR THE OIL TO BE MOBILE
          WRITE (15,9540) SWMAX,SAR,SPMAX,SPR
          WRITE (*,9540) SWMAX,SAR,SPMAX,SPR
          STOP 'ERROR: RUN REBUILD AND READ THE *.HSS OUTPUT FILE'
        END IF
        CALL PKE (SPMAX,SWMAX,XKPMX)
        CPART = COINI
        IF (QP.GT.0..OR.QW.GT.0.) CPART = COINI*QP/(QP + XXK1*QW)
      ELSE IF (IAT.EQ.2) THEN
        THPZ = DPL - DPA
        SPMAX = PVOL/ETA/THPZ
        IF (IWR.EQ.1.AND.SPMAX.LT.SPR) WRITE (15,9505) THPZ,SPMAX
        IF (SPMAX.GT.1.-SWMAX-SAR) THEN
          SPMAX = 1.-SWMAX-SAR
          PVOL = SPMAX*ETA*THPZ
          PMAS = PVOL*AC*PRHO*DMKG/DVSV
          IF (IWR.EQ.1) WRITE (15,9510) SPMAX,PVOL,PMAS
        END IF
        CALL PKE (SPMAX,SWMAX,QP)
        XKPMX = QP
        Z = DPL
        DZ = THPZ*0.1
        DO 15 I=101,91,-1
          ZI(I,2) = Z
          ZI(I,1) = Z
          ZZ(I,2) = Z
          Z = Z - DZ
 15     CONTINUE
        Z = DPL
        DO 20 J=1,11
          ZC(J,2) = Z
          ZC(J,1) = Z
          ZZ(J,3) = Z
          Z = Z - DZ
 20     CONTINUE
        CPART = PVOL*COINI/RD(SPMAX,SWMAX)/THPZ
      END IF
      IF (QP.LE.0..AND.IAT.NE.3.AND.IAT.NE.4) THEN
        NEQ(1) = 0
        NEQ(2) = 0
      ELSE IF (QP.LE.XKPMX.AND.IAT.NE.3.AND.IAT.NE.4) THEN
C       ***KINEMATIC OIL FLOW
        SL = SPR
        IF (SL.GT.SPR) SL = QP/PKS
C
C       ***DETERMINE THE OIL SATURATION CORRESPONDING TO THE GIVEN FLUX
        TOL = 1.E-3
 45     CONTINUE
          CALL BISEC (PKE,SL,1.-SWMAX-SAR,TOL,110,QP,SWMAX,SPMAX,XX,IE)
          CALL PKE (SPMAX,SWMAX,XKPMX)
C         ***ASSURE THAT SPMAX GIVES FLUX CLOSE TO LOADING RATE
          IF (ABS(XKPMX-QP).GT.5.E-5) THEN
            IF (TOL.LT.1.E-10) STOP 'OIL-TOLERANCE'
            TOL = TOL*0.1
            GO TO 45
          END IF
C
        DZPMDT = QP/ETA/SPMAX
      ELSE IF (IAT.EQ.3.OR.IAT.EQ.4.OR.QP.GT.XKPMX) THEN
C       ***PRESSURIZED OIL FLOW
        EE = XKPMX*(1.-SPMAX*ETA)
        FF = XKPMX*QP
        ZZ(1,1) = DPA+0.001
        IGA = 1
C       C0(1) = COINI
        CHS(1) = 0.000001
        CHS(2) = 0.
C       ***DETERMINE THE APPROXIMATE CORRECTION FOR CAPILLARY SUCTION
C       ***AT THE OIL FRONT
        CALL CAPSUC (PHIF)
        IF (IAT.EQ.3.OR.IAT.EQ.4) THEN
C         ***SET CONSTANTS FOR USE IN SUBROUTINE GAHS
          GA1 = SPMAX*ETA/XKPMX
          GA2 = ALOG( HS + PHIF)
          GA3 = HS + PHIF
          GA4 = GA3*GA2
          GGA1 = DBLE(SPMAX*ETA/XKPMX)
          GGA3 = DBLE(HS + PHIF)
          GGA2 = DLOG(GGA3)
          GGA4 = GGA3*GGA2
C         ***INITIAL CONDITION
          ZZ(1,1) = DPA
        END IF
        IF (IAT.EQ.4) THEN
          NEQ(1) = 3
          TPEM = TPE
          TPE  = 2.+TPE
          ZZ(3,1) = HS
        END IF
        DO = 0.
        IF (DLO.GT.0.) THEN
          DO = DLO
          DLO = 0.
        END IF
      END IF
      IF (IAT.EQ.1.AND.IGA.EQ.0) THEN
        PMAS = PVOL*AC*PRHO*DMKG/DVSV
        IF (IWR.EQ.1) WRITE (15,9520) QP,PVOL,PMAS
        IF (IWR.EQ.1.AND.COINI.GT.0.)
     *  WRITE (15,9525) AC*QP*DTP*COINI*CVSV
      END IF
      CALL SLOPE (SPMAX,SWMAX,DZPDDT)
```

```
      DZPDDT = DZPDDT/ETA
      ZGA = DPA + QP*DTP/SPMAX/ETA
      PS(1) = SPMAX
      NEQ(2) = 0
      IF (COINI.GT.0..AND.DLO.GT.0.) THEN
        CODM = XXK1*CPART*RD(SPMAX,SWMAX)/RD(0.,SWMAX)
        IF (IWR.EQ.1) WRITE (15,9530) CODM
      END IF
C     ***CONSTANTS FOR MULL'S RECTANGULAR PROFILE MODEL
      VPA = 0.
C
C     ***INITIALIZE OILENS CONSTANTS
      CALL INTLNS (NEQ,ZZ,ND)
C
C     ***INITIALIZE CHARACTERISTICS
      CALL INTCHR (NEQ,ZZ,ND)
C
C     ***WRITE OUT THE CAPILLARY PRESSURE CURVES
      CALL WPC (IWR,XLAMB,SWR,SPR,HWE,HOE)
C
C     ***WRITE HEADINGS FOR OUTPUT FILES IF NECESSARY
      IF (IWR.EQ.1) CALL HEADS (ZZ,ND)
C
C     ***CREATE TEMPORARY FILES FOR THE PLOT FILES
      IF (IWR.EQ.1) CALL PFILE2
C
      RETURN
C
 9400 FORMAT (15X,30HTRAPPED AIR SATURATION      = ,G10.4,' (*)')
 9420 FORMAT (10X,'***LOADING EXCEEDS KINEMATIC RATE CAPACITY OF SOIL'/
     *        10X,'***EXCESS IS ASSUMED TO RUN OFF')
 9500 FORMAT (15X,30HWATER SATURATION            = ,G10.4,' (*)'/
     *        15X,30HWATER FLUX                  = ,G10.4,' (M/D)'/
     *        15X,30HMAX. OIL CONDUCTIVITY       = ,G10.4,' (M/D)')
 9505 FORMAT (15X,30HTHICKNESS OF POLLUTED ZONE  = ,G10.4,' (M)'/
     *        15X,30HMAXIMUM OIL SATURATION      = ,G10.4,' (*)')
 9510 FORMAT (15X,34HOIL SAT. REDUCED TO MAX. ALLOWABLE/
     *        15X,30HADJUSTED OIL SATURATION     = ,G10.4,' (*)'/
     *        15X,30HADJUSTED OIL VOLUME/AREA    = ,G10.4,' (M)'/
     *        15X,30HADJUSTED TOTAL OIL MASS     = ,G10.4,' (KG)'/)
 9520 FORMAT (15X,30HPOLLUANT VOLUME FLUX        = ,G10.4,' (M/D)'/
     *        15X,30HTOTAL OIL LOADING, VOL/AREA = ,G10.4,' (M)'/
     *        15X,30HTOTAL OIL MASS              = ,G10.4,' (KG)')
 9525 FORMAT (15X,30HTOTAL CONSTITUENT MASS      = ,G10.4,' (KG)')
 9530 FORMAT (15X,30HMAX. CONC. DUE TO OIL DECAY = ,G10.4)
 9540 FORMAT (//10X,'THE WATER AND AIR SATURATIONS ARE TOO HIGH'/
     *          10X,'FOR THERE TO BE MOBILE NAPL'/
     *          10X,'WATER SATURATION          = ',F10.4,' (*)'/
     *          10X,'TRAPPED AIR SATURATION    = ',F10.4,' (*)'/
     *          10X,'NAPL SATURATION           = ',F10.4,' (*)'/
     *          10X,'RESIDUAL NAPL SATURATION  = ',F10.4,' (*)'/)
      END
```

```
      SUBROUTINE INTCHR (NEQ,ZZ,ND)
      DIMENSION NEQ(*),ZZ(ND,*)
C     ************************************************************
C     *
C     *  INTCHR INITIALIZES THE VARIABLES WHICH DEFINE THE CONSTITUENT
C     *  AND OIL PHASE CHARACTERISTICS
C     *
C     *  INPUT ARGUMENTS: NONE
C     *
C     *  OUTPUT ARGUMENTS.....................................
C     *  NEQ(J)  = ARRAY OF NUMBER OF EQUATIONS FOR EACH GROUP
C     *  ZZ(I,J) = DEPTH FOR EQUATION I OF GROUP J
C     *  ND      = ROW DIMENSION OF ZZ
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES:  INQSET, SLOPE (ARGUMENT OF INQSET)
C     *  THIS ROUTINE IS CALLED BY: INITIA
C     ************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /OCHR/ DLO,OLAG,SI(101,2),S0(101),T0(101),ZI(101,2)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RLTV/ KRF,KRFO,XLAMB,EPS,EM1,EM2,CC1,REM1
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      EXTERNAL SLOPE,PKE,WKE
      NUM(56) = NUM(56) + 1
C
      IL = 90
      IF (IAT.NE.2) IL = 101
      DO 4  J=1,2
        DO 3 I=1,IL
          ZI(I,J) = DPA
          ZC(I,J) = DPA
          SI(I,J) = SPMAX
          CC(I,J) = CPART
 3      CONTINUE
 4    CONTINUE
C
C     ***OIL PHASE CHARACTERISTICS
C     ***CHARACTERISTIC NUMBERS DECREASE FROM THE LEADING EDGE TO THE
SURFACE
C     ***1.)  DURING THE LOADING
      DT = DTP*0.1
      T  = TPB - DT
      DO 10 I=101,91,-1
        T = T + DT
        T0(I) = T
        S0(I) = SPMAX
 10   CONTINUE
C     ***2.)  AFTER THE END OF THE EVENT
      DS = (SPMAX-SPR)/40.
      SO = SPR - DS
      DO 15 I=1,90
        SO = SO + DS
        T0(I) = TPE
        S0(I) = SO
        SI(I,1) = SO
        SI(I,2) = SO
 15   CONTINUE
C
C     ***CHARACTERISTICS FOR THE CONSTITUENTS
      IF (COINI.EQ.0.) THEN
        NEQ(3) = 0
      ELSE IF (COINI.LT.0.) THEN
        STOP 'INTCHR---COINI.LT.0'
      ELSE IF (COINI.GT.0.) THEN
C       ***CONSITIUENT CHARACTERISTIC NUMBERS INCREASE FROM THE LEADING
C       ***EDGE TO THE SURFACE
        T = TPB - DT
        DO 25 J=1,11
          T = T + DT
          TC(J) = T
          C0(J) = CPART
          NEQ(3) = J
 25     CONTINUE
        IF (IGA.EQ.1) THEN
          C0(1) = COINI
          CC(1,2) = COINI
          CC(1,1) = COINI
C         ***ADD MORE CONSTITUENT CHARACTERISTICS FOR GREEN-AMPT CASES
          DO 40 I=NEQ(3),2,-1
            TC(I+4) = TC(I)
            C0(I+4) = C0(I)
 40       CONTINUE
          DT = DT/5.
          T = TC(1)
          DO 50 I=2,5
            T = T + DT
            TC(I) = T
            C0(I) = CPART
 50       CONTINUE
          NEQ(3) = NEQ(3) + 4
        END IF
      END IF
C
C     ***VOLATILIZATION MODEL PARAMETERS
      IF (XXK3.GT.0.) THEN
        CN = 10./3.
        IF (IAT.EQ.2) NEQ(1) = 2
        ZZ(2,1) = DPA
      END IF
C
C     ***INITIALIZE THE INQUAD ROUTINE FOR THE OIL CHARACTERISTICS
```

```
      CALL INQSET (SLOPE,SPR,1.-SWMAX,SWMAX,50,WP)
C
C     ***SET UP CHARACTERISTIC PATTERN FOR DECAYING OIL PHASE
      IF (DLO.GT.0..AND.QP.GT.0.) NEQ(2) = 101
C
      RETURN
      END
```

```
      SUBROUTINE UNITS
C     ****************************************************************
C     *
C     *  UNITS
C     *
C     *  BUILT IN UNIT CONVERSIONS
C     *  UNIT CONVERSION FACTORS FOR CONCENTRATIONS IN MG/L,
C     *  SIMULATION IN METERS AND DAYS, DENSITY IN GR/CC
C     *  ALL FACTORS ARE IN THE FORM OF THE FOLLOWING RATIOS:
C     *
C     *
C     *
C     * INPUT ARGUMENTS.....(NONE).........................
C     * OUTPUT ARGUMENTS.....(NONE).........................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  INPUT
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      NUM(53) = NUM(53) + 1
C
C
C
C     ***1.) CONCENTRATION MASS/DENSITY MASS
      CMDM = 0.001
C     ***2.) CONCENTRATION MASS/KILOGRAM
      CMKG = 1.E-6
C     ***3.) CONCENTRATION VOLUME/SIMULATION VOLUME
      CVSV = 0.001
C     ***3B.) CONCENTRATION VOLUME/DENSITY VOLUME
      CVDV = 1000.
C     ***4.) DENSITY MASS/GRAMS
      DMGR = 1.
C     ***4B.) DENSITY MASS/KILOGRAMS
      DMKG = 0.001
C     ***5.) DENSITY VOLUME/CUBIC CENTIMETERS
      DVCC = 1.
C     ***6.) DENSITY VOLUME/SIMULATION VOLUME
      DVSV = 1.E-6
C     ***7.) SOLID WATER PARTITION VOLUME/DENSITY VOLUME
      PVDV = 1000.
C     ***8.) DENSITY MASS/SOLID WATER PARTITION MASS
      DMPM = 0.001
C     ***FOR MIXED UNIT CONVERSION OF OILENS OIL LENS HYDROCARBON
C     ***VOLUME AND VOLUME FLUX LOSSES
      UC1 = CMDM*DVSV/CVSV
C     ***FOR MIXED UNIT CONVERSION OF OILENS OUTPUT
      UC2 = CMKG/CVSV
C     ***FOR MIXED UNIT CONVERSION OF SOIL PARTITION COEFFICIENT
      UC3 = PVDV*DMPM
C     ***FOR EMPIRICAL VAPOR PRESSURE MODEL
      UC4 = DMGR*DVCC
C     ***FOR DISSOLUTION OF HYDROCARBONS (CONVERTS MG/L TO GR/CC)
      UC5 = CMDM/CVDV
C
      RETURN
      END
```

```
      SUBROUTINE VG2BC (VGALPA,XN,XLAMB,HWE)
C     ****************************************************************
C     *
C     * VG2BC PERFORMS AN APPROXIMATE CONVERSION BETWEEN VAN GENUCHTEN
C     * AND BROOKS AND COREY PARAMETERS USING A METHOD
C     * DEVELOPED BY LENHARD ET AL. 1989
C     *
C     * INPUT PARAMETERS
C     * VGALPA = VAN GENUCHTENS ALPHA
C     * XN     = VAN GENUCHTENS N
C     * OUTPUT ARGUMENTS
C     * XLAMB  = BROOKS AND COREY LAMBDA
C     * HWE    = BROOKS AND COREY AIR ENTRY HEAD
C     *
C     * JIM WEAVER, VI-7-92
C     * REQUIRED ROUTINES: NONE
C     * ANSI STD X3.9-1978 FORTRAN 77
C     *
C     ****************************************************************
C
C     ***DEFINITION OF M
      XM = 1. - 1./XN
      IF (XM.GT.0.01) THEN
C       ***LAMBDA AT MATCH POINT OF SE = 0.5
        XLAMB = XM/(1.-XM)
        XLAMB = XLAMB*(1.-(0.5**(1./XM)))
C       ***EMPIRICAL MATCH POINT FOR THE ENTRY HEAD DETERMINATION
        SEB = 0.72 - 0.35*EXP(-XN*XN*XN*XN)
C       ***ENTRY HEAD AT MATCH POINT
        HWE = (SEB**(-1./XM) - 1.)**(1.-XM)
        HWE = HWE * (SEB**(1./XLAMB))/VGALPA
      ELSE
        STOP 'VG M OUT OF RANGE'
C       HWE = 1./VALPHA
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE WFFS
C     ****************************************************************
C     *
C     *  WFFS
C     *
C     *  DETERMINE THE WATER SATURATION CORRESPONDING TO THE GIVEN FLUX
C     *
C     *
C     * INPUT ARGUMENTS.................(NONE).....................
C     * OUTPUT ARGUMENTS................(NONE).....................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  BISEC,WKE
C     * CALLED BY  INITIA
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /RLTV/ KRF,KRFO,XLAMB,EPS,EM1,EM2,CC1,REM1
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      EXTERNAL WKE
      NUM(54) = NUM(54) + 1
C
      IF (KRF.EQ.1) THEN
C       ***ANALYTIC FUNCTION INVERSION FOR BURDINE BROOKS AND COREY
        SWMAX = SWR
        IF (QW.GT.0.) SWMAX = SWR + OMSWR*((QW/WKS)**(1./EPS))
      ELSE IF (KRF.EQ.2) THEN
C       ***NUMERIC FUNCTION INVERSION FOR BURDINE VAN GENUCHTEN
        SL = SWR
        IF (SL.GT.SWR) SL = QW/WKS
        TOL = 1.E-3
 40     CONTINUE
          CALL BISEC (WKE,SL,1.,TOL,110,QW,WKS,SWMAX,XX,IE)
C         ***ASSURE THAT SWMAX GIVES FLUX CLOSE TO LOADING RATE
          CALL WKE (SWMAX,WKS,XWMX)
          IF (ABS(XWMX-QW).GT.5.E-5) THEN
C           ***SOLUTION NOT CLOSE ENOUGH
C           ***CUT TOLERANCE ANE TRY AGAIN
            IF (TOL.LT.1.E-10) STOP 'WATER-TOLERANCE'
            TOL = TOL*0.1
            GO TO 40
          END IF
        CONTINUE
      END IF
C
      RETURN
```

```
END
```

```
      SUBROUTINE KRCHK (IRS)
C     ***************************************************************
C     *
C     *  KRCHK  CHECKS THE DERIVATIVE OF RELATIVE PERMEABILITY
C     *
C     *
C     * INPUT ARGUMENTS.................(NONE).......................
C     * OUTPUT ARGUMENTS.............................................
C     *  IRS    REAR SHOCK FLAG  0 = NO REAR SHOCK FORMS
C     *                          1 = REAR SHOCK FORMS
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  SLOPE
C     * CALLED BY   INPUT
C     * IXX-31-1989
C     ***************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(58) = NUM(58) + 1
C
C     ***SET DEFAULTS
      IRS = 0
      SLOLD = 0.0
C
C     ***CHECK DERIVATIVE OF OIL RELATIVE PERMEABILITY
C     ***FOR CURVATURE
      DO 10 SO=SPR+0.0001,SPR+0.001,0.0001
        CALL SLOPE (SO,SWMAX,DKODSO)
        IF (DKODSO.LT.SLOLD) THEN
C          ***SLOPE OF KRO IS NOT ALWAYS MONOTONICALLY INCREASING
C          ***REAR SHOCK WILL FORM
          IRS = 1
          GO TO 25
        END IF
 10   CONTINUE
      DO 15 SO=SPR+0.001,SPR+0.01,0.001
        CALL SLOPE (SO,SWMAX,DKODSO)
        IF (DKODSO.LT.SLOLD) THEN
C          ***SLOPE OF KRO IS NOT ALWAYS MONOTONICALLY INCREASING
C          ***REAR SHOCK WILL FORM
          IRS = 1
          GO TO 25
        END IF
 15   CONTINUE
      DO 20 SO=SPR+0.01,1.-SWMAX,0.01
        CALL SLOPE (SO,SWMAX,DKODSO)
        IF (DKODSO.LT.SLOLD) THEN
C          ***SLOPE OF KRO IS NOT ALWAYS MONOTONICALLY INCREASING
C          ***REAR SHOCK WILL FORM
          IRS = 1
          GO TO 25
        END IF
 20   CONTINUE
 25   CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INTLNS (NEQ,ZZ,ND)
C     ****************************************************************
C     *                                                              *
C     * INTLNS                                                       *
C     *                                                              *
C     * INTLNS INITIALIZES VARIOUS CONSTANTS FOR THE OILENS MODULE   *
C     *                                                              *
C     * INPUT ARGUMENTS.....(NONE).............................      *
C     * OUTPUT ARGUMENTS.......................................      *
C     *   ZZ(I,J) = DEPTH FOR EQUATION I GROUP J                     *
C     *   ND      = ROW DIMENSION OF ZZ(),Z1(), & F()                *
C     *   NEQ(J)  = NUMBER OF EQUATION IN EQCH GROUP                 *
C     *                                                              *
C     *                                                              *
C     * JIM WEAVER                                                   *
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY             *
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY                *
C     * ADA, OKLAHOMA  74820                                         *
C     *                                                              *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77               *
C     * REQUIRED ROUTINES: RD,PKE                                    *
C     * CALLED BY INITIA                                             *
C     *                                                              *
C     ****************************************************************
      DIMENSION NEQ(*),ZZ(ND,*)
C     ***INITIALIZE VARIABLES FOR OILENS CALCUATIONS
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /COUN/ NUM(100)
      COMMON /GLMB/ CCAL,CCMD,CCRA,CCRB,CHLENS,CHLOSS,CHRES
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RESV/ VTRESA,VTRESB
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /VLNS/ ALPHA,AP1,FRACT,FRING,FRING3,PI,RADI,RMF,SPI4,
     *              SQRTPI,VFRING,VIN1,VIN2,CIN1,CIN2,CLTIME,HLTIME
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      EXTERNAL SLOPE,PKE,WKE
      NUM(55) = NUM(55) + 1
C
C     ***INITIALIZE OILENS MODEL PARAMETERS
      RHMAX=0.
      CMDC=0.
      CMDH=0.
      BFLXC=-1.
      BFLXH=-1.
      FRACT=0.
      QLOSSR=0.
      QLOSSH=0.
      VIN1=0.
      VIN2=0.
      VTRESA = 0.
      VTRESB = 0.
C
      XMSOL = SOLENS
      SOLENS = 1.-SWMAX
      IF (SOLENS.GT.XMSOL) SOLENS = XMSOL
      THETO  = ETA*SOLENS
      TRA = ETA*SPR
      TRB = ETA*SPRB
      FRING3 = FRING
      CALL PKE (SOLENS,1.-SOLENS,PERMO)
C
C     ***MODIFY FRING3 BASED UPON WATER TABLE FLUCTUATION
      IF (WFLUC-FRING3.GT.0.) THEN
        VO = (WFLUC-FRING3)*SPR*ETA
        XL = VO/SOLENS/ETA
        FRING3 = FRING3 + XL
      END IF
C
      PERMO  = PERMO*RKS
      RT     = RMF*RADI
      SPI4   = SQRT(PI/4.)
      SQRTPI = SQRT(PI)
      AC     = PI*RADI*RADI
      VFRING = AC*FRING3*ETA*SOLENS
      BW     = RD(SOLENS,1.-SOLENS)
      ZZ(1,4) = 0.
      ZZ(2,4) = 0.
      ZZ(3,4) = RT
      NEQ(4) = 1
      ZZ(1,5) = 0.
      ZZ(2,5) = RT
      NEQ(5) = 1
      CHLENS = 0.
      CHRES  = 0.
      CHLOSS = 0.
      HLTIME = 0.
C
      RETURN
      END
```

```
      SUBROUTINE INCAP (QP,XKPMX,ZF,DPA,PHIF,TPE,TT,QGA,OFLX,RO)
C     ************************************************************
C
C     ***INCAP CALCULATES THE OIL FLUX AND RUNOFF BASED
C     ***ON THE GREEN-AMPT APRROXIMATION OF THE INFILTRATION CAPACITY
C     ***OIL FLUXES GREATER THAN THE GREEN-AMPT FLUX CREATE RUNOFF
C
C     INPUT ARGUMENTS
C     QP    = OIL FLUX AT SURFACE
C     XKPMX = MAXIMUM OIL CONDUCTIVITY
C     ZF    = DEPTH OF OIL FRONT
C     DPA   = DEPTH AT WHICH OIL IS APPLIED
C     PHIF  = CAPILLARY SUCTION AT FRONT
C     TPE   = OIL EVENT ENDING TIME
C     TT    = CURRENT SIMULATION TIME
C
C     OUTPUT ARGUMENTS
C     QGA   = GREEN-AMPT OIL FLUX
C     OFLX  = OIL FLUX IN SOIL
C     RO    = RUNOFF VOLUME FLUX PER UNIT AREA
C
C     JIM WEAVER, X-11-89
C
C     SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     REQUIRED ROUTINES:   NONE
C     THIS ROUTINE IS CALLED BY:  OEQS
C
C     ************************************************************
      COMMON /COUN/ NUM(100)
      NUM(62) = NUM(62) + 1
C
C     ***INITIALIZATION
      RO   = 0.
      OFLX = QP
      IF (TT.GT.TPE) OFLX = XKPMX
      QGA  = XKPMX
C
C     ***GREEN-AMPT FLUX
      IF (ZF.GT.0.) QGA = XKPMX*(ZF-DPA+PHIF)/(ZF-DPA)
C
C     ***OIL FLUX AND RUN-OFF CALCULATION
      IF (QGA.GT.OFLX) RETURN
      RO = OFLX - QGA
      OFLX = QGA
C
      RETURN
      END
```

```
      SUBROUTINE CAPSUC (SUCTER)
C     ************************************************************
C     *                                                          *
C     * CAPSUC  APPROXIMATE CAPILLARY SUCTION AT THE OIL FRONT   *
C     *                                                          *
C     *                                                          *
C     * INPUT ARGUMENTS.....................................     *
C     * OUTPUT ARGUMENTS....................................     *
C     *  SUCTER  CAPILLARY SUCTION CONTRIBUTION TO FLUX          *
C     *                                                          *
C     * JIM WEAVER                                               *
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY         *
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY            *
C     * ADA, OKLAHOMA  74820                                     *
C     *                                                          *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77           *
C     * REQUIRED ROUTINES   GLQ,CAFN                             *
C     * CALLED BY  INITIA                                        *
C     *                                                          *
C     * X-4-1989                                                 *
C     * IIX-9-1990 CORRECTED                                     *
C     *                                                          *
C     ************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /CPSA/ HOE,HWE,OSIG,PHIF,WSIG,XLAMBO
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RLTV/ KRF,KRFO,XLAMB,A2A,A3A,A4A,A5A,A6A
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      EXTERNAL CAFN
      NUM(59) = NUM(59) + 1
C
C
C     ***APPROXIMATE CAPILLARY SUCTION AT OIL WETTING FRONT
C     ***FOR THE BROOKS AND COREY CAPILLARY PRESSURE FUNCTION
      SUCTER = 0.
      S1 = SPR+SWMAX
      ST = SPMAX+SWMAX
C
C     ***SUCTION INTEGRAL AT FRONT
C     ***THE RESIDUAL OIL SATURATION IS ASSUMED TO BE ZERO FOR
C     ***THIS CALCULATION
      SPRTMP = SPR
      OMSTMP = OMSOR
      SPR = 0.
      OMSOR = 1.
      CALL GLQ (CAFN,S1,ST,VAL)
C
C     ***RESET THE RESIDUAL OIL SATURATION
      SPR = SPRTMP
      OMSOR = OMSTMP
      FINT = HOE*VAL/XLAMB/OMSWR
C
C     ***CAPILLARY SUCTION AT ST = SPMAX+SWMAX
      FIM = - HOE*(((ST-SWR)/OMSWR)**(-1./XLAMB))
C     ***APPROXIMATE CAPILLARY SUCTION AT OIL WETTING FRONT
C     ***SUCTER IS ADDED TO THE HEAD TERM IN THE GREEN-AMPT SOLUTION
      SUCTER = FINT/XKPMX - FIM
C
C
      RETURN
      END
```

```
      SUBROUTINE CAFN (ST,F)
C     ***************************************************************
C     *
C     *  CAFN    FUNCTION THAT IS INTEGRATED TO DETERMINE THE CAPILLARY
C     *          SUCTION APPROXIMATION (USED WITH GLQ)
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   ST    = TOTAL LIQUID SATURATION (OIL + WATER)
C     * OUTPUT ARGUMENTS............................................
C     *   F     = INTEGRAND VALUE
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES
C     * CALLED BY  GLQ(CAPSUC)
C     *
C     ***************************************************************
      COMMON /COUN/ NUM(100)
      COMMON /CPSA/ HOE,HWE,OSIG,PHIF,WSIG,XLAMBO
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C     DATA I /0/
      NUM(60) = NUM(60) + 1
C
      IF (NUM(60).EQ.1) THEN
        I = 0
      END IF
C
C     ***INITIALIZE
      IF (I.EQ.0) THEN
        X1 = -(1. + 1./XLAMBO)
        I = 1
      END IF
C
C     ***FUNCTION VALUE
      SO = ST - SWMAX
      CALL PKE (SO,SWMAX,OKE)
      F = OKE*((ST-SWR)/OMSWR)**X1
C
      RETURN
      END
```

```
      SUBROUTINE POST (ETIME,CLTIME,HLTIME,ZZ,ND)
C     ***************************************************************
C     *
C     * POST   POSTPROCESSING FOR OILENS
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   ETIME   ENDING TIME OF SIMULATION
C     *   CLTIME  TIME THAT CONSTITUENT LEACHES FROM THE LENS
C     *   HLTIME  TIME THAT OIL LENS IS CREATED
C     *   ZZ(,)   VALUES OF THE DEPENDENT VARIABLES
C     *   ND      ROW DIMENSION OF ZZ
C     *
C     * OUTPUT ARGUMENTS.......(NONE)..............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  HSSM
C     *
C     ***************************************************************
      REAL ZZ(ND,*)
      COMMON /COUN/ NUM(100)
      COMMON /CNVS/ CMDM,CMKG,CVSV,CVDV,DMGR,DMKG,DVCC,DVSV,PVDV,DMPM,
     *              UC1,UC2,UC3,UC4,UC5
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /LFLX/ CINL,CMDC,CMDH,FLXGWH,RMC,SOLENS,THETO,TRA,TRB,QINL
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
      NUM(43) = NUM(43) + 1
C
C
C     ***RETURN WHEN NO OILENS
      IF (WTABLE.LT.0.) RETURN
C
      CTIME = ETIME-CLTIME
      HTIME = ETIME-HLTIME
C     ***POST PROCESSING
C     ***FOR OILENS:
      IF (HTIME.GT.0.) HMF = CMDH/HTIME
      IF (CTIME.GT.0.) CMF = CMDC/CTIME
      IF (KSTOP.EQ.3) CMF = BFLXC
C
C     ***RETURN IF NO OUTPUT DESIRED
C     CALL ZMASS (ETIME,WTABLE,SH,NEQ,IW,ZZ,ND,VVM,CVL)
C
      IF (IWR.NE.1) RETURN
      IF (ZZ(2,4)+ZZ(1,5).LE.0..AND.ZZ(1,5).LE.0.) RETURN
C
C     ***WRITE OUT OILENS RESULTS
      IF (ZZ(2,4)+ZZ(1,5).GT.0.) THEN
        WRITE (8,9100)
        WRITE (8,9101) ETIME,QW
        WRITE (8,9102) KSTOP
        IF (KSTOP.EQ.4) WRITE (8,9103) OPERC
        WRITE (8,9104)
      END IF
      IF (KSTOP.NE.3) WRITE (8,9107)
      IF (ZZ(2,4).GT.0.) WRITE (8,9108) CMDH*UC2,HMF*UC2,ZZ(3,4)
      IF (KSTOP.EQ.3) WRITE (8,9115)
      IF (ZZ(1,5).GT.0.)
     1 WRITE (8,9120) CMDC*UC2,CMF*UC2,BFLXC*UC2,ZZ(2,5)
C
      RETURN
 9100 FORMAT (//10X,'KOPT/OILENS MODEL POST-PROCESSING'/
     2         10X,'================================='/)
 9101 FORMAT (10X,'ENDING TIME OF SOLUTION                      = ',
     1            G12.4,' DAYS'/
     2        10X,'INFILTRATION RATE                            = ',
     3            G12.4,' M/D '/)
 9102 FORMAT (10X,'ENDING CRITERIA, KSTOP                       = ',I5)
 9103 FORMAT (10X,'FOR KSTOP = 4 ENDING FACTOR, OPERC           = ',
     1           G12.4)
 9104 FORMAT (10X,'KSTOP VALUES--KOPT/OILENS IS STOPPED BY: '/
     4        10X,'1  USER SPECIFIED ENDING TIME'/
     5        10X,'2  OIL LENS MOTION CEASES'/
     6        10X,'3  CONTAMINANT MASS FLUX BEGINS TO BE REDUCED'/
     7        10X,'   FROM ITS MAXIMUM VALUE'/
     8        10X,'4  CONTAMINANT MASS IN THE OIL LENS IS REDUCED'/
     9        10X,'   TO (OPERC)*TOTAL CUMULATIVE CONTAMINANT MASS'/
     1        10X,'   FLUX INTO THE LENS'/)
 9107 FORMAT (/10X,'***DISSOLUTION FLUXES AVERAGED OVER SIMULATION ',
     1            'TIME***'/)
 9108 FORMAT (10X,'CUMULATIVE HYDROCARBON DISSOLVED INTO AQUIFER = ',
     1            G12.4,' KG '/
     2        10X,'AVERAGE HYDROCARBON DISSOLUTION FLUX         = ',
     3            G12.4,' KG/D'/
     4        10X,'MAXIMUM RADIUS OF HYDROCARBON LENS           = ',
     5            G12.4,' M'/)
 9115 FORMAT (10X,'DISSOLUTION FLUX GIVEN AS MAXIMUM FLUX OVER'/
     1        10X,'SIMULATION TIME ')
 9120 FORMAT (10X,'CUMULATIVE CONTAMINANT DISSOLVED INTO AQUIFER = ',
     1            G12.4,' KG '/
     2        10X,'AVERAGE CONTAMINANT DISSOLUTION FLUX         = ',
     3            G12.4,' KG/D'/
     4        10X,'MAXIMUM CONTAMINANT DISSOLUTION FLUX         = ',
     5            G12.4,' KG/D'/
     6        10X,'MAXIMUM RADIUS OF CONTAMINANT                = ',
     7            G12.4,' M'/)
      END
```

```
      SUBROUTINE COUNT
C     ****************************************************************
C     *
C     *  COUNT   COUNTS THE NUMBER OF TIMES CERTAIN SUBROUTINES ARE CALLED
C     *
C     *
C     *
C     * INPUT ARGUMENTS.................(NONE).......................
C     * OUTPUT ARGUMENTS................(NONE).......................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  HSSM,CHK
C     *
C     ****************************************************************
      CHARACTER NT*15
      COMMON /COUN/ NUM(100)
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /NAME/ NT(15)
      COMMON /OUTP/ ERCMX,EROMX,NPT,NREG
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      CHARACTER CN(100)*6
      DATA (CN(I),I=1,66) /'MAIN','EQS','CHK','YWRITE','GAHS',
     1     'RD','CVEL','CONCE','OCP','ZMASS','OIL','HIST',
     2     'SFLUX','INPUT','INITIA','COUNT','MRKF12',' ','INQUAD',
     3     'BINARY','BISEC','WKE',' ','PKE','SLOPE',' ','KRSET',
     4     'INQSET','OEQS','CEQS','OCHK','SCHK','SCHK2','OCHK2','CCHK',
     5     'CCHK2','OFLUX','OILENS','LCHK','LCHK2','VOLT','ERF','POST',
     6     'CLENS','HMASS','CMASS','PROLNS','DUPUIT','MAXSAV',
     7     'REPACK','MESSAG','DATA','UNITS','WFFS','INTLNS','INTCHR',
     8     'HEADS','KRCHK','CAPSUC','CAFN','GLQ','INCAP','DHCHK2','DSO',
     *     'LFLUX',' '/
      NUM(16) = NUM(16) + 1
C     ***RUN INFORMATION
C
C
C     ***SUBROUTINE CALLS
      IF (IWR.EQ.1) WRITE (8,9500) (NT(II),II=1,15)
      IF (IWR.EQ.1) WRITE (8,9501)
      N1 = 0
      NEND = 61
 90   CONTINUE
        IF (N1.GT.NEND) GO TO 100
        N1 = N1 + 1
        IF (CN(N1).EQ.' ') GO TO 90
        N2 = N1 + 1
 95     CONTINUE
          IF (CN(N2).EQ.' '.AND.N1.NE.NEND) THEN
            N2 = N2 + 1
            GO TO 95
          END IF
          IF (IWR.EQ.1.AND.CN(N2).NE.' ') THEN
            WRITE (8,9503) N1,CN(N1),NUM(N1),N2,CN(N2),NUM(N2)
          ELSE
            IF (IWR.EQ.1) WRITE (8,9503) N1,CN(N1),NUM(N1)
          END IF
          N1 = N2
          GO TO 90
 100  CONTINUE
C
C     ***RUN STATISTICS
      X1 = FLOAT(NUM(2))
      X2 = FLOAT(NPT+NREG)
      X3 = FLOAT(NPT)
      X4 = 100.*FLOAT(NREG)/X3
      X5 = FLOAT(ISE)/X1
      IF (IWR.EQ.1)
     1 WRITE (8,9820) NPT+NREG,NPT,NREG,X4,NUM(2),X1/X2,X1/X3,X5
      IF (IWR.EQ.1) WRITE (8,9900)
C
      RETURN
C     ***OUTPUT FORMATS
 9500 FORMAT (1H1//17X,'KOPT--KINEMATIC OILY POLLUTANT TRANSPORT'/
     2        10X,20X,15HRUN INFORMATION/
     3        10X,54(1H*)/
     4        3(12X,5A10/)/)
 9501 FORMAT (10X,20HROUTINE        CALLS,14X,20HROUTINE        CALLS/
     1        10X,20(1H=),14X,20(1H=)/)
 9503 FORMAT (8X,I4,3X,A6,2X,I7,12X,I4,3X,A6,2X,I7)
 9820 FORMAT (//15X,'RUNGE-KUTTA-FEHLBERG SOLVER PERFORMANCE'/
     1        10X,54(1H=)//
     2        10X,'NUMBER OF TIME STEPS TAKEN               ',I10/
     3        10X,'ACCEPTED STEPS                           ',I10/
     4        10X,'REJECTED STEPS                           ',I10/
     5        10X,'PERCENT STEPS REJECTED                   ',F10.4/
     6        10X,'FUNCTION EVALUATIONS                     ',I10/
     7        10X,'AVERAGE NO. EVALUATIONS PER STEP         ',F10.4/
     8        10X,'AVERAGE NO. EVALUATIONS PER ACCEPTED STEP ',F10.4/
     9        10X,'AVERAGE NO. EQUATIONS SOLVED PER EVAL.    ',F10.4/)
 9900 FORMAT (//10X,50(1H*)/
     4        10X,'SUCCESSFUL EXECUTION'/
     5        10X,50(1H*))
      END
```

```
      SUBROUTINE WPC (IWR,XLAMB,SWR,SOR,HWE,HOE)
C     ****************************************************************
C     *
C     *  WPC   WRITES OUT THE WATER/AIR AND OIL/AIR CAPILLARY PRESSURE
C     *        CURVES USED BY HSSM
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   IWR    INTEGER WRITE FLAG
C     *   XLAMB  BROOKS AND COREY LAMBDA
C     *   SWR    RESIDUAL WATER SATURATION
C     *   SOR    RESIDUAL OIL SATURATION
C     *   HWE    WATER/AIR ENTRY HEAD
C     *   HOE    OIL/AIR ENTRY HEAD
C     * OUTPUT ARGUMENTS.........(NONE).............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  INPUT
C     *
C     ****************************************************************
C
C
C
      IF (IWR.EQ.0) RETURN
C
C
C     ***SET CONSTANTS
      X1 = (-1./XLAMB)
      OMSWR = 1.-SWR
      OMSOR = 1.-SOR
C
C
C     ***WATER/AIR CURVE
      WRITE (15,9100)
      DO 10 SW=SWR,0.98,0.02
        SWE = (SW-SWR)/OMSWR
        IF (SW.LE.SWR) GO TO 10
        HCW = HWE*(SWE**X1)
        SO = SW
        SOE = (SO-SWR)/OMSWR
        HCO = HOE*(SOE**X1)
        WRITE (15,9110) SW,HCW,HCO
 10   CONTINUE
      WRITE (15,9110) 1.0,HWE,HOE
      WRITE (15,9130)
C
C
      RETURN

9100 FORMAT (1H1///10X,'WATER-AIR, NAPL-AIR CAPILLARY PRESSURE CURVE'/
     *              10X,'****************************************'//
     *              10X,'WATER or NAPL',1X,'CAPILLARY      '
     *                 ,1x,'CAPILLARY'/
     *              10X,'SATURATION   ',1X,'HEAD (CM WATER)'
     *                 ,1x,'HEAD (CM NAPL)'/
     *              10X,'================================='
     *                 ,'================'/)
9110 FORMAT (10X,3(F10.4,5X))
9120 FORMAT (1H1///10X,'NAPL-AIR CAPILLARY PRESSURE CURVE'/
     *              10X,'*********************************'//
     *              10X,'NAPL      ',1X,'CAPILLARY'/
     *              10X,'SATURATION',1X,'HEAD (CM NAPL)'/
     *              10X,'================================'/)
9130 FORMAT (/)
      END
```

```
      SUBROUTINE WTHICK
C     ****************************************************************
C     *                                                              *
C     * WTHICK                                                       *
C     *                                                              *
C     * THE WTHICK SUBROUTINE GENERATES AN INPUT DATA SET FOR THE    *
C     * NTHICK UTILITY.   (NTHICK ESTIMATES THE OIL SATURATION IN THE*
C     * NAPL LENS.)                                                  *
C     *                                                              *
C     * INPUT ARGUMENTS.................(NONE).....................  *
C     * OUTPUT ARGUMENTS................(NONE).....................  *
C     *                                                              *
C     * JIM WEAVER                                                   *
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY             *
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY                *
C     * ADA, OKLAHOMA  74820                                         *
C     *                                                              *
C     * 11-2-92                                                      *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77               *
C     * REQUIRED ROUTINES NONE                                       *
C     * CALLED BY INPUT                                              *
C     *                                                              *
C     ****************************************************************
      CHARACTER*40 OFILE(7),MFILE(3)
      CHARACTER C1*1,NAME*12,TF*40
      COMMON /CPSA/ HOE,HWE,OSIG,PHIF,WSIG,XLAMBO
      COMMON /FILE/ MFILE,OFILE
      COMMON /LENS/ A,AC,B,BW,RHMAX,PERMO,PRHO,QLOSSR,QLOSSH,SPRB,ZLN(2)
      COMMON /MATR/ ETA,OMSOR,OMSWR,SAR,SR,SWR
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      COMMON /RLTV/ KRF,KRFO,XLAMB,A2A,A3A,A4A,A5A,A6A
      COMMON /TRAN/ DWV,EVAP,HLC,HLO,PMU,RH,RHOS,TEMP,WMU,WRHO,XXKS,
     *              XXKSH,XXKV
C
C
C
C     ***EXTRACT THE NECESSARY FILE NAME
      TF = OFILE(1)
      DO 100 I=1,40
C       ***FIND THE DECIMAL POINT IN THE FILE NAME
        C1 = TF(I:I)
        IF (C1.EQ.'.') THEN
          INDEC = I
          GO TO 101
        END IF
  100 CONTINUE
  101 CONTINUE
C
C     ***BUILD THE FILE NAME BEGINNING 8 CHARACTERS TO THE LEFT OF THE .
      IN = 0
C     ***GENERATE DO LOOP INDICES
      IBEG = INDEC-8
      IF (IBEG.LE.0) THEN
C       ***STARTING INDEX COULD BE LT ZERO IF NO PATH AND THE
C       ***FILE NAME IS LESS THAN 8 CHARACTERS
        IBEG = 1
      END IF
      IEND = INDEC
C
C
C     ***SEE IF THERE IS A PATH INCLUDED
      IPNAME = 0
      DO 150 I=IBEG,IEND
        IF (TF(I:I).EQ.'\') THEN
          IPNAME = I
        END IF
  150 CONTINUE
C
      IF (IPNAME.NE.0) THEN
        IBEG = IPNAME
      END IF
C
C     **BUILD THE NAME
      DO 200 I=IBEG,IEND
        IF (TF(I:I).NE.' ') THEN
          IN = IN + 1
          NAME(IN:IN) = TF(I:I)
        ELSE IF (TF(I:I).EQ.'.') THEN
          IN = IN + 1
          NAME(IN:IN) = TF(I:I)
          GO TO 201
        END IF
  200 CONTINUE
  201 CONTINUE
C
C     ***ADD THE EXTENSION "NTH"
      NAME(IN+1:IN+1) = 'N'
      NAME(IN+2:IN+2) = 'T'
      NAME(IN+3:IN+3) = 'H'
C
C
C     ***OPEN THE FILE FOR THE NTHICK DATA SET
      OPEN (45,FILE=NAME,STATUS='UNKNOWN')
C
C     ***WRITE THE DATA NEEDED BY NTHICK
      WRITE (45,9100) ETA,HWE,XLAMB,SWR
      WRITE (45,9100) SPR,SPRB
      WRITE (45,9100) WSIG,OSIG
      WRITE (45,9100) WRHO,PRHO
C
      CLOSE (45)
C
 9100 FORMAT (4F10.4)
```

```
RETURN
END
```

```
      SUBROUTINE TSGP1 (IFILE,OFILE,KKSTOP,IFACE,
     *DLONG,DTRAN,VDISP,TAQU,PMAX,CMINW,BEGT,ENDT,
     *TINC,NWELL,XWELL,YWELL,VEL,ETA,XXKS,RHOS,ZLAM)
C     ************************************************************
C     *
C     *  TSGP1    BEGINS (BUT DOES NOT END) THE CREATION OF THE TSGPLUME
C     *           INPUT DATA FILE
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   IFILE   HSSM-KO INPUT DATA FILE NAME
C     *   OFILE   HSSM-KO OUTPUT FILE NAMES
C     *   KKSTOP  INTEGER HSSM-KO ENDING FLAG
C     *   IFACE   CHARACTER DATA-CREATION INTERFACE FLAG
C     *   DLONG   LONGITUDINAL DISPERSION COEFFICIENT
C     *   DTRAN   TRANSVERSE DISPERSION COEFFICIENT
C     *   VDISP   VERTICAL TRANSVERSE DISPERSIVITY
C     *   TAQU    THICKNESS OF THE AQUIFER
C     *   PMAX    PERCENT MAX OIL LENS SIZE
C     *   CMINW   MINIMUM TSGPLUME OUTPUT CONCENTRATION
C     *   BEGT    SIMULATION BEGINNING TIME
C     *   ENDT    SIMULATION ENDING TIME
C     *   TINC    SIMULATION TIME INCREMENT
C     *   NWELL   NUMBER OF EXPOSURE POINTS
C     *   XWELL   X-COORDINATES OF THE EXPOSURE POINTS
C     *   YWELL   Y-COORDINATES OF THE EXPOSURE POINTS
C     *   VEL     SEEPAGE VELOCITY
C     *   ETA     POROSITY
C     *   XXKS    SOIL/WATER PARTITION COEFFICIENT
C     *   RHOS    BULK DENSITY
C     *   ZLAM    DECAY COEFFICIENT FOR DILUATION
C     * OUTPUT ARGUMENTS.....................(NONE).................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES
C     * CALLED BY
C     *
C     ************************************************************
      CHARACTER IFACE*1,NT*15
      CHARACTER*40 IFILE,OFILE(*)
      DIMENSION XWELL(*),YWELL(*)
      COMMON /NAME/ NT(15)
C
      R = 1. + RHOS*XXKS/ETA
      VS = VEL/ETA
C
      WRITE (14,9109) IFILE,OFILE(1),OFILE(5),OFILE(6),
     *OFILE(7),KKSTOP,IFACE
      WRITE (14,9111) (NT(I),I=1,15)
      WRITE (14,9110) DLONG,DTRAN,VDISP,VS,ETA,TAQU
      WRITE (14,9110) R,PMAX,CMINW,ZLAM
      WRITE (14,9110) BEGT,ENDT,TINC
      WRITE (14,9120) NWELL
      DO 10 I=1,NWELL
        WRITE (14,9110) XWELL(I),YWELL(I)
 10   CONTINUE
C
      RETURN
 9109 FORMAT (5(1X,A40/),1X,I10,1X,A1)
 9110 FORMAT (1X,6(G10.4,1X))
 9111 FORMAT (5A10/5A10/5A10)
 9120 FORMAT (1X,I10)
      END
```

```
      SUBROUTINE TSGP3 (QW)
C     ****************************************************************
C     *
C     *
C     *  TSGP3    ADD THE RECHARGE RATE TO THE TSGPLUME INPUT FILE
C     *
C     *
C     * INPUT ARGUMENTS.........................................
C     * OUTPUT ARGUMENTS........................................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  INPUT
C     *
C     ****************************************************************
C
C
      WRITE (14,9110) QW
C
      RETURN
 9110 FORMAT (1X,6(G10.4,1X))
      END
```

```
      SUBROUTINE HEADS (ZZ,ND)
C     ****************************************************************
C     *
C     *  HEADS   PLACES HEADINGS IN THE OUTPUT FILES
C     *
C     * INPUT ARGUMENTS.............................................
C     *    ZZ   VALUES OF DEPENDENT VALUES
C     *    ND   ROW DIMENSION OF ARRAY ZZ
C     * OUTPUT ARGUMENTS.....................(NONE).................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY   INITIA
C     *
C     ****************************************************************
      DIMENSION ZZ(ND,*)
      CHARACTER NT*15
      COMMON /COUN/ NUM(100)
      COMMON /CCHR/ CPART,CC(101,2),C0(101),TC(101),ZC(101,2)
      COMMON /CONC/ CLAG,COINI,DLC,FCV,XXK1,XXK2,XXK3,XXKO
      COMMON /FRNT/ ISE,PF(2),PRO(2),PS(2),PT(2),PZ(2)
      COMMON /VAPO/ ABL,AKS,CN,CV(2),CVF(2),DAIR,XJV(2),ZV(2)
      COMMON /HIPR/ IHIST,NTIMES,PR(10),NZS,HI(5)
      COMMON /GAEQ/ CHS(2),DO,EE,FF,GA1,GA2,GA3,GA4,HS,IGA,IP,TPEM,ZGA
      COMMON /GRND/ COXY,VDISP,VEL,WFLUC,WTABLE
      COMMON /NAME/ NT(15)
      COMMON /POLL/ DPA,DPL,DTP,DZPMDT,DZPDDT,IAT,PKS,PVOL,QP,SPMAX,SPR,
     1              THPZ,TPB,TPE,TPR,WP(50),XKPMX,XK22
      NUM(57) = NUM(57) + 1
C     ***OIL FRONT
      WRITE (15,9580) (NT(II),II=1,15)
      WRITE (15,9585)
      IF (IGA.EQ.1) THEN
        WRITE(15,9588)1,TPB,ZZ(1,1),SPMAX,PF(2),0.,0.,HS
      ELSE
        WRITE (15,9588)1,TPB,ZZ(1,1),SPMAX,QP,0.,0.
      END IF
C
C     ***CONSTITUENT FRONT
      IF (COINI.GT.0.) THEN
        WRITE (11,9590) (NT(II),II=1,15)
        WRITE (11,9595)
      END IF
C
C     ***HISTORIES
      IF (NZS.GT.0) THEN
        WRITE (10,9100) (NT(I),I=1,15),(HI(I),I=1,NZS)
        IF (NZS.EQ.1) WRITE (10,9102)
        IF (NZS.EQ.2) WRITE (10,9103)
        IF (NZS.EQ.3) WRITE (10,9104)
        IF (WTABLE.GT.0) WRITE (10,9105)
      END IF
C
      RETURN
 9580 FORMAT (1H1//8X,50(1H*)/
     *        21X,'LOCATION OF THE NAPL FRONT'/
     *        8X,50(1H*)/
     *        3(8X,5A10/)/)
 9585 FORMAT (/16X,23X,'NAPL'/
     *        18X,48(1H-)/
     *        1X,' STEP',4X,'TIME',2X,3X,'DEPTH',2X,'SATURATION',5X,
     *           'FLUX',2X,3X,'RUNOFF',1X,3X,'MASS',3X,1X,'PONDING',
     *           2X,'UCHAR SPD.','FRONT SPD.','DCHAR SPD.'/
     *        10X,'(D)',1X,2X,3X,1X,'(M)',1X,2X,4X,'(*)',3X,5X,'(M/D)',
     *           1X,3X,'(KG)',4X,1X,1X,'(KG)',3X,3X,'(M)',4X,
     *           2X,'(M/D)',3X,2X,'(M/D)',3X,2X,'(M/D)',3X/
     *        1X,76(1H=)/)
 9588 FORMAT (1X,I5,12F10.4)
 9590 FORMAT (1H1//8X,50(1H*)/
     *        21X,'LOCATION OF THE CONSTITUENT FRONT'/
     *        8X,50(1H*)/
     *        3(8X,5A10/)/)
 9595 FORMAT (/
     *        16X,19X,'CONSTITUENT'/
     *        18X,38(1H-)/
     *        1X,' STEP',4X,'TIME',2X,9X,'DEPTHS',5X,'CONC-WATER',1X,
     *        'MASS'/
     *        16X,4X,'LOWER',1X,4X,'UPPER'/
     *        1X,55(1H=)/)
 9100 FORMAT (1H1,//10X,50(1H*)/
     *        10X,'      SATURATION AND CONCENTRATION HISTORIES'/
     *        10X,50(1H*)/
     *        3(10X,5A10/)/
     *        1X,10H   TIME   ,12X,'DEPTH(S)'/
     *        10X,3(10X,F10.4,10X))
 9102 FORMAT (10X,(4X,'SAT',3X,4X,'FLUX',2X,3X,'CONC',3X)/
     *        1X,9(1H=),(1X,29(1H=)))
 9103 FORMAT (10X,2(4X,'SAT',3X,4X,'FLUX',2X,3X,'CONC',3X)/
     *        1X,9(1H=),2(1X,29(1H=)))
 9104 FORMAT (10X,3(4X,'SAT',3X,4X,'FLUX',2X,3X,'CONC',3X)/
     *        1X,9(1H=),3(1X,29(1H=)))
 9105 FORMAT (10X,'WARNING:  WHEN HISTORIES ARE WITHIN AN OIL LENS'/
     *        10X,'          ZERO OIL FLUX IS GIVEN'/)
      END
```

```
      SUBROUTINE YWRITE (TT)
C     ****************************************************************
C     *
C     *
C     * WRITE SCREEN MESSAG AT THE END OF EACH YEARS SIMULATION
C     *
C     *
C     *
C     * INPUT ARGUMENTS.........................................
C     *   TT     TIME IN DAYS
C     * OUTPUT ARGUMENTS......(NONE)..............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES:  NONE
C     * CALLED BY:  CHK
C     *
C     ****************************************************************
      COMMON /COUN/ NUM(100)
C     DATA YRS /730./
      NUM(4) = NUM(4) + 1
C
C
C
      IF (NUM(4).EQ.1) THEN
        YRS = 730.
      END IF
C
C
C     ***WRITES A SCREEN MESSAGE AFTER THE SIMULATION OF EACH YEAR
      IF (TT.GT.YRS) THEN
        CALL MESS2 (17,YRS/365.)
        YRS = YRS + 365.
      END IF
C
C
      RETURN
      END
```

```
      SUBROUTINE MESSAG (IM)
C     ****************************************************************
C     ***WRITE MESSAGES TO THE SCREEN
C     ***REQUIRED ROUTINES:   NONE
C     ****************************************************************
      CHARACTER*40 STR
      COMMON /COUN/ NUM(100)
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      NUM(51) = NUM(51) + 1
C
C     ***ENTRY MESS2 IS USED TO OUTPUT A REAL*4 NUMBER WITH A MESSAG
      ENTRY MESS2 (IM,X)
C
C     ***ENTRY MESS3 IS USED TO OUTPUT A CHARACTER*40 STRING WITH MESSAGE
      ENTRY MESS3 (IM,STR)
C
C     ***ENTRY MESS4 IS USED TO OUTPUT AN INTEGER WITH A MESSAGE
      ENTRY MESS4 (IM,INTE)
C
      IF (IM.EQ.1) THEN
C       ***KOPT BANNER
        WRITE (*,9100)
        WRITE (*,9101)
        READ (*,*)
        WRITE (*,9102)
        READ (*,*)
      END IF
C
      IF (IWR.EQ.0) THEN
        IF (IM.EQ.30) THEN
C          ***MONTE CARLO SIMULATION NUMBER
          WRITE (*,9330) INTE
        END IF
        RETURN
      END IF
C
      IF (IM.EQ.2) THEN
C       ***DATA INITIALIZATION
        WRITE (*,9120)
      ELSE IF (IM.EQ.3) THEN
C       ***NUMERICAL CALCUATION
        WRITE (*,9130)
      ELSE IF (IM.EQ.4) THEN
C       ***POST PROCESSING
        WRITE (*,9140)
      ELSE IF (IM.EQ.5) THEN
C       ***KOPT-END
        WRITE (*,9150)
      ELSE IF (IM.EQ.6) THEN
C       ***DATA INPUT
        WRITE (*,9160)
      ELSE IF (IM.EQ.7) THEN
C       ***OILENS STARTS
        WRITE (*,9170)
      ELSE IF (IM.EQ.8) THEN
C       ***OILENS ENDS
        WRITE (*,9180)
      ELSE IF (IM.EQ.9) THEN
C       ***CHEMICAL REACHES LENS
        WRITE (*,9190)
      ELSE IF (IM.EQ.10) THEN
C       ***OIL INFILTRATION
        WRITE (*,9200)
      ELSE IF (IM.EQ.11) THEN
C       ***OIL REDISTRIBUTION
        WRITE (*,9210)
      ELSE IF (IM.EQ.12) THEN
C       ***SIMULATION END
        WRITE (*,9212)
      ELSE IF (IM.EQ.13) THEN
C       ***FILE IO
        WRITE (*,9213) STR
      ELSE IF (IM.EQ.14) THEN
C       ***INVALID DATA FILE
        WRITE (*,9214)
      ELSE IF (IM.EQ.15) THEN
C       ***PROFILE AT TIME X
        WRITE (*,9215) X
      ELSE IF (IM.EQ.16) THEN
C       ***TSGPLUME DATA FILE
        WRITE (*,9216) STR
      ELSE IF (IM.EQ.17) THEN
C       ***SIMULATION OF EACH YEAR
        WRITE (*,9217) X
      ELSE IF (IM.EQ.18) THEN
C       ***KOPT/OILENS PLOT FILE CREATION
        WRITE (*,9218) STR
      ELSE IF (IM.EQ.19) THEN
C       ***PLOT FILE CHECKING
        WRITE (*,9219)
      ELSE IF (IM.EQ.20) THEN
C       ***REPACKING FILE 18
        WRITE (*,9220)
      ELSE IF (IM.EQ.21) THEN
C       ***REPACKING FILE 19
        WRITE (*,9221)
      END IF
C
      RETURN
 9100 FORMAT (/
     *       15X,'*************************************************'/
     1       15X,'*                                               *'/
     *       15X,'*                     HSSM                      *'/
```

```
     3        15X,'*                                                  *'/
     *        15X,'*         HYDROCARBON SPILL SCREENING MODEL        *'/
     5        15X,'*                                                  *'/
     3        15X,'*  INCLUDING THE KOPT, OILENS AND TSGPLUME MODELS  *'/
     6        15X,'*                                                  *'/
     7        15X,'*                JAMES W. WEAVER                   *'/
     8        15X,'*  UNITED STATES ENVIRONMENTAL PROTECTION AGENCY    *'/
     9        15X,'*   R.S. KERR ENVIRONMENTAL RESEARCH LABORATORY     *')
 9101 FORMAT (15X,'*              ADA, OKLAHOMA 74820                 *'/
     2        15X,'*                                                  *'/
     5        15X,'* RANDALL CHARBENEAU, SUSAN SHULTZ, MIKE JOHNSON   *'/
     *        15X,'*  ENVIRONMENTAL AND WATER RESOURCES ENGINEERING   *'/
     6        15X,'*        THE UNIVERSITY OF TEXAS AT AUSTIN         *'/
     7        15X,'*                                                  *'/
     *        15X,'*                RSKERL V1.00                      *'/
     *        15X,'**************************************************'/
     1       /15X,'PRESS RETURN')
 9102 FORMAT (15X,'**************************************************'/
     *        15X,'*                  WARNING:                        *'/
     *        15X,'* THIS PROGRAM SIMULATES IDEALIZED BEHAVIOR OF     *'/
     *        15X,'* OILY-PHASE CONTAMINANTS IN IDEALIZED POROUS      *'/
     *        15X,'* MEDIA, AND IS NOT INTENDED FOR APPLICATION TO    *'/
     *        15X,'* HETEROGENEOUS SITES.                             *'/
     *        15X,'* THE MODEL RESULTS HAVE NOT BE VERIFIED BY EITHER*'/
     *        15X,'* LAB OR FIELD STUDIES.                            *'/
     *        15X,'* READ USER GUIDE FOR FURTHER INFORMATION BEFORE   *'/
     *        15X,'* ATTEMPTING TO USE THIS PROGRAM.                  *'/
     *        15X,'* NEITHER THE AUTHOR, THE UNIVERSITY OF TEXAS,     *'/
     *        15X,'* NOR THE UNITED STATES GOVERNMENT ACCEPTS ANY     *'/
     *        15X,'* LIABILITY RESULTING FROM ITS USAGE.              *'/
     *        15X,'* THE U.S. E.P.A. DOES NOT OFFICIALLY ENDORSE THE  *'/
     *        15X,'* USE OF THIS CODE.                                *'/
     *      15X,'**************************************************'/
     *      /15X,'PRESS RETURN')
 9120 FORMAT (10X,'*** DATA INITIALIZATION          ')
 9130 FORMAT (10X,'*** SIMULATION BEGINNING         ')
 9140 FORMAT (10X,'*** POST PROCESSING              ')
 9150 FORMAT (10X,'*** HSSM END                     '//)
 9160 FORMAT (/10X,'*** DATA INPUT                   ')
 9170 FORMAT (10X,'*** OIL LENS FORMS               ')
 9180 FORMAT (10X,'*** OIL LENS MOTION STOPS        ')
 9190 FORMAT (10X,'*** CHEMICAL REACHES WATER TABLE ')
 9200 FORMAT (10X,'*** OIL INFILTRATION             ')
 9210 FORMAT (10X,'*** OIL REDISTRIBUTION           ')
 9212 FORMAT (10X,'*** SIMULATION END               ')
 9213 FORMAT (10X,'*** CREATING OUTPUT FILE: '/
     *        10X,'*** ',A40)
 9214 FORMAT (10X,'*** INVALID INPUT DATA FILE     ')
 9215 FORMAT (10X,'*** PROFILING AT ',F12.4,' DAYS ')
 9216 FORMAT (10X,'*** CREATING HSSM-T DATA FILE: '/
     *        10X,'*** ',A40)
 9217 FORMAT (10X,'*** SIMULATION OF ',F5.1,' YEARS COMPLETED')
 9218 FORMAT (10X,'*** CREATING HSSM-KO PLOT FILE: '/
     *        10X,'*** ',A40)
 9219 FORMAT (10X,'*** PROCESSING PLOT FILE CONTENTS')
 9220 FORMAT (10X,'*** REPACKING FILE 18')
 9221 FORMAT (10X,'*** REPACKING FILE 19')
 9330 FORMAT (10X,'*** (MONTE CARLO/SENSITIVITY) SIMULATION ',I10)
      END
```

```
      SUBROUTINE SOPEN (IFILE)
C     ****************************************************************
C     *
C     *  OPEN THE INPUT DATA FILE WITH STANDARD FORTRAN 77 COMMANDS
C     *  THE USER IS PROMPTED FOR AN INPUT FILE NAME
C     *
C     *
C     * INPUT ARGUMENTS.....................................
C     * OUTPUT ARGUMENTS....................................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES   PREKOP,DIR,MESSAG
C     * CALLED BY  DATA
C     *
C     ****************************************************************
      CHARACTER*40 IFILE
      LOGICAL EXIST
      COMMON /COUN/ NUM(100)
C
C
C
      IFILE = ' '
 5    CONTINUE
C
C     ***PROMPT FOR THE NAME OF THE INPUT DATA FILE
      WRITE (*,*) ' '
      WRITE (*,*) 'ENTER THE NAME OF THE INPUT DATA FILE'
      WRITE (*,*) '----+----*-40-CHARACTER-LIMIT*----+----*'
      READ (*,'(A)') IFILE
C
C     ***CHECK FOR EXISTENCE OF FILE NAME
      INQUIRE (FILE=IFILE,EXIST=EXIST)
      IF (EXIST.EQV..FALSE.) THEN
        WRITE (*,*) 'INPUT DATA FILE DOES NOT EXIST--REENTER'
        WRITE (*,*) ' '
        GO TO 5
      END IF
C
      OPEN (4,FILE=IFILE,STATUS='OLD')
C
      RETURN
      END
```

```
      SUBROUTINE NSOPEN (IFILE)
C     ****************************************************************
C     *
C     *  NSOPEN   NONSTANDARD OPENING OF THE INPUT FILE
C     *           FOR COMPATIBILITY WITH HSSM-WIN
C     *
C     *
C     *
C     * INPUT ARGUMENTS........(NONE).............................
C     * OUTPUT ARGUMENTS.....................................
C     *   IFILE    INPUT DATA FILE NAME
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  DATA
C     *
C     ****************************************************************
      CHARACTER*40 IFILE
C
C     ***OPEN THE HSSM-KO INPUT FILE
      OPEN (4,FILE=' ',STATUS='OLD')
      INQUIRE (4,NAME=IFILE)
C
      RETURN
      END
```

```
      SUBROUTINE PKCON (OFILE1,OFILE2,OFILE3,IWR)                       C
C     ****************************************************************  C
C     *                                                                 C
C     *  PKCON    CONCATENATES TEMPORARY FILES TO CREATE THE PLOT FILES
C     *                                                                 C
C     *                                                                 
C     *                                                                 
C     * INPUT ARGUMENTS.............................................    C
C     *   OFILE1   SATURATION PROFILE PLOT FILE NAME                    
C     *   OFILE2   OIL LENS PROFILE PLOT FILE NAME                      C
C     *   OFILE3   LENS RADIUS AND CONSTITUENT DATA FILE NAME           
C     *   IWR      WRITE FLAG (WRITE IF IWR = 1)                        
C     * OUTPUT ARGUMENTS............................................    C
C     *                                                                 
C     * JIM WEAVER                                                      
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY                
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY                   C
C     * ADA, OKLAHOMA  74820                                            C
C     *                                                                 
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77                  
C     * REQUIRED ROUTINES  CMD,MESS3                                    
C     * CALLED BY  PFILE                                                
C     *                                                                 
C     ****************************************************************  
      CHARACTER*40 OFILE1,OFILE2,OFILE3                                 
      CHARACTER*256 STR                                                 C
      INTEGER I4                                                        
      LOGICAL EXIST                                                     
      COMMON /FLAG/ ICONC,IKOPT,ILENS,ITSGP                             C
C                                                                       C
C     ***CONCANTINATE OUTPUT FILES                                      
      IF (IWR.EQ.0) RETURN                                              
C                                                                       
      CALL MESS3 (18,OFILE1)                                            C
C     ***PLOT FILE 1 CONTAINING THE KOPT SATURATION/DEPTH PROFILES      C
      REWIND 20                                                         C
      REWIND 16                                                         
      REWIND 22                                                         C
C                                                                       
C     ***CHECK FOR EXISTENCE OF THE OUTPUT FILES                        
      IE = 0                                                            C
      INQUIRE (FILE='T20.TMP',EXIST=EXIST)                              
      IF (EXIST.EQV..FALSE.) IE = IE + 1                                C
      INQUIRE (FILE='T16.TMP',EXIST=EXIST)                              
      IF (EXIST.EQV..FALSE.) IE = IE + 1                                C
      INQUIRE (FILE='T22.TMP',EXIST=EXIST)                              
      IF (EXIST.EQV..FALSE.) IE = IE + 1                                
C                                                                       
      IF (IE.EQ.0) THEN                                                 
        STR = 'COPY T20.TMP+T16.TMP+T22.TMP '//OFILE1//' > OFILE1'      C
C                                                                       C
C       ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION          
        CLOSE (20)                                                      
        CLOSE (16)                                                      
        CLOSE (22)                                                      
C                                                                       C
C       ***EXECUTE OPERATING SYSTEM COMMAND, STR                        
C       ***RETURNS ERROR CODE, I4                                       
        CALL CMD (STR,I4)                                               
C                                                                       
        STR = 'DEL OFILE1'                                              
        CALL CMD  (STR,I4)                                              
C                                                                       
      END IF                                                            
C                                                                       
      IF (ILENS.EQ.1) THEN                                              
        CALL MESS3 (18,OFILE2)                                          
C       ***PLOT FILE 2 CONTAINING THE OILENS RADIAL PROFILES            
        REWIND 21                                                       
        REWIND 17                                                       
        REWIND 22                                                       
C                                                                       C
C       ***CHECK FOR EXISTENCE OF THE OUTPUT FILES                      C
        IE = 0                                                          
        INQUIRE (FILE='T21.TMP',EXIST=EXIST)                            
        IF (EXIST.EQV..FALSE.) IE = IE + 1                              
        INQUIRE (FILE='T17.TMP',EXIST=EXIST)                            
        IF (EXIST.EQV..FALSE.) IE = IE + 1                              
        INQUIRE (FILE='T22.TMP',EXIST=EXIST)                            
        IF (EXIST.EQV..FALSE.) IE = IE + 1                              
C                                                                       
        IF (IE.EQ.0) THEN                                               
          STR = 'COPY T21.TMP+T17.TMP+T22.TMP '//OFILE2//' > OFILE2'    
C                                                                       C
C         ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION        
          CLOSE (21)                                                    
          CLOSE (17)                                                    
          CLOSE (22)                                                    
C                                                                       C
C         ***EXECUTE OPERATING SYSTEM COMMAND, STR                      C
C         ***RETURNS ERROR CODE, I4                                     
          CALL CMD (STR,I4)                                             
C                                                                       
          STR = 'DEL OFILE2'                                            
          CALL CMD (STR,I4)                                             
C                                                                       
        END IF                                                          
C                                                                       
        CALL MESS3 (18,OFILE3)                                          
C       ***PLOT FILE 3 CONTAINING THE OILENS RADIUS AND CONSITUENT DATA 
        REWIND 21                                                       
        REWIND 18                                                       
        REWIND 19                                                       
        REWIND 22                                                       
C                                                                       C
C       ***CHECK FOR EXISTENCE OF THE OUTPUT FILES                      
        IE = 0                                                          
        INQUIRE (FILE='T21.TMP',EXIST=EXIST)                            
        IF (EXIST.EQV..FALSE.) IE = IE + 1                              
        INQUIRE (FILE='T18.TMP',EXIST=EXIST)                            
```

```
          IF (EXIST.EQV..FALSE.) IE = IE + 1
          INQUIRE (FILE='T19.TMP',EXIST=EXIST)
          IF (EXIST.EQV..FALSE.) IE = IE + 1
          INQUIRE (FILE='T22.TMP',EXIST=EXIST)
          IF (EXIST.EQV..FALSE.) IE = IE + 1

C
          IF (IE.EQ.0) THEN
          STR='COPY T21.TMP+T18.TMP+T19.TMP+T22.TMP '//OFILE3//' > OFILE3'
C
C           ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
            CLOSE (21)
            CLOSE (18)
            CLOSE (19)
            CLOSE (22)
C
C           ***EXECUTE OPERATING SYSTEM COMMAND, STR
C           ***RETURNS ERROR CODE, I4
            CALL CMD (STR,I4)
C
            STR = 'DEL OFILE3'
            CALL CMD (STR,I4)
C
          END IF
        END IF
C
        RETURN
        END
```

```
      SUBROUTINE DELF
C     ****************************************************************   C
C     *                                                                    
C     *  DELF   DELETES TEMPORARY OUTPUT FILES                             
C     *                                                                    
C     *                                                                    
C     *                                                                    
C     * INPUT ARGUMENTS.............(NONE)..........................       
C     * OUTPUT ARGUMENTS............(NONE)..........................       
C     *                                                                    
C     * JIM WEAVER                                                         
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY                   
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY                      
C     * ADA, OKLAHOMA  74820                                               
C     *                                                                    
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77                     
C     * REQUIRED ROUTINES  NONE                                            
C     * CALLED BY  HSSM                                                    
C     *                                                                    
C     ********************************************************************
C
      REWIND 4
C
C     ***REOPEN FILES AND DELETE THEM
      OPEN (7,FILE='T7.TMP',STATUS='OLD')
      CLOSE (7,STATUS='DELETE')
      OPEN (8,FILE='T8.TMP',STATUS='OLD')
      CLOSE (8,STATUS='DELETE')
      OPEN (9,FILE='T9.TMP',STATUS='OLD')
      CLOSE (9,STATUS='DELETE')
      OPEN (10,FILE='T10.TMP',STATUS='OLD')
      CLOSE (10,STATUS='DELETE')
      OPEN (11,FILE='T11.TMP',STATUS='OLD')
      CLOSE (11,STATUS='DELETE')
      OPEN (12,FILE='T12.TMP',STATUS='OLD')
      CLOSE (12,STATUS='DELETE')
      OPEN (13,FILE='T13.TMP',STATUS='OLD')
      CLOSE (13,STATUS='DELETE')
      OPEN (14,FILE='T14.TMP',STATUS='OLD')
      CLOSE (14,STATUS='DELETE')
      OPEN (15,FILE='T15.TMP',STATUS='OLD')
      CLOSE (15,STATUS='DELETE')
      OPEN (16,FILE='T16.TMP',STATUS='OLD')
      CLOSE (16,STATUS='DELETE')
      OPEN (17,FILE='T17.TMP',STATUS='OLD')
      CLOSE (17,STATUS='DELETE')
      OPEN (18,FILE='T18.TMP',STATUS='OLD')
      CLOSE (18,STATUS='DELETE')
      OPEN (19,FILE='T19.TMP',STATUS='OLD')
      CLOSE (19,STATUS='DELETE')
      OPEN (20,FILE='T20.TMP',STATUS='OLD')
      CLOSE (20,STATUS='DELETE')
      OPEN (21,FILE='T21.TMP',STATUS='OLD')
      CLOSE (21,STATUS='DELETE')
      OPEN (22,FILE='T22.TMP',STATUS='OLD')
      CLOSE (22,STATUS='DELETE')
      RETURN
      END
```

```
      SUBROUTINE IOPOST (OFILE,IWR,IREO)
C     ************************************************************   C
C     *                                                                C
C     *  IOPOST   CONCATENATES THE HSSM-KO OUTPUT FILE (*.HSS)         C
C     *                                                                C
C     *                                                                C
C     *                                                                C
C     * INPUT ARGUMENTS.............................................
C     *   OFILE   HSSM-KO OUTPUT FILE NAME
C     *   IWR     WRITE FLAG (WRITE OUTPUT FILES IF IWR = 1)
C     *   IREO    ECHO PRINT ONLY FLAG (IF IREO = 1)
C     * OUTPUT ARGUMENTS.........(NONE)............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  CMD,MESS3
C     * CALLED BY  HSSM
C     *
C     ************************************************************
      CHARACTER*40 OFILE
      CHARACTER*256 STR
      INTEGER IN(8)
      INTEGER I4
      DATA IN /15,11,12,13,7,9,10,8/
C
C     ***CONCANTINATE OUTPUT FILES
      IF (IWR.EQ.0) RETURN
      CALL MESS3 (13,OFILE)
C
      DO 5 I=7,15
        REWIND I
  5   CONTINUE
C
      IEND = 8
      STR = 'COPY T15.TMP+T11.TMP+T12.TMP+T13.TMP+T7.TMP+T9.TMP+'//
     *'T10.TMP+T8.TMP '//OFILE//' > OFILE'
      IF (IREO.EQ.0) THEN
        IEND = 1
        STR = 'COPY T15.TMP '//OFILE//' > OFILE'
      END IF
C
C     ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
      DO 30 I=1,IEND
        CLOSE (IN(I))
 30   CONTINUE
C
C     ***EXECUTE OPERATING SYSTEM COMMAND, STR
C     ***RETURNS ERROR CODE, I4
      CALL CMD (STR,I4)
C
      STR = 'DEL OFILE'
      CALL CMD (STR,I4)
C
      RETURN
      END
```

```
      SUBROUTINE TSGP2 (TFILE,IWR,IREO)
C     ****************************************************************
C     *
C     *  TSGP2   CREATES THE TSGPLUME INPUT FILE
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   TFILE   HSSM-T INPUT FILE NAME
C     *   IWR     WRITE FLAG (WRITE OUTPUT FILES IF IWR = 1)
C     *   IREO    ECHO PRINT ONLY FLAG (IF IREO = 1)
C     * OUTPUT ARGUMENTS............................................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES
C     * CALLED BY
C     *
C     ****************************************************************
      CHARACTER*40 TFILE
      CHARACTER*256 STR
      INTEGER I4
C
C     ***CONCANTINATE OUTPUT FILES
      IF (IWR.EQ.0.OR.IREO.EQ.0) RETURN
      CALL MESS3 (16,TFILE)
C
      REWIND 14
      REWIND 7
C
      STR = 'COPY T14.TMP+T7.TMP '//TFILE//' > TFILE'
C
C     ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
      CLOSE (14)
      CLOSE (7)
C
C     ***EXECUTE OPERATING SYSTEM COMMAND, STR
C     ***RETURNS ERROR CODE, I4
      CALL CMD (STR,I4)
C
      STR = 'DEL TFILE'
      CALL CMD (STR,I4)
C
      RETURN
      END
```

```
      SUBROUTINE OPNFLE
C     ****************************************************************
C     *
C     *  OPNFLE
C     *
C     *  OPEN THE TEMPORARY FILES NEEDED BY HSSM FOR OUTPUTTING MODEL C
C     *   RESULTS
C     *  CHECK FOR PRIOR EXISTENCE OF INPUT FILES
C     *
C     *
C     *
C     * INPUT ARGUMENTS........(NONE).............................
C     * OUTPUT ARGUMENTS.......(NONE).............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  INPUT
C     *
C     ****************************************************************
      LOGICAL FEXIST
C
C     ***AQUEOUS CONCENTRATIONS FROM THE LENS
      INQUIRE (FILE='T7.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (7,FILE='T7.TMP',STATUS='UNKNOWN')
        CLOSE (7,STATUS='DELETE')
      END IF
      OPEN (7,FILE='T7.TMP',STATUS='UNKNOWN')
C
C     ***POST PROCESSING
      INQUIRE (FILE='T8.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (8,FILE='T8.TMP',STATUS='UNKNOWN')
        CLOSE (8,STATUS='DELETE')
      END IF
      OPEN (8,FILE='T8.TMP',STATUS='UNKNOWN')
C
C     ***FILE 9 IS THE PROFILE FILE
      INQUIRE (FILE='T9.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (9,FILE='T9.TMP',STATUS='UNKNOWN')
        CLOSE (9,STATUS='DELETE')
      END IF
      OPEN (9,FILE='T9.TMP',STATUS='UNKNOWN')
C
C     ***
      INQUIRE (FILE='T10.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (10,FILE='T10.TMP',STATUS='UNKNOWN')
        CLOSE (10,STATUS='DELETE')
```

```
      END IF
      OPEN (10,FILE='T10.TMP',STATUS='UNKNOWN')
C
C     ***CONSTITUENT IN THE VADOSE ZONE
      INQUIRE (FILE='T11.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (11,FILE='T11.TMP',STATUS='UNKNOWN')
        CLOSE (11,STATUS='DELETE')
      END IF
      OPEN (11,FILE='T11.TMP',STATUS='UNKNOWN')
C
C     ***OIL LENS OUTPUT
      INQUIRE (FILE='T12.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (12,FILE='T12.TMP',STATUS='UNKNOWN')
        CLOSE (12,STATUS='DELETE')
      END IF
      OPEN (12,FILE='T12.TMP',STATUS='UNKNOWN')
C
C     ***AQUEOUS PHASE CONCENTRATION FROM THE LENS (TITLES)
      INQUIRE (FILE='T13.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (13,FILE='T13.TMP',STATUS='UNKNOWN')
        CLOSE (13,STATUS='DELETE')
      END IF
      OPEN (13,FILE='T13.TMP',STATUS='UNKNOWN')
C
      INQUIRE (FILE='T14.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (14,FILE='T14.TMP',STATUS='UNKNOWN')
        CLOSE (14,STATUS='DELETE')
      END IF
      OPEN (14,FILE='T14.TMP',STATUS='UNKNOWN')
C
C     ***FILE 15 IS THE DATA ECHO PRINT AND MAIN KOPT OUTPUT FILE
      INQUIRE (FILE='T15.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (15,FILE='T15.TMP',STATUS='UNKNOWN')
        CLOSE (15,STATUS='DELETE')
      END IF
      OPEN (15,FILE='T15.TMP',STATUS='UNKNOWN')

C     ***FILE 16 IS THE SATURATION PROFILE PLOT FILE
      INQUIRE (FILE='T16.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (16,FILE='T16.TMP',STATUS='UNKNOWN')
        CLOSE (16,STATUS='DELETE')
      END IF
      OPEN (16,FILE='T16.TMP',STATUS='UNKNOWN')
C
C     ***FILE 17 IS THE OILENS RADIAL SECTION PLOT FILE
      INQUIRE (FILE='T17.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (17,FILE='T17.TMP',STATUS='UNKNOWN')
        CLOSE (17,STATUS='DELETE')
      END IF
      OPEN (17,FILE='T17.TMP',STATUS='UNKNOWN')
C
C     ***FILE 18 IS THE OILENS RADIUS PLOT FILE
      INQUIRE (FILE='T18.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (18,FILE='T18.TMP',STATUS='UNKNOWN')
        CLOSE (18,STATUS='DELETE')
      END IF
      OPEN (18,FILE='T18.TMP',STATUS='UNKNOWN')
C
C     ***FILE 19 IS THE OILENS CONTAMINANT PLOT FILE
      INQUIRE (FILE='T19.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (19,FILE='T19.TMP',STATUS='UNKNOWN')
        CLOSE (19,STATUS='DELETE')
      END IF
      OPEN (19,FILE='T19.TMP',STATUS='UNKNOWN')
C
      DO 10 I=7,19
        REWIND I
 10   CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DIR
C     ****************************************************************
C     *
C     *  DIR ISSUES A DOS DIRECTORY COMMAND
C     *
C     *
C     *
C     * INPUT ARGUMENTS......(NONE)..............................
C     * OUTPUT ARGUMENTS.....(NONE)..............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  CMD
C     * CALLED BY   DPF,SOPEN
C     *
C     ****************************************************************
      CHARACTER*256 STR
      INTEGER I4
C
C     ***DISPLAY DIRECTORY
      STR = 'DIR |MORE'
      CALL CMD (STR,I4)
C
      WRITE (*,*) ' '
C
      RETURN
      END
```

```
      Interface to integer*2 function system [c]
     &  (string[reference])
      character*1 string
      end
      subroutine cmd(a,k)
c     ****************************************************************
c     *
c     *  cmd   issues a dos command from fortran code
c     *
c     *
c     *
c     * input arguments.............................................
c     *   a     character string containing dos command
c     * output arguments............................................
c     *   k     length of command string
c     *
c     * jim cloud
c     * computer services corp
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * single precision fortran 77
c     * required routines   none
c     * called by  delf,iopost,tsgp2,pkcon
c     *
c     *
c     *
c     ****************************************************************
      character a*(*), cstr*256,CSTR2*256
      integer k
      integer*2 system
      integer*1 nul
      data nul/0/
c
      k=len_trim(a)
      if(k.gt.255)return
      cstr=a(1:k)//nul
c
c     ***TURN OFF SCREEN ECHO OF COMMAND
      CSTR2='ECHO OFF'//NUL
      K=INT(SYSTEM(CSTR2))
c
c     ***EXECUTE DOS COMMAND
      k=int(system(cstr))
c
c     ***TURN ON SCREEN ECHO OF COMMANDS
      CSTR2='ECHO ON'//NUL
      K=INT(SYSTEM(CSTR2))
c
      return
      end
```

3.2 Source Code for HSSM-T

```
      PROGRAM HSSMT
C   ************************************************************
C   ************************************************************
C   **
C   **    HSSMT   THE TRANSIENT SOURCE GAUSSIAN PLUME MODULE
C   **            OF THE Hydrocarbon Spill Screening Model (HSSM)
C   **
C   **   THIS PROGRAM SOLVES THE TWO-DIMENSIONAL ADVECTION-
C   **   DISPERSION EQUATION FOR VARIABLE OR CONSTANT INPUT
C   **   SOURCE CONCENTRATIONS. THE SOURCE IS MODELED AS A
C   **   GAUSSIAN DISTRIBUTION WITH THE GIVEN CONCENTRATION AT
C   **   MEAN LOCATION. THE MODEL ASSUMES A CONSTANT SEEPAGE
C   **   VELOCITY, CONSTANT LONGITUDINAL AND TRANSVERSE
C   **   DISPERSION COEFFICIENTS, CONSTANT RETARDATION FACTOR
C   **   AND FIRST ORDER DECAY RATE. ALL SUBROUTINES AND
C   **   FUNCTIONS CARRY OUT A SPECIFIC TASK AND IS EXPLAINED
C   **   AT THE BEGINNING OF THE CODING.
C   **
C   **   THIS VERSION OF TSGPLUME HAS BEEN ADAPTED FOR USE WITH
C   **   THE HSSM MODEL AND ITS INTERFACES.
C   **
C   **
C   **
C   ************************************************************
C   **
C   **   WARNING:
C   **   THE RESULTS FROM THIS CODE HAVE NOT BEEN VERIFIED BY
C   **   EITHER LABORATORY OR FIELD STUDIES.
C   **   THIS PROGRAM SIMULATES IDEALIZED BEHAVIOR OF OILY-PHASE
C   **   CONTAMINANTS IN IDEALIZED POROUS MEDIA, AND IS NOT INTENDED
C   **   FOR APPLICATION TO HETEROGENEOUS SITES.
C   **   THE SUITIBILITY OF THIS PROGRAM FOR ANY PURPOSE IS NOT IM-
C   **   PLIED.  NEITHER THE AUTHOR, THE UNIVERSITY OF TEXAS, NOR
C   **   THE UNITED STATES GOVERNMENT ACCEPTS ANY LIABILITY RESULTING
C   **   FROM ITS USAGE.
C   **
C   ************************************************************
C   **
C   **
C   **   REFERENCES:
C   **   Weaver, J.W., R.J. Charbeneau, and B.K. Lien, A Screening
C   **     Model for Vadose Zone Nonaqueous Phase Liquid Transport
C   **     Using Green-Ampt and Kinematic Wave Theory, Water Resources
C   **     Research, 30(1), 1994
C   **   Weaver, J.W., R.J. Charbeneau, J.D. Tauxe, B.K. Lien, and
C   **     J.B. Provost, The Hydrocarbon Spill Screening Model (HSSM)
C   **     Volume 1:  User's Guide, US EPA, 1994, EPA/600/R-94/39a
C   **  Charbeneau, R.J. and J.W. Weaver, The Hydrocarbon Spill Screening
C   **     Model Volume 2: Theoretical Background and Source Codes,
C   **     US EPA, 1994, EPA/600/R-94/039b
C   **
C   **
C   **   AUTHOR: MIKE JOHNSON
C   **           U.T. AUSTIN
C   **           SPRING 1989
C   **
C   **   MODIFIED FOR INCLUSION IN HSSM BY
C   **      JIM WEAVER
C   **      US EPA
C   **      ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C   **      ADA, OKLAHOMA   74820
C   **
C   **   DOUBLE PRECISION FORTRAN 77
C   **   REQUIRED ROUTINES: INFO,REELIM,LOSORT,ADVECT,REDUE,PREPLOT,
C   **                      PRELIM,NONDIM,SOLN,ROMINT,BINARY,NEWOUT,
C   **                      ENDPLT,MESSAG,DPF,MBISEC,MEPS,RFUNC,FILES,
C   **                      DIRECT,CMD,NSOPEN,PCON,DELFI,CHKFLE,FN,BIN
C   **
C   ************************************************************
C   ************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      CHARACTER*40 FNAME(2)
      CHARACTER*40 TEXT(2)
      CHARACTER*40 IFILE
      DIMENSION CMAX(6),IN(2),NC(6),TCMAX(6)
      COMMON /AUTO/ ADVT,DADVT
      COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
      COMMON /DIMXYT/ XX(6),XY(6)
      COMMON /XYT/ X(6),Y(6)
      COMMON /SOURCE/ B1(31),TB1(31),XB(1000),TB(1000)
      COMMON /PROPS/ R,AL,AT,SIG,VEL,ZLAM,POR,THCK
      COMMON /OTHR/ D,CLAM,CMINW
C
C
C
      CALL MESSAG (1)
C
C     ***CHECK FOR PRIOR EXISTENCE OF OUTPUT FILES
      CALL CHKFLE
C
C     ***OPEN FILE FOR I/O
      NFILES = 2
      DATA IN /42,43/
      ISD = 1
C
C
C
 5    CONTINUE
C
C
C     ***NONSTANDARD INPUT USED ON THE FIRST FILE
C     ***STANDARD INPUT USED ON SUBSEQUENT (FROM A USER COMMAND
C     ***TO CHANGE INPUT DATA SET)
      IF (ISD.EQ.0) THEN
C       ***STANDARD FORTRAN INPUT
```

```
        TEXT(1) = 'TSGPLUME OUTPUT FILE'
        TEXT(2) = 'TSGPLUME PLOTTING FILE'
        CALL FILES (NFILES,TEXT,IN,FNAME)
      ELSE IF (ISD.EQ.1) THEN
C       ***NONSTANDARD INPUT
        CALL NSOPEN (IFILE)
        ISD = 0
      END IF
C
      ICOMP = 0
      CALL INFO (ISD,ICOMP,FNAME,RMAX,TMAX,IEND,IINP)
      IF (IEND.EQ.1) STOP 'END TSGPLUME'
      IF (IINP.EQ.1) GO TO 5
      NWA = 0
C
C     ***DETERMINE THE FLOATING POINT [DOUBLE] PRECISION
C     ***NEEDED FOR SOUBROUTINE MBISEC
      CALL MEPS (EPPS)
C
C     ***DETERMINE THE CONCENTRATION HISTORY FOR EACH RECEPTOR POINT
C     ***(MAXIMUM OF NWELL)
      DO 10 IWELL=1,NWELL
C
        CALL MESS5 (3,IWELL)
        CALL MESS4 (12,XX(IWELL)+RMAX,XY(IWELL))
C
C       ***DO NO CALCULATION FOR POINTS WITHIN THE RADIUS OF THE LENS
        IF (XX(IWELL).LT.0.) THEN
          CALL MESS4 (17,XX(IWELL)+RMAX,XY(IWELL))
          GO TO 10
        END IF
C
C       ***LOWER BOUND FOR ADVECTIVE TRAVEL TIME TO RECEPTOR
C       ***SOLUTION FROM LOWER BOUND UP TO (NOT INCLUDING) ADVECTIVE
C       ***TRAVEL TIME
        CALL ADVECT (IWELL,EPPS,TMAX,RMAX,NC(IWELL),ICOMP)
        IF (NC(IWELL).EQ.0) GO TO 10
C
C       ***DETERMINE THE CONCENTRATION HISTORY AT THE RECEPTOR
C       ***FROM THE ADVECTIVE TRAVEL TIME TO WHEN THE CONCENTRATION
C       ***FALLS BELOW CMINW
C       ***ICOMP = 0 if no concentrations computed
C       ***ICOMP = 1 IF AT LEAST ONE CONCENTRATION COMPUTED
C       ***CMAX  = MAXIMUM CONCENTRATION
C       ***TCMAX = TIME TO OCCURANCE OF MAXIMUM CONCENTRATION
C       ***IWELL = RECEPTOR POINT INDEX
C       ***RMAX  = INPUT MASS FLUX RADIUS
C       ***ISET  = FLAG SET TO ZERO FOR EACH CALL TO SOLN
        ISET = 0
        CALL SOLN (ICOMP,CMAX(IWELL),TCMAX(IWELL),IWELL,RMAX,ISET)
C
C       ***END THE PLOT FILE FOR THIS RECEPTOR LOCATION
        CALL ENDPLT (ICOMP,NWELL)
C
 10   CONTINUE
C
C
C
C
C     ***COMPLETE THE PLOT FILE
      ICOMP = 0
      CALL ENDPLT (ICOMP,NWELL)
      CALL PREPLOT (ICOMP,NUMT,CMAX,TCMAX,TB,XB,NWELL,NC,NUMTT)
C     ***CREATE THE OUTPUT FILES
      CALL PCON (FNAME(1),FNAME(2))
C     ***DELETE THE TEMPORARY FILES
      CALL DELFI
      CALL MESSAG (4)
C
C
C
C
      STOP '         *** SUCCESSFUL EXECUTION OF HSSM-T'
      END
```

```
      SUBROUTINE INFO (ISD,ICOMP,FNAME,RMAX,TMAX,IEND,IINP)
C   ***************************************************************
C   ***************************************************************
C   **
C   **  P U R P O S E
C   **
C   **   THIS ROUTINE READS THE INPUT DATA
C   **
C   **
C   **  I N P U T    P A R A M E T E R S
C   **
C   **         NUMX... INTEGER. THE NUMBER OF X-COORDINATES
C   **         NUMY... INTEGER. THE NUMBER OF Y-COORDINATES
C   **         NUMB... INTEGER. THE NUMBER OF INPUT CONCENTRATIONS
C   **         XX .... REAL ARRAY CONTAINING THE X-COORDINATES
C   **         XY .... REAL ARRAY CONTAINING THE Y-COORDINATES
c   **         btime.. real beginning time for output
c   **         etime.. real ending time for output
c   **         tinte.. real time interval between outputs
C   **         XB .... REAL ARRAY CONTAINING THE INPUT MASS FLUX (M/T)
C   **         TB .... REAL ARRAY CONTAINING THE TIMES FOR INPUTS
C   **         R  .... REAL. THE RETARDATION FACTOR
C   **         AL .... REAL. THE LONGITUDINAL DISPERSIVITY (L)
C   **         AT .... REAL. THE TRANSVERSE DISPERSIVITY (L)
C   **         SIG.... REAL. THE STANDARD DEVIATION OF INPUT SOURCE (L)
C   **         VEL.... REAL. THE AVERAGE (UNIFORM) SEEPAGE VELOCITY (L/T)
C   **         ZLAM... REAL. THE DECAY COEFFICIENT OF THE SOLUTE (1/T)
C   **         POR.... REAL. THE POROSITY OF THE MEDIUM
C   **         THCK... REAL. THE THICKNESS OF THE PLUME
C   **         ISD ... INTEGER FILE OPEN FLAG 0=STANDARD FORTRAN
C   **                                        1=NONSTANDARD
C   **
C   **
C   **  O U T P U T    P A R A M E N T E R S
C   **
C   **         X  .... REAL ARRAY CONTAINING THE NONDIMENSIONAL X-VALUES
C   **         Y  .... REAL ARRAY CONTAINING THE NONDIMENSIONAL Y-VALUES
C   **         T  .... REAL ARRAY CONTAINING THE NONDIMENSIONAL TIMES
C   **         B1 .... REAL ARRAY CONTAINING THE NONDIMENSIONAL INPUT
C   **                 MASS FLUX
C   **         TB1.... REAL ARRAY CONTAINING THE NONDIMENSIONAL INPUT
C   **                 TIMES
C   **         D  .... REAL. THE NONDIMENSIONAL DISPERSION COEFFICIENT
C   **         CLAM... REAL. THE NONDIMENSIONAL DECAY COEFFICIENT
C   **
C   **     INPUT ARGUMENTS.....................................
C   **        ISD      FLAG FOR STANDARD OR NONSTANDARD OPENING OF FILES
C   **        ICOMP    COMPUTED CONCENTRATION FLAG
C   **                 ICOMP = 0, NO CONCENTRATIONS COMPUTED
C   **                 ICOMP = 1, CONCENTRATIONS COMPUTED
C   **        FNAME()  FNAME(1) IS THE OUTPUT FILE NAME
C   **                 FNAME(2) IS THE PLOT FILE NAME
C   **
C   **     OUTPUT AQGUMENTS.....................................
C   **       RMAX    EFFECTIVE LENS RADIUS FOR THE SIMULATION
C   **       TMAX    TIME OF MAXIMUM MASS FLUX INTO THE AQUIFER
C   **       IEND    ENDING FLAG (END PROGRAM IF IEND = 1)
C   **       IINP    FLAG FOR FILE CHANGE
C   **
C   **
C   **    AUTHOR: MIKE JOHNSON
C   **            U.T. AUSTIN
C   **    MODIFIED BY JIM WEAVER, US EPA
C   **    LANGUAGE: DOUBLE PRECISION FORTRAN 77
C   **    REQUIRED ROUTINES: DPF,LOSORT,MESSAG,NONDIM,PCON,PRELIM,REELIM,
C   **                       REDUCE
C   **
C   ***************************************************************
C   ***************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      CHARACTER IFACE*1,NT*15
      CHARACTER*40 IFILE,OFILE(3),MFILE,FNAME(*)
      CHARACTER*5 ENDCHR
      CHARACTER*1 ANS
      DIMENSION TM(3)
      EXTERNAL RFUNC
      COMMON /AUTO/ ADVT,DADVT
      COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
      COMMON /DIMXYT/ XX(6),XY(6)
      COMMON /XYT/ X(6),Y(6)
      COMMON /SOURCE/ B1(31),TB1(31),XB(1000),TB(1000)
      COMMON /TIME/ BTIME,ETIME,TINTE
      COMMON /PROPS/ R,AL,AT,SIG,VEL,ZLAM,POR,THCK
      COMMON /OTHR/ D,CLAM,CMINW
      COMMON /NAME/ NT(15)
      COMMON /INPU/ MFIRST,NHITS
C
C
C     ***READ THE INPUT DATA
      READ (5,'(A40)') IFILE
      READ (5,'(A40)') MFILE
      READ (5,'(A40)') OFILE(1)
      READ (5,'(A40)') OFILE(2)
      READ (5,'(A40)') OFILE(3)
      READ (5,9000) KKSTOP,IFACE
C
C
C
      IF (ISD.EQ.1.OR.ISD.EQ.0) THEN
C        ***NON-STANDARD OPENING OF FILES
        OPEN (42,FILE='T42.TMP',STATUS='UNKNOWN')
        OPEN (43,FILE='T43.TMP',STATUS='UNKNOWN')
        OPEN (44,FILE='T44.TMP',STATUS='UNKNOWN')
        REWIND 42
        REWIND 43
        FNAME(1) = OFILE(2)
        FNAME(2) = OFILE(3)
      END IF
C
```

```
C
C
      NOF = 3
C      ***DISPLAY TO THE SCREEN A MENU OF CHOICES
      CALL DPF (IFILE,MFILE,OFILE,NOF,IFACE,IEND,IPRE,IINP)
      IF (IEND.EQ.1) RETURN
      IF (IINP.EQ.1) RETURN
C
C
      CALL MESSAG (2)
      READ (5,9100) (NT(I),I=1,15)
      WRITE (42,9100) (NT(I),I=1,15)
      WRITE (43,9100) (NT(I),I=1,15)
      WRITE (44,9100) (NT(I),I=1,15)
C
C
      WRITE(42,4)
      WRITE (42,1) IFILE,MFILE,OFILE(1),OFILE(2),OFILE(3)
      WRITE (42,5) KKSTOP,IFACE
      IF (KKSTOP.NE.4) THEN
        WRITE (42,14)
        STOP
      END IF
C
C
      READ (5,*) AL,AT,AV,VEL,POR,TAQU
      WRITE(42,6) AL,AT,AV,VEL,POR,TAQU
C
C
      READ (5,*) R,PMAX,CMINW,ZLAM
      WRITE (42,7) R,PMAX,CMINW,ZLAM
C
C
C      ***CONVERT CMINW FROM MG/L TO KG/M3
      CMINW = CMINW/1000.
C
C
      READ (5,*) BTIME,ETIME,TINTE
      WRITE (42,8) BTIME,ETIME,TINTE
C
C
      READ (5,*) NWELL
      WRITE (42,9) NWELL
C
C
      DO 100 I=1,NWELL
        READ (5,*) XX(I),XY(I)
        WRITE (42,10) XX(I),XY(I)
C        ***ZERO THE NUMBER OF OUTPUT CONCENTRATIONS
        NUMT(I) = 0
 100    CONTINUE
C
C      ***RECHARGE RATE
      READ (5,*) XINF
      WRITE (42,13) XINF
C
C
C
C
C      ***READ THE MASS FLUX INPUT DATA
      RMAX = 0.
      FMAX = 0.
      TMAX = 0.
      II = 0
       NHITS = 0
 150    CONTINUE
        II = II + 1
        IF (II.GT.999) STOP 'TOO MANY INPUT MASS FLUXES'
        READ (5,*,ERR=200,END=200) TI,RC,HF,CF
        XB(II) = CF
        TB(II) = TI
C         WRITE (42,10) TB(II),XB(II)
        IF (RC.GT.RMAX) THEN
          RMAX=RC
          TRMAX=TI
        END IF
        IF (CF.GT.FMAX) THEN
          FMAX=CF
          TMAX=TI
          RRMX=RC
           NHITS = 1
         ELSE IF (CF.EQ.FMAX) THEN
           NHITS = NHITS + 1
        END IF
        GO TO 150
 200    CONTINUE
C
       MFIRST = 0
C      ***DETERMINE IF THE MAXIMUM MASS FLUX IS FIRST
       IF (FMAX.EQ.XB(1)) THEN
         MFIRST = 1
       END IF
C
C
C
C
      IF (II.LE.1) THEN
C        ***END THE SIMULATION IF NO INPUT FLUX DISTRIBUTION
        WRITE (44,9110)
        ENDCHR = 'END  '
        ENDCHR(4:4) = CHAR(13)
        ENDCHR(5:5) = CHAR(10)
        WRITE (44,9120) ENDCHR
        CALL PCON (FNAME(1),FNAME(2))
```

```
          CALL MESSAG (13)
          CALL MESSAG (4)
          STOP
        END IF
        NUMB = II
        XB(NUMB) = 0.
        TB(NUMB) = TB(NUMB-1) + 1.
C
        IF (RMAX.EQ.0..OR.FMAX.EQ.0..OR.TMAX.EQ.0.) THEN
          WRITE (42,15)
          STOP
        END IF
C
C
C
        CALL MESSAG (6)
C       ***EXTRACT SMALLER NUMBER OF INPUT MASS FLUXES FROM THE
C       ***DISTRIBUTION JUST READ IN
        IF (NUMB.GT.31) THEN
C         ***NUMBER OF POINTS BEFORE THE PEAK, NPB
          NPB = 5
C         ***NUMBER OF POINTS AFTER THE PEAK, NPA
          NPA = 25
C
          IF (MFIRST.EQ.1) THEN
            NPB = 1
            NPA = 29
          END IF
C
          CALL REDUCE (NPB,NPA,TMAX,FMAX,TB,XB,NUMB)
C
        END IF
C
C
C
C       ***WRITE OUT THE MODIFIED MASS FLUX ARRAY
        WRITE (42,12)
        DO 300 I=1,NUMB
          WRITE (42,23) I,TB(I),XB(I)
 300    CONTINUE
C
C
C
C       ***MODIFY THE TIME INCREMENT IF THE DURATION
C       ***OF MASS FLUX INTO THE AQUIFER IS TOO LONG
        DUR = TB(NUMB)-TB(1)
        IF (TINTE.LT.0.1*DUR) THEN
C         ***SET THE TIME INCREMENT TO 365 DAYS IF THE
C         ***MASS FLUX DURATION IS GREATER THAN TEN YEARS
          WRITE (*,22) TINTE,0.1*DUR
          READ (*,'(A)') ANS
C
C         ***THE USERS REACTION TO THE NEW STEP
          IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
C           ***USE THE SUGGESTED STEP
            TINTE = 0.1*DUR
          ELSE IF (ANS.EQ.'N'.OR.ANS.EQ.'n') THEN
C           ***ASK IF THE OLD STEP SHOULD BE USED
            WRITE (*,24)
            READ (*,'(A)') ANS
            IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
              CONTINUE
            ELSE IF (ANS.EQ.'N'.OR.ANS.EQ.'n') THEN
C             ***IF NO PROMPT FOR  NEW TIME STEP VALUE
              WRITE (*,25)
              READ (*,*) TINTE
            END IF

          END IF
C
C         ***OUTPUT THE STEP THAT IS TO BE USED
          WRITE (42,21) TINTE
C
        END IF
        WRITE (42,17) RMAX,TRMAX,RRMX,TMAX
C
C
C
C
C       ***MODIFY MAXIMUM RADIUS AND PERCENT MAX RADIUS
C       ***IF USER HAS SPECIFIED PMAX .GE. 101.
C       ***THIS OPTION CAUSES THE PROGRAM TO USE THE RADIUS
C       ***OF THE MAXIMUM MASS FLUX INPUT TO THE AQUIFER
        IF (PMAX.GE.101.) THEN
          PMAX = 100.
          RMAX = RRMX
        END IF
C
C
C
C
C       ***SUBTRACT OFF THE RADIUS OF THE LENS FROM THE X
C       ***COORDINATE OF THE RECEPTOR LOCATION
C       ***(KOPT AND OILENS ASSUME THAT THE CENTER OF THE SOURCE IS (0,0)
C       ***TSGPLUME ASSUMES THAT THE DOWNGRADIENT EDGE OF THE LENS IS
C       ***(0,0)
        DO 350 I=1,NWELL
          XX(I) = XX(I) - RMAX
 350    CONTINUE
C
C
C
C
C       ***PRELIMINARY CALCULATIONS
        CALL PRELIM (RMAX,PMAX,AV,TAQU,XINF,VEL,POR,SIG,THCK)
C
C       ***LENS SIZE FOR MAXIMUM FLUX
        ERAD = PMAX*RMAX/100.
        EARE = ERAD*ERAD*3.14159
        WRITE (42,18) ERAD,EARE,THCK
```

```
C
C       ***ESTIMATED MAXIMUM CONCENTRATION BASED ON MIXING VOLUME
C       ***UNDER THE LENSE
        VOL = EARE*THCK
        CMX = 1000.*FMAX/VEL/ERAD/2./THCK
C       ***FLUX BALANCE
        FLENS = FMAX/EARE
        FSTRI = FMAX/ERAD/2./THCK
C       WRITE (42,19) FMAX,VOL,VEL,CMX,FLENS,FSTRI
C
C
C
C       ***SORT THE WELL LOCATIONS SO THAT THEY ARE INCREASING IN
C       ***DISTANCE FROM THE ORIGIN
        CALL LOSORT (XX,XY,NWELL,1)
C
C
C
C
C       ***ELIMINATE ANY RECEPTOR POINT THAT IS WITHIN THE LENS
        CALL REELIM (XX,XY,NWELL,RMAX)
C
C
C
C
C       NONDIMENSIONALIZE THE DATA
C
        CALL NONDIM
C
        RETURN
 1      FORMAT (//3X,'HSSM-KO INPUT DATA FILE  ',A40/
     *          3X,'HSSM-KO OUTPUT FILE        ',A40/
     *          3X,'HSSM-T INPUT FILE          ',A40/
     *          3X,'HSSM-T OUTPUT FILE         ',A40/
     *          3X,'HSSM-T PLOT FILE           ',A40)
 4      FORMAT(//1X,'TSGPLUME'//
     *           3X,'INPUT DATA:'/
     *           3X,'==========='/)
 5      FORMAT (3X,'HSSM ENDING PARAMETER, KKSTOP ',I10/
     *          3X,'INTERFACE FLAG                ',A1)
 6      FORMAT(/,3X,'LONG. DISPERSIVITY   =',G10.4,' (M)'/,3X,
     1            'TRANS. DISPERSIVITY  =',G10.4,' (M)'/,3X,
     2            'VERT. DISPERSIVITY   =',G10.4,' (M)'/3X,
     3            'SEEPAGE VELOCITY     =',G10.4,' (M/D)'/,3X,
     4            'POROSITY             =',G10.4,' (*)'/3X,
     7            'AQUIFER THICKNESS    =',G10.4,' (M)'/)
 7      FORMAT (/3X,'RETARDATION FACTOR   =',G10.4,' (*)'/3X,
     *            'P.C. MAX RADIUS      =',G10.4,' (*)'/3X,
     *            'MIN. AQUIFER CONC.   =',G10.4,' (MG/L)'/3X,
     *            'DECAY COEFFICIENT    =',G10.4,' (1/D)'/)
 8      FORMAT (/3X,'BEGINNING TIME       =',G10.4,' (D)'/3X,
     *            'ENDING TIME          =',G10.4,' (D)'/3X,
     *            'TIME INCREMENT       =',G10.4,' (D)'/)
 9      FORMAT (/3X,'NO. OBS. WELLS       =',I8,'   (*)'//3X,
     *          'X-LOCATION',1X,'Y-LOCATION'/3X,
     *          '----------',1X,'----------')
10      FORMAT (3x,G10.4,1X,G10.4)
11      FORMAT (//3X,'INPUT MASS FLUX HISTORY'//3X,
     *            '    TIME  ',1X,'MASS FLUX'/3X,
     *            '    (D)  ',1X,' (KG/D)  '/3X,
     *            '---------',1X,'---------')
12      FORMAT (///4X,'REDUCED INPUT MASS FLUX '/
     *          4X,'HISTORY USED FOR COMPUTATION '/
     *          4X,'============================'//4X,
     *            '   TIME  ',1X,'MASS FLUX'/4X,
     *            '   (D)   ',1X,' (KG/D)  '/4X,
     *            '---------',1X,'---------')
13      FORMAT (/3X,'RECHARGE RATE       =',F8.2,' (M/D)')
14      FORMAT (3X,'TO RUN TSGPLUME, HSSM MUST BE RUN WITH KKSTOP = 4')
15      FORMAT (3X,'INVALID KOPT/OILENS MASS FLUX DISTRIBUTION'/
     *          3X,'LENS RADIUS, MASS FLUX, AND/OR TIME TO PEAK'
     *          3X,'MASS FLUX ARE ZERO')
16      FORMAT (3X,'ADVECTIVE TRAVEL TIME TO RECEPTOR POINT ',G10.4)
17      FORMAT (//3X,'MAXIMUM RADIUS           =',g10.4,' (M)'/
     *          3X,'MAX. RADIUS TIME          =',g10.4,' (D)'/
     *          3X,'RADIUS AT MAX. FLUX       =',G10.4,' (D)'/
     *          3X,'MAX. FLUX TIME            =',g10.4,' (D)')
18      FORMAT (3X,'EFFECTIVE RADIUS          =',g10.4,' (M)'/
     *          3X,'EFFECTIVE AREA            =',g10.4,' (M^2)'/
     *          3X,'PENETRATION THICKNESS     =',g10.4,' (M)'/)
19      FORMAT (3X,'MAX. FLUX                 =',g10.4,' (KG/D)'/
     *          3X,'MIXING VOLUME             =',g10.4,' (M^3)'/
     *          3X,'SEEPAGE VELOCITY          =',g10.4,' (M/D)'/
     *          3X,'REPRESENATIVE MAX. CONC.  =',g10.4,' (MG/L)'/
     *          3X,'LENS FLUX PER AREA        =',G10.4/
     *          3X,'STRIP FLUX PER AREA       =',G10.4/
     *          3X,'EFFECTIVE CONCENTRATION   =',G10.4/)
20      FORMAT(7X,F8.2,5X,F9.4)
21      FORMAT (//
     *         3X,'TIME STEP TOO SMALL RELATIVE TO MASS FLUX DURATION'/
     *         3X,'MODIFIED TIME STEP       =',G10.4,' (D)'//)
22      FORMAT (10X,
     *          '*** TSGPLUME RECOMMENDS CHANGING THE TIME INCREMENT'/
     *          10X,'*** FROM ',G10.4,' DAYS TO ',G10.4,' DAYS'/
     *          10X,'*** ACCEPT THE CHANGE ? (Y OR N)')
23      FORMAT (1x,i2,1x,g10.4,1x,g10.4)
24      FORMAT (10X,'*** DO YOU WISH TO USE THE ORIGINAL TIME ',
     *            'INCREMENT ?  (Y OR N)')
25      FORMAT (10X,'*** ENTER A NEW TIME INCREMENT IN DAYS ')
9000    FORMAT (1X,I10,1X,A1)
9100    FORMAT (5A10/5A10/5A10)
9110    FORMAT ('BEGIN')
9120    FORMAT (A5)
        END
```

```
      SUBROUTINE REELIM (XX,YY,NWELL,RMAX)
C     ************************************************************
C     *
C     * REELIM
C     *
C     * REELIM ELIMINATES ANY RECEPTOR POINT THAT IS WITHIN
C     * THE EFFECTIVE RADIUS OF THE OIL LENS.
C     *
C     * INPUT ARGUMENTS
C     * XX()     ARRAY OF X LOCATIONS OF THE RECEPTORS
C     * YY()     ARRAY OF Y LOCATIONS OF THE RECEPTORS
C     * NWELL    NUMBER OF RECEPTOR WELLS
C     * RMAX     EFFECTIVE LENS RADIUS FOR THE SIMULATION
C     *
C     * OUTPUT ARGUMENTS
C     * XX()     MODIFIED ARRAY OF X LOCATIONS OF THE RECEPTORS
C     * YY()     MODIFIED ARRAY OF Y LOCATIONS OF THE RECEPTORS
C     *
C     * JIM WEAVER
C     * US EPA/ORD/OEPER/RSKERL
C     * ADA, OKLAHOMA
C     * 12-22-92
C     * ANSI STD. X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES: NONE
C     * CALLED BY  INFO
C     *
C     ************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      DIMENSION XX(*),YY(*)
C
C
C
      I = 0
 10   CONTINUE
        I = I + 1
C
C       ***DISTANCE FROM CENTER OF SOURCE (0,0) TO RECEPTOR POINT
        IF (XX(I).LE.0.) THEN
C
          CALL MESS4 (17,XX(I)+RMAX,YY(I))
C         ***ELIMINATE THIS POINT BECAUSE IT IS BEHIND THE
C         ***ASSUMED POSITION OF THE AQUIFER SOURCE
          DO 5 J=I,NWELL-1
            XX(J) = XX(J+1)
            YY(J) = YY(J+1)
 5        CONTINUE
          NWELL = NWELL - 1
          I = I - 1
C
        END IF
C
      IF (I.LT.NWELL-1) GO TO 10
C
C
C
      RETURN
      END
```

```
      SUBROUTINE LOSORT (XX,YY,NWELL,IDIST)
C     ************************************************************
C     *
C     * LOSORT
C     *
C     * LOSORT SORTS THE RECEPTOR WELL LOCATIONS SO THAT THEY ARE
C     * IN ORDER OF INCREASING X.   SORTING IS REQUIRED FOR THE
C     * CALCULATION OF THE MINIMIMUN CONCENTRATION AT THE RECEPTOR
C     * PERFORMED IN SR. ADVECT.
C     *
C     * INPUT ARGUMENTS
C     * XX()     ARRAY OF X LOCATIONS OF THE RECEPTORS
C     * YY()     ARRAY OF Y LOCATIONS OF THE RECEPTORS
C     * NWELL    NUMBER OF RECEPTOR WELLS
C     * IDIST    SORT FLAG
C     *          0 RETURN IN INCREASING ORDER OF XX
C     *          1 RETURN IN INCREASING ORDER OF DISTANCE
C     *
C     * OUTPUT ARGUMENTS
C     * XX()     SORTED ARRAY OF X LOCATIONS OF THE RECEPTORS
C     * YY()     SORTED ARRAY OF Y LOCATIONS OF THE RECEPTORS
C     *
C     * JIM WEAVER
C     * US EPA/ORD/OEPER/RSKERL
C     * ADA, OKLAHOMA
C     * 12-18-92
C     * ANSI STD. X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES: NONE
C     * CALLED BY  ADVECT,INFO
C     *
C     ************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      DIMENSION XX(*),YY(*)
C
C
      IF (NWELL.EQ.1) RETURN
C
C
C     ***SORT THE RECEPTOR WELL LOCATIONS
      NLAST = NWELL - 1
      DO 15 J=1,NWELL
        IH = 0
        DO 14 I=1,NLAST
          DS1 = XX(I)
          DS2 = XX(I+1)
          IF (IDIST.EQ.1) THEN
            DS1 = DS1*XX(I) + YY(I)*YY(I)
            DS2 = DS2*XX(I+1) + YY(I+1)*YY(I+1)
          END IF
          IF (DS1.GT.DS2) THEN
            IH = 1
            PTEM = XX(I)
            XX(I) = XX(I+1)
            XX(I+1) = PTEM
            PTEM = YY(I)
            YY(I) = YY(I+1)
            YY(I+1) = PTEM
          END IF
   14   CONTINUE
        IF (IH.EQ.0) GO TO 16
        NLAST = NLAST - 1
   15 CONTINUE
   16 CONTINUE
C
C
C
      RETURN
      END
```

```
      SUBROUTINE ADVECT (IWELL,EEPS,TMAX,RMAX,NCONC,ICOMP)
C     ************************************************************
C     *   ADVECT                                                 *
C     *                                                          *
C     *  ADVECT DETERMINES THE ADVECTIVE TRAVEL TIME TO A RECEPTOR POINT.
C     *  DUE TO DISPERSION THE ACTUAL ARRIVAL TIME OF THE PEAK   *
C     *  CONCENTRATION WILL BE LATER THAN THE VALUE CALCULATED BY ADVECT
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................  *
C     *   IWELL    INDEX OF THE RECEPTOR POINT LOCATION          *
C     *   EEPS     MACHINE FLOATING POINT [DOUBLE] PRECISION     *
C     *   TMAX     TIME OF MAXIMUM MASS FLUX INTO THE AQUIFER    *
C     *   RMAX     EFFECTIVE LENS RADIUS FOR THE SIMULATION      *
C     *                                                          *
C     *   OUTPUT ARGUMENTS.....................................  *
C     *   NCONC    NOCONCENTRATION FLAG                          *
C     *            0  NO CONCENTRATION ABOVE THE THRESHOLD (cminw)
C     *            1  CONCENTRATION FOUND ABOVE THRESHOLD        *
C     *   ICOMP    COMPUTED CONCENTRATION FLAG                   *
C     *            0  NO CONCENTRATIONS COMPUTED                 *
C     *            1  AT LEAST ONE CONCENTRATION COMPUTED        *
C     *                                                          *
C     *   JIM WEAVER                                             *
C     *   R.S. KERR ENVIRONMENTAL RESEARCH LABORATORY            *
C     *   U.S. ENVIRONMENTAL PROTECTION AGENCY                   *
C     *   ADA, OKLAHOMA   74820                                  *
C     *                                                          *
C     *   DOUBLE PRECISION FORTRAN 77                            *
C     *   NOVEMBER 12, 1992                                      *
C     *   REQUIRED ROUTINES:  LOSORT,MBISEC,MESSAG,NEWOUT        *
C     *   CALLED BY   HSSMT                                      *
C     *                                                          *
C     ************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      CHARACTER NT*15
      CHARACTER*40 IFILE,OFILE(3),MFILE
      DIMENSION TM(3)
      EXTERNAL RFUNC
      COMMON /AUTO/ ADVT,DADVT
      COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
      COMMON /DIMXYT/ XX(6),XY(6)
      COMMON /XYT/ X(6),Y(6)
      COMMON /SAVE/ VSAVE(100),ZSAVE(100)
      COMMON /SOURCE/ B1(31),TB1(31),XB(1000),TB(1000)
      COMMON /TIME/ BTIME,ETIME,TINTE
      COMMON /PROPS/ R,AL,AT,SIG,VEL,ZLAM,POR,THCK
      COMMON /OTHR/ D,CLAM,CMINW
      COMMON /NAME/ NT(15)
C
C
C
C     ***DETERMINE THE ADVECTIVE TRAVEL TIME FOR AUTOMATION OF
C     ***TSGPLUME, IF ONE RECEPTOR WELL IS SPECIFIED
C     ***THE MINIMUM TRAVEL TIME FROM THE  FIRST TIME THAT MASS ENTERS
C     ***THE AQUIFER (TB(1)) TO THE RECEPTOR POINT AT XX(IWELL)
C     ***THE MASS ENTERS AT THE LEADING EDGE OF THE LENS SO THAT
C     ***THE DISTANCE IS REDUCED BY THE LENS RADIUS (RMAX)
C     ADVT = TMAX + (XX(IWELL)-RMAX)*R/VEL
C     ***REDUCTION ALREADY DONE IN TINE INFO
      ADVT = TB(1) + XX(IWELL)*R/VEL
C
C
C
C     ***DETERMINE THE EARLIEST TIME THAT THE CONCENTRATION IS ABOVE
C     ***THE THRESHOLD (CMINW)
      IIC = 0
      NCONC = 0
      CALL MESSAG (9)
      TM(1) = X(IWELL)
      TM(2) = Y(IWELL)
      PI = 4.*ATAN(1.0)
      TM(3) = SQRT(4.*PI)
C     ***SINCE THE RECEPTOR WELL LOCATIONS ARE SORTED W.R.T. THEIR
C     ***X LOCATION EACH SUCESSIVE IS FURTHER AWAY FROM THE SOURCE
C     ***THEREFORE THE MINIMUM TIME MUST BE GREATER THAN THAT FOR
C     ***THE PREVIOUS WELL.   THE TIME FOR THE PREVIOUS WELL (STIME)
C     ***IS A LOWER BOUND FOR THE NEW SOLUTION
      TM1 = TB(1)
C     IF (IWELL.GT.1) TM1 = 0.98*STIME
      CONV  = 0.0001
      NVS = 0
      NCONC = 0
C
C
C     ***ASSURE THAT THE TIME INTERVAL IS SMALLER THAN THE
C     ***TIME BETWEEN THE ADVECTIVE TRAVEL TIME AND THE BEGINNING TIME
      DT = 0.2*(ADVT-TM1)
      DO 100 TIME = TM1,ADVT,DT
        TND = VEL*TIME/R/AL
        CALL RFUNC (TND,TM,CONC)
        NVS = NVS + 1
        ZSAVE(NVS) = TND
        VSAVE(NVS) = CONC
        IF (CONC.GT.CMINW.AND.IIC.EQ.0) THEN
C          ***FIRST CONCENTRATION ABOVE THE THRESHOLD CMINW
          T1 = TIME-DT
          IF (T1.LT.TB(1)) T1 = TB(1)
          T1 = VEL*T1/R/AL
          T2 = VEL*TIME/R/AL
          CALL MBISEC (RFUNC,T1,T2,CONV,EEPS,100,CMINW,TM,BTIME,CONC,
     *    IE,NVS,VSAVE,ZSAVE)
          IIC = 1
          IF (IE.EQ.0) THEN
            NCONC = 1
```

```
            STIME = BTIME*R*AL/VEL
          END IF
        END IF
 100  CONTINUE
C
C
C
C     ***NO CONCENTRATION FOUND: RETURN WITH FLAG SET TO 0
C     ***THE SOLUTION WILL NOT BE CALCULATED FOR THIS LOCATION
      IF (NCONC.EQ.0) THEN
        CALL MESSAG (14)
        RETURN
      END IF
      ICOMP = 1
C
C
C
C     ***SORT THE SOLUTION TIMES
      CALL LOSORT (ZSAVE,VSAVE,NVS,0)
C
C
C
C     ***ELIMINATE ANY CONCENTRATIONS BELOW CMINW
      NN = 0
 110  CONTINUE
        NN = NN + 1
        IF (VSAVE(NN).LT.CMINW) THEN
          DO  120 J=NN,NVS-1
            ZSAVE(J) = ZSAVE(J+1)
            VSAVE(J) = VSAVE(J+1)
 120       CONTINUE
          NVS = NVS - 1
          NN = NN - 1
        END IF
      IF (NN.LT.NVS) GO TO 110
C
C     ***ELIMINATE ANY DUPLICATE ENTRIES
      NN = 0
 130  CONTINUE
        NN = NN + 1
        IF (ABS(ZSAVE(NN)-ZSAVE(NN+1)).LT.0.001) THEN
          DO 140 J=NN,NVS-1
            ZSAVE(J+1) = ZSAVE(J+2)
            VSAVE(J+1) = VSAVE(J+2)
 140       CONTINUE
          NVS = NVS - 1
          NN = NN -1
        END IF
      IF (NN.LT.NVS-2) GO TO 130
C
C
C
C     ***SET NVS TO 1 TO WRITE OUT ONLY THE FIRST CONC. ABOVE CMINW
      NVS = 1
C
C     ***OUTPUT RESULTS
      DO 175 I=1,NVS
        ZSAVE(I) = ZSAVE(I)*R*AL/VEL
 175  CONTINUE
C
      CALL NEWOUT (1,ZSAVE(1),VSAVE(1),IWELL,RMAX)
      DO 200 I=2,NVS
        CALL NEWOUT (0,ZSAVE(I),VSAVE(I),IWELL,RMAX)
 200  CONTINUE
      NUMT(IWELL) = NVS
C
      RETURN
      END
```

```
      SUBROUTINE REDUCE (NPB,NPA,TMAX,FMAX,T,F,NUMB)
C     ************************************************************
C     *
C     *  EXTRACT A SMALLER NUMBER OF MASS FLUXES FROM THE INPUT
C     *          DISTRIBUTION
C     *
C     * INPUT ARGUMENTS.....................................
C     *   NPB   = NUMBER OF POINTS BEFORE THE PEAK
C     *   NPA   = NUMBER OF POINTS AFTER THE PEAK
C     *   TMAX  = TIME AT WHICH MAXIMUM MASS FLUX OCCURS
C     *   FMAX  = MAXIMUM MASS FLUX
C     *   T()   = TIMES
C     *   F()   = MASS FLUX
C     *   NUMB  = NUMBER OF MASS FLUXES AND TIMES
C     * OUTPUT ARGUMENTS....................................
C     *   T()   = MODIFIED TIMES
C     *   F()   = MODIFIED MASS FLUX
C     *   NUMB  = NUMBER OF MASS FLUXES AND TIMES
C     *
C     *  JIM WEAVER, USEPA
C     *  DOUBLE PRECISION FORTRAN 77
C     *  REQUIRED ROUTINES:  BINARY
C     *  CALLED BY  INFO,PREPLOT
C     *
C     ************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      DIMENSION T(*),F(*)
      DIMENSION TEMPT(76),TEMPF(76)
C
C
      IF (NPA+NPB.GT.75) STOP 'INCORRECT USE OF SUBROUTINE REDUCE'
C     ***PLACE NPB POINTS IN BETWEEN THE INITAL FLUX AND THE PEAK
      TEMPT(1) = T(1)
      TEMPF(1) = F(1)
      TEMPT(NPB+1) = TMAX
      TEMPF(NPB+1) = FMAX
      DT = (TMAX-T(1))/FLOAT(NPB)
      TIME = T(1) + DT
      DO 10 I=2,NPB
        CALL BINARY (TIME,T,NUMB,1,IN)
        SL = (F(IN+1)-F(IN))/(T(IN+1)-T(IN))
        FF = F(IN) + SL*(TIME-T(IN))
        TEMPT(I) = TIME
        TEMPF(I) = FF
        TIME = TIME + DT
 10   CONTINUE
C
C
C
C     ***PLACE NPA POINTS BETWEEN THE PEAK FLUX AND THE END FLUX
      TEMPT(NPB+NPA+1) = T(NUMB)
      TEMPF(NPB+NPA+1) = F(NUMB)
      DT = (T(NUMB)-TMAX)/FLOAT(NPA)
      TIME = TMAX+DT
      DO 20 I=NPB+2,NPB+NPA
        CALL BINARY (TIME,T,NUMB,1,IN)
        SL = (F(IN+1)-F(IN))/(T(IN+1)-T(IN))
        FF = F(IN) + SL*(TIME-T(IN))
        TEMPT(I) = TIME
        TEMPF(I) = FF
        TIME = TIME + DT
 20   CONTINUE
C
C
C
C     ***PUT DATA INTO THE ARRAYS TO BE RETURNED
      DO 30 I=1,NPB+NPA+1
        T(I) = TEMPT(I)
        F(I) = TEMPF(I)
 30   CONTINUE
      NUMB = NPA+NPB+1
C
C
C
C     ***CHECK FOR ANOMOLOUS INPUT MASS FLUX
C     ***1.  THE MAXIMUM MASS FLUX IS THE FIRST INPUT POINT
      IX = 0
      DO 50 I=1,NUMB
        IF (T(I).EQ.T(I+1)) THEN
          IX = IX + 1
        END IF
 50   CONTINUE
C
C
      DO 60 I=1,NUMB-IX
        T(I) = T(I+IX)
        F(I) = F(I+IX)
 60   CONTINUE
      NUMB = NUMB - IX
C
C
      RETURN
      END
```

```
      SUBROUTINE PREPLOT (ICOMP,NUMT,CMAX,TCMAX,TB,XB,NWELL,NC,
     *  NUMTT)
C     ************************************************************
C     *
C     *  PREPLOT PREPARES FILE 43 FOR PLOTTING BY HSSM-WIN
C     *  THE NUMBER OF POINTS IS REDUCED
C     *
C     * INPUT ARGUMENTS.............................................
C     *    ICOMP  = COMPUTATION FLAG = 0 IF NO CONCENTRATIONS COMPUTED
C     *    NUMT() = NUMBER OF TIME-CONCENTRATION PAIRS
C     *    CMAX() = MAXIMUM CONCENTRATION
C     *    TCMAX()= TIME TO PEAK OF MAX CONCENTRATION
C     *    TB()   = ARRAY TO STORE TIMES
C     *    XB()   = ARRAY TO STORE CONCENTRATIONS
C     *    NWELL  = NUMBER OF RECEPTOR LOCATIONS WITH CONCENTRATION
C     *             DISTRIBUTIONS
C     *    NC()   = ARRAY OF WELL FLAGS
C     *             NC(I) = 1 IF CONCENTRATION DISTRIBUTION CALCULATED
C     *             NC(I) = 0 IF NO CONCENTRATION DISTRIBUTION
C     * OUTPUT ARGUMENTS............................................
C     *    TB() = REDUCED TIMES
C     *    XB() = REDUCED CONCENTRATIONS
C     *    NUMTT = NUMBER OF POINTS (51)
C     *
C     *  JIM WEAVER   7-27-92
C     *  REQUIRED ROUTINES:   REDUCE
C     *  CALLED BY  HSSMT
C     *
C     ************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      DIMENSION CMAX(*),NC(*),NUMT(*),TB(*),TCMAX(*),XB(*)
      CHARACTER*50 CHAR5
      REWIND 43
      REWIND 44
C
      DO 10 I=1,5
        READ (43,'(A50)') CHAR5
        WRITE (44,9090) CHAR5
 10    CONTINUE
C
      DO 50 II=1,NWELL
        IF (NC(II).EQ.0) GO TO 50
C
        YMAX = 0.0
        DO 20 I=1,NUMT(II)
          READ (43,*,ERR=20,END=20) TB(I),XB(I)
 20      CONTINUE
C
        IF (NUMT(II).EQ.0) GO TO 50
        NUMTT = NUMT(II)
        IF (NUMT(II).GE.71) THEN
          CALL MESS5 (11,II)
          CALL REDUCE (24,45,TCMAX(II),CMAX(II)*1000.,TB,XB,NUMTT)
        END IF
C
        DO 40 I=1,NUMTT
          WRITE (44,9100) TB(I),XB(I)
 40      CONTINUE
C
        DO 45 I=1,2
          READ (43,'(A50)') CHAR5
          WRITE (44,9090) CHAR5
 45      CONTINUE
 50    CONTINUE
C
      RETURN
 9090  FORMAT(A50)
 9100  FORMAT(1X,F10.2,3X,6(G10.4,1X))
       END
```

```
      SUBROUTINE PRELIM (RMAX,PMAX,AV,TAQU,XI,VEL,POR,SIG,THCK)
C     ****************************************************************
C     *
C     *  PRELIM  PERFORMS PRELIMINARY CALCULATIONS
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *   RMAX    EFFECTIVE SOURCE RADIUS FOR SIMULATION
C     *   PMAX    PERCENT MAXIMUM RADIUS FOR CALCULATION
C     *   AV      VERTICAL DISPERSIVITY
C     *   TAQU    AQUIFER THICKNESS
C     *   XI      RECHARGE RATE
C     *   VEL     GROUNDWATER SEEPAGE VELOCITY
C     *   POR     POROSITY
C     * OUTPUT ARGUMENTS............................................
C     *   SIG     STANDARD DEVIATION FOR GAUSSIAN DISTRIBUTION OF
C     *            SOURCE
C     *   THCK    PENETRATION THICKNESS
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * DOUBLE PRECISION FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  INFO
C     *
C     ****************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
C
C      ***PRELIMINARY CALCULATIONS
      XL = 2.*RMAX*PMAX/100.
      SIG = 0.25*XL
C
      VD = VEL*POR
      THCK = SQRT(2.*AV*XL) + TAQU*(1.-EXP(-XI*XL/TAQU/VD))
      IF (THCK.GT.TAQU) THCK = TAQU
C
      RETURN
      END
```

```
      SUBROUTINE NONDIM
C     ****************************************************************
C     ****************************************************************
C     **
C     **  P U R P O S E
C     **
C     **   THIS ROUTINE EXPRESSES THE INPUT PARAMETERS IN NONDIMENSIONAL
C     **   FORM.
C     **
C     **  I N P U T / O U T P U T   P A R A M E T E R S
C     **
C     **       (ALL INPUT/OUTPUT DATA ARE THE SAME AS IN SUBROUTINE INFO)
C     **
C     **   AUTHOR: MIKE JOHNSON
C     **           U.T. AUSTIN
C     **
C     **   LANGUAGE: FORTRAN 77
C     **
C     **   REQUIRED ROUTINES: (NONE)
C     **
C     ****************************************************************
C     ****************************************************************
        IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
        COMMON /AUTO/ ADVT,DADVT
        COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
        COMMON /DIMXYT/ XX(6),XY(6)
        COMMON /XYT/ X(6),Y(6)
        COMMON /SOURCE/ B1(31),TB1(31),XB(1000),TB(1000)
        COMMON /PROPS/ R,AL,AT,SIG,VEL,ZLAM,POR,THCK
        COMMON /OTHR/ D,CLAM,CMINW
C
        PI=4.*ATAN(1.0)
C
C
C     CALCULATE DIMENSIONLESS DISPERSION COEFFICIENT
C
        D = AL * AT / SIG**2.
C
C     CALCULATE DIMENSIONLESS DECAY-RATE COEFFICIENT
C
        CLAM = R * ZLAM * AL / VEL
C
C     CALCULATE DIMENSIONLESS X's
C
        DO 1 I=1,NWELL
 1        X(I) = XX(I) / AL
C
C     CALCULATE DIMENSIONLESS Y's
C
        DO 2 J=1,NWELL
 2        Y(J) = XY(J) / SIG
C
```

```
C    CALCULATE DIMENSIONLESS TIMES
C       ***7-26-92 THE NONDIMENSIONLESS TIME ARRAYS HAVE BEEN REMOVED
C       ***TIMES AND NONDIMENSIONAL TIMES ARE CACULATED AS NEEDED
C       DO 3 L=1,NUMT
C3        T(L) = VEL * XT(L) / ( R * AL )
C
C
C    CALCULATE DIMENSIONLESS TIMES FOR INFLOW CONCENTRATION PROFILE
C
          DO 6 I=1,NUMB
 6        TB1(I) = VEL * TB(I) / ( R * AL )
C
C    CALCULATE DIMENSIONLESS MASS FLUX
          DO 8 I=1,NUMB
 8        B1(I) = XB(I) / ( SQRT( PI / 2. ) * THCK * POR * VEL * SIG *
     +                  ( 1. + SQRT(1. + 4. * CLAM ) ) )
C
          RETURN
          END
```

```
      SUBROUTINE SOLN (ICOMP,CMAX,TCMAX,IWELL,RMAX,ISET)
C   ******************************************************************
C   ******************************************************************
C   **
C   **  P U R P O S E
C   **
C   **   THIS ROUTINE COMPUTES THE CONCENTRATION PROFILES AT EACH
C   **   LOCATION AND TIME SPECIFIED IN THE INPUT.
C   **
C   **
C   **  I N P U T    P A R A M E T E R S
C   **
C   **   (ALL NONDIMENSIONALIZED PARAMETERS ARE USED IN THIS ROUTINE)
C   **       IWELL   RECEPTOR WELL INDEX
C   **       RMAX    EFFECTIVE RADIUS FOR SIMULATION
C   **       ISET    FLAG SET TO ZERO UPON EACH CALL TO SOLN
C   **
C   **
C   **  O U T P U T    P A R A M E T E R S
C   **
C   **       C .... REAL ARRAY CONTAINING THE CONCENTRATION RATIOS
C   **              COMPUTED FOR EACH LOCATION THROUGH TIME.
C   **       ICOMP  = 0 IF NO CONCENTRATIONS COMPUTED
C   **       ICOMP  = 1 IF AT LEAST ONE CONCENTRATION COMPUTED
C   **       CMAX   = MAXIMUM DIMENSIONAL CONCENTRATION COMPUTED
C   **       TCMAX  = DIMENSIONAL TIME TO PEAK CONCENTRATION
C   **       ISET   = FLAG SET TO ONE AFTER RESETTING THE TIME STEP
C   **
C   **   AUTHOR: MIKE JOHNSON
C   **           U.T AUSTIN
C   **   MODIFIED BY JIM WEAVER, USEPA
C   **
C   **   LANGUAGE: DOUBLE PRECISION FORTRAN 77
C   **   REQUIRED ROUTINES: NEWOUT,ROMINT
C   **   CALLED BY  HSSMT
C   **
C   ******************************************************************
C   ******************************************************************
   IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      COMMON /AUTO/ ADVT,DADVT
      COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
      COMMON /DIMXYT/ XX(6),XY(6)
      COMMON /XYT/ X(6),Y(6)
      COMMON /SAVE/ VSAVE(100),ZSAVE(100)
      COMMON /SOURCE/ B1(31),TB1(31),XB(1000),TB(1000)
      COMMON /TIME/ BTIME,ETIME,TINTE
      COMMON /PROPS/ R,AL,AT,SIG,VEL,ZLAM,POR,THCK
      COMMON /OTHR/ D,CLAM,CMINW
C
C
C
C     ***CALCULATE PI
      PI=4.*ATAN(1.0)
      CX1 = SQRT(4.*PI)
C
C
C
C     ***DETERMINE THE CONCENTRATION PROFILE VIA NUMERICAL INTEGRATION
      CMAX = 0.0
      ITIME = 0
      ICFAC = 0
      TFACT = 1.15
      ODT = TINTE
C
C
C
C     ***SET PREVIOUS TWO TIMES (TPREV1 AND TPREV2)
C     ***GET ASSOCIATED CONCENTRATIONS (CPREV1 AND CPREV2)
C     ***FIRST PREVIOUS TIME
      TPREV1 = ZSAVE(1)
      CPREV1 = VSAVE(1)
C
C
C
C     ***SECOND PREVIOUS TIME
      TPREV2 = TPREV1 + TINTE
      TND = TPREV2*VEL/R/AL
      CALL ROMINT (TND,X(IWELL),Y(IWELL),XI)
      CPREV2 = XI*X(IWELL)/CX1
C
C
C
C     ***SET NEW TIME STEP AND GET NEW CONCENTRATION
      TIME = TPREV2 + TINTE
      TND = TIME*VEL/R/AL
C     ***FIND THE NEW CONCENTRATION
      CALL ROMINT (TND,X(IWELL),Y(IWELL),XI)
      C = XI*X(IWELL)/CX1
      CMAX = C
      TCMAX = TIME
      ICOMP = 1
      NUMT(IWELL) = 1
C
C
C
 10   CONTINUE
C
C
C       ***CHECK FOR ALL CONCENTRATIONS MONOTONICALLY INCREASING OR
C       ***DECREASING
        IF ((CPREV1.LT.CPREV2.AND.CPREV2.LT.C).OR.
     *    (CPREV1.GT.CPREV2.AND.CPREV2.GT.C)) THEN
C
C
```

```
C
C         ***GRADUALLY INCREASE THE TIME STEP AFTER FINDING THE PEAK
          IF (ICFAC.EQ.1) THEN
C
            TINTE = TFACT*TINTE
C
C           ***INFLATE THE TIME INFLATION FACTOR
            IF (ABS(CPREV2-C)/C.GT.0.05) THEN
C             ***DO NOT INCREASE THE INFLATION FACTOR IF THE
C             ***CONCENTRATION
C             ***CHANGES TOO MUCH
              TFACT = 0.95
            ELSE
C             ***OTHERWISE INCREASE THE INFLATION FACTOR
              TFACT = TFACT*1.05
            END IF
C
C           ***LIMIT MAXIMUM TIME STEP
            STEPMAX = ODT
            IF (STEPMAX.LT.365.25) STEPMAX = 365.25
            IF (TINTE.GT.STEPMAX) THEN
              TINTE = STEPMAX
              TFACT = 1.15
            END IF
C
          END IF
C
C
C
C         ***RESET TIME STEP (TINTE) TO 1/10 OF THE ORIGINAL VALUE
          IF (C.LT.0.98*CMAX.AND.TINTE.LT.ODT.AND.ICFAC.EQ.0) THEN
            TINTE = .1*ODT
            ICFAC = 1
          END IF
C
C
C
C         ***OUTPUT RESULTS ONLY IF THERE IS AT LEAST 1% CHANGE IN CONC.
C         ***AND THIS IS NOT THE FIRST STEP (I.E., TPREV1 = ZSAVE(1))
          IF (ABS((CPREV2-C)/C).GT.0.0.AND.TPREV1.NE.ZSAVE(1)) THEN
            CALL NEWOUT (0,TPREV1,CPREV1,IWELL,RMAX)
C           ***RECORD MAXIMUM CONCENTRATION
            IF (C.GT.CMAX) THEN
              CMAX = C
              TCMAX = TIME
            END IF
          END IF
C
C         ***CHECK FOR ENDING (C DROPS BELOW CMINW)
          IF (C.LE.CMINW) THEN
            CALL NEWOUT (0,TIME,C,IWELL,RMAX)
C           ***RECORD MAXIMUM CONCENTRATION
            IF (C.GT.CMAX) THEN
              CMAX = C
              TCMAX = TIME
              END IF
              TINTE = ODT
              RETURN
            END IF
C
          END IF
C
C         ***CHECK FOR CONCENTRATION C BEING LESS THAN CPREV2
C         ***(INDICATES THAT THE PEAK IS INBETWEEN TPREV1 AND TIME)
C
C
          ICHK = 0
C         ***CHECK FOR THE THIRD POINT BEING LOWER THAN THE PREVIOUS TWO
C         ***WHICH INDICATES THE PRESENCE OF A PEAK
          IF (CPREV1.LT.CPREV2.AND.CPREV2.GT.C) ICHK = 1
C         ***STOP CHECKING WHEN CPREV2 IS WITHIN 0.1% OF C
C         ***THIS CHECK CAN OVERRIDE THE PREVIOUS SETTING OF ICHK
          IF (CPREV1.LT.CPREV2.AND.CPREV2.GT.C.AND.
     *  ABS((CPREV2-C)/C).LT.0.0005) ICHK = 0
          IF (ICHK.EQ.1) THEN
C           ***CONDITION FOUND--DECREASE THE TIME STEP
C           ***RESTART CALCULATIONS
            IF (ISET.EQ.0) THEN
              ODT = TINTE
              ISET = 1
            END IF
            TINTE = TINTE/5.
C
C           ***SECOND PREVIOUS EVALUATION MUST BE RESET
            TPREV2 = TPREV1 + TINTE
            TND = TPREV2*VEL/R/AL
            CALL ROMINT (TND,X(IWELL),Y(IWELL),XI)
            CPREV2 = XI*X(IWELL)/CX1
C
C           ***NEW EVALUATION TIME
            TIME = TPREV2 + TINTE
            TND = TIME*VEL/R/AL
            CALL ROMINT (TND,X(IWELL),Y(IWELL),XI)
            C =  XI*X(IWELL)/CX1
C
            GO TO 10
          END IF
C
C
C         ***ROLL THE OLD TIMES AND CONCENTRATIONS
          TPREV1 = TPREV2
          CPREV1 = CPREV2
          TPREV2 = TIME
          CPREV2 = C
C
C         ***SET STEP FOR NEW EVALUATION
          TIME = TIME + TINTE
          TND = TIME*VEL/R/AL
          CALL ROMINT (TND,X(IWELL),Y(IWELL),XI)
          C = XI*X(IWELL)/CX1
```

```
C
        GO TO 10
C
 50   CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE ROMINT(BB,X,Y,XI)
C     ***********************************************************************
C     ***********************************************************************
C     **
C     **  P U R P O S E
C     **
C     **   THIS ROUTINE PEFORMS A NUMERICAL INTEGRATION BASED ON A
C     **   ROMBERG ALGORITHM. THE FUNCTION IS INTEGRATED FOR EACH GIVEN
C     **   TIME (UPPER BOUND OF INTEGRATION) AND LOCATION IN SPACE
C     **
C     **
C     **  I N P U T    P A R A M E T E R S
C     **
C     **       B .... THE UPPER LIMIT OF INTEGRATION (THE TIME PASSED
C     **               IN FROM SUBROUTINE SOLN)
C     **       X .... THE X-COORDINATE
C     **       Y .... THE Y-COORDINATE
C     **      (THE OTHER PARAMETERS PASSED THROUGH COMMON BLOCKS ARE
C     **       THE SAME AS IN THE ABOVE ROUTINES)
C     **
C     **
C     **  O U T P U T    P A R A M E T E R S
C     **
C     **       XI.... THE NUMERICAL SOLUTION TO THE INTEGRAL AT THE GIVEN
C     **              LOCATION AND TIME
C     **
C     **
C     **  O T H E R   P A R A M E T E R S
C     **
C     **       T1.... REAL ARRAY CONTAINING INTEGRAL APPROXIMATIONS
C     **       EPS... REAL VARIABLE. THE RELATIVE ERROR FOR WHICH
C     **              CONVERGENCE IS CHECKED
C     **       A .... REAL VARIABLE. THE LOWER BOUND OF INTEGRATION (=0.)
C     **       BT.... REAL DUMMY VARIABLE. EQUALS THE UPPER BOUND OF
C     **              INTEGRATION
C     **
C     **
C     **   AUTHOR: MIKE JOHNSON
C     **           U.T. AUSTIN
C     **
C     **   LANGUAGE: FORTRAN 77
C     **
C     **   REQUIRED ROUTINES: FN
C     **
C     ***********************************************************************
C     ***********************************************************************
        IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
          COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
          COMMON /SOURCE/ B1(31),TB1(31),XB(1000),TB(1000)
          COMMON /OTHR/ D,CLAM,CMINW
C
          dimension T1(25,25)
```

```
C
C   SET TOLERANCE AND MAXIMUM ITERATIONS
C
C
C   TOLERANCE
C
        EPS=0.00001
C
C   MAXIMUM NUMBER OF ITERATIONS
C
        NUMITS=23
C
C
C     ***SET LIMITS OF INTEGRATION
      B  = BB
      BT = BB
      A=0.
C
C     ***ADJUST LIMITS OF INTEGRATION
      IF (A.LT.TB1(1)) THEN
C       ***SET THE LOWER LIMIT OF INTEGRATION TO THE FIRST
C       ***NONDIMENSIONAL TIME THAT MASS ENTERS THE AQUIFER
        A = TB1(1)
      END IF
      IF (BB.GT.TB1(NUMB)) THEN
C       ***ONLY INTEGRATE WHEN MASS IS ENTERING THE AQUIFER
C       ***ALL CONTRIBUTIONS TO THE INTEGRAL ARE ZERO
C       ***WHEN NO MASS ENTERS THE AQUIFER
C       ***NO MASS ENTERS THE AQUIFER AFTER THE END OF THE MASS FLUX
        B = TB1(NUMB)
      END IF
C
C   CALCULATE THE AVERAGE TIME VALUE BETWEEN ZERO AND
C   UPPER BOUND OF INTEGRATION
C
        ABAV=(B+A)/2.
C
C   CALCULATE THE FIRST SET OF INTEGRAL 'PANELS'
C   AT THE LOWER BOUND, AVERAGE TIME, AND UPPER BOUND
C
        T1(1,1)=(B-A)/2.*(FN(A,X,Y,BT)+FN(B,X,Y,BT))
        T1(1,2)=T1(1,1)/2.+(B-A)/2.*(FN(ABAV,X,Y,BT))
        T1(2,1)=1./3.*(4.*T1(1,2)-T1(1,1))
C
c
C    BEGIN ITERATION TO GET BEST APPROXIMATION FOR THE INTEGRAL
C
        DO 1 J=3,25
C
C    SET THE INTERVAL DISTANCE
C
          DLAM=(B-A)/(2**(J-1))
C
C    CALCULATE ANOTHER 'PANEL' LOCATION
C
          XLAM=A-DLAM
          N=2**(J-2)
C
C    BEGIN INTEGRATION
C
          SUM=0.
          DO 2 I=1,N
C
C   CALCULATE THE STEP SIZE
C
            XLAM=XLAM+2.*DLAM
            SUM=SUM+FN(XLAM,X,Y,BT)
 2        CONTINUE
C
C   CALCULATE NEW PANELS
C
          T1(1,J)=T1(1,J-1)/2.+DLAM*SUM
          DO 3 L=2,J
            K=J+1-L
            T1(L,K)=(4**(L-1)*T1(L-1,K+1)-T1(L-1,K))/(4**(L-1)-1.)
 3        CONTINUE
C
C   CHECK FOR CONVERGENCE BASED ON RELATIVE ERROR CRITERIA
C   IF CONVERGENCE IS MET THE INTEGRAL VALUE HAS BEEN
C   DETERMINED AND RETURN TO SOLN
C
          TD = T1(J,1)
          IF (TD.EQ.0.) TD = 1.
          IF(ABS((T1(J,1)-T1(J-1,1))/TD).LE.EPS)THEN
            XI=T1(J,1)
            GO TO 10
          ELSE
            ITS=J
          ENDIF
 1      CONTINUE
C
C     IF CONVERGENCE WAS NEVER ACHIEVED FOR THIS TIME STEP
C     (STOP THE PROGRAM RUN)
C
        IF(ITS.GE.NUMITS)THEN
c         WRITE(9,*)'CONVERGENCE WAS NOT FOUND...'
          STOP
        ELSE
        ENDIF
 10     RETURN
        END
```

```
      FUNCTION FN(Z,X,Y,TIME)
C     ***********************************************************************
C     ***********************************************************************
C     **                                                                   **
C     **  P U R P O S E                                                    **
C     **                                                                   **
C     **   THIS ROUTINE EVALUATES THE FUNCTI0N (INTEGRAND) AT A GIVEN      **
C     **   TIME AND LOCATION.                                              **
C     **                                                                   **
C     **                                                                   **
C     **  I N P U T    P A R A M E T E R S                                 **
C     **                                                                   **
C     **       Z .... REAL VARIABLE. THE TIME FOR WHICH THE FUNCTION       **
C     **              IS TO BE EVALUATED                                   **
C     **       X .... REAL VARIABLE. THE X-COORDINATE                      **
C     **       Y .... REAL VARIABLE. THE Y-COORDINATE                      **
C     **       TIME.. REAL VARIABLE. THE TIME AT WHICH THE SOLUTION        **
C     **              IS BEING DETERMINED                                  **
C     **                                                                   **
C     **                                                                   **
C     **  O U T P U T    P A R A M E T E R S                               **
C     **                                                                   **
C     **       FN.... REAL VARIABLE. THE VALUE OF THE FUNCTION FOR THE     **
C     **              INPUT PARAMETERS                                     **
C     **                                                                   **
C     **   AUTHOR: MIKE JOHNSON                                            **
C     **           U.T. AUSTIN                                             **
C     **                                                                   **
C     **   LANGUAGE: FORTRAN 77                                            **
C     **                                                                   **
C     **   REQUIRED ROUTINES: BIN                                          **
C     **                                                                   **
C     ***********************************************************************
C     ***********************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)

      COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
      COMMON /SOURCE/ B1(31),TB1(31),XB(1000),TB(1000)
      COMMON /OTHR/ D,CLAM,CMINW
C
C   THE FUNCTION EQUALS ZERO FOR THE SINGULARITY, SINCE
C   THE EVALUATION REQUIRES THE DIVISION BY TIME-Z
C
      IF(Z.GE.TIME)THEN
        FN=0.
      ELSE
C
C   FOR ALL NON-SINGULAR POINTS DETERMINE THE VALUE
C   OF THE FUNCTION; BREAK UP THE FUNCTION INTO
C   NUMERATOR, DENOMENATOR COMBINE TO GIVE THE VALUE
C   OF FN
C
C
        FNA=-(X**2./(4.*(TIME-Z)))-(Y**2./(2+4.*D*(TIME-Z)))
     1      -(TIME-Z)/4.-CLAM*(TIME-Z) + X/2.
        FNB=SQRT(((TIME-Z)**3.)*(1.+2.*D*(TIME-Z)))
        FN=BIN(Z)*EXP(FNA)/FNB
      ENDIF
C
C
      RETURN
      END
```

```
      FUNCTION BIN(Z)
C     ******************************************************************
C     ******************************************************************
C     **
C     **  P U R P O S E
C     **
C     **   THE PURPOSE OF THIS ROUTINE IS TO DETERMINE THE INPUT
C     **   CONCENTRATIONS FOR GIVEN TIMES. THIS IS DONE BY LINEAR
C     **   INTERPOLATION BETWEEN THE GIVEN INPUT DATA POINTS
C     **
C     **
C     **  I N P U T    P A R A M E T E R S
C     **
C     **       Z .... REAL VARIABLE. THE TIME FOR WHICH THE INPUT
C     **              CONCENTRATION IS DETERMINED
C     **
C     **
C     **  O U T P U T    P A R A M E T E R S
C     **
C     **       BIN... REAL VARIABLE. THE VALUE OF THE MASS FLUX
C     **              AT THE GIVEN TIME
C     **
C     **
C     **   AUTHOR: MIKE JOHNSON
C     **           U.T. AUSTIN
C     **   MODIFIED BY JIM WEAVER, USEPA
C     **
C     **   LANGUAGE: DOUBLE PRECISION FORTRAN 77
C     **   REQUIRED ROUTINES: BINARY
C     **   CALLED BY FN
C     **
C     ******************************************************************
C     ******************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
      COMMON /SOURCE/ B1(31),TB1(31),XB(1000),TB(1000)
C
      BIN = 0.
C     ***THERE IS NO MASS FLUX BEFORE THE INITIAL MASS FLUX TO THE
C     ***AQUIFER
C     ***THERE IS NO MASS FLUX AFTER THE LAST MASS FLUX TO THE AQUIFER
      IF (Z.LT.TB1(1).OR.TB1(NUMB).LT.Z) THEN
        RETURN
      END IF
C
C     ***INTERPOLATE THE INPUT MASS FLUX DISTRIBUTION
      CALL BINARY (Z,TB1,NUMB,1,IN)
      SLOPE = (B1(IN+1)-B1(IN))/(TB1(IN+1)-TB1(IN))
      BIN = B1(IN) + SLOPE*(Z-TB1(IN))
C
      RETURN
      END
```

```
      SUBROUTINE BINARY (U,X,NX,INCX,IN)
C     ******************************************************************
C     *                                                                *
C     *   BINARY                                                       *
C     *                                                                *
C     *                                                                *
C     *   PURPOSE....THIS SUBROUTINE PERFORMS A BINARY SEARCH ON       *
C     *   THE ARRAY X TO FIND THE INDEX OF THE X VALUE WHICH LIES      *
C     *   JUST BELOW U                                                 *
C     *                                                                *
C     *   INPUT ARGUMENTS.....................................        *
C     *   U      = POINT SEARCHED FOR                                  *
C     *   X      = ARRAY OF VALUES                                     *
C     *   NX     = NUMBER OF X VALUES                                  *
C     *   INCX   = INCREMENT (MUST = 1)                                *
C     *                                                                *
C     *   OUTPUT ARGUMENTS.....................................        *
C     *   IN     = INDEX OF X VALUE JUST BELOW U                       *
C     *                                                                *
C     *                                                                *
C     *   AUTHOR  JIM WEAVER                                           *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN                    *
C     *                                                                *
C     *   REFERENCE:  FORSYTHE, MALCOLM AND MOLER, 1977, PG. 76        *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77             *
C     *   REQUIRED ROUTINES...NONE                                     *
C     *                                                                *
C     ******************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      COMMON /COUN/ NUM(100)
      DIMENSION X(*)
      DATA I/1/
C
      NUM(20) = NUM(20) + 1
      IF (INCX.NE.1) STOP 'BINARY-INDEX'
C     ***X'S ORDERED BY INCREASING VALUE
      IF (X(1).LT.X(NX)) THEN
        IF (I.GT.NX) I = 1
        IF (U.LT.X(I).OR.X(I+1).LT.U) THEN
          IF (U.LT.X(I)) THEN
            J = I
            I = 1
          ELSE
            I = I + 1
            J = NX
          END IF
 2        CONTINUE
            K = (I + J)/2
            IF (U.LT.X(K)) J = K
            IF (U.GE.X(K)) I = K
          IF (J.GT.I+1) GO TO 2
        END IF
        IN = I
        IF (IN.EQ.NX) IN = IN - 1
      ELSE IF (X(1).GT.X(NX)) THEN
```

```
C       ***X'S ORDERED BY DECREASING VALUES
        IF (I.GT.NX) I = 1
        IF (U.LT.X(I+1).OR.X(I).LT.U) THEN
          IF (U.LT.X(I+1)) THEN
            J = NX
            I = I + 1
          ELSE
            J = I
            I = 1
          END IF
 5         CONTINUE
            K = (I + J)/2
            IF (U.LT.X(K)) I = K
            IF (U.GE.X(K)) J = K
          IF (J.GT.I+1) GO TO 5
        END IF
        IN = I
      END IF
      RETURN
      END
```

```
      SUBROUTINE NEWOUT (ICALL,TIME,C,IWELL,RMAX)
C     ******************************************************************
C     ******************************************************************
C     **
C     **  P U R P O S E
C     **
C     **   THIS ROUTINE PRINTS THE OUTPUT RESULTS
C     **
C     **
C     **   INPUT ARGUMENTS
C     **      ICALL  WRITE HEADING ON OUTPUT FILE PARAMETER
C     **              1     = HEADING
C     **              OTHER = NO HEADING
C     **     TIME    SIMULATION TIME (DAYS)
C     **     C()     CONCENTRATION VECTOR (KG/M3)
C     **     IWELL   WELL INDEX
C     **     RMAX    RADIUS OF SOURCE
C     **
C     **
C     **   LANGUAGE: DOUBLE PRECISION FORTRAN 77
C     **   REQUIRED ROUTINES: MESS2
C     **   CALLED BY: ADVECT,SOLN
C     **
C     ******************************************************************
C     ******************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      COMMON /DIMXYT/ XX(6),XY(6)
      COMMON /NUM/ NWELL,NUMT(6),NUMB,ITYPE
      COMMON /INPU/ MFIRST,NHITS
      DIMENSION CMAXX(6)
      DATA CMAXX,IWX /6*0.,0/
C
C     ***WRITE HEADINGS
      IF (ICALL.EQ.1) THEN
        WRITE (42,1) XX(IWELL)+RMAX,XY(IWELL)
C       ***WRITE TO PLOT FILE FOR AUTOMATED PLOTTING
        IF (IWX.EQ.0) THEN
          WRITE (43,11) XX(IWELL)+RMAX,XY(IWELL)
          IWX = 1
        ELSE
          WRITE (43,12) XX(IWELL)+RMAX,XY(IWELL)
        END IF
      END IF
C
      CALL MESS2 (5,TIME)
C
C     ***CONVERT CONCENTRATIONS TO MG/L ON OUTPUT
      WRITE (42,10) TIME,C*1000.
C
C
C
C     ***THE PLOT FILE:::
C     ***SAVE MAXIMUM CONCENTRATION
C     ***A DATA STATEMENT INITIALLIZES CMAXX
      IF (NHITS.GT.5) THEN
C       ***MULTIPLE INPUT MASS FLUXES WERE THE SAME (S.R. INFO)
C       ***OCCURS WHEN NO LENS FORMS
        WRITE (43,10) TIME,C*1000.
        NUMT(IWELL) = NUMT(IWELL) + 1
      ELSE
C       ***A SINGLE PEAK MASS FLUX
C       ***OCCURS WHEN THE OILENS FORMS
        IF (C.GT.CMAXX(IWELL)) CMAXX(IWELL) = C
        IF (C.GE.CMAXX(IWELL)) THEN
C
C         ***WRITE OUT EACH STEP FOR INCREASING CONCENTRATIONS
          WRITE (43,10) TIME,C*1000.
          CPREV = C
          NUMT(IWELL) = NUMT(IWELL) + 1
        ELSE IF (C.LT.CMAXX(IWELL)) THEN
C
C         ***ONLY WRITE OUT DECREASING CONCENTRATIONS IF ON FALLING LIMB
          IF (C.LT.CPREV*.99) THEN
            WRITE (43,10) TIME,C*1000.
            CPREV = C
            NUMT(IWELL) = NUMT(IWELL) + 1
          END IF
        END IF
      END IF
C
C
C
 1    FORMAT (//1X,'AQUIFER CONCENTRATION HISTORIES'/
     *            1X,'============================='//
     *            4X,'TIME',5X,'RECEPTOR LOCATION'/
     *            13X,'( X',(1X,F10.2)' )'/
     *            13X,'( Y',(1X,F10.2)' )'/
     *            1X,10(1H-),3X,(10(1H-),1X))
 10   FORMAT(1X,G10.4,3X,6(G10.4,1X))
 11   FORMAT ('BEGIN'/1X,G10.4,2X,G10.4)
 12   FORMAT (1X,G10.4,2X,G10.4)
      RETURN
      END
```

```
      SUBROUTINE ENDPLT (ICOMP,NWELL)
C     ***************************************************************
C     *
C     *  ENDPLT    COMPLETES THE PLOT FILE
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *    ICOMP   FLAG FOR CONCENTRATIONS COMPUTED
C     *    NWELL   NUMBER OF RECEPTOR POINTS
C     * OUTPUT ARGUMENTS............................................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  HSSMT
C     *
C     ***************************************************************
      CHARACTER*5 ENDCHR
C
C     ***FINISH OFF PLOT FILE FOR HSSM-WIN
C     IF (NWELL.GT.1) RETURN
C
      IF (ICOMP.EQ.1) THEN
        WRITE (43,9100)
      ELSE IF (ICOMP.EQ.0) THEN
        ENDCHR = 'END  '
        ENDCHR(4:4) = CHAR(13)
        ENDCHR(5:5) = CHAR(10)
        WRITE (43,9110) ENDCHR
      END IF
C
      RETURN
 9100 FORMAT ('STOP')
 9110 FORMAT (A5)
      END
```

```
      SUBROUTINE MESSAG (IM)
C     ***************************************************************
C     ***WRITE MESSAGES TO THE SCREEN
C     ***************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      CHARACTER*40 STR
C
C     ***ENTRY MESS2 IS USED TO OUTPUT A REAL*4 NUMBER WITH A MESSAG
      ENTRY MESS2 (IM,X)
C
C     ***ENTRY MESS3 IS USED TO OUTPUT A CHARACTER*40 STRING WITH MESSAGE
      ENTRY MESS3 (IM,STR)
C
C     ***ENTRY MESS4 IS USED TO OUTPUT TWO REAL*4 NUMBERS
      ENTRY MESS4 (IM,X,Y)
C
C     ***ENTRY MESS5 IS USED TO OUTPUT AN INTEGER
      ENTRY MESS5 (IM,II)
C
      IF (IM.EQ.1) THEN
C       ***TSGPLUME BANNER
        WRITE (*,9100)
        WRITE (*,9101)
        WRITE (*,9102)
        READ (*,*)
      END IF
C
C     IF (IWR.EQ.0) RETURN
C
      IF (IM.EQ.2) THEN
C       ***DATA INPUT
        WRITE (*,9202)
      ELSE IF (IM.EQ.3) THEN
C       ***SIMULATION
        WRITE (*,9203) II
      ELSE IF (IM.EQ.4) THEN
C       ***ENDING
        WRITE (*,9204)
      ELSE IF (IM.EQ.5) THEN
C       ***COMPUTING AT
        IF (X.GT.365.25) THEN
          I1 = IFIX(X)
          IT1 = IFIX(X)/365.25
          T1 = FLOAT(IT1)
          X22 = T1*365.25
          T2 = X-T1*365.25
          IF (T2.LT.0.) THEN
            CONTINUE
          END IF
          IF (IT1.EQ.1) THEN
            WRITE (*,9306) IT1,T2
          ELSE
```

```
          WRITE (*,9307) IT1,T2
        END IF
      ELSE
        WRITE (*,9305) X
      END IF
    ELSE IF (IM.EQ.6) THEN
C     ***DATA INITIALIZATION
      WRITE (*,9206)
    ELSE IF (IM.EQ.7) THEN
C     ***NAME OF THE OUTPUT FILE
      WRITE (*,9207) STR
    ELSE IF (IM.EQ.8) THEN
C     ***NAME OF THE PLOT FILE
      WRITE (*,9208) STR
    ELSE IF (IM.EQ.9) THEN
C     ***CALCULATING THE TOE OF THE HISTORY
      WRITE (*,9209)
    ELSE IF (IM.EQ.10) THEN
C     ***CALCULATING THE PEAK OF THE HISTORY
      WRITE (*,9210)
    ELSE IF (IM.EQ.11) THEN
C     ***REDUCING THE PLOT FILE
      WRITE (*,9211) II
    ELSE IF (IM.EQ.12) THEN
C     ***RECEPTOR LOCATIONS
      WRITE (*,9212) X,Y
    ELSE IF (IM.EQ.13) THEN
C     ***NO INPUT MASS FLUX
      WRITE (*,9213)
    ELSE IF (IM.EQ.14) THEN
C     ***NO MINIMUM CONCENTRATION FOUND
      WRITE (*,9214)
    ELSE IF (IM.EQ.15) THEN
C     ***MACHINE PRECISION
      WRITE (*,9215)
    ELSE IF (IM.EQ.16) THEN
C     ***SEARCH ALGORITHM ITERATIONS
      WRITE (*,9216) II
    ELSE IF (IM.EQ.17) THEN
C     ***RECEPTOR WITHIN THE LENS
      WRITE (*,9217) X,Y
    END IF
C
    RETURN
 9100 FORMAT (/
     *       15X,'*************************************************'/
     1       15X,'*                                               *'/
     2       15X,'*                   TSGPLUME                    *'/
     3       15X,'*                                               *'/
     4       15X,'*      TRANSIENT SOURCE GAUSSIAN PLUME MODEL    *'/
     5       15X,'*                                               *'/
     6       15X,'*                                               *'/
     *       15X,'*                MIKE JOHNSON                   *'/
     *       15X,'*              RANDALL CHARBENEAU               *'/
     6       15x,'*        THE UNIVERSITY OF TEXAS AT AUSTIN      *')
```

```
9101 FORMAT (15X,'*                                               *'/
    *        15x,'*                  JIM WEAVER                   *'/
    *        15X,'* ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY*'/
    *        15X,'*  UNITED STATES ENVIRONMENTAL PROTECTION AGENCY  *'/
    *        15X,'*                                               *'/
    *        15X,'*                  RSKERL V1.00                 *'/
    *        15X,'*************************************************'/
    *       //)
9102 FORMAT (10X,'PRESS ENTER TO CONTINUE')
9202 FORMAT (/10X,'*** DATA INPUT                  ')
9203 FORMAT (10X,'***'/
    *        10X,'*** COMPUTATION BEGINNING FOR RECEPTOR ',I5)
9204 FORMAT (10X,'*** TSGPLUME END                 '/)
9305 FORMAT (10X,'*** COMPUTATION AT ',F7.2' DAYS COMPLETED')
9306 FORMAT (10X,'*** COMPUTATION AT ',I5,' YEAR  ',F7.2,
    *            ' DAYS COMPLETED')
9307 FORMAT (10X,'*** COMPUTATION AT ',I5,' YEARS ',F7.2,
    *            ' DAYS COMPLETED')
9206 FORMAT (10X,'*** DATA INITIALIZATION          ')
9207 FORMAT (10X,'*** OUTPUT FILE: '/
    *        10X,'***   ',A40)
9208 FORMAT (10X,'*** PLOT FILE: '/
    *        10X,'***   ',A40)
9209 FORMAT (10X,'*** CALCULATING THE TOE TIME OF THE HISTORY')
9210 FORMAT (10X,'*** CALCULATING THE PEAK TIME OF THE HISTORY')
9211 FORMAT (10X,'*** REDUCING THE SIZE OF THE PLOT FILE-RECEPTOR',
    *           I5)
9212 FORMAT (10X,'*** LOCATION X = ',G10.4/
    *        10X,'***          Y = ',G10.4)
9213 FORMAT (10X,'*** NO INPUT MASS FLUX DISTRIBUTION')
9214 FORMAT (10X,'*** NO MINIMUM CONCENTRATION AT THIS RECEPTOR')
9215 FORMAT (10X,'*** CALCULATING FLOATING POINT PRECISION')
9216 FORMAT (10X,'*** SEARCH ALGORITHM COMPLETED IN ',I4,' ITERATIONS')
9217 FORMAT (10X,'*** '/
    *        10X,'*** RECEPTOR WITHIN THE RADIUS OF THE LENS'/
    *        10X,'***    NO CALCULATION FOR X = ',G10.4/
    *        10X,'***                       Y = ',G10.4/)
     END
```

```
      SUBROUTINE DPF (IFILE,MFILE,OFILE,NOF,IFACE,IEND,IPRE,IINP)
C     ************************************************************
C     *
C     *  DPF    DISPLAY THE FILE NAMES AND PROMPT FOR NEW FILES
C     *
C     *                                                          
C     *
C     * INPUT ARGUMENTS.....................................
C     *   IFILE   INPUT FILE NAME
C     *   OFILE   OUTPUT FILE NAMES
C     *   NOF     NUMBER OF OUTPUT FILE NAMES
C     *   IFACE   CHARACTER*1 INTERFACE FLAG
C     *   IEND    EXIT FLAG (DO NOT RUN HSSM IF IEND = 1)
C     *   IPRE    PREKOPT RUN FLAG (=1 IF PREKOPT HAS RUN)
C     *   IINP    FILE NAME HAS BEEN CHANGED IF IINP = 1
C     * OUTPUT ARGUMENTS....................................
C     *   IFILE   INPUT FILE NAME
C     *   OFILE   OUTPUT FILE NAMES
C     *   IEND    EXIT FLAG (DO NOT RUN HSSM IF IEND = 1)
C     *   IPRE    PREKOPT RUN FLAG (=1 IF PREKOPT HAS RUN)
C     *   IINP    FILE NAME HAS BEEN CHANGED IF IINP = 1
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  DIRECT
C     * CALLED BY   INFO
C     *
C     ************************************************************
      CHARACTER IFACE*1
      CHARACTER*40 IFILE,MFILE,OFILE(*),ANS*1
      CHARACTER*60 OTEXT(8)
      LOGICAL EXIST
C
      IEND = 0
      IPRE = 0
      IINP = 0
C
 5    CONTINUE
C
C     ***PROMPT FOR NEW FILE NAMES
      WRITE (*,9000) IFILE,MFILE,(OFILE(II),II=1,NOF)
      WRITE (*,*) ' '
C
      WRITE (*,*) 'TO RUN TSGPLUME         ENTER <RETURN>'
      IX = 0
      IF (IFACE.EQ.'W'.OR.IFACE.EQ.'w') IX = 1
      IF (IX.EQ.0) THEN
        WRITE (*,*) 'TO CHANGE INPUT FILE    ENTER F'
        WRITE (*,*) 'TO VIEW DIRECTORY       ENTER D'
      END IF
      WRITE (*,*) 'TO EXIT                 ENTER 1'
      WRITE (*,*) ' '
      READ (*,'(A)') ANS
C
      IF (IX.EQ.0.AND.(ANS.EQ.'f'.OR.ANS.EQ.'F')) THEN
        IINP = 1
        IFILE = ' '
      ELSE IF (ANS.EQ.'1') THEN
C       ***EXIT WITHOUT RUNNING TSGPLUME
        WRITE (*,*) ' '
        IEND = 1
        RETURN
      ELSE IF (IX.EQ.0.AND.(ANS.EQ.'D'.OR.ANS.EQ.'d')) THEN
        CALL DIRECT
        GO TO 5
      END IF
C
      RETURN
 9000 FORMAT (/1X,'OUTPUT AND PLOT FILE NAMES:'//
     *         1X,'HSSM-KO INPUT DATA FILE ',A40/
     *         1X,'KOPT/OILENS OUTPUT      ',A40/
     *         1X,'TSGPLUME INPUT          ',A40/
     *         1X,'TSGPLUME OUTPUT         ',A40/
     *         1X,'TSGPLUME PLOT           ',A40/)
      END
```

```
      SUBROUTINE MBISEC (FNC,ZB,ZE,CONV,EPS,IS,SV,TM,ZM,VM,IE,
     *NVS,VSAVE,ZSAVE)
C     ************************************************************
C     *                                                          *
C     *   BISEC                                                  *
C     *                                                          *
C     *                                                          *
C     *   PURPOSE....THIS SUBROUTINE USES THE BISECTION METHOD TO  *
C     *   FIND THE ZERO OF A FUNCTION                            *
C     *                                                          *
C     *   INPUT ARGUMENTS.....................................   *
C     *   FNC    = SUBROUTINE NAME CONTAINING FUNCTION           *
C     *   ZB     = LOWER LIMIT OF SEARCH                         *
C     *   ZE     = UPPER LIMIT OF SEARCH                         *
C     *   CONV   = CONVERGENCE TOLERANCE                         *
C     *   EPS    = MACHINE FLOATING POINT PRECISION (FROM ROUTINE MEPS)
C     *   IS     = MAXIMUM NUMBER OF ITERATIONS                  *
C     *   SV     = FUNCTION VALUE TO BE MATCHED BY BISEC ITERATION *
C     *   TM     = REAL PARAMETER ARRAY PASSED TO FNC
C     *            TM(1) = XDIST
C     *            TM(2) = YDIST
C     *            TM(3) = CX1
C     *   NVS    = NUMBER OF SAVED RESULTS (PREVIOUS)
C     *                                                          *
C     *   OUTPUT ARGUMENTS....................................   *
C     *   ZM     = BISEC RESULT (ZB.LE.ZM.LE.ZE)                 *
C     *   VM     = FUNCTION VALUE AT ZM (SHOULD MATCH SV)        *
C     *   IE     = ERROR CODE IE=-1: BISEC FAILED                *
C     *   NVS    = NUMVER OF SAVED RESULTS (PREVIOUS + CURRENT)
C     *   VSAVE()= ARRAY OF SAVED FUNCTION VALUES
C     *   ZSAVE()= ARRAY OF SAVED INDEPENDENT VARIABLES
C     *                                                          *
C     *                                                          *
C     *   AUTHOR  JIM WEAVER                                     *
C     *           THE UNIVERSITY OF TEXAS AT AUSTIN              *
C     *                                                          *
C     *   REFERENCE: FORSYTHE, MALCOLM & MOLER, 1977, PG. 157    *
C     *   SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77       *
C     *   REQUIRED ROUTINES...SEE INPUT ARGUMENTS                *
C     *   THANKS TO SHERYL FRANKLIN                              *
C     ************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      DIMENSION TM(*)
      DIMENSION VSAVE(*),ZSAVE(*)
C
      IE = -1
      XF = 1.
      C = 0.
      A = ZB
      B = ZE
C
      IF (IS.GT.100) STOP 'MAXIMUM BISEC ITERATIONS 100'
C
C     ***ENDPOINT VALUES
      CALL FNC (A,TM,V1)
      FA = V1-SV
      NVS=NVS+1
      ZSAVE(NVS) = A
      VSAVE(NVS) = V1
      CALL FNC (B,TM,V2)
      FB = V2-SV
      NVS=NVS+1
      ZSAVE(NVS) = B
      VSAVE(NVS) = V2
      IF (FA*FB.GT.0.) RETURN
      IE = 0
      II=NVS
      FC = FB
C
C     ***ITERATE FOR A MAXIMUM NUMBER OF ITERATIONS
      DO 1  I=1,IS
        IF (FB*FC.GT.0.) THEN
          C = A
          FC = FA
          D = B - A
          E = D
        END IF
        IF (ABS(FC).LT.ABS(FB)) THEN
          A = B
          FA = FB
          B = C
          FB = FC
          C = A
          FC = FA
        END IF
        TL1 = 2.*EPS*ABS(B) + 0.5*CONV
        XM = 0.5*(C-B)
        IF (ABS(XM).LE.TL1.OR.FB.EQ.0.) THEN
          ZM = B
          VM = V2
          IE = 0
          CALL MESS5 (16,I)
          RETURN
        END IF
        IF (ABS(E).GE.TL1.AND.ABS(FA).GT.ABS(FB)) THEN
          S = FB/FA
          IF (A.EQ.C) THEN
            P = 2.*XM*S
            Q = 1.-S
          ELSE
            T = FA/FC
            R = FB/FC
            P = S*((C-B)*T*(T-R) - (B-A)*(R-1.))
            Q = (R-1.)*(S-1.)*(T-1.)
          END IF
          IF (P.GT.0.) Q = -Q
          P = ABS(P)
          IF (2.*P.LT.MIN(3.*XM*Q-ABS(TL1*Q),ABS(E+Q))) THEN
            E = D
            D = P/Q
```

```
          ELSE
            D = XM
            E = D
          END IF
        ELSE
          D = XM
          E = D
        END IF
        A = B
        FA = FB
        IF (ABS(D).GT.TL1) THEN
          B = B+D
        ELSE
          B = B+SIGN(TL1,XM)
        END IF
        CALL FNC (B,TM,V2)
        FB = V2-SV
        NVS = NVS + 1
        ZSAVE(NVS) = B
        VSAVE(NVS) = V2
1     CONTINUE
      STOP 'MBISEC: FAIL'
      END
```

```
      SUBROUTINE MEPS (EPS)
C     ******************************************************************
C     *
C     * MEPS   MACHINE FLOATING POINT PRECISION
C     *        SEE  FORSYTHE, MALCOLM AND MOLER, PG.14
C     *
C     * INPUT ARGUEMENTS......(NONE).................................
C     * OUTPUT ARGUMENT.............................................
C     *    EPS   ESTIMATE OF MACHINE PRECISION
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  MESSAG
C     * CALLED BY  HSSMT
C     *
C     ******************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
C
C
      CALL MESSAG (15)
C
      EPS = 1.
 10   CONTINUE
        EPS = 0.5*EPS
        EP1 = EPS + 1.
      IF (EP1.GT.1) GO TO 10
C
      RETURN
      END
```

```
      SUBROUTINE RFUNC (TIME,TM,CONC)
C     ****************************************************************
C     *
C     *  RFUNC  IS USED IN MBISEC TO DETERMINE THE CONCENTRATION FOR
C     *         A GIVEN TIME
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............................................
C     *    TIME    SOLUTION TIME
C     *    TM      PARAMETERS PASSED TO RFUNC
C     * OUTPUT ARGUMENTS............................................
C     *    CONC    CONCENTRATION
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  ROMINT
C     * CALLED BY  MBISEC
C     *
C     ****************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
      DIMENSION TM(*)
C
      XDIST=TM(1)
      YDIST=TM(2)
      CX1=TM(3)
C
      CALL ROMINT(TIME,XDIST,YDIST,XI)
      CONC=XI*XDIST/CX1
C
      RETURN
      END
```

```
      SUBROUTINE FILES (NFILES,TEXT,IN,FNAME)
C     ****************************************************************
C     ***
C     ***FILES PROMTS THE USER FOR THE DESIRED INPUT DATA FILE AND
C     ***OUTPUT FILE NAMES.  IF THE INPUT FILE DOES NOT EXIST THE USER IS
C     ***PROMPTED FOR ANOTHER. IF THE OUTPUT FILE ALREADY EXISTS THE USER
C     ***IS ASKED IF THE FILE SHOULD BE OVERWRITTEN.
C     ***VII-31-1989, V-31-90
C     ***
C     ***INPUT ARGUMENTS
C     ***NFILES = NUMBER OF OUTPUT FILES
C     ***IN()   = ARRAY OF OUTPUT FILE NUMBERS (KEYED TO WRITE STATEMENTS
C     ***         IN PROGRAM
C     ***
C     ****************************************************************
      CHARACTER*40 IFILE,OFILE,ANS*1
      DIMENSION IN(*)
      CHARACTER*40 FNAME(*)
      CHARACTER*40 TEXT(*)
      LOGICAL EXIST
      DATA IFILE /' '/
C
 5    CONTINUE
      IEND = 0
C
C     ***CHOOSE TO READ INPUT FILE,
      WRITE (*,*) ' '
      WRITE (*,*) 'ENTER THE NAME OF THE INPUT FILE'
      WRITE (*,*) '(40 CHARACTER LIMIT)'
      WRITE (*,*) '*---+---*---+---*---+---*---+---*'
      READ (*,'(A)') IFILE
C
C     ***CHECK FOR EXISTENCE OF FILE NAME
      INQUIRE (FILE=IFILE,EXIST=EXIST)
      IF (EXIST.EQV..FALSE.) THEN
        WRITE (*,*) 'INPUT DATA FILE DOES NOT EXIST--REENTER'
        WRITE (*,*) ' '
        GO TO 5
      END IF
C
C     ***OPEN AND REWIND INPUT FILES
      OPEN (5,FILE=IFILE,STATUS='OLD')
      REWIND 5
C
      RETURN
      END
```

```
      SUBROUTINE DIRECT
C     ****************************************************************
C     *
C     *  DIRECT ISSUES A DOS DIRECTORY COMMAND
C     *
C     *
C     *
C     * INPUT ARGUMENTS......(NONE)..............................
C     * OUTPUT ARGUMENTS.....(NONE)..............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  CMD
C     * CALLED BY   DPF
C     *
C     ****************************************************************
      CHARACTER*256 STR
      INTEGER*4 I4
C
C     ***EXECUTE VMS DIR COMMAND
      STR = 'DIR |MORE'
C
C     ***EXECUTE OPERATING SYSTEM COMMAND, STR
C     ***RETURNS ERROR CODE, I4
      CALL CMD (STR,I4)
C
      RETURN
      END
```

```
      Interface to integer*2 function system [c]
     &  (string[reference])
      character*1 string
      end
      subroutine cmd(a,k)
c     ****************************************************************
c     *
c     *  cmd   issues a dos command from fortran code
c     *
c     *
c     *
c     * input arguments..........................................
c     *   a     character string containing dos command
c     * output arguments.........................................
c     *   k     length of command string
c     *
c     * jim cloud
c     * computer services corp
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * single precision ansi std x3.9-1978 fortran 77
c     * required routines   none
c     * called by  direct,pcon,delfi
c     *
c     *
c     *
c     ****************************************************************
      character a*(*), cstr*256,CSTR2*256
      integer k
      integer*2 system
      integer*1 nul
      data nul/0/
C
      k=len_trim(a)
      if(k.gt.255)return
      cstr=a(1:k)//nul
C
C     ***TURN OFF SCREEN ECHO OF COMMAND
      CSTR2='ECHO OFF'//NUL
      K=INT(SYSTEM(CSTR2))
C
C     ***EXECUTE DOS COMMAND
      k=int(system(cstr))
C
C     ***TURN ON SCREEN ECHO OF COMMANDS
      CSTR2='ECHO ON'//NUL
      K=INT(SYSTEM(CSTR2))
C
      return
      end
```

```
      SUBROUTINE NSOPEN (IFILE)
C     ****************************************************************
C     *
C     *  NSOPEN   NONSTANDARD OPENING OF THE INPUT FILE
C     *           FOR COMPATIBILITY WITH HSSM-WIN
C     *
C     *
C     *
C     * INPUT ARGUMENTS........(NONE).............................
C     * OUTPUT ARGUMENTS.........................................
C     *   IFILE    INPUT DATA FILE NAME
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  HSSMT
C     *
C     ****************************************************************
      CHARACTER*40 IFILE
C
      OPEN (5,FILE=' ',STATUS='OLD')
      INQUIRE (5,NAME=IFILE)
C
      RETURN
      END
```

```
      SUBROUTINE PCON (OFILE1,OFILE2)
C     ****************************************************************
C     *
C     *  PCON CREATES THE OUTPUT FILES FROM THE TEMPORARY FILES
C     *
C     *
C     *
C     * INPUT ARGUMENTS.........................................
C     *   OFILE1   MAIN OUTPUT FILE
C     *   OFILE2   PLOT FILE
C     * OUTPUT ARGUMENTS.............(NONE).....................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  CMD,MESS3
C     * CALLED BY  HSSMT
C     *
C     ****************************************************************
      CHARACTER*40 OFILE1,OFILE2
      CHARACTER*256 STR
      INTEGER*4 I4
C
C     ***CONCANTINATE OUTPUT FILES
C
C     ***MAIN OUTPUT FILE
      CALL MESS3 (7,OFILE1)
      REWIND 42
C
      STR = 'COPY T42.TMP '//OFILE1//' > OFILE3'
C
C     ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
      CLOSE (42)
C
C     ***EXECUTE OPERATING SYSTEM COMMAND, STR
C     ***RETURNS ERROR CODE, I4
      CALL CMD (STR,I4)
C
C     ***DELETE CONCATENATION MESSAGE FILES
      STR = 'DEL OFILE3'
      CALL CMD (STR,I4)
C
C     ***PLOT FILE
      CALL MESS3 (8,OFILE2)
      REWIND 44
C
      STR = 'COPY T44.TMP '//OFILE2//' > OFILE4'
C
C     ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
```

```
      CLOSE (44)
C
C     ***EXECUTE OPERATING SYSTEM COMMAND, STR
C     ***RETURNS ERROR CODE, I4
      CALL CMD (STR,I4)
C
      STR = 'DEL OFILE4'
      CALL CMD (STR,I4)
C
      RETURN
      END
```

```
      SUBROUTINE DELFI
C     ****************************************************************
C     *
C     *  DELFI   DELETES TEMPORARY OUTPUT FILES
C     *
C     *
C     *
C     * INPUT ARGUMENTS.............(NONE)..........................
C     * OUTPUT ARGUMENTS............(NONE)..........................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  HSSMT
C     *
C     ****************************************************************
C
C     ***REOPEN FILES AND DELETE THEM
      OPEN (42,FILE='T42.TMP',STATUS='OLD')
      CLOSE (42,STATUS='DELETE')
      OPEN (43,FILE='T43.TMP',STATUS='OLD')
      CLOSE (43,STATUS='DELETE')
      OPEN (44,FILE='T44.TMP',STATUS='OLD')
      CLOSE (44,STATUS='DELETE')
C
      REWIND 5
C
      RETURN
      END
```

```
      SUBROUTINE CHKFLE
C     ****************************************************************
C     *
C     *  CHKFLE
C     *
C     * OPEN THE TEMPORARY FILES NEEDED BY HSSM FOR
C     * OUTPUTTING MODEL RESULTS
C     *  CHECK FOR PRIOR EXISTENCE OF INPUT FILES
C     *
C     *
C     *
C     * INPUT ARGUMENTS........(NONE)..............................
C     * OUTPUT ARGUMENTS.......(NONE)..............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  HSSMT
C     *
C     ****************************************************************
      LOGICAL FEXIST
C
C     ***FILE 42 IS THE MAIN OUTPUT FILE
      INQUIRE (FILE='T42.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (42,FILE='T42.TMP',STATUS='UNKNOWN')
        CLOSE (42,STATUS='DELETE')
      END IF
C
C     ***FILE 43 IS THE TEMPORARY PLOT FILE
      INQUIRE (FILE='T43.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (43,FILE='T43.TMP',STATUS='UNKNOWN')
        CLOSE (43,STATUS='DELETE')
      END IF
C
C     ***FILE 44 IS THE PLOT FILE
      INQUIRE (FILE='T44.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (44,FILE='T44.TMP',STATUS='UNKNOWN')
        CLOSE (44,STATUS='DELETE')
      END IF
C
      RETURN
      END
```

3.3 Source Code for NTHICK

```
      program nthick
c     *******************************************************
c     *
c     * nthick
c     *
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma  74820
c     * 405-436-8545
c     *
c     * 9-9-93
c     *
c     * double precision, ansi x3.9-1978 FORTRAN 77
c     * required routines io,ioproc,meps,nsopen,satura,specif
c     *
c     *******************************************************
      implicit double precision (a-h), double precision (o-z)
      character*1 ans
c
c
c     ***read the input data
      call io (eta,hwe,xlamb,swr,spr,sprb,wsig,osig,wrho,prho,owsig)
c
c
c     ***calculate the constants
      call ioproc (hwe,wsig,osig,wrho,prho,owsig,hceow,hceao)
c
c
c
c     ***write out headings
      write (*,8999)
      write (4,8999)
      write (*,*) ' '
      write (*,*) 'Press any key to continue'
      read (*,*)
c
c     ***echo data to the screen
      write (*,9200) eta,hwe,xlamb,swr
      write (*,9201) spr,sprb
      write (*,9202) wsig,osig
      write (*,9203) wrho,prho
      write (*,9204) owsig
c     ***echo data to the file
      write (4,9200) eta,hwe,xlamb,swr
      write (4,9201) spr,sprb
      write (4,9202) wsig,osig
      write (4,9203) wrho,prho
      write (4,9204) owsig
c
      write (*,*) ' '
      write (*,*) 'Press any key to continue'
      read (*,*)
c
c
      write (*,9000)
      write (4,9000)
c
c     ***minimum observation well thickness
      bohat = owsig/wsig/(1.-prho/wrho)
      bohat = bohat - osig*wrho/wsig/prho
      bohat = bohat*hwe
c
c
c     ***generate a sequence of napl saturations (somax) and
c     ***averaged formation thicknesses (do) from
c     ***observation well thicknesses (bo)
      do 100 bo = bohat,2.0,0.15
c
c
c       ***calculate the average formation thickness and
c       ***oil saturation in the lens
        call satura (bo,eta,xlamb,swr,spr,sprb,osig,owsig,wrho,prho,
     *                   hceao,hceow,do,somax)
c
c
c
        if (somax.gt.1.0) go to 100
        if (somax.lt.0.0) go to 100
        if (do.lt.0.0) go to 100
c
c       ***output the results
        write (*,9001) bo,do,somax
        write (4,9001) bo,do,somax
c
 100  continue
c
c
c
c     ***exit the program
      write (*,*) 'Exit the program ? (Y or N) '
      read (*,'(a)') ans
c
c
c
      if (ans.eq.'N'.or.ans.eq.'n') then
c       ***do not exit yet
c
c
c       ***find the value of somax for a specific do
        call specif (bohat,eta,xlamb,swr,spr,sprb,osig,owsig,
     *               wrho,prho,hceao,hceow,do,bo,somax)
c
c
      end if
c
c
```

```
c
c
 8999 format (///2x,'*****************************************'/
     *           2x,'Estimate of NAPL saturation in OILENS'/
     *           2x,'*****************************************'/)
 9000 format (//2x,'Observation ',1x,'Averaged    ',1x,'   NAPL   '/
     *          2x,'    Well    ',1x,'Formation   ',1x,'Saturation'/
     *          2x,'  Thickness ',1x,'Thickness   ',1x/
     *          2x,'   (m)      ',1x,'   (m)      '/
     *          2x,'============',1x,'============',1x,'==========')
 9001 format (2x,f10.4,1x,f10.4,1x,f10.4)
 9200 format (2x,'Porosity                       ',f10.4,' (*)'/
     *        2x,'Air entry head                 ',f10.4,' (m)'/
     *        2x,'Brooks and Corey lambda        ',f10.4,' (*)'/
     *        2x,'Residual water saturation      ',f10.4,' (*)')
 9201 format (2x,'Vadose zone residual NAPL sat. ',f10.4,' (*)'/
     *        2x,'Aquifer residual NAPL sat.     ',f10.4,' (*)')
 9202 format (2x,'Water surface tension          ',f10.4,' (dyne/cm)'/
     *        2x,'NAPL surface tension           ',f10.4,' (dyne/cm)')
 9203 format (2x,'Water density                  ',f10.4,' (g/cc)'/
     *        2x,'NAPL density                   ',f10.4,' (g/cc)')
 9204 format (2x,'NAPL/water interfacial tension ',f10.4,' (dyne/cm)')
c
      write (*,*) ' '
      stop '         ***successful exection of nthick'
      end
```

```
      subroutine satura (bo,eta,xlamb,swr,spr,sprb,osig,owsig,wrho,prho,
     *                   hceao,hceow,do,somax)
c     ****************************************************************
c     *
c     *  io  inputs data to nthick                                     c
c     *                                                                c     ***averaged formation thickness
c     *                                                                      do = alpha + beta*bo
c     * input arguments                                                c
c     *  none                                                          c
c     *                                                                c     ***napl saturation in the lens
c     * output arguments                                                     somax = do/eta/bo
c     *  eta     porosity                                              c
c     *  hwe     air entry head in meters                              c
c     *  xlamb   Brooks and Corey's lambda                             c
c     *  swr     residual water saturation                                   return
c     *  spr     vadose zone residual NAPL saturation                        end
c     *  sprb    aquifer residual NAPL saturation
c     *  wsig    water/air surface tension
c     *  osig    NAPL/air surface tension
c     *  wrho    water density
c     *  prho    NAPL density
c     *  owsig   water/NAPL interfacial tension
c     *
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma  74820
c     * 405-436-8545
c     *
c     * 9-9-93
c     *
c     * double precision, ansi x3.9-1978 FORTRAN 77
c     * required routines:  none
c     * called by  nthick,specif
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
c
c
c
c
c     ***the alpha constant
      alpha = eta*(xlamb*(1.-swr) - sprb)*hceow
      alpha = alpha - xlamb*eta*(1.-swr-spr)*hceao
      alpha = alpha/(1.-xlamb)
c
c     ***the chi constant
      chi = osig/owsig
      chi = chi*(wrho-prho)/prho
      chi = chi*(  ((1.-swr-spr)/(1.-swr-sprb))**(1./xlamb)  )
c
c
c     ***the beta(bo)  function
      beta = eta*(1.-swr)
      beta = beta + chi*eta*spr/(1.-chi)
      t1   = eta*(1.-swr-sprb)/(1.-xlamb)
      t2   = ((1.-chi)*hceow/bo)**xlamb
      beta = beta - t1*t2
```

```
      subroutine io (eta,hwe,xlamb,swr,spr,sprb,wsig,osig,wrho,
     *prho,owsig)
c     ****************************************************************
c     *
c     *  io  inputs data to nthick
c     *
c     *
c     * input arguments
c     *  none
c     *
c     * output arguments
c     *  eta     porosity
c     *  hwe     air entry head in meters
c     *  xlamb   Brooks and Corey's lambda
c     *  swr     residual water saturation
c     *  spr     vadose zone residual NAPL saturation
c     *  sprb    aquifer residual NAPL saturation
c     *  wsig    water/air surface tension
c     *  osig    NAPL/air surface tension
c     *  wrho    water density
c     *  prho    NAPL density
c     *  owsig   water/NAPL interfacial tension
c     *
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma  74820
c     * 405-436-8545
c     *
c     * 9-9-93
c     *
c     * double precision, ansi x3.9-1978 FORTRAN 77
c     * required routines:  nsopen
c     * called by nthick
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
      character*40 ifile
c
c
c     ***open up the input data file
      call nsopen (ifile)
c
c
c
c     ***read the input data from the file
      read (4,*) eta,hwe,xlamb,swr
      read (4,*) spr,sprb
      read (4,*) wsig,osig
      read (4,*) wrho,prho
c
c
c     ***read the napl-water interfacial tension
c     ***if in the input file:
      owsig = 0.0
      read (4,*,end=20,err=20) owsig
c
      return
c
 20   continue
c
c     ***prompt for the value, since it is not in the input data file
      write (*,*) ' '
      write (*,*) 'Enter the NAPL/water interfacial tension value'
      write (*,*) '(in dyne/cm)'
      read (*,*) owsig
c
c     ***close the file
      close (4)
c
c     ***reopen the file and write all the data to the file
c     ***(including the NAPL/water interfacial tension)
      open (4,file=ifile,status='OLD')
c
c     ***write the data to the file
c     ***read the input data from the file
      write (4,9100) eta,hwe,xlamb,swr
      write (4,9100) spr,sprb
      write (4,9100) wsig,osig
      write (4,9100) wrho,prho
      write (4,9100) owsig
c
 9100 format (4f10.4)
      return
      end
```

```
      SUBROUTINE NSOPEN (IFILE)
C     ****************************************************************
C     *
C     *  NSOPEN   NONSTANDARD OPENING OF THE INPUT FILE
C     *           FOR COMPATIBILITY WITH HSSM-WIN
C     *
C     *
C     *
C     * INPUT ARGUMENTS........(NONE).............................
C     * OUTPUT ARGUMENTS.......................................
C     *   IFILE    INPUT DATA FILE NAME
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  IO
C     *
C     ****************************************************************
      CHARACTER*40 IFILE
C
      OPEN (4,FILE=' ',STATUS='OLD')
      INQUIRE (4,NAME=IFILE)
C
      RETURN
      END
```

```
      subroutine ioproc (hwe,wsig,osig,wrho,prho,owsig,hceow,hceao)
c     ****************************************************************
c     *                                                              *
c     *  io  calculates constants for nthick                         *
c     *                                                              *
c     *                                                              *
c     * input arguments.............................................*
c     *  hwe    air entry head (water/air) in cm                     *
c     *  wsig    water/air surface tension                           *
c     *  osig    NAPL/air surface tension                            *
c     *  wrho    water density                                       *
c     *  prho    NAPL density                                        *
c     *  owsig   water/NAPL interfacial tension                      *
c     *                                                              *
c     * output arguments............................................*
c     *  hceow   NAPL/water entry head                               *
c     *  hceao   air/NAPL entry head                                 *
c     *                                                              *
c     *                                                              *
c     * jim weaver                                                   *
c     * robert s. kerr environmental research laboratory            *
c     * united states environmental protection agency                *
c     * ada, oklahoma  74820                                        *
c     * 405-436-8545                                                *
c     *                                                              *
c     * 11-2-93                                                      *
c     *                                                              *
c     * double precision, ansi x3.9-1978 FORTRAN 77                  *
c     * required routines:  none                                     *
c     * called by  nthick                                            *
c     *                                                              *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
c
c
c     ***the NAPL/water entry head
      hceow = wrho*owsig*hwe/wsig/(wrho-prho)
c
c
c     ***the air/NAPL entry head
      hceao = wrho*osig*hwe/prho/wsig
c
c
      return
      end
```

```
      subroutine specif (bohat,eta,xlamb,swr,spr,sprb,osig,owsig,
     *              wrho,prho,hceao,hceow,do,bo,somax)
c     ****************************************************************
c     *                                                              *
c     *  specif   determines the value of somax given an input dmatch*
c     *                                                              *
c     *                                                              *
c     * input arguments.............................................
c     *   bohat  minimum observation well thickness                  *
c     *   eta    porosity                                            *
c     *   xlamb  brooks and corey lambda                             *
c     *   swr    residual water saturation                           *
c     *   spr    residual NAPL saturation in the vadose zone         *
c     *   sprb   residual NAPL saturation in the aquifer             *
c     *   osig   NAPL surface tension                                *
c     *   owsig  NAPL/water interfacial tension                      *
c     *   wrho   water density                                       *
c     *   prho   NAPL density                                        *
c     *   hceao  entry head in an air/NAPL system                    *
c     *   hceow  entry head in an NAPL/water system                  *
c     * output arguments............................................
c     *   do      formation NAPL thickness                           *
c     *   bo      observation well thickness                         *
c     *   somax   NAPL saturation                                    *
c     *                                                              *
c     * jim weaver                                                   *
c     * robert s. kerr environmental research laboratory            *
c     * united states environmental protection agency                *
c     * ada, oklahoma 74820                                         *
c     *                                                              *
c     * single precision ansi std x3.9-1978 fortran 77               *
c     * required routines  meps,satura                               *
c     * called by  nthick                                            *
c     *                                                              *
c     ****************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)
c
c     ***determine the machine epislon
      call meps (eps)
c
c
      write (*,*) ' '
      write (*,*)
     *  'Enter the average NAPL thickness in the formation (m)'
      read (*,*) dmatch
c
c     ***bisection search parameters
      zb = bohat
      ze = 10.*bohat
      conv = 1.e-6
      is = 200
```

```
      sv = dmatch
c
c     ***find the value of observation well thickness that
c     ***corresponds to the given average formation thickness
c     ***bisection search from
C
      IE = -1
      XF = 1.
      C = 0.
      A = ZB
      B = ZE
C
      IF (IS.GT.250) STOP 'MAXIMUM BISEC ITERATIONS 250'
C
C     ***ENDPOINT VALUES
      call satura (a,eta,xlamb,swr,spr,sprb,osig,owsig,wrho,prho,
     *                   hceao,hceow,v1,somax)
c
      FA = V1-SV
      call satura (b,eta,xlamb,swr,spr,sprb,osig,owsig,wrho,prho,
     *                   hceao,hceow,v2,somax)
c
c
      FB = V2-SV
      IF (FA*FB.GT.0.) RETURN
      IE = 0
      FC = FB
C
C     ***ITERATE FOR A MAXIMUM NUMBER OF ITERATIONS
      DO 1  I=1,IS
        IF (FB*FC.GT.0.) THEN
          C = A
          FC = FA
          D = B - A
          E = D
        END IF
        IF (ABS(FC).LT.ABS(FB)) THEN
          A = B
          FA = FB
          B = C
          FB = FC
          C = A
          FC = FA
        END IF
        TL1 = 2.*EPS*ABS(B) + 0.5*CONV
        XM = 0.5*(C-B)
        IF (ABS(XM).LE.TL1.OR.FB.EQ.0.) THEN
          ZM = B
          VM = V2
          IE = 0
          go to 20
        END IF
        IF (ABS(E).GE.TL1.AND.ABS(FA).GT.ABS(FB)) THEN
          S = FB/FA
          IF (A.EQ.C) THEN
            P = 2.*XM*S
            Q = 1.-S
          ELSE
            T = FA/FC
            R = FB/FC
            P = S*((C-B)*T*(T-R) - (B-A)*(R-1.))
            Q = (R-1.)*(S-1.)*(T-1.)
          END IF
          IF (P.GT.0.) Q = -Q
          P = ABS(P)
          IF (2.*P.LT.MIN(3.*XM*Q-ABS(TL1*Q),ABS(E+Q))) THEN
            E = D
            D = P/Q
          ELSE
            D = XM
            E = D
          END IF
        ELSE
          D = XM
          E = D
        END IF
        A = B
        FA = FB
        IF (ABS(D).GT.TL1) THEN
          B = B+D
        ELSE
          B = B+SIGN(TL1,XM)
        END IF
c       CALL FNC (B,TM,V2)
        call satura (b,eta,xlamb,swr,spr,sprb,osig,owsig,wrho,prho,
     *                   hceao,hceow,v2,somax)
        FB = V2-SV
 1    CONTINUE
      STOP 'MBISEC: FAIL'
 20   continue
c
c     ***calculate the value of so
      do = vm
      bo = zm
      somax = do/eta/bo
c
c
      write (*,9400) do,somax
c
 9400 format (/2x,'Specified avg. NAPL thickness in the formation = ',
     *             f10.4,' (m)'/
     *         2x,'NAPL lens saturation                            = ',
     *             f10.4,' (*)'//)
      RETURN
      END
```

```
      SUBROUTINE MEPS (EPS)
C     ****************************************************************
C     *
C     * MEPS   MACHINE FLOATING POINT PRECISION
C     *        SEE  FORSYTHE, MALCOLM AND MOLER, PG.14
C     *
C     * INPUT ARGUEMENTS......(NONE).................................
C     * OUTPUT ARGUMENT..............................................
C     *    EPS   ESTIMATE OF MACHINE PRECISION
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  INITIA
C     *
C     ****************************************************************
      IMPLICIT DOUBLE PRECISION (A-H), DOUBLE PRECISION (O-Z)

C
      EPS = 1.
 10   CONTINUE
        EPS = 0.5*EPS
        EP1 = EPS + 1.
      IF (EP1.GT.1) GO TO 10
C
      RETURN
      END
```

3.4 Source Code for RAOULT

```
      program raoult
c     ****************************************************************
c     *
c     * raoult
c     *
c     * raoult uses raoults law and an approximate gasoline composition
c     * to estimate the gasoline/water partition coefficient
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma  74820
c     * 9-19-93
c     *
c     * double precision ansi std. x3.9-1978 FORTRAN 77
c     * required routines: bufchk,datachg,dataout,indata,readfl
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
c
c
      character*1 ans,chg
      character*20 name(200)
      dimension co(200),omega(200),sol(200),gamma(200)
c
c     ***exit raoult flag
      ans = 'N'
c     ***change the data flag
      chg = 'Y'
c
c
c
c     ***read the input data
      call indata (name,co,sol,omega,gamma,ncomp)
c
c
c
c     ***perform multiple calculations of the partition coefficient
      do 1000 while (ans.eq.'N'.or.ans.eq.'n')
c
c       ***unit conversions
c       ***miligrams per gram
        ucmgpg = 1000.
c       ***cubic centimeters per liter
        ucccpl = 1000.
c
        write (*,9100)
c
```

```
      chg = 'Y'
      do 55 while (chg.eq.'Y'.or.chg.eq.'y')
c
c       ***write data to screen
        call dataout (name,sol,co,omega,gamma,ncomp)

c
c       ***prompt for data change
        write (*,*)
        write (*,*) 'Change the input data ? (Y or N)'
        read (*,'(a)') chg
c
        if (chg.eq.'Y'.or.chg.eq.'y') then
c         ***change the data
          call datachg (name,sol,co,omega,gamma,ncomp)
        end if
c
 55     continue
c
c       ***calculate the gasoline density
        rho = 0.

        do 60 j=1,ncomp
          rho = rho + co(j)
 60     continue
        write (*,9115) rho
c
c       ***calculate weighted concentration and average molecular weight
        sum = 0.
        avgmw = 0.
        avgco = 0.
        do 100 j=1,ncomp
          if (omega(j).ne.0.) then
            sum = sum + co(j)/omega(j)
          end if
          avgmw = avgmw + omega(j)*co(j)
          avgco = avgco + co(j)
 100    continue
        if (avgco.ne.0.) then
          avgmw = avgmw/avgco
        end if
        write (*,9120) avgmw
c
c
        write (*,*) ' '
        write (*,*) ' '
        write (*,*) 'Select constituent of interest by number'
        read (*,*) k


c
c
        if (sol(k).ne.0.) then
          xko = omega(k)*sum/sol(k)/gamma(k)
          xko = xko*ucmgpg*ucccpl
c         ***solubility in moles per liter (smolpl)
          if (omega(k).ne.0.) then
            smolpl = sol(k)/omega(k)
            xko2 = rho/avgmw/smolpl
            xko2 = xko2*ucccpl*ucmgpg
          end if
          write (*,9150) xko,xko2
        else
          write (*,9155)
        end if
c
c
c       ***allow another calculation
        write (*,*) ' '
        write (*,*) 'Exit ? (Y or N)'
        read (*,'(a)') ans
c
        close (3)
c
 1000  end do
c
c
       write (*,*) ' '
       write (*,*) ' '
c
c
       stop '        *** sucessful execution of raoult'
c
c
c
 9100 format (/
     *       5x,50(1h*)/
     *       5x,'Raoults Law Partitioning Calculation'/
     *       5x,50(1h*)//)
 9115 format (/5x,'Hydrocarbon density   = ',f10.4)
 9120 format ( 5x,'Avg. Molecular Weight = ',f10.4)
 9150 format (///5x,'Calculated Hydrocarbon/Water ',
     *              'Partition Coefficient:'/
     *           5x,'Composition basis:'/
     *           5x,f10.4/
     *           5x,'Average molecular weight basis:'/
     *           5x,f10.4/)
 9155 format (///5x,'Consituent Solubility not in data base:'/
     *           5x,'No calculation possible')
      end
```

```
      subroutine indata (name,co,sol,omega,gamma,ncomp)
c     ************************************************************
c     *
c     *  indata  input the data for the raoult program
c     *
c     *
c     *
c     * input arguments..............(none).......................
c     * output arguments.........................................
c     *   name()  chemical names
c     *   co()    concentration
c     *   sol()   solubility
c     *   omega() molecular weight
c     *   gamma() activity coefficient
c     *   ncomp   number of chemicals
c     *
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * double precision ansi std x3.9-1978 fortran 77
c     * required routines   readfl
c     * called by  raoult
c     *
c     ************************************************************
      implicit double precision (a-h), double precision (o-z)
c
c
      character*20 name(200)
      dimension co(200),omega(200),sol(200),gamma(200)
c
c
c     ***open file raoult.dat
      open (3,file='raoult.dat',status='OLD')
c
c     ***read the number of data items
      read (3,*) ncomp
c
c
c     ***read the data items as mixed character and number strings
      call readfl (ncomp,name,sol,co,omega,gamma)
c
      return
      end
```

```
      subroutine readfl (ncomp,name,sol,co,omega,gamma)
c     ****************************************************************
c     *
c     *  readfl   reads the mixed character and real data from raoult.dat
c     *
c     *
c     *
c     * input arguments.............................................
c     * output arguments............................................
c     *   name()  chemical names
c     *   co()    concentration
c     *   sol()   solubility
c     *   omega() molecular weight
c     *   gamma() activity coefficient
c     *   ncomp   number of chemicals
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * double precision ansi std x3.9-1978 fortran 77
c     * required routines  bufchk
c     * called by  indata
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
      character buffer*80
      character*20 tname,name(*)
      dimension sol(*),co(*),omega(*),gamma(*)
      dimension num(10)
c
c
c
c     ***these data are extracted from the string,rather than
c     ***being read directly due to problems in reading
c     ***free format input from lines with both character and
c     ***integer data
      do 300 j=1,ncomp
        iv = 0
c
c       ***read a line
        read (3,'(a)') buffer
c
        tname = ' '
        inlen = 0
        ichar = 0
        icont = 0
        ideci = 0
        nnumb = 0
        do 10 i=1,80
c
c         ***get the next character
          call bufchk (buffer(i:i),itype,ivar)
c
c
c         ***the chemical name lies between
c         ***the first nonblank character and the first blank
          if (inlen.gt.0) then
c           ***inlen>0 when characters have been found
            if (itype.eq.0.and.ichar.eq.0) then
c             ***ends the accumulation of characters for the chem. name
              ichar = 1
              name(j) = tname
            end if
          end if
c
          if (ichar.eq.0.and.itype.ne.0) then
            inlen = inlen + 1
c           ***accumulate the characters of the chemical name
            tname(inlen:inlen) = buffer(i:i)
          end if
c
c
c
c         ***construct the real numbers
          if (ichar.ge.1) then
c
            if (istart.eq.0.and.
     *          (itype.eq.2.or.itype.eq.3)) then
c             ***start accumulating the number
              icont = 0
              if (itype.eq.2) then
                nnumb = nnumb + 1
              end if
              istart = 1
            end if
c
c
c
c
c
            if (itype.eq.0.and.istart.eq.1) then
c             ***end the accumulation of the number
              istart = 0
c
c
c             ***process the number
              rnumb = 0.
              do 5 ii=1,icont
                rnumb = rnumb + float(num(ii))*(10.**(ideci-ii))
 5            continue
c
c
c             ***store the number
              if (nnumb.eq.1) then
                co(j) = rnumb
              else if (nnumb.eq.2) then
                sol(j)  = rnumb
              else if (nnumb.eq.3) then
```

```
                  omega(j) = rnumb
                else if (nnumb.eq.4) then
                  gamma(j) = rnumb
                  go to 300
                end if
c
              end if
c
c
c
c
              if (istart.eq.1) then
c               ***accumulate the location of the decimal point
c               ***and the digits of the number
c               ***if turned on by istart
                if (itype.eq.3) then
c                 ***location of the decimal point
c                 ***the decimal is after the numeral number icont
                  ideci = icont
                else if (itype.eq.2) then
c                 ***the string is a number that is not in the chemical
name
                  icont = icont + 1
                  if (icont.eq.11) then
                    stop 'input real number has too many digits'
                  end if
                  num(icont) = ivar
                end if
              end if
c
c
           end if
c
 10     continue
c
 300  continue
c
c
c
c
      return
      end
```

```
      subroutine bufchk (char1,itype,ivar)
C     ****************************************************************
C     *
c     * bufchk  checks the type (character,number,decimal point) of a
c     *         singel character
c     *
c     *
c     * input arguments.................................................
c     *   char1     a single character
c     * output arguments................................................
c     *   itype    character type
c     *            0 = blank (delimeter)
c     *            1 = character
c     *            2 = integer number
c     *            3 = decimal point
c     *   iver     an integer if itype = 2
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * single precision ansi std x3.9-1978 fortran 77
c     * required routines  none
c     * called by  readfl
c     *
c     ****************************************************************
      character*1 char1
c
c
      ivar = 0
c
c     ***itype = 1 for a character
      itype = 1
c
      if (char1.eq.' ') then
c       ***do nothing for a blank
        itype = 0
        return
c
c
c     ***decimal point
      else if (char1.eq.'.') then
        itype = 3
c     ***numbers
      else if (char1.eq.'0') then
        itype = 2
        ivar = 0
      else if (char1.eq.'1') then
        itype = 2
        ivar = 1
      else if (char1.eq.'2') then
        itype = 2
        ivar = 2
      else if (char1.eq.'3') then
        itype = 2
        ivar = 3
      else if (char1.eq.'4') then
        itype = 2
        ivar = 4
      else if (char1.eq.'5') then
        itype = 2
        ivar = 5
      else if (char1.eq.'6') then
        itype = 2
        ivar = 6
      else if (char1.eq.'7') then
        itype = 2
        ivar = 7
      else if (char1.eq.'8') then
        itype = 2
        ivar = 8
      else if (char1.eq.'9') then
        itype = 2
        ivar = 9
      end if
c
c
      return
      end
```

```
      subroutine datachg (name,sol,co,omega,gamma,ncomp)
c     ****************************************************************
c     *
c     *  datachg   changes the chemical data
c     *
c     *
c     *
c     * input arguments.........................................
c     *   name()  chemical names
c     *   co()    concentration
c     *   sol()   solubility
c     *   omega() molecular weight
c     *   gamma() activity coefficient
c     *   ncomp   number of chemicals
c     * output arguments.......(none)..............................
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * double precision ansi std x3.9-1978 fortran 77
c     * required routines   none
c     * called by  raoult
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
      character*1 answ
      character*20 name(*)
      dimension sol(*),co(*),omega(*),gamma(*)
c
 10   continue
      write (*,*) ' '
      write (*,*) 'Select item to change by number'
      read (*,*) item
c
      if (item.lt.1.or.item.gt.ncomp) then
        write (*,*) 'Selection out of range'
        go to 10
      end if
c
      answ = 'Y'
c
      do 60 while (answ.eq.'Y'.or.answ.eq.'y')
c
        write (*,*) ' '
        write (*,*) 'Select data item to change'
        write (*,*) '1  Name'
        write (*,*) '2  Solubility'
        write (*,*) '3  Concentration'
        write (*,*) '4  Molecular Weight'
        write (*,*) '5  Activity Coefficient'
        read (*,*) isel
        write (*,*) ' '
c
        if (isel.eq.1) then
          write (*,*) 'Enter the new name'
          read (*,'(a20)') name(item)
        else if (isel.eq.2) then
          write (*,*) 'Enter the new solubility in mg/L'
          read (*,*) sol(item)
        else if (isel.eq.3) then
          write (*,*) 'Enter the new concentration '
          read (*,*) co(item)
        else if (isel.eq.4) then
          write (*,*) 'Enter the new molecular weight'
          read (*,*) omega(item)
        else if (isel.eq.5) then
          write (*,*) 'Enter the new activity coefficient'
          read (*,*) gamma(item)
        else
          write (*,*) 'Selection out of range'
        end if
c
        write (*,*) ' '
        write (*,*) 'Change another data item? (Y or N)'
        read (*,'(a)') answ
c
 60   continue
c
c
      return
      end
```

```
      subroutine dataout (name,sol,co,omega,gamma,ncomp)
c     ****************************************************************
c     *
c     *  dataout   writes chemical data to the screen
c     *
c     *
c     *
c     * input arguments.........................................
c     *   name()  chemical names
c     *   co()    concentration
c     *   sol()   solubility
c     *   omega() molecular weight
c     *   gamma() activity coefficient
c     *   ncomp   number of chemicals
c     * output arguments..........(none)............................
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * double precision ansi std x3.9-1978 fortran 77
c     * required routines  none
c     * called by   raoult,datachg
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
c
      character*20 name(*)
      dimension sol(*),co(*),omega(*),gamma(*)
c
c
c
      write (*,9105)
      do 50 j=1,ncomp
        write (*,9110) j,name(j),sol(j),co(j),omega(j),gamma(j)
 50   continue
c
c
c
c
      return
 9105 format(5x,'Chemical             ','Solubility ','Conc.      ',
     *             'Molecular ',1x,'Activity '/
     *       5x,'                    ','  (mg/L)   ','(g/cm3)    ',
     *             'Weight    ',1x,'Coefficient'/
     *       5x,20(1h=),4(1x,10(1h=)))
 9110 format (1x,i3,1x,a20,4(1x,f10.4))
      end
```

```
9
benzene          0.0082         1750.    78.    1.
toluene          0.0426          535.    92.    1.
xylenes          0.0718          167.   106.    1.
1-hexene         0.0159            0.    84.    1.
cyclohexane      0.0021            0.    84.    1.
n-hexane         0.0204            0.    86.    1.
other_aromatics  0.0740            0.   106.    1.
other_parafins   0.3367            0.   97.2    1.
heavy_ends       0.1451            0.   128.    1.
```

3.5 Source Code for REBUILD

```
      PROGRAM HSSMR
C     ****************************************************************
C     *
C     *HSSMR  CONCANTENATES THE HSSM *.TMP FILES WHEN AN ERROR HAS
C     *       OCURRED IN RUNNING THE PROGRAM
C     *       HSSMR CREATES THE *.HSS FILE FOR THE USER TO CHECK
C     *       THE MODEL INPUT AND RESULTS
C     *       HSSMR CREATES THE *.TSG FILE FOR THE USER TO CHECK
C     *       THE MODEL INPUT AND RESULTS
C     *
C     *
C     *
C     *       JIM WEAVER
C     *       ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *       UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     *       P.O. BOX 1198
C     *       ADA, OKLAHOMA    74820
C     *
C     *       SINGLE PRECISION FORTRAN 77
C     *       REQUIRED ROUTINES MESSAG,GENMES,CHKHSS,CHKTSG,CHKPLOT,DEL2,
C     *                        IOPOST,TSGP2,PFILE,RP18,RP19,PFILE2,PKCON,
C     *                            DEL2,CMD,DIR
C     *       IIX-10-92
C     *       JANUARY 14, 1993   RECOVERY OF *.TSG FILE, MARK LEE
C     *
C     ****************************************************************
C
C     ******************************
C     ***CALLS THE SUBROUTINE MESSAG
C     ***AND DISPLAYS OPENING MESSAGE.
C     ******************************
      CALL MESSAG (1)
C
C     ***GENERATE THE WARNING MESSAGE FOR THE *.HSS FILE
      CALL GENMES
C
C     ******************************
C     ***CALLS THE SUBROUTINE CHKHSS.
C     ******************************
      CALL CHKHSS ()
C
C     ******************************
C     ***CALLS THE SUBROUTINE CHKTSG.
C     ******************************
      CALL CHKTSG ()
C
C     ******************************
C     ***CALLS THE SUBROUTINE CHKPLOT.
C     ******************************
      CALL CHKPLOT ()
C
C     ******************************
C     ***CALLS THE SUBROUTINE DEL2.
C     ***
C     ***NOT IN USE AT PRESENT TO ALLOW
C     *** FOR EXAMINATION OF TEMP FILES.
C     ******************************
      CALL DEL2
c
      STOP '         *** REBUILD ENDED'
      END
```

```
      SUBROUTINE GENMES
C     ****************************************************************
C     *
C     *  GENMES   GENERATES A WARNING MESSAGE THAT THE FILE WAS
C     *           PRODUCED BY HSSMR
C     *
C     *
C     *
C     * INPUT ARGUMENTS.........(NONE).............................
C     * OUTPUT ARGUMENTS........(NONE).............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION FORTRAN 77
C     * REQUIRED ROUTINES  NONE
C     * CALLED BY  HSSMR
C     *
C     ****************************************************************
      LOGICAL FEXIST
C
C     ***GENERATES A WARNING MESSAGE IN FILE TWA.TMP
C     ***WARNING NOTICE
      INQUIRE (FILE='TWA.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (44,FILE='TWA.TMP',STATUS='UNKNOWN')
        CLOSE (44,STATUS='DELETE')
      END IF
C
C     ***OPEN THE FILE
      OPEN (44,FILE='TWA.TMP',STATUS='UNKNOWN')
C
C     ***WRITE THE MESSAGE
      WRITE (44,9100)
C
C     ***CLOSE THE FILE
      CLOSE (44)
C
C
 9100 FORMAT (//10X,50(1H*)/
     *          10X,'THIS FILE WAS PRODUCED BY REBUILD.EXE'/
     *          10X,'FROM AN INTERRUPTED HSSM RUN'/
     *          10X,'THE FILE MAY BE INCOMPLETE OR DISORDERED'/
     *          10X,50(1H*)///)
      RETURN
      END
```

```
      SUBROUTINE CHKHSS ()
C
C*********************************************************************
C     ***THIS SUBROUTINE CHECKS FOR THE TEMP FILES NEEDED TO CREATE THE
C     ****.HSS FILE. THE *.HSS FILE IS THEN CHECKED FOR AND CREATED IF IT
C     ***DOES NOT EXIST.
C     ***
C     ***REQUIRED ROUTINES:   IOPOST
C     ***THIS ROUTINE IS CALLED BY:   PROGRAM HSSMR
C
C*********************************************************************
C
      LOGICAL LEXIST
      CHARACTER*40 OFILE
      CHARACTER*16 S1
      CHARACTER*19 S2
C
C     ***CHECK FOR EXISTENCE OF THE *.HSS OUTPUT FILES
      IE = 0
      INQUIRE (FILE='T15.TMP',EXIST=LEXIST)
      IF (LEXIST.EQV..TRUE.) THEN
        IE = IE + 1
      ELSE IF (LEXIST.EQV..FALSE.) THEN
        IE = IE
        CALL MESSAG (9)
        RETURN
      END IF
C
C     ***FILE T15.TMP IS THE MAIN FILE OF INTEREST FOR *.HSS
      IF (IE.NE.0) THEN
C       ***CHECK FOR THE EXISTENCE OF THE OTHER *.TMP FILES
        INQUIRE (FILE='T11.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IE = IE + 1
        INQUIRE (FILE='T12.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IE = IE + 1
        INQUIRE (FILE='T13.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IE = IE + 1
        INQUIRE (FILE='T7.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IE = IE + 1
        INQUIRE (FILE='T9.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IE = IE + 1
        INQUIRE (FILE='T10.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IE = IE + 1
        INQUIRE (FILE='T8.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IE = IE + 1
C
C       ***READ THE OUTPUT FILE NAME FROM FILE T15.TMP
        IF (IE.GT.1) THEN
          OPEN (15,FILE='T15.TMP',STATUS='OLD')
 10       CONTINUE
            READ (15,9100,ERR=19,END=19) S1,S2
            IF (S2.EQ.'HSSM-KO OUTPUT:') THEN
              BACKSPACE (15)
              READ (15,9110) OFILE
              CALL MESS3 (8,OFILE)
              GO TO 20
            END IF
            GO TO 10
 19         CONTINUE
            STOP 'ERROR IN FILE NAME SEARCH (*.HSS)'
 20         CONTINUE
C       ***DELETE EXISTING OUTPUT FILE CONTAINING THE NAME
        INQUIRE (FILE=OFILE,EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) THEN
          CALL MESS3 (12,OFILE)
        ELSE
C
C         ***CREATE THE OUTPUT FILE WITH AS MANY OF THE TEMPORARY FILES
AS
C         ***POSSIBLE
          IWR = 1
          IREO = 1
          IF (IE.EQ.1) IREO = 0
            CALL IOPOST (OFILE,IWR,IREO)
          END IF
        END IF
 9100 FORMAT (A16,A15)
 9110 FORMAT (35X,A40)
      RETURN
      END
```

```
      SUBROUTINE CHKTSG ()                                                    
C     ***************************************************************     
C     ***THIS SUBROUTINE CHECKS FOR THE TEMP FILES NEEDED TO CREATE THE      
C     ****.TSG FILE. THE *.TSG FILE IS THEN CHECKED FOR AND CREATED IF IT    
C     ***DOES NOT EXIST.                                                    
C     ***                                                                   
C     ***REQUIRED ROUTINES:   TSGP2                                         
C     ***THIS ROUTINE IS CALLED BY:   PROGRAM HSSMR                         
C     ***************************************************************     
C                                                                           
      LOGICAL LEXIST                                                        
      CHARACTER*40 TFILE                                                    
      CHARACTER*3 S3                                                        
      CHARACTER*15 S4                                                       
                                                                            
C     ***CHECK FOR THE EXISTENCE OF THE *.TSG FILE                          
      ID = 0                                                                
      INQUIRE (FILE='T42.TMP',EXIST=LEXIST)                                 
      IF (LEXIST.EQV..TRUE.) THEN                                           
        ID = ID + 1                                                         
      ELSE IF (LEXIST.EQV..FALSE.) THEN                                     
        ID = ID                                                             
        CALL MESSAG (10)                                                    
        RETURN                                                              
      END IF                                                                
C                                                                           
C     ***READ THE OUTPUT FILE NAME FROM FILE T42.TMP                        
      IF (ID.EQ.1) THEN                                                     
C       ***RECOVER THE *.TSG FILE                                           
        OPEN (42,FILE='T42.TMP',STATUS='OLD')                               
  40     CONTINUE                                                           
          READ (42,9120,ERR=49,END=49) S3,S4                                
          IF (S4.EQ.'HSSM-T OUTPUT') THEN                                   
            BACKSPACE (42)                                                  
            READ (42,9130) TFILE                                            
            CALL MESS3 (8,TFILE)                                            
            GO TO 50                                                        
          END IF                                                            
          GO TO 40                                                          
  49       CONTINUE                                                         
          STOP 'ERROR IN FILE NAME SEARCH (*.TSG)'                          
  50       CONTINUE                                                         
C     ***DELETE EXISTING OUTPUT FILE CONTAINING THE NAME                    
      INQUIRE (FILE=TFILE,EXIST=LEXIST)                                     
      IF (LEXIST.EQV..TRUE.) THEN                                           
      CALL MESS3 (12,TFILE)                                                 
      ELSE                                                                  
C                                                                           
C       ***CREATE THE TSGPLUME FILE IF POSSIBLE                             
        IWR = 1                                                             
        IREO = 1                                                            
        CALL TSGP2 (TFILE,IWR,IREO)                                         
      END IF                                                                
      END IF                                                                
C                                                                           
      RETURN                                                                
 9120 FORMAT (A3,A13)                                                       
 9130 FORMAT (31x,A40)                                                      
      END                                                                   
```

```
      SUBROUTINE IOPOST (OFILE,IWR,IREO)
C     ******************************************************************
C     ***
C     ***THIS SUBROUTINE CONCATINATES THE DESIGNATED TEMP FILES INTO THE
C     ***OUTPUT FILE *.HSS
C     ***
C     ***REQUIRED ROUTINES:
C     ***THIS ROUTINE IS CALLED BY:   SUBROUTINE CHKHSS
C     ******************************************************************
C
      CHARACTER*40 OFILE
      CHARACTER*256 STR
      INTEGER IN(8)
      INTEGER*4 I4
      DATA IN /15,11,12,13,7,9,10,8/
      CALL MESS3 (3,OFILE)
C
C     ***CONCANTINATE OUTPUT FILES
      IF (IWR.EQ.0) RETURN
C
      DO 5 I=7,15
        REWIND I
  5   CONTINUE
C
      IEND = 8
c     STR = 'COPY TWA.TMP+T15.TMP+T11.TMP+T12.TMP+T13.TMP+T7.TMP+'//
c    *'T9.TMP+T10.TMP+T8.TMP '//OFILE//' > OFILE'

      STR = 'COPY T15.TMP+T11.TMP+T12.TMP+T13.TMP+T7.TMP+T9.TMP+'//
     *'T10.TMP+T8.TMP '//OFILE//' > OFILE'

      IF (IREO.EQ.0) THEN
        IEND = 1
        STR = 'COPY T15.TMP '//OFILE//' > OFILE'
      END IF
C
C     ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
      DO 30 I=1,IEND
        CLOSE (IN(I))
 30   CONTINUE
C
C     ***EXECUTE OPERATING SYSTEM COMMAND, STR
C     ***RETURNS ERROR CODE, I4
      CALL CMD (STR,I4)
C
      RETURN
      END
C
```

```
      SUBROUTINE TSGP2 (TFILE,IWR,IREO)
C     ******************************************************************
C     ***
C     ***THIS SUBROUTINE CONCATINATES THE DESIGNATED TEMP FILES INTO THE
C     ***OUTPUT FILE *.TSG
C     ***
C     ***REQUIRED ROUTINES:
C     ***THIS ROUTINE IS CALLED BY:   SUBROUTINE CHKTSG
C     ******************************************************************
C
      CHARACTER*40 TFILE
      CHARACTER*256 STR
      INTEGER*4 I4
C
C     ***CONCANTINATE OUTPUT FILES
      CALL MESS3 (4,TFILE)
      IF (IWR.EQ.0.OR.IREO.EQ.0) RETURN
C
      REWIND 42
C
      IEND = 7
      STR = 'COPY T42.TMP '//TFILE//' > TFILE'
C
C     ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
      CLOSE (42)
C
C     ***EXECUTE OPERATING SYSTEM COMMAND, STR
C     ***RETURNS ERROR CODE, I4
      CALL CMD (STR,I4)
C
      RETURN
      END
C     ******************************************************************
C
```

```
      SUBROUTINE CHKPLOT ()
C     ******************************************************************
C     ***THIS SUBROUTINE CHECKS FOR THE TEMP FILES NEEDED TO CREATE THE
C     ****.PL* FILES. THE *.PL* FILES ARE THEN CHECKED FOR AND CREATED IF
C     ***THEY DO NOT EXIST.
C     ***
C     ***REQUIRED ROUTINES:   PFILE
C     ***THIS ROUTINE IS CALLED BY:   PROGRAM HSSMR
C     ******************************************************************
      LOGICAL LEXIST
      CHARACTER*40 OFILE1
      CHARACTER*40 OFILE2
      CHARACTER*40 OFILE3
      CHARACTER*16 S1
      CHARACTER*19 S2
C
C     ***CHECK FOR EXISTENCE OF THE *.PL* OUTPUT FILES
      IPT = 0
      INQUIRE (FILE='T20.TMP',EXIST=LEXIST)
      IF (LEXIST.EQV..TRUE.) THEN
          IPT = IPT + 1
      ELSE IF (LEXIST.EQV..FALSE.) THEN
        IPT = 0
        CALL MESSAG (11)
        RETURN
      END IF
C
C     ***FILE T15.TMP IS THE MAIN FILE OF INTEREST FOR *.PL*
      IF (IPT.NE.0) THEN
C       ***CHECK FOR THE EXISTENCE OF THE OTHER *.TMP FILES
        INQUIRE (FILE='T16.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IPT = IPT + 1
        INQUIRE (FILE='T17.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IPT = IPT + 1
        INQUIRE (FILE='T18.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IPT = IPT + 1
        INQUIRE (FILE='T19.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IPT = IPT + 1
        INQUIRE (FILE='T20.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IPT = IPT + 1
        INQUIRE (FILE='T21.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.) IPT = IPT + 1
        INQUIRE (FILE='T22.TMP',EXIST=LEXIST)
        IF (LEXIST.EQV..TRUE.)  IPT = IPT + 1
C
C
C       ***READ THE OUTPUT FILE NAME FROM FILE T15.TMP
        IF (IPT.GT.1) THEN
          OPEN (15,FILE='T15.TMP',STATUS='OLD')
 2710       CONTINUE
            READ (15,9100,ERR=2719,END=2719) S1,S2
            IF (S2.EQ.'HSSM-KO PLOT 1:') THEN
              BACKSPACE (15)
              READ (15,9110) OFILE1
              CALL MESS3 (8,OFILE1)
```

```
              GO TO 2720
            END IF
            GO TO 2710
 2719         CONTINUE
            STOP 'ERROR IN FILE NAME SEARCH (*.PL1)'
 2720         CONTINUE
          INQUIRE (FILE=OFILE1,EXIST=LEXIST)
          IF (LEXIST.EQV..TRUE.) THEN
          CALL MESS3 (12,OFILE1)
          ELSE
            CONTINUE
          END IF
C     ***********************************************
C       ***READ THE OUTPUT FILE NAME FROM FILE T15.TMP
          IF (IPT.GT.1) THEN
            OPEN (15,FILE='T15.TMP',STATUS='OLD')
 2810       CONTINUE
            READ (15,9100,ERR=2819,END=2819) S1,S2
            IF (S2.EQ.'HSSM-KO PLOT 2:') THEN
              BACKSPACE (15)
              READ (15,9110) OFILE2
              CALL MESS3 (8,OFILE2)
              GO TO 2820
            END IF
            GO TO 2810
 2819         CONTINUE
            STOP 'ERROR IN FILE NAME SEARCH (*.PL2)'
 2820         CONTINUE
          END IF
          INQUIRE (FILE=OFILE2,EXIST=LEXIST)
          IF (LEXIST.EQV..TRUE.) THEN
          CALL MESS3 (12,OFILE2)
          ELSE
            CONTINUE
          END IF
C     ***********************************************
C       ***READ THE OUTPUT FILE NAME FROM FILE T15.TMP
          IF (IPT.GT.1) THEN
            OPEN (15,FILE='T15.TMP',STATUS='OLD')
 2910       CONTINUE
            READ (15,9100,ERR=2919,END=2919) S1,S2
            IF (S2.EQ.'HSSM-KO PLOT 3:') THEN
              BACKSPACE (15)
              READ (15,9110) OFILE3
              CALL MESS3 (8,OFILE3)
              GO TO 2920
            END IF
            GO TO 2910
 2919         CONTINUE
            STOP 'ERROR IN FILE NAME SEARCH (*.PL3)'
 2920         CONTINUE
          END IF
C       ***DELETE EXISTING OUTPUT FILE CONTAINING THE NAME
          INQUIRE (FILE=OFILE3,EXIST=LEXIST)
          IF (LEXIST.EQV..TRUE.) THEN
          CALL MESS3 (12,OFILE3)
          ELSE
            CONTINUE
          END IF
C
            INQUIRE (FILE=OFILE1,EXIST=LEXIST)
            IF (LEXIST.EQV..TRUE.) THEN
              INQUIRE (FILE=OFILE2,EXIST=LEXIST)
              IF (LEXIST.EQV..TRUE.) THEN
                INQUIRE (FILE=OFILE3,EXIST=LEXIST)
                IF (LEXIST.EQV..TRUE.) THEN
                  RETURN
                END IF
              ELSE
                GO TO 2950
              END IF
            END IF
C     ********************************************************************
C         ***CREATE THE OUTPUT FILE WITH AS MANY OF THE TEMPORARY FILES
AS
C         ***POSSIBLE
 2950     CONTINUE
          IWR = 1
          IREO = 1
          IF (IPT.EQ.1) IREO = 0
            CALL PFILE (OFILE1,OFILE2,OFILE3)
          END IF
      END IF
 9100 FORMAT (A16,A15)
 9110 FORMAT (35X,A40)
      RETURN
      END
C
C
```

```
      SUBROUTINE PFILE (OFILE1,OFILE2,OFILE3)
C     ***************************************************************
C     *
C     *  PFILE REPACKS FILES T18.TMP AND T19.TMP IF THEY ARE
C     *  TOO LARGE.  IT ALSO OPENS AND CREATES THE 3 PLOT FILES.
C     *
C     *  INPUT ARGUMENTS.....................................
C     *  OFILE1  CHARACTER*40 *.PL1 FILE NAME
C     *  OFILE2  CHARACTER*40 *.PL2 FILE NAME
C     *  OFILE3  CHARACTER*40 *.PL3 FILE NAME
C     *
C     *  OUTPUT ARGUMENTS: NONE
C     *
C     *  JIM WEAVER
C     *  ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     *
C     *  SINGLE PRECISION, ANSI STD. X3.9-1978 FORTRAN 77
C     *  REQUIRED ROUTINES:  PFILE2,PKCON,RP18,RP19
C     *  THIS ROUTINE IS CALLED BY:  SUBROUTINE CHKPLOT
C     ***************************************************************
      CHARACTER*40 OFILE1,OFILE2,OFILE3,NT*15
      COMMON /NAME/ NT(15)
      COMMON /FLAG/ ICONC,IKOPT,ILENS,ITSGP
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
C     ***CONTROL MODULE***
C     ***REPACK FILES T18.TMP AND T19.TMP IF THEY ARE TOO LARGE
      IF (ILENS.EQ.1) THEN
      CALL RP18
      CALL RP19
      END IF
C
C     **OPEN FILES WITH HEADINGS FOR THE THREE PLOT FILES
      CALL PFILE2
C
C     ***CREATE KOPT/OILENS PLOT FILES FOR HSSM-WIN PROGRAM
C     ***NONSTANDARD CONCATENATION ROUTINE FOR CREATING OUTPUT FILE
C     ***FROM THE HEADING FILES AND THE PREVIOUSLY OUTPUTED DATA
      CALL PKCON (OFILE1,OFILE2,OFILE3,IWR)
C
      RETURN
      END
```

```
      SUBROUTINE RP18
C     ****************************************************************
C     ***REPACK FILE T18.TMP
C     ***IF NUMBER OF RECORDS > 75, REDUCE THE NUMBER OF RECORDS TO 51
C     ***INPUT
C     *** T1            INDEPENDENT REAL ARRAY TO BE REPACKED
C     *** Y1,Y2,Y3,Y4   DEPENDENT REAL ARRAYS TO BE REPACKED
C     ***
C     ***OUTPUT
C     *** T1            ADJUSTED INDEPENDENT REAL ARRAY
C     *** Y1,Y2,Y3,Y4   ADJUSTED DEPENDENT REAL ARRAYS
C     ***
C     ***WORK
C     ***  TT1,YY1,YY2,YY3,YY4       VARIABLES FOR CURRENT RECORD
C     ***  TT1P,YY1P,YY2P,YY3P,YY4P  VARIABLES IN PREVIOUS RECORD
C     ***  DT       NEW SPACING INCREMENT FOR INDEPENDENT VARIABLE (TIME)
C     ***  NREC     NUMBER OF RECORDS IN A FILE (INCLUDING TRAILER
C     ***           RECORD)
C     ***  SEOF     CHARACTER VARIABLE TO DETECT 'S' IN TRAILER RECORD
C     ***  NMAX     (NMAX+1) = NUMBER OF RECORDS FOR A FILE AFTER
C     ***           REPACKING
C     ***  IREC     RECORD NUMBER OF CURRENT RECORD
C     ***  INEW     POINTER TO NEXT AVAILABLE POSITION IN T1 ARRAY
C     ***
C     *REQUIRED ROUTINES:   NONE
C     *THIS ROUTINE IS CALLED BY:  PFILE
C     ****************************************************************
      DIMENSION T1(52),Y1(52)
      CHARACTER*4 SEOF
      CHARACTER ENDCHR*5
      ENDCHR = 'END  '
      ENDCHR(4:4) = CHAR(13)
      ENDCHR(5:5) = CHAR(10)
C
      NMAX = 50
C     ***ADD 'END' STATEMENT TO THE END OF FILE 18 FOR HSSM-WIN
      WRITE (18,420)
C     ***REPACK FILE 18
      CLOSE (18)
      OPEN(18,FILE='T18.TMP',STATUS='OLD')
C
C     ***CHECK FOR EMPTY FILE
      READ (18,9010) SEOF
      IF (SEOF.EQ.'END') THEN
      READ (18,9010) SEOF
      IF (SEOF.EQ.'BEGI') THEN
C         ***THERE IS NO DATA IN FILE 18
        RETURN
      END IF
      END IF
C
      REWIND 18
C
C     ***READ FIRST RECORD OF THE FILE
      READ(18,411) SEOF,T1(1),Y1(1)
C     ***READ THROUGH THE LAST DATA RECORD AND THE TRAILER RECORD
      READ(18,411) SEOF,TT1,YY1
      NREC = 2
5     CONTINUE
      IF(SEOF.NE.'END') THEN
      T1(NMAX) = TT1
      Y1(NMAX) = YY1
      NREC = NREC + 1
      READ(18,411) SEOF,TT1,YY1
      GO TO 5
      ENDIF
C
C     ***CHECK TO SEE IF FILE 18 NEEDS TO BE REPACKED
      IF(NREC.GT.1.5*NMAX) THEN
C       ***CALCULATE DT
       DT = (T1(NMAX) - T1(1))/FLOAT(NMAX - 1)
C       ***READ FIRST 2 RECORDS
       REWIND 18
       READ(18,412) TT1P,YY1P
       READ(18,412) TT1,YY1
       IREC = 2
       INEW = 2
C        ***LOOP FOR ALL THE NEW POINTS UP TO THE LAST POINT
10       CONTINUE
       IF(INEW.LT.NMAX) THEN
C          ***CALCULATE NEXT VALUE OF T1
         T1(INEW) = T1(INEW-1) + DT
C          ***WHILE LOOP TO SEARCH FOR TT1 WHICH IS JUST GREATER THAN
C          *** T1(INEW)
15         CONTINUE
         IF(TT1.LE.T1(INEW)) THEN
           TT1P=TT1
           YY1P=YY1
           READ(18,412) TT1,YY1
           IREC = IREC + 1
           GO TO 15
         END IF
C          ***CALCULATE NEXT VALUE OF Y1
         SL = (YY1 - YY1P)/(TT1 - TT1P)
         Y1(INEW) = YY1P + SL * (T1(INEW) - TT1P)
         INEW = INEW + 1
       GO TO 10
       END IF
C        ***WRITE THE NEW FILE 18 FROM T1 AND Y1 ARRAYS
       REWIND 18
       DO 90 I = 1,INEW
         WRITE(18,412) T1(I),Y1(I)
90       CONTINUE
       WRITE(18,420)
```

```
      END IF
C
      RETURN
411   FORMAT(A4,5(G10.4,1X))
412   FORMAT(4X,5(G10.5,1X))
419   FORMAT('BEGIN')
420   FORMAT('END'/'BEGIN')
 9002 FORMAT (A5)
 9010 FORMAT (A4)
      END
```

```
      SUBROUTINE RP19
C     ****************************************************************
C     ***REPACK FILE T19.TMP
C     ***IF NUMBER OF RECORDS > 75, REDUCE THE NUMBER OF RECORDS TO 51
C     ***INPUT
C     *** T1            INDEPENDENT REAL ARRAY TO BE REPACKED
C     *** Y1,Y2,Y3,Y4   DEPENDENT REAL ARRAYS TO BE REPACKED
C     ***
C     ***OUTPUT
C     *** T1            ADJUSTED INDEPENDENT REAL ARRAY
C     *** Y1,Y2,Y3,Y4   ADJUSTED DEPENDENT REAL ARRAYS
C     ***
C     ***WORK
C     ***  TT1,YY1,YY2,YY3,YY4       VARIABLES FOR CURRENT RECORD
C     ***  TT1P,YY1P,YY2P,YY3P,YY4P  VARIABLES IN PREVIOUS RECORD
C     ***  DT       NEW SPACING INCREMENT FOR INDEPENDENT VARIABLE (TIME)
C     ***  NREC     NUMBER OF RECORDS IN A FILE (INCLUDING TRAILER
C     ***            RECORD)
C     ***  SEOF     CHARACTER VARIABLE TO DETECT 'S' IN TRAILER RECORD
C     ***  NMAX     (NMAX+1) = NUMBER OF RECORDS FOR A FILE AFTER
C     ***            REPACKING
C     ***  IREC     RECORD NUMBER OF CURRENT RECORD
C     ***  INEW     POINTER TO NEXT AVAILABLE POSITION IN T1 ARRAY
C     ***
C     *REQUIRED ROUTINES:   NONE
C     *THIS ROUTINE IS CALLED BY:  PFILE
C     ****************************************************************
      DIMENSION T1(52),Y1(52),Y2(52),Y3(52),Y4(52)
      CHARACTER*4 SEOF
      CHARACTER ENDCHR*5
      ENDCHR = 'END  '
      ENDCHR(4:4) = CHAR(13)
      ENDCHR(5:5) = CHAR(10)
C
      NMAX = 50
C     ***REPACK FILE 19
C
C     ***ADD 'END' STATEMENT TO THE END OF FILE 19 FOR HSSM-WIN
      WRITE (19,9002) ENDCHR
      CLOSE (19)
      OPEN(19,FILE='T19.TMP',STATUS='OLD')
C
C     ***CHECK FOR EMPTY FILE
      READ (19,9010) SEOF
      IF (SEOF.EQ.'END') THEN
      READ (19,9010) SEOF
      IF (SEOF.EQ.'BEGI'.OR.SEOF.EQ.'    ') THEN
C         ***THERE IS NO DATA IN FILE 1
        RETURN
      END IF
      END IF
C
      REWIND 19
C
C     ***READ FIRST RECORD OF THE FILE
      READ(19,411) SEOF,T1(1),Y1(1),Y2(1),Y3(1),Y4(1)
C     ***READ THROUGH THE LAST DATA RECORD AND THE TRAILER RECORD
      READ(19,411) SEOF,TT1,YY1,YY2,YY3,YY4
      NREC = 2
      FMAX = 0.
      TMAX = 0.
50    CONTINUE
      IF(SEOF.NE.'END') THEN
      T1(NMAX) = TT1
      Y1(NMAX) = YY1
      Y2(NMAX) = YY2
      Y3(NMAX) = YY3
      Y4(NMAX) = YY4
      NREC = NREC + 1
      READ(19,411) SEOF,TT1,YY1,YY2,YY3,YY4
      IF (YY2.GT.FMAX) THEN
        FMAX = YY2
        TMAX = TT1
          IMAX = NREC
      END IF
      GO TO 50
      ENDIF
C
C     ***CHECK TO SEE IF FILE NEEDS TO BE REPACKED
      IF (NREC.GT.1.5*NMAX) THEN
C
C       ***DETERMINE THE TIME INTERVAL DT FROM THE START TO THE PEAK

      DT = (TMAX-T1(1))/5.
C
C       ***READ FIRST 2 RECORDS
      REWIND 19
      READ(19,412) TT1P,YY1P,YY2P,YY3P,YY4P
      READ(19,412) TT1,YY1,YY2,YY3,YY4
      IREC = 2
      INEW = 2

C       ***LOOP FOR ALL THE NEW POINTS UP TO THE LAST POINT
100     CONTINUE

      IF (INEW.LT.NMAX) THEN
C         ***RECALCULATE DT
        IF (T1(INEW-1).GE.TMAX) DT = (T1(NMAX)-TMAX)/FLOAT(NMAX-6)
C
C         ***CALCULATE NEXT VALUE OF T1
        T1(INEW) = T1(INEW-1) + DT
C
C         ***WHILE LOOP TO SEARCH FOR TT1 WHICH IS JUST GREATER THAN
T1(INEW)
```

```
150       CONTINUE

          IF(TT1.LE.T1(INEW)) THEN
            TT1P=TT1
            YY1P=YY1
            YY2P=YY2
            YY3P=YY3
            YY4P=YY4
            READ(19,412,end=1000) TT1,YY1,YY2,YY3,YY4
            IREC = IREC + 1
            GO TO 150
          END IF

C           ***CALCULATE NEXT VALUES OF Y2,Y3,Y4
          SL = (YY1 - YY1P)/(TT1 - TT1P)
          Y1(INEW) = YY1P + SL * (T1(INEW) - TT1P)
          SL = (YY2 - YY2P)/(TT1 - TT1P)
          Y2(INEW) = YY2P + SL * (T1(INEW) - TT1P)
          SL = (YY3 - YY3P)/(TT1 - TT1P)
          Y3(INEW) = YY3P + SL * (T1(INEW) - TT1P)
          SL = (YY4 - YY4P)/(TT1 - TT1P)
          Y4(INEW) = YY4P + SL * (T1(INEW) - TT1P)
          INEW = INEW + 1
          GO TO 100
        END IF
C
C         ***WRITE NEW FILE 19 FROM T1,Y1,Y2,Y3,Y4 ARRAYS
        REWIND 19
        DO 900 I = 1,INEW
          WRITE(19,412) T1(I),Y1(I),Y2(I),Y3(I),Y4(I)
900       CONTINUE
        END IF
C
        RETURN
 1000   CONTINUE
        WRITE (*,*) 'irec, tt1p',irec,tt1p
        RETURN
411     FORMAT(A4,5(G10.4,1X))
412     FORMAT(4X,5(G10.5,1X))
419     FORMAT('BEGIN')
420     FORMAT('END'/'BEGIN')
 9002   FORMAT (A5)
 9010   FORMAT (A4)
        END
```

```
      SUBROUTINE PFILE2
C     ****************************************************************
C     ***ROUTINE WILL CREATE KOPT/OILENS PLOT FILES FOR HSSM-WIN PROGRAM
C     ***REQUIRED ROUTINES:  NONE
C     ***THIS ROUTINE IS CALLED BY:  SUBROUTINE PFILE
C     ****************************************************************
      CHARACTER NT*15,ENDCHR*5,STOPCHR*5
      COMMON /NAME/ NT(15)
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      COMMON /WTER/ IRT,RKS,SWMAX,QW,WKS,XMKRW
C
C     ***OPEN FILES WITH HEADINGS FOR THE THREE PLOT FILES
      OPEN (22,FILE='T22.TMP',STATUS='UNKNOWN')
      OPEN (23,FILE='T23.TMP',STATUS='UNKNOWN')
      REWIND 22
      REWIND 23
C
      STOPCHR = 'STOP '
      STOPCHR(5:5) = CHAR(10)
      ENDCHR = 'END  '
      ENDCHR(4:4) = CHAR(13)
      ENDCHR(5:5) = CHAR(10)
      WRITE (22,9003) ENDCHR
      WRITE (23,9002) STOPCHR
C
      RETURN
 9000 FORMAT (5A10/5A10/5A10/F10.4/'BEGIN')
 9001 FORMAT (5A10/5A10/5A10/'BEGIN')
 9002 FORMAT (/A5)
 9003 FORMAT (/A5)
      END
```

```
      SUBROUTINE PKCON (OFILE1,OFILE2,OFILE3,IWR)
C     ****************************************************************
C     ***THIS SUBROUTINE CHECKS FOR THE TEMP FILES NEEDED TO CREATE THE
C     ****.PL* FILES.  THE *.PL* FILES ARE CREATED BY CONCATINATION OF
C     ***THE TEMP FILES NECESSARY FOR EACH OF THE THREE PLOT FILES.
C     ***
C     ***REQUIRED ROUTINES:   NONE
C     ***THIS ROUTINE IS CALLED BY:   SUBROUTINE PFILE
C     ****************************************************************
      CHARACTER*40 OFILE1
      CHARACTER*40 OFILE2
      CHARACTER*40 OFILE3
      CHARACTER*256 STR
      INTEGER*4 I4
      LOGICAL EXIST
      COMMON /FLAG/ ICONC,IKOPT,ILENS,ITSGP
C
C     ***CONCANTINATE OUTPUT FILES
C      IF (IWR.EQ.0) RETURN
C
      CALL MESS3 (5,OFILE1)
C     ***PLOT FILE 1 CONTAINING THE KOPT SATURATION/DEPTH PROFILES
      REWIND 20
      REWIND 16
      REWIND 23
      REWIND 22
C
C     ***CHECK FOR EXISTENCE OF THE OUTPUT FILES
      IPL = 0
      INQUIRE (FILE='T20.TMP',EXIST=EXIST)
      IF (EXIST.EQV..FALSE.) IPL = IPL + 1
      INQUIRE (FILE='T16.TMP',EXIST=EXIST)
      IF (EXIST.EQV..FALSE.) IPL = IPL + 1
      INQUIRE (FILE='T22.TMP',EXIST=EXIST)
      IF (EXIST.EQV..FALSE.) IPL = IPL + 1
      INQUIRE (FILE='T23.TMP',EXIST=EXIST)
      IF (EXIST.EQV..FALSE.) IPL = IPL + 1
C
      IF (IPL.EQ.0) THEN
       STR = 'COPY T20.TMP+T16.TMP+T23.TMP+T22.TMP'//OFILE1//' > OFILE1'
C
C       ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
        CLOSE (20)
        CLOSE (16)
        CLOSE (23)
        CLOSE (22)
C
C       ***EXECUTE OPERATING SYSTEM COMMAND, STR
C       ***RETURNS ERROR CODE, I4
        CALL CMD (STR,I4)
C
        STR = 'DEL OFILE1'
```

```
        CALL CMD  (STR,I4)
C
      END IF
C
C      IF (ILENS.EQ.1) THEN
        CALL MESS3 (6,OFILE2)
C       ***PLOT FILE 2 CONTAINING THE OILENS RADIAL PROFILES
        REWIND 21
        REWIND 17
        REWIND 23
        REWIND 22
C
C       ***CHECK FOR EXISTENCE OF THE OUTPUT FILES
        IPM = 0
        INQUIRE (FILE='T21.TMP',EXIST=EXIST)
        IF (EXIST.EQV..FALSE.) IPM = IPM + 1
        INQUIRE (FILE='T17.TMP',EXIST=EXIST)
        IF (EXIST.EQV..FALSE.) IPM = IPM + 1
        INQUIRE (FILE='T22.TMP',EXIST=EXIST)
        IF (EXIST.EQV..FALSE.) IPM = IPM + 1
        INQUIRE (FILE='T23.TMP',EXIST=EXIST)
        IF (EXIST.EQV..FALSE.) IPM = IPM + 1
C
      IF (IPM.EQ.0) THEN
      STR = 'COPY T21.TMP+T17.TMP+T23.TMP+T22.TMP'//OFILE2//' > OFILE2'
C
C         ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
          CLOSE (21)
          CLOSE (17)
          CLOSE (23)
          CLOSE (22)
C
C         ***EXECUTE OPERATING SYSTEM COMMAND, STR
C         ***RETURNS ERROR CODE, I4
          CALL CMD (STR,I4)
C
          STR = 'DEL OFILE2'
          CALL CMD (STR,I4)
C
        END IF
C
        CALL MESS3 (7,OFILE3)
C       ***PLOT FILE 3 CONTAINING THE OILENS RADIUS AND CONSITUENT DATA
        REWIND 21
        REWIND 18
        REWIND 19
        REWIND 24
        REWIND 22
C
C       ***CHECK FOR EXISTENCE OF THE OUTPUT FILES
        IPN = 0
        INQUIRE (FILE='T21.TMP',EXIST=EXIST)
        IF (EXIST.EQV..FALSE.) IPN = IPN + 1
        INQUIRE (FILE='T18.TMP',EXIST=EXIST)
        IF (EXIST.EQV..FALSE.) IPN = IPN + 1
        INQUIRE (FILE='T19.TMP',EXIST=EXIST)
        IF (EXIST.EQV..FALSE.) IPN = IPN + 1
        INQUIRE (FILE='T22.TMP',EXIST=EXIST)
        IF (EXIST.EQV..FALSE.) IPN = IPN + 1
C
        IF (IPN.EQ.0) THEN
        STR='COPY T21.TMP+T18.TMP+T19.TMP+T22.TMP'//OFILE3//' > OFILE3'


C
C         ***CLOSE OUTPUT FILES IN PREPARATION FOR CONCATINATION
          CLOSE (21)
          CLOSE (18)
          CLOSE (19)
          CLOSE (22)
C
C         ***EXECUTE OPERATING SYSTEM COMMAND, STR
C         ***RETURNS ERROR CODE, I4
          CALL CMD (STR,I4)
C
          STR = 'DEL OFILE3'
          CALL CMD (STR,I4)
C
C        END IF
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE MESSAG (IM)
C     ******************************************************************
C     ***WRITE MESSAGES TO THE SCREEN
C     ***REQUIRED ROUTINES:   NONE
C     ******************************************************************
      CHARACTER*40 STR
      COMMON /COUN/ NUM(100)
      COMMON /SIMU/ BFLXC,BFLXH,DM,DTPR,IWR,KKSTOP,KSTOP,OPERC,TM
      NUM(51) = NUM(51) + 1
C
C     ***ENTRY MESS2 IS USED TO OUTPUT A REAL*4 NUMBER WITH A MESSAG
      ENTRY MESS2 (IM,X)
C
C     ***ENTRY MESS3 IS USED TO OUTPUT A CHARACTER*40 STRING WITH MESSAGE
      ENTRY MESS3 (IM,STR)
C
C     ***ENTRY MESS4 IS USED TO OUTPUT AN INTEGER WITH A MESSAGE
      ENTRY MESS4 (IM,INTE)
C
      IF (IM.EQ.1) THEN
C       ***KOPT BANNER
      WRITE (*,9500)
      WRITE (*,9501)
      READ (*,*)
      END IF
C
C
      IF (IM.EQ.8) THEN
C       ***SEARCHING FOR FILE NAME
      WRITE (*,9580) STR
      ELSE IF (IM.EQ.3) THEN
C       ***REBUILDING *.HSS FILE
      WRITE (*,9530) STR
      ELSE IF (IM.EQ.4) THEN
C       ***REBUILDING *.TSG FILE
      WRITE (*,9540) STR
      ELSE IF (IM.EQ.5) THEN
C       ***REBUILDING *.PL1 FILE
      WRITE (*,9550) STR
      ELSE IF (IM.EQ.6) THEN
C       ***REBUILDING *.PL2 FILE
      WRITE (*,9560) STR
      ELSE IF (IM.EQ.7) THEN
C       ***REBUILDING *.PL3 FILE
      WRITE (*,9570) STR
      ELSE IF (IM.EQ.9) THEN
C       ***FILES DO NOT EXIST
      WRITE (*,9590)
      ELSE IF (IM.EQ.10) THEN
C       ***FILES DO NOT EXIST
      WRITE (*,9610)
      ELSE IF (IM.EQ.11) THEN
C       ***FILES DO NOT EXIST
      WRITE (*,9611)
      ELSE IF (IM.EQ.12) THEN
C       ***FILE ALREADY EXISTS
      WRITE (*,9612) STR
      END IF
C
      RETURN
 9500 FORMAT (/
     *        15X,'*************************************************'/
     *        15X,'*                                               *'/
     *        15X,'*                   REBUILD                     *'/
     *        15X,'*                                               *'/
     *        15X,'*             HSSM RECOVERY PROGRAM             *'/
     *        15X,'*                                               *'/
     *        15X,'*                                               *'/
     *        15X,'*        JAMES W. WEAVER AND MARK B. LEE        *'/
     *        15X,'*  R. S. KERR ENVIRONMENTAL RESEARCH LABORATORY  *'/
     *        15X,'*  UNITED STATES ENVIRONMENTAL PROTECTION AGENCY  *'/
     *        15X,'*             ADA, OKLAHOMA 74820               *')
 9501 FORMAT (15X,'*                                               *'/
     *        15X,'*              FEBRUARY 16, 1993                *'/
     *        15X,'*                                               *'/
     *        15X,'*                VERSION 1.00                   *'/
     *        15x,'*                                               *'/
     *        15X,'*************************************************'/
     *       /15X,'PRESS RETURN')
 9530 FORMAT (10X,'*** REBUILDING *.HSS FILE: '/
     *        10X,'***  ',A40)
 9540 FORMAT (10X,'*** REBUILDING *.TSG FILE: '/
     *        10X,'***  ',A40)
 9550 FORMAT (10X,'*** REBUILDING *.PL1 FILE: '/
     *        10X,'***  ',A40)
 9560 FORMAT (10X,'*** REBUILDING *.PL2 FILE: '/
     *        10X,'***  ',A40)
 9570 FORMAT (10X,'*** REBUILDING *.PL3 FILE: '/
     *        10X,'***  ',A40)
 9580 FORMAT (10X,'*** SEARCHING FOR FILE NAME: '/
     *        10X,'***  ',A40)
 9590 FORMAT (10X,'*** TEMP FILES DO NOT EXIST FOR *.HSS ')
 9610 FORMAT (10X,'*** TEMP FILES DO NOT EXIST FOR *.TSG ')
 9611 FORMAT (10X,'*** TEMP FILES DO NOT EXIST FOR *.PL* ')
 9612 FORMAT (10X,'*** FILE ALREADY EXISTS-RECOVERY NOT NEEDED FOR: '/
     *        10X,'***  ',A40)
      END
```

```
      SUBROUTINE DEL2 ()
C     ****************************************************************
C     ****************************************************************
C     ***
C     ***THIS SUBROUTINE REMOVES EXISTING *.TMP FILES.
C     ***
C     ***REQUIRED ROUTINES:
C     ***THIS ROUTINE IS CALLED BY:   PROGRAM HSSMR
C     ****************************************************************
C
C     ***OPEN THE TEMPORARY FILES NEEDED BY HSSM FOR OUTPUTTING MODEL
C     ***RESULTS CHECK FOR PRIOR EXISTENCE OF INPUT FILES
C
      LOGICAL FEXIST
C
      INQUIRE (FILE='T7.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (7,FILE='T7.TMP',STATUS='UNKNOWN')
        CLOSE (7,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T8.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (8,FILE='T8.TMP',STATUS='UNKNOWN')
        CLOSE (8,STATUS='DELETE')
      END IF
C
C     ***FILE 9 IS THE PROFILE FILE
      INQUIRE (FILE='T9.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (9,FILE='T9.TMP',STATUS='UNKNOWN')
        CLOSE (9,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T10.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (10,FILE='T10.TMP',STATUS='UNKNOWN')
        CLOSE (10,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T11.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (11,FILE='T11.TMP',STATUS='UNKNOWN')
        CLOSE (11,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T12.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (12,FILE='T12.TMP',STATUS='UNKNOWN')
        CLOSE (12,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T13.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (13,FILE='T13.TMP',STATUS='UNKNOWN')
        CLOSE (13,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T14.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (14,FILE='T14.TMP',STATUS='UNKNOWN')
        CLOSE (14,STATUS='DELETE')
      END IF
C
C     ***FILE 15 IS THE DATA ECHO PRINT AND MAIN KOPT OUTPUT FILE
      INQUIRE (FILE='T15.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (15,FILE='T15.TMP',STATUS='UNKNOWN')
        CLOSE (15,STATUS='DELETE')
      END IF

C     ***FILE 16 IS THE SATURATION PROFILE PLOT FILE
      INQUIRE (FILE='T16.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (16,FILE='T16.TMP',STATUS='UNKNOWN')
        CLOSE (16,STATUS='DELETE')
      END IF
C
C     ***FILE 17 IS THE OILENS RADIAL SECTION PLOT FILE
      INQUIRE (FILE='T17.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (17,FILE='T17.TMP',STATUS='UNKNOWN')
        CLOSE (17,STATUS='DELETE')
      END IF
C
C     ***FILE 18 IS THE OILENS RADIUS PLOT FILE
      INQUIRE (FILE='T18.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (18,FILE='T18.TMP',STATUS='UNKNOWN')
        CLOSE (18,STATUS='DELETE')
      END IF
C
C     ***FILE 19 IS THE OILENS CONTAMINANT PLOT FILE
      INQUIRE (FILE='T19.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (19,FILE='T19.TMP',STATUS='UNKNOWN')
        CLOSE (19,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T20.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (20,FILE='T20.TMP',STATUS='UNKNOWN')
        CLOSE (20,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T21.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (21,FILE='T21.TMP',STATUS='UNKNOWN')
        CLOSE (21,STATUS='DELETE')
      END IF
```

```
C                                                                       END
      INQUIRE (FILE='T22.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (22,FILE='T22.TMP',STATUS='UNKNOWN')
        CLOSE (22,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T23.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (23,FILE='T23.TMP',STATUS='UNKNOWN')
        CLOSE (23,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T24.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (24,FILE='T24.TMP',STATUS='UNKNOWN')
        CLOSE (24,STATUS='DELETE')
      END IF
C
C
      INQUIRE (FILE='T30.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (30,FILE='T30.TMP',STATUS='UNKNOWN')
        CLOSE (30,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T42.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (42,FILE='T42.TMP',STATUS='UNKNOWN')
        CLOSE (42,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T43.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (43,FILE='T43.TMP',STATUS='UNKNOWN')
        CLOSE (43,STATUS='DELETE')
      END IF
C
      INQUIRE (FILE='T44.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (44,FILE='T44.TMP',STATUS='UNKNOWN')
        CLOSE (44,STATUS='DELETE')
      END IF
C
C     ***WARNING NOTICE
      INQUIRE (FILE='TWA.TMP',EXIST=FEXIST)
      IF (FEXIST.EQV..TRUE.) THEN
        OPEN (44,FILE='TWA.TMP',STATUS='UNKNOWN')
        CLOSE (44,STATUS='DELETE')
      END IF
C
      RETURN
```

```
      Interface to integer*2 function system [c]
     &  (string[reference])
      character*1 string
      end
      subroutine cmd(a,k)
c     ****************************************************************
c     *
c     *  cmd   issues a dos command from fortran code
c     *
c     *
c     *
c     * input arguments.........................................
c     *   a     character string containing dos command
c     * output arguments........................................
c     *   k     length of command string
c     *
c     * jim cloud
c     * computer services corp
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma 74820
c     *
c     * single precision ansi std x3.9-1978 fortran 77
c     * required routines   none
c     * called by  delf,iopost,tsgp2,pkcon
c     *
c     *
c     *
c     ****************************************************************
      character a*(*), cstr*256,CSTR2*256
      integer k
      integer*2 system
      integer*1 nul
      data nul/0/
C
      k=len_trim(a)
      if(k.gt.255)return
      cstr=a(1:k)//nul
C
C     ***TURN OFF SCREEN ECHO OF COMMAND
      CSTR2='ECHO OFF'//NUL
      K=INT(SYSTEM(CSTR2))
C
C     ***EXECUTE DOS COMMAND
      k=int(system(cstr))
C
C     ***TURN ON SCREEN ECHO OF COMMANDS
      CSTR2='ECHO ON'//NUL
      K=INT(SYSTEM(CSTR2))
C
      return
      end
```

```
      SUBROUTINE DIR
C     ****************************************************************
C     *
C     *  DIR ISSUES A DOS DIRECTORY COMMAND
C     *
C     *
C     *
C     * INPUT ARGUMENTS......(NONE).............................
C     * OUTPUT ARGUMENTS.....(NONE).............................
C     *
C     * JIM WEAVER
C     * ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
C     * UNITED STATES ENVIRONMENTAL PROTECTION AGENCY
C     * ADA, OKLAHOMA  74820
C     *
C     * SINGLE PRECISION ANSI STD X3.9-1978 FORTRAN 77
C     * REQUIRED ROUTINES  CMD
C     * CALLED BY   DPF,SOPEN
C     *
C     ****************************************************************
      CHARACTER*256 STR
      INTEGER I4
C
C     ***DISPLAY DIRECTORY
      STR = 'DIR |MORE'
      CALL CMD (STR,I4)
C
      WRITE (*,*) ' '
C
      RETURN
      END
```

3.6 Source Code for SOPROP

```
      program soprop
c     ****************************************************************
c     *
c     * soprop
c     *
c     * soprop uses the regression equations developed by rawls and
c     * brakensiek (1985) to estimate soil hydraulic properties
c     * given the percent sand, clay and porosity
c     * soprop estimates the hydraulic conductivity, and brooks and corey
c     * parameters (air entry head,lambda and the residual water content)
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma  74820
c     * 405-436-8545
c     *
c     * 9-9-93
c     *
c     * double precision, ansi x3.9-1978 FORTRAN 77
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
      dimension b(4,0:2,0:2,0:2),b0(4)
      dimension p(4)
c
c
      call io (ps,pc,por)
c
      call bdata (b,b0)
c
c
c
c     ***generate the approximate parameter values
c     ***1 = ln(hydraulic conductivity)
c     ***2 = residual moisture content
c     ***3 = ln(air entry head)
c     ***4 = ln(lambda)
c
      do 100 ii=1,4
        p(ii) = b0(ii)
        do 75 i=0,2
          do 50 j=0,2
            do 25 k=0,2
              p(ii) = p(ii) + b(ii,i,j,k)*(ps**i)*(pc**j)*(por**k)
 25         continue
 50       continue
 75     continue
 100  continue
c
c
c     ***exponentiate hydraulic conductivity, entry head and lambda
      p(1) = exp(p(1))
      p(3) = exp(p(3))
      p(4) = exp(p(4))
c
c     ***convert hydraulic conductivity from cm/hr to m/d
      p(1) = p(1)*24./100.
c     ***convert entry head from cm to m
      p(3) = p(3)/100.
c     ***convert the residual mositure content to residual
c     ***moisture saturation
      p(2) = p(2)/por
c
c
      write (*,8999) ps,pc,por
      write (*,9000) (p(ii),ii=1,4)
c
 8999 format (///5x,'*******************************************'/
     *           5x,'Estimate of soil hydraulic properties'/
     *           5x,'from Rawls and Brakensiek (1985)'/
     *           5x,'regression equations'/
     *           5x,'*******************************************'/
     *           5x,'for the soil with:'/
     *           5x,f10.4,' percent sand'/
     *           5x,f10.4,' percent clay'/
     *           5x,f10.4,' porosity'/
     *           5x,'the estimated hydraulic parameters are:'/)
 9000 format (/5x,'hydraulic conductivity       ',g10.4,' m/d '/
     *           5x,'Brooks and Corey parameters:',/
     *           5x,'residual water saturation    ',g10.4/
     *           5x,'air entry head               ',g10.4,' m '/
     *           5x,'pore size distribution index ',g10.4///)
c
c
      stop '         ***successful exection of soprop'
      end
```

```
      subroutine io (ps,pc,por)
c     ****************************************************************
c     *
c     *  io  inputs data to soprop
c     *
c     *
c     * input arguments
c     *  none
c     *
c     * output arguments
c     * ps     percent sand
c     * pc     percent clay
c     * por  porosity
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma  74820
c     * 405-436-8545
c     *
c     * 9-9-93
c     *
c     * double precision, ansi x3.9-1978 FORTRAN 77
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
c
c
c     ***inquire for the input data
      write (*,*) ' '
c
 10   continue
c     ***percent sand
      write (*,*) ' '
      write (*,*) 'Enter the percent sand'
      read (*,*) ps
c
      if (ps.lt.5.) then
c       ***the percent sand must be greater than 5.0
       write (*,*) 'The percent sand must be greater than 5.0'
       go to 10
      end if
c
      if (ps.gt.70.) then
c       ***the percent sand cannot exceed 70.
       write (*,*) 'The percent sand cannot exceed 70.0'
       go to 10
      end if
c
c
c
c
 20   continue
c     ***percent clay
      write (*,*) ' '
      write (*,*) 'Enter the percent clay'
      read (*,*) pc
c
      if (pc.lt.5.) then
c       ***the percent clay must be greater than 5.0
       write (*,*) 'The percent clay must be greater than 5.0'
       go to 20
      end if
c
      if (pc.gt.60.) then
c       ***the percent clay cannot exceed 60.
       write (*,*) 'The percent clay cannot exceed 60.0'
       go to 20
      end if
c
      if (ps+pc.gt.100.) then
c       ***the sand and clay contents are greater than 100.
       write (*,*) 'The percent sand & clay cannot be greater than 100'
       go to 10
      end if
c
c
c     ***sat water content
      write (*,*) ' '
      write (*,*) 'Enter the porosity'
      read (*,*) por
c
c
c
      return
      end
```

```
      subroutine bdata (b,b0)
c     ****************************************************************
c     *
c     *  io  inputs data to soprop
c     *
c     *
c     * input arguments
c     *  none
c     *
c     * output arguments
c     * b(i,j,k)    rawls and brakensiek (1985) regression coefficients
c     * b0(i)       rawls and brakensiek (1985) constant b0
c     *
c     * jim weaver
c     * robert s. kerr environmental research laboratory
c     * united states environmental protection agency
c     * ada, oklahoma  74820
c     * 405-436-8545
c     *
c     * 9-9-93
c     *
c     * double precision, ansi x3.9-1978 FORTRAN 77
c     *
c     ****************************************************************
      implicit double precision (a-h), double precision (o-z)
      dimension b(4,0:2,0:2,0:2),b0(4)
c
c     ***set the coefficients for the regression equations
c
c     ***zero out the arrays
      do 100 ii=1,4
        b0(ii) = 0.0
        do 75 i=0,2
          do 50 j=0,2
            do 25 k=0,2
              b(ii,i,j,k) = 0.0
 25         continue
 50       continue
 75     continue
 100  continue
c
c     ***set the regression coefficients
c     ***b(1,....) is for hydraulic conductivity)
      b0(1) = -8.96847
      b(1,1,0,0) = 0.0
      b(1,0,1,0) = -0.028212
      b(1,0,0,1) = 19.52348
      b(1,2,0,0) = 0.00018107
      b(1,0,2,0) = -0.0094125
      b(1,0,0,2) = -8.395215
      b(1,1,1,0) = 0.0
      b(1,1,0,1) = 0.077718
      b(1,0,1,1) = 0.0
      b(1,2,1,0) = 0.0000173
      b(1,0,2,1) = 0.02733
      b(1,2,0,1) = 0.001434
      b(1,1,2,0) = -0.0000035
      b(1,0,1,2) = 0.0
      b(1,2,0,2) = -0.00289
      b(1,0,2,2) = -0.019492
c
c
c     ***b(2,....) is for residual moisture content
      b0(2) = -0.0182482
      b(2,1,0,0) = 0.00087269
      b(2,0,1,0) = 0.00513488
      b(2,0,0,1) = 0.02939286
      b(2,2,0,0) = 0.0
      b(2,0,2,0) = -0.00015395
      b(2,0,0,2) = 0.0
      b(2,1,1,0) = 0.0
      b(2,1,0,1) = -0.0010827
      b(2,0,1,1) = 0.0
      b(2,2,1,0) = 0.0
      b(2,0,2,1) = 0.00030703
      b(2,2,0,1) = 0.0
      b(2,1,2,0) = 0.0
      b(2,0,1,2) = -0.0023584
      b(2,2,0,2) = 0.0
      b(2,0,2,2) = -0.00018233
c
c     ***b(3,....) is for the entry head
      b0(3) = 5.3396738
      b(3,1,0,0) = 0.0
      b(3,0,1,0) = 0.1845038
      b(3,0,0,1) = -2.48394546
      b(3,2,0,0) = 0.0
      b(3,0,2,0) = -0.00213853
      b(3,0,0,2) = 0.0
      b(3,1,1,0) = 0.0
      b(3,1,0,1) = -0.0435649
      b(3,0,1,1) = -0.61745089
      b(3,2,1,0) = -0.00001282
      b(3,0,2,1) = 0.00895359
      b(3,2,0,1) = -0.00072472
      b(3,1,2,0) = 0.0000054
      b(3,0,1,2) = 0.50028060
      b(3,2,0,2) = 0.00143598
      b(3,0,2,2) = -0.00855375
c
c     ***b(4,....) is for lambda
      b0(4) = -0.7842831
      b(4,1,0,0) = 0.0177544
      b(4,0,1,0) = 0.0
```

```
      b(4,0,0,1) = -1.062498
      b(4,2,0,0) = -0.00005304
      b(4,0,2,0) = -0.00273493
      b(4,0,0,2) = 1.11134946
      b(4,1,1,0) = 0.0
      b(4,1,0,1) = -0.03088295
      b(4,0,1,1) = 0.0
      b(4,2,1,0) = -0.00000235
      b(4,0,2,1) = 0.00798746
      b(4,2,0,1) = 0.0
      b(4,1,2,0) = 0.0
      b(4,0,1,2) = -0.00674491
      b(4,2,0,2) = 0.00026587
      b(4,0,2,2) = -0.00610522
c
      return
      end
```

3.7 Compilation with Microsoft FORTRAN

The HSSM modules and the utilities were compiled with Microsoft FORTRAN version 5.1 compilier. With one exception, ANSI standard x3.9-1978 FORTRAN was used in writing these routines. Compliance with the standard was verified by using the /4Ybs compile option on the Microsoft compiler. The full compilation statement was

```
fl /4Ybs /Zi /Od /FPi file.FOR
```

The `/FPi` option was used so that HSSM would take advantage of a math coprocessor if available, but not require the coprocessor. The CMD routine was compiled with the /4Yb option which does not enforce strict ANSI compliance. CMD issues an operating system command to DOS, a feature that is not available in standard FORTRAN 77. The following routines require the CMD routine:

NSOPEN
PKCON
IOPOST
TSGP2
DIR

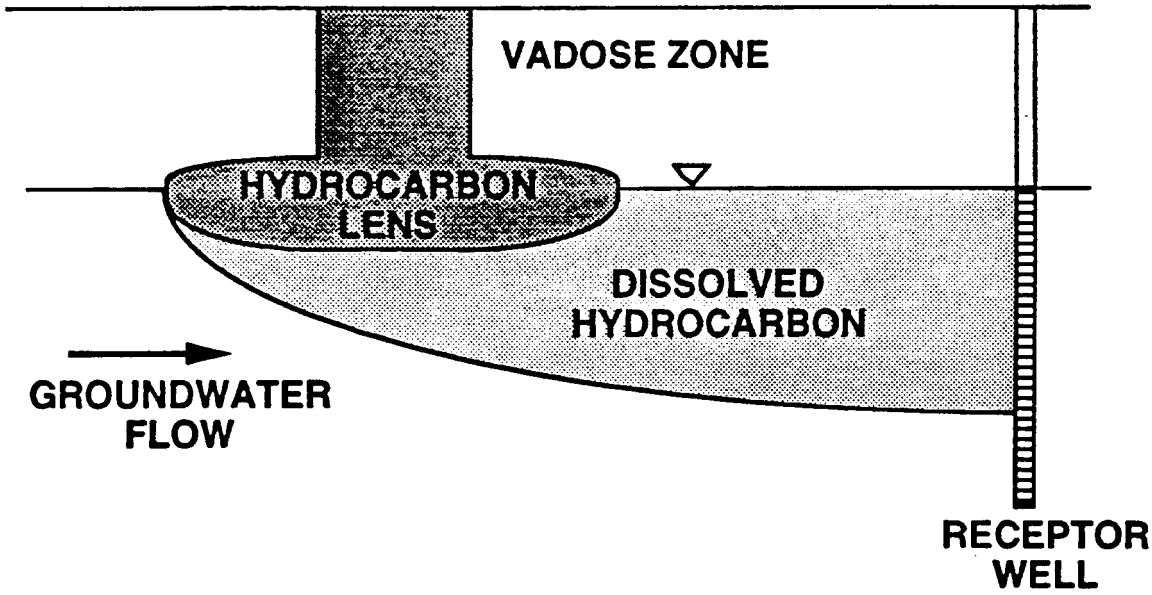

Figure 1 Schematic view of the Hydrocarbon Spill Screening Model (HSSM) scenario

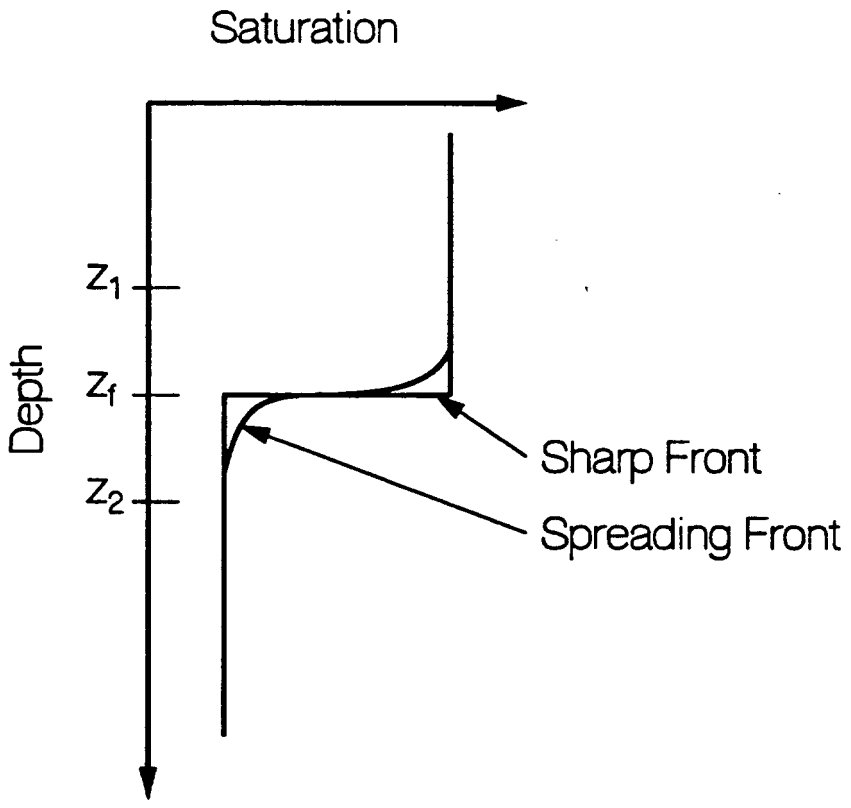

Figure 2 Schematic comparison of sharp and diffuse fronts

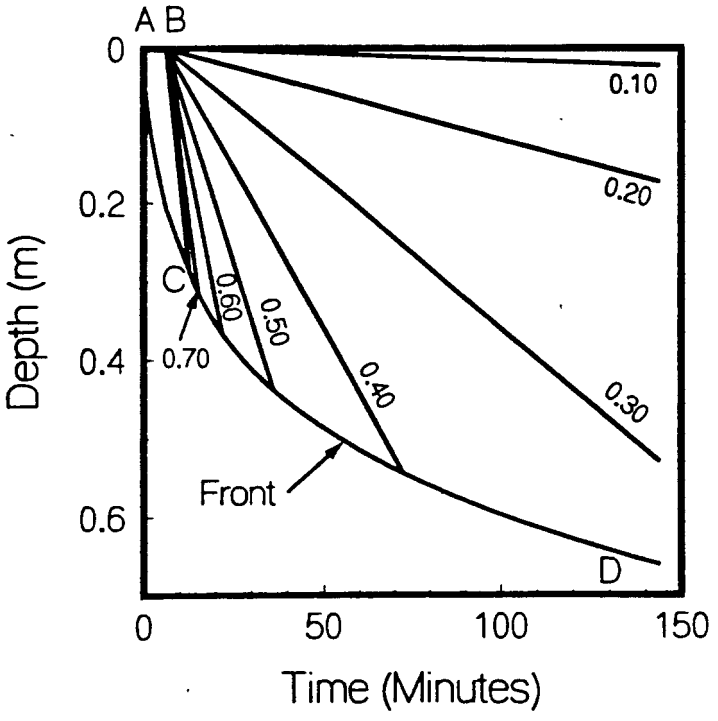

Figure 3 Base characteristic plane

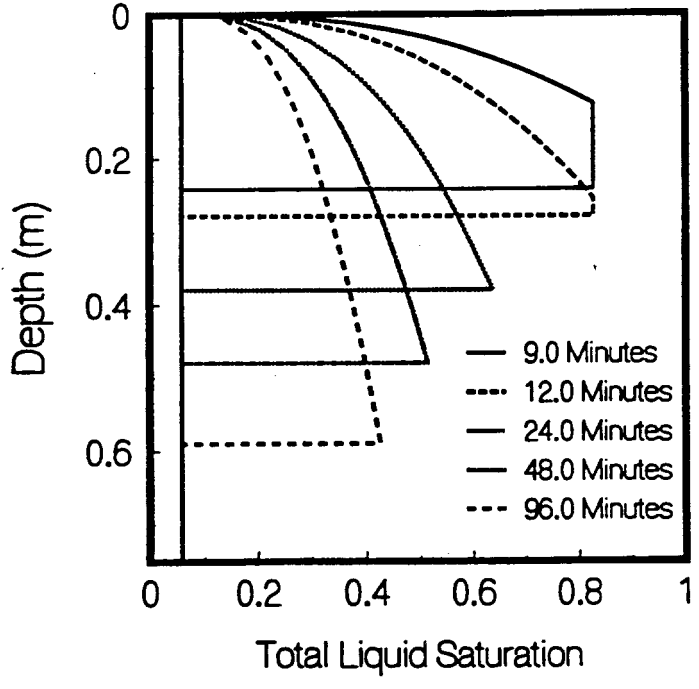

Figure 4 Total liquid profiles

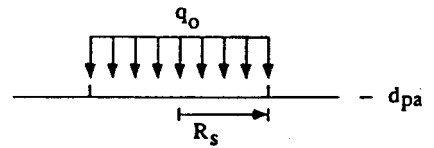

Figure 5 KOPT model release scenarios

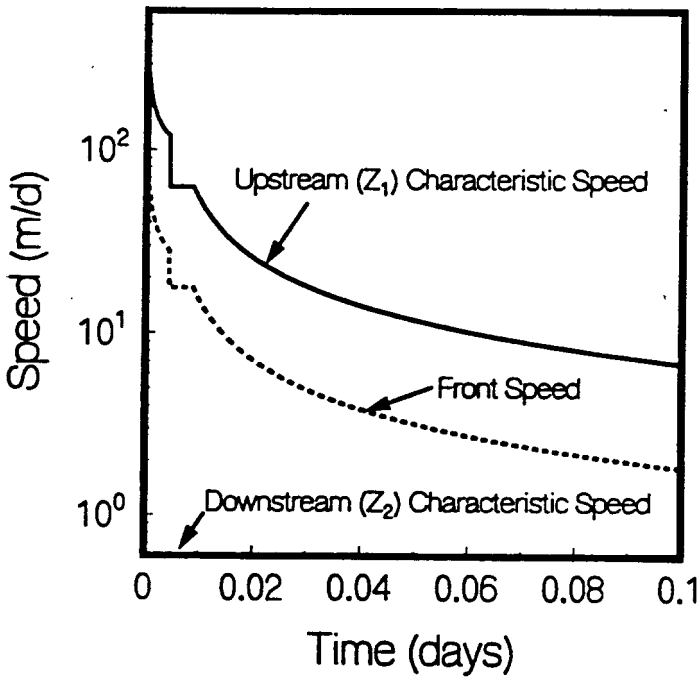

Figure 6 Upstream characteristic and front speeds

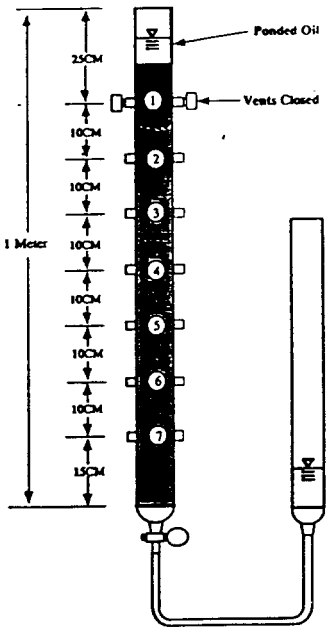

Figure 7 Glass column used for the laboratory evaluation of KOPT

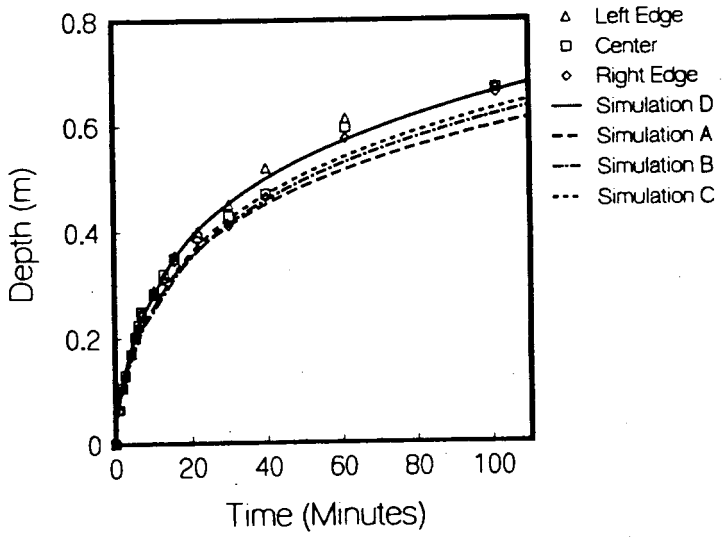

Figure 8 Measured NAPL position at right-hand edge, center and left-hand edge of column.

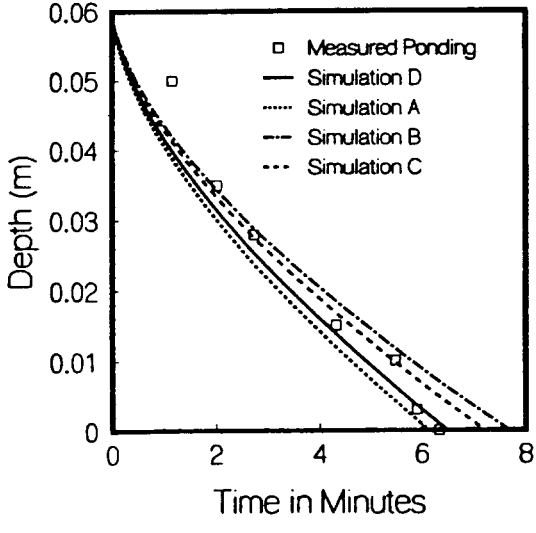

Figure 9 Measured NAPL ponding depth at the surface of the sand

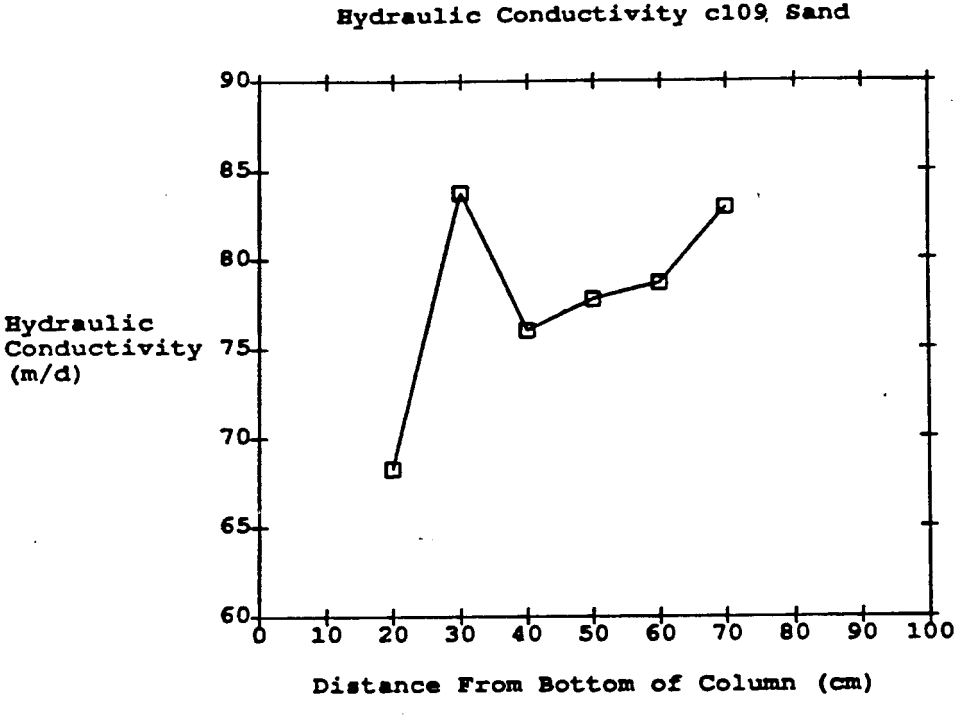

Figure 10 Measured distribution of hydraulic conductivity in the sand pack

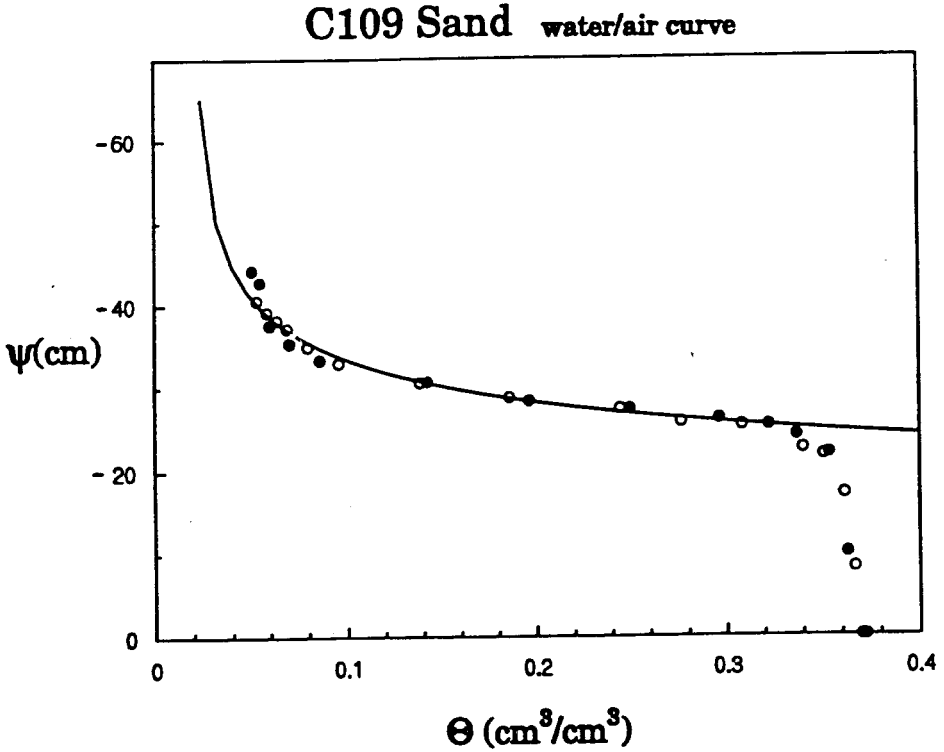

Figure 11 Data from two measured capillary pressure curves for the c109 sand and the fitted Brooks and Corey model (solid line)

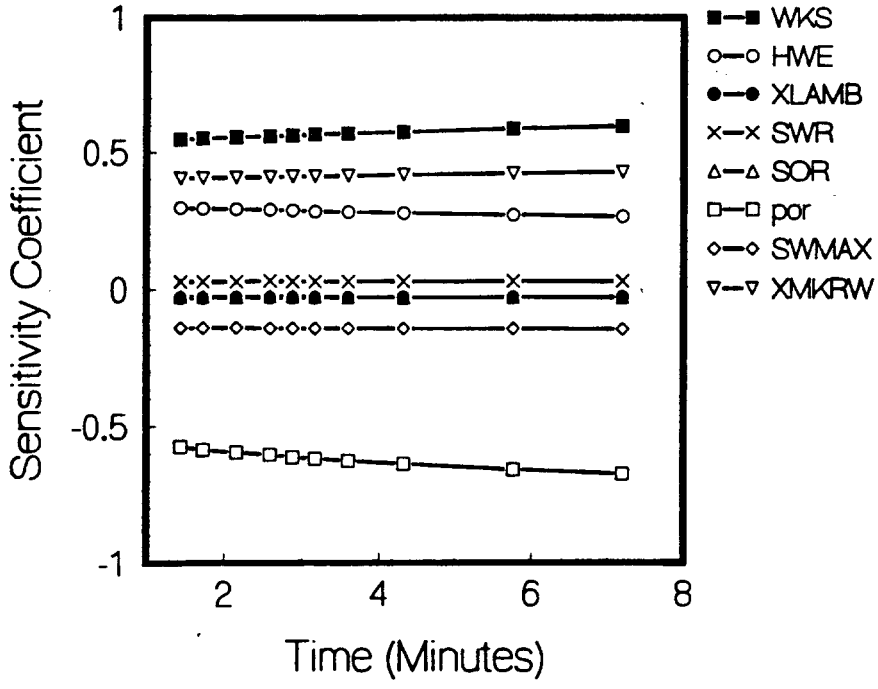

Figure 12 Nondimensional sensitivity coefficients for the Green-Ampt portion of the simulation B parameter set

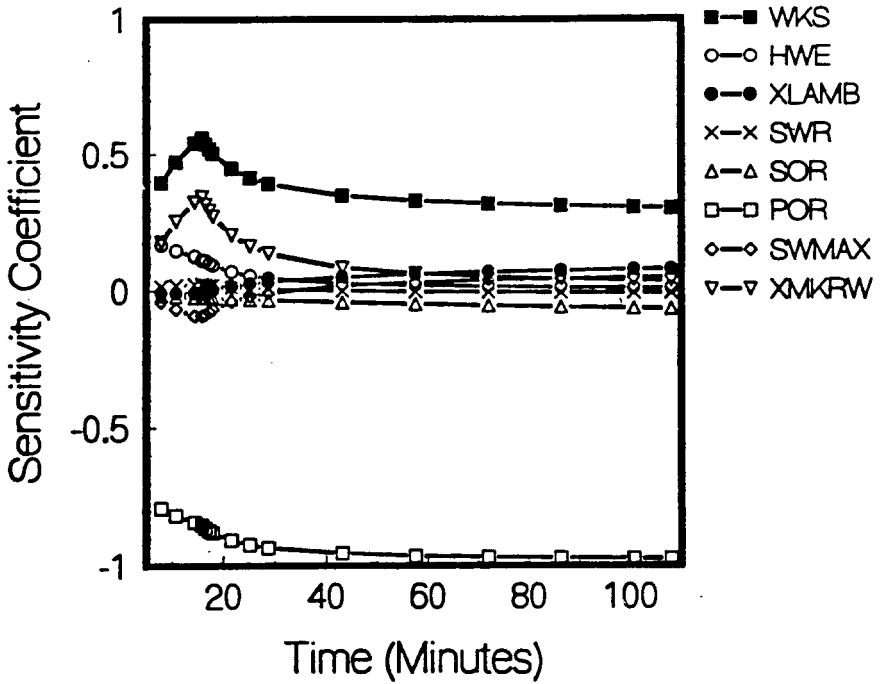

Figure 13 Nondimensional sensitivity coefficients for the kinematic portion of the simulation B parameter set

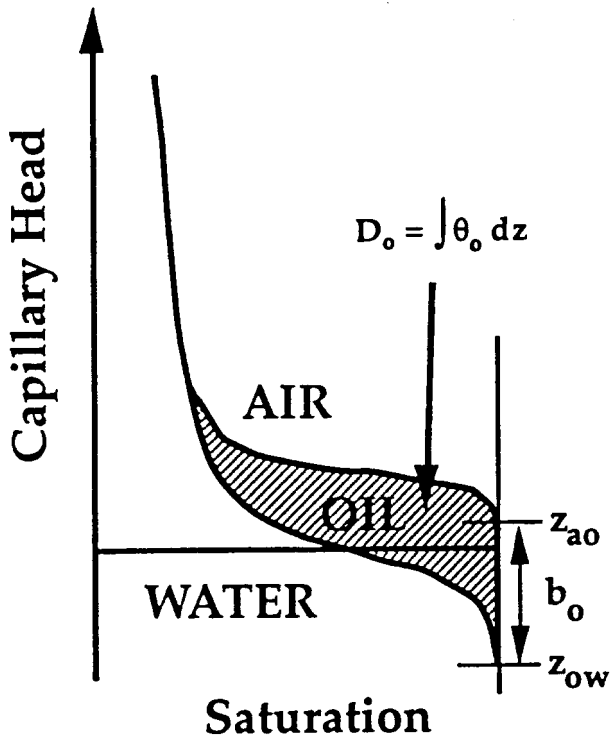

Figure 14 Calculation of LNAPL thickness in an oil lens

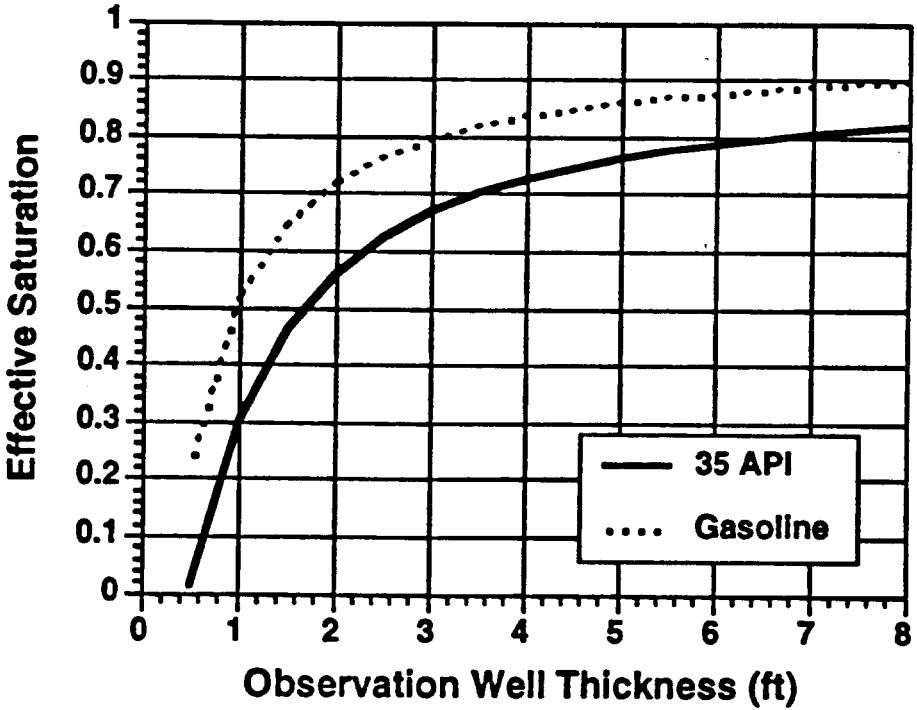

Figure 15 Effective saturation of a hydrocarbon in a sand

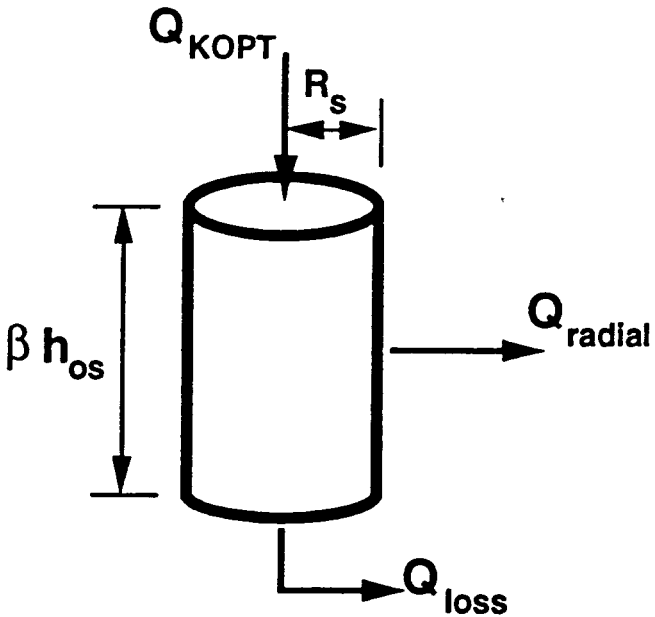

Figure 16 Volume balance for the source cylinder

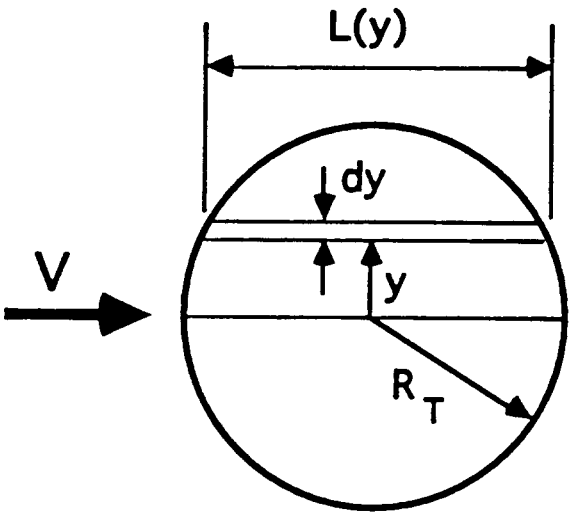

Figure 17 Plan view of the oil lens

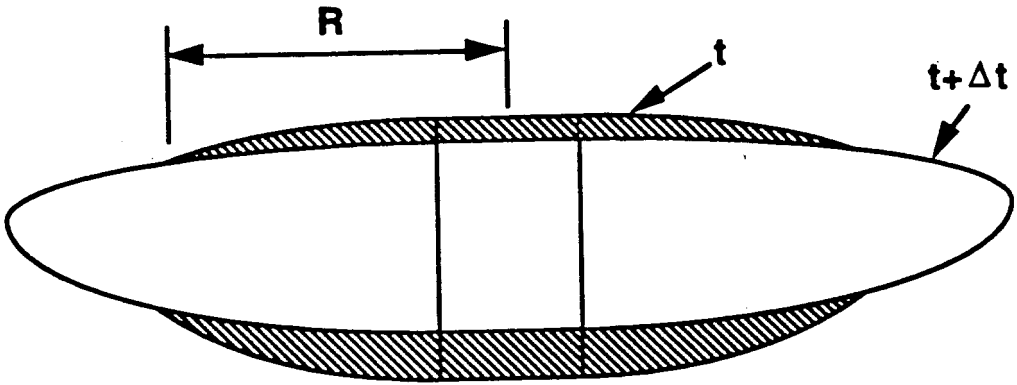

Figure 18 Residual volume for decaying mound

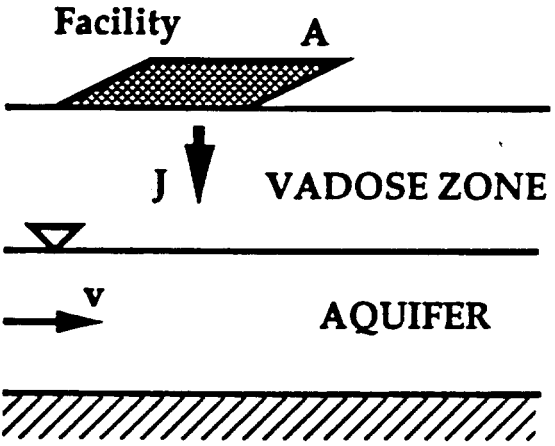

Figure 19 Basic setup of the gaussian-source plume model

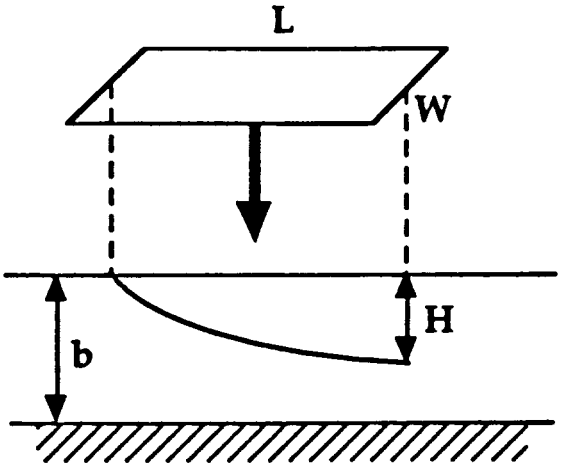

Figure 20 Development of mixing zone beneath the facility

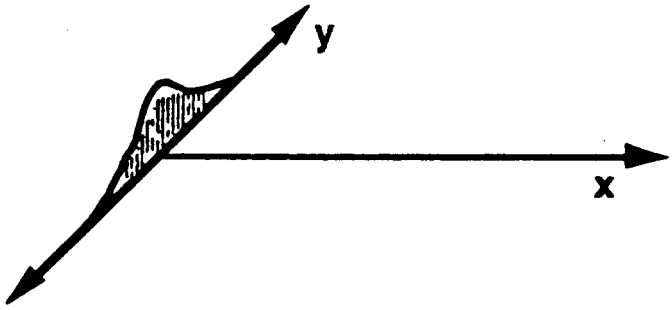

Figure 21 Gaussian distribution which is taken as the boundary condition at the downstream extent of the area beneath the facility

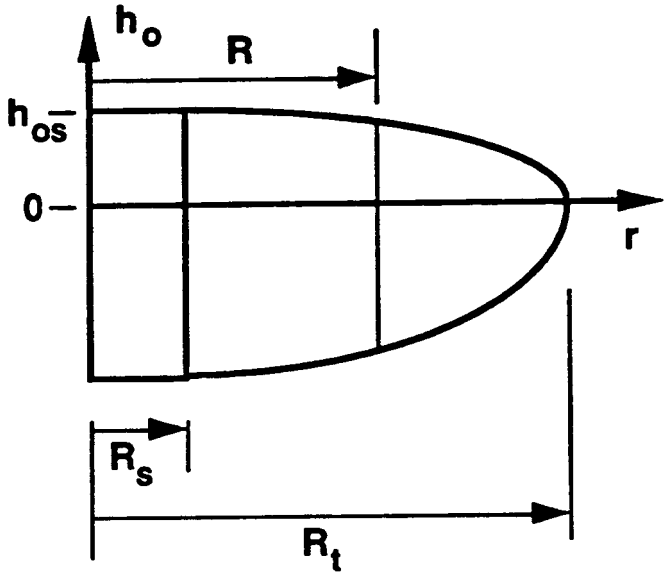

Figure 62 Representation for the lens volume