SAND84-0878
Unlimited Release
Printed May 1985

Distribution
UC-70

PETROS -- A PROGRAM FOR CALCULATING TRANSPORT OF HEAT, WATER, WATER VAPOR AND AIR THROUGH A POROUS MATERIAL

G. R. Hadley
Fluid Mechanics and Heat Transfer Division II
Sandia National Laboratories
Albuquerque, NM 87185

Abstract

The one-dimensional code PETROS computes the transport of water, water vapor, an inert gas, and heat through a partially saturated porous medium. The mass flux of liquid water is driven by gradients in saturation, temperature, and gas pressure as well as the force of gravity. Gas transport includes effects due to Knudsen diffusion and binary gaseous diffusion of each gas component, plus Darcy flow of the gas mixture. Evaporation and condensation are accounted for, both in the fluid mass balance and the heat equation. This report includes a description of the model assumptions and the resulting equations, together with the numerical techniques used to obtain problem solutions. Included also are instructions for r code, and a sample problem.

CONTENTS

NOMENCLATURE

a	constant appearing in Eq. (1)
c	evaporation rate coefficient [Eq. (3)]
c_p	specific heat at constant pressure $(\frac{J}{kg\ K})$
D_{vK}	Knudsen diffusion coefficient $(\frac{m^2}{s})$
D_{va}	molecular binary gas diffusion coefficient $(\frac{m^2}{s})$
F	evaporation rate $(\frac{kg}{m^3 s})$
$\mathscr{F}$	total evaporation rate [Eq. (58)]
g	acceleration of gravity $(\frac{m}{s^2})$
h	mass transfer coefficient in Eq. (79) $(\frac{kg}{m^2 s})$
J	mass flux $(\frac{kg}{m^2 s})$
k	relative permeability (subscripted)
k	Boltzmann's constant (nonsubscripted)
K	thermal conductivity $(\frac{W}{mK})$
L	latent heat of vaporization $(\frac{J}{kg})$
m	molecular mass (kg)
n	geometry indicator [Eq. (5)]
p	pressure (Pa)
Q	heat flux $(\frac{J}{m^2 s})$
r	position coordinate (m)
R	mesh ratio [Eq. (62)]
$\mathscr{R}$	average pore radius [Eq. (24)] (m)
s	saturation
$\mathscr{S}$	constant defined by Eq. (21)
t	time (s)
T	temperature (K)
$\mathscr{V}$	total liquid volume [Eq. (59)] (m^3)

Greek

α	mixing factor (Section VI), thermal diffusivity [Eq. (70)]
γ	mass density $(\frac{kg}{m^3})$
ε	truncation error [Eq. (57)]
κ	absolute permeability (m^2)
μ	viscosity $(\frac{kg}{ms})$
ξ	function defined by Eq. (33)
σ	surface tension $(\frac{N}{m})$
τ	tortuosity factor
ϕ	porosity

Subscripts

a	air
c	capillary
ℓ	liquid
i	quantity evaluated at mesh point i
v	water vapor, gas (in the case of μ_v)
0	reference value

FIGURES

TABLES

I. INTRODUCTION

The work described in this report was performed by Sandia National Laboratories as a part of the Nevada Nuclear Waste Storage Investigations (NNWSI) project. Sandia is one of the principal organizations participating in the project, which is managed by the U. S. Department of Energy's Nevada Operations Office. The project is a part of the Department of Energy's program to safely dispose of the radioactive waste from nuclear power plants.

The Department of Energy has determined that the safest and most feasible method currently known for the disposal of such wastes is to emplace them in mined geologic repositories. The NNWSI project is conducting detailed studies of an area on and near the Nevada Test Site (NTS) in southern Nevada to determine the feasibility of developing a repository.

PETROS is a one-dimensional finite difference code which calculates the transport of heat plus three fluid components (water, water vapor, and air) through a partially saturated porous material. This report includes a discussion of the model used, the numerical methods, and instructions for the potential user.

This code was originally written to study the transport of water and water vapor in a nuclear waste repository. If such repositories are located above the water table, then the movement of all three fluid components can become important. Also, change of phase for the water component must be included because of its role in heat and mass transport. The inclusion of such a broad range of phenomena has also made the code useful for a variety of other problems, two of which are the study of porous heat pipes and the drying of porous materials.

PETROS calculates the time dependent transport of liquid water (due to capillary forces and gradients of temperature and gas pressure), including the liquid mass loss resulting from phase change. The treatment of the binary gas system (assumed to be in steady state) includes effects due to Darcy flow as well as Knudsen diffusion of each gas species and binary diffusion of one gas relative to the other. This more complete treatment of the gas phase was motivated by the desire to handle transport in a tight geological material which displays a small average pore size. Heat

transport includes conduction, convection by all three phases, and latent heat effects.

All transport equations except the heat equation have been written in integral form and then differenced on a stationary Eulerian grid. This procedure has allowed the solution of problems with extremely steep saturation gradients (i.e., evaporation fronts). Differencing in time is fully implicit using a predictor-corrector scheme to handle the nonlinear terms. The time step is controlled internally in order to maintain good solution accuracy and prevent unwanted instabilities. Provisions have been made for the inclusion of several user-defined properties such as capillary pressure, relative permeability, and thermal conductivity. MKS units are used throughout the code.

Sections I-III of this manual describe the basic physical assumptions made in arriving at a mathematical model and the resulting differential equations. Sections IV-VIII cover the numerical methods, the resulting difference equations, and the content of internally supplied properties routines. Section IX describes code checkout together with some indication of expected accuracy. The casual user who is interested only in running the code should skim over Sections V-VIII and then read the Appendices, where user instructions and coding of a sample problem are given and error messages are explained.

II. MODELING ASSUMPTIONS

The following is a description of the model used, together with relevant simplifying assumptions:

1) All modeling is one-dimensional in planar, cylindrical, or spherical geometry. The porous medium is thus composed of slabs, cylindrical or spherical shells, each of which may be a different homogeneous and isotropic porous material. In each case the porous matrix is assumed to be nondeformable.

2) Liquid motion is transient and is assumed to occur as a noninertial (Darcy) flow which results from an imposed pressure gradient plus the force of gravity. The pressure gradient is made up of the following components:

   i) Surface tension effects (capillary forces). Differences in pressure between the gas and liquid phases are assumed to be representable in terms of a single function of the form

$$p_c = p_c(s)\left(1 + a(T-T_o)\right) \qquad (1)$$

   where s and T are the saturation and temperature, respectively. No hysteresis effects are included, and although $p_c(s)$ may be different for each material, a and T_o are constant and apply to all materials during a problem. Thus from (1), liquid motion may result from saturation gradients or temperature gradients.

   ii) Effects resulting from gas pressure gradients. Unlike the derivation of Richard's equation [1], the gas pressure is not assumed to be constant. Thus liquid motion may be driven by gradients in the gas pressure. Change of phase is allowed and included in the liquid mass balance. Darcy's law employs a relative permeability which depends only on s.

3) The motion of both gas phases is treated as noninertial and steady state. The latter condition follows from the assumption that gas velocities are much higher than liquid velocities and equilibration times are correspondingly shorter. The two gas species are assumed ideal, and their motion includes the following effects [2]:

   i) Darcy flow of the combined mixture using a relative permeability which depends only on s.

ii) Knudsen diffusion of each component relative to the porous medium, assuming a single average pore size for each material.

iii) Binary gaseous diffusion of one gas relative to the other. A constant tortuosity may be inserted if desired for both types of diffusion.

Change of phase is included in the water vapor mass balance.

4) The rate of phase change is modeled in two different ways which are subject to user choice. In the equilibrium model, the water vapor pressure is set equal to the equilibrium value when liquid is present, i.e.,

$$p_v = \begin{cases} p_{sat}(T) & (s > s_1) \\ \text{determined from momentum eqn.} & (s < s_1) \end{cases} , \qquad (2)$$

where s_1 is a small but nonzero cutoff saturation. For this case, evaporation rates are determined from the divergence of the water vapor flux. The other option is the nonequilibrium model, in which the evaporation rate is given by

$$F = cs(p_{sat}(T) - p_v) \qquad (3)$$

with c constant. For this case p_v is determined from the solution of a pressure equation. Here c must be increased until the solution no longer changes (becomes independent of c); as a guide, use $c = 10^5\sqrt{\kappa}$. Values of c which are excessively large may cause problems in calculation caused by the subtraction of two nearly equal quantities [Equation (3)].

5) Heat transfer is assumed to be due to conduction and convection only, with dispersion effects neglected. The mixture is assigned a single conductivity, which may depend on position and time directly or indirectly through other problem variables. Convection of all fluid phases is included. The heat equation is solved transiently and includes latent heat effects. The latent heat of vaporization is allowed to depend linearly on temperature.

III. GOVERNING EQUATIONS

Equations consistent with the modeling assumptions listed in Section II have been derived elsewhere [3] using volume averaging and mixture theory, and will only be listed here. The continuity equation for liquid water is (in one dimension)

$$\phi\gamma_\ell \frac{\partial s}{\partial t} + \frac{1}{r^n}\frac{\partial}{\partial r}\left(r^n J_\ell\right) = -F \tag{4}$$

where F is the evaporation rate (mass/vol time) and J_ℓ is the liquid mass flux averaged over the porous material. The geometry indicator n takes on the value

$$n = \begin{cases} 0 & \text{planar geometry} \\ 1 & \text{cylindrical geometry} \\ 2 & \text{spherical geometry} \end{cases} \tag{5}$$

In accordance with Darcy's law, the flux J_ℓ is given by

$$J_\ell = -\frac{\gamma_\ell \kappa k_\ell}{\mu_\ell}\left[\frac{\partial p_\ell}{\partial r} - \gamma_\ell g\right] , \tag{6}$$

where the force of gravity is in the direction of positive r, and is generally expected to be set to zero for cases involving curvilinear geometries. We eliminate p_ℓ in Equation (6) from the definition of capillary pressure:

$$p_c(s,T) = p - p_\ell \tag{7}$$

where p is the total gas pressure

$$p = p_v + p_a \tag{8}$$

Equation (6) thus becomes

$$J_\ell = -\frac{\gamma_\ell \kappa k_\ell}{\mu_\ell}\left[\frac{\partial p}{\partial r} - \frac{\partial p_c}{\partial s}\frac{\partial s}{\partial r} - \frac{\partial p_c}{\partial T}\frac{\partial T}{\partial r} - \gamma_\ell g\right] . \tag{9}$$

We write

$$p_c(s,T) = p_c(s,T_o)\left[1 + \frac{d\ell n\sigma}{dT}\,(T-T_o)\right] \tag{10}$$

where σ is the surface tension and for convenience we set $T_o = 300$ K. Then Equation (9) may be further expanded to read

$$J_\ell = -\frac{\gamma_\ell K k_\ell}{\mu_\ell}\left[\frac{\partial p}{\partial r} - \frac{\partial p_c}{\partial s}\,f(T)\,\frac{\partial s}{\partial r} - p_c\,\frac{d\ell n\sigma}{dT}\,\frac{\partial T}{\partial r} - \gamma_\ell g\right] \tag{11}$$

$$\text{where } f(T) \equiv 1 + \frac{d\ell n\sigma}{dT}\,(T-T_o)\quad , \tag{12}$$

and $p_c(s,T_o)$ has been abreviated to p_c. Equations (3), (4), (5), and (11), together with the auxiliary functions $k_\ell(s)$, $\mu_\ell(T)$, and $p_c(s,T_o)$ thus specify the liquid transport. The heat transport equation is

$$\gamma c_p\big|_{mix}\,\frac{\partial T}{\partial t} + \left(\underline{c}_p \cdot \underline{J}\right)\frac{\partial T}{\partial r} = \frac{1}{r^n}\,\frac{\partial}{\partial r}\left(r^n K\,\frac{\partial T}{\partial r}\right) - FL(T)\quad , \tag{13}$$

where K is the mixture conductivity, L(T) is the latent heat of vaporization, and $\underline{c}_p$ and $\underline{J}$ are defined by

$$\underline{c}_p = \begin{pmatrix} c_{p\ell} \\ c_{pa} \\ c_{pv} \end{pmatrix} \tag{14}$$

$$\underline{J} = \begin{pmatrix} J_\ell \\ J_a \\ Jv \end{pmatrix}\quad . \tag{15}$$

The mixture heat capacity is

$$\gamma c_p\big|_{mix} = (1-\phi)\gamma_m c_{pm} + \phi s\gamma_\ell c_{p\ell} \tag{16}$$

where the subscript m refers to the porous matrix. The two gas components (water vapor and an inert component which we refer to as "air") satisfy the steady state continuity equation

$$\frac{1}{r^n}\frac{\partial}{\partial r}\left(r^n J_v\right) = F \tag{17}$$

$$\frac{1}{r^n}\frac{\partial}{\partial r}\left(r^n J_a\right) = 0 \quad . \tag{18}$$

The fluxes are given by [2]

$$J_v = -\frac{\phi(1-s)m_v D_{v\kappa}}{kT}\frac{\partial p_v}{\partial r} - \frac{p_v m_v \kappa k_v}{\mu_v kT}\frac{\partial p}{\partial r} + \frac{D_{v\kappa}}{pD_{va}}\left(\frac{p_v J_a}{\mathscr{S}^2} - p_a J_v\right) \tag{19}$$

$$J_a = -\frac{\phi(1-s)m_v \mathscr{S} D_{v\kappa}}{kT}\frac{\partial p_a}{\partial r} - \frac{p_a \mathscr{S}^2 m_v \kappa k_v}{\mu_v kT}\frac{\partial p}{\partial r} + \frac{D_{v\kappa}}{pD_{va}}\left(p_a \mathscr{S} J_v - \frac{p_v J_a}{\mathscr{S}}\right) \tag{20}$$

$$\text{with } \mathscr{S} = \sqrt{\frac{m_a}{m_v}} \quad . \tag{21}$$

Equations (19) and (20) may be solved for the fluxes:

$$J_v = -\frac{\phi(1-s)m_v D_{v\kappa}}{kT}\left[\frac{\frac{\partial p_a}{\partial r} + \left(1 + \frac{p\mathscr{S}D_{va}}{p_v D_{v\kappa}}\right)\frac{\partial p_v}{\partial r}}{1 + \frac{p_a \mathscr{S}}{p_v} + \frac{p\mathscr{S}D_{va}}{p_v D_{v\kappa}}}\right] - \frac{\kappa k_v m_v p_v}{\mu_v kT}\frac{\partial p}{\partial r} \tag{22}$$

$$J_a = -\frac{\phi(1-s)m_v \mathscr{S} D_{v\kappa}}{kT}\left[\frac{\frac{\partial p_v}{\partial r} + \left(1 + \frac{pD_{va}}{p_a D_{v\kappa}}\right)\frac{\partial p_a}{\partial r}}{1 + \frac{p_v \mathscr{S}}{p_a} + \frac{pD_{va}}{p_a D_{v\kappa}}}\right] - \frac{\kappa k_v m_v \mathscr{S}^2 p_a}{\mu_v kT}\frac{\partial p}{\partial r} \quad . \tag{23}$$

Equations (17), (18), (22), and (23) together comprise a set of coupled nonlinear elliptic differential equations for the two pressures p_v and p_a.

The Knudsen diffusion coefficient is given by [2]

$$D_{v\kappa} = \frac{2}{3}\left(\mathscr{R}/\tau\right)\sqrt{\frac{8kT}{\pi m_v}} \tag{24}$$

where $\mathscr{R}$ is the average pore radius and τ the tortuosity factor. The binary diffusion coefficient D_{va} is given by [4]

$$D_{va} = 2.3 \times 10^{-5} \left(\frac{p_o}{p\tau}\right)\left(\frac{T}{T_o}\right)^{1.81} \frac{m^2}{s} \quad , \tag{25}$$

where $p_o = 9.8 \times 10^4$ N/m^2 and $T_o = 256$ K. As an aid to the user, an equivalent pore radius for Equation (24) may be obtained from the capillary tube model

$$\mathscr{R} = \sqrt{\frac{8\phi\tau}{\kappa}} \quad . \tag{26}$$

IV. DIFFERENCE EQUATIONS

We first construct the mesh shown in Figure 1. All primary variables (p_v, p_a, s, and T) are defined at the node points, and all fluxes (J_v, J_a, and J_ℓ) are defined at the midnode points indicated by dashed lines. These lines represent boundaries of a control volume over which we shall integrate the transport equations. Beginning with Equation (4), we have

$$\gamma_\ell \frac{\partial}{\partial t} \int_{i-1/2}^{i+1/2} \phi r^n s dr + r^n J_\ell \Big|_{i-1/2}^{i+1/2} = - \int_{i-1/2}^{i+1/2} r^n F dr \tag{27}$$

where we have multiplied through by r^n before integration. If we approximate s and F as being piecewise linear and ϕ constant between mesh points, then the integrals in Equation (27) are

$$\int_{i-1/2}^{i+1/2} \phi r^n s dr = r_i^n s_i \frac{\phi_i \Delta r_i + \phi_{i-1} \Delta r_{i-1}}{2} \tag{28}$$

$$\int_{i-1/2}^{i+1/2} r^n F dr = r_i^n F_i \frac{\Delta r_i + \Delta r_{i-1}}{2} \tag{29}$$

where Δr_i is defined as shown in Figure 1. The correction terms to (28) and (29) are of order $\frac{n}{4} \left(\frac{\Delta r}{r}\right)$. Thus Equations (28) and (29) are quite adequate except for curvilinear geometries at positions very near the origin. Substituting Equations (28), (29), and (11) into (27) gives the difference equation for s:

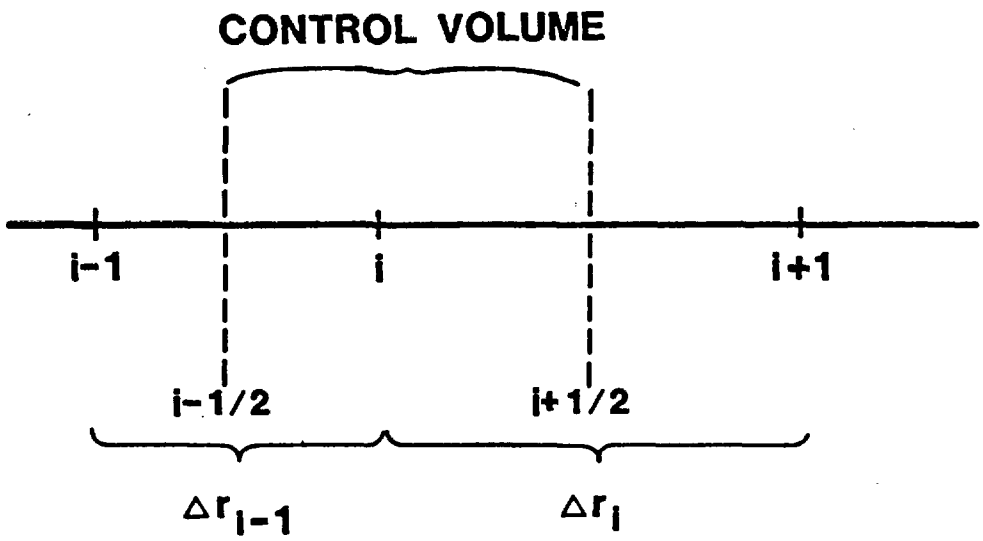

Figure 1. Nomenclature used for numerical grid.

$$s_i' - s_i = \frac{\Delta t}{r_i^n \frac{1}{2}(\phi_i \Delta r_i + \phi_{i-1}\Delta r_{i-1})} \left\{ \frac{r_{i+1/2}^n \kappa_i k_\ell^{i+1/2}}{\mu_\ell^{i+1/2}} \right.$$

$$\left(\frac{p_{i+1} - p_i}{\Delta r_i} - \gamma_\ell g - \left.\frac{\partial p_c}{\partial s}\right|_{i+1/2} \frac{s_{i+1}' - s_i'}{\Delta r_i} f(T_{i+1/2}) \right.$$

$$\left. - \left. p_c \right|_{i+1/2} \frac{d\ell n\sigma}{dT} \frac{T_{i+1} - T_i}{\Delta r_i} \right) - \frac{r_{i-1/2}^n \kappa_{i-1} k_\ell^{i-1/2}}{\mu_\ell^{i-1/2}} \left(\frac{p_i - p_{i-1}}{\Delta r_{i-1}} - \gamma_\ell g \right.$$

$$\left. - \left.\frac{\partial p_c}{\partial s}\right|_{i-1/2} \frac{s_i' - s_{i-1}'}{\Delta r_{i-1}} f(T_{i-1/2}) - \left. p_c \right|_{i-1/2} \frac{d\ell n\sigma}{dT} \frac{T_i - T_{i-1}}{\Delta r_{i-1}} \right)$$

$$\left. - \frac{r_i^n F_i}{2\gamma_\ell} (\Delta r_i + \Delta r_{i-1}) \right\} . \qquad (30)$$

The convention of using primes to represent values advanced in time by one time step Δt shows that Equation (30) has been differenced in a fully implicit manner. It is conservative in form [5] so that if $F_i = 0$ everywhere, liquid mass is exactly conserved. Boundary conditions of either Dirichlet or Neumann type may be imposed. For Dirichlet conditions, s(1) and/or s(IMAX) are specified. For a Neumann boundary condition, the mass flux at $i = \frac{3}{2}$ or $i = IMAX - \frac{1}{2}$ (or both) is specified, leading to the boundary equations

$$s_2 - s_1 = \frac{\Delta r_1}{f(T_{3/2}) \left.\frac{dp_c}{ds}\right|_{3/2}} \left\{ \left. \frac{p_2 - p_1}{\Delta r_1} + \frac{J_\ell \mu_\ell}{\gamma_\ell \kappa k_\ell} \right|_{3/2} \right.$$

$$\left. - \left. p_c \right|_{3/2} \frac{d\ell n\sigma}{dT} \frac{T_2 - T_1}{\Delta r_1} - \gamma_\ell g \right\} \qquad (31)$$

$$s_N - s_{N-1} = \frac{\Delta r_{N-1}}{f(T_{N-1/2}) \left.\frac{dp_c}{ds}\right|_{N-1/2}} \left\{ \frac{p_N - p_{N-1}}{\Delta r_{N-1}} + \left.\frac{J_\ell \mu_\ell}{\gamma_\ell \kappa k_\ell}\right|_{N-1/2} \right.$$

$$\left. - \left. p_c \right|_{N-1/2} \frac{d\ell n\sigma}{dT} \frac{T_N - T_{N-1}}{\Delta r_{N-1}} - \gamma_\ell g \right\} \qquad (32)$$

where N = IMAX is the number of mesh points.

The heat Equation (13) is differenced in a similar manner, by multiplying through by r^n and integrating over the control volume shown in Figure 1. Using the abbreviation

$$\xi_i \equiv \frac{1}{2} \left(\Delta r_{i-1} \gamma c_p \Big|_{mix,i-1} + \Delta r_i \gamma c_p \Big|_{mix,i} \right) \quad , \qquad (33)$$

the heat equation becomes

$$r_i^n \xi_i \frac{\partial T_i}{\partial t} + \underline{c}_p \cdot \int_{i-1/2}^{i+1/2} r^n \underline{J} \frac{\partial T}{\partial r} \, dr = \left[r^n K \frac{\partial T}{\partial r} \right]_{i-1/2}^{i+1/2}$$

$$- FL(T_i) \frac{r_i^n}{2} (\Delta r_i + \Delta r_{i-1}) \quad . \qquad (34)$$

Since $\underline{J}$ is defined at the control volume boundaries, we approximate $\frac{\partial T}{\partial r}$ as piecewise linear between mesh points, so that

$$\int_{i-1/2}^{i+1/2} r^n \underline{J} \frac{\partial T}{\partial r} \, dr = \frac{T_i - T_{i-1}}{\Delta r_{i-1}} \underline{J}_{i-1/2} \int_{i-1/2}^{i} r^n dr$$

$$+ \frac{T_{i+1} - T_i}{\Delta r_i} \underline{J}_{i+1/2} \int_{i}^{i+1/2} r^n dr \quad . \qquad (35)$$

This reduces to

$$\int_{i-1/2}^{i+1/2} r^n \underline{J} \frac{\partial T}{\partial r}\, dr = \frac{1}{2}\underline{J}_{i-1/2}(T_i - T_{i-1})r_i^n + \frac{1}{2}\underline{J}_{i+1/2}(T_{i+1} - T_i)r_i^n \quad , \tag{36}$$

where the truncation error is again of order $(\frac{\Delta r}{r})$. The resulting differenced heat equation is

$$T_i' - T_i + \frac{\Delta t}{2\xi_i}\left[\underline{c}_p \cdot \underline{J}_{i+1/2}(T_{i+1}' - T_i') + \underline{c}_p \cdot \underline{J}_{i-1/2}(T_i' - T_{i-1}')\right] = \frac{\Delta t}{r_i^n \xi_i}\left[r_{i+1/2}^n K_{i+1/2}\frac{T_{i+1}' - T_i'}{\Delta r_i} - r_{i-1/2}^n K_{i-1/2}\frac{T_i' - T_{i-1}'}{\Delta r_{i-1}}\right] - \frac{F_i L(T_i)\frac{1}{2}(\Delta r_i + \Delta r_{i-1})\Delta t}{\xi_i} \quad . \tag{37}$$

The latent heat term has been modeled as

$$L(T) = L_o + \frac{dL}{dT}(T - T_o) \tag{38}$$

with $T_o = 373$ K. Equation (37) has been differenced in a fully implicit manner, and, as is the case with the saturation Equation (29), may be solved using a tri-diagonal algorithm, provided the nonlinear terms are appropriately treated. As before, boundary conditions for Equation (37) may be of either the Dirichlet or Neumann type. For Neumann boundary conditions, the heat flux at mesh location $\frac{3}{2}$ or $N - \frac{1}{2}$ is specified, leading to either

$$T_2 - T_1 = -\frac{\Delta r_1 Q_{3/2}}{K_{3/2}} \tag{39}$$

or

$$T_N - T_{N-1} = - \frac{\Delta r_{N-1} Q_{N-1/2}}{K_{N-1/2}} \quad . \tag{40}$$

Prior to differencing the gas momentum Equations (17), (18), (22), and (23), we first rewrite those equations in a more compact form using the following definitions:

$$\underline{\Phi} \equiv \begin{pmatrix} J_v \\ J_a \end{pmatrix}, \quad \underline{p} \equiv \begin{pmatrix} p_v \\ p_a \end{pmatrix}, \text{ and } \quad \underline{d} \equiv \begin{pmatrix} r^{n_F} \\ 0 \end{pmatrix} . \tag{41}$$

The fluxes may then be expressed as

$$\Phi_j = \sum_k T_{jk} \frac{\partial p_k}{\partial r} \quad , \tag{42}$$

where

$$T_{11} \equiv - \frac{\phi(1-s) m_v D_{v\kappa}}{kT} \frac{p_v + \dfrac{p\mathscr{S} D_{va}}{D_{v\kappa}}}{p_v + p_a\mathscr{S} + \dfrac{p\mathscr{S} D_{va}}{D_{v\kappa}}} - \frac{\kappa k_v m_v p_v}{\mu_v kT} \tag{43}$$

$$T_{12} \equiv - \frac{\phi(1-s) m_v D_{v\kappa}}{kT} \frac{p_v}{p_v + p_a\mathscr{S} + \dfrac{p\mathscr{S} D_{va}}{D_{v\kappa}}} - \frac{\kappa k_v m_v p_v}{\mu_v kT} \tag{44}$$

$$T_{21} \equiv - \frac{\phi(1-s) m_v \mathscr{S} D_{v\kappa}}{kT} \frac{p_a}{p_a + \dfrac{p_v}{\mathscr{S}} + \dfrac{pD_{va}}{D_{v\kappa}}} - \frac{\kappa k_v m_v \mathscr{S}^2 p_a}{\mu_v kT} \tag{45}$$

$$T_{22} \equiv - \frac{\phi(1-s) m_v \mathscr{S} D_{v\kappa}}{kT} \frac{p_a + \dfrac{pD_{va}}{D_{v\kappa}}}{p_a + \dfrac{p_v}{\mathscr{S}} + \dfrac{pD_{va}}{D_{v\kappa}}} - \frac{\kappa k_v m_v \mathscr{S}^2 p_a}{\mu_v kT} \quad . \tag{46}$$

Equations (17) and (18) are multiplied by r^n and integrated over the control volume as before. The resulting difference equations, when expressed in tensor notation using the definitions (41) through (46) are written as the single equation

$$r^n_{i+1/2} T^{i+1/2}_{jk} \frac{p_k^{i+1} - p_k^i}{\Delta r_i} - r^n_{i-1/2} T^{i-1/2}_{jk} \frac{p_k^i - p_k^{i-1}}{\Delta r_{i-1}}$$

$$= d_j \frac{1}{2} (\Delta r_i + \Delta r_{i-1}) \quad . \tag{47}$$

In Equation (47), repeated tensor indices are assumed to be summed over. Since the coefficients T_{jk} in Equation (47) are functions of the pressures, we linearize according to

$$T^{i\pm1/2}_{jk} = {}^{\circ}T^{i\pm1/2}_{jk} + \frac{1}{2} {}^{\circ}T^{i\pm1/2}_{jk,\ell} (p_\ell^i + p_\ell^{i\pm1} - {}^{\circ}p_\ell^i - {}^{\circ}p_\ell^{i\pm1}) \quad . \tag{48}$$

Here we have used the notation

$$T_{jk,\ell} \equiv \frac{\partial T_{jk}}{\partial p_\ell} \tag{49}$$

and the superscript $^{\circ}$ refers to values at the last time step.
The final difference equation for the gas momentum equations is thus

$$r^n_{i+1/2} \left[{}^{\circ}T^{i+1/2}_{jk} \frac{p_k^{i+1} - p_k^i}{\Delta r_i} + \frac{1}{2} {}^{\circ}T^{i-1/2}_{jk,\ell} (p_\ell^i + p_\ell^{i+1} - {}^{\circ}p_\ell^i - {}^{\circ}p_\ell^{i+1}) \right.$$

$$\left. \frac{{}^{\circ}p_k^{i+1} - {}^{\circ}p_k^i}{\Delta r_i} \right] - r^n_{i-1/2} \left[{}^{\circ}T^{i-1/2}_{jk} \frac{p_k^i - p_k^{i-1}}{\Delta r_{i-1}} + \frac{1}{2} {}^{\circ}T^{i-1/2}_{jk,\ell} (p_\ell^i + p_\ell^{i-1} \right.$$

$$\left. - {}^{\circ}p_\ell^i - {}^{\circ}p_\ell^{i-1}) \frac{{}^{\circ}p_k^i - {}^{\circ}p_k^{i-1}}{\Delta r_{i-1}} \right] = d_j \frac{1}{2} (\Delta r_i + \Delta r_{i-1}) \quad . \tag{50}$$

Equation (50) may now be solved using a linear block tri-diagonal algorithm. It is still effectively nonlinear, however, and iteration must

be employed until convergence is achieved. Boundary conditions for Equation (50) are expressed as

$$\begin{pmatrix} p_v \\ p_a \end{pmatrix}^{(1)} = \begin{pmatrix} EE_{11} & EE_{12} \\ EE_{21} & EE_{22} \end{pmatrix} \begin{pmatrix} p_v \\ p_a \end{pmatrix}^{(2)} + \begin{pmatrix} FF_1 \\ FF_2 \end{pmatrix} \tag{51}$$

$$\begin{pmatrix} p_v \\ p_a \end{pmatrix}^{(N)} = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} \begin{pmatrix} p_v \\ p_a \end{pmatrix}^{(N-1)} + \begin{pmatrix} H_1 \\ H_2 \end{pmatrix} \tag{52}$$

where $\underline{\underline{EE}}$ and $\underline{\underline{G}}$ are 2 x 2 matrices and $\underline{FF}$ and $\underline{H}$ are vectors, all defined by the user.

V. MATERIAL INTERFACES

For the equations developed so far, all variables have been treated as continuous. However, saturation is known to be discontinuous across a surface dividing two materials with different capillary pressure curves. This occurs because the liquid and gas pressures (and therefore the capillary pressure) must be continuous across the interface. If the capillary pressure curves are then different for each material, it follows that a discontinuity in saturation is required. This discontinuity is numerically described by defining the saturation at the interface node as the arithmetic mean of the discontinuous values. That is,

$$s_i = \frac{1}{2}\left(s_i^+ + s_i^-\right) \quad . \tag{53}$$

The continuity of capillary pressure across the interface may be expressed as

$$p_c^-(s_i^-) = p_c^+(s_i^+) \quad , \tag{54}$$

which may be rewritten as

$$p_c^-(s_i^-) = p_c^+(2s_i - s_i^-) \quad . \tag{55}$$

Equation (55) may be solved for s_i^- if s_i is known, and then s_i^+ found from Equation (53). The values s_i^+ and s_i^- are inserted into the flux expressions, and are also used for property evaluations.

VI. TIME-STEP PROCEDURE

The difference equations presented in the last section could be used to advance the problem in time simply by advancing s, then T, followed by a pressure solution. However, the strong nonlinear coupling between equations, together with possible nonlinear boundary conditions, makes this simple procedure inadequate to prevent numerical oscillations for difficult problems, even with a carefully controlled time step. Consequently, an elaborate time-step procedure has been incorporated, which will allow well-behaved solutions for most problems of interest. In general, problems which produce high evaporation/condensation rates cause the greatest difficulty due to the coupling between the gas momentum and heat equations. However, for some problems a simpler time integration procedure is adequate, and may be employed according to user option.

Although the user specifies an initial time step, successive time steps are calculated internally, following, in part, the procedure documented by Gresho, et al [6]. Upon completing a step, the norm of an effective relative error is first determined according to

$$|y_n| \equiv \max|s_n - s_{n-1}| + \frac{1}{300}\max|T_n - T_{n-1}| \quad , \tag{56}$$

where the maximum over the mesh is implied. Then, for a required truncation error ε, the next time step Δt_{n+1} is estimated from

$$\Delta t_{n+1} = \Delta t_n \left[\frac{3\varepsilon\left(1 + \frac{\Delta t_{n-1}}{\Delta t_n}\right)}{y_n}\right]^{1/3} . \tag{57}$$

The ratio $\frac{\Delta t_{n+1}}{\Delta t_n}$ is, however, restricted to be ≤ 1.25, and if a value below 0.8 is obtained, the entire time step is repeated using the new value Δt_{n+1}.

Once the new time step is computed, all variables are advanced in time using the method described below. Prior to that description, however, some terminology needs to be defined. "Old" value refers to values of the

dependent variables at the end of the last successful time step. The next time step will in general be subdivided into several smaller time steps, and the solutions at the end of each of these subdivisions are called "provisional" values for the variables s and T.

Prior to the n^{th} time step advancement, the computed time step Δt_n is divided by the factor $2^{NPROV-1}$ to provide a smaller time step for use in a predictor-corrector type scheme. The time step is successively doubled with new predictions of the variables made each time, until the computed time step Δt_n is recovered. The overall algorithm is then as follows:

1) set temporary values = provisional values (or old values for first iteration)
2) advance s and T from the old time to obtain new provisional values using temporary values in nonlinear terms
3) mix provisional and temporary values linearly as follows to get new provisional values

$$PROV = \alpha PROV + (1-\alpha)TEMP$$

4) solve for p_v and p_a using provisional values in nonlinear terms. Update fluxes, evaporation rates
5) DT = DT*2 until full time step Δt_n is recovered.

The reason for this admittedly elaborate procedure is to obtain estimates of the advanced variables to use in the nonlinear terms so as to avoid numerical instabilities. If the mixing parameter α is set to unity, then the nonlinear terms are always evaluated at the midpoint of the next time subinterval, as illustrated by the following example: (refer to Figure 2 for time positions)

EXAMPLE: NPROV = 3

iteration	Δt	evaluation of nonlinear terms	solution
1	$\frac{\Delta t_n}{4}$	0	0 → 1
2	$\frac{\Delta t_n}{2}$	1	0 → 2
3	Δt_n	2	0 → 3 (final solution)

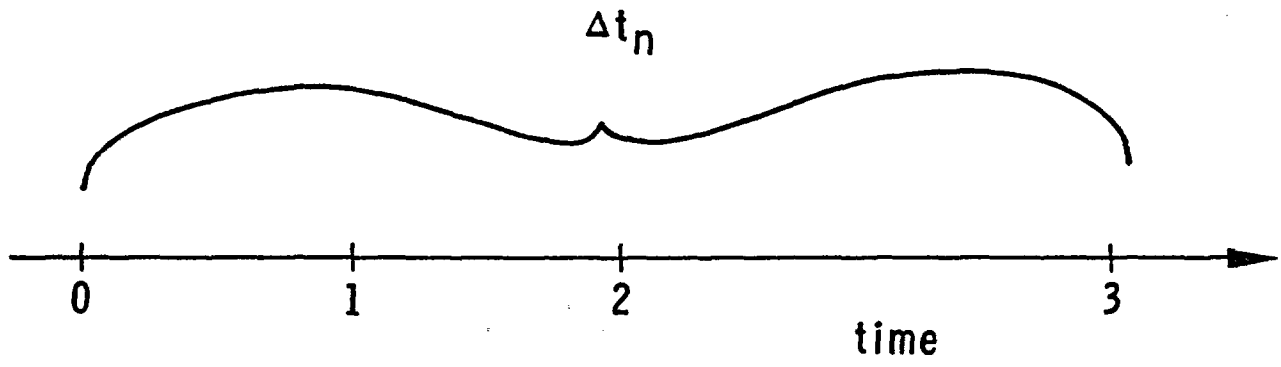

Figure 2. Illustration of time integration method (see example).

For values of α less than unity, the nonlinear terms are advanced even more slowly, as might be necessary to reduce numerical instabilities in problems with high evaporation rates. Notice that the more standard single-pass method is recovered if NPROV = 1.

Each edit displays two auxiliary quantities, the total evaporation rate and the total liquid volume, which can often provide useful information concerning the solution as well as a check on the numerical accuracy being obtained during the time integration. They are defined by

$$\mathscr{F} = \int F r^n dr \tag{58}$$

$$\mathscr{V} = \int \phi s r^n dr \quad . \tag{59}$$

These quantities satisfy the equations

$$\gamma_\ell \frac{\partial \mathscr{V}}{\partial t} + \left[r^n (J_\ell + J_v)\right]_{\text{left}}^{\text{right}} = 0 \tag{60}$$

$$\mathscr{F} = \left[r^n J_v\right]_{\text{left}}^{\text{right}} \tag{61}$$

where "left" and "right" denote the problem boundaries. For a problem with mass flux boundary conditions, the displayed quantities $\mathscr{F}$ and $\mathscr{V}$, together with Equations (60) and (61), provide a convenient check of the code's mass conservation. The latter should hold to good accuracy provided that s does not exceed unity. In that case s is truncated to unity and mass will not be conserved. For such problems, the degree of nonconservation may be monitored by watching the transient behavior of $\mathscr{V}$.

VII. MESH GENERATION

The problem region for PETROS consists of IMAX mesh points, with mesh point 1 representing the smallest value of the coordinate r and IMAX the largest. The region in between may be divided into NREG regions ($1 \leq NREG \leq 10$), each of which may be a separate material. All boundaries separating regions are coincident with some mesh point. Within each region, the mesh spacings must satisfy the geometrical progression

$$\frac{\Delta r_i}{\Delta r_{i-1}} = R \quad , \tag{62}$$

where R may be different for each region. It is advisable that adjacent mesh spacing ratios (even across region boundaries) stay close to unity (say within the interval [0.8 to 1.25]), since for large ratios the center of each control volume is no longer near a mesh point, and solution accuracy may suffer. More typical values are R = [0.9 to 1.10].

The user may want to determine the mesh size at the ends of a region, knowing the number of mesh points, the region width, and the zoning ratio, in order to anticipate the disparity between mesh sizes crossing region boundaries. To this end, the following formulas are useful for a region of width x and zoning ratio R. The first zone is of width

$$\Delta r_1 = \begin{cases} \dfrac{x(R-1)}{R^n - 1} \quad , & R \neq 1 \\ \dfrac{x}{n} \quad , & R = 1 \end{cases} \tag{63}$$

where n is the number of zones in the region. The last zone is of width

$$\Delta r_n = \Delta r_1 R^{n-1} \quad . \tag{64}$$

VIII. INTERNALLY SUPPLIED ROUTINES

Because of the many property functions required to run a problem, user specification of each function would lead to unreasonable demands upon the user who intends to run a simple scoping problem. Consequently, the code may be run entirely from the data file with no other functions or subroutines supplied, if desired. In this mode, properties are set internally with scaling accomplished by parameters from the data file. Below is a list of all internally supplied property routines, including some which may not be changed by the user (marked with an asterisk). MKS units are used throughout.

A. Viscosity of Water*

This function subprogram determines the viscosity of water from 293 K to 573 K at saturation using a quadratic interpolator and data table. Outside this interval, the value used is that at the nearest endpoint. The data were taken from Eckert and Drake [4] and the Handbook of Chemistry and Physics [7].

B. Saturation Vapor Pressure of Water*

This function subprogram computes the equilibrium vapor pressure of water at saturation for temperatures from 293 K to 643 K using a quadratic interpolator and data table. Outside this interval, the value used is that at the nearest endpoint. The data were taken from the Handbook of Chemistry and Physics [7].

C. Knudsen Diffusion Coefficient*

This function subprogram computes the Knudsen diffusion coefficient for water vapor in a porous material of pore size $\mathscr{R}$ at temperature T, according to the formula [2]

$$D_{vK} = 22.8 \frac{\mathscr{R}}{\tau} \sqrt{T} \quad , \tag{65}$$

where τ is the (user-defined) tortuosity factor.

D. Binary Diffusion Coefficient*

This function subprogram computes the binary diffusion coefficient for water vapor diffusing through air at pressure p and temperature T. Eckert and Drake [4] give this formula as

$$D_{va} = 2.3 \times 10^{-5} \frac{9.8 \times 10^4}{p\tau} \left(\frac{T}{256}\right)^{1.81} \tag{66}$$

where τ is the (user-defined) tortuosity factor.

E. Liquid Relative Permeability

This function subprogram computes the relative permeability of the liquid phase according to the formula [1]

$$k_\ell = \left[\frac{s - s_o}{1 - s_o}\right]^n \tag{67}$$

with n and s_o (the irreducible saturation) input from the data file. The subprogram limits k_ℓ to be ≤ 1. For $s \leq s_o$, k_ℓ is set to 1.E-9.

F. Gas Relative Permeability

This function subprogram sets the gas phase relative permeability [1] to

$$k_v = 1 - k_\ell \quad , \tag{68}$$

with k_ℓ computed as described above.

G. Capillary Pressure

This function subprogram computes capillary pressure by evaluating the Leverett function for drainage [1]. This evaluation is made via quadratic interpolation from a data table and scaled by a value of capillary pressure read from the data file.

H. Capillary Pressure Derivative

This function subprogram evaluates the derivative of the capillary pressure curve, discussed in G (Leverett Function), with respect to s. Evaluation proceeds using quadratic interpolation and a data table, and is scaled by the same parameter which scales the capillary pressure curve.

I. Thermal Conductivity

This function subprogram calculates the thermal conductivity of the multiphase medium consisting of matrix, water, and gas using a simple linear model

$$K_{mix} = K_{dry}(1-s) + sK_{wet} \quad , \tag{69}$$

where K_{dry} is the thermal conductivity of the matrix plus gas-filled pores and K_{wet} that of the fully saturated matrix.

IX. CODE CHECKOUT

Because PETROS solves a set of coupled nonlinear equations, comparison with analytic solutions for a complete problem is impossible. In addition, other established codes which contain the same capability, and which could therefore provide reliable solutions for checking, do not exist. Consequently, code checkout has been restricted to comparisons with a selection of uncoupled problems which check the code a section at a time. Thus, below we describe comparisons with analytic solutions for heat transfer and liquid mass transfer separately. Then the vapor transport equations are checked against published numerical solutions.

A. Heat Transfer

For this test we uncouple the heat transport equation by using a test problem with zero evaporation rate and zero mass flux. For this case heat transport takes place through conduction alone. For constant values of thermal conductivity and specific heat, analytic solutions may be obtained. We consider the transient problem shown in Figure 3. Initially, the region between $r = 0$ and $r = r_o$ contains a solid at temperature T_o. At $t = 0$ a fluid at temperature T_f begins flowing past the right boundary while the left boundary is kept adiabatic. The resulting temperature profile is given by Eckert and Drake [4] for both planar and cylindrical geometry:

planar

$$\frac{T - T_f}{T_o - T_f} = 2 \sum_{n=1}^{\infty} \frac{\sin\lambda_n r_o \cos\lambda_n r}{\lambda_n r_o + \sin\lambda_n r_o \cos\lambda_n r_o} e^{-\alpha\lambda_n^2 t} \tag{70}$$

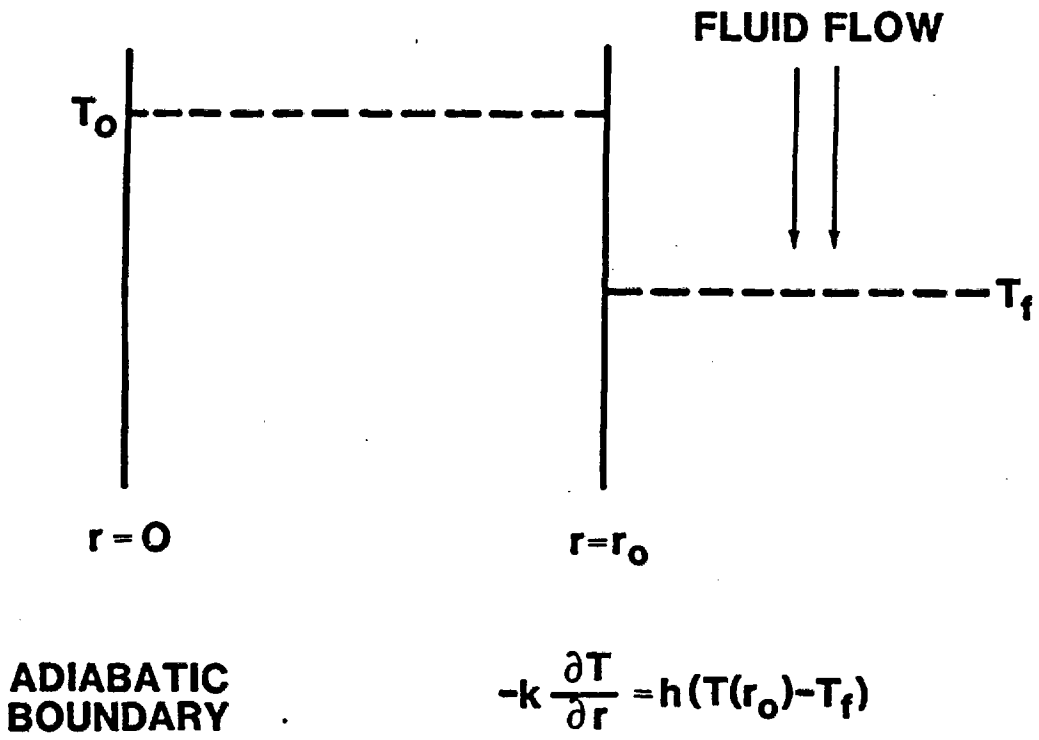

Figure 3. Schematic of test problem for heat and mass transfer sections of code.

with the eigenvalue equation

$$\cot(\lambda r_o) = \frac{\lambda K}{h} \quad , \tag{71}$$

where K and α are the thermal conductivity and diffusivity, respectively.

<u>cylindrical</u>

$$\frac{T - T_f}{T_o - T_f} = 4 \sum_{n=1}^{\infty} \frac{J_o(\lambda_n r)}{(\lambda_n^2 r_o^2 + 4) J_o(\lambda_n r_o)} e^{-\alpha \lambda_n^2 t} \tag{72}$$

with the eigenvalue equation

$$\frac{\lambda K}{h} = \frac{J_o(\lambda r_o)}{J_1(\lambda r_o)} \quad . \tag{73}$$

In Equations (72) and (73), J_o and J_1 are the Bessel functions of orders 0 and 1, respectively.

We use the following values:

$$K = 1 \frac{W}{mK} \qquad h = 20 \frac{W}{m^2 K}$$

$$T_o = 350 \text{ K}$$

$$\rho c_p = 10^5 \frac{J}{m^3}$$

$$T_f = 300 \text{ K}$$

$$r_o = 0.1 \text{ m} \qquad \text{IMAX} = 75$$

The resulting comparison with PETROS given for r = 0 and r = 0.1 is presented in Table I for both planar and cylindrical geometry. Times have been chosen large enough so that at most two terms of the series need be evaluated. The largest deviation from the analytic solution is 1.2%, indicating a satisfactory numerical solution.

Table I.

t	T(0)	$T_{analytic}^{(0)}$	% Error	T(0.1)	$T_{analytic}^{(0.1)}$	% Error
Planar						
69.72	349.73	349.67	0.11	330.1	329.87	0.46
944.25	319.1	319.7	1.2	308.97	309.35	0.76
Cylindrical						
61.28	349.51	349.23	0.54	329.09	328.94	0.3
549.68	316.61	316.40	0.44	307.40	307.46	0.13

B. Liquid Mass Transport

Although the liquid mass transport equation is of a linear form, it is normally rendered nonlinear because of the complicated functional form of the capillary pressure, relative permeability, etc. We may, however, produce a linear equation identical in form to the heat equation by choosing simple functional forms or constants for these quantities. The saturation equation thus takes the form

$$\frac{\partial s}{\partial t} = \frac{\kappa k_\ell C}{\phi \mu_\ell} \frac{1}{r^n} \frac{\partial}{\partial r} \left(r^n \frac{\partial s}{\partial r} \right) \quad , \tag{74}$$

where we have set $\frac{dP_c}{ds} = -C$ (a constant), and $g = 0$, $F = 0$ and $\frac{\partial p}{\partial r} = 0$ in Equations (4) and (9). For simplicity we have solved a problem identical to that sketched in Figure 3 with the identifications

$$K \rightarrow \frac{\gamma_\ell \kappa k_\ell C}{\mu_\ell} \quad , \tag{75}$$

$$\alpha \rightarrow \frac{\kappa k_\ell C}{\phi \mu_\ell} \quad . \tag{76}$$

With the problem thus defined, the analytic solutions in the previous section apply. We set the parameters as follows:

$r_o = 0.1$ m $\qquad C = 5.E4$ Pa

$h = 0.1 \frac{kg}{m^2 s} \qquad s_r = 0.$

$\kappa = 10^{-12} m^2 \qquad s_o = 0.8$

$k_\ell = 0.1 \qquad \phi = 0.5$

In addition, we set the problem temperature to a value of 293 K, which will yield a constant $\mu_\ell = 1.E\text{-}3 \frac{kg}{m\text{-}s}$. These parameters yield the same eigenvalues as before. The resulting comparison is shown in Table II, and agreement is within 1.6% for all points checked.

Table II.

	r = 0			r = 0.1		
t	s(0)	$S_{analytic}^{(0)}$	% Error	s(0.1)	$S_{analytic}^{(0.1)}$	% Error
planar:						
101.01	0.7872	0.7891	0.24	0.4417	0.4409	0.1
715.62	0.3981	0.4111	1.6	0.1907	0.1948	0.5
cylindrical:						
77.428	0.7849	0.7838	0.13	0.4352	0.4344	0.094
565.05	0.2524	0.2521	0.03	0.1147	0.1148	0.015

C. Vapor Transport

To check the numerical solution of the two coupled vapor pressure equations, previous numerical calculations [1] are used for comparison. These previous calculations were performed using a different numerical method entirely, but solving identical equations. They involve a steady state boundary value problem, in which water vapor from a fully saturated region diffuses through a dry region of a porous medium towards a boundary maintained at 1 atmosphere of dry air. This problem was run for a wide range of permeabilities and thus incorporates all three vapor transport mechanisms (Darcy flow, binary gaseous diffusion, and Knudsen diffusion). The solutions for a constant temperature of 383 K (110°C) are shown in Figure 4. These solutions were reproduced so closely by PETROS that the two sets of curves would be indistinguishable if plotted together. Parameters used in this run were

$$\mathcal{R} = \sqrt{\frac{8\kappa}{\phi}}$$

$\phi = 0.25$

IG = 0 (planar geometry)

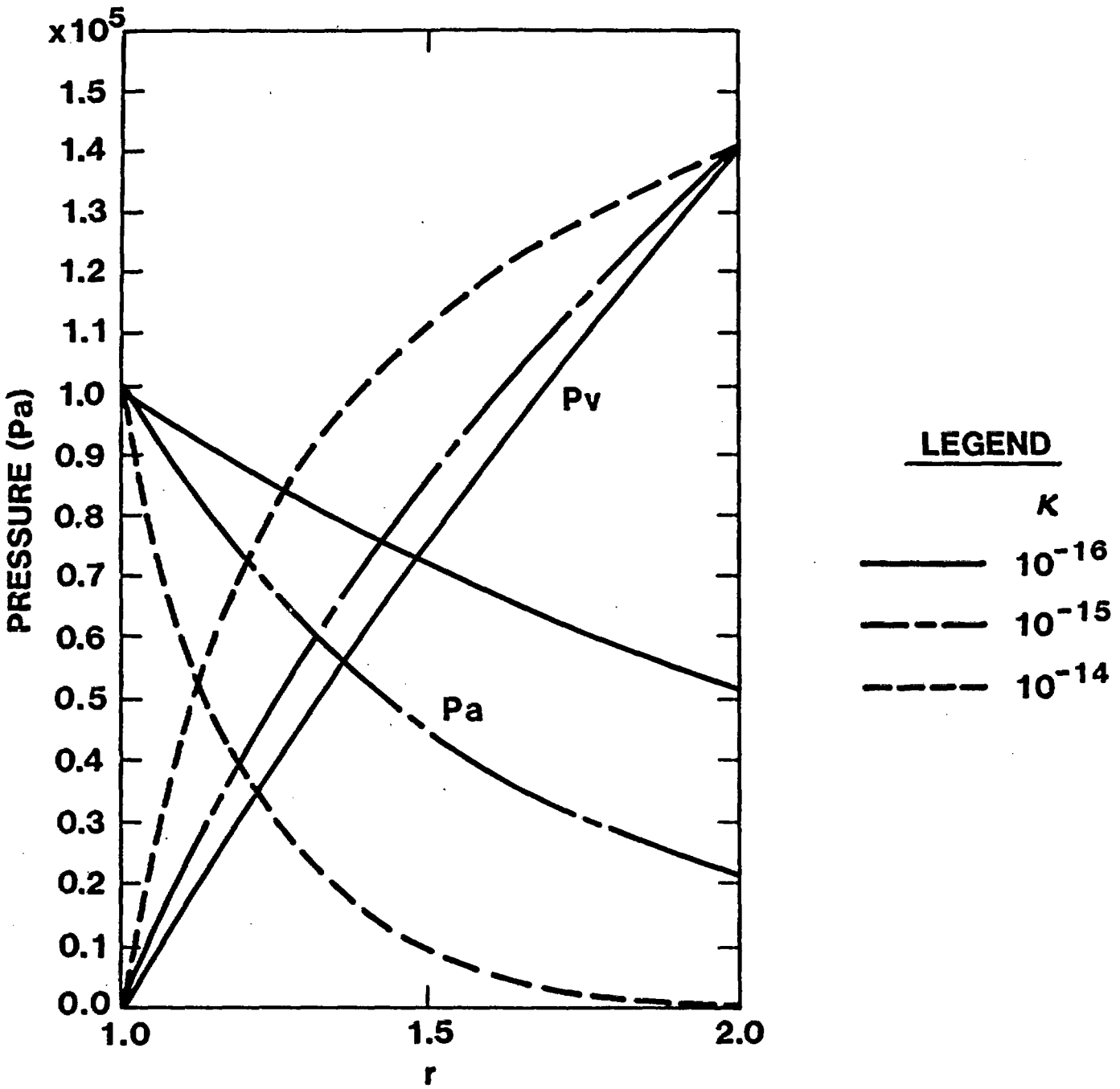

Figure 4. Pressure solver test solutions (permeabilities are in m^2)

APPENDIX A: USER INPUT INSTRUCTIONS

Data Deck

The input data deck is used to specify all information needed to run a problem, with the possible exception of user supplied subroutines or functions (described later). The deck is read in free format (list-directed format) with blanks or commas used to separate numbers. Floating point numbers may be listed in either decimal or scientific notation (E-notation), but must contain a decimal point. Several identical numbers may be listed by the use of an asterisk as a repetition character; thus

4.3 *3 5.6 7 *2

would be read as

4.3 4.3 4.3 4.3 5.6 7 7 7 .

If data for a single card occupies more than 80 columns, a continuation card may be used to list the remaining data, subject to the following rules

1) the break must be between numbers, not dividing a single number,

2) the continuation character "C" should be separated from the last digit read by a space or comma.

Example:

4.3 6.4E4 7.93 5.6E2 C

3.66E5 5.593

Each data section is introduced by a command card which must begin in Column 1 and appear exactly as shown below. Data pertinent to each command card then follows that card in the order designated. These command card "data sections" may appear in any order except that the $REGION card and data must precede the $PROP data section. This is necessary to specify how many regions there are, and consequently how many property values are to be read in on each card. Certain data sections may sometimes be omitted

altogether. For example, if the code is to be run without advancing the temperature, then the $TEMPBC data section, which specifies temperature boundary conditions, may be omitted. Below is a list of input variables for each data section:

$HEADER

PROBLEM DESCRIPTION

The problem description may contain up to 80 characters

$REGION

NREG,IG,IREG(2),IREG(3),....IREG(NREG+1)

RR(1),RR(2),....RR(NREG+1)

RATIO(1),RATIO(2),....RATIO(NREG)

These variables are as follows:

NREG = number of problem regions ($1 \leq NREG \leq 10$)

IG = indicator for problem geometry

$$= \begin{cases} 0 & \text{planar} \\ 1 & \text{cylindrical} \\ 2 & \text{spherical} \end{cases}$$

IREG(L) = mesh point number that begins region L. IREG(1) = 1 always, so is not input. IREG(NREG+1) = IMAX the total number of mesh points in the problem (≤ 101)

RR(L) = spatial position that begins region L

RR(1) = left boundary position

RR(NREG+1) = right boundary position

RATIO(L) = zoning ratio for region L

>1 mesh size increasing with r

<1 mesh size decreasing with r

=1 mesh size constant

$INIT

TEMP,SAT (or "VAR")

IEDIT,ISTOP,TSTOP,DTINIT

TEMP = initial constant problem temperature

SAT = initial constant problem saturation

If initial conditions for S,T are not constant these variables are replaced by a card containing "VAR" in columns 1-3, and a subroutine ICOND provided (see Appendix B).

IEDIT = number of time steps between edits

ISTOP = total number of problem time steps allowed (typically a few hundred)

TSTOP = maximum problem time (sec)

Execution halts and an edit is displayed whenever ISTOP time steps have been computed or real problem time = TSTOP, whichever comes first.

DTINIT = initial time step

This variable should usually be set smaller than the user thinks is appropriate, in order to get the run off to a smooth start. If initially set too small, the time step will accelerate to more reasonable values in just a few steps.

$SATBC

ILEFT,IRIGHT,VLEFT,VRIGHT (or "VAR")

ILEFT = type of saturation boundary condition (B.C.) at left boundary ($r = r_{min}$)

0 -- Dirichlet B.C., s = constant

1 -- Neumann B.C., liquid mass flux = constant

IRIGHT = type of saturation boundary condition at right boundary ($r = r_{max}$)

0 -- Dirichlet B.C., s = constant

1 -- Neumann B.C., liquid mass flux = constant

VLEFT = value of left boundary condition (either saturation or mass flux). Mass flux positive in the direction of increasing r.

VRIGHT = value of right boundary condition (either saturation or mass flux). Mass flux positive in the direction of increasing r.

If either boundary condition is not one of these types, this card is replaced by a card containing "VAR" in Columns 1-3, and a subrouting SATBC provided (see Appendix B).

$VAPBC

EE(1,1),EE(1,2),EE(2,1),EE(2,2),FF(1),FF(2)
G(1,1),G(1,2),G(2,1),G(2,2),H(1),H(2)
} (or "VAR")

The matrices EE and G and vectors FF and H are as defined in Equations (51) and (52). For more general boundary conditions, replace both cards by a single card containing "VAR" in Columns 1-3, and provide subroutine VAPBC (see Appendix B).

$TEMPBC

ILEFT,IRIGHT,VLEFT,VRIGHT (or "VAR")

ILEFT = temperature boundary condition type at left boundary ($r = r_{min}$)

0 -- Dirichlet B.C. T = constant

1 -- Neumann B.C. heat flux = constant

IRIGHT = temperature boundary condition type at right boundary ($r = r_{max}$)

0 -- Dirichlet B.C. T = constant

1 -- Neumann B.C. heat flux = constant

VLEFT = value of left boundary condition (either temperature or heat flux). Heat flux positive in the direction of increasing r.

VRIGHT = value of right boundary condition (either temperature or heat flux). Heat flux positive in the direction of increasing r.

If either boundary condition is not one of these types, this card is replaced by a card containing "VAR" in Columns 1-3, and a subroutine TEMPBC provided (see Appendix B).

$PROP

CAPPA(1),CAPPA(2),....CAPPA(NREG)

PHI(1),PHI(2),....PHI(NREG)

PORE(1),PORE(2),....PORE(NREG)

PC(1),PC(2),....PC(NREG) (or "VAR")

IRELW,IRELG

TCWET(1),TCWET(2),....TCWET(NREG)

TCDRY(1),TCDRY(2),....TCDRY(NREG) } (or "VAR")

RCPMAT(1),RCPMAT(2),....RCPMAT(NREG)

CAPPA(L) = permeability of region L in m^2

PHI(L) = porosity of region L

PORE(L) = average pore radius for region L. This parameter is denoted $\mathscr{R}$ in Equation (65) and is used only to compute the Knudsen diffusion coefficient

PC(L) = capillary pressure for region L. This number is used to scale the Leverett curve included internally in the code. If the user wishes to supply a curve, this card is replaced by a single card containing "VAR" in Columns 1-3 and functions PCAP and DPCDS provided (see Appendix B).

IRELW =
- 0 internally supplied relative permeability curve for water
- 1 user supplied relative permeability curve for water. Requires function XKL (see Appendix B)

IRELG =
- 0 internally supplied relative permeability curve for gas
- 1 user-supplied relative permeability curve for gas. Requires function XKV (see Appendix B)

TCWET(L) = thermal conductivity for fully saturated medium (region L)

TCDRY(L) = thermal conductivity for dry medium (region L). If the user wishes to supply a variable thermal conductivity, both cards are replaced by a single card containing "VAR" in columns 1-3, and function TC provided (see Appendix B)

RCPMAT(L) = product of density and specific heat for matrix material only (region L). Units are J/(m**3-K)

```
$CON
I,CON(I)
J,CON(J)
  .
  .
  .
etc.
DONE
```

This data section is optional and allows the user to change problem constants stored in the equivalenced arrays CON and ICON. A complete list of problem constants accessible to the user as well as their default values is included in Appendix C. (Other CON array elements are used for storing data read in from the data deck.)

I = array index (see Appendix C)

CON(I) = desired value (may be fixed or floating point depending on parameter)

DONE = literal occupying Columns 1-4 to inform data deck reader that all data from this section has been read

$END

Data deck terminator (mandatory)

Control Cards

The existing version of PETROS is written in FORTRAN IV, and intended for execution on the CDC 7600 system. No external math libraries are required unless the user desires to use additional software (such as spline-fitting routines) to compliment some user-defined subroutine. The user then needs only to compile the source code plus any user-defined functions or subroutines and then execute. For Sandia users this is accomplished on the CDC 7600 using the following job stream:

```
ATTACH,FXMATH.
FTN,I=INPUT,L=0.
FTN,I=INPUT.
LIBRARY,FXMATH.
LGO,PL=77000.
```

The second FTN card is only included if a file of user-supplied subroutines or functions is intended to be used. The FXMATH cards are only needed if such routines are needed by user-defined functions. The input records are then inserted in the order

PETROS source	(record #1)
user-supplied FORTRAN routines	(record #2)
data deck	(record #3)

APPENDIX B: USER-SUPPLIED ROUTINES

Below is a list of optional subroutines or functions that the code may expect, depending on the values of certain parameters in the data deck or CON list (see Appendix A or C).

FUNCTION XKL(S,IR)

This function subprogram evaluates the liquid relative permeability XKL in region IR at a point where the saturation is S.

FUNCTION XKV(S,IR)

This function subprogram evaluates the gas phase relative permeability XKV in region IR at a point where the saturation is S.

FUNCTION PCAP(S,IR)

This function subprogram evaluates the capillary pressure (Pa) at a point in region IR where the saturation is S. The user is referred to Appendix D for information on methods for interpolating data tables.

FUNCTION DPCDS(S,IR)

This function subprogram evaluates the derivative with respect to S of the capillary pressure curve provided by FUNCTION PCAP. If it is desired to numerically differentiate using calls to PCAP, care should be taken so that the resulting derivative is not discontinuous at mesh points. Otherwise, it may be advisable to differentiate the curve prior to runtime, store the derivatives in a DATA statement, and interpolate the derivative data table.

FUNCTION TC(S,T,IR)

This function subprogram evaluates the thermal conductivity at a point in region IR whose saturation is S and whose temperature is T.

SUBROUTINE TEMPBC(ILEFT,IRIGHT,VLEFT,VRIGHT,TM,T,S,NB,IMAX)

DIMENSION T(IMAX),S(IMAX)

This subroutine sets the temperature boundary condition parameters ILEFT,IRIGHT,VLEFT,VRIGHT (described in Appendix A) at time TM. The saturation and temperature arrays are S and T. The parameter NB has the value 1 when left boundary conditions are to be set, and 2 when right boundary conditions are to be set. (This feature may be ignored if speed is unimportant.) IMAX = mesh point at right boundary.

SUBROUTINE SATBC(ILEFT,IRIGHT,VLEFT,VRIGHT,TM,T,S,NB,IMAX)

DIMENSION T(IMAX),S(IMAX)

This subroutine sets the saturation boundary condition parameters ILEFT,IRIGHT,VLEFT,VRIGHT (described in Appendix A) at time TM. Other parameters are identical to those for TEMPBC.

SUBROUTINE VAPBC(EE,FF,G,H,TM,T,S,R,DR,NB,IMAX)

DIMENSION EE(2,2),G(2,2),FF(2),H(2),T(IMAX),S(IMAX),R(IMAX),DR(IMAX)

This subroutine evaluates the boundary condition parameters for vapor transport described by Equations (51) and (52). R(I) is the position coordinate at mesh point I, and DR(I) = R(I+1) - R(I). Other parameters are identical to those for TEMPBC. If needed, the user may employ the internal function PVAP(T), which evaluates the equilibrium vapor pressure at temperature T.

```
SUBROUTINE ICOND(IRG,T,S,R,IMAX)
DIMENSION IRG(IMAX),T(IMAX),S(IMAX),R(IMAX)
```

This subroutine evaluates the arrays T, S to define initial conditions on temperature and saturation. R(I) is the position coordinate at mesh point I, is preset by the code, and must not be changed. IRG(I) is the region number between mesh points I and I+1.

APPENDIX C: CON ARRAY

Here we list elements of the equivalenced arrays CON, ICON which may be changed by the user (all units MKS).

ARRAY INDEX	NAME	DEFAULT VALUE	MEANING
1	SS	1.25	square root of mass ratio $\sqrt{\frac{m_a}{m_v}}$ ($\mathscr{S}$ in Equation (23))
2	CEVA	1.E-5	evaporation rate coefficient (c in Equation (3)). Set $c = 10^5 \sqrt{\kappa}$
3	GAMMAL	1000.	density of liquid water
4	EPS	1.E-3	time step truncation error (ε in Equation (57))
5	NPROV	5	number of time step subintervals (Section VI)
6	IAIR	1	0 no air in problem 1 air included
10	G	9.792	acceleration of gravity
11	SMIN	0.005	dryout threshold (for problem termination)
12	CPL	4200.	specific heat for liquid water
13	CPA	1000.	specific heat for air (constant pressure)
14	CPV	2000.	specific heat for water vapor (constant pressure)
15	DLDT	-3573.	derivative of latent heat of vaporization w.r.t. temperature
16	LO	2.256E6	latent heat of vaporization of water at 373 K
17	ERRV	1.E-7	relative error tolerance (water vapor component) for convergence of pressure solver
18	ERRA	1.E-3	relative error tolerance (air component) for convergence of pressure solver

19	XMV	3.E-26	mass of water molecule
20	XMUV	2.E-5	gas mixture viscosity
21	XK	1.38E-23	Boltzmann's constant
78	DLNSDT	-2.42E-3	temperature derivative of ℓn (surface tension)
156	PAGUESS	1.E5	initial guess for air pressure
157	COEF	22.8	coefficient for Knudsen diffusion (see Equation (65))
158	SZERO	0.	saturation cutoff for internal liquid relative permeability model (s_o in Equation (67))
180	SONE	0.005	saturation cutoff for equilibrium model (s_1 in Equation (2))
181	IEQUIL	0	0 non-equilibrium model 1 equilibrium model
182	ALPHA	1.	mixing factor for time step advance algorithm (see Section VI)
183	POWER	4.	exponent of s in expression for liquid relative permeability (n in Equation (67))
204	ITSOLVE	1	indicator for temperature time advancement - 0 no temperature advancement; 1 advances temperature with time
205	ISSOLVE	1	indicator for saturation solution - 0 no call to solver; 1 calls solver
206	IPSOLVE	1	indicator for pressure solution - 0 no call to solver; 1 calls solver
207	ITERMAX	20	max. no. of pressure solver iterations
208	TORT	1.	tortuosity factor for gas diffusion coefficients

APPENDIX D: INTERPOLATION OF DATA TABLES

In many situations, user-defined functions are not easily represented analytically. In such situations it is convenient to provide a data table covering the abscissas of interest, and then interpolate the table. This section is provided to assist the user who may be unfamiliar with such procedures. Two examples are given for supplying the function PCAP using data table interpolation. The first utilizes a quadratic interpolator internal to the code. The second example uses spline fitting routines available from MATHLIB. The latter routines are more complicated to code, but provide continuous first derivatives, since they represent cubic fits.

Example 1

```
      FUNCTION PCAP(S,IR)
      DIMENSION C(11)
      DATA C/2.5,0.68,0.55,0.51,0.48,0.47,0.46,
     X0.45,0.43,0.405,0./,N,DS/11,.1/
      CALL INTERP(C,N,DS,S,PCAP,SLP)
      RETURN
      END
```

The arguments for INTERP are

C	name of array to be interpolated
N	dimension of array C
DS	abscissa spacing (must be uniform)
S	input abscissa - abscissa of C(1)
PCAP	output ordinate
SLP	curve derivative at S

Since INTERP is a quadratic interpolator, the derivative may not be continuous across a mesh point. This condition may pose problems if the slope is to be used (say, for a capillary pressure derivative computation).

Example 2

```
      FUNCTION PCAP(S,IR)
      DIMENSION PARR(15),SARR(15),YP(15),YPP(15),W(50)
      DATA SARR/0.04,0.256,0.398,0.47,0.635,0.725,
     X0.751,0.78,0.815,0.855,0.905,0.961,0.985,0.997,1./
      DATA PARR/1510.,510.,280.,210.,100.,60.,50.,
     X40.,30.,20.,10.,0.,-10.,-20.,-30./
      DATA INIT,N/0,15/
      IF(INIT.EQ.1) GO TO 10
      CALL SPLIFT(SARR,PARR,YP,YPP,N,W,IERR,0,0,0,0,0)
      INIT=1
   10 CALL SPLINT (SARR,PARR,YPP,N,S,PCAP,PD,YPPI,1,KERR)
      PCAP=PCAP*9.714E3
      RETURN
      END
```

More detail concerning the call list for the routines SPLIFT and SPLINT is available in Reference 8.

APPENDIX E: SAMPLE PROBLEM

Consider an initially saturated multilayered soil, as shown in Figure 5. At time t = 0, the soil is dried from above by applying a constant heat flux and a drying atmosphere blowing across the top surface at zero relative humidity. We then wish to compute the saturation and temperature profiles as functions of time.

For this problem, we will utilize the internal capillary pressure function and simply scale the capillary pressure in each region by [7]

$$p_c = \sigma \sqrt{\frac{\phi}{\kappa}} \tag{77}$$

where $\sigma = 0.072$ nt/m is the surface tension of water. Also we will assume that pore radii follow the law [1]

$$\mathscr{R} = \sqrt{\frac{8\kappa}{\phi}} \tag{78}$$

Boundary conditions will be modeled as follows:

1) Heat

We impose a constant heat flux of 1000 W/m^2 at the top boundary and an adiabatic bottom boundary.

2) Mass

Here we use a simple mass flux boundary condition of the form

$$J_\ell = hs(o) \ , \ x = 0, \tag{79}$$

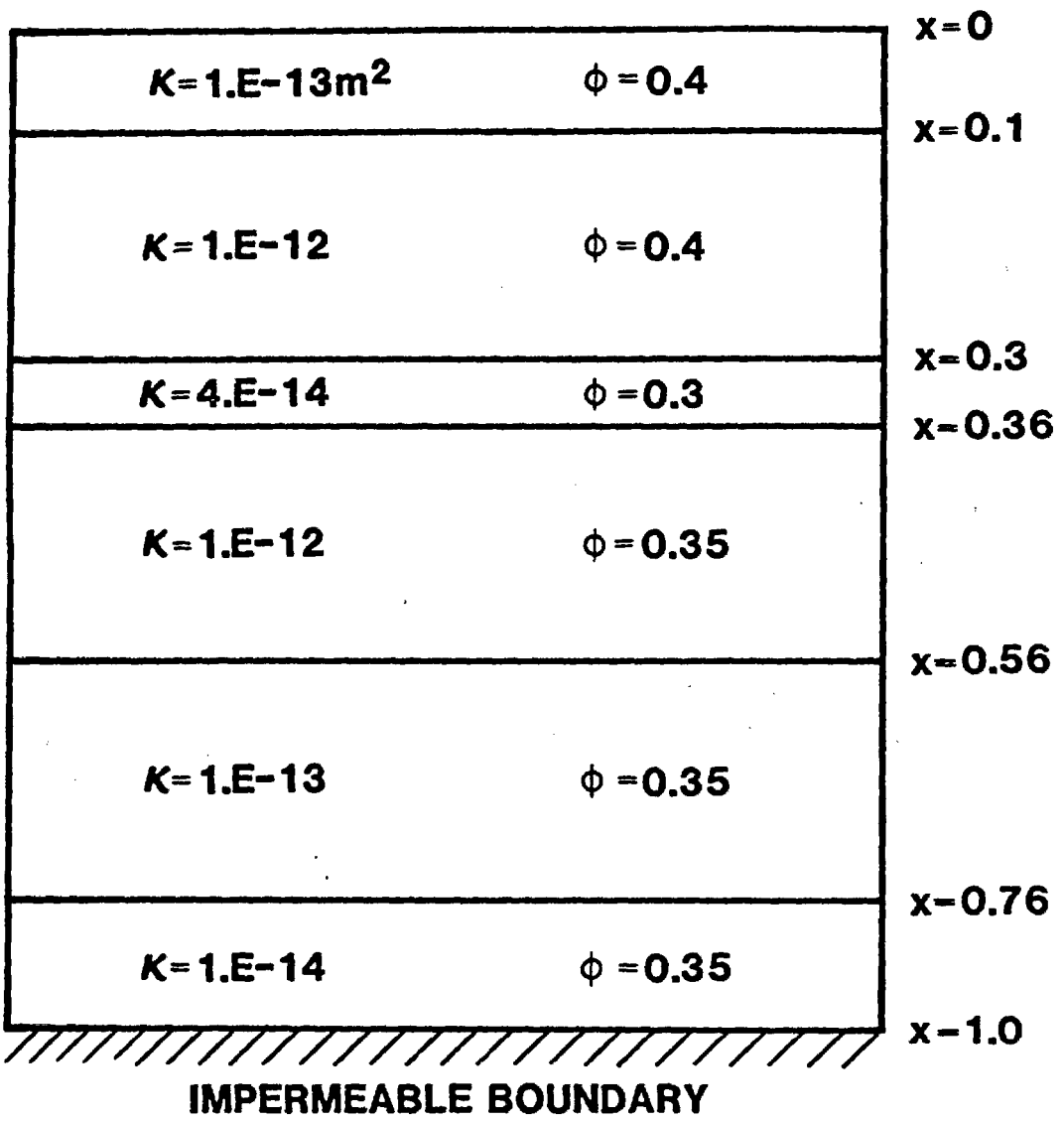

Figure 5. Sample problem: drying of multilayered soil. Numbers shown are soil porosity and permeability. Depths are in m and permeabilities in m^2.

where h is a mass transfer coefficient, and

$$J_\ell = 0 \ , \ x = 1. \tag{80}$$

3) Vapor Flow

Here we set

$$\left.\begin{aligned} p_a &= 1.0 \times 10^5 - p_v \\ p_v &= s(o)^{0.2} p_{vap}(T(0)) \end{aligned}\right\} \quad x = 0 \tag{81}$$

and zero mass flux at x = 1. The exponent 0.2 has been chosen to drop p_v to zero smoothly as the top surface dries.

The data deck and auxiliary subroutines necessary to run this problem are shown in Figures 6 and 7. The zoning was chosen to give approximately uniform zones except near the drying surface where gradients are often steep. This problem ran in 400 seconds on the CDC 7600 computer. The resulting saturation profiles are shown in Figure 8. Code output (Table III) consists of a series of short lines giving time step information, followed by periodic edits. The first two lines of each edit give the step number, time, time step, net evaporation rate and total liquid volume. These latter two quantities are explained in detail in Section VI. For each mesh point are then listed (in order, left to right) position, temperature (K), saturation, water vapor pressure, saturation pressure, air pressure, total pressure $(p_v + p_a)$, water vapor mass flux, liquid mass flux, evaporation rate, and air mass flux. All units are MKS. Note that whereas

```
$HEADER
SAMPLE DRYING PROBLEM
$REGION
6  0  21  40  46  66  86  101
0.  0.1  0.3  0.36  0.56  0.76  1.0
1.09  1.*3  1.07
$INIT
300.  0.999
50  700  5.E5  1.
$SATBC
VAR
$VAPBC
VAR
$TEMPBC
1  1  1000.  0.
$PROP
1.E-13,1.E-12,4.E-14,1.E-12,1.E-13,1.E-14
0.4,0.4,0.3,0.35,*2
1.413E-6,4.46E-6,1.033E-6,4.77E-6,1.51E-6,C
4.77E-7
1.44E5,4.55E4,1.97E5,4.27E4 C
1.35E5,4.27E5
0 0
1.5 *5
0.8 *5
2.E6 *5
$CON
182   0.5
183   3
DONE
$END
```

Figure 6. Data deck for sample problem.

```
      SUBROUTINE SATBC(ILEFT,IRIGHT,VLEFT,VRIGHT,TM,T,S,NB,IMAX)
      DIMENSION T(IMAX),S(IMAX)
      ILEFT=1
      IRIGHT=1
      VLEFT=-1.E-2*S(1)
      VRIGHT=0.
      RETURN
      END

      SUBROUTINE VAPBC(EE,FF,G,H,TM,T,S,R,DR,NB,IMAX)
      DIMENSION EE(2,2),G(2,2),FF(2),H(2),T(IMAX),S(IMAX),R(IMAX),DR(IMAX)
      IF(NB.EQ.2)GO TO 10
      DO 1  I=1,2
      DO 1  J=1,2
      EE(I,J)=0.
    1 FF(I)=0.
      FF(1)=(S(1)**0.2)*PVAP(T(1))
      FF(2)=1.E5-FF(1)
      RETURN
   10 DO 2  I=1,2
      DO 2 J=1,2
      G(I,J)=0.
    2 H(I)=0.
      G(1,1)=1.
      G(2,2)=1.
      RETURN
      END
```

Figure 7. User defined subroutines for sample problem

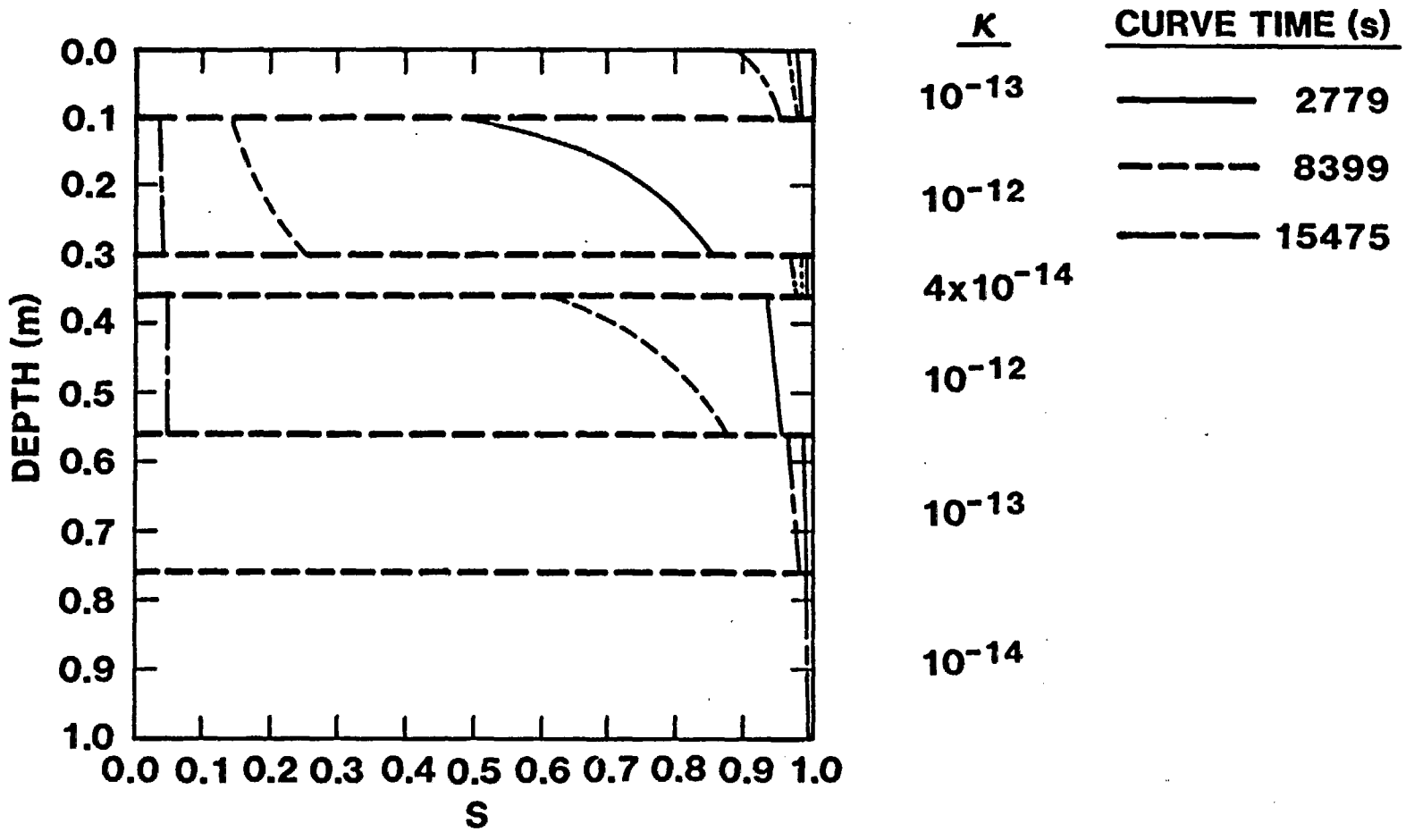

Figure 8. Resulting saturation profiles at different times for sample problem. Permeabilities of soil layers (units are m^2) are shown to aid in interpretation.

Table III. Output from Sample Problem

```
STEP COMPLETE--DT=  .53173E+03
STEP COMPLETE--DT=  .56075E+03
STEP COMPLETE--DT=  .58997E+03
STEP COMPLETE--DT=  .61952E+03
STEP COMPLETE--DT=  .64931E+03
STEP COMPLETE--DT=  .67896E+03
STEP COMPLETE--DT=  .70867E+03
STEP COMPLETE--DT=  .73844E+03
STEP COMPLETE--DT=  .76830E+03
STEP COMPLETE--DT=  .79828E+03
STEP COMPLETE--DT=  .82902E+03
STEP COMPLETE--DT=  .85953E+03
STEP COMPLETE--DT=  .88993E+03
STEP COMPLETE--DT=  .92045E+03
STEP COMPLETE--DT=  .95123E+03
STEP COMPLETE--DT=  .98238E+03
STEP COMPLETE--DT=  .10140E+04
STEP COMPLETE--DT=  .10474E+04
STEP COMPLETE--DT=  .10803E+04
STEP COMPLETE--DT=  .11140E+04
STEP COMPLETE--DT=  .11485E+04
STEP COMPLETE--DT=  .11840E+04

STEP=   50     T=  .24054E+05     DT=  .11840E+04

NET EVAPORATION=  .69573E-15     TOTAL LIQUID VOL=  .21125E-01
```

I	R	T	S	PV	PVAP	PA	P	VFLUX	LFLUX	EVAP	AFLUX
1	0.00000	300.00	.7507	.35457E+04	.35040E+04	.10000E-03	.35457E+04	0.	.14090E-10	0.	0.
2	.00136	300.19	.7506	.35457E+04	.35451E+04	.10000E-03	.35457E+04	-.68560E-08	-.12965E-08	-.49310E-05	-.23269E-17
3	.00278	300.39	.7506	.35887E+04	.35887E+04	.10000E-03	.35887E+04	-.70374E-08	-.88571E-08	-.12423E-06	-.23865E-17
4	.00428	300.60	.7505	.36350E+04	.36350E+04	.10000E-03	.36350E+04	-.71242E-08	-.16242E-07	-.56622E-07	-.24138E-17
5	.00585	300.82	.7504	.36842E+04	.36842E+04	.10000E-03	.36842E+04	-.72149E-08	-.23316E-07	-.56387E-07	-.24423E-17
6	.00750	301.06	.7504	.37365E+04	.37365E+04	.10000E-03	.37365E+04	-.73114E-08	-.30042E-07	-.57100E-07	-.24726E-17
7	.00923	301.30	.7503	.37922E+04	.37922E+04	.10000E-03	.37922E+04	-.74142E-08	-.36387E-07	-.57912E-07	-.25048E-17
8	.01104	301.56	.7503	.38515E+04	.38515E+04	.10000E-03	.38515E+04	-.75238E-08	-.42322E-07	-.58820E-07	-.25391E-17
9	.01295	301.83	.7502	.39147E+04	.39147E+04	.10000E-03	.39147E+04	-.76409E-08	-.47824E-07	-.59835E-07	-.25757E-17
10	.01496	302.11	.7502	.39822E+04	.39822E+04	.10000E-03	.39822E+04	-.77661E-08	-.52877E-07	-.60964E-07	-.26149E-17
11	.01706	302.41	.7502	.40542E+04	.40541E+04	.10000E-03	.40542E+04	-.78994E-08	-.57468E-07	-.61791E-07	-.26565E-17
12	.01927	302.72	.7501	.41311E+04	.41311E+04	.10000E-03	.41311E+04	-.78858E-08	-.61599E-07	.59952E-08	-.26485E-17
13	.02159	303.05	.7501	.42117E+04	.42117E+04	.10000E-03	.42117E+04	-.72747E-08	-.66082E-07	.25697E-06	-.24400E-17
14	.02403	303.40	.7501	.42899E+04	.42898E+04	.10000E-03	.42899E+04	-.75042E-08	-.69759E-07	-.91895E-07	-.25134E-17
15	.02659	303.77	.7501	.43745E+04	.43745E+04	.10000E-03	.43745E+04	-.77550E-08	-.73022E-07	-.95683E-07	-.25935E-17
16	.02927	304.16	.7501	.44665E+04	.44665E+04	.10000E-03	.44665E+04	-.80249E-08	-.76026E-07	-.98033E-07	-.26796E-17
17	.03209	304.57	.7501	.45664E+04	.45664E+04	.10000E-03	.45664E+04	-.83159E-08	-.78827E-07	-.10066E-06	-.27721E-17
18	.03505	305.00	.7501	.46753E+04	.46752E+04	.10000E-03	.46753E+04	-.86303E-08	-.81484E-07	-.10341E-06	-.28719E-17
19	.03816	305.45	.7501	.47939E+04	.47939E+04	.10000E-03	.47939E+04	-.89710E-08	-.84061E-07	-.10690E-06	-.29797E-17
20	.04143	305.94	.7501	.49235E+04	.49235E+04	.10000E-03	.49235E+04	-.93411E-08	-.86626E-07	-.11059E-06	-.30965E-17
21	.04485	306.45	.7501	.50653E+04	.50653E+04	.10000E-03	.50653E+04	-.97441E-08	-.89244E-07	-.11472E-06	-.32234E-17
22	.04845	306.99	.7501	.52208E+04	.52207E+04	.10000E-03	.52208E+04	-.10184E-07	-.91982E-07	-.11932E-06	-.33615E-17
23	.05223	307.56	.7501	.53915E+04	.53915E+04	.10000E-03	.53915E+04	-.10666E-07	-.94905E-07	-.12447E-06	-.35124E-17

the primitive variables p,s, and T are defined at mesh points, all fluxes are defined halfway between mesh points. Thus, for example, fluxes listed opposite node 23 would be appropriate at the control volume boundary between nodes 23 and 24. For this reason, fluxes opposite IMAX are undefined and always listed as zero. In a similar manner, the evaporation rate is defined at interior mesh points and listed as zero at I=1 and IMAX.

APPENDIX F: CODE MESSAGES

PETROS contains a modest number of error messages and execution diagnostics which can aid the user as to what is taking place during an execution. They are as follows:

Input Errors	Interpretation
COMMAND CARD ERROR	One of the command cards is incorrect. See Appendix A.
DATA CARD ERROR IN XXXXX SECTION	One of the data cards in the named section has too many or too few numbers.
INPUT DATA ERROR --- ILLEGAL CHARACTER	The last input line printed has an illegal character.
INPUT DATA ERROR --- MISSING CARD	Reader encountered an EOF when trying to read a data card.
INPUT DATA ERROR --- NO MANTISSA WITH EXPONENT	The letter E has been read without a number preceeding it.
Execution Errors	
NO PRESSURE CONVERGENCE AFTER XX ITERATIONS MAX ERROR IN PV = XXXXX MAX ERROR IN PA = XXXXX	The pressure solver was unable to produce a self-consistent solution for p_v and p_a after ITERMAX iterations. Relative errors listed show how close the solver was to convergence and which variable failed to converge. This diagnostic occurs most often when temperatures pass the boiling point in a high permeability material.
Execution Diagnostics	
STEP COMPLETE -- DT=XXXX	A time step has been successfully completed with time step of XXXX.
REPEATING STEP -- NEW DT = XXXX	New time step too large, step is being repeated with a smaller time step.
PROBLEM TERMINATION -- TOTAL DRYOUT	The problem has been terminated because S < SMIN everywhere.
PRESSURE CONVERGENCE IN XX ITERATIONS	Pressure convergence achieved. This message is only printed when more than 10 iterations were required.

References

1. A. E. Sheidegger, The Physics of Flow Through Porous Media, U. of Toronto Press, Canada, p. 62 (1974).
2. G. R. Hadley, "Theoretical Treatment of Evaporation Front Drying," Int. J. Heat Mass Transf., Vol. 25, No. 10, pp. 1511-1522 (1982).
3. R. K. Wilson, G. R. Hadley, J. W. Nunziato, "Modeling Multiphase Mixtures: The Volume Averaging Method and the Continuum Theory of Mixtures," to be submitted for publication.
4. E. R. G. Eckert, R. M. Drake, Jr., Analysis of Heat and Mass Transfer, McGraw Hill (New York) p. 787 (1972).
5. P. J. Roache, Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, NM, p. 28 (1972).
6. P. M. Gresho, R. L. Lee and R. L. Sani, "On the Time Dependent Solution of the Incompressible Navier-Stokes Equations in Two- and Three-Dimensions," Recent Advances in Numerical Methods in Fluids, Vol. I, Pineridge Press, Swansea, U.K. (1979).
7. Handbook of Chemistry and Physics, Chemical Rubber Publishing Company, Cleveland, Ohio, 42nd Edition (1961).
8. K. H. Haskell, W. H. Vandevender, "Brief Instruction for Using the Sandia Mathematical Subroutine Library," Version 8.0, SAND79-2382 (1980).

DISTRIBUTION LIST

B. C. Rusche (RW-1)
Director
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

J. W. Bennett (RW-22)
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

Ralph Stein (RW-23)
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

J. J. Fiore, (RW-22)
Program Management Division
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

M. W. Frei (RW-23)
Engineering & Licensing Division
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

E. S. Burton (RW-25)
Siting Division
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, D.C. 20585

C. R. Cooley (RW-24)
Geosciences & Technology Division
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

T. P. Longo (RW-25)
Program Management Division
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

Cy Klingsberg (RW-24)
Geosciences and Technology Division
Office of Geologic Repositories
U. S. Department of Energy
Forrestal Building
Washington, DC 20585

B. G. Gale (RW-25)
Siting Division
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, D.C. 20585

R. J. Blaney (RW-22)
Program Management Division
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

R. W. Gale (RW-40)
Office of Policy, Integration, and
Outreach
U.S. Department of Energy
Forrestal Building
Washington, D.C. 20585

J. E. Shaheen (RW-44)
Outreach Programs
Office of Policy, Integration and
Outreach
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

J. O. Neff, Manager
Salt Repository Project Office
U.S. Department of Energy
505 King Avenue
Columbus, OH 43201

D. C. Newton (RW-23)
Engineering & Licensing Division
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

O. L. Olson, Manager
Basalt Waste Isolation Project Office
U.S. Department of Energy
Richland Operations Office
Post Office Box 550
Richland, WA 99352

D. L. Vieth, Director (4)
Waste Management Project Office
U.S. Department of Energy
Post Office Box 14100
Las Vegas, NV 89114

D. F. Miller, Director
Office of Public Affairs
U.S. Department of Energy
Post Office Box 14100
Las Vegas, NV 89114

D. A. Nowack (12)
Office of Public Affairs
U.S. Department of Energy
Post Office Box 14100
Las Vegas, NV 89114

B. W. Church, Director
Health Physics Division
U.S. Department of Energy
Post Office Box 14100
Las Vegas, NV 89114

Chief, Repository Projects Branch
Division of Waste Management
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555

Document Control Center
Division of Waste Management
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555

S. A. Mann, Manager
Crystalline Rock Project Office
U.S. Department of Energy
9800 South Cass Avenue
Argonne, IL 60439

K. Street, Jr.
Lawrence Livermore National
Laboratory
Post Office Box 808
Mail Stop L-209
Livermore, CA 94550

L. D. Ramspott (3)
Technical Project Officer for NNWSI
Lawrence Livermore National
Laboratory
P.O. Box 808
Mail Stop L-204
Livermore, CA 94550

W. J. Purcell (RW-20)
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

D. T. Oakley (4)
Technical Project Officer for NNWSI
Los Alamos National Laboratory
P.O. Box 1663
Mail Stop F-671
Los Alamos, NM 87545

W. W. Dudley, Jr. (3)
Technical Project Officer for NNWSI
U.S. Geological Survey
Post Office Box 25046
418 Federal Center
Denver, CO 80225

NTS Section Leader
Repository Project Branch
Division of Waste Management
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555

V. M. Glanzman
U.S. Geological Survey
Post Office Box 25046
913 Federal Center
Denver, CO 80225

P. T. Prestholt
NRC Site Representative
1050 East Flamingo Road
Suite 319
Las Vegas, NV 89109

J. B. Wright
Technical Project Officer for NNWSI
Westinghouse Electric Corporation
Waste Technology Services Division
Nevada Operations
Post Office Box 708
Mail Stop 703
Mercury, NV 89023

M. E. Spaeth
Technical Project Officer for NNWSI
Science Applications
International, Corporation
2769 South Highland Drive
Las Vegas, NV 89109

ONWI Library
Battelle Columbus Laboratory
Office of Nuclear Waste Isolation
505 King Avenue
Columbus, OH 43201

SAIC-T&MSS Library (2)
Science Applications
International, Corporation
2950 South Highland Drive
Las Vegas, NV 89109

W. M. Hewitt, Program Manager
Roy F. Weston, Inc.
2301 Research Blvd., 3rd Floor
Rockville, MD 20850

W. S. Twenhofel, Consultant
Science Applications
International, Corp.
820 Estes Street
Lakewood, CO 80215

H. D. Cunningham
General Manager
Reynolds Electrical &
Engineering Co., Inc.
Post Office Box 14400
Mail Stop 555
Las Vegas, NV 89114

A. E. Gurrola
General Manager
Energy Support Division
Holmes & Narver, Inc.
Post Office Box 14340
Las Vegas, NV 89114

T. Hay, Executive Assistant
Office of the Governor
State of Nevada
Capitol Complex
Carson City, NV 89710

J. A. Cross, Manager
Las Vegas Branch
Fenix & Scisson, Inc.
Post Office Box 15408
Las Vegas, NV 89114

R. R. Loux, Jr., Director (3)
Nuclear Waste Project Office
State of Nevada
Capitol Complex
Carson City, NV 89710

N. E. Carter
Battelle Columbus Laboratory
Office of Nuclear Waste Isolation
505 King Avenue
Columbus, OH 43201

C. H. Johnson, Technical
Program Manager
Nuclear Waste Project Office
State of Nevada
Capitol Complex
Carson City, NV 89710

John Fordham
Desert Research Institute
Water Resources Center
Post Office Box 60220
Reno, NV 89506

Dr. Martin Mifflin
Desert Research Institute
Water Resources Center
Suite 1
2505 Chandler Avenue
Las Vegas, NV 89120

Department of Comprehensive Planning
Clark County
225 Bridger Avenue, 7th Floor
Las Vegas, NV 89155

Planning Department
Nye County
Post Office Box 153
Tonopah, NV 89049

Lincoln County Commission
Lincoln County
Post Office Box 90
Pioche, NV 89043

Economic Development Department
City of Las Vegas
400 East Stewart Avenue
Las Vegas, NV 89101

Community Planning and Development
City of North Las Vegas
Post Office Box 4086
North Las Vegas, NV 89030

Director of Community Planning
City of Boulder City
Post Office Box 367
Boulder City, NV 89005

City Manager
City of Henderson
Henderson, NV 89015

Commission of the European Communities
200 Rue de la Loi
B-1049 Brussels
BELGIUM

N. A. Norman
Project Manager
Bechtel National Inc.
P. O. Box 3965
San Francisco, CA 94119

R. Harig
Parsons Brinckerhoff Quade & Douglas, Inc.
1625 Van Ness Ave.
San Francisco, CA 94109-3678

V. J. Cassella (RW-22)
Office of Geologic Repositories
U.S. Department of Energy
Forrestal Building
Washington, DC 20585

Technical Information Center
Roy F. Weston, Inc.
2301 Research Boulevard, Third Floor
Rockville, MD 20850

K. Pruess
Earth Sciences Department
Lawrence Berkeley Laboratory
Berkeley, CA 94720

1510 J. W. Nunziato
1511 G. G. Weigand
1511 R. E. Benner
1511 N. E. Bixler
1511 R. R. Eaton
1511 G. R. Hadley (10)
1511 M. J. Martinez
1511 R. K. Wilson
1512 J. C. Cummings
1513 D. W. Larson
1520 D. J. McCloskey
1530 L. W. Davison
1540 W. C. Luth
6300 R. W. Lynch
6310 T. O. Hunter
6310 NNWSICF

6311 L. W. Scully
6311 L. Perrine (2)
6312 F. W. Bingham
6312 N. K. Hayden
6312 R. R. Peters
6313 T. E. Blejwas
6313 E. A. Klavetter
6313 R. M. Zimmerman
6314 J. R. Tillerson
6314 A. J. Mansure
6315 S. Sinnock
6330 W. D. Weart
6331 A. R. Lappin
6332 L. D. Tyler
6332 WMT Library (20)
6334 D. R. Anderson

6430 N. R. Ortiz
3141 C. M. Ostrander (5)
3151 W. L. Garner (3)
8024 M. A. Pound
DOE/TIC (28)
(3154-3, C. H. Dalin)

Org.	Bldg.	Name	Rec'd by	Org.	Bldg.	Name	Rec'd by

Sandia National Laboratories