Preliminary Assessment of Injection, Storage, and Recovery of Freshwater in the Lower Hawthorn Aquifer, Cape Coral, Florida

By Vicente Quiñones-Aponte *and* Eliezer J. Wexler

U.S. Geological Survey

Water-Resources Investigations Report 94-4121

Prepared in cooperation with the

City of Cape Coral and the
South Florida Water Management District

U.S. DEPARTMENT OF THE INTERIOR
MARCH 3, 1849

Tallahassee, Florida
1995

U.S. DEPARTMENT OF THE INTERIOR
BRUCE BABBITT, Secretary

U.S. GEOLOGICAL SURVEY
Gordon P. Eaton, Director

For additional information
write to:

District Chief
U.S. Geological Survey
Suite 3015
227 N. Bronough Street
Tallahassee, FL 32301

Copies of this report can be
purchased from:

U.S. Geological Survey
Earth Science Information Center
Open-File Reports Section
Box 25286, MS 517
Denver Federal Center
Denver, CO 80225

CONTENTS

FIGURES

TABLES

CONVERSION FACTORS, VERTICAL DATUM, AND ADDITIONAL ABBREVIATIONS

Multiply	By	To obtain
millimeter (mm)	0.03937	inch
millimeter per year (mm/yr)	0.03937	inch per year
meter (m)	3.281	foot
meter per second (m/s)	3.281	foot per second
meter per day (m/d)	3.281	foot per day
kilometer (km)	0.6214	mile
square meter (m^2)	10.76	square foot
meter squared per second (m^2/s)	10.76	foot squared per second
meter squared per day (m^2/d)	10.76	foot squared per day
square kilometer (km^2)	0.3861	square mile
cubic meter (m^3)	35.31	cubic foot
cubic meter (m^3)	264.2	gallon
cubic meter per second (m^3/s)	264.2	gallon per second
cubic meter per day (m^3/d)	264.2	gallon per day
liter per second per meter (L/s/m)	4.831	gallon per minute per foot
kilogram per meter per second (kg/m/s)	0.6716	pound mass per foot per second
kilogram per meter per second squared ($kg/m/s^2$)	0.6716	pound mass per foot per second squared
kilogram per cubic meter (km/m^3)	0.0624	pound per cubic foot

Sea level: In this report, "sea level" refers to the National Geodetic Vertical Datum of 1929 (NGVD of 1929)—a geodetic datum derived from a general adjustment of the first-order level nets of both the United States and Canada, formerly called Sea Level Datum of 1929.

The standard unit for transmissivity (T) is cubic meter per day per square meter times meter of aquifer thickness. This mathematical expression reduces to meter squared per day.

Temperature in degrees Celsius (°C) can be converted to degrees Fahrenheit (°F) as follows:
°F = 1.8(°C) + 32

Additional Abbreviations

RO = reverse osmosis
SISRF = subsurface injection, storage, and recovery of freshwater
SUTRA = Saturated-Unsaturated **TRA**nsport
mg/L = milligrams per liter

Preliminary Assessment of Injection, Storage, and Recovery of Freshwater in the Lower Hawthorn Aquifer, Cape Coral, Florida

By Vicente Quiñones-Aponte *and* Eliezer J. Wexler

Abstract

A preliminary assessment of subsurface injection, storage and recovery of fresh canal water was made in the naturally brackish lower Hawthorn aquifer in Cape Coral, southwestern Florida. A digital modeling approach was used for this preliminary assessment, incorporating available data on hydrologic conditions, aquifer properties, and water quality to simulate density-dependent ground-water flow and advective-dispersive transport of a conservative ground-water solute (chloride ion).

A baseline simulation was used as reference to compare the effects of changing various operational factors on the recovery efficiency. A recovery efficiency of 64 percent was estimated for the baseline simulation. Based on the model, the recovery efficiency increases if the injection rate and recovery rates are increased and if the ratio of recovery rate to injection rate is increased. Recovery efficiency decreases if the amount of water injected is increased; slightly decreases if the storage time is increased; is not changed significantly if the water is injected to a specific flow zone; increases with successive cycles of injection, storage, and recovery; and decreases if the chloride concentrations in either the injection water or native aquifer water are increased. In everal hypothetical tests, the recovery efficiency fluctuated between 22 and about 100 percent.

Two successive cycles could bring the recovery efficiency from 60 to about 80 percent. Interlayer solute mass movement across the upper and lower boundaries seems to be the most important factor affecting the recovery efficiency. A sensitivity analysis was performed applying a technique in which the change in the various factors and the corresponding model responses are normalized so that meaningful comparisons among the responses could be made. The general results from the sensitivity analysis indicated that the permeabilities of the upper and lower flow zones were the most important factors that produced the greatest changes in the relative sensitivity of the recovery efficiency. Almost equally significant changes occurred in the relative sensitivity of the recovery efficiency when all porosity values of the upper and lower flow zones and the leaky confining units and the vertical anisotropy ratio were changed.

The advective factors are the most important in the Cape Coral area according to the sensitivity analysis. However, the dispersivity values used in the model were extrapolated from studies conducted at the nearby Lee County Water Treatment Plant, and these values might not be representative of the actual dispersive characteristics of the lower Hawthorn aquifer in the Cape Coral area.

INTRODUCTION

Cape Coral, a coastal suburban community in western Lee County (fig. 1), is a fast growing city in southwestern Florida, with the population increasing at a rate of 8.5 percent during the year ending in April 1989 (City of Cape Coral, Planning Division, written commun., 1989). The city had less than 500 residents in 1960, but became the largest city in Lee County by 1983. The number of permanent residents in 1990 was estimated at more than 73,600. Temporary residents from the northern United States and Canada typically increase the population by about 20 percent during the winter months (City of Cape Coral, Planning Division, 1988).

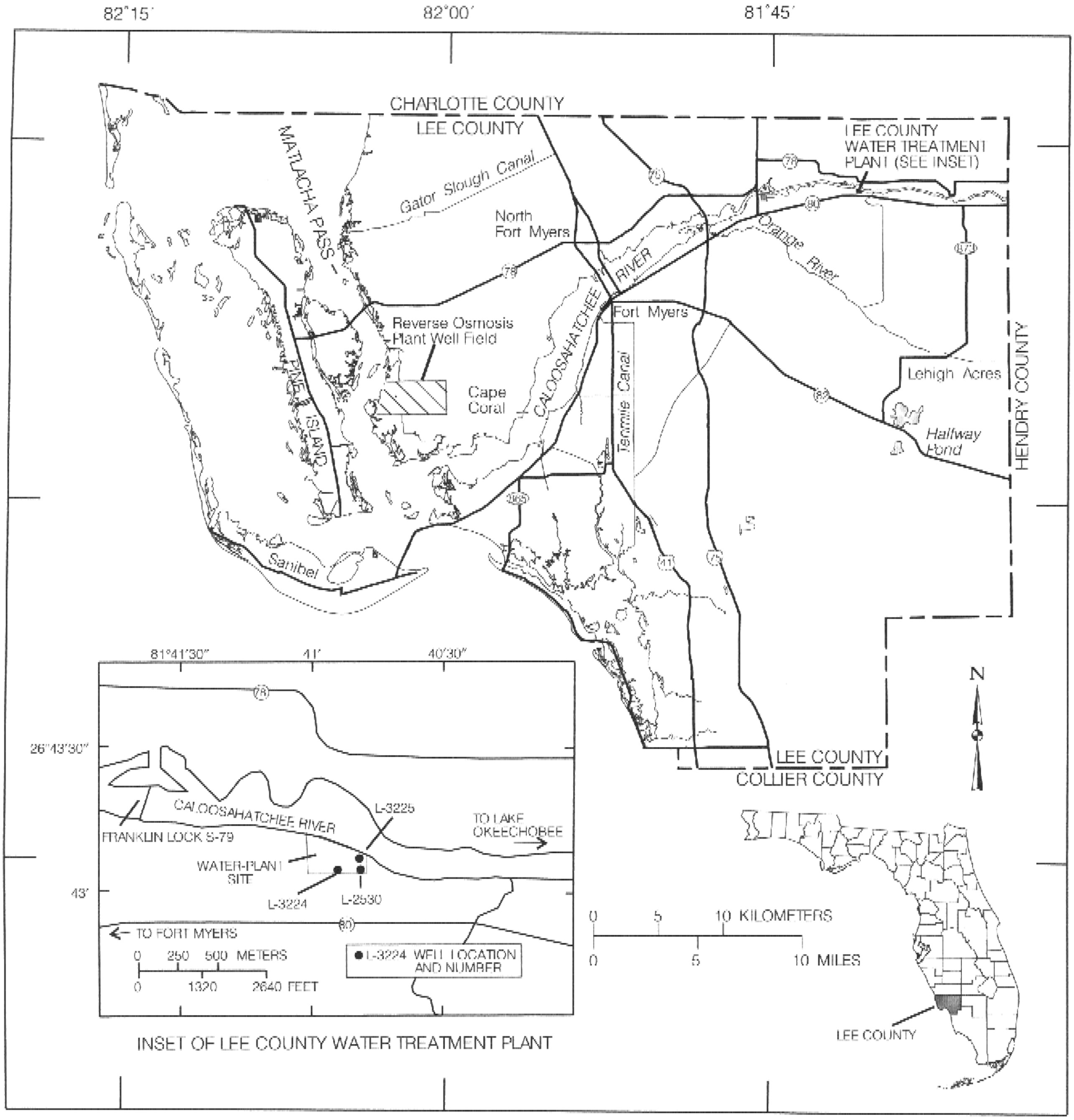


Figure 1. Location of the Cape Coral study area, wells, and the Lee County Water Treatment Plant site.

The rapidly increasing population has placed a stress on the potable water supply for Lee County. The upper Hawthorn aquifer (also referred to as the mid-Hawthorn aquifer) is the principal source of fresh ground water in Cape Coral. This aquifer is moderately permeable and has been subjected to severe drawdowns, particularly during a recent 3-year drought period (1989-91). At present (1994), the most reliable municipal water supply to Cape Coral (and nearby Pine Island) is brackish water from the lower Hawthorn aquifer that is treated at a 52,990 m^3/d reverse-osmosis (RO) plant. Drawdowns in this moderately permeable aquifer have also been substantial. Increased population and water demands in Charlotte County to the north and upgradient of Cape Coral could have an effect on the amount of water available in the two aquifers.

Demand for water is seasonal with peak use occurring during the dry season (November-April) when monthly precipitation averages less than 51 mm (National Oceanic and Atmospheric Administration, 1944-88). Lawn, golf course, agricultural irrigation, and public-supply demands are highest during this period. Temporary water-use restrictions have been implemented occasionally during recent years because of drought conditions and could become permanent as the demand for water becomes more acute.

Alternative water supplies or a means of augmenting existing supplies is a major concern to water-management officials. For this reason, the U.S. Geological Survey, in cooperation with the City of Cape Coral and the South Florida Water Management District, began a study in October 1986 to assess the feasibility of subsurface injection, storage, and recovery of freshwater (SISRF) in Cape Coral. The objectives of the study were to: (1) define the runoff pattern of the freshwater canal system, (2) assess quantities of excess runoff occurring during the wet season, and (3) assess the feasibility of conserving the excess runoff through subsurface storage. This report involves the development and testing of a digital model for assessing hypothetical SISRF tests in Cape Coral.

Although a site seems favorable for SISRF, the recovery efficiency at a particular site can only be determined by establishing a full-scale test facility and conducting full cycle testing under various conditions. Pilot tests are generally too expensive for preliminary assessments, such as this study. However, recent SISRF tests conducted by the U.S. Geological Survey at the Lee County Water Treatment Plant (Fitzpatrick, 1986a) can provide information, which when supplemented with less expensive computer-modeling techniques, yield usable preliminary information on recovery efficiency for an SISRF operation in Cape Coral.

Purpose and Scope

This report presents the results of a preliminary assessment of the subsurface injection, storage, and recovery operation in the lower Hawthorn aquifer in Cape Coral, Fla., using a digital modeling technique. Model simulations were made to assess: (1) recovery efficiencies for injected water; (2) the effect of repeated cycles, length of storage period, injection rates, and volumes of injected water on recovery efficiency; and (3) the relation between recovery efficiencies and the uncertainty in values for hydrogeologic properties. Hydrogeologic data from boreholes in Cape Coral and at the Lee County Water Treatment Plant were used to estimate hydraulic characteristics of the lower Hawthorn aquifer.

A modified SUTRA (Saturated-Unsaturated **TRA**nsport) ground-water flow and solute-transport digital model was used for the simulations. Data from an earlier study at the Lee County Water Treatment Plant were used to calibrate and test the model, and the model was then applied to simulate a hypothetical injection and recovery operation in Cape Coral. Nearly 30 simulations calculated recovery efficiencies for various changes in injection and recovery rates, volumes of water injected, storage time, and solute concentrations.

Description of Study Area

The city of Cape Coral occupies an area of 259 km^2 in Lee County, southwestern Florida (fig. 1). The development of the area, originally a low-lying pineland subject to frequent flooding, began in 1958 and continued to the early 1960's with the construction of a 724-km drainage canal system that interlaces the entire area (Knapp and others, 1984).

The Cape Coral watershed is similar to most southern Florida watersheds in that it is characterized by sheetflow runoff conditions and swamp type vegetation. Surface-water runoff in these watersheds is exclusively derived from rainfall. Rainfall is subdivided into surface-water runoff, evapotranspiration, and natural recharge to the shallow surficial aquifer. Some of the recharge to shallow aquifers returns to the drainage canals in Cape Coral. Many of the canals (totaling about 193 km in length) convey saltwater because they are affected by tidal reaches of the Caloosahatchee River and bays in the Gulf of Mexico. The remaining canals on higher lands convey surface-water runoff collected from the watershed. Although canals that convey fresh surface-water runoff and those that contain saltwater are connected, the movement of saltwater into the freshwater canals is impeded by a series of weir structures with crests that are above sea level.

The freshwater canal system contains two different systems, the north Cape Coral canal system and the south Cape Coral canal system. The canal systems are separated by U.S. Highway 78 with the northern system bounded by Gator Slough. Dredge spoil obtained during canal construction was used to raise land surface as much as 0.62 m in some areas (Fitzpatrick, 1986b). H.R. La Rose indicates that flow through the canals responds to seasonal patterns (U.S. Geological Survey, written commun., 1994). Records for the north Cape Coral canal system indicate that canal flow (not including flood peaks) ranges from 0.85 to 2.83 m^3/s during wet seasons and can be as low as 0.003 m^3/s during dry seasons.

Cape Coral has a subtropical climate with temperatures that are moderated by the Gulf of Mexico. The average annual temperature is 23°C with monthly averages ranging between 28°C in August and 18°C in January. Annual precipitation averages 1,372 mm. Hurricanes have caused damage in the past with high-velocity winds, rainfall, and tidal surges in Lee County. Additional data on local climate are available in a summary report by the Lee County Planning Department (1977).

Subsurface Injection, Storage, and Recovery of Freshwater Concept

Subsurface injection, storage, and recovery of freshwater in saline aquifers underlying southern Florida is a method of water-supply augmentation that has received increased attention in recent years. The SISRF concept is particularly suited for southern Florida where there is: (1) a surplus of freshwater during the wet season; (2) lack of suitable surface storage reservoirs because of the high cost of land, low relief, and high rates of evapotranspiration; and (3) availability of moderately permeable aquifers near the surface which contain brackish water (defined in the table below).

The average monthly rainfall in Cape Coral is more than 178 mm during the wet season (May-October). Most of this water ultimately discharges to the tidal reach of the Caloosahatchee River or Matlacha Pass through an extensive network of drainage canals totaling about 483 km. In the SISRF concept, part of the surface freshwater discharge is intercepted, treated for removal of suspended solids, chlorinated, and then injected through wells into the lower Hawthorn aquifer or deeper aquifers. Water is stored in the aquifers for 3 to 6 months and recovered during the dry season (November-April) to augment supply or meet peak demand. This cyclic procedure of injection, storage, and recovery is repeated on an annual basis.

Success of an SISRF cycle is measured by the recovery efficiency—defined as the volume of mixed injected and native aquifer waters recovered that meets a prescribed chemical standard, expressed as a percentage of the volume of water initially injected (Meyer, 1989). Most recent studies of SISRF, including this study, have assumed the Florida Department of Environmental Protection (1993) recommended level of 250 mg/L (milligrams per liter) for chloride ion as the standard which is equivalent to about 500 to 600 mg/L total dissolved solids. Generally, the degree of water is expressed as a percent of seawater in terms of total dissolved solids. The U.S. Geological Survey has adopted the following classification:

Classification	Total dissolved solids concentration (milligrams per liter)	Percent seawater
Freshwater	<1,000	<2.9
Slightly saline (brackish water)	1,000 - 3,000	2.9 - 8.6
Moderately saline (brackish water)	3,000 - 10,000	8.6 - 29
Very saline (saltwater)	10,000 - 35,000	29 - 100
Brine	>35,000	>100

Factors Affecting Recovery Efficiency

Merritt (1985) and Merritt and others (1983) studied the potential for SISRF in southern Florida and described a number of physical mechanisms that control the recoverability of freshwater and determine the suitability of the receiving aquifer for SISRF. The three dominant processes are buoyancy stratification, mixing due to hydrodynamic dispersion, and downgradient displacement of the injected freshwater.

Buoyancy stratification describes the tendency for the lighter freshwater to rise through the aquifer as it moves outward from the injection well and overrides the denser, native saltwater. Native saltwater in the lower part of the injection zone is drawn into the well during recovery, whereas potable water remains in the upper part of the zone. Buoyancy stratification is controlled by several factors, including: (1) the density contrast between native and injected waters, (2) permeability of the injection zone, and (3) the thickness of the injection zone (Merritt, 1985). These studies indicate that the effect of buoyancy stratification is smaller in relatively thin aquifers of moderate permeability and containing native water of low total dissolved solids concentration. These type of aquifers, therefore, are suitable for SISRF. Confinement of the injection zone by low-permeability materials can also aid in limiting the upward movement of freshwater.

Hydrodynamic dispersion is the mixing of solutes between zones of high and low solute concentrations as a result of molecular diffusion and mechanical dispersion. Molecular diffusion is caused by the flux of solute particles from areas of high solute concentration to areas of low solute concentration. The effect of molecular diffusion is independent of the fluid velocity. Mechanical dispersion is caused by mixing of solutes due to variations in fluid velocities at the microscopic scale. Enhanced mechanical dispersion or macrodispersion is caused by fluid velocity variations resulting from local differences in hydraulic conductivity.

Mechanical dispersion is dependent on the fluid velocity. At the relatively large fluid velocities during injection and recovery cycles, the effects of mechanical dispersion are generally greater than those of molecular diffusion.

Dispersive mixing causes the formation of a transition zone between the native and injected waters. The size of this zone depends on the rate of injection, length of injection period, and the solute-concentration difference between native and injected waters. Because fluid velocities are highest near the well, most of the mixing occurs at the beginning of the injection process. As injection continues, the transition zone moves outward at continually decreasing fluid velocities, leading to decreasing dispersive mixing. Merritt (1985) reported that the growth of the transition zone did not keep pace with the growth of the freshwater zone for long injection periods, thus providing for enhancement of the recovery by injecting larger volumes of water.

The effect of downgradient movement of the freshwater zone on recovery efficiency depends on the length of the cycle and the regional ground-water flow velocities. It is possible to design multiple-well injection systems in situations where flow velocities are high and storage periods are long, similar to those described by Merritt (1985) or Kimbler and others (1975). These multiple well systems can be used to offset the effects of downgradient movement.

The lower Hawthorn aquifer beneath Cape Coral seems to meet most of the criteria for consideration in an SISRF scheme. The aquifer has moderate permeability with mean values representing the vertical distribution of hydraulic conductivity that ranges from 21.3 to 41.4 m/d (estimated using data from Missimer and Associates, Inc., 1985). The aquifer, confined by low-permeability leaky units on the top and bottom, has a thickness of about 60 m. The native water is brackish with chloride concentrations (500-600 mg/L), total dissolved solids concentrations (greater than 1,000 mg/L), and densities (1,001 kg/m^3) not much different from the treated surface water that is proposed to be injected. Rates of regional movement of ground water are generally lower in the northern part of Cape Coral and are higher in the vicinity of the RO wells to the south (fig. 1). Other factors in favor of SISRF are: (1) the artesian heads to be overcome by forced pumping are relatively low; (2) the aquifer is moderately permeable, allowing reasonable rates of pumping be maintained; and (3) well-construction costs would probably not be much higher than for typical water-supply wells in the area.

Another factor that can affect SISRF efficiency is clogging of the aquifer around the injection wellbore. This clogging can be caused by bacterial growth, suspended sediments in the injected water, and chemical precipitation of solutes caused by chemical reactions between the injected fluid and the aquifer material or native water. Removal of sediments and disinfection of the water would likely be required before injecting surface waters. Geochemical models can be used to predict the reactions likely to occur during rock-water interaction and mixing of injected and native waters; additional treatment requirements for the injected water could then be determined. However, the analysis of the well-clogging potential was beyond the scope of this study.

GENERAL HYDROGEOLOGIC SETTING

The geology of Lee County and the Cape Coral area has been described by previous investigators, including Wedderburn and others (1982), Knapp and others (1984), and Missimer and Associates, Inc. (1984). The upper 228 m of sediments in the Cape Coral area are composed of the upper part of the Suwannee Limestone of Oligocene age, the Tampa Limestone and the Hawthorn Formation of Miocene age, the Tamiami Formation of Pliocene age, and undifferentiated deposits chiefly of Pleistocene and Holocene age (fig. 2).

The Suwannee Limestone underlying Cape Coral is predominantly a very pale orange to tan medium-grained limestone, but tends to be sandy and slightly phosphatic (Knapp and others, 1984). The top of the unit generally dips to the south-southeast and ranges from 183 m below sea level at the northern border of Cape Coral to about 229 m below sea level at the southeastern end (Missimer and Associates, Inc., 1984). The base of the unit lies between 274 and 366 m below sea level although few wells in the area penetrate beyond the upper part of the Suwannee Limestone.

Earlier reports by the U.S. Geological Survey divide the Miocene age sediments into two units, the Tampa Limestone and Hawthorn Formation. Recent studies (Wedderburn and others, 1982; Missimer and Associates, Inc., 1984) refer to the Tampa Limestone as the Tampa Formation and, although lithologically distinctive, include these sediments within the Hawthorn Formation.

The Tampa Limestone is present from about 150 to 200 m below land surface and is described by Wedderburn and others (1982) as a very light orange to white,

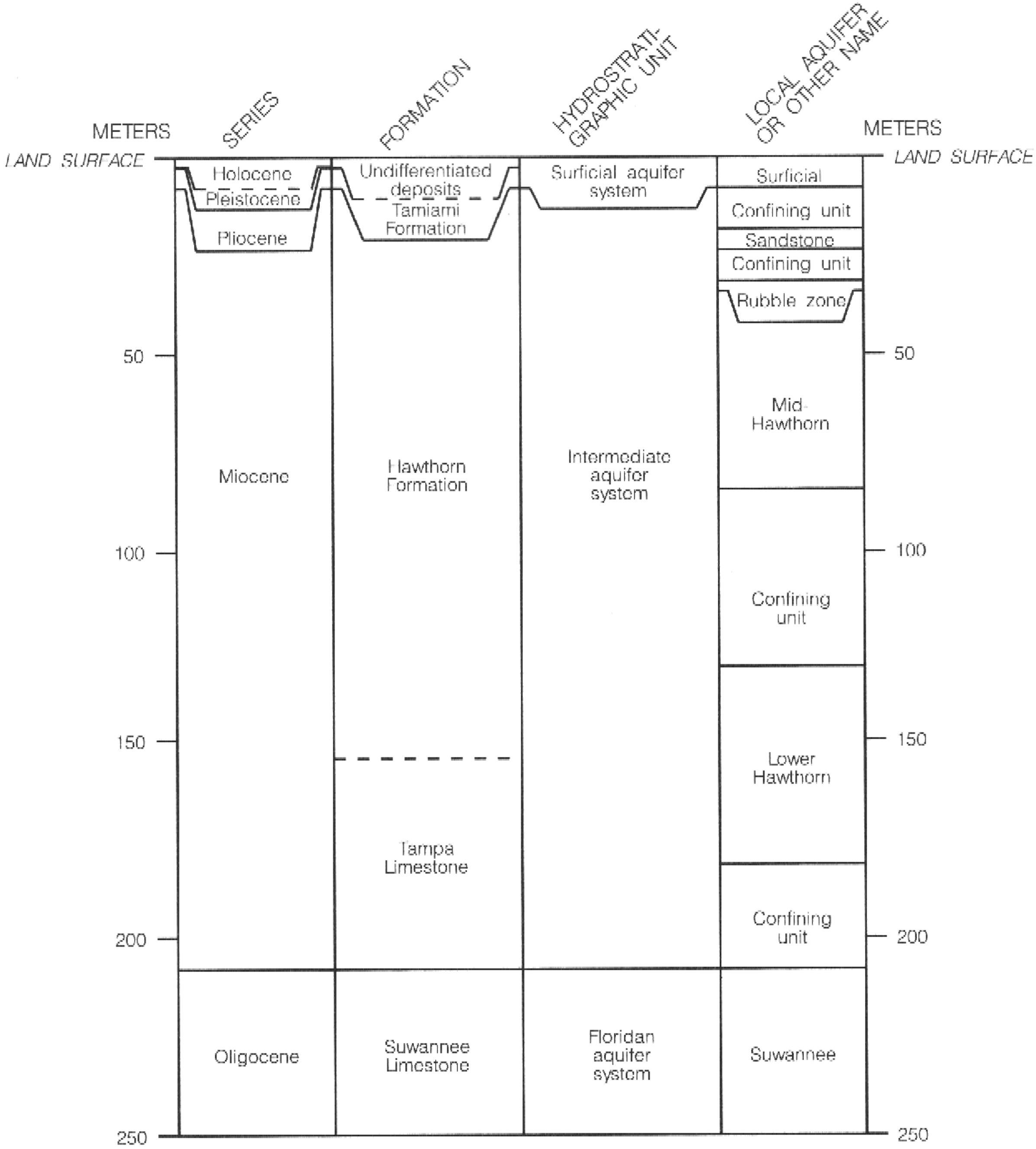


Figure 2. Profile showing geologic formations, hydrostratigraphic units, and local aquifers underlying Cape Coral (modified from La Rose, 1990).

biogenic, micritic, very fine grained limestone with up to 10 percent quartz sand. The Hawthorn Formation is a predominantly clastic unit. The thickness of the formation is about 150 m (Wedderburn and others, 1982). The Hawthorn Formation consists of a series of highly heterogeneous, interbedded clayey phosphatic dolosilts and phosphatic sandy dolomites and limestones (Wedderburn and others, 1982). The upper part of the Hawthorn Formation is a slightly sandy, dolomitic, phosphatic limestone with a maximum thickness of 46 m (Wedderburn and others, 1982). The top of this bed is about 30 m below sea level beneath Cape Coral and dips primarily to the southeast reaching 53 m below sea level in the southeastern corner of Cape Coral. Local names for zones within the upper part of the Hawthorn Formation have been listed by Missimer and Associates, Inc. (1984) and include the Cape Coral clay, Lehigh Acres sandstone, and Fort Myers clay.

Pliocene and Pleistocene age sediments range from 6.1 to 12.2 m thick in the study area (Missimer and Associates, Inc., 1984). Locally, four geologic formations occur within these undifferentiated sediments: (1) the Pamlico sand, (2) the Fort Thompson formation, (3) the Caloosahatchee formation, and (4) the Pinecrest member of the Tamiami Formation. Detailed stratigraphic descriptions are given by Missimer and Associates, Inc. (1984).

Hydrogeology of the Lower Hawthorn Aquifer

The lower Hawthorn aquifer occurs in the lower part of the Hawthorn Formation and the upper part of the Tampa Limestone (fig. 2). The lower Hawthorn aquifer in Cape Coral occurs from about 128 to 188 m below land surface, having an average thickness of 60 m. However, the thickness of its water-yielding zone is less than 30 m (La Rose, 1990). The lower Hawthorn aquifer is confined by thick, leaky clay sequences above and below. Because of this confinement and the higher heads in the upgradient recharge area, this aquifer is considered to be an artesian system with a producing capacity ranging from 0.019 to 0.032 m^3/s in large-diameter wells under natural flow conditions.

Although abundant water is available from the lower Hawthorn aquifer, high chloride concentrations (greater than 500 mg/L) preclude its direct use for public-water supply. Water from the lower Hawthorn aquifer is used to feed RO desalination plants in Cape Coral. According to an interpretation of the hydrogeologic system by La Rose (1990), recharge to the lower Hawthorn aquifer comes from the mid-Hawthorn aquifer north of the study area where the upper confining unit pinches out in Hillsborough, Polk, Manatee, and Hardee Counties.

Hydraulic Characteristics of the Lower Hawthorn Aquifer

Three individual flow zones in the lower Hawthorn aquifer at the Lee County Water Treatment Plant are identified by Fitzpatrick (1986a) using data from geophysical logs (caliper, flow velocity, fluid resistivity, and fluid temperature) during pumping and injection conditions (table 1).

The percentages of flow from the individual zones at the Lee County Water Treatment Plant (table 1) are estimated from caliper/velocity borehole studies conducted by Fitzpatrick (1986a). The aquifer is characteristic of a leaky confined aquifer with hydraulic characteristics as follows (Fitzpatrick, 1986a):

$$T = 7.526 \times 10^{-4}\ m^2/s \text{ to } 8.601 \times 10^{-4}\ m^2/s,$$

$$S = 1 \times 10^{-4}, \text{ and}$$

$$K_v'/b' = 0.01 \text{ per day} = 864 \text{ per second}$$

where,

T is transmissivity;

S is storage coefficient;

K_v' is vertical hydraulic conductivity of the confining beds; and

b' is thickness of the confining beds.

The hydraulic characteristics of the individual flow zones at the Lee County Water Treatment Plant are estimated using the following procedure:

$$Q_T = Q_1 + Q_2 + Q_3 \tag{1}$$

where,

Q_T is the total flow rate through the well; and

Q_i (i = 1,2,3) represents the flow components from the different flow zones.

Table 1. General hydrogeologic characteristics of flow zones and confining units in the lower Hawthorn aquifer at the Lee County Water Treatment Plant and Cape Coral

Location	Flow zones and leaky confining units (meters below land surface)	Thickness (meters)	Percent of flow from this zone	Hydraulic conductivity (meters per second)	Intrinsic permeability (square meters)
Lee County Water Treatment Plant	153.9–160.0	6.1	30	3.775×10^{-5}	3.846×10^{-12}
	160.0–167.6	7.6	5	5.044×10^{-6}	5.140×10^{-13}
	167.6–176.8	9.2	65	5.468×10^{-5}	5.572×10^{-12}
Cape Coral	198.0–213.3	15.3	34	1.065×10^{-4}	1.085×10^{-11}
	213.3–222.5	9.2	2	1.041×10^{-5}	1.061×10^{-12}
	222.5–231.6	9.1	64	3.370×10^{-4}	3.435×10^{-11}

For each flow zone:

$$Q_i = 2\pi r T_i \frac{dh_i}{dr} \qquad (2)$$

where,

r is radial distance from pumping well;
dh_i is the head change in the different flow zones; and
dr is the change in distance from the pumping well.

Assuming no head gradient among the flow zones, $dh_i/dr = dh/dr$, and uniform head in the wellbore:

$$Q_T = 2\pi r (T_1 + T_2 + T_3) \frac{dh}{dr} = 2\pi r T \frac{dh}{dr} \qquad (3)$$

and

$$T = T_1 + T_2 + T_3 = K_1 b_1 + K_2 b_2 + K_3 b_3 . \qquad (4)$$

For example, if T is the composite transmissivity estimated from an aquifer test, assuming that equation 4 can be applied, $Q_i/Q_T = T_i/T$ and $T_i = K_i b_i$ gives the hydraulic conductivity of each zone. If $T = 7.68 \times 10^{-4}$ m^2/s (aquifer test), 30 percent of the total flow (Q_T) comes from zone 1 (flowmeter survey), and this zone has a thickness of 6.1 m:

$$T_1 = \frac{Q_i}{Q_T} T = 0.30 \times 7.68 \times 10^{-4} (\mathrm{m}^2/\mathrm{s}) = 2.30 \times 10^{-4}\ (\mathrm{m}^2/\mathrm{s}),$$

and $$K_1 = \frac{T_1}{b_1} = \frac{2.30 \times 10^{-4}\ (\mathrm{m}^2/s)}{6.1\ \mathrm{m}} = 3.775 \times 10^{-5}\ (\mathrm{m/s}),$$

The hydraulic conductivity (K_i) values for the different flow zones are given in table 1. Aquifer matrix permeability (k_i, intrinsic permeability) values from table 1 are then computed using:

$$k_i = \frac{\mu}{\rho} \frac{K_i}{g} \qquad (5)$$

where,

μ is dynamic viscosity of the fluid [M/LT];
ρ is fluid density [M/L^3]; and
g is gravitational acceleration [L/T^2].

Although the general hydrogeologic framework of the lower Hawthorn aquifer at the two sites (Cape Coral and the Lee County Water Treatment Plant) is similar, the magnitude of the hydraulic characteristics is somewhat different. Analysis of flow velocity and caliper borehole logs (fig. 3) in Cape Coral indicated a similar flow zoning, occurring at different depths below land surface and with different thicknesses and hydraulic coefficients (table 1). The upper flow zone and the low permeability unit seem to be thicker in Cape Coral, but the distribution of flow across these hydrogeologic units is almost the same (table 1).

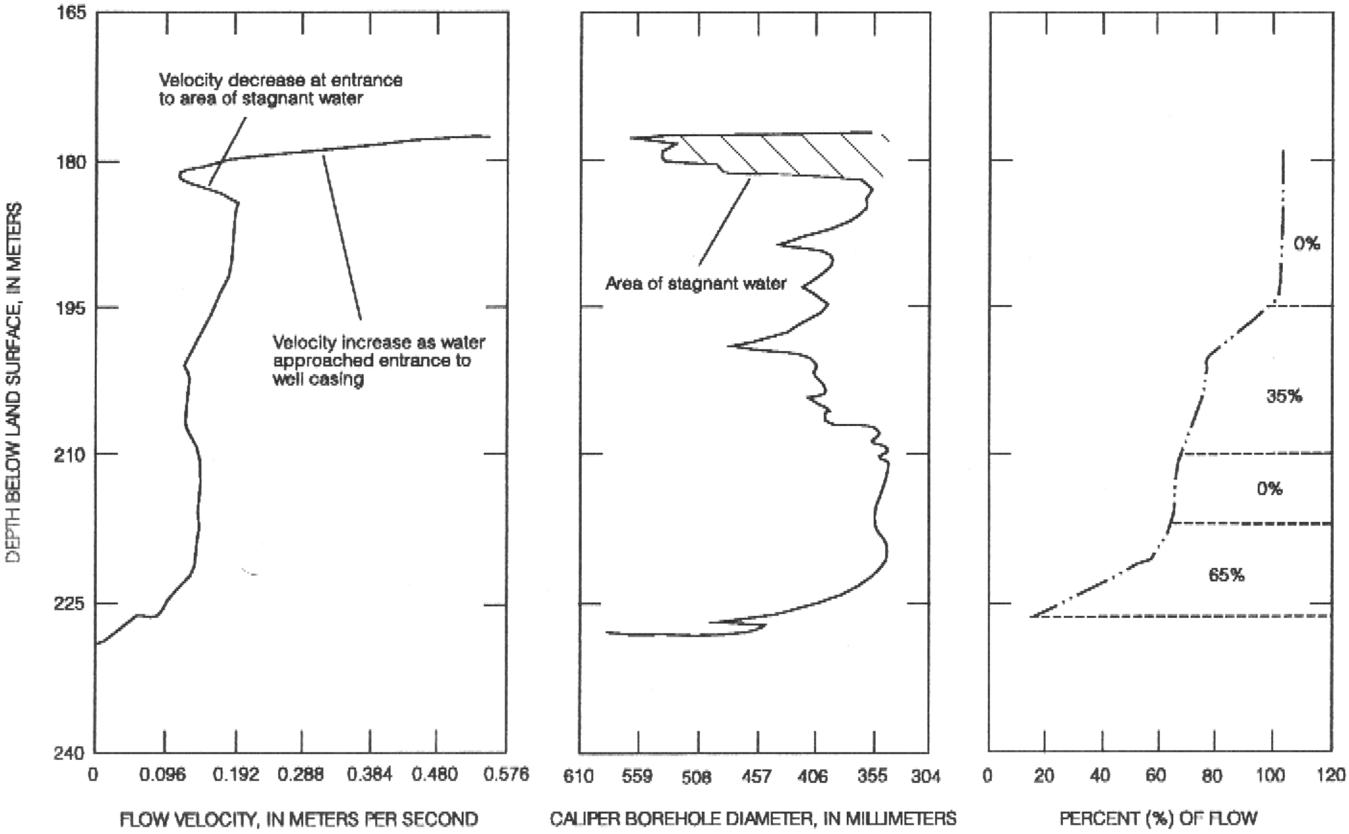


Figure 3. Percent of total flow estimated using velocity and caliper borehole logs for well L-M-2426 at Cape Coral.

Apparent transmissivity values are estimated for several wells in Cape Coral (table 2), using specific capacity values from step-drawdown tests conducted by Missimer and Associates, Inc. (1985), and the empirical equation by Brown (1963). Estimated transmissivity values range from 149 to about 807 m^2/d (fig. 4 and table 2) with a geometric mean value of about 414 m^2/d. Values of hydraulic conductivity and intrinsic permeability are estimated for the lower Hawthorn aquifer in Cape Coral (table 1), using the geometric mean of the transmissivity values and equations 1 to 5.

Table 2. Specific capacity and apparent transmissivity values for wells completed in the lower Hawthorn aquifer at Cape Coral

[Specific capacity values from Missimer and Associates, Inc. (1985); apparent transmissivity values estimated using the empirical equation by Brown (1963)]

Well identification number	Specific capacity (liters per second per meter)	Apparent transmissivity (meters squared per day)
L-M-2417	4.74	496.7
L-M-2418	5.20	546.4
L-M-2419	3.97	409.8
L-M-2420	5.55	583.6
L-M-2421	3.35	347.7
L-M-2422	4.57	496.7
L-M-2423	1.74	149.0
L-M-2424	2.24	223.5
L-M-2425	2.84	273.2
L-M-2426	7.64	807.2
L-M-2427	7.27	782.3
L-M-2428	4.14	397.4
Geometric mean		414.3
Standard deviation		203.7

THEORETICAL BACKGROUND

The ability to assess whether SISRF could be an economical water-supply alternative is enhanced by the capability to predict the movement of water and solutes under the conditions of injection, storage, and recovery. Digital models have been developed by the U.S. Geological Survey and others to simulate the density-dependent flow of ground water and the transport of solutes in ground-water systems. These models can utilize data on fluid and aquifer properties to estimate recovery efficiencies under conditions expected at a particular study area.

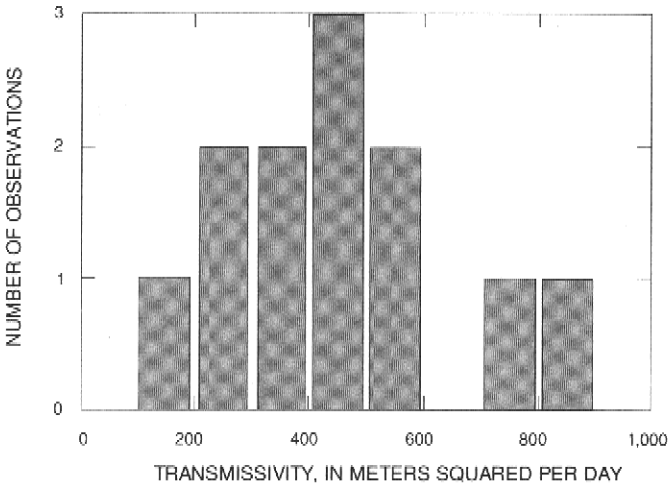


Figure 4. Histogram of apparent transmissivity values estimated from wells tapping the lower Hawthorn aquifer at Cape Coral.

Simulation of density-dependent ground-water flow and solute transport requires the solution of two governing partial differential equations subject to appropriate boundary and initial conditions. The first equation describes transient ground-water flow under conditions where density differences due to solute concentrations can affect flow. The second equation describes the movement and spread of solutes within the flowing ground water using data on the distribution of ground-water velocities obtained by solving the first equation. The two equations are solved iteratively, as the distribution of solute concentrations needed to solve the first equation is initially estimated and updated after solving the second equation. The theoretical background of the governing equations is discussed in the next section.

Density-Dependent Ground-Water Flow Equation

The rate of ground-water flow is assumed to be governed by Darcy's law, which when written in terms of fluid pressure (rather than piezometric head), is:

$$q = -k(\nabla p - \rho g z)/\mu \qquad (6)$$

where,

q is specific discharge (flow rate per unit cross-sectional area) [L/T];
k is the intrinsic permeability of the aquifer materials [L^2];
∇ is the gradient operator [1/L];
p is the fluid pressure [M/LT^2];
ρ is the fluid density [M/L^3];
g is the gravitational acceleration vector [L/T^2];
z is the elevation above a reference datum [L]; and
μ is the dynamic viscosity of the fluid [M/LT].

Using Darcy's law and the principle of conservation of fluid mass, a mass-balance equation can be written as:

$$\frac{\partial (n\rho)}{\partial t} = -\nabla \bullet (\rho q) \pm Q_p \qquad (7)$$

where,

n is aquifer porosity [dimensionless], and

Q_p is mass of fluid injected (+) or withdrawn (-) per unit time per unit volume of aquifer [M/L^3T].

The dependence of fluid density on solute concentration has an important effect on the mass-balance equation, which can be seen by expanding the first term in equation 7:

$$\frac{\partial (n\rho)}{\partial t} = \frac{\rho \partial n \partial p}{\partial p \partial t} + \frac{n \partial \rho \partial p}{\partial p \partial t} + \frac{n \partial \rho \partial c}{\partial c \partial t}, \qquad (8)$$

or

$$\frac{\partial (n\rho)}{\partial t} = \frac{S_s \partial p}{\partial t} + \frac{n \partial \rho \partial c}{\partial c \partial t} \qquad (9)$$

where,

c is solute concentration (mass of solute/mass of water) [dimensionless]; and

S_s is specific pressure storativity of the aquifer

given by $S_s = [(1-n)\,\alpha + n\,\beta]$ for an unconsolidating aquifer [LT^2/M] where,

α is compressibility of the aquifer solid matrix [LT^2/M], and

β is compressibility of water [LT^2/M].

The determination of fluid pressures at any given time, which affects the rates of fluid movement, requires the prior or simultaneous determination of the rate of change in fluid concentration over time. The specific discharge, as determined by Darcy's law, is also dependent on solute concentration through the density and viscosity terms (eq. 6), which is only slightly dependent on solute concentration.

A system of equations, such as equation 7, can be simultaneously solved for a given set of boundary conditions, aquifer properties, fluid densities, and rates of recharge or withdrawals from the aquifer. The solution will be in terms of the pressure at all points in the aquifer. The average pore velocity, v, can then be determined from the distribution of hydraulic head by Darcy's law:

$$v = -\frac{q}{n} \qquad (10)$$

where,

n is the effective porosity of the aquifer [dimensionless].

Advection and Hydrodynamic Dispersion

Movement of solutes through a porous medium is controlled by advection and hydrodynamic dispersion. Advective transport describes the movement of solute particles along the mean direction of fluid flow at a rate equal to the average pore-water velocity. Hydrodynamic dispersion describes the spread of solute particles along and transverse to the direction of average fluid flow in response to molecular diffusion and mechanical dispersion.

Molecular diffusion produces a flux of solute particles from areas of high to low solute concentrations; its effect is independent of the fluid velocity. Mechanical dispersion is the mixing of solutes caused by variations in fluid velocities at the microscopic scale. Velocity variations are caused by several factors, including: (1) velocity distributions within the pore space, (2) variations in pore size, (3) differences in path lengths for different solute particles, and (4) the effect of converging and diverging flow paths (Bear, 1979). Mechanical dispersion is dependent on fluid velocity, and at the relatively large pore-water velocities expected during injection and recovery phases, the effects are greater than those of molecular diffusion. Fluid movement during the storage phase is mainly from buoyancy forces, and at these low velocities, molecular diffusion can have a more significant role in solute movement.

Dispersive flux, J, can be described by Fick's first law as:

$$J = -D_m \nabla c \qquad (11)$$

where,

c is the volumetric concentration of solute [M/L^3]; and

D_m is the second rank tensor containing the coefficients of mechanical dispersion [L^2/T].

Mechanical dispersion coefficients are related to the average pore velocity by the dispersivity of the medium (Scheidegger, 1961). The coefficients of dispersivity are dependent on properties of the medium including permeability, length of a characteristic flow path, and tortuosity. In an isotropic medium (with respect to dispersion), the coefficients of mechanical dispersion can be expressed in terms of two components: (1) longitudinal dispersivity (α_L), which represents dispersion in the direction of the flow path; and (2) transverse dispersivity (α_T), which represents dispersion in the direction perpendicular to the flow path. Transverse dispersivities are usually smaller than longitudinal dispersivities by a factor of 5 to 20 (Freeze and Cherry, 1979).

The nine components of the symmetric mechanical dispersion tensor can be expressed in terms of v (the average pore-water velocity vector) and the velocity components v_x, v_y, and v_z (Bear, 1979). In a system where ground-water flow is horizontal (v_z=0), the components of the mechanical dispersion tensor are:

$$\begin{aligned} D_{xx} &= (\alpha_L v_x^2 + \alpha_T v_y^2) / |v| \\ D_{xy} &= D_{yx} = (\alpha_L - \alpha_T) v_x v_y / |v| \\ D_{yy} &= (\alpha_L v_y^2 + \alpha_T v_x^2) / |v| \\ D_{xz} &= D_{zx} = D_{yz} = D_{zy} = 0 \\ D_{zz} &= \alpha_T |v|. \end{aligned} \qquad (12)$$

For radially symmetric irrational flow (v_Θ=0) systems, subscripts x and y are replaced by r and z, respectively.

The hydrodynamic dispersion tensor can be written as:

$$D_h = D_m + D_d \underline{I} \qquad (13)$$

where,

D_h is the second order hydrodynamic dispersion tensor [L^2/T];
D_m is the mechanical dispersion tensor [L^2/T];
D_d is the coefficient of molecular diffusion [L^2/T]; and
$\underline{I}$ is the identity tensor.

Macrodispersion

Longitudinal dispersivities typically range from 0.100 to 10.00 mm in laboratory experiments with homogeneous materials and have been estimated as much as 90 m from field studies of contaminant plumes (Freeze and Cherry, 1979). The larger values in field studies are related to increased mixing (on a macroscopic scale) because of local variations in aquifer hydraulic and dispersive characteristics.

Most studies of radial injection have assumed that macrodispersive fluxes can be represented by Fick's law with a constant dispersion coefficient. However, recent studies of transport in porous media have indicated that dispersion can increase away from the source and reach an asymptotic value after travel distances of hundreds or thousands of feet (Gelhar and Axness, 1983). Dispersivities are scale dependent at short distances with values increasing away from the contaminant source as larger scale heterogeneities occur (Gelhar and others, 1979). Recent developments in the macrodispersion theory are discussed by Anderson (1984).

In this study, aquifer dispersivity values were estimated from the analysis of field test data from a previous study (Fitzpatrick, 1986a). Values of aquifer dispersivity used in the different simulations and sensitivity analyses are discussed in later sections. Limitations of the advective-dispersive model must be recognized along with the other limitations introduced because of uncertainties in aquifer properties.

Advective-Dispersive Solute-Transport Equation

A version of the variable-density advective-dispersive solute-transport equation modified for saturated flow and conservative solute species presented by Voss (1984) is:

$$\frac{\partial (n\rho c)}{\partial t} = -\nabla \bullet (n\rho v c) + \nabla \bullet [ns(Dd\,\underline{I} + Dm) \bullet \nabla C] + Q' c^* \qquad (14)$$

where,

Q' is the volumetric injection rate per unit area of aquifer [L/T]; and
c^* is volumetric solute concentration in the injected fluid [M/L^3].

When applying equation 14 to freshwater injection in an aquifer, flow can be assumed to be either: (1) radially symmetric about the injection well (regional flow is negligible), or (2) horizontal and the solute concentration and fluid density are vertically uniform (regional flow is considered). In the latter case, the term c represents the vertically averaged concentration at a point in the aquifer. For this study, the first option was used.

The term $Q' c^*$ represents only sources of solute mass. Withdrawals of fluid from the aquifer do not need to be considered in the transport equation because the concentration of solute in the fluid withdrawn from the aquifer c^* is identical to the solute concentration c. The source term from equation 14 is incorporated as part of the boundary conditions.

PRELIMINARY ASSESSMENT OF INJECTION, STORAGE AND RECOVERY OF FRESHWATER

Solution of the two governing partial-differential equations generally requires sophisticated digital models. These models use numerical approximation techniques that determine aquifer pressure and solute concentrations at a finite number of points and at specified time intervals. SUTRA (Saturated-Unsaturated TRAnsport), a computer code based on the Galerkin finite element technique (Voss, 1984), was applied in this study. Modifications were made to the code to compute the solution in terms of a regular rectangular grid with the intention of minimizing computer storage and time (apps. 1 and 2). Appendix 1 contains the hierarchic levels of subprograms in the original SUTRA version and in the modified SUTRA version, hereafter referred to as QSUTRA.

Subprograms PLOT, CONNEC, BANWID, NCHECK, and PINCHB were not included. All of these subprograms, except for PLOT, were used in the original SUTRA version to process information related to the irregularity of element shapes forming the mesh or grid. A new subprogram (FOPEN) was added to open files and assign unit numbers (apps. 1 and 2) (C.I. Voss, U.S. Geological Survey, written commun., 1994). Subprogram SOLVEB, which includes the algorithms to solve the system of equations (eqs. 7 and 15), was substituted by subprograms SOLVEC and LSORA (apps. 1 and 3). SOLVEC uses the incomplete Cholesky-conjugate gradient method (Kuiper, 1987) to solve a system of ground-water flow equations (eq. 7). LSORA uses the line successive overrelaxation method (Young, 1954) to solve a system of solute-transport equations (eq. 15). Some other changes to the code are highlighted in the program listing (app. 2).

QSUTRA was tested by applying it to Henry's (1964) density-dependent flow problem described in Voss (1984, p. 196-203). This problem was selected because it provides a good opportunity to test the capabilities of QSUTRA in solving nonlinearities occurring in variable density flow problems. Comparison of results from QSUTRA and SUTRA for Henry's (1964) problem are presented in appendix 4. As shown in appendix 4, concentration profiles from QSUTRA and SUTRA are identical. Also, QSUTRA and SUTRA estimates of flux across one model boundary compare very well.

Simulations of freshwater injection, storage, and recovery in the lower Hawthorn aquifer were made using the QSUTRA code with a radial coordinates grid. The following assumptions are made: (1) the effect of the background hydraulic gradient is negligible, (2) the aquifer is divided into vertically adjacent layers characterized in the model as homogeneous with respect to the hydraulic and transport characteristics, (3) the hydraulic and transport characteristics are homogeneous along the radial direction of flow, and (4) the aquifer characteristics are isotropic along the horizontal (radial) direction. Assumptions 2 and 3 are made because of the lack of information on the spatial variability of the hydraulic and transport characteristics. Estimates of the transport characteristics of the lower Hawthorn aquifer were made using data from previous freshwater injection tests (Fitzpatrick, 1986a) conducted at the Lee County Water Treatment Plant (fig. 1).

Grid Design

Although the configuration of the lower Hawthorn aquifer at the Lee County Water Treatment Plant and Cape Coral are similar, differences on the thickness of the flow zones and on the magnitude of the hydraulic properties precluded the use of the same model grid for both sites. Two finite-element grids were required. The first grid was used for calibrating and testing the model with data from field tests conducted at the Lee County Water Treatment Plant and documented (Fitzpatrick, 1986a). The second grid was used to represent the hydrogeologic conditions at the Cape Coral site. Transport characteristics obtained from simulating Fitzpatrick's tests were extrapolated to the Cape Coral area.

The Lee County Water Treatment Plant site grid consists of 1,400 elements and 1,491 nodes (fig. 5A), and the Cape Coral grid consists of 2,100 elements and 2,201 nodes (fig. 6A). Both grids extend out radially to 10,384 m (figs. 5A and 6A). The Cape Coral grid was used to conduct hypothetical tests of freshwater injection, storage, and recovery in the lower Hawthorn aquifer in the study area (fig. 1). The grids are very fine (2 m) in the vicinity of the injection well so as to avoid errors associated with numerical dispersion (artificial dispersion introduced by inappropriate spatial discretization) and high aspect ratios (large difference between sides of an element). At a distance of 100 m, element lengths increased to 4 and 8 m at 120 m from the well. Beyond 160 m, element lengths were successively doubled until a maximum length of 4,096 m was reached. The thickness of elements remained constant (2 m). The part of the finite-element grids extending to a distance of 160 m from the injection well is shown in figures 5B and 6B, and the entire finite-element grids are shown in figures 5A and 6A.

Boundary and Initial Conditions

Boundary conditions were set at $r=0$, $r=10,384$ m, $z=144.8$ m below land surface, and $z=184.8$ m below land surface for the Lee County Water Treatment Plant model, and set at $r=0$, $r=10,384$ m, $z=186$ m below land surface, and $z=246$ m below land surface for the Cape Coral model—the limits of the finite-element grids (figs. 5 and 6). Boundaries at the top and bottom of the aquifer (upper and lower limits of the modeled zone) were set constant for pressure and concentration. The solute concentration was set equal to the solute concentration of the native water at these boundaries, and the pressures were set equal to the hydrostatic pressures at the specific depths where the boundaries were located.

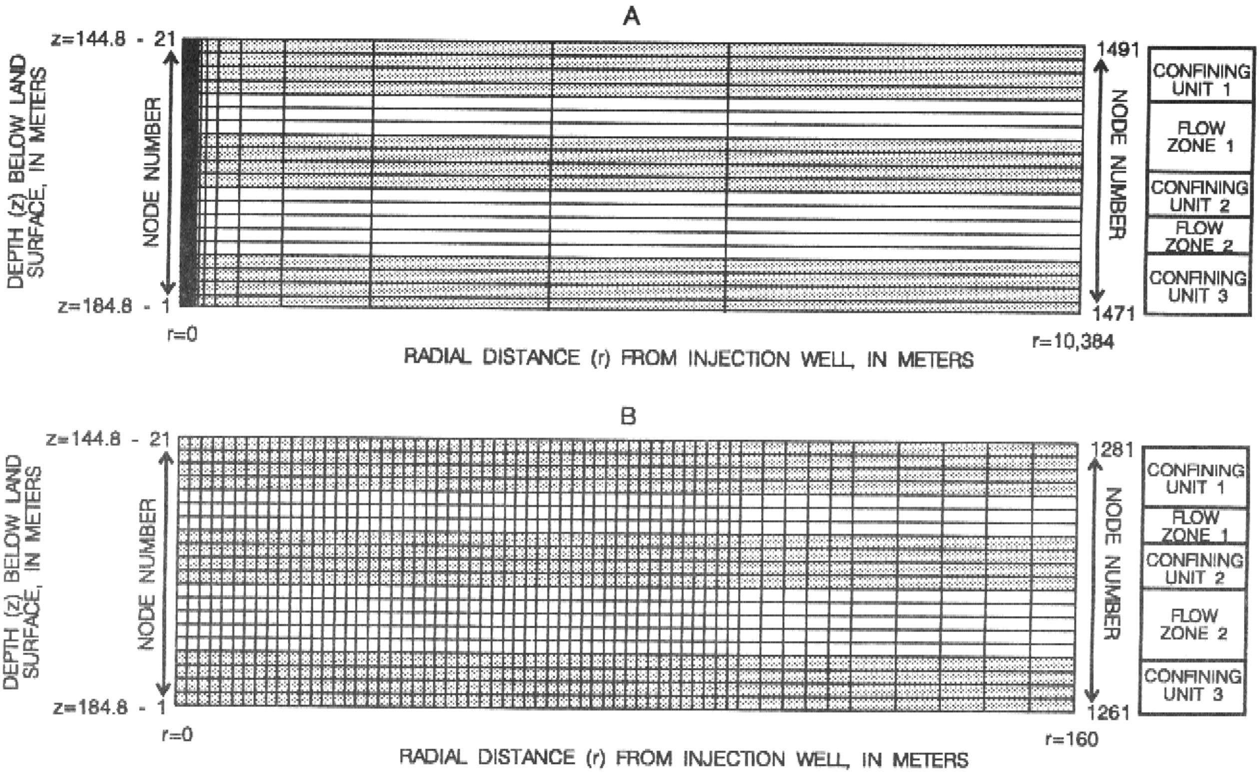


Figure 5. Sectional views of the cylindrical coordinate finite-element grid used to study previous subsurface injection, storage, and recovery of freshwater in the lower Hawthorn aquifer at the Lee County Water Treatment Plant.

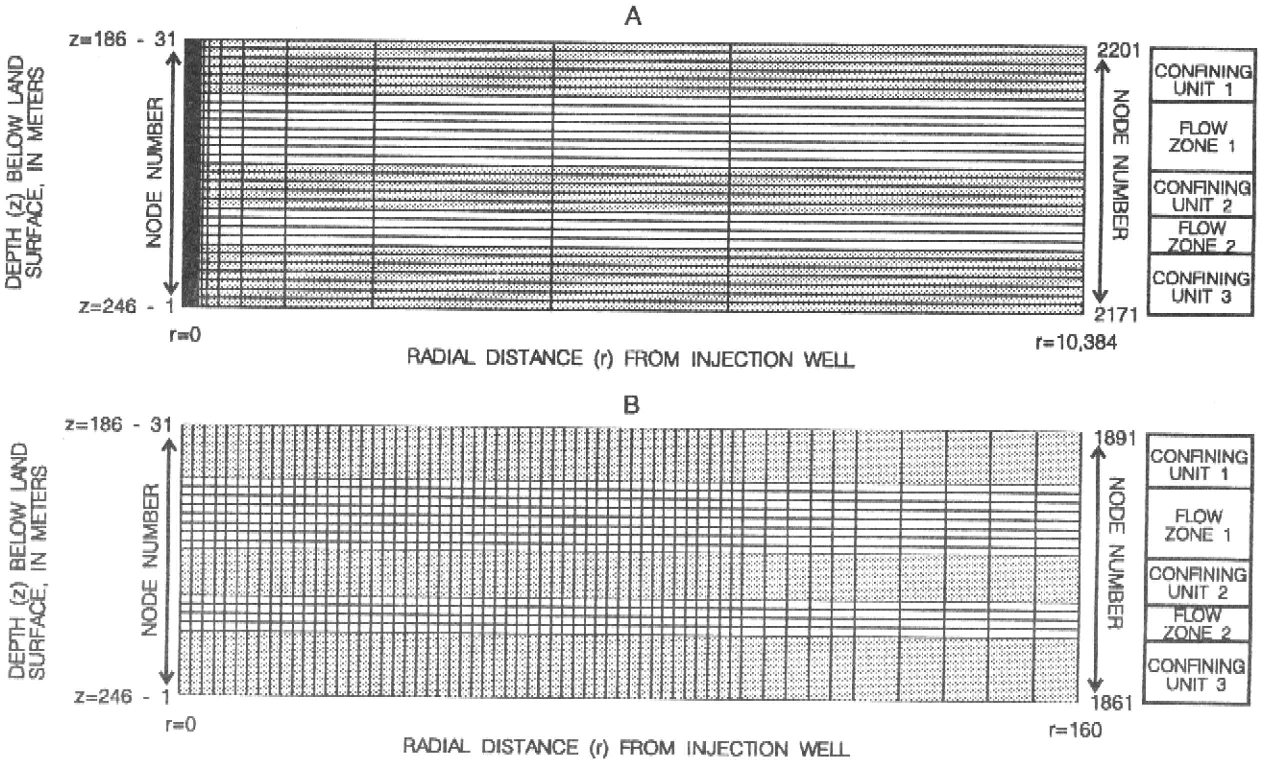


Figure 6. Sectional views of the cylindrical coordinate finite-element grid used to study hypothetical subsurface injection, storage, and recovery of freshwater in the lower Hawthorn aquifer at Cape Coral.

One limitation setting of these types of boundary conditions (constant pressure and concentration on top and bottom) is that if injected or mixed water passes across these boundaries, the model would be unable to consider it during the recovery pumping because the concentrations along these boundaries are assumed to represent a constant value. However, for the present study, these boundary conditions yielded the best representation of the actual aquifer in terms of approximating measured pressure and concentration changes in observation wells and in the injection well during recovery. Also, these boundary conditions would yield more conservative estimates of recovery efficiency. The lack of detailed hydrogeologic information beyond these boundaries precluded the location of the boundaries farther from flow zones receiving the injection water. An attempt was made to locate the boundaries farther from the injection source by extrapolating the hydrogeologic information, but the results were discouraging in terms of matching field measured pressure and concentration changes.

At r=10,384 m, no-flow/no-transport boundary conditions were specified. This boundary was intentionally located far from the injection source to prevent any effect that it might have on the determination of pressures and concentrations in the aquifer segment affected by the injection. Boundary conditions at the well (r=0) were set by specifying a mass flux equal to the injection rate. The flux was proportionally distributed among the boundary nodes along the length of the injection zone using the aquifer hydraulic characteristics (K) as a weighting factor. The solute concentration in the injected water during injection was specified at the well boundary (r=0). A flux average concentration for water withdrawn during recovery was calculated from concentration values at boundary nodes representing the well.

The hydraulic conductivity value of the upper and lower confining zones was modified using the model to replicate the effect of these leaky units on pressure and concentration changes in the main flow zones (discussed later). Although more sophisticated boundary types are currently available, they are not available in QSUTRA, and this study lacks the field data to justify their application. For large volumes of water injected (larger than those used in this study), the vertical and horizontal boundaries can become invalid yielding unrealistic model results.

Initial pressures were assumed to be hydrostatic and set equal to an equivalent freshwater head of 10.49 m above sea level for the Lee County Water Treatment Plant model and 7.62 m for the Cape Coral area model. Initial solute concentration was set equal to solute concentration in the native water. For this study, fluid density was assumed to depend only on solute concentration. Fluid density was calculated by the model based on initial solute concentrations and the following functional relation between density and solute concentration:

$$\rho = \rho_i + (\rho_n - \rho_i)\,[(C-C_i)/(C_n-C_i)] \qquad (15)$$

where,

ρ_i is density of injected water [M/L^3];

ρ_n is density of native water [M/L^3]l;

C is solute concentration in the mixed water [M/L^3];

C_i is solute concentration in the injected water [M/L^3]; and

C_n is solute concentration in the native water [M/L^3].

Solute Source

Chloride ion, the dominant conservative anion in the native aquifer water and the injected surface water, was selected as the solute to be modeled. Chloride concentrations in water samples from the lower Hawthorn aquifer ranged from 500 to 550 mg/L at the Lee County Water Treatment Plant and from 350 to 750 mg/L in Cape Coral (Missimer and Associates, Inc., 1985). The model computes relative or normalized concentrations that range from 0.1 to 1, where 0.1 represents concentration in the injected water and 1 represents concentration in the native water.

Time Steps

Initial time-step sizes were kept equal or smaller than 400 seconds to avoid numerical dispersion associated with a large time-step size. The time-step size was increased during the injection phase in such a way that the injected water front (neglecting dispersion) moved a constant distance during each successive time step. The final time-step size from the injection phase was used and kept constant for the entire simulation of the storage period. During the recovery phase, the time-step size was gradually reduced from its maximum value as the injected water front moved closer to the well. Generally, except for the first time step in each run, only two iterations per time step were needed to resolve the nonlinearities of the density-dependent flow equation (eq. 7).

Model Simulation Results for the Lee County Water Treatment Plant—Calibration and Testing

Data from a study by Fitzpatrick (1986a) were used in this study to define the hydrogeologic system and to provide a basis for estimating the hydraulic and transport characteristics for the lower Hawthorn aquifer in Cape Coral. The conceptual model for the Lee County Water Treatment Plant site was developed on the basis of interpretation of velocity, caliper, fluid resistivity, and fluid temperature borehole logs and interpretation of aquifer-test data (Fitzpatrick, 1986a). The conceptual model consists of two main flow zones and three leaky confining units (fig. 5). Aquifer hydraulic characteristics, boundary conditions, and nodes subject to them were previously described.

Two injection, storage, and recovery tests and results (table 3) from the study by Fitzpatrick (1986a) were useful in calibrating the model (tests 2 and 3). Test 3 was used for model calibration and test 2 for model testing. Data for test 3 were obtained for the injection well (L-3225) and two observation wells (L-2530 and L-3224), about 43 and 102 m, respectively, from the injection well. The calibration of the model was performed using the classical interactive process in which the model variables were changed within realistic limits, until a satisfactory match to the measured data was obtained. Initial model variables were set according to data presented in table 1 and information previously described in this report.

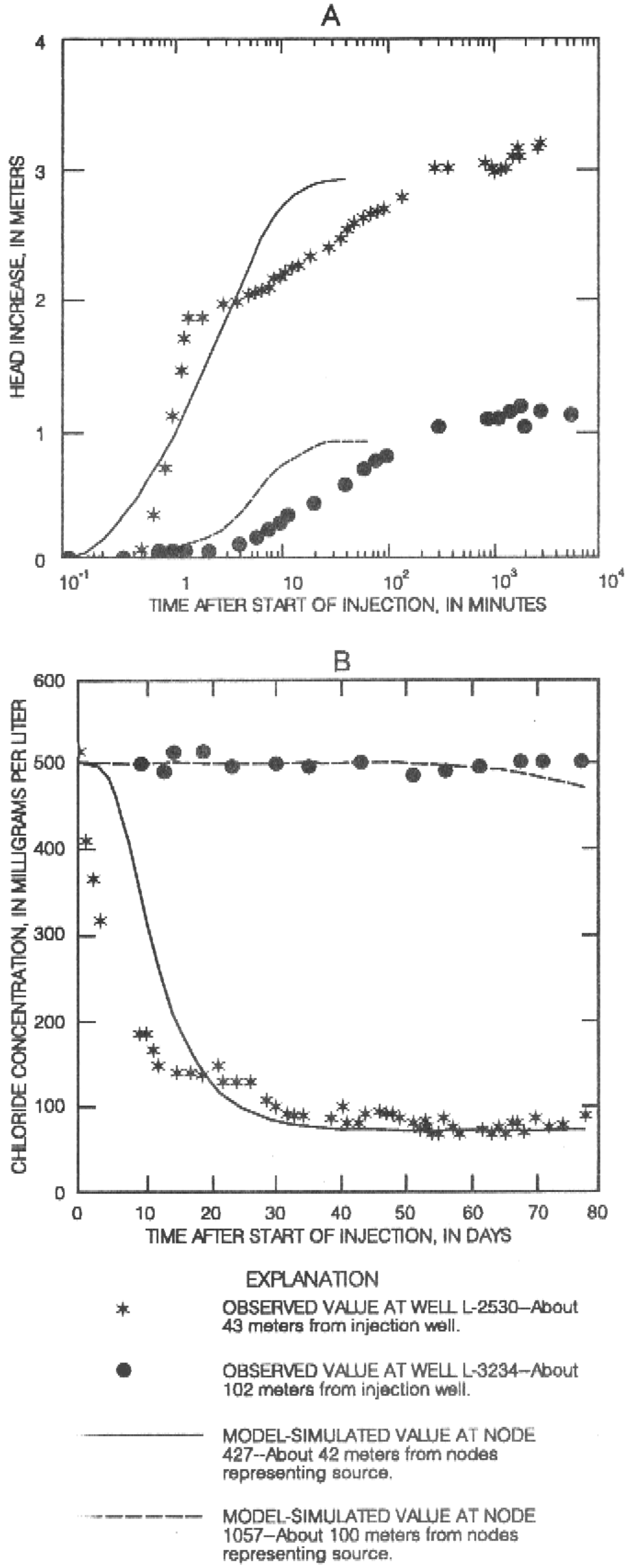
Increases in hydraulic head at observation wells L-2530 and L-3224 were used to calibrate the hydraulic variables. The permeability of the flow zones is assumed to be isotropic, and no attempt was made to change it. However, the permeability of the leaky confining units was decreased from an estimated value of 1.89×10^{-13} to 1.67×10^{-13} m^2 to obtain a satisfactory match between observed and modeled head change data (fig. 7A). The permeability is assumed to be isotropic in the vertical and horizontal directions. Following the hydraulic calibration, data on chloride concentration changes in the two observation wells (L-2530 and L-3224) were used to calibrate the transport model for effective porosity and longitudinal and transverse dispersivities. The model yielded better results when using an effective porosity of 0.12, a longitudinal dispersivity (α_T) of 3.0 m, and a transverse dispersivity (α_T) of 0.3 m for a ratio of $\alpha_T/\alpha_L = 0.1$ (fig. 7B). However, the model did not fit the field test data for the early arrival times of the injected water front at well L-2530 (fig. 7B). Several simulations were made varying the effective porosity, dispersivity values (α_T and α_T), and the aquifer permeability without obtaining a good match to the field data from well L-2530, while simultaneously matching the field data from well L-3224. This is probably because of the nature of flow in a part of the aquifer, which according to the borehole velocity logs (fig. 3), seems to have cavernous porosity, whereas the model is based on equations that are developed for a porous media system.

Model results for test 3 were compared with field data at the injection well (L-3225) for the recovery phase. Although a satisfactory match was obtained for breakthrough at observation wells L-2530 and L-3224, model predicted values for recovery chloride concentrations at the injection well (L-3225) were low compared to field measured values. Different porosity values were assigned to the main flow zones and the leaky confining units in an attempt to improve the model predictions at the injection well while keeping a good match at the two observation wells. A combination of porosity values of 0.15 for the main flow zones and 0.05 for the leaky confining units yielded satisfactory results (fig. 8). The characteristics used in the calibrated model and the fluid, solute, and rock matrix properties used in the simulations are listed in tables 4 and 5, respectively.

Table 3. Results of two injection, storage, and recovery of freshwater tests conducted from a previous study in the lower Hawthorn aquifer at the Lee County Water Treatment Plant

[Tests conducted by Fitzpatrick (1986a). Recovery time indicates time since the beginning of recovery when chloride concentration of recovered water approached background concentration of native aquifer water]

Test number	Average injection rate (cubic meters per day)	Average recovery rate (cubic meters per day)	Total volume injected (cubic meters)	Injection time (days)	Storage time (days)	Recovery time (days)	Average chloride concentration of injected water (milligrams per liter)
2	1,635	899	26,160	16	47	50	171
3	1,423	818	109,571	77	98	150	69



e 7. Observed and model simulated head increase g the first 7 days of injection and chloride concentration breakthrough curves at observation wells L-2530 and L-3224 during the injection phase of test 3 at the Lee County Water Treatment Plant.

The model was tested using chloride concentration data at the injection well (L-3225) during the recovery phase of test 2 (Fitzpatrick, 1986a). The test simulation was made using the same hydraulic and transport characteristics from the calibration run for test 3. The resulting chloride concentration breakthrough curve produced by the model was low compared to the field data (fig. 9). In an attempt to provide a closer match of the field data, the longitudinal and transverse dispersivity values were increased from 3.0 and 0.3 m to 5.0 and 0.5 m, respectively. This change resulted in a good match of the field measured data by the model-generated data (fig. 9). According to the present knowledge on the scale dependency of the dispersion coefficient (Gelhar and others, 1979; Gelhar and Axness, 1983; and Mercado, 1984), the value used to effectively simulate test 2 was expected to be smaller than its counterpart for test 3. However, the dispersivity value from test 2 was larger than that from test 3, but the difference between the values was small ($\alpha_L = 3.0$ m and $\alpha_T = 0.3$ m for test 3; $\alpha_L = 5.0$ m and $\alpha_T = 0.5$ m for test 2). No further attempt was made in this study to explain the differences in the dispersivity values between the two tests because detailed field information was unavailable.

Model Simulation Results for Cape Coral—Effects of Operational Factors on Recovery Efficiency

A series of hypothetical SISRF tests were made for the lower Hawthorn aquifer in Cape Coral using the digital modeling technique. Estimates of the hydrologic and transport characteristics from the analysis of previous test data (Fitzpatrick, 1986a) were used in a baseline simulation with other factors represented by values from studies in similar geologic units. The baseline simulation was used as a reference to study the effects of changing a series of SISRF operational factors on the recovery efficiency. The hydrologic and transport characteristics used in the baseline simulation were selected as the best possible representation of the actual field values in Cape Coral. These characteristic values might not necessarily represent the entire spatial spectrum of possible values in the lower Hawthorn aquifer. Therefore, the characteristic values used in the simulations are subject to some uncertainty. The effects on the recovery efficiency of the rates of injection and recovery; volume of water injected; storage time; injection into selected flow zones; successive cycles of injection, storage, and recovery; and chloride concentrations of injected and native waters were also studied using the digital model.

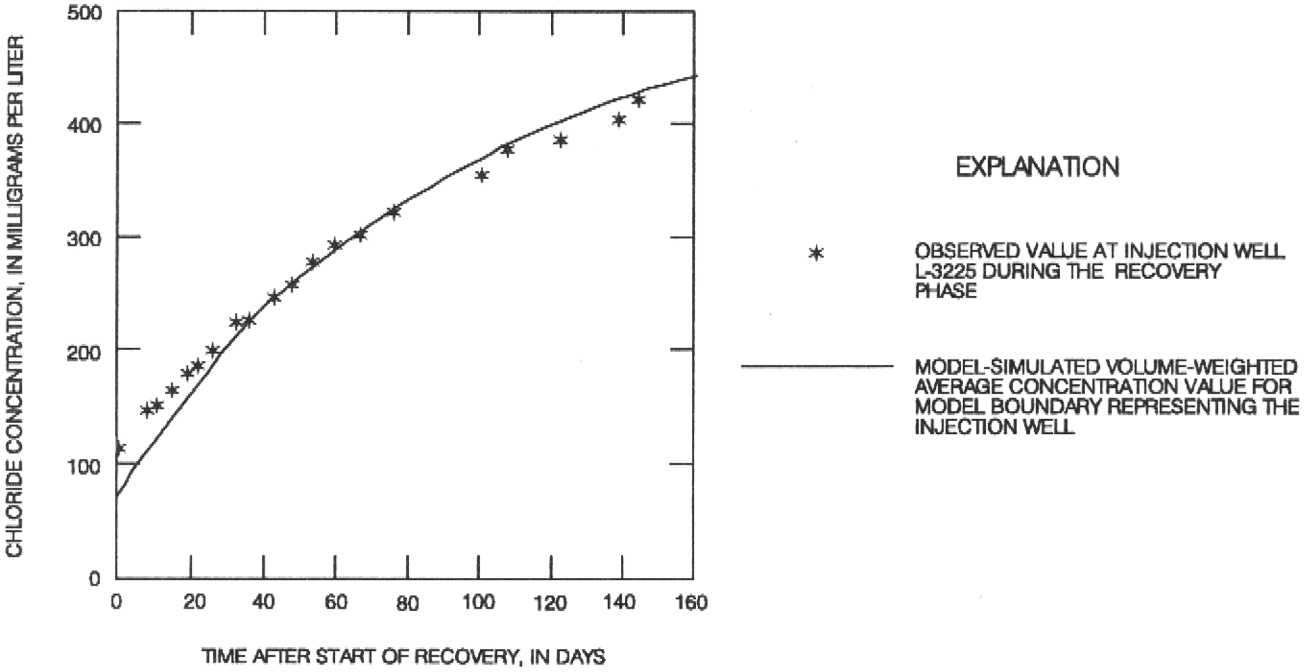


Figure 8. Observed and model-simulated chloride concentration in water recovered from injection well L-3225 during the recovery phase of test 3 at the Lee County Water Treatment Plant (data from Fitzpatrick, 1986a).

Table 4. Characteristics of flow zones and confining units used to model the lower Hawthorn aquifer at the Lee County Water Treatment Plant

Flow zones and leaky confining units (meters below land surface)	Permeability (square meters)	Effective porosity (percent)	Specific pressure storativity (kilograms per meter per second squared)$^{-1}$	Longitudinal dispersivity (meters)	Transverse dispersivity (meters)
144.8–152.8	1.670×10^{-13}	5	1.36×10^{-10}	3.0	0.3
152.8–158.8	3.846×10^{-12}	15	1.68×10^{-10}	3.0	.3
158.8–166.8	5.140×10^{-13}	5	1.36×10^{-10}	3.0	.3
166.8–176.8	5.572×10^{-12}	15	1.68×10^{-10}	3.0	.3
176.8–184.8	1.670×10^{-13}	5	1.36×10^{-10}	3.0	.3

Table 5. Fluid, solute, and rock matrix properties used in the simulations

Property	Value
Dynamic viscosity of native water, in kilograms per meter per second	0.001
Dynamic viscosity of injected water, in kilograms per meter per second	0.001
Density of native water, in kilograms per cubic meter	1,001.0
Density of injected water, in kilograms per cubic meter	1,000.1
Coefficient of molecular diffusion, in meters squared per second	5.0×10^{-10}
Fluid compressibility, in (kilograms per meter per second squared)$^{-1}$	4.4×10^{-10}
Rock matrix compressibility, in (kilograms per meter per second squared)$^{-1}$	1.2×10^{-10}

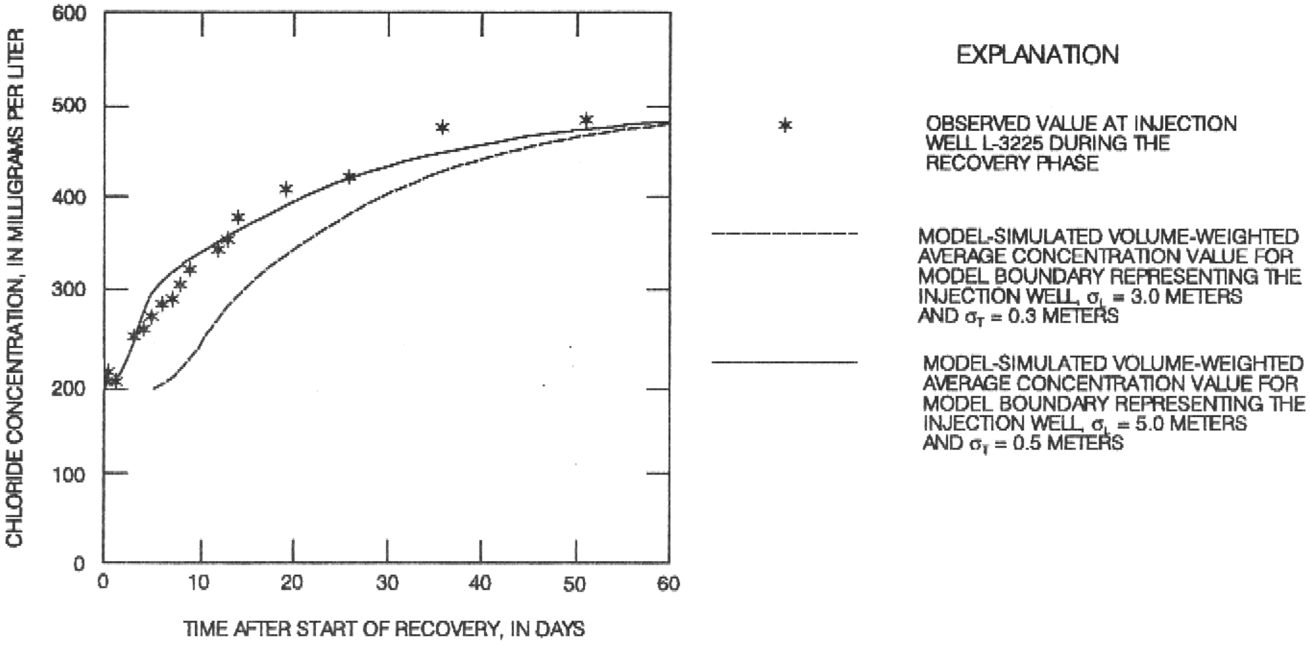


Figure 9. Observed and model-simulated chloride concentration in water recovered from injection well L-3225 during the recovery phase of test 2 at the Lee County Water Treatment Plant (observed data from Fitzpatrick, 1986a).

Baseline Simulation

A baseline simulation was made using the previously described model grid (fig. 6), estimated hydraulic and transport characteristics (tables 1, 5, and 6), and the conditions presented in table 7. The growth of the injected water body and the chloride distribution profiles (mixing zone) during the injection phase of the baseline simulation are depicted in figure 10. Although the injected water body in the lower main flow zone has twice the radial extent of its counterpart in the upper main flow zone, the difference between the chloride distribution profiles of the two flow zones was not significant (fig. 10). The injected water front was about 50 m from the injection well in the lower main flow zone and 25 m from the injection well in the upper main flow zone at the end of the injection phase (fig. 10D). A vector representation of the pore-water velocity field was generated by the model (fig. 11). This velocity vector shows that the injected water, in general, is moving: (1) horizontally outward along the two main flow zones, (2) vertically upward from the upper main flow zone into the upper confining zone, (3) vertically downward from the lower main flow zone into the lower confining zone, and (4) vertically upward from the lower main flow zone through the middle confining zone into the upper main flow zone (fig. 11). A similar vector representation was generated by the model during the recovery phase, but the vectors point in the opposite direction (fig. 12).

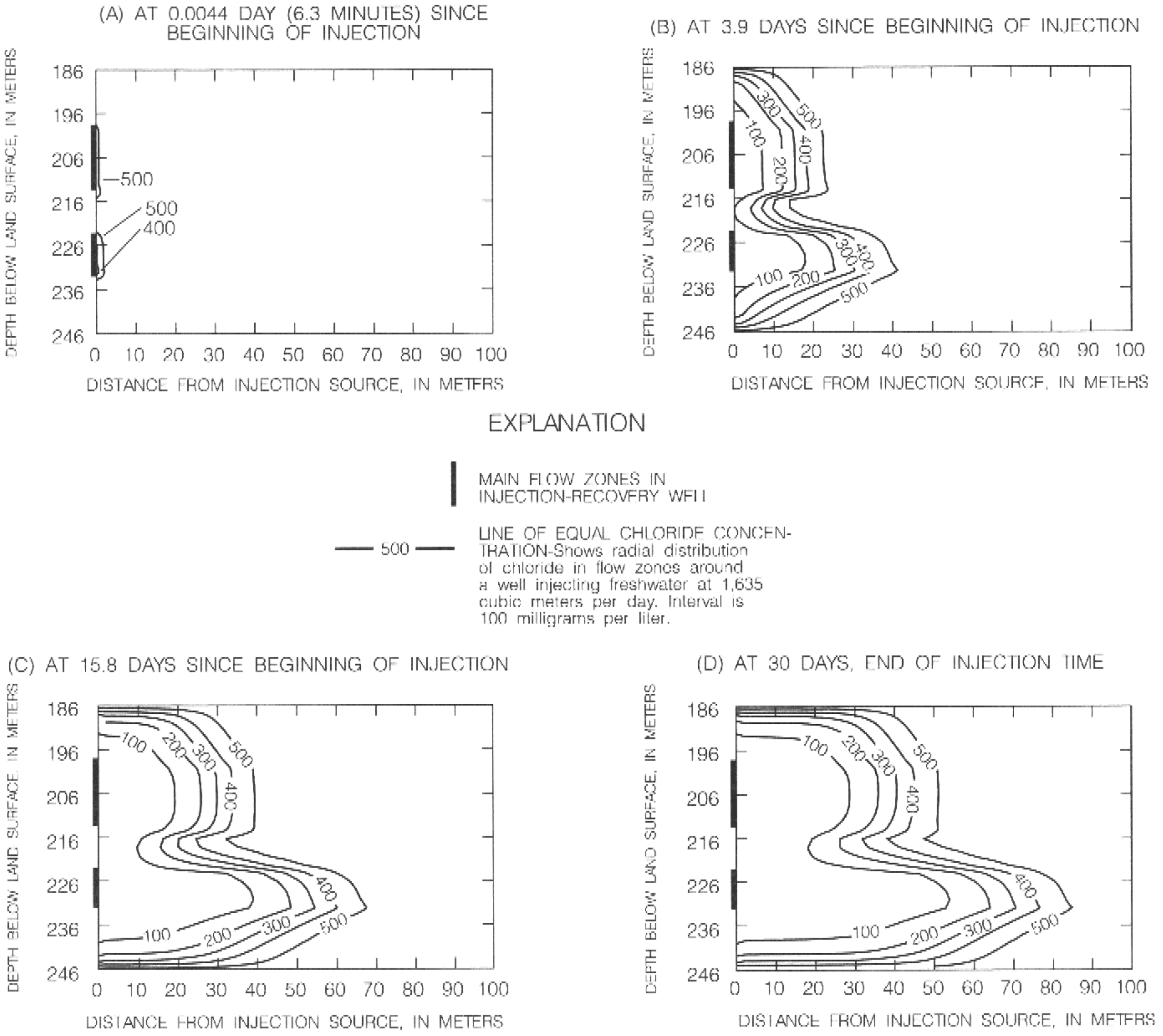
Table 6. Characteristics of flow zones and confining units used to model the lower Hawthorn aquifer at Cape Coral

Flow zones and leaky confining units (meters below land surface)	Permeability (square meters)	Effective porosity (percent)	Specific pressure storativity (kilograms per meter per second squared)$^{-1}$	Longitudinal dispersivity (meters)	Transverse dispersivity (meters)
186.0–198.0	1.061×10^{-12}	5	1.36×10^{-10}	3.0	0.3
198.0–214.0	1.085×10^{-11}	15	1.68×10^{-10}	3.0	.3
214.0–224.0	1.061×10^{-12}	5	1.36×10^{-10}	3.0	.3
224.0–232.0	3.435×10^{-11}	15	1.68×10^{-10}	3.0	.3
232.0–246.0	1.061×10^{-12}	5	1.36×10^{-10}	3.0	.3

Table 7. Conditions and results for recovery times and efficiencies for the baseline simulation and other simulations of subsurface freshwater injection, storage, and recovery for the lower Hawthorn aquifer at Cape Coral

[Recovery time is when the preestablished chloride concentration limit of 250 milligrams per liter is reached]

Simulation number	Injection rate (cubic meters per day)	Recovery rate (cubic meters per day)	Recovery rate/ injection rate (dimensionless)	Volume of injected water (cubic meters)	Injection time (days)	Storage time (days)	Chloride concentrations, milligrams per liter		Recovery time (days)	Recovery efficiency (percent)
							Injection water	Native aquifer water		
Baseline Simulation										
1	1,635.2	1,635.2	1.00	49,055	30	0	50	500	19.2	64
Changes in Rates of Injection and Recovery										
2	408.8	408.8	1.00	49,055	120.0	0	50	500	76.1	63
3	817.6	817.6	1.00	49,055	60.0	0	50	500	37.5	63
4	3,270.3	3,270.3	1.00	49,055	15.0	0	50	500	10.2	68
5	6,540.7	6,540.7	1.00	49,055	7.5	0	50	500	6.9	92
Changes in Recovery Rate/Injection Rate Ratio										
6	1,635.2	408.8	.25	49,055	120.0	0	50	500	74.3	62
7	1,635.2	817.6	.50	49,055	60.0	0	50	500	37.5	63
8	1,635.2	3,270.3	2.00	49,055	15.0	0	50	500	10.2	68
9	1,635.2	6,540.7	4.00	49,055	7.5	0	50	500	7.0	93
Changes in Volume of Water Injected										
10	1,635.2	1,635.2	1.00	12,264	7.5	0	50	500	7.6	100
11	1,635.2	1,635.2	1.00	24,528	15.0	0	50	500	12.1	81
12	1,635.2	1,635.2	1.00	98,110	60.0	0	50	500	33.5	56
13	1,635.2	1,635.2	1.00	196,221	120.0	0	50	500	51.8	43
Changes in Storage Time										
14	1,635.2	1,635.2	1.00	49,055	30	5	50	500	19.2	64
15	1,635.2	1,635.2	1.00	49,055	30	30	50	500	19.0	63
16	1,635.2	1,635.2	1.00	49,055	30	90	50	500	18.6	62
17	1,635.2	1,635.2	1.00	49,055	30	180	50	500	18.1	60
Injection into Upper Flow Zone (198–214 meters)										
18	1,635.2	1,635.2	1.00	49,055	30	0	50	500	18.3	61
Injection into Lower Flow Zone (224–232 meters)										
19	1,635.2	1,635.2	1.00	49,055	30	0	50	500	18.6	62
Five Successive Cycles										
20	1,635.2	1,635.2	1.00	49,055	30	180	50	500	18.3	61
21	1,635.2	1,635.2	1.00	49,055	30	180	50	500	23.4	78
22	1,635.2	1,635.2	1.00	49,055	30	180	50	500	25.0	83
23	1,635.2	1,635.2	1.00	49,055	30	180	50	500	25.8	86
24	1,635.2	1,635.2	1.00	49,055	30	180	50	500	26.6	89
Different Injected and Native Water Chloride Concentrations										
25	1,635.2	1,635.2	1.00	49,055	30	0	100	500	16.6	55
26	1,635.2	1,635.2	1.00	49,055	30	0	200	500	9.0	30
27	1,635.2	1,635.2	1.00	49,055	30	0	50	1,000	10.7	36
28	1,635.2	1,635.2	1.00	49,055	30	0	50	2,000	6.5	22



(A) AT 0.0044 DAY (6.3 MINUTES) SINCE BEGINNING OF INJECTION

(B) AT 3.9 DAYS SINCE BEGINNING OF INJECTION

(C) AT 15.8 DAYS SINCE BEGINNING OF INJECTION

(D) AT 30 DAYS, END OF INJECTION TIME

Figure 10. Chloride distribution profiles at different times during the injection phase of the baseline simulation.

A 64 percent recovery efficiency value was obtained for the baseline simulation for the preselected 250-mg/L chloride concentration limit. The thickness of the mixing zone at the end of the recovery phase grew from 1.5 to 2 times compared to its thickness at the end of the injection phase (figs. 13 and 10D). Some residual injected water was still inside the injection zones at the end of recovery (fig. 13). A subsequent injection phase would result in a wider mixing zone and a higher recovery efficiency.

A simulation was made with the same parameters that were used in the baseline simulation but using no-flow/no-transport boundaries at the top and bottom limits of the model. This simulation was conducted to test the effect on the recovery efficiency of using a constant pressure/constant concentration boundary condition to represent interlayer solute mass movement across these boundaries. The simulation yielded a recovery efficiency of 83 percentage points, which is 19 percentage points higher than the value estimated from the baseline simulation (64 percentage points). This indicates that the constant pressure/constant concentration boundaries are important in the determination of the recovery efficiency and that this type of boundary would yield more conservative estimates of the recovery efficiency.

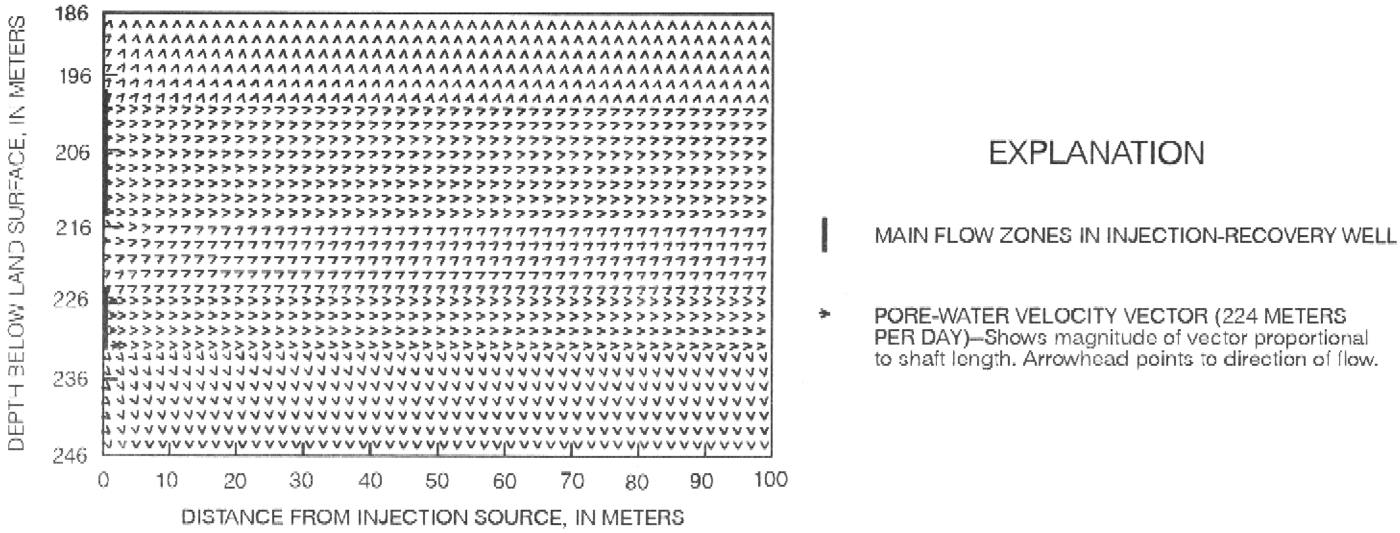


Figure 11. Vector field representing pore-water velocities in a radial section of the flow zones at the end of the injection phase of the baseline simulation.

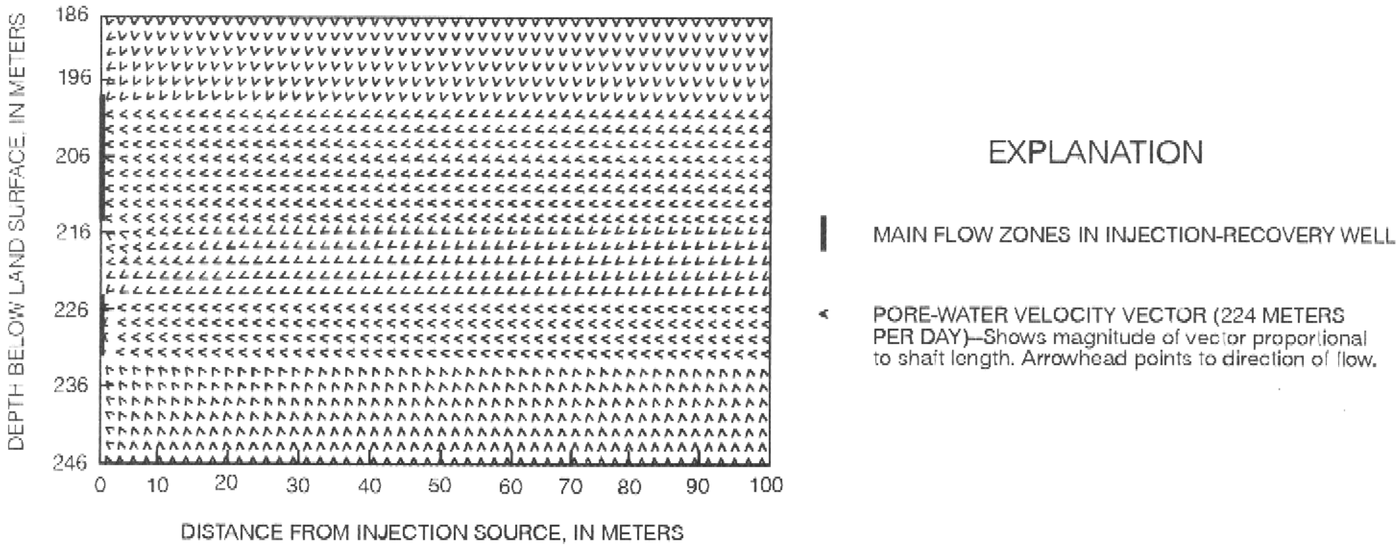


Figure 12. Vector field representing pore-water velocities in a radial section of the flow zones at the end of the recovery phase of the baseline simulation.

Rates of Injection and Recovery

The effect of the rates of injection and recovery on the recovery efficiency was studied with eight simulations using different injection and recovery rates and injection rate/recovery rate ratios (simulations 2-9 in table 7). In simulations 2 to 5, the injection rate (Q_I) and the recovery rate (Q_R) were each changed by 25, 50, 200, and 400 percent from the value used in the baseline simulation. In simulations 6 to 9, the ratio of Q_R/Q_I was changed by 25, 50, 200, and 400 percent from the baseline simulation ratio ($Q_R/Q_I = 1$).

The results of the simulations indicated that when the injection rate was decreased by 25 and 50 percent while keeping Q_R/Q_I equal to 1, an insignificant decrease in the recovery efficiency occurred (fig. 14 and table 7). However, when the injection and recovery rates were increased by 200 and 400 percent, the recovery efficiency increased from 64 percent (for the baseline simulation) to 68 and 92 percent, respectively (fig. 14 and table 7). Although in a previous hypothetical study (Merritt, 1985) no relation was reported between the rates of injection and recovery and the

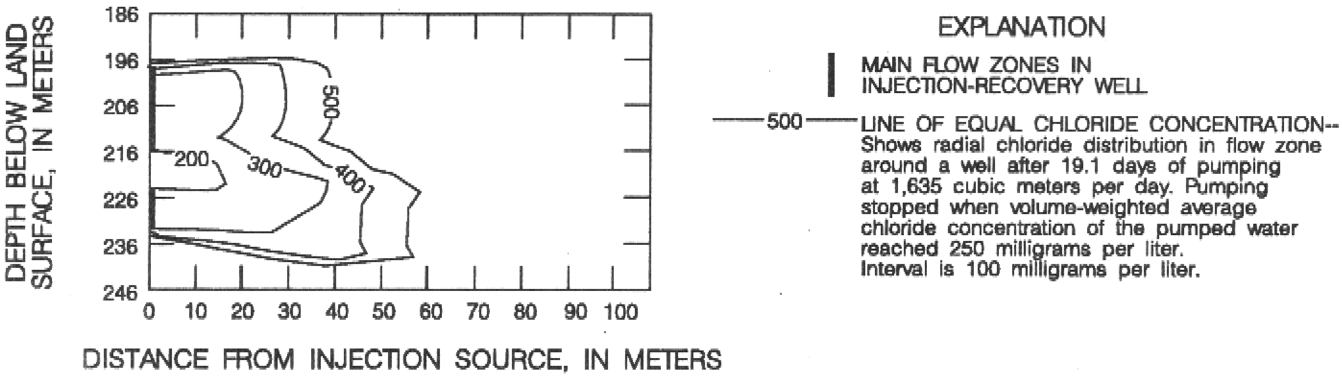


Figure 13. Chloride distribution profile at the end of the recovery phase of the baseline simulation.

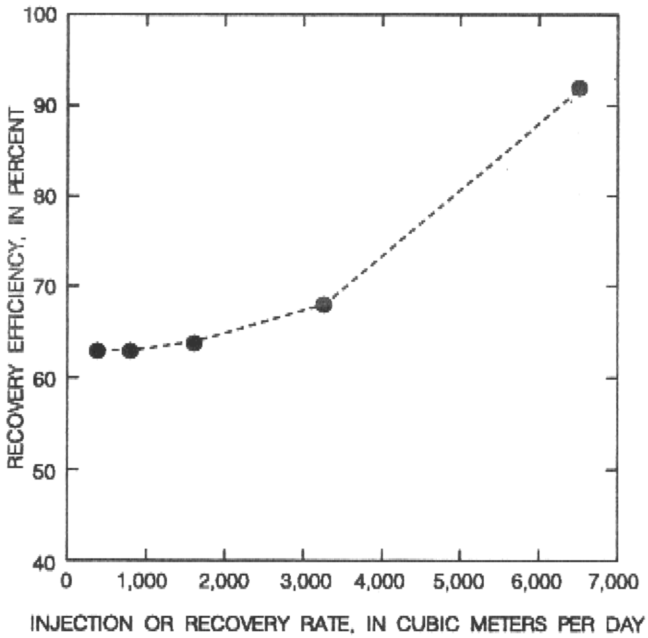


Figure 14. Relation between recovery efficiency and injection or recovery rate in the lower Hawthorn aquifer for $Q_R/Q_I = 1$.

recovery efficiency, the injected solute mass was confined by the upper and lower boundaries, keeping the injected mass near the well region and precluding mass migration across the upper and lower model boundaries. Because the mass of injected water was confined, no vertical movement occurred, and therefore, the duration and rate of injection and recovery were not important. Leakance occurs in most confined aquifers, and interlayer solute mass movement provides mechanics for mass migration, thereby affecting the recovery efficiency.

For the recovery rate/injection rate (Q_R/Q_I) ratios of 25 and 50 percent, the recovery efficiency decreased slightly (fig. 15 and table 7). For Q_R/Q_I ratios of 200 and 400 percent, the recovery efficiency increased from 64 percent (for the baseline simulation value) to 68 and 93 percent, respectively (fig. 15 and table 7). This relation can be explained by the fact that vertical mass transfer in leaky aquifers can be significant. For fast recovery rates, the vertical migration of mass would be smaller, providing for higher recoverability.

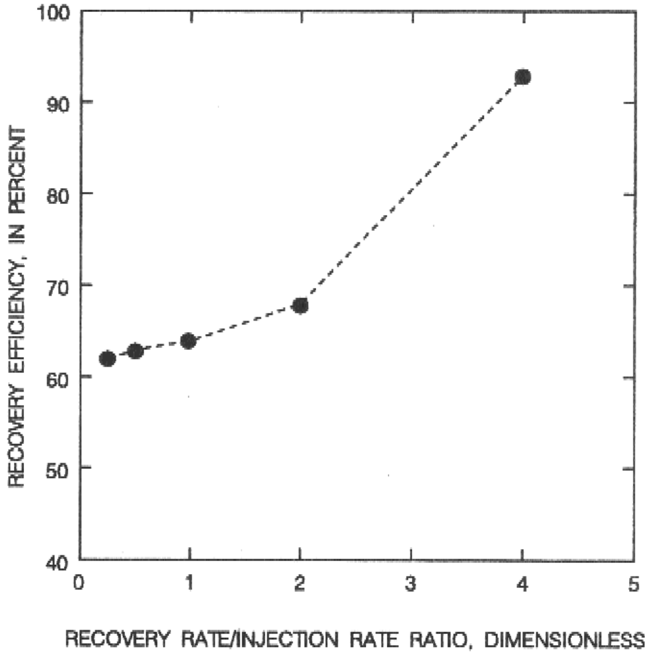


Figure 15. Relation between recovery efficiency and recovery rate/injection rate ratio in the lower Hawthorn aquifer.

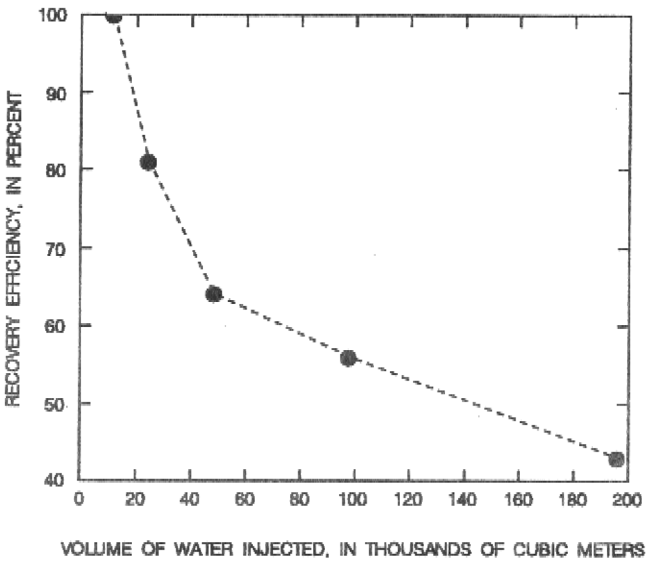


Figure 16. Relation between recovery efficiency and volume of water injected in the lower Hawthorn aquifer.

Volume of Water Injected

The effect of injecting different size volumes was studied using four simulations in which the injected volume was changed by 25, 50, 200, and 400 percent from the baseline simulation value (simulations 10-13 in table 7). This was accomplished by decreasing or increasing the injection time, while keeping the same injection rate used in the baseline simulation. Results from these simulations show that for the range of injected volumes tested in this study the recovery efficiency decreases as the volume of water injected increases (fig. 16 and table 7). Initially, the recovery efficiency decreases at a great rate as the volume of water injected is increased, but an asymptote is approached at a recovery efficiency value of about 40 percent (fig. 16); however, this result cannot be generalized. Some investigators (Merritt, 1985; Quiñones-Aponte and others, 1989) reported that the relation between the volume of water injected and the recovery efficiency can change direction for different ranges of volumes of water injected. For instance, the recovery efficiency for a range of small volumes of water injected can increase as the volume of water injected increases, and the recovery efficiency for a range of large volumes of water injected can decrease as the volume of injected water increases. The type of aquifer (confined or leaky) and boundary conditions can also affect the relation between volume of water injected and recovery efficiency. The leaky nature of the aquifer represented in this study model provides for transfer of injected water into low-permeability units. For longer injection times, larger volumes of water would migrate into and across the low-permeability units, thus reducing the potential for freshwater recovery.

Storage Time

The effect of storage time duration was assessed by increasing the duration of the storage time from the baseline simulation value of 0 days. Four simulations were made using storage times of 5, 30, 90, and 180 days (simulations 14-17 in table 7). Results from the simulations indicated that the storage time did not greatly affect the recovery efficiency, showing only a 4 percentage point decrease in recovery efficiency when the storage time was increased from 0 to 180 days (fig. 17 and table 7). However, the present model does not consider the regional background flow, which, combined with the storage time, could significantly reduce the recovery efficiency. Quiñones-Aponte and others (1989) interpreted actual SISRF tests and suggested that the recovery efficiency generally decreases as the storage time increases, but the rate of decrease in recovery efficiency would also depend on the volume of water injected. When small volumes of water are injected, the storage time has a stronger effect on reducing the recovery efficiency than when large volumes are injected (Quiñones-Aponte and others, 1989). The effect of storage time on the recovery efficiency would become overshadowed by the effect of the volume of water injected when the volume injected is large.

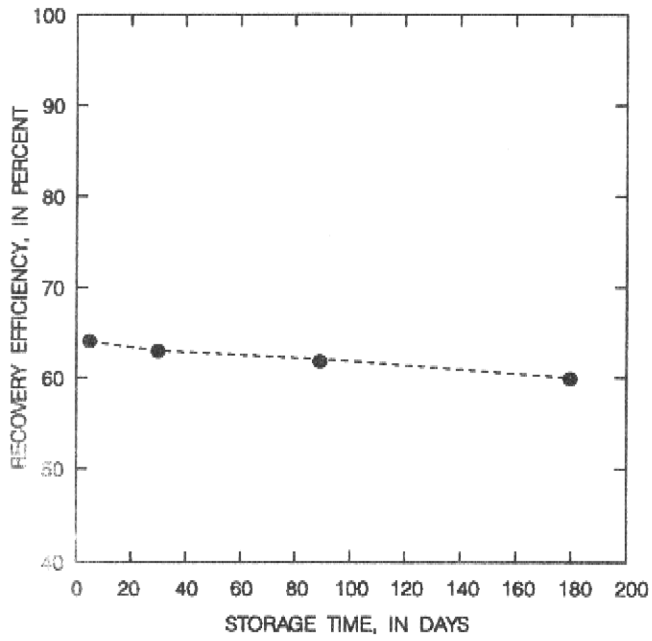


Figure 17. Relation between recovery efficiency and storage time in the lower Hawthorn aquifer.

Water Injected into Selected Flow Zones

The effect of injection into selected flow zones on recovery efficiency was studied by individually injecting the same volume of water at the same rate (volume and rate used for the baseline simulation) to each of the two more permeable flow zones. Two simulations were made—injection into the upper flow zone (198-213.3 m) and injection into the lower flow zone (222.5-231.6 m). The recovery efficiency did not change significantly; however, the configuration of the lines of equal chloride concentration at the end of the injection phase for both simulation cases revealed a sharp contrast between cases (fig. 18) and compared to their counterpart for the baseline simulation (figs. 18 and 10D). Model results indicated that the recovery efficiency in both cases decreased by a very small amount (simulations 18 and 19 in table 7) compared with the baseline simulation, which is not significant if the errors associated with the numerical method are taken into consideration. Merritt (1985) reported similar results; however, to generalize these results, a more-detailed study focusing on this aspect (injection into different flow zones) is needed.

When the injection well is open to all of the flow zones, a potential problem is the occurrence of interflow from higher to lower permeability zones through the wellbore during storage time. Water from flow zones under higher hydraulic pressure flows through the wellbore into flow zones under lower hydraulic pressure. This potential problem was not assessed by the model presented in this report; however, it should be considered for the design of actual injection wells.

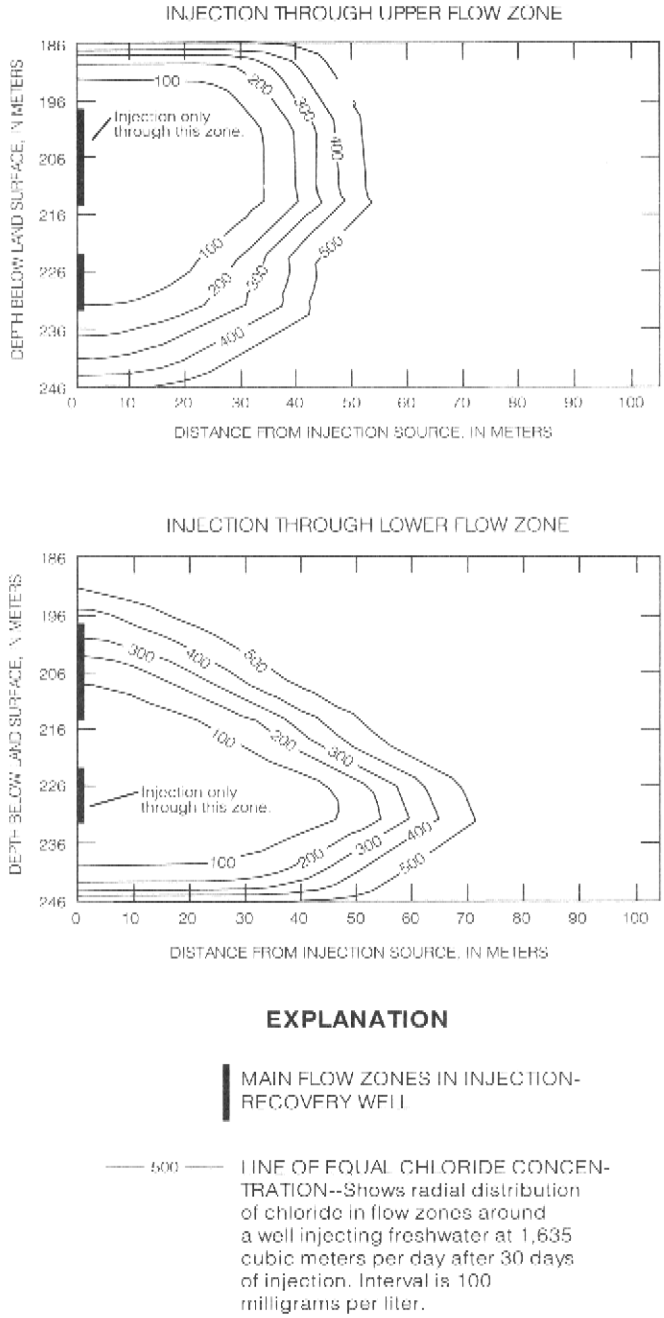


Figure 18. Chloride distribution profiles at the end of a 30-day injection period for the cases of injection into the upper and lower flow zones.

Successive Cycles of Injection, Storage, and Recovery

Five consecutive simulations were made to study the effect of successive cycles of injection, storage, and recovery on the recovery efficiency. The different factors were not changed from the baseline simulation values; however, a storage time of 180 days was used for each cycle (simulations 20-24 in table 7). Results from the preceding cycle were used as initial values for simulating the following cycle. Model results were similar to those reported by Merritt (1985). The rate of improvement on recovery efficiency with successive SISRF cycles was higher during the early cycles, increasing from about 60 to 84 percent during the first three cycles (fig. 19). Recovery efficiency increased from about 84 to 88 percent for cycles 3, 4, and 5 (fig. 19). It can be inferred from Merritt (1985, fig. 12) that the relation between recovery efficiency and the number of SISRF cycles approaches an asymptote after a certain number of cycles, where for practical purposes, no improvement of recovery efficiency occurs. The asymptote is reached at earlier cycle numbers for aquifers having small longitudinal dispersivity values (Merritt, 1985).

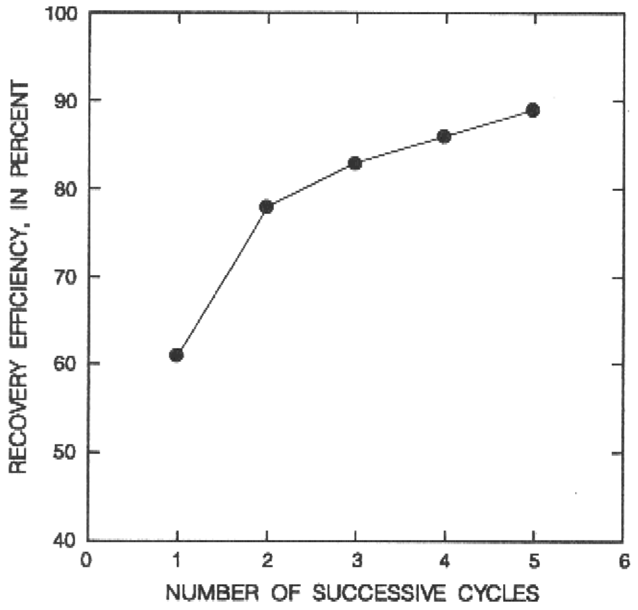


Figure 19. Relation between recovery efficiency and successive subsurface injection, storage, and recovery of freshwater cycles in the lower Hawthorn aquifer.

Chloride Concentrations in Injected and Native Waters

Four simulations were made to study the effects of different chloride concentrations in the injected and native waters. The chloride concentration in the injected water was changed in two simulations by increasing the value used in the baseline simulation by 200 and 400 percent. The chloride concentration in the native water in the remaining two simulations was changed in the same manner. The recovery efficiency in all of the simulations indicated reductions ranging from 9 percentage points (from the baseline simulation value) for 100 mg/L of chloride concentration in injected water to 42 percentage points (from the baseline simulation value) for 2,000 mg/L of chloride concentration in native water (simulations 25-28 in table 7). The analysis indicates: (1) the changes in the quality of injected water could result in reduction of the recovery efficiency; and (2) increases in the chloride concentration in native water because of saltwater intrusion, upconing, or other factors can decrease the recovery efficiency (table 7).

Sensitivity Analysis

Simulations were made to determine the sensitivity of the model-predicted recovery efficiency to variation in modeled aquifer characteristics, including permeability, ratios of anisotropy, longitudinal and transverse dispersivities, molecular diffusion, and effective porosity. The sensitivity analysis was conducted to assess the uncertainty of estimating the aquifer hydraulic and transport properties. A sensitivity analysis provides the means to identify the most important aquifer characteristics.

The relative sensitivity approach developed by Simon (1988) was applied in this sensitivity study. In the relative sensitivity approach, modeled aquifer characteristics are varied from an optimum or calibrated value by different arbitrarily selected percentages. An objective function is used to represent the overall changes in model results because of a change in the optimum aquifer characteristic value.

For this sensitivity analysis, the recovery efficiency was used as an objective function. Relative changes in the objective function values (recovery efficiency values) were related to relative changes in the different aquifer characteristics. Each of the selected aquifer characteristic values was changed individually while keeping the other values unchanged. According to Simon (1988), the first relative change in the recovery efficiency value from the baseline simulation value can be defined by:

$$REFFREL_i = \frac{ACV_b(REFF_i - REFF_b)}{REFF_b(ACV_i - ACV_b)} \qquad (16)$$

where,

$REFFREL_i$ is the relative change in the recovery efficiency;

$REFF_i$ is the recovery efficiency for a given change in an aquifer characteristic value;

$REFF_b$ is the recovery efficiency for the baseline simulation;

ACV_i is the changed or modified aquifer characteristic value; and

ACV_b is the aquifer characteristic value used in the baseline simulation.

Subsequent relative changes can be defined by:

$$REFFREL_i = \frac{ACV_b(REFF_i - REFF_{i-1})}{REFF_b(ACV_i - ACV_{i-1})}. \qquad (17)$$

For this sensitivity analysis, the parameters were divided into two categories—hydraulic and transport. The general results from the two categories, which are described in the following sections, indicated that the permeability values of the upper and lower flow zones were the most important factors and produced the greatest changes in the relative sensitivity of the recovery efficiency (fig. 20A-C). In second place of importance,

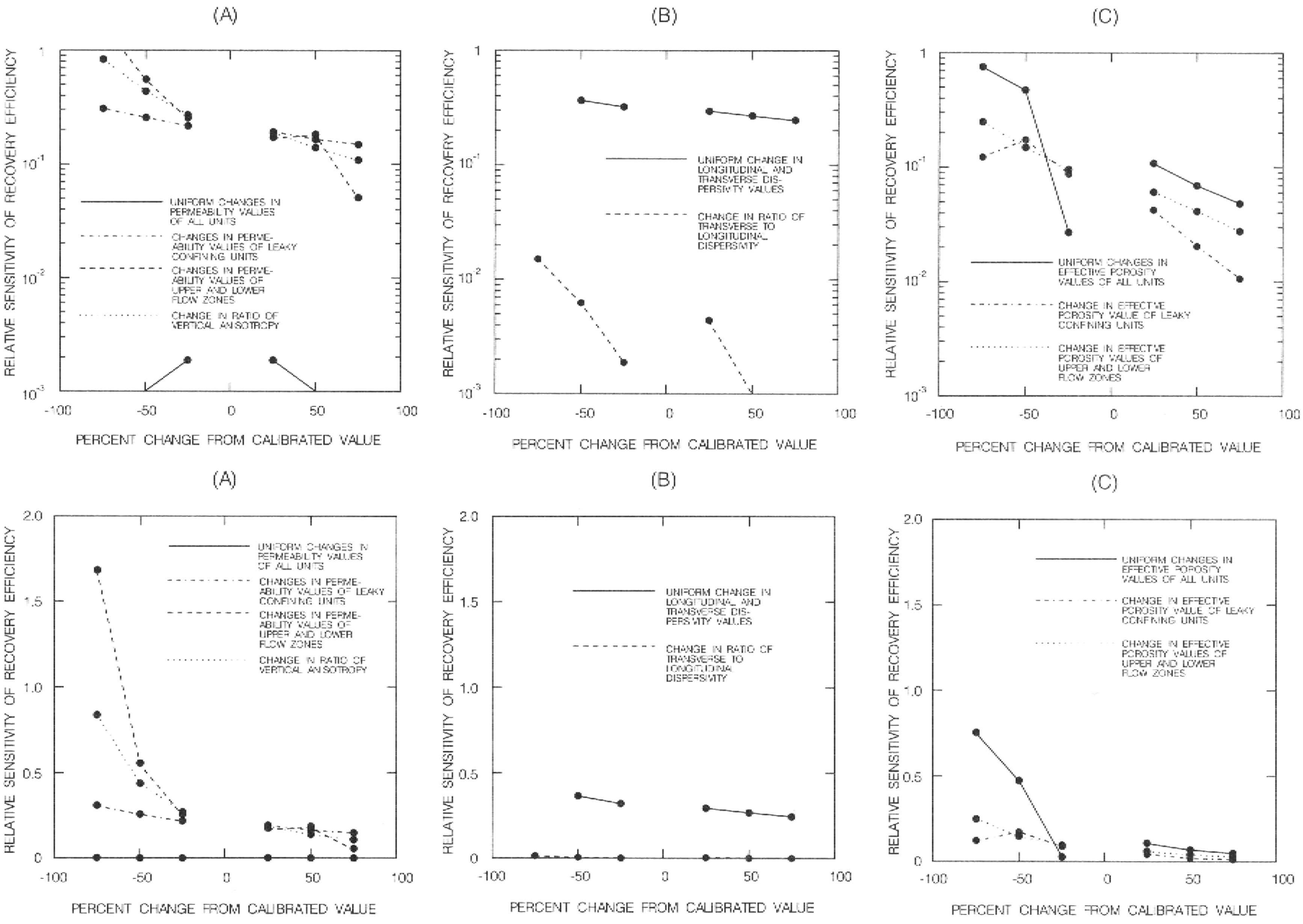


Figure 20. Relative sensitivity of recovery efficiency to variations in (A) permeability values and vertical anisotropy ratio, (B) longitudinal and transverse dispersivities and the ratio of transverse to longitudinal dispersivities, and (C) effective porosity.

but of about equal significance between them, are changes in the relative sensitivity of the recovery efficiency, produced by changing all the porosity values (porosity values of the upper and lower flow zones and the leaky confining units) and those produced by changing the vertical anisotropy ratio (fig. 20C). The fact that permeability, vertical anisotropy, and porosity are the most important factors indicates that the advection process is the most important transport process for this study. Another general observation is that the effect of changing the characteristic values on the relative sensitivity of the recovery efficiency increases when the values are decreased and decreases when the values are increased for all cases (figs. 20A-C).

Permeability and Vertical Anisotropy

The aquifer permeability determines the specific discharge or Darcy's velocity (eq. 6), which in turn, is combined with the effective aquifer porosity to determine the average pore-water velocity (eq. 10). The average pore-water velocity is directly used in the advective term of the transport equation (eq. 14) and indirectly used through the hydrodynamic dispersion tensor (eq. 12) in the dispersive term of the transport equation (eq. 14). Uncertainty in the permeability value would, therefore, affect the advective and dispersive components of the transport computations. The sensitivity of the model to the permeability value was limited to changing the magnitude of the permeability tensor and the vertical anisotropy. Other factors having potential effects on the permeability, such as horizontal anisotropy and heterogeneity, were not considered in this analysis because of the lack of available information.

The magnitude of the permeability was changed in three different ways: (1) uniform changes in all permeability values, (2) changes in permeability values of the leaky confining units, and (3) changes in permeability values of the upper and lower flow zones. The permeability values of the upper and lower flow zones (seemingly the most important in the permeability category) produced the greater changes in the relative sensitivity of the recovery efficiency when the calibrated value was decreased or increased, but showed greater effects when the permeability values were decreased (fig. 20A). Changes in the permeability value of the leaky confining units indicated some sensitivity when the value was increased or decreased by 25 percent, but for greater changes the relative sensitivity was not significantly affected (fig. 20A). It can be inferred from figure 20A that a uniform change in the permeabilities of model layers representing all flow zones and leaky confining units produced an insignificant effect on the recovery efficiency.

Vertical anisotropy, the ratio of vertical to horizontal permeability, was also studied. Changes in the ratio of horizontal to vertical permeability produced the second greatest changes in the relative sensitivity of the recovery efficiency in the permeability category (fig. 20A).

Hydrodynamic Dispersion and Effective Porosity

The hydrodynamic dispersion tensor describes the combined effects of the flow field, aquifer matrix, and molecular diffusion on the transport of solute particles (eq. 13). Flow field and aquifer matrix effects are represented by mechanical dispersion (eq. 12), whereas molecular diffusion is described by Fick's law. The effect of hydrodynamic dispersion on the relative sensitivity of recovery efficiency was studied through the different components of the hydrodynamic dispersion coefficient. The longitudinal and transverse dispersivities represent the dispersive mechanisms of the process. Although molecular diffusion is also a component of the hydrodynamic dispersion coefficient, it is widely recognized among scientists that the effect of molecular diffusion is negligible when compared to longitudinal and transverse dispersivities. Therefore, no attempt was made to study the effects of changing the coefficient of molecular diffusion in this study.

Two different tests were made for the longitudinal and transverse dispersivity values. Both dispersivity values were simultaneously changed by the same percentage in the first test, keeping the ratio of transverse to longitudinal dispersivity equal to 1/10. The ratio of transverse to longitudinal dispersivity was changed in the second test, keeping the longitudinal dispersivity value constant while changing the transverse dispersivity value. The results from the analysis indicated that the uniform change in both transverse and longitudinal dispersivity values produced more significant changes in the relative sensitivity of the recovery efficiency than when the ratio of transverse to longitudinal dispersivities was changed (fig. 20B). In both cases, the relative sensitivity of the recovery efficiency decreased as the dispersivity values or ratio of transverse to longitudinal dispersivity were increased (fig. 20B).

Effective porosity is a factor in the ground-water hydraulic equation (eq. 7) and the advective-dispersive solute-transport equation (eq. 14) in the storage term. However, this porosity has a double effect on the advective dispersive solute-transport equation (eq. 14). In addition to the effect on the storage term for the transport equation, the effective porosity value is combined with the specific discharge (obtained from the ground-water flow equation) to determine the average pore-water velocities, which are used to represent the advection term in the transport equation (eq. 14).

The effective porosity values were changed in three different ways: (1) the porosity values of all the different layers representing the hydrogeologic units were changed by the same percentage from their calibrated values, (2) changes were made to porosity values of the upper and lower flow zones, and (3) changes were made to porosity values of the leaky confining units. Results from the analysis indicated that the most significant changes in the relative sensitivity of the recovery efficiency (and seemingly the most important in the hydrodynamic dispersion category) were produced by changing the porosity values of all layers using the same percentage (figs. 20B and 20C). The second most significant changes to the relative sensitivity of the recovery efficiency were produced by changing the porosity of the upper and lower flow zones (fig. 20C). Smaller changes in the relative sensitivity of the recovery efficiency were produced when porosity values of the leaky confining units were changed (fig. 20C). These results suggest that a specific combination of porosity values of the flow zones and the leaky confining units is needed to provide an adequate representation of the transport system.

LIMITATIONS

Confidence in the model and in the resulting simulations is limited by a number of factors. These factors can be segregated into two categories—the hydrogeologic information and the aspects of the model code. Among the hydrogeologic information, the most important limiting factors in this study were lack of:

- Complete understanding about the spatial variability of the hydraulic conductivity or permeability values (heterogeneity),
- Field information on changes in the magnitude of the hydraulic conductivity or permeability in the horizontal and vertical directions (horizontal and vertical anisotropies),
- Field information on the porosity values,
- Knowledge about the potential effect of fractures or solution cavities on the flow and transport processes (result of effective secondary porosity),
- Real SISRF tests in the Cape Coral area, and
- Assumptions made to represent the top and bottom boundary conditions as having constant solute concentration and pressure.

The computer code (QSUTRA) used in this study has some intrinsic limitations:

- The fact that the code provides only for two-dimensional simulations precluded the study of the effect of background regional flow on the displacement of the injected water when the cylindrical (radial) coordinate option was used;
- When the Cartesian coordinate option is used, the assumption of vertical homogeneity and isotropy must be made, and such an assumption would be unrealistic for the Cape Coral site; and
- In QSUTRA, the solute-transport equation for transient compressible fluid flow is represented by an analogous numerical expression where porosity, thickness, and fluid density are kept constant by producing a mass-balance error. This affects the determination of velocities and dispersion coefficients (Goode, 1990; 1992). However, this intrinsic error is not expected to greatly affect the simulation of field-scale problems in which the uncertainty and variability of the modeled aquifer characteristics overshadow the potential effects from the intrinsic mass-balance error.

SUMMARY AND CONCLUSIONS

A preliminary assessment of subsurface injection, storage, and recovery of freshwater (SISRF) was made as a potential alternative to the growing water-supply problems of Cape Coral in Lee County, southwestern Florida. A digital modeling approach was used for this preliminary assessment to research the actual potential of SISRF without having to spend the large amounts of money required for real field testing of this technique.

The hydrogeologic framework used for this study was modified or developed from the interpretation of data from previous studies. Aquifer characteristics were estimated from interpretation of data from previous studies. A combination of caliper and flow-velocity borehole geophysical logs was used to estimate the percentages of flow entering different flow zones. These percentages of flow and information on the aquifer transmissivity were used to estimate permeability values for the different flow zones.

A general presentation was made of the density-dependent ground-water flow and advective dispersive solute-transport equations. A modified version of the computer code SUTRA (QSUTRA) and a cylindrical coordinates grid were used for this preliminary assessment because of the lack of information required to represent the real three-dimensional ground-water flow and transport system.

Dispersive characteristics were estimated on the basis of data from a previous study at the Lee County Water Treatment Plant. This was accomplished by calibrating a model for the Lee County Water Treatment Plant site and testing this model using field data from a previous study. A second model was made for the Cape Coral area using local hydraulic characteristics and adopting the dispersive characteristics estimated for the Lee County Water Treatment Plant site model.

A series of 28 hypothetical tests of subsurface injection, storage, and recovery of freshwater were made for the lower Hawthorn aquifer in Cape Coral using the digital modeling technique to assess the efficiency of this operation in the subject aquifer. A baseline simulation was used as reference to compare the effects of changing some operational factors on the recovery efficiency. A recovery efficiency of 64 percent was estimated for the baseline simulation. This recovery efficiency represents the total amount of water pumped during the recovery phase before the 250-milligrams per liter chloride limit is reached divided by the total amount of injected water. The effects of the following operational factors were assessed using the model: rates of injection and recovery; volume of water injected; storage time; injection into selected flow zones; successive cycles of injection, storage, and recovery; and chloride concentrations in injected and native aquifer waters.

A summary of the simulation results from the model, which is based on the limited knowledge of the aquifer, indicates that the recovery efficiency increased when the injection rate and recovery rates were increased, and when the ratio of recovery rate to injection rate was increased. Recovery efficiency decreased when the amount of water injected was increased; decreased slightly when the storage time was increased; was not changed significantly when the water was injected to a specific flow zone; increased with successive cycles of injection, storage, and recovery; and decreased when the chloride concentrations in either the injected water or native aquifer water were increased. The different simulation results for storage time might be unrealistic because the cylindrical coordinates used in the model did not consider the regional background flow, which was an important factor in previous studies.

The higher recovery efficiencies were obtained for three simulation tests for which the duration of injection and recovery phases was shorter. This is expected because of the nature of the conceptual system in which migration of the solute particles to areas beyond the vertical boundaries will reduce the recoverability for tests of longer duration. The recovery efficiency fluctuated from its baseline value of 64 percent to an upper value of about 100 percent and to a lower value of 22 percent in all of the simulations.

Interlayer solute mass movement across the upper and lower boundaries seems to be the most important factor affecting the recovery efficiency. A simulation that was conducted with the same parameters used for the baseline simulation, but representing the top and bottom boundaries as impermeable (no flow and no solute transport), yielded a recovery efficiency value of 83 percentage points. This value is 19 percentage points higher than the estimated value from the baseline simulation showing that this boundary is important in determining the recovery efficiency, and that using constant pressure and constant solute concentration, boundaries will yield more conservative estimates of the recovery efficiency.

The sensitivity analysis was performed applying the relative sensitivity technique in which changes in the different factors and model responses are normalized to make a meaningful comparison of the model responses due to changes in the different factors. Two categories of factors were recognized for the sensitivity analysis—aquifer permeability and hydrodynamic dispersion. Several combinations of changes were made for factors of the two categories. For instance, a factor was changed only for a specific flow zone. The general results from the sensitivity analysis indicated that the permeability values of the upper and lower flow zones are the most important factors, producing the overall greater changes in the relative sensitivity of the recovery efficiency. In second place of importance, but of about equal significance between them, are changes in the relative sensitivity of the recovery efficiency, produced by changing all the porosity values (porosity values of the upper and lower flow zones and the confining beds) and those produced by changing the vertical anisotropy ratio.

Model results indicate that high recovery efficiencies (from 64 to about 100 percent) can be achieved for different SISRF operational schemes. Two successive injection, storage, and recovery cycles can increase the recovery efficiency from 60 to about 80 percent. Combinations of different operational factors also can be used to maintain high recovery efficiencies. The advective factors (pore-water velocities derived from permeability and porosity values) were apparently the most important to the model sensitivity in the Cape Coral area. However, the dispersivity values used for the lower Hawthorn aquifer in the Cape Coral area model were not field values, but values that were extrapolated from the model of the lower Hawthorn aquifer at the Lee County Water Treatment Plant site. These dispersivity values might not be representative of the actual dispersive characteristics of the lower Hawthorn aquifer in the Cape Coral area. The model presented in this report is a generalized version of the actual hydrogeologic system and could be refined if additional information on the advective and dispersive characteristics of the aquifer is made available.

REFERENCES CITED

Anderson, M.P., 1984, Movement of contaminants in groundwater: Groundwater transport-advection and dispersion, *in* Groundwater Contamination: Washington, D.C., National Academy Press, p. 37-45.

Bear, J., 1979, Hydraulics of Groundwater: New York, McGraw-Hill, 569 p.

Brown, R.H., 1963, Estimating the transmissivity of an artesian aquifer from the specific capacity of a well, *in* Methods of Determining Permeability, Transmissivity, and Drawdown: U.S. Geological Survey Water-Supply Paper 1536-I, p. 336-338.

City of Cape Coral Planning Division, 1988, Comprehensive plan—future land use element—Volume 1. Population and Demographics, 31 p.

Fitzpatrick, D.J., 1986a, Tests for injecting, storing, and recovering freshwater in a saline artesian aquifer, Lee County, Florida: U.S. Geological Survey Water-Resources Investigations Report 85-4249, 53 p.

——— 1986b, Hydrogeologic conditions and saline-water intrusion, Cape Coral, Florida, 1978-81: U.S. Geological Survey Water-Resources Investigations Report 85-4231, 31 p.

Florida Department of Environmental Protection, 1993, Drinking water standards, monitoring and reporting: Chapter 17-550, Florida Administrative Code, 38 p.

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

Gelhar, L.W., and Axness, C.L., 1983, Three-dimensional stochastic analysis of macrodispersion in aquifers. Water Resources Research, v. 19, no. 1, p. 161-180.

Gelhar, L.W., Gutjahr, A.L., and Naff, R.L., 1979, Stochastic analysis of macrodispersion in a stratified aquifer: Water Resources Research, v. 15, no. 6, p. 1387-1397.

Goode, D.J., 1990, Governing equations and model approximation errors associated with the effects of fluid-storage transients on solute transport in aquifers: U.S. Geological Survey Water-Resources Investigations Report 90-4156, 20 p.

——— 1992, Modeling transport in transient ground-water flow: An unacknowledged approximation: Ground Water, v. 30, no. 2, p. 257-261.

Kimbler, O.K., Kazmann, R.G., and Whitehead, W.R., 1975, Cyclic storage of fresh water in saline aquifers: Louisiana Water Resources Research Institute Bulletin 10, Baton Rouge, Louisiana State University, 78 p.

Knapp, M.S., Burns, W.S., Sharp, T.S., and Shih, George, 1984, Preliminary water resources assessment of the mid and lower Hawthorn aquifers in western Lee County, Florida: South Florida Water Management District Technical Report 84-10, 106 p.

Kuiper, L.K., 1987, Computer program for solving groundwater flow equations by the preconditioned conjugate gradient method: U.S. Geological Survey Water-Resources Investigations Report 87-4091, 34 p.

La Rose, H.R., 1990, Geohydrologic framework and an analysis of a well-plugging program, Lee County, Florida: U.S. Geological Survey Water-Resources Investigations Report 90-4063, 26 p., 1 pl.

Lee County Planning Department, 1977, Climate of Lee County: Lee County Planning Department report, 62 p.

Mercado, A., 1984, A note on micro and macrodispersion: Ground Water, v. 22, no. 6, p. 790-791.

Merritt, M.L., 1985, Subsurface storage of freshwater in south Florida: A digital model analysis of recoverability: U.S. Geological Survey Water Supply Paper 2261, 44 p.

Merritt, M.L., Meyer, F.W., Sonntag, W.H., and Fitzpatrick, D.J., 1983, Subsurface storage of freshwater in south Florida: A prospectus: U.S. Geological Survey Water-Resources Investigations Report 83-4214, 69 p.

Meyer, F.W., 1989, Subsurface storage of liquids in the Floridan aquifer system in south Florida: U.S. Geological Survey Open-File Report 88-477, 25 p.

Missimer and Associates, Inc., 1984, Hydrogeology of the lower part of the Hawthorn aquifer system, Cape Coral Florida: Consultant's Report, 84 p.

——— 1985, Cape Coral reverse osmosis wellfield final construction report and operational and maintenance recommendations: v. 1, 58 p.

National Oceanic and Atmospheric Administration, 1944-88, Climatological data, Florida: Monthly summaries for 1984 to 1988.

Quiñones-Aponte, Vicente, Whiteside, D.V., and Zack, Allen, 1989, Single well injection and recovery of freshwater from an aquifer containing saline water in Arecibo, Puerto Rico: U.S. Geological Survey Water-Resources Investigations Report 88-4037, 19 p.

Scheidegger, A.E., 1961, General theory of dispersion in porous media: Journal of Geophysical Research, v. 66, p. 3273-3278.

Simon, E., 1988, Parameter sensitivity analysis for hydrologic simulation models: Water International, v. 13, p. 46-56.

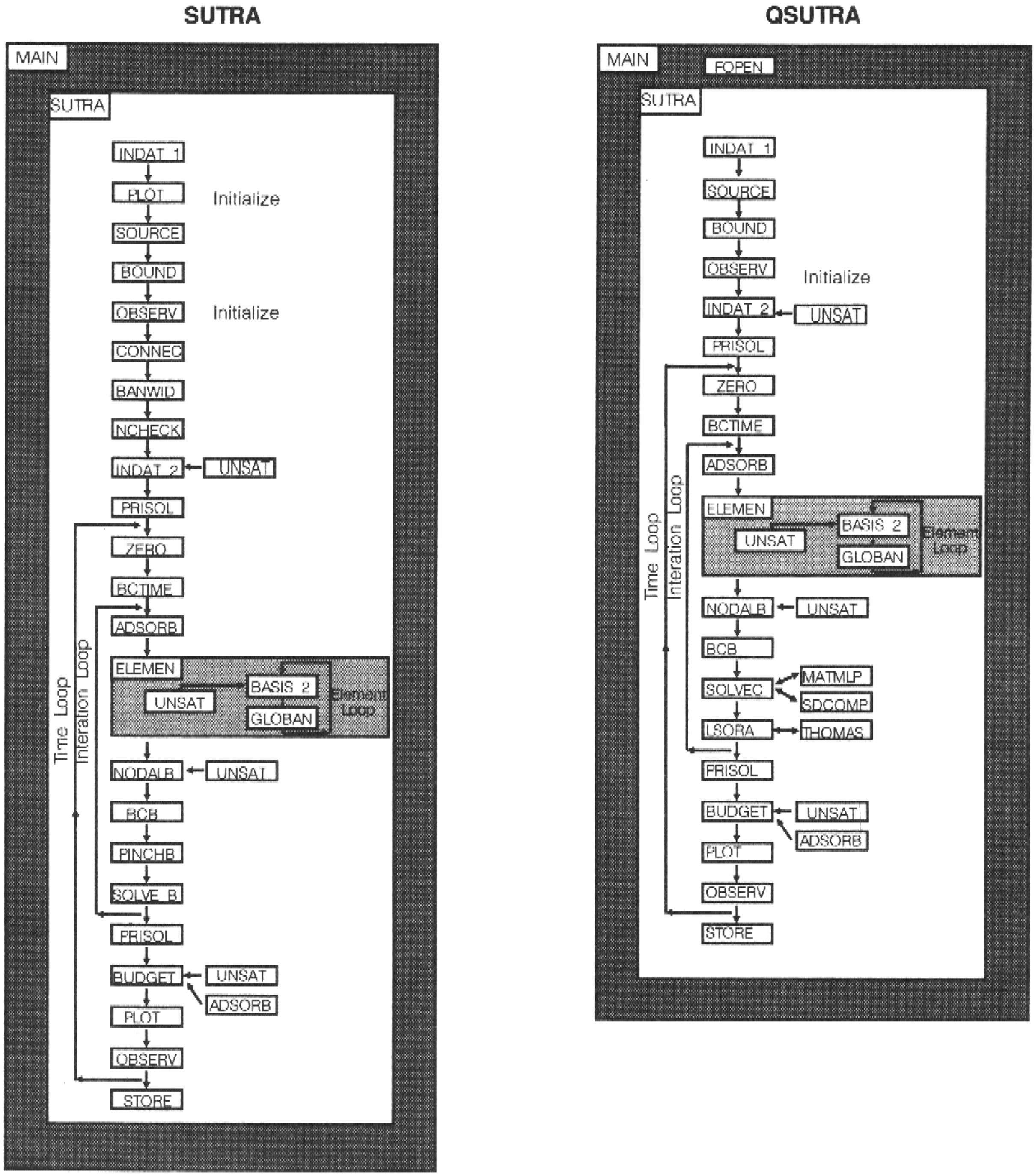
Voss, C.I., 1984, A finite-element simulation model for saturated-unsaturated, fluid density-dependent groundwater flow with energy transport or chemically reactive single species solute transport: U.S. Geological Survey Water-Resources Investigations Report 84-4369, 409 p.

Wedderburn, L.A., Knapp, M.S., Waltz, D.P., and Burns, W.S., 1982, Hydrogeologic reconnaissance of Lee County, Florida—Part 1: South Florida Water Management District Technical Report 82-1, 192 p.

Young, D.M., 1954, Iterative method for solving partial differential equations of the elliptic type: Transactions American Mathematical Society, v. 76, p. 92-111.

Appendix I

APPENDIX 1. HIERARCHIC LEVELS OF SUBPROGRAMS IN SUTRA NAD QSUTRA SHOWING MAJOR CHANGES TO THE ORIGINAL CODE (SUTRA)


SUTRA

QSUTRA

Appendix II

QSUTRA Program Listing (Model Version 1284-2DICG Modified for Regular Grid)

[MODIFIED, Changes as per updated version of SUTRA, version V06902D; NEW, Changes made as part of QSUTRA implementation, version 1284-2DICG]

```
C     SUTRA              M A I N   P R O G R A M   SUTRA-VERSION 1284-2D A10.....
C______________________________________________________________________ A20.....
C|                                                                    |A30.....
C|                                                                    |A40.....
C|                   UNITED STATES GEOLOGICAL SURVEY                  |A50.....
C|   GROUND-WATER FLOW AND ENERGY OR SOLUTE TRANSPORT SIMULATION MODEL |A60.....
C|                                                                    |A70.....
C|                                                                    |A80.....
C|                                                                    |A90.....
C|                                                                    |A100....
C|                                                                    |A110....
C|                    _______________________                         |A110....
C|                   |                       |                        |A120....
C|                   |   S   U   T   R   A   |                        |A130....
C|                   |_______________________|                        |A140....
C|                                                                    |A150....
C|                                                                    |A160....
C|                 Saturated    Unsaturated    TRAnsport              |A170....
C|                 =            =              ===                    |A180....
C|                                                                    |A190....
C|                                                                    |A200....
C|                                                                    |A210....
C|   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     |A220....
C|   * ->saturated and/or unsaturated groundwater flow          *     |A230....
C|   * ->either single species reactive solute transport        *     |A240....
C|   *    or thermal energy transport                           *     |A250....
C|   * ->two-dimensional areal or cross-sectional simulation    *     |A260....
C|   * ->either cartesian or radial/cylindrical coordinates     *     |A270....
C|   * ->hybrid galerkin-finite-element method and              *     |A280....
C|   *    integrated-finite-difference method                   *     |A290....
C|   *    with two-dimensional quadrilateral finite elements    *     |A300....
C|   * ->finite-difference time discretization                  *     |A310....
C|   * ->non-linear iterative, sequential or steady-state       *     |A320....
C|   *    solution modes                                        *     |A330....
C|   * ->optional fluid velocity calculation                    *     |A340....
C|   * ->optional observation well output                       *     |A350....
C|   * ->modified for regular grid only  - to minimize storage  *     |NEW
C|   * ->optional fluid mass and solute mass or energy budget   *     |A370....
C|   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     |A380....
C|                                                                    |A390....
C|                                                                    |A400....
C|                                                                    |A410....
C|     Complete explanation of function and use of this code         |A420....
C|     is given in :                                                  |A430....
C|                                                                    |A440....
C|     Voss, Clifford I., 1984, SUTRA: A Finite-Element               |A450....
C|          Simulation Model for Saturated-Unsaturated                |A460....
C|          Fluid-Density-Dependent Ground-Water Flow                 |A470....
C|          with Energy Transport or Chemically-Reactive              |A480....
C|          Single-Species Solute Transport, U.S. Geological          |A490....
C|          Survey Water-Resources Investigations Report              |A500....
C|          84-4369.                                                  |A510....
C|                                                                    |A520....
C|                                                                    |A530....
C|                                                                    |A540....
C|     Users who wish to be notified of updates of the SUTRA         |A550....
C|     code and documentation may be added to the mailing            |A560....
C|     by sending a request to :                                      |A570....
C|                                                                    |A580....
C|                       Chief Hydrologist - SUTRA                    |A590....
C|                        U.S. Geological Survey                      |A600....
C|                         431 National Center                        |A610....
C|                        Reston, Virginia 22092                      |A620....
```

```
C|                              USA                                    |A630....
C|                                                                     |A640....
C|     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     |A650....
C|     * The SUTRA code and documentation were prepared under a  *     |A660....
C|     * joint research project of the U.S. Geological Survey,   *     |A670....
C|     * Department of the Interior, Reston, Virginia, and the   *     |A680....
C|     * Engineering and Services Laboratory,  U.S. Air Force    *     |A690....
C|     * Engineering and Services Center, Tyndall A.F.B.,        *     |A700....
C|     * Florida.  The SUTRA code and documentation are          *     |A710....
C|     * available for unlimited distribution.                   *     |A720....
C|     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     |A730....
C|                                                                     |A740....
C|                                                                     |A750....
C|                                                                     |A760....
C|_____________________________________________________________________|A770....
C                                                                       A780....
C                                                                       A790....
C                                                                       A800....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               A810....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8                     MODIFIED
      COMMON/LGE/ RM,RV,IMV                                             A820....
C     COMMON/LGEV/ RV                                                   A830....
C     COMMON/LGEMV/ IMV                                                 A840....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          NEW
     1   NSOP,NSOU,NBCN
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            NEW
      COMMON/OBS/ NOBSN,NTOBSN,NOBCYC,ITCNT                             A890....
      CHARACTER*1 TITLE1(80),TITLE2(80)                                 A900....
      CHARACTER*6 SIMULA(2)                                             A910....
      CHARACTER*80 UNAME,ENAME,FNAME                                    MODIFIED
        INTEGER RMDIM,RVDIM,IMVDIM,RMDIMA,RVDIMA,IMVDMA                 NEW
      DIMENSION KRV(100),FNAME(8),IUNIT(8)                              MODIFIED
C                                                                       A930....
C                                                                       A940....
C_____________________________________________________________________  A950....
C|* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *|  A960....
C|* ***************************************************************  *|  A970....
C|* *                                                              * *|  A980....
C|* *   The three arrays that need be dimensioned                  * *|  A990....
C|* *    are dimensioned as follows:                               * *|  A1000...
C|* *                                                              * *|  A1010...
C|* *   DIMENSION  RM( RMDIM), RV( RVDIM), IMV(IMVDIM)             * *|  A1020...
C|* *                                                              * *|  A1030...
C|* *   RMDIM >= NN*(NBIP + NBIS + 9)                                   NEW
C|* *                                                              * *|  A1050...
C|* *   RVDIM >= (( NNV*NN + (NEV+8)*NE + NBCN*3                   * *|  A1060...
C|* *               + (NOBS+1)*(NTOBS+2)*2 + NTOBS + 5 ))          * *|  A1070...
C|* *                                                              * *|  A1080...
C|* *   IMVDIM >= (( NE*8 + NN +  NSOP + NSOU                      * *|  NEW
C|* *               + NBCN*2 + NOBS + NTOBS + 12 ))                * *|  A1100...
C|* *                                                              * *|  A1110...
C|* *   where:                                                     * *|  A1120...
C|* *                                                              * *|  A1130...
C|* *    NNV = 30                                                  * *|  A1140...
C|* *    NEV = 10                                                  * *|  A1150...
C|* *    NBCN = NPBC + NUBC                                        * *|  A1160...
C|* *                                                              * *|  A1170...
C|* *    NBIP = 5    for a regular grid                            * *|  NEW
C|* *    NBIS = 9    for a regular grid                            * *|  NEW
C|* *    NBIA = 13   for a regular grid                            * *|  NEW
```

```
C|* *                                                                    * *|
C|* *   and:                                                             * *|   A1180...
C|* *                                                                    * *|   A1190...
C|* *    NN = number of nodes in finite element mesh                     * *|   A1200...
C|* *    NE = number of elements in finite element mesh                  * *|   A1210...
C|* *    NOBS = number of observation nodes in mesh                      * *|   A1220...
C|* *    NTOBS = maximum number of time steps with observations          * *|   A1230...
C|* *    NSOP = number of fluid mass source nodes in mesh                * *|   A1250...
C|* *    NSOU = number of energy or solute mass source nodes             * *|   A1260...
C|* *    NPBC = number of specified pressure nodes in mesh               * *|   A1270...
C|* *    NUBC   number of specified concentration or temperature         * *|   A1280...
C|* *           nodes in mesh                                            * *|   A1290...
C|* *                                                                    * *|   A1300...
C|* *                                                                    * *|   A1310...
C|* *   The three arrays must be given dimensions just below.            * *|   A1320...
C|* *                                                                    * *|   A1330...
C|* *                                                                    * *|   A1330...
C|* *   REMEMBER also to change the dimension values,                    * *|  A1330.1MODIFIED
C|* *    RMDIM, RVDIM and IMVDIM in the three assignment                 * *|  A1330.2MODIFIED
C|*.*    statements below the DIMENSION statement!                       * *|  A1330.3MODIFIED
C|* *                                                                    * *|  A1330.4MODIFIED
C|* *   AND ALSO :                                                       * *|  A1330.5MODIFIED
C|* *    Two files must be permanently assigned just below for           * *|  A1330.6MODIFIED
C|* *    your computer installation.  One file captures error            * *|  A1330.7MODIFIED
C|* *    output written during subsequient file opening.  The            * *|  A1330.8MODIFIED
C|* *    other file contains the unit numbers and file names             * *|  A1330.9MODIFIED
C|* *    to be assigned as SUTRA input and output files                  * *|  A1331.0MODIFIED
C|* *    for each simulation.                                            * *|  A1331.1MODIFIED
C|* *                                                                    * *|  A1331.2MODIFIED
C|* *    STANDARD ASSIGNMENTS TO BE MADE:                                * *|  A1331.3MODIFIED
C|* *    for Error Output:                                               * *|  A1331.4MODIFIED
C|* *        Filename is contained in ENAME                              * *|  A1331.5MODIFIED
C|* *        Unit Number is contained in K00                             * *|  A1331.6MODIFIED
C|* *    for Simulation Units and Files:                                 * *|  A1331.7MODIFIED
C|* *        Filename is contained in UNAME                              * *|  A1331.8MODIFIED
C|* *        Unit Number is contained in K0                              * *|  A1332.9MODIFIED
C|* *                                                                    * *|  A1333.0MODIFIED
C|* ********************  D I M E N S I O N S  ******************** *|  A1341..MODIFIED
C|* ***************************************************************** *|   A1340...
      DIMENSION   RM(1100000), RV(2250000), IMV(500000)                     A1350...
        RMDIMA=1100000                                                      MODIFIED
        RVDIMA=2250000                                                      MODIFIED
        IMVDMA= 500000                                                      MODIFIED
C
C|* *****  S T A N D A R D   F I L E   A S S I G N M E N T S  ***** *|  A1346..MODIFIED
C|*   E R R O R   O U T P U T                                      * *|  A1347..MODIFIED
      ENAME = 'SUTRA.ERR'                                                  A1348..MODIFIED
      K00 = 1                                                              A1349..MODIFIED
C|*   S I M U L A T I O N   U N I T S   A N D   F I L E S          * *|  A1350..MODIFIED
      UNAME = 'SUTRA.FIL'                                                  A1351..MODIFIED
CDJGOODE      K0 = 100                                                     A1352..MODIFIED
      K0 = 99
C|*                                                                 * *|  A1352.1MODIFIED
C     -------> Required Format of Unit K0 :                         * *|  A1352.2MODIFIED
                                                                    * *|  A1352.3MODIFIED
                 V A R I A B L E          F O R M A T               * *|  A1352.4MODIFIED
C|*                                                                 * *|  A1352.5MODIFIED
C|*              Unit Number for K1          (free format)          * *|  A1352.6MODIFIED
C|*               File Name for K1            (A80)                 * *|  A1352.7MODIFIED
```

```
C|*             Unit Number for K2        (free format)         * *|  A1352.8MODIFIED
C|*              File Name for K2          (A80)                * *|  A1352.9MODIFIED
C|*             Unit Number for K3        (free format)         * *|  A1353..MODIFIED
C|*              File Name for K3          (A80)                * *|  A1353.5MODIFIED
C|*             Unit Number for K4        (free format)         * *|  A1354..MODIFIED
C|*              File Name for K4          (A80)                * *|  A1355..MODIFIED
C|*                                                             * *|  A1356..MODIFIED
C|*                                                             * *|  A1357..MODIFIED
C|*   The last two lines need not be included if UNIT-K4 will not * *|  A1358..MODIFIED
C|*   be used.  This file has six or eight lines.               * *|  A1359..MODIFIED
C|* ************************************************************ *|  A1360...
C|* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *|  A1370...
C|_______________________________________________________________|  A1380...
C                                                                    A1390...
C                                                                    A1400...
C ---> Programmers making code changes that affect dimensions must   A1401..MODIFIED
C --->  check and change the following assignments for NNV and NEV:  A1402..MODIFIED
C                                                                    A1403..MODIFIED
C                                                                    A1408..MODIFIED
C                                                                    A1409..MODIFIED
C                                                                    A1410...
C.....ASSIGN UNIT NUMBERS AND OPEN FILE UNITS FOR THIS SIMULATION    A1412..MODIFIED
      CALL FOPEN(UNAME,ENAME,FNAME,IUNIT,NFILE)                      A1414..MODIFIED
C                                                                    A1416..MODIFIED
C                                                                    A1410...
C.....INPUT DATASET 1:  INPUT DATA HEADING                           A1420...
C.....( SET ME=-1 FOR SOLUTE TRANSPORT, ME=+1 FOR ENERGY TRANSPORT ) A1430...
      READ(K1,100) SIMULA                                            A1440...
  100 FORMAT(2A6)                                                    A1450...
      WRITE(K3,110)                                                  A1460...
  110 FORMAT(1H1,131(1H*)////3(132(1H*)////)////                     A1470...
     1   47X,' SSSS   UU  UU  TTTTTT  RRRRR     AA  '/               A1480...
     2   47X,'SS    S UU  UU  T TT T  RR  RR   AAAA '/               A1490...
     3   47X,'SSSS    UU  UU    TT    RRRRR   AA  AA'/               A1500...
     4   47X,'    SS  UU  UU    TT    RR R    AAAAAA'/               A1510...
     5   47X,'SS  SS  UU  UU    TT    RR RR   AA  AA'/               A1520...
     6   47X,' SSSS    UUUU     TT    RR  RR  AA  AA'/               A1530...
     7   7(/),37X,'U N I T E D    S T A T E S   ',                   A1540...
     8   'G E O L O G I C A L   S U R V E Y'////                     A1550...
     9   45X,'SUBSURFACE FLOW AND TRANSPORT SIMULATION MODEL'/       A1560...
     *   //59X,'-VERSION 1284-2DICG MODIFIED FOR A REGULAR GRID-'/// A1570NEW
     A   36X,'*  SATURATED-UNSATURATED FLOW AND SOLUTE OR ENERGY',   A1580...
     B   ' TRANSPORT  *'////4(////1X,132(1H*)))                      A1590...
C                                                                    A1600...
      IF(SIMULA(1).NE.'SUTRA ') GOTO 115                             A1610...
      IF(SIMULA(2).EQ.'SOLUTE') GOTO 120                             A1620...
      IF(SIMULA(2).EQ.'ENERGY') GOTO 140                             A1630...
  115 WRITE(K3,116)                                                  A1640...
  116 FORMAT(1H1/////20X,'* * * * *   ERROR IN FIRST DATA CARD--',   A1650...
     1   '------DATA INPUT HALTED FOR CORRECTIONS   * * * * *')      A1660...
      ENDFILE(K3)                                                    A1661...
      STOP                                                           A1670...
  120 ME=-1                                                          A1680...
      WRITE(K3,130)                                                  A1690...
  130 FORMAT(1H1//132(1H*)///20X,'* * * * *   S U T R A   S O L U ', A1700...
     1   'T E   T R A N S P O R T   S I M U L A T I O N   * * * * *'// A1710...
     2   /132(1H*)/)                                                 A1720...
      GOTO 160                                                       A1730...
  140 ME=+1                                                          A1740...
      WRITE(K3,150)                                                  A1750...
```

```
  150 FORMAT(1H1//132(1H*)///20X,'* * * * *   S U T R A   E N E R ',    A1760...
    1   'G Y   T R A N S P O R T   S I M U L A T I O N   * * * * *'//  A1770...
    2   /132(1H*)/)                                                     A1780...
  160 CONTINUE                                                          A1790...
C                                                                       A1800...
C.....INPUT DATASET 2:  OUTPUT HEADING                                   A1810...
      READ(K1,170) TITLE1,TITLE2                                        A1820...
  170 FORMAT(80A1/80A1)                                                 A1830...
      WRITE(K3,180) TITLE1,TITLE2                                       A1840...
  180 FORMAT(////1X,131(1H-)//26X,80A1//26X,80A1//1X,131(1H-))          A1850...
C.....OUTPUT FILE UNIT ASSIGNMENTS                                      A1850.5MODIFIED
      WRITE(K3,202) (IUNIT(NF),FNAME(NF),NF=1,3)                        A1851..MODIFIED
  202 FORMAT(/////11X,'F I L E   U N I T   A S S I G N M E N T S'//     A1852..MODIFIED
     1   13X,'INPUT UNITS:'/                                            A1853..MODIFIED
     2   13X,' SIMULATION DATA    ',I3,4X,'ASSIGNED TO ',A80/           A1854..MODIFIED
     3   13X,' INITIAL CONDITIONS ',I3,4X,'ASSIGNED TO ',A80//          A1855..MODIFIED
     4   13X,'OUTPUT UNITS:'/                                           A1856..MODIFIED
     5   13X,' SIMULATION RESULTS ',I3,4X,'ASSIGNED TO ',A80)           A1857..MODIFIED
      IF(NFILE.EQ.4) WRITE(K3,203) IUNIT(4),FNAME(4)                    A1858..MODIFIED
  203 FORMAT(13X,' RESTART DATA       ',I3,4X,'ASSIGNED TO ',A80)       A1859..MODIFIED
C.....INPUT AND OUTPUT DATASET 4:  SIMULATION MODE OPTIONS              A1865..MODIFIED
      READ(K1,200) IS,JT,NBI,NPINCH,NPBC,NUBC,NSOP,NSOU,NOBS,NTOBS      A1860NEW
        NN=IS*JT                                                        NEW
        NE=(IS-1)*(JT-1)                                                NEW
      READ(K1,200) IUNSAT,ISSFLO,ISSTRA,IREAD,ISTORE,ITIME              A1870NEW
  200 FORMAT(16I5)                                                      A1880...
      WRITE(K3,205)                                                     A1890...
  205 FORMAT(/////11X,'S I M U L A T I O N   M O D E   ',               A1900...
     1   'O P T I O N S'/)                                              A1910...
      IF(ISSTRA.EQ.1.AND.ISSFLO.NE.1) THEN                              A1920...
       WRITE(K3,210)                                                    A1930...
  210  FORMAT(////11X,'STEADY-STATE TRANSPORT ALSO REQUIRES THAT ',     A1940...
     1    'FLOW IS AT STEADY STATE.'//11X,'PLEASE CORRECT ISSFLO ',     A1950...
     2    'AND ISSTRA IN THE INPUT DATA, AND RERUN.'////////            A1960...
     3    45X,'S I M U L A T I O N   H A L T E D   DUE TO INPUT ERROR') A1970...
       ENDFILE(K3)                                                      A1980...
       STOP                                                             A1990...
      ENDIF                                                             A2000...
      IF(IUNSAT.EQ.+1) WRITE(K3,215)                                    A2010...
      IF(IUNSAT.EQ.0) WRITE(K3,216)                                     A2020...
  215 FORMAT(11X,'- ALLOW UNSATURATED AND SATURATED FLOW:  UNSATURATED',A2030...
     1   ' PROPERTIES ARE USER-PROGRAMMED IN SUBROUTINE   U N S A T')   A2040...
  216 FORMAT(11X,'- ASSUME SATURATED FLOW ONLY')                        A2050...
      IF(ISSFLO.EQ.+1.AND.ME.EQ.-1) WRITE(K3,219)                       A2060...
      IF(ISSFLO.EQ.+1.AND.ME.EQ.+1) WRITE(K3,220)                       A2070...
      IF(ISSFLO.EQ.0) WRITE(K3,221)                                     A2080...
  219 FORMAT(11X,'- ASSUME STEADY-STATE FLOW FIELD CONSISTENT WITH ',   A2090...
     1   'INITIAL CONCENTRATION CONDITIONS')                            A2100...
  220 FORMAT(11X,'- ASSUME STEADY-STATE FLOW FIELD CONSISTENT WITH ',   A2110...
     1   'INITIAL TEMPERATURE CONDITIONS')                              A2120...
  221 FORMAT(11X,'- ALLOW TIME-DEPENDENT FLOW FIELD')                   A2130...
      IF(ISSTRA.EQ.+1) WRITE(K3,225)                                    A2140...
      IF(ISSTRA.EQ.0) WRITE(K3,226)                                     A2150...
  225 FORMAT(11X,'- ASSUME STEADY-STATE TRANSPORT')                     A2160...
  226 FORMAT(11X,'- ALLOW TIME-DEPENDENT TRANSPORT')                    A2170...
      IF(IREAD.EQ.-1) WRITE(K3,230)                                     A2180...
      IF(IREAD.EQ.+1) WRITE(K3,231)                                     A2190...
  230 FORMAT(11X,'- WARM START - SIMULATION IS TO BE ',                 A2200...
     1   'CONTINUED FROM PREVIOUSLY-STORED DATA')                       A2210...
  231 FORMAT(11X,'- COLD START - BEGIN NEW SIMULATION')                 A2220...
      IF(ISTORE.EQ.+1) WRITE(K3,240)                                    A2230...
```

```
      IF(TSTORE.EQ.0) WRITE(K3,241)                                     A2240...
  240 FORMAT(11X,'- STORE RESULTS AFTER EACH TIME STEP ON UNIT-66',      A2250...
     1   ' AS BACK-UP AND FOR USE IN A SIMULATION RE-START')            A2260...
  241 FORMAT(11X,'- DO NOT STORE RESULTS FOR USE IN A ',                A2270...
     1   'RE-START OF SIMULATION')                                      A2280...
C                                                                       A2290...
      IF(ME.EQ.-1)                                                      A2300...
     1   WRITE(K3,245) NN,NE,NPBC,NUBC,NSOP,NSOU,NOBS,NTOBS             A2310NEW
  245 FORMAT(////11X,'S I M U L A T I O N   C O N T R O L   ',          A2320...
     1   'N U M B E R S'//11X,I6,5X,'NUMBER OF NODES IN FINITE-',       A2330...
     2   'ELEMENT MESH'/11X,I6,5X,'NUMBER OF ELEMENTS IN MESH'/         A2340...
     5   11X,I6,5X,'EXACT NUMBER OF NODES IN MESH AT WHICH ',           A2370...
     6   'PRESSURE IS A SPECIFIED CONSTANT OR FUNCTION OF TIME'/        A2380...
     7   11X,I6,5X,'EXACT NUMBER OF NODES IN MESH AT WHICH ',           A2390...
     8   'SOLUTE CONCENTRATION IS A SPECIFIED CONSTANT OR ',            A2400...
     9   'FUNCTION OF TIME'//11X,I6,5X,'EXACT NUMBER OF NODES AT',      A2410...
     *   ' WHICH FLUID INFLOW OR OUTFLOW IS A SPECIFIED CONSTANT',      A2420...
     A   ' OR FUNCTION OF TIME'/11X,I6,5X,'EXACT NUMBER OF NODES AT',   A2430...
     B   ' WHICH A SOURCE OR SINK OF SOLUTE MASS IS A SPECIFIED ',      A2440...
     C   'CONSTANT OR FUNCTION OF TIME'//11X,I6,5X,'EXACT NUMBER OF ',  A2450...
     D   'NODES AT WHICH PRESSURE AND CONCENTRATION WILL BE OBSERVED',  A2460...
     E   /11X,I6,5X,'MAXIMUM NUMBER OF TIME STEPS ON WHICH ',           A2470...
     F   'OBSERVATIONS WILL BE MADE')                                   A2480...
C                                                                       A2490...
      IF(ME.EQ.+1)                                                      A2500...
     1     WRITE(K3,255) NN,NE,NPBC,NUBC,NSOP,NSOU,NOBS,NTOBS           A2510NEW
  255 FORMAT(////11X,'S I M U L A T I O N   C O N T R O L   ',          A2520...
     1   'N U M B E R S'//11X,I6,5X,'NUMBER OF NODES IN FINITE-',       A2530...
     2   'ELEMENT MESH'/11X,I6,5X,'NUMBER OF ELEMENTS IN MESH'/         A2540...
     5   11X,I6,5X,'EXACT NUMBER OF NODES IN MESH AT WHICH ',           A2570...
     6   'PRESSURE IS A SPECIFIED CONSTANT OR FUNCTION OF TIME'/        A2580...
     7   11X,I6,5X,'EXACT NUMBER OF NODES IN MESH AT WHICH ',           A2590...
     8   'TEMPERATURE IS A SPECIFIED CONSTANT OR ',                     A2600...
     9   'FUNCTION OF TIME'//11X,I6,5X,'EXACT NUMBER OF NODES AT',      A2610...
     *   ' WHICH FLUID INFLOW OR OUTFLOW IS A SPECIFIED CONSTANT',      A2620...
     A   ' OR FUNCTION OF TIME'/11X,I6,5X,'EXACT NUMBER OF NODES AT',   A2630...
     B   ' WHICH A SOURCE OR SINK OF ENERGY IS A SPECIFIED CONSTANT',   A2640...
     C   ' OR FUNCTION OF TIME'//11X,I6,5X,'EXACT NUMBER OF NODES ',    A2650...
     D   'AT WHICH PRESSURE AND TEMPERATURE WILL BE OBSERVED'           A2660...
     E   /11X,I6,5X,'MAXIMUM NUMBER OF TIME STEPS ON WHICH ',           A2670...
     F   'OBSERVATIONS WILL BE MADE')                                   A2680...
C                                                                       A2690...
C                                                                       A2700...
C.....CALCULATE DIMENSIONS FOR POINTERS                                 A2710...
C                                                                       A2720...
      NBCN=NPBC+NUBC+1                                                  A2730...
      NSOP=NSOP+1                                                       A2740...
      NSOU=NSOU+1                                                       A2750...
      NPINCH=1                                                          A2760NEW
      NBIP=5                                                            NEW
      NBIS=9                                                            NEW
      MATDMP=NN*NBIP                                                    A2770NEW
      MATDMS=NN*NBIS                                                    A2770NEW
      NIN=NE*8                                                          A2780...
      NOBSN=NOBS+1                                                      A2790...
      NTOBSN=NTOBS+2                                                    A2800...
      MATOBS=NOBSN*NTOBSN                                               A2810...
      NE4=NE*4                                                          A2820...
C                                                                       A2830...
C                                                                       A2840...
C.....SET UP POINTERS FOR REAL MATRICES                                 A2850...
C                                                                       A2860...
```

```
      KRM1=1                                                           A2870...
      KRM2=KRM1+    MATDMP                                             A2880NEW
      KRM3=KRM2+    MATDMS                                             A2890NEW
      KRM4=KRM3+NN                                                     NEW
      KRM5=KRM4+NN                                                     NEW
      KRM6=KRM5+NN                                                     NEW
      KRM7=KRM6+NN                                                     NEW
C     KRM8=KRM7+NN*9                                                   NEW
C     NOTE: THE LAST POINTER IN THE ABOVE LIST, CURRENTLY, KRM8,       A2900...
C           MAY  N E V E R  BE PASSED TO SUTRA.  IT POINTS TO THE      A2910...
C           STARTING ELEMENT OF THE NEXT NEW MATRIX TO BE ADDED.       A2920...
C           PRESENTLY, SPACE IS ALLOCATED FOR (7) MATRICES.            A2930...
C                                                                      A2940...
C                                                                      A2950...
C.....SET UP POINTERS FOR REAL VECTORS                                 A2960...
C                                                                      A2970...
C     NNV IS NUMBER OF REAL VECTORS THAT ARE NN LONG                   A2980...
      NNV=30                                                           A2990...
C     NEV IS NUMBER OF REAL VECTORS THAT ARE NE LONG                   A3000...
      NEV=10                                                           A3010...
C                                                                      A3020...
      M2=1                                                             A3030...
      KRV(1)=1                                                         A3040...
      M1=M2+1                                                          A3050...
      M2=M2+        ( NNV )                                            A3060...
      DO 400 J=M1,M2                                                   A3070...
  400 KRV(J)=KRV(J-1)+  NN                                             A3080...
      M1=M2+1                                                          A3090...
      M2=M2+        ( NEV )                                            A3100...
      DO 410 J=M1,M2                                                   A3110...
  410 KRV(J)=KRV(J-1)+  NE                                             A3120...
      M1=M2+1                                                          A3130...
      M2=M2+        ( 3 )                                              A3140...
      DO 420 J=M1,M2                                                   A3150...
  420 KRV(J)=KRV(J-1)+  NBCN                                           A3160...
      M1=M2+1                                                          A3170...
      M2=M2+        ( 2 )                                              A3180...
      DO 430 J=M1,M2                                                   A3190...
  430 KRV(J)=KRV(J-1)+  MATOBS                                         A3200...
      M2=M2+        ( 1 )                                              A3210...
      KRV(M2)=KRV(M2-1)+NTOBSN                                         A3220...
      M1=M2+1                                                          A3230...
      M2=M2+        ( 2 )                                              A3240...
      DO 440 J=M1,M2                                                   A3250...
  440 KRV(J)=KRV(J-1)+  NE4                                            A3260...
C     NOTE: THE LAST POINTER IN THE ABOVE LIST, CURRENTLY, KRV(J=49),  A3270...
C           MAY  N E V E R  BE PASSED TO SUTRA.  IT POINTS TO THE      A3280...
C           STARTING ELEMENT OF THE NEXT NEW REAL VECTOR TO BE ADDED.  A3290...
C           PRESENTLY, SPACE IS ALLOCATED FOR (48) VECTORS.            A3300...
C                                                                      A3310...
C                                                                      A3320...
C.....SET UP POINTERS FOR INTEGER VECTORS                              A3330...
C                                                                      A3340...
      KIMV1=1                                                          A3350...
      KIMV2=KIMV1+    NIN                                              A3360...
      KIMV3=KIMV2+    NPINCH*3                                         A3370...
      KIMV4=KIMV3+    NSOP                                             A3380...
      KIMV5=KIMV4+    NSOU                                             A3390...
      KIMV6=KIMV5+    NBCN                                             A3400...
      KIMV7=KIMV6+    NBCN                                             A3410...
      KIMV8=KIMV7+    NN                                               A3420...
      KIMV9=KIMV8+    NOBSN                                            A3430...
```

```
C     KIMV10=KIMV9+    NTOBSN                                           A3440...
C     NOTE: THE LAST POINTER IN THE ABOVE LIST, CURRENTLY, KIMV10,      A3450...
C           MAY  N E V E R  BE PASSED TO SUTRA.  IT POINTS TO THE       A3460...
C           STARTING ELEMENT OF THE NEXT NEW INTEGER VECTOR TO BE ADDED.A3470...
C           PRESENTLY, SPACE IS ALLOCATED FOR (8) INTEGER VECTORS.      A3480...
C                                                                       A3490...
C                                                                       NEW
C                 CHECK FOR CORRECT DIMENSIONS                          NEW
        RMDIM = NN*(NBIP+NBIS+9)                                        NEW
        RVDIM = (( NNV*NN + (NEV+8)*NE + NBCN*3                         NEW
     1              + (NOBS+1)*(NTOBS+2)*2 + NTOBS + 5 ))               NEW
        IMVDIM = (( NE*8 + NN + NPINCH*3 + NSOP + NSOU                  NEW
     1              + NBCN*2 + NOBS + NTOBS + 12 ))                     NEW
      IF(RMDIM.GT.RMDIMA.OR.RVDIM.GT.RVDIMA.OR.IMVDIM.GT.IMVDMA) THEN   NEW
        WRITE(*,*) 'MAXIMUM DIMENSIONS EXCEEDED, PLEASE CORRECT'        NEW
        STOP 101                                                        NEW
      END IF                                                            NEW
C                                                                       A3500...
C.....PASS POINTERS TO MAIN CONTROL ROUTINE, SUTRA                      A3510...
      CALL SUTRA(RM(KRM1),RM(KRM2),RM(KRM3),RM(KRM4),RM(KRM5),
     1   RM(KRM6),RM(KRM7),                                             A3520NEW
     1   RV(KRV(1)),RV(KRV(2)),RV(KRV(3)),RV(KRV(4)),RV(KRV(5)),        A3530...
     2   RV(KRV(6)),RV(KRV(7)),RV(KRV(8)),RV(KRV(9)),RV(KRV(10)),       A3540...
     3   RV(KRV(11)),RV(KRV(12)),RV(KRV(13)),RV(KRV(14)),RV(KRV(15)),   A3550...
     4   RV(KRV(16)),RV(KRV(17)),RV(KRV(18)),RV(KRV(19)),RV(KRV(20)),   A3560...
     5   RV(KRV(21)),RV(KRV(22)),RV(KRV(23)),RV(KRV(24)),RV(KRV(25)),   A3570...
     6   RV(KRV(26)),RV(KRV(27)),RV(KRV(28)),RV(KRV(29)),RV(KRV(30)),   A3580...
     7   RV(KRV(31)),RV(KRV(32)),RV(KRV(33)),RV(KRV(34)),RV(KRV(35)),   A3590...
     8   RV(KRV(36)),RV(KRV(37)),RV(KRV(38)),RV(KRV(39)),RV(KRV(40)),   A3600...
     9   RV(KRV(41)),RV(KRV(42)),RV(KRV(43)),RV(KRV(44)),RV(KRV(45)),   A3610...
     *   RV(KRV(46)),RV(KRV(47)),RV(KRV(48)),                           A3620...
     1   IMV(KIMV1),IMV(KIMV2),IMV(KIMV3),IMV(KIMV4),IMV(KIMV5),        A3630...
     2   IMV(KIMV6),IMV(KIMV7),IMV(KIMV8),IMV(KIMV9) )                  A3640...
C                                                                       A3650...
C                                                                       A3660...
      ENDFILE(K3)                                                       A3670...
      STOP                                                              A3680...
      END                                                               A3690...
C     SUBROUTINE       S  U  T  R  A          SUTRA - VERSION 1284-2D B10.....
C                                                                       B20.....
C *** PURPOSE :                                                         B30.....
C ***  MAIN CONTROL ROUTINE FOR SUTRA SIMULATION.                       B40.....
C ***  ORGANIZES DATA INPUT, INITIALIZATION, CALCULATIONS FOR           B50.....
C ***  EACH TIME STEP AND ITERATION, AND VARIOUS OUTPUTS.               B60.....
C ***  CALLS MOST OTHER SUBROUTINES.                                    B70.....
C                                                                       B80.....
      SUBROUTINE SUTRA( PMAT,UMAT,CWRK,CWRK2,CWRK3,CWRK4,CWRK5,         B90NEW
     1   PITER,UITER,PM1,UM1,UM2,PVEL,SL,SR,                            B100....
     2   X,Y,THICK,VOL,POR,CS1,CS2,CS3,SW,DSWDP,RHO,SOP,                B110....
     3   QIN,UIN,QUIN,PVEC,UVEC,RCIT,RCITM1,CC,XX,YY,                   B120....
     4   ALMAX,ALMIN,ATAVG,VMAG,VANG,                                   B130....
     5   PERMXX,PERMXY,PERMYX,PERMYY,PANGLE,                            B140....
     6   PBC,UBC,QPLITR,POBS,UOBS,OBSTIM,GXSI,GETA,                     B150....
     7   IN,IPINCH,IQSOP,IQSOU,IPBC,IUBC,INDEX,IOBS,ITOBS )             B160....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               B170....
      CHARACTER*10 ADSMOD                                               B180....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6                           MODIFIED
      COMMON/MODSOR/ ADSMOD                                             B190....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          B200NEW
     1   NSOP,NSOU,NBCN                                                 B210
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        B220....
```

```
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                         B230....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,          B240....
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME             B250NEW
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,B260....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2        B270....
      COMMON/ITERAT/ RPM,RPMAX,RUM,RUMAX,ITER,ITRMAX,IPWORS,IUWORS,      B280....
     1               ICON,ITRMX2,OMEGA,RPMX2,RUMX2                       NEW
      COMMON/KPRINT/ KNODAL,KELMNT,KINCID,KPLOTP,KPLOTU,KVEL,KBUDG       B290....
      COMMON/OBS/ NOBSN,NTOBSN,NOBCYC,ITCNT                              B300....
      DIMENSION QIN(NN),UIN(NN),IQSOP(NSOP),QUIN(NN),IQSOU(NSOU)         B310....
      DIMENSION IPBC(NBCN),PBC(NBCN),IUBC(NBCN),UBC(NBCN),QPLITR(NBCN)   B320....
      DIMENSION IN(NIN),IPINCH(1,3)                                      B330NEW
      DIMENSION X(NN),Y(NN),THICK(NN),SW(NN),DSWDP(NN),RHO(NN),SOP(NN),  B340....
     1   POR(NN),PVEL(NN)                                                B350....
      DIMENSION PERMXX(NE),PERMXY(NE),PERMYX(NE),PERMYY(NE),PANGLE(NE),  B360....
     1   ALMAX(NE),ALMIN(NE),ATAVG(NE),VMAG(NE),VANG(NE),                B370....
     2   GXSI(NE,4),GETA(NE,4)                                           B380....
      DIMENSION VOL(NN),PMAT(NN,NBIP),PVEC(NN),UMAT(NN,NBIS),UVEC(NN)    B390NEW
      DIMENSION CWRK(NN),CWRK2(NN),CWRK3(NN),CWRK4(NN),CWRK5(NN,5)       AQUI
      DIMENSION PM1(NN),UM1(NN),UM2(NN),PITER(NN),UITER(NN),             B400....
     1   RCIT(NN),RCITM1(NN),CS1(NN),CS2(NN),CS3(NN)                     B410....
      DIMENSION CC(NN),INDEX(NN),XX(NN),YY(NN)                           B420....
      DIMENSION POBS(NOBSN,NTOBSN),UOBS(NOBSN,NTOBSN),OBSTIM(NTOBSN),    B430....
     1   IOBS(NOBSN),ITOBS(NTOBSN)                                       B440....
      DATA IT/0/                                                         B450....
C                                                                        B460....
C                                                                        B470....
C                                                                        B480....
C.....INPUT SIMULATION DATA FROM UNIT-5 (DATASETS 3 THROUGH 15B)          B490....
      CALL INDAT1(X,Y,THICK,POR,ALMAX,ALMIN,ATAVG,PERMXX,PERMXY,         B500....
     1   PERMYX,PERMYY,PANGLE,SOP,IN)                                    B510NEW
C                                                                        B550....
C.....INPUT FLUID MASS, AND ENERGY OR SOLUTE MASS SOURCES                 B560....
C        (DATASETS 17 AND 18)                                            B570....
      CALL ZERO(QIN,NN,0.0D0)                                            B580....
      CALL ZERO(UIN,NN,0.0D0)                                            B590....
      CALL ZERO(QUIN,NN,0.0D0)                                           B600....
      IF(NSOP-1.GT.0.OR.NSOU-1.GT.0)                                     B610....
     1    CALL SOURCE(QIN,UIN,IQSOP,QUIN,IQSOU,IQSOPT,IQSOUT)            B620....
C                                                                        B630....
C.....INPUT SPECIFIED P AND U BOUNDARY CONDITIONS (DATASETS 19 AND 20)    B640....
      IF(NBCN-1.GT.0) CALL BOUND(IPBC,PBC,IUBC,UBC,IPBCT,IUBCT)          B650....
C                                                                        B660....
C.....SET FLAG FOR TIME-DEPENDENT SOURCES OR BOUNDARY CONDITIONS.         B670....
C     WHEN IBCT=+4, THERE ARE NO TIME-DEPENDENT SPECIFICATIONS.          B680....
      IBCT=IQSOPT+IQSOUT+IPBCT+IUBCT                                     B690....
C                                                                        B700....
C.....INPUT OBSERVATION NODE DATA (DATASET 21)                            B710....
      IF(NOBSN-1.GT.0) CALL OBSERV(0,IOBS,ITOBS,POBS,UOBS,OBSTIM,        B720....
     1   PVEC,UVEC,ISTOP)                                                B730....
      WRITE(K3,4000)                                                     NEW
 4000 FORMAT(////////1X,132(1H-)///42X,'E N D   O F   I N P U T   ',     NEW
     1   'F R O M   U N I T - 5'//132(1H-))                              NEW
C                                                                        B830....
C.....INPUT INITIAL OR RESTART CONDITIONS AND INITIALIZE PARAMETERS       B840....
C        (READ UNIT-55 DATA)                                             B850....
      CALL INDAT2(PVEC,UVEC,PM1,UM1,UM2,CS1,CS2,CS3,SL,SR,RCIT,SW,DSWDP,B860....
     1   PBC,IPBC,IPBCT)                                                 B870....
C                                                                        B880....
C.....SET STARTING TIME OF SIMULATION CLOCK                               B890....
C     TSEC=TSTART                                                        B900....
      TSECP0=TSEC                                                        B910....
```

```
      TSECU0=TSEC                                                       B920....
      TMIN=TSEC/60.D0                                                   B930....
      THOUR=TMIN/60.D0                                                  B940....
      TDAY=THOUR/24.D0                                                  B950....
      TWEEK=TDAY/7.D0                                                   B960....
      TMONTH=TDAY/30.4375D0                                             B970....
      TYEAR=TDAY/365.25D0                                               B980....
      DELT0=DELT                                                        NEW
C                                                                       B990....
C.....OUTPUT INITIAL CONDITIONS OR STARTING CONDITIONS                  B1000...
      IF(ISSTRA.NE.1) CALL PRISOL(0,0,0,PVEC,UVEC,VMAG,VANG,SW)         B1010...
C                                                                       B1020...
C.....SET SWITCHES AND PARAMETERS FOR SOLUTION WITH STEADY-STATE FLOW  B1030...
      IF(ISSFLO.NE.1) GOTO 1000                                         B1040...
      ML=1                                                              B1050...
      NOUMAT=0                                                          B1060...
      ISSFLO=2                                                          B1070...
      ITER=0                                                            B1080...
      DLTPM1=DELTP                                                      B1090...
      DLTUM1=DELTU                                                      B1100...
      BDELP=0.0D0                                                       B1110...
      BDELU=0.0D0                                                       B1120...
      GOTO 1100                                                         B1130...
C                                                                       B1140...
C                                                                       B1150...
C **********************************************************************B1160...
C.....BEGIN TIME STEP **************************************************B1170...
C **********************************************************************B1180...
 1000 IT=IT+1                                                           B1190...
      ITER=0                                                            B1200...
      ML=0                                                              B1210...
      NOUMAT=0                                                          B1220...
C.....SET NOUMAT TO OBTAIN U SOLUTION WITHOUT REFORMULATING THE MATRIX  B1230NEW
C        BEGINNING ON SECOND TIME STEP AFTER A PRESSURE SOLUTION        B1240...
C        IF THE SOLUTION IS NON-ITERATIVE (ITRMAX=1)                    B1250...
C     IF(MOD(IT-1,NPCYC).NE.0.AND.MOD(IT,NPCYC).NE.0.AND.IT.GT.2        B1260...
C    1    .AND.ITRMAX.EQ.1) NOUMAT=1                                    B1270...
C.....CHOOSE SOLUTION VARIABLE ON THIS TIME STEP:                       B1280...
C        ML=0 FOR P AND U, ML=1 FOR P ONLY, AND ML=2 FOR U ONLY.        B1290...
      IF(IT.EQ.1.AND.ISSFLO.NE.2) GOTO 1005                             B1300...
      IF(MOD(IT,NPCYC).NE.0) ML=2                                       B1310...
      IF(MOD(IT,NUCYC).NE.0) ML=1                                       B1320...
C.....MULTIPLY TIME STEP SIZE BY DTMULT EACH ITCYC TIME STEPS           B1330...
C.....THE FOLLOWING CARDS WERE ADDED TO ALLOW FOR THE TIME STEP         NEW
C.....TO YIELD A CONSTANT DISTANCE INCREMENT IN A RADIAL FLOW SYSTEM    NEW
      IF(ITIME.EQ.0) THEN                                               NEW
        IF(MOD(IT,ITCYC).EQ.0.AND.IT.GT.1) DELT=DELT*DTMULT             B1340...
      END IF                                                            NEW
      IF(ITIME.EQ.1 .AND. IT.GT.1) THEN                                 NEW
C       DELT0 = THE INITIAL TIME INCREMENT                              NEW
C       ITCYC = A FLAG -- FOR ITCYC>0, TIME STEP SIZE IS INCREASED      NEW
C       FOR ITCYC<0, TIME STEP SIZE IS DECREASED, WHERE -ITCYC =        NEW
C                    MAXIMUM NUMBER OF TIME STEPS IN PREVIOUS RUN       NEW
C       FOR ITCYC=0, TIME STEP SIZE IS HELD CONSTANT                    NEW
        IF(ITCYC.GT.0) DELT=DELT0*(2.0*IT-1)                            NEW
        IF(ITCYC.LT.0) DELT=DELT0*(2.*(-ITCYC-IT+1)-1)                  NEW
      END IF                                                            NEW
C.....SET TIME STEP SIZE TO MAXIMUM ALLOWED SIZE, DTMAX                 B1350...
      IF(DELT.GT.DTMAX) DELT=DTMAX                                      B1360...
C.....INCREMENT SIMULATION CLOCK, TSEC, TO END OF NEW TIME STEP         B1370...
 1005 TSEC=TSEC+DELT                                                    B1380...
```

```
      TMIN=TSEC/60.D0                                                   B1390...
      THOUR=TMIN/60.D0                                                  B1400...
      TDAY=THOUR/24.D0                                                  B1410...
      TWEEK=TDAY/7.D0                                                   B1420...
      TMONTH=TDAY/30.4375D0                                             B1430...
      TYEAR=TDAY/365.25D0                                               B1440...
C                                                                       B1450...
C.....SET TIME STEP FOR P AND/OR U, WHICHEVER ARE SOLVED FOR            B1460...
C        ON THIS TIME STEP                                              B1470...
      IF(ML-1) 1010,1020,1030                                           B1480...
 1010 DLTUM1=DELTU                                                      B1490...
      DLTPM1=DELTP                                                      B1500...
      GOTO 1040                                                         B1510...
 1020 DLTPM1=DELTP                                                      B1520...
      GOTO 1040                                                         B1530...
 1030 DLTUM1=DELTU                                                      B1540...
 1040 CONTINUE                                                          B1550...
      DELTP=TSEC-TSECP0                                                 B1560...
      DELTU=TSEC-TSECU0                                                 B1570...
C.....SET PROJECTION FACTORS USED ON FIRST ITERATION TO EXTRAPOLATE     B1580...
C        AHEAD ONE-HALF TIME STEP                                       B1590...
      BDELP=(DELTP/DLTPM1)*0.50D0                                       B1600...
      BDELU=(DELTU/DLTUM1)*0.50D0                                       B1610...
      BDELP1=BDELP+1.0D0                                                B1620...
      BDELU1=BDELU+1.0D0                                                B1630...
C.....INCREMENT CLOCK FOR WHICHEVER OF P AND U WILL BE SOLVED FOR        B1640...
C        ON THIS TIME STEP                                              B1650...
      IF(ML-1) 1060,1070,1080                                           B1660...
 1060 TSECP0=TSEC                                                       B1670...
      TSECU0=TSEC                                                       B1680...
      GOTO 1090                                                         B1690...
 1070 TSECP0=TSEC                                                       B1700...
      GOTO 1090                                                         B1710...
 1080 TSECU0=TSEC                                                       B1720...
 1090 CONTINUE                                                          B1730...
C                                                                       B1740...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - B1750...
C.....BEGIN ITERATION - - - - - - - - - - - - - - - - - - - - - - - - - B1760...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - B1770...
 1100 ITER=ITER+1                                                       B1780...
C                                                                       B1790...
      IF(ML-1) 2000,2200,2400                                           B1800...
C.....SHIFT AND SET VECTORS FOR TIME STEP WITH BOTH P AND U SOLUTIONS    B1810...
 2000 DO 2025 I=1,NN                                                    B1820...
      PITER(I)=PVEC(I)                                                  B1830...
      PVEL(I)=PVEC(I)                                                   B1840...
      UITER(I)=UVEC(I)                                                  B1850...
      RCITM1(I)=RCIT(I)                                                 B1860...
 2025 RCIT(I)=RHOW0+DRWDU*(UITER(I)-URHOW0)                             B1870...
      DO 2050 IP=1,NPBC                                                 B1880...
      I=IABS(IPBC(IP))                                                  B1890...
      QPLITR(IP)=GNU*(PBC(IP)-PITER(I))                                 B1900...
 2050 CONTINUE                                                          B1910...
      IF(ITER.GT.1) GOTO 2600                                           B1920...
      DO 2075 I=1,NN                                                    B1930...
      PITER(I)=BDELP1*PVEC(I)-BDELP*PM1(I)                              B1940...
      UITER(I)=BDELU1*UVEC(I)-BDELU*UM1(I)                              B1950...
      PM1(I)=PVEC(I)                                                    B1960...
      UM2(I)=UM1(I)                                                     B1970...
 2075 UM1(I)=UVEC(I)                                                    B1980...
      GOTO 2600                                                         B1990...
C.....SHIFT AND SET VECTORS FOR TIME STEP WITH P SOLUTION ONLY          B2000...
 2200 DO 2225 I=1,NN                                                    B2010...
```

```
      PVEL(I)=PVEC(I)                                                   B2020...
 2225 PITER(I)=PVEC(I)                                                  B2030...
      IF(ITER.GT.1) GOTO 2600                                           B2040...
      DO 2250 I=1,NN                                                    B2050...
      PITER(I)=BDELP1*PVEC(I)-BDELP*PM1(I)                              B2060...
      UITER(I)=UVEC(I)                                                  B2070...
      RCITM1(I)=RCIT(I)                                                 B2080...
      RCIT(I)=RHOW0+DRWDU*(UITER(I)-URHOW0)                             B2090...
 2250 PM1(I)=PVEC(I)                                                    B2100...
      GOTO 2600                                                         B2110...
C.....SHIFT AND SET VECTORS FOR TIME STEP WITH U SOLUTION ONLY          B2120...
 2400 IF(NOUMAT.EQ.1) GOTO 2480                                         B2130...
      DO 2425 I=1,NN                                                    B2140...
 2425 UITER(I)=UVEC(I)                                                  B2150...
      IF(ITER.GT.1) GOTO 2600                                           B2160...
      DO 2450 I=1,NN                                                    B2170...
      PITER(I)=PVEC(I)                                                  B2180...
      PVEL(I)=PVEC(I)                                                   B2190...
      UITER(I)=BDELU1*UVEC(I)-BDELU*UM1(I)                              B2200...
 2450 RCITM1(I)=RCIT(I)                                                 B2210...
      DO 2475 IP=1,NPBC                                                 B2220...
      I=IABS(IPBC(IP))                                                  B2230...
      QPLITR(IP)=GNU*(PBC(IP)-PITER(I))                                 B2240...
 2475 CONTINUE                                                          B2250...
 2480 DO 2500 I=1,NN                                                    B2260...
      UM2(I)=UM1(I)                                                     B2270...
 2500 UM1(I)=UVEC(I)                                                    B2280...
 2600 CONTINUE                                                          B2290...
C                                                                       B2300...
C.....INITIALIZE ARRAYS WITH VALUE OF ZERO                              B2310...
      MATDMP=NN*NBIP                                                    B2320...
      MATDMS=NN*NBIS                                                    B2320...
      IF(ML-1) 3000,3000,3300                                           B2330...
 3000 CALL ZERO(PMAT,MATDMP,0.0D0)                                      B2340...
      CALL ZERO(PVEC,NN,0.0D0)                                          B2350...
      CALL ZERO(VOL,NN,0.0D0)                                           B2360...
      IF(ML-1) 3300,3400,3300                                           B2370...
 3300 IF(NOUMAT) 3350,3350,3375                                         B2380...
 3350 CALL ZERO(UMAT,MATDMS,0.0D0)                                      B2390...
 3375 CALL ZERO(UVEC,NN,0.0D0)                                          B2400...
 3400 CONTINUE                                                          B2410...
C                                                                       B2420...
C.....SET TIME-DEPENDENT BOUNDARY CONDITIONS, SOURCES AND SINKS         B2430...
C        FOR THIS TIME STEP                                             B2440...
      IF(ITER.EQ.1.AND.IBCT.NE.4)                                       B2450...
     1   CALL BCTIME(IPBC,PBC,IUBC,UBC,QIN,UIN,QUIN,IQSOP,IQSOU,        B2460...
     2   IPBCT,IUBCT,IQSOPT,IQSOUT,UM1)                                 B2470NEW
C                                                                       B2480...
C.....SET SORPTION PARAMETERS FOR THIS TIME STEP                        B2490...
      IF(ML.NE.1.AND.ME.EQ.-1.AND.NOUMAT.EQ.0.AND.                      B2500...
     1   ADSMOD.NE.'NONE      ') CALL ADSORB(CS1,CS2,CS3,SL,SR,UITER)   B2510...
C                                                                       B2520...
C.....DO ELEMENTWISE CALCULATIONS IN MATRIX EQUATION FOR P AND/OR U     B2530...
      IF(NOUMAT.EQ.0)                                                   B2540...
     1 CALL ELEMEN(ML,IN,X,Y,THICK,PITER,UITER,RCIT,RCITM1,POR,         B2550...
     2   ALMAX,ALMIN,ATAVG,PERMXX,PERMXY,PERMYX,PERMYY,PANGLE,          B2560...
     3   VMAG,VANG,VOL,PMAT,PVEC,UMAT,UVEC,GXSI,GETA,PVEL,CWRK)         B2570NEW
C                                                                       B2580...
C.....DO NODEWISE CALCULATIONS IN MATRIX EQUATION FOR P AND/OR U        B2590...
      CALL NODALB(ML,VOL,PMAT,PVEC,UMAT,UVEC,PITER,UITER,PM1,UM1,UM2,   B2600...
     1   POR,QIN,UIN,QUIN,CS1,CS2,CS3,SL,SR,SW,DSWDP,RHO,SOP)           B2610...
C.....SET SPECIFIED P AND U CONDITIONS IN MATRIX EQUATION FOR P AND/OR UB2630...
```

```
      CALL BCB(ML,PMAT,PVEC,UMAT,UVEC,IPBC,PBC,IUBC,UBC,QPLITR)         B2640...
 4200 CONTINUE                                                          B2690...
C                                                                       B2700...
C.....MATRIX EQUATION FOR P AND/OR U ARE COMPLETE, SOLVE EQUATIONS:     B2710...
      IF(ML-1) 5000,5000,5500                                           B2750...
C.....SOLVE FOR P                                                       B2760...
 5000 IPS=0                                                             B2770NEW
      CALL SOLVEC(NBIP,PMAT,PM1,PVEC,CWRK,CWRK2,CWRK3,CWRK4,CWRK5)      NEW
C.....P SOLUTION NOW IN PVEC                                            B2790...
      IF(ML-1) 5500,6000,5500                                           B2800...
C.....SOLVE FOR U                                                       B2810...
 5500 IPS=1                                                             B2820NEW
 5700  CALL LSORA(NBIS,UMAT,UVEC,UITER,CWRK,CWRK2,CWRK5)                NEW
C.....U SOLUTION NOW IN UVEC                                            B2860...
 6000 CONTINUE                                                          B2870...
C                                                                       B2880...
C.....CHECK PROGRESS AND CONVERGENCE OF ITERATIONS                      B2890...
C           AND SET STOP AND GO FLAGS:                                  B2900...
C                ISTOP = -1   NOT CONVERGED - STOP SIMULATION           B2910...
C                ISTOP =  0   ITERATIONS LEFT OR CONVERGED - KEEP SIMULATING B2920...
C                ISTOP =  1   LAST TIME STEP REACHED - STOP SIMULATION   B2930...
C                ISTOP =  2   MAXIMUM TIME REACHED - STOP SIMULATION    B2940...
C                IGOI = 0   P AND U CONVERGED, OR NO ITERATIONS DONE    B2950...
C                IGOI = 1   ONLY P HAS NOT YET CONVERGED TO CRITERION   B2960...
C                IGOI = 2   ONLY U HAS NOT YET CONVERGED TO CRITERION   B2970...
C                IGOI = 3   BOTH P AND U HAVE NOT YET CONVERGED TO CRITERIA B2980...
      ISTOP=0                                                           B2990...
      IGOI=0                                                            B3000...
      IF(ITRMAX-1) 7500,7500,7000                                       B3010...
 7000 RPM=0.D0                                                          B3020...
      RUM=0.D0                                                          B3030...
      IPWORS=0                                                          B3040...
      IUWORS=0                                                          B3050...
      IF(ML-1) 7050,7050,7150                                           B3060...
 7050 DO 7100 I=1,NN                                                    B3070...
      RP=DABS(PVEC(I)-PITER(I))                                         B3080...
      IF(RP-RPM) 7100,7060,7060                                         B3090...
 7060 RPM=RP                                                            B3100...
      IPWORS=I                                                          B3110...
 7100 CONTINUE                                                          B3120...
      IF(RPM.GT.RPMAX) IGOI=IGOI+1                                      B3130...
 7150 IF(ML-1) 7200,7350,7200                                           B3140...
 7200 DO 7300 I=1,NN                                                    B3150...
      RU=DABS(UVEC(I)-UITER(I))                                         B3160...
      IF(RU-RUM) 7300,7260,7260                                         B3170...
 7260 RUM=RU                                                            B3180...
      IUWORS=I                                                          B3190...
 7300 CONTINUE                                                          B3200...
      IF(RUM.GT.RUMAX) IGOI=IGOI+2                                      B3210...
 7350 CONTINUE                                                          B3220...
      IF(IGOI.GT.0.AND.ITER.EQ.ITRMAX) ISTOP=-1                         B3230...
      IF(IGOI.GT.0.AND.ISTOP.EQ.0) GOTO 1100                            B3240...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - B3250...
C.....END ITERATION - - - - - - - - - - - - - - - - - - - - - - - - - - B3260...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - B3270...
C                                                                       B3280...
 7500 CONTINUE                                                          B3290...
      IF(ISTOP.NE.-1.AND.IT.EQ.ITMAX) ISTOP=1                           B3300...
      IF(ISTOP.NE.-1.AND.TSEC.GE.TMAX) ISTOP=2                          B3310...
C                                                                       B3320...
C.....OUTPUT RESULTS FOR TIME STEP EACH NPRINT TIME STEPS               B3330...
      IF(IT.GT.1.AND.MOD(IT,NPRINT).NE.0.AND.ISTOP.EQ.0) GOTO 8000      B3340...
```

```
C.....PRINT P AND/OR U, AND MAYBE SW AND/OR V                           B3350...
      CALL PRISOL(ML,ISTOP,IGOI,PVEC,UVEC,VMAG,VANG,SW)                 B3360...
C.....CALCULATE AND PRINT FLUID MASS AND/OR ENERGY OR SOLUTE MASS BUDGETB3370...
      IF(KBUDG.EQ.1)                                                    B3380...
     1   CALL BUDGET(ML,IBCT,VOL,SW,DSWDP,RHO,SOP,QIN,PVEC,PM1,         B3390...
     2   PBC,QPLITR,IPBC,IQSOP,POR,UVEC,UM1,UM2,UIN,QUTN,TQSOU,UBC,     B3400...
     3   CS1,CS2,CS3,SL,SR)                                             B3410...
 8000 CONTINUE                                                          B3500...
C                                                                       B3510...
C.....MAKE OBSERVATIONS AT OBSERVATION NODES EACH NOBCYC TIME STEPS     B3520...
      IF(NOBSN-1.GT.0) CALL OBSERV(1,IOBS,ITOBS,POBS,UOBS,OBSTIM,       B3530...
     1                   PVEC,UVEC,ISTOP)                               B3540...
C                                                                       B3550...
C.....STORE RESULTS FOR POSSIBLE RESTART OF SIMULATION EACH TIME STEP   B3560...
      IF(ISTORE.NE.1) GOTO 8150                                         B3570...
      CALL STORE(PVEC,UVEC,PM1,UM1,CS1,RCIT,SW,PBC)                     B3580...
C                                                                       B3590...
 8150 IF(ISTOP.EQ.0) GOTO 1000                                          B3600...
C **********************************************************************B3610...
C.....END TIME STEP ****************************************************B3620...
C **********************************************************************B3630...
C                                                                       B3640...
C                                                                       B3650...
C.....COMPLETE OUTPUT AND TERMINATE SIMULATION                          B3660...
      IF(ISTORE.EQ.1) WRITE(K3,8100)                                    B3670...
 8100 FORMAT(//////11X,'*** LAST SOLUTION HAS BEEN STORED ',            B3680...
     1   'ON UNIT 66 ***')                                              B3690...
C                                                                       B3700...
C.....OUTPUT RESULTS OF OBSERVATIONS                                    B3710...
 8200 IF(NOBSN-1.GT.0) CALL OBSERV(2,IOBS,ITOBS,POBS,UOBS,OBSTIM,       B3720...
     1                       PVEC,UVEC,ISTOP)                           B3730...
C                                                                       B3740...
C.....OUTPUT END OF SIMULATION MESSAGE AND RETURN TO MAIN FOR STOP      B3750...
      IF(ISTOP.GT.0) GOTO 8400                                          B3760...
      IF(IGOI-2) 8230,8260,8290                                         B3770...
 8230 WRITE(K3,8235)                                                    B3780...
 8235 FORMAT(////////11X,'SIMULATION TERMINATED DUE TO ',               B3790...
     1   'NON-CONVERGENT PRESSURE',                                     B3800...
     2              /11X,'********** ********** *** ** ',               B3810...
     3   '************** ********')                                     B3820...
      RETURN                                                            B3830...
 8260 IF(ME) 8262,8262,8266                                             B3840...
 8262 WRITE(K3,8264)                                                    B3850...
 8264 FORMAT(////////11X,'SIMULATION TERMINATED DUE TO ',               B3860...
     1   'NON-CONVERGENT CONCENTRATION',                                B3870...
     2              /11X,'********** ********** *** ** ',               B3880...
     3   '************** *************')                                B3890...
      RETURN                                                            B3900...
 8266 WRITE(K3,8268)                                                    B3910...
 8268 FORMAT(////////11X,'SIMULATION TERMINATED DUE TO ',               B3920...
     1   'NON-CONVERGENT TEMPERATURE',                                  B3930...
     2              /11X,'********** ********** *** ** ',               B3940...
     3   '************** ***********')                                  B3950...
      RETURN                                                            B3960...
 8290 IF(ME) 8292,8292,8296                                             B3970...
 8292 WRITE(K3,8294)                                                    B3980...
 8294 FORMAT(////////11X,'SIMULATION TERMINATED DUE TO ',               B3990...
     1   'NON-CONVERGENT PRESSURE AND CONCENTRATION',                   B4000...
     2              /11X,'********** ********** *** ** ',               B4010...
     3   '************** ******** *** *************')                   B4020...
      RETURN                                                            B4030...
 8296 WRITE(K3,8298)                                                    B4040...
```

```
 8298 FORMAT(////////11X,'SIMULATION TERMINATED DUE TO ',              B4050...
     1   'NON-CONVERGENT PRESSURE AND TEMPERATURE',                    B4060...
     2               /11X,'********** ********** *** ** ',             B4070...
     3   '************** ******** *** ***********')                    B4080...
      RETURN                                                           B4090...
C                                                                      B4100...
 8400 IF(ISTOP.EQ.2) GOTO 8500                                         B4110...
      WRITE(K3,8450)                                                   B4120...
 8450 FORMAT(////////11X,'SUTRA SIMULATION TERMINATED AT COMPLETION ', B4130...
     1   'OF TIME STEPS'/                                              B4140...
     2                11X,'***** ********** ********** ** ********** ', B4150...
     3   '** **** *****')                                              B4160...
      RETURN                                                           B4170...
 8500 WRITE(K3,8550)                                                   B4180...
 8550 FORMAT(////////11X,'SUTRA SIMULATION TERMINATED AT COMPLETION ', B4190...
     1   'OF TIME PERIOD'/                                             B4200...
     2                11X,'***** ********** ********** ** ********** ', B4210...
     3   '** **** ******')                                             B4220...
      RETURN                                                           B4230...
C                                                                      B4240...
      END                                                              B4250...
C     SUBROUTINE        I  N  D  A  T  1        SUTRA - VERSION 1284-2D C10.....
C                                                                      C20.....
C *** PURPOSE :                                                        C30.....
C ***  TO INPUT ,OUTPUT, AND ORGANIZE A MAJOR PORTION OF               C40.....
C ***  UNIT-5 INPUT DATA (DATASET 5 THROUGH DATASET 15B)               C50.....
C                                                                      C60.....
      SUBROUTINE INDAT1(X,Y,THICK,POR,ALMAX,ALMIN,ATAVG,PERMXX,PERMXY, C70.....
     1   PERMYX,PERMYY,PANGLE,SOP,IN)                                  C80.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                              C90.....
      CHARACTER*10 ADSMOD                                              C100....
      CHARACTER*14 UTYPE(2)                                            C110....
      CHARACTER*6 STYPE(2)                                             C120....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6                          MODIFIED
      COMMON/MODSOR/ ADSMOD                                            C130....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,         C140NEW
     1   NSOP,NSOU,NBCN                                                C150....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,       C160....
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                       C170....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,       C180....
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME           C190NEW
      COMMON/ITERAT/ RPM,RPMAX,RUM,RUMAX,ITER,ITRMAX,IPWORS,IUWORS,    C200....
     1               ICON,ITRMX2,OMEGA,RPMX2,RUMX2                     NEW
      COMMON/TENSOR/ GRAVX,GRAVY                                       C210....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,C220....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2      C230....
      COMMON/SATPAR/ PCENT,SWRES,PCRES,SSLOPE,SINCPT                   C240....
      COMMON/KPRINT/ KNODAL,KELMNT,KINCID,KPLOTP,KPLOTU,KVEL,KBUDG      C250....
      DIMENSION X(NN),Y(NN),THICK(NN),POR(NN),SOP(NN),IN(NIN)          C260NEW
      DIMENSION PERMXX(NE),PERMXY(NE),PERMYX(NE),PERMYY(NE),PANGLE(NE), C270....
     1   ALMAX(NE),ALMIN(NE),ATAVG(NE)                                 C280....
      DIMENSION IIN(4)                                                 NEW
      DATA UTYPE(1)/' TEMPERATURES '/,UTYPE(2)/'CONCENTRATIONS'/       C290....
      DATA STYPE(1)/'ENERGY'/,STYPE(2)/'SOLUTE'/                       C300....
C                                                                      C310....
      INSTOP=0                                                         C320....
C                                                                      C330....
C.....INPUT DATASET 5: NUMERICAL CONTROL PARAMETERS                    C340....
      READ(K1,50) UP,GNU                                               C350....
   50 FORMAT(G10.0,G15.0)                                              C360....
      WRITE(K3,70) UP,GNU                                              C370....
   70 FORMAT(////11X,'N U M E R I C A L   C O N T R O L   D A T A'//   C380....
```

```
     1   11X,F15.5,5X,'"UPSTREAM WEIGHTING" FACTOR'/                    C390....
     2   11X,1PD15.4,5X,'SPECIFIED PRESSURE BOUNDARY CONDITION FACTOR') C400....
C                                                                       C410....
C.....INPUT DATASET 6: TEMPORAL CONTROL AND SOLUTION CYCLING DATA       C420....
      READ(K1,100) ITMAX,DELT,TMAX,ITCYC,DTMULT,DTMAX,NPCYC,NUCYC       C430....
  100 FORMAT(I5,2G15.0,I10,G10.0,G15.0,2I5)                             C440....
      WRITE(K3,120) ITMAX,DELT,TMAX,ITCYC,DTMULT,DTMAX,NPCYC,NUCYC      C450....
  120 FORMAT(1H1////11X,'T E M P O R A L   C O N T R O L   A N D   ',   C460....
     1   'S O L U T I O N   C Y C L I N G   D A T A',                   C470....
     2   //11X,I15,5X,'MAXIMUM ALLOWED NUMBER OF TIME STEPS'            C480....
     3   /11X,1PD15.4,5X,'INITIAL TIME STEP (IN SECONDS)'               C490....
     4   /11X,1PD15.4,5X,'MAXIMUM ALLOWED SIMULATION TIME (IN SECONDS)' C500....
     5   //11X,I15,5X,'TIME STEP MULTIPLIER CYCLE (IN TIME STEPS)'      C510....
     6   /11X,0PF15.5,5X,'MULTIPLICATION FACTOR FOR TIME STEP CHANGE'   C520....
     7   /11X,1PD15.4,5X,'MAXIMUM ALLOWED TIME STEP (IN SECONDS)'       C530....
     8   //11X,I15,5X,'FLOW SOLUTION CYCLE (IN TIME STEPS)'             C540....
     9   /11X,I15,5X,'TRANSPORT SOLUTION CYCLE (IN TIME STEPS)')        C550....
      IF(NPCYC.GE.1.AND.NUCYC.GE.1) GOTO 140                            C560....
      WRITE(K3,130)                                                     C570....
  130 FORMAT(//11X,'* * * * ERROR DETECTED :  BOTH NPCYC AND ',         C580....
     1   'NUCYC MUST BE SET GREATER THAN OR EQUAL TO 1.')               C590....
      INSTOP=INSTOP-1                                                   C600....
  140 IF(NPCYC.EQ.1.OR.NUCYC.EQ.1) GOTO 160                             C610....
      WRITE(K3,150)                                                     C620....
  150 FORMAT(//11X,'* * * * ERROR DETECTED :  EITHER NPCYC OR ',        C630....
     1   'NUCYC MUST BE SET TO 1.')                                     C640....
      INSTOP=INSTOP-1                                                   C650....
  160 CONTINUE                                                          C660....
C.....SET MAXIMUM ALLOWED TIME STEPS IN SIMULATION FOR                   C670....
C          STEADY-STATE FLOW AND STEADY-STATE TRANSPORT SOLUTION MODES  C680....
      IF(ISSFLO.EQ.1) THEN                                              C690....
       NPCYC=ITMAX+1                                                    C700....
       NUCYC=1                                                          C710....
       ENDIF                                                            C720....
      IF(ISSTRA.EQ.1) ITMAX=1                                           C730....
C                                                                       C740....
C.....INPUT DATASET 7: OUTPUT CONTROLS AND OPTIONS                      C750....
      READ(K1,170) NPRINT,KNODAL,KELMNT,KINCID,KPLOTP,KPLOTU,KVEL,KBUDG C760....
  170 FORMAT(16I5)                                                      C770....
      WRITE(K3,172) NPRINT                                              C780....
  172 FORMAT(////11X,'O U T P U T   C O N T R O L S   A N D   ',        C790....
     1   'O P T I O N S'//11X,I6,5X,'PRINTED OUTPUT CYCLE ',            C800....
     2   '(IN TIME STEPS)')                                             C810....
      IF(KNODAL.EQ.+1) WRITE(K3,174)                                    C820....
      IF(KNODAL.EQ.0) WRITE(K3,175)                                     C830....
  174 FORMAT(/11X,'- PRINT NODE COORDINATES, THICKNESSES AND',          C840....
     1   ' POROSITIES')                                                 C850....
  175 FORMAT(/11X,'- CANCEL PRINT OF NODE COORDINATES, THICKNESSES AND',C860....
     1   ' POROSITIES')                                                 C870....
      IF(KELMNT.EQ.+1) WRITE(K3,176)                                    C880....
      IF(KELMNT.EQ.0) WRITE(K3,177)                                     C890....
  176 FORMAT(11X,'- PRINT ELEMENT PERMEABILITIES AND DISPERSIVITIES')   C900....
  177 FORMAT(11X,'- CANCEL PRINT OF ELEMENT PERMEABILITIES AND ',       C910....
     1   'DISPERSIVITIES')                                              C920....
      IF(KINCID.EQ.+1) WRITE(K3,178)                                    C930....
      IF(KINCID.EQ.0) WRITE(K3,179)                                     C940....
  178 FORMAT(11X,'- PRINT NODE INCIDENCES IN EACH ELEMENT')             C950NEW
  179 FORMAT(11X,'- CANCEL PRINT OF NODE INCIDENCES IN EACH ELEMENT')   C970NEW
      IME=2                                                             C1030...
      IF(ME.EQ.+1) IME=1                                                C1040...
      IF(KVEL.EQ.+1) WRITE(K3,184)                                      C1090...
      IF(KVEL.EQ.0) WRITE(K3,185)                                       C1100...
```

```
  184 FORMAT(/11X,'- CALCULATE AND PRINT VELOCITIES AT ELEMENT ',       C1110...
     1   'CENTROIDS ON EACH TIME STEP WITH OUTPUT')                     C1120...
  185 FORMAT(/11X,'- CANCEL PRINT OF VELOCITIES')                       C1130...
      IF(KBUDG.EQ.+1) WRITE(K3,186) STYPE(IME)                          C1140...
      IF(KBUDG.EQ.0) WRITE(K3,187)                                      C1150...
  186 FORMAT(/11X,'- CALCULATE AND PRINT FLUID AND ',A6,' BUDGETS ',    C1160...
     1   'ON EACH TIME STEP WITH OUTPUT')                               C1170...
  187 FORMAT(/11X,'- CANCEL PRINT OF BUDGETS')                          C1180...
C                                                                       C1190...
C.....INPUT DATASET 8: ITERATION CONTROLS                               C1200...
      READ(K1,190) ITRMAX,RPMAX,RUMAX,ICON,ITRMX2,OMEGA,RPMX2,RUMX2     C1210NEW
  190 FORMAT(I10,2G10.0,2I10,3G10.0)                                    C1220NEW
      IF(ITRMAX.EQ.1) WRITE(K3,193)                                     C1230NEW
  193 FORMAT(////11X,'I T E R A T I O N   C O N T R O L   D A T A',     C1250...
     1   //11X,'  NO ITERATION FOR NON-LINEARITIES')                    C1260...
      WRITE(K3,195) ITRMAX,RPMAX,RUMAX                                  C1280...
  195 FORMAT(////11X,'I T E R A T I O N   C O N T R O L   D A T A',     C1290...
     1   //11X,I15,5X,'MAXIMUM NUMBER OF ITERATIONS PER TIME STEP',     C1300...
     2   /11X,1PD15.4,5X,'ABSOLUTE CONVERGENCE CRITERION FOR FLOW',     C1310...
     3   ' SOLUTION'/11X,1PD15.4,5X,'ABSOLUTE CONVERGENCE CRITERION',   C1320...
     4   ' FOR TRANSPORT SOLUTION')                                     C1330...
      WRITE(K3,1951) ICON,ITRMX2,OMEGA,RPMX2,RUMX2                      NEW
 1951 FORMAT(////11X,'I T E R A T I V E   S O L V E R   D A T A',       NEW
     4   //11X,I15,5X,'OPTION NUMBER FOR PRECONDITIONED CONJUGATE ',    NEW
     5   ' GRADIENT SOLVER'/11X,I15,5X,                                 NEW
     6   'MAXIMUM NUMBER OF ITERATIONS FOR ITERATIVE SOLVERS'/11X,      NEW
     7   1P1E15.4,5X,'ACCELERATION FACTOR FOR LSOR SOLUTION',           NEW
     2   /11X,1PD15.4,5X,'ABSOLUTE CONVERGENCE CRITERION FOR FLOW',     NEW
     3   ' SOLUTION'/11X,1PD15.4,5X,'ABSOLUTE CONVERGENCE CRITERION',   NEW
     4   ' FOR TRANSPORT SOLUTION')                                     NEW
      CONTINUE                                                          C1340...
C                                                                       C1350...
C.....INPUT DATASET 9: FLUID PROPERTIES                                 C1360...
      READ(K1,200) COMPFL,CW,SIGMAW,RHOW0,URHOW0,DRWDU,VISC0            C1370...
C.....INPUT DATASET 10: SOLID MATRIX PROPERTIES                         C1380...
      READ(K1,200) COMPMA,CS,SIGMAS,RHOS                                C1390...
  200 FORMAT(8G10.0)                                                    C1400...
      IF(ME.EQ.+1)                                                      C1410...
     1   WRITE(K3,210)COMPFL,COMPMA,CW,CS,VISC0,RHOS,RHOW0,DRWDU,URHOW0,C1420...
     2                  SIGMAW,SIGMAS                                   C1430...
  210 FORMAT(1H1////11X,'C O N S T A N T   P R O P E R T I E S   O F',  C1440...
     1   '   F L U I D   A N D   S O L I D   M A T R I X'               C1450...
     2   //11X,1PD15.4,5X,'COMPRESSIBILITY OF FLUID'/11X,1PD15.4,5X,    C1460...
     3   'COMPRESSIBILITY OF POROUS MATRIX'//11X,1PD15.4,5X,            C1470...
     4   'SPECIFIC HEAT CAPACITY OF FLUID',/11X,1PD15.4,5X,             C1480...
     5   'SPECIFIC HEAT CAPACITY OF SOLID GRAIN'//13X,'FLUID VISCOSITY',C1490...
     6   ' IS CALCULATED BY SUTRA AS A FUNCTION OF TEMPERATURE IN ',    C1500...
     7   'UNITS OF [kg/(m*s)]'//11X,1PD15.4,5X,'VISC0, CONVERSION ',    C1510...
     8   'FACTOR FOR VISCOSITY UNITS,  [desired units] = VISC0*',       C1520...
     9   '[kg/(m*s)]'//11X,1PD15.4,5X,'DENSITY OF A SOLID GRAIN'        C1530...
     *   //13X,'FLUID DENSITY, RHOW'/13X,'CALCULATED BY ',              C1540...
     1   'SUTRA IN TERMS OF TEMPERATURE, U, AS:'/13X,'RHOW = RHOW0 + ', C1550...
     2   'DRWDU*(U-URHOW0)'//11X,1PD15.4,5X,'FLUID BASE DENSITY, RHOW0' C1560...
     3   /11X,1PD15.4,5X,'COEFFICIENT OF DENSITY CHANGE WITH ',         C1570...
     4   'TEMPERATURE, DRWDU'/11X,1PD15.4,5X,'TEMPERATURE, URHOW0, ',   C1580...
     5   'AT WHICH FLUID DENSITY IS AT BASE VALUE, RHOW0'               C1590...
     6   //11X,1PD15.4,5X,'THERMAL CONDUCTIVITY OF FLUID'               C1600...
     7   /11X,1PD15.4,5X,'THERMAL CONDUCTIVITY OF SOLID GRAIN')         C1610...
      IF(ME.EQ.-1)                                                      C1620...
     1   WRITE(K3,220)COMPFL,COMPMA,VISC0,RHOS,RHOW0,DRWDU,URHOW0,SIGMAWC1630...
  220 FORMAT(1H1////11X,'C O N S T A N T   P R O P E R T I E S   O F',  C1640...
```

```
     1   '  F L U I D   A N D   S O L I D   M A T R I X'                C1650...
     2   //11X,1PD15.4,5X,'COMPRESSIBILITY OF FLUID'/11X,1PD15.4,5X,    C1660...
     3   'COMPRESSIBILITY OF POROUS MATRIX'                             C1670...
     4   //11X,1PD15.4,5X,'FLUID VISCOSITY'                             C1680...
     4   //11X,1PD15.4,5X,'DENSITY OF A SOLID GRAIN'                    C1690...
     5   //13X,'FLUID DENSITY, RHOW'/13X,'CALCULATED BY ',              C1700...
     6   'SUTRA IN TERMS OF SOLUTE CONCENTRATION, U, AS:',              C1710...
     7   /13X,'RHOW = RHOW0 + DRWDU*(U-URHOW0)'                         C1720...
     8   //11X,1PD15.4,5X,'FLUID BASE DENSITY, RHOW0'                   C1730...
     9   /11X,1PD15.4,5X,'COEFFICIENT OF DENSITY CHANGE WITH ',         C1740...
     *   'SOLUTE CONCENTRATION, DRWDU'                                  C1750...
     1   /11X,1PD15.4,5X,'SOLUTE CONCENTRATION, URHOW0, ',              C1760...
     4   'AT WHICH FLUID DENSITY IS AT BASE VALUE, RHOW0'               C1770...
     5   //11X,1PD15.4,5X,'MOLECULAR DIFFUSIVITY OF SOLUTE IN FLUID')   C1780...
C                                                                       C1790...
C.....INPUT DATASET 11: ADSORPTION PARAMETERS                           C1800...
      READ(K1,230) ADSMOD,CHI1,CHI2                                     C1810...
  230 FORMAT(A10,2G10.0)                                                C1820...
      IF(ME.EQ.+1) GOTO 248                                             C1830...
      IF(ADSMOD.EQ.'NONE      ') GOTO 234                               C1840...
      WRITE(K3,232) ADSMOD                                              C1850...
  232 FORMAT(////11X,'A D S O R P T I O N   P A R A M E T E R S'        C1860...
     1   //16X,A10,5X,'EQUILIBRIUM SORPTION ISOTHERM')                  C1870...
      GOTO 236                                                          C1880...
  234 WRITE(K3,235)                                                     C1890...
  235 FORMAT(////11X,'A D S O R P T I O N   P A R A M E T E R S'        C1900...
     1   //16X,'NON-SORBING SOLUTE')                                    C1910...
  236 IF((ADSMOD.EQ.'NONE ').OR.(ADSMOD.EQ.'LINEAR    ').OR.            C1920...
     1   (ADSMOD.EQ.'FREUNDLICH').OR.(ADSMOD.EQ.'LANGMUIR  ')) GOTO 238 C1930...
      WRITE(K3,237)                                                     C1940...
  237 FORMAT(//11X,'* * * * ERROR DETECTED :  TYPE OF SORPTION MODEL ', C1950...
     1   'IS NOT SPECIFIED CORRECTLY.'/11X,'CHECK FOR TYPE AND ',       C1960...
     2   'SPELLING, AND THAT TYPE IS LEFT-JUSTIFIED IN INPUT FIELD')    C1970...
      INSTOP=INSTOP-1                                                   C1980...
  238 IF(ADSMOD.EQ.'LINEAR    ') WRITE(K3,242) CHI1                     C1990...
  242 FORMAT(11X,1PD15.4,5X,'LINEAR DISTRIBUTION COEFFICIENT')          C2000...
      IF(ADSMOD.EQ.'FREUNDLICH') WRITE(K3,244) CHI1,CHI2                C2010...
  244 FORMAT(11X,1PD15.4,5X,'FREUNDLICH DISTRIBUTION COEFFICIENT'       C2020...
     1   /11X,1PD15.4,5X,'SECOND FREUNDLICH COEFFICIENT')               C2030...
      IF(ADSMOD.EQ.'FREUNDLICH'.AND.CHI2.LE.0.D0) THEN                  C2040...
       WRITE(K3,245)                                                    C2050...
  245  FORMAT(11X,'* * * * ERROR DETECTED :  SECOND COEFFICIENT ',      C2060...
     1    'MUST BE GREATER THAN ZERO')                                  C2070...
       INSTOP=INSTOP-1                                                  C2080...
       ENDIF                                                            C2090...
      IF(ADSMOD.EQ.'LANGMUIR  ') WRITE(K3,246) CHI1,CHI2                C2100...
  246 FORMAT(11X,1PD15.4,5X,'LANGMUIR DISTRIBUTION COEFFICIENT'         C2110...
     1   /11X,1PD15.4,5X,'SECOND LANGMUIR COEFFICIENT')                 C2120...
C                                                                       C2130...
C.....INPUT DATASET 12: PRODUCTION OF ENERGY OR SOLUTE MASS             C2140...
  248 READ(K1,200) PRODF0,PRODS0,PRODF1,PRODS1                          C2150...
      IF(ME.EQ.-1) WRITE(K3,250) PRODF0,PRODS0,PRODF1,PRODS1            C2160...
  250 FORMAT(////11X,'P R O D U C T I O N   A N D   D E C A Y   O F   ',C2170...
     1   'S P E C I E S   M A S S'//13X,'PRODUCTION RATE (+)'/13X,      C2180...
     2   'DECAY RATE (-)'//11X,1PD15.4,5X,'ZERO-ORDER RATE OF SOLUTE ', C2190...
     3   'MASS PRODUCTION/DECAY IN FLUID'/11X,1PD15.4,5X,               C2200...
     4   'ZERO-ORDER RATE OF ADSORBATE MASS PRODUCTION/DECAY IN ',      C2210...
     5   'IMMOBILE PHASE'/11X,1PD15.4,5X,'FIRST-ORDER RATE OF SOLUTE ', C2220...
     3   'MASS PRODUCTION/DECAY IN FLUID'/11X,1PD15.4,5X,               C2230...
     4   'FIRST-ORDER RATE OF ADSORBATE MASS PRODUCTION/DECAY IN ',     C2240...
     5   'IMMOBILE PHASE')                                              C2250...
      IF(ME.EQ.+1) WRITE(K3,260) PRODF0,PRODS0                          C2260...
```

```
  260 FORMAT(////11X,'P R O D U C T I O N   A N D   L O S S   O F   ',  C2270...
     1   'E N E R G Y'//13X,'PRODUCTION RATE (+)'/13X,                  C2280...
     2   'LOSS RATE (-)'//11X,1PD15.4,5X,'ZERO-ORDER RATE OF ENERGY ',  C2290...
     3   'PRODUCTION/LOSS IN FLUID'/11X,1PD15.4,5X,                     C2300...
     4   'ZERO-ORDER RATE OF ENERGY PRODUCTION/LOSS IN ',               C2310...
     5   'SOLID GRAINS')                                                C2320...
C.....SET PARAMETER SWITCHES FOR EITHER ENERGY OR SOLUTE TRANSPORT      C2330...
      IF(ME) 272,272,274                                                C2340...
C     FOR SOLUTE TRANSPORT:                                             C2350...
  272 CS=0.0D0                                                          C2360...
      CW=1.D00                                                          C2370...
      SIGMAS=0.0D0                                                      C2380...
      GOTO 278                                                          C2390...
C     FOR ENERGY TRANSPORT:                                             C2400...
  274 ADSMOD='NONE      '                                               C2410...
      CHI1=0.0D0                                                        C2420...
      CHI2=0.0D0                                                        C2430...
      PRODF1=0.0D0                                                      C2440...
      PRODS1=0.0D0                                                      C2450...
C     DIVIDE SIGMA TO CANCEL MULTIPLICATION BY RHOW*CW                  C2460...
C        IN SUBROUTINE ELEMEN.                                          C2470...
      RC0=RHOW0*CW                                                      C2480...
      SIGMAW=SIGMAW/RC0                                                 C2490...
      SIGMAS=SIGMAS/RC0                                                 C2500...
  278 CONTINUE                                                          C2510...
C                                                                       C2520...
C.....INPUT DATASET 13: ORIENTATION OF COORDINATES TO GRAVITY           C2530...
      READ(K1,200) GRAVX,GRAVY                                          C2540...
      WRITE(K3,320) GRAVX,GRAVY                                         C2550...
  320 FORMAT(////11X,'C O O R D I N A T E   O R I E N T A T I O N   ',  C2560...
     1   'T O   G R A V I T Y'//13X,'COMPONENT OF GRAVITY VECTOR',      C2570...
     2   /13X,'IN +X DIRECTION, GRAVX'/11X,1PD15.4,5X,                  C2580...
     3   'GRAVX = -GRAV * D(ELEVATION)/DX'//13X,'COMPONENT OF GRAVITY', C2590...
     4   ' VECTOR'/13X,'IN +Y DIRECTION, GRAVY'/11X,1PD15.4,5X,         C2600...
     5   'GRAVY = -GRAV * D(ELEVATION)/DY')                             C2610...
C                                                                       C2620...
C.....INPUT DATASETS 14A AND 14B: NODEWISE DATA                         C2630...
      READ(K1,330) SCALX,SCALY,SCALTH,PORFAC                            C2640...
  330 FORMAT(10X,4G10.0)                                                C2650...
      DO 450 I=1,NN                                                     C2660...
      READ(K1,400) II,X(II),Y(II),THICK(II),POR(II)                     C2670...
  400 FORMAT(I5,5X,4G10.0)                                              C2680NEW
      X(II)=X(II)*SCALX                                                 C2690...
      Y(II)=Y(II)*SCALY                                                 C2700...
      THICK(II)=THICK(II)*SCALTH                                        C2710...
      POR(II)=POR(II)*PORFAC                                            C2720...
C     SET SPECIFIC PRESSURE STORATIVITY, SOP.                           C2730...
  450 SOP(II)=(1.D0-POR(II))*COMPMA+POR(II)*COMPFL                      C2740...
  460 IF(KNODAL.EQ.0) WRITE(K3,469) SCALX,SCALY,SCALTH,PORFAC           C2750...
  469 FORMAT(1H1////11X,'N O D E   I N F O R M A T I O N'//16X,         C2760...
     1   'PRINTOUT OF NODE COORDINATES, THICKNESSES AND POROSITIES ',   C2770...
     2   'CANCELLED.'//16X,'SCALE FACTORS :'/33X,1PD15.4,5X,'X-SCALE'/  C2780...
     1   33X,1PD15.4,5X,'Y-SCALE'/33X,1PD15.4,5X,'THICKNESS FACTOR'/    C2790...
     2   33X,1PD15.4,5X,'POROSITY FACTOR')                              C2800...
      IF(KNODAL.EQ.+1)WRITE(K3,470)(I,X(I),Y(I),THICK(I),POR(I),I=1,NN) C2810...
  470 FORMAT(1H1//11X,'N O D E   I N F O R M A T I O N'//13X,           C2820...
     1   'NODE',7X,'X',16X,'Y',17X,'THICKNESS',6X,'POROSITY'//          C2830...
     2   (11X,I6,3(3X,1PD14.5),6X,0PF8.5))                              C2840...
C                                                                       C2850...
C.....INPUT DATASETS 15A AND 15B: ELEMENTWISE DATA                      C2860...
      READ(K1,490) PMAXFA,PMINFA,ANGFAC,ALMAXF,ALMINF,ATAVGF            C2870...
  490 FORMAT(10X,6G10.0)                                                C2880...
```

```
      IF(KELMNT.EQ.+1) WRITE(K3,500)                                    C2890...
  500 FORMAT(1H1//11X,'E L E M E N T   I N F O R M A T I O N'//         C2900...
     1   11X,'ELEMENT',4X,'MAXIMUM',9X,'MINIMUM',12X,                   C2910...
     2   'ANGLE BETWEEN',3X,'      MAXIMUM',5X,'   MINIMUM',5X,         C2920...
     3   '   AVERAGE'/22X,'PERMEABILITY',4X,'PERMEABILITY',4X,          C2930...
     4   '+X-DIRECTION AND',3X,'LONGITUDINAL',3X,'LONGITUDINAL'3X,      C2940...
     5   '  TRANSVERSE'/50X,'MAXIMUM PERMEABILITY',3X,'DISPERSIVITY',   C2950...
     6   3X,'DISPERSIVITY',3X,'DISPERSIVITY'/58X,'(IN DEGREES)'//)      C2960...
      DO 550 LL=1,NE                                                    C2970...
      READ(K1,510) L,PMAX,PMIN,ANGLEX,ALMAX(L),ALMIN(L),ATAVG(L)        C2980...
  510 FORMAT(I5,5X,6G10.0)                                              C2990NEW
      PMAX=PMAX*PMAXFA                                                  C3000...
      PMIN=PMIN*PMINFA                                                  C3010...
      ANGLEX=ANGLEX*ANGFAC                                              C3020...
      ALMAX(L)=ALMAX(L)*ALMAXF                                          C3030...
      ALMIN(L)=ALMIN(L)*ALMINF                                          C3040...
      ATAVG(L)=ATAVG(L)*ATAVGF                                          C3050...
      IF(KELMNT.EQ.+1) WRITE(K3,520) L,PMAX,PMIN,ANGLEX,                C3060...
     1   ALMAX(L),ALMIN(L),ATAVG(L)                                     C3070...
  520 FORMAT(11X,I7,2X,2(1PD14.5,2X),8X,4(0PF10.3,5X))                  C3080...
C                                                                       C3090...
C.....ROTATE PERMEABILITY FROM MAXIMUM/MINIMUM TO X/Y DIRECTIONS        C3100...
      RADIAX=1.745329D-02*ANGLEX                                        C3110...
      SINA=DSIN(RADIAX)                                                 C3120...
      COSA=DCOS(RADIAX)                                                 C3130...
      SINA2=SINA*SINA                                                   C3140...
      COSA2=COSA*COSA                                                   C3150...
      PERMXX(L)=PMAX*COSA2+PMIN*SINA2                                   C3160...
      PERMYY(L)=PMAX*SINA2+PMIN*COSA2                                   C3170...
      PERMXY(L)=(PMAX-PMIN)*SINA*COSA                                   C3180...
      PERMYX(L)=PERMXY(L)                                               C3190...
      PANGLE(L)=RADIAX                                                  C3200...
  550 CONTINUE                                                          C3210...
      IF(KELMNT.EQ.0)                                                   C3220...
     1   WRITE(K3,569) PMAXFA,PMINFA,ANGFAC,ALMAXF,ALMINF,ATAVGF        C3230...
  569 FORMAT(////11X,'E L E M E N T   I N F O R M A T I O N'//          C3240...
     1   16X,'PRINTOUT OF ELEMENT PERMEABILITIES AND DISPERSIVITIES ',  C3250...
     2   'CANCELLED.'//16X,'SCALE FACTORS :'/33X,1PD15.4,5X,'MAXIMUM ', C3260...
     1   'PERMEABILITY FACTOR'/33X,1PD15.4,5X,'MINIMUM PERMEABILITY ',  C3270...
     2   'FACTOR'/33X,1PD15.4,5X,'ANGLE FROM +X TO MAXIMUM DIRECTION',  C3280...
     3   ' FACTOR'/33X,1PD15.4,5X,'MAXIMUM LONGITUDINAL DISPERSIVITY',  C3290...
     4   ' FACTOR'/33X,1PD15.4,5X,'MINIMUM LONGITUDINAL DISPERSIVITY',  C3300...
     5   ' FACTOR'/33X,1PD15.4,5X,'TRANSVERSE DISPERSIVITY FACTOR')     C3310...
C                                                                       C3320...
C.....END SIMULATION FOR CORRECTIONS TO UNIT-5 DATA IF NECESSARY        C3330...
      IF(INSTOP.EQ.0) GOTO 1000                                         C3340...
      WRITE(K3,999)                                                     C3350...
  999 FORMAT(////////11X,'PLEASE CORRECT INPUT DATA AND RERUN.',        C3360...
     1   ///22X,'S I M U L A T I O N   H A L T E D',                    C3370...
     2     /22X,'*******************   ***********')                    C3380...
      ENDFILE(K3)                                                       C3390...
      STOP                                                              C3400...
C                                                                       C3410...
C                                                                       C3420...
1000  IF(KINCID.EQ.0) WRITE(K3,1)                                       NEW
    1 FORMAT(1H1////11X,'M E S H   C O N N E C T I O N   D A T A'//     NEW
     1   16X,'PRINTOUT OF NODAL INCIDENCES CANCELLED.')                 NEW
      IF(KINCID.EQ.+1) WRITE(K3,2)                                      NEW
    2 FORMAT(1H1////11X,'M E S H   C O N N E C T I O N   D A T A',      NEW
     1   ///11X,'**** NODAL INCIDENCES ****'///)                        NEW
C                                                                       NEW
```

```
C.....CALCULATE INCIDENCES FOR REGULAR GRID                             NEW
      NEX=IS-1                                                          NEW
      NEY=JT-1                                                          NEW
      NELEMN=0                                                          NEW
      DO 560 IE2=1,NEX                                                  NEW
      DO 560 IE1=1,NEY                                                  NEW
      NELEMN=NELEMN+1                                                   NEW
      N0=IE1+(IE2-1)*JT                                                 NEW
      IIN(1)=N0                                                         NEW
      IIN(2)=N0+JT                                                      NEW
      IIN(3)=N0+JT+1                                                    NEW
      IIN(4)=N0+1                                                       NEW
C                                                                       NEW
C.....PREPARE NODE INCIDENCE LIST FOR MESH, IN.                         NEW
      DO 570 II=1,4                                                     NEW
      III=II+(NELEMN-1)*4                                               NEW
  570 IN(III)=IIN(II)                                                   NEW
C                                                                       NEW
      IF(KINCID.EQ.0) GOTO 560                                          NEW
      WRITE(K3,650) NELEMN,(IIN(M),M=1,4)                               NEW
  650 FORMAT(11X,'ELEMENT',I6,5X,' NODES AT : ',6X,'CORNERS ',          NEW
     1    5(1H*),4I6,1X,5(1H*))                                         NEW
C                                                                       NEW
  560 CONTINUE                                                          NEW
C                                                                       NEW
C ***       NOTE: BANDWIDTH FOR A REGULAR GRID IS FIXED                 NEW
      WRITE(K3,2500) NBIP,NBIS                                          NEW
 2500 FORMAT(////13X,'BANDWIDTH FOR PRESSURE MATRIX, ',I4/              NEW
     1 13X,'BANDWIDTH FOR TRANSPORT MATRIX, ',I4)                       NEW
C                                                                       NEW
C     SET UP POINTER ARRAYS FOR MATRICES                                NEW
      NPT(1)=-JT                                                        NEW
      NPT(2)=1-JT                                                       NEW
      NPT(3)=2-JT                                                       NEW
      NPT(4)=0                                                          NEW
      NPT(5)=1                                                          NEW
      NPT(6)=2                                                          NEW
      NPT(7)=JT                                                         NEW
      NPT(8)=1+JT                                                       NEW
      NPT(9)=JT+2                                                       NEW
C                                                                       NEW
      RETURN                                                            NEW
      END                                                               C3440...
C     SUBROUTINE         S  O  U  R  C  E         SUTRA - VERSION 1284-2D E10.....
C                                                                       E20.....
C *** PURPOSE :                                                         E30.....
C ***  TO READ AND ORGANIZE FLUID MASS SOURCE DATA AND ENERGY OR        E40.....
C ***  SOLUTE MASS SOURCE DATA.                                         E50.....
C                                                                       E60.....
      SUBROUTINE SOURCE(QIN,UIN,IQSOP,QUIN,IQSOU,IQSOPT,IQSOUT)         E70.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               E80.....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6                           MODIFIED
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          E90NEW
     1    NSOP,NSOU,NBCN                                                E100....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        E110NEW
     1    NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME           E120NEW
      DIMENSION QIN(NN),UIN(NN),IQSOP(NSOP),QUIN(NN),IQSOU(NSOU)        E130....
C                                                                       E140....
```

```
C.....NSOPI IS ACTUAL NUMBER OF FLUID SOURCE NODES
C.....NSOUI IS ACTUAL NUMBER OF SOLUTE MASS OR ENERGY SOURCE NODES
      NSOPI=NSOP-1
      NSOUI=NSOU-1
      IQSOPT=1
      IQSOUT=1
      NIQP=0
      NIQU=0
      IF(NSOPI.EQ.0) GOTO 1000
      IF(ME) 50,50,150
   50 WRITE(K3,100)
  100 FORMAT(1H1////11X,'F L U I D   S O U R C E   D A T A'
     1   ////11X,'**** NODES AT WHICH FLUID INFLOWS OR OUTFLOWS ARE ',
     2   'SPECIFIED ****'//11X,'NODE NUMBER',10X,
     3   'FLUID INFLOW(+)/OUTFLOW(-)',5X,'SOLUTE CONCENTRATION OF'
     4   /11X,'(MINUS INDICATES',5X,'(FLUID MASS/SECOND)',
     5   12X,'INFLOWING FLUID'/12X,'TIME-VARYING',39X,
     6   '(MASS SOLUTE/MASS WATER)'/12X,'FLOW RATE OR'/12X,
     7   'CONCENTRATION)'//)
      GOTO 300
  150 WRITE(K3,200)
  200 FORMAT(1H1////11X,'F L U I D   S O U R C E   D A T A'
     1   ////11X,'**** NODES AT WHICH FLUID INFLOWS OR OUTFLOWS ARE ',
     2   'SPECIFIED ****'//11X,'NODE NUMBER',10X,
     3   'FLUID INFLOW(+)/OUTFLOW(-)',5X,'TEMPERATURE [DEGREES CELCIUS]'
     4   /11X,'(MINUS INDICATES',5X,'(FLUID MASS/SECOND)',12X,
     5   'OF INFLOWING FLUID'/12X,'TIME-VARYING'/12X,'FLOW OR'/12X,
     6   'TEMPERATURE)'//)
C
C.....INPUT DATASET 17
  300 CONTINUE
      READ(K1,400) IQCP,QINC,UINC
  400 FORMAT(I10,2G15.0)
      IF(IQCP.EQ.0) GOTO 700
      NIQP=NIQP+1
      IQSOP(NIQP)=IQCP
      IF(IQCP.LT.0) IQSOPT=-1
      IQP=IABS(IQCP)
      QIN(IQP)=QINC
      UIN(IQP)=UINC
      IF(IQCP.GT.0) GOTO 450
      WRITE(K3,500) IQCP
      GOTO 600
  450 IF(QINC.GT.0) GOTO 460
      WRITE(K3,500) IQCP,QINC
      GOTO 600
  460 WRITE(K3,500) IQCP,QINC,UINC
  500 FORMAT(11X,I10,13X,1PE14.7,16X,1PE14.7)
  600 GOTO 300
  700 IF(NIQP.EQ.NSOPI) GOTO 890
C.....END SIMULATION IF THERE NEED BE CORRECTIONS TO DATASET 17
      WRITE(K3,750) NIQP,NSOPI
  750 FORMAT(////11X,'THE NUMBER OF FLUID SOURCE NODES READ, ',I5,
     1    ' IS NOT EQUAL TO THE NUMBER SPECIFIED, ',I5////
     2   11X,'PLEASE CORRECT DATA AND RERUN'////////
     3   22X,'S I M U L A T I O N   H A L T E D'/
     4   22X,'__________________________________')
      ENDFILE(K3)
      STOP
  890 IF(IQSOPT.EQ.-1) WRITE(K3,900)
  900 FORMAT(////11X,'THE SPECIFIED TIME VARIATIONS ARE ',
     1   'USER-PROGRAMMED IN SUBROUTINE  B C T I M E .')
```

```
C                                                                        E770....
C                                                                        E780....
 1000 IF(NSOUI.EQ.0) GOTO 9000                                           E790....
      IF(ME) 1050,1050,1150                                              E800....
 1050 WRITE(K3,1100)                                                     E810....
 1100 FORMAT(////////11X,'S O L U T E   S O U R C E   D A T A'           E820....
     1   ////11X,'**** NODES AT WHICH SOURCES OR SINKS OF SOLUTE ',      E830....
     2   'MASS ARE SPECIFIED ****'//11X,'NODE NUMBER',10X,               E840....
     3   'SOLUTE SOURCE(+)/SINK(-)'/11X,'(MINUS INDICATES',5X,           E850....
     4   '(SOLUTE MASS/SECOND)'/12X,'TIME-VARYING'/12X,                  E860....
     5   'SOURCE OR SINK)'//)                                            E870....
      GOTO 1300                                                          E880....
 1150 WRITE(K3,1200)                                                     E890....
 1200 FORMAT(////////11X,'E N E R G Y   S O U R C E   D A T A'           E900....
     1   ////11X,'**** NODES AT WHICH SOURCES OR SINKS OF ',             E910....
     2   'ENERGY ARE SPECIFIED ****'//11X,'NODE NUMBER',10X,             E920....
     3   'ENERGY SOURCE(+)/SINK(-)'/11X,'(MINUS INDICATES',5X,           E930....
     4   '(ENERGY/SECOND)'/12X,'TIME-VARYING'/12X,                       E940....
     5   'SOURCE OR SINK)'//)                                            E950....
C                                                                        E960....
C.....INPUT DATASET 18                                                   E970....
 1300 CONTINUE                                                           E980....
      READ(K1,400) IQCU,QUINC                                            E990....
      IF(IQCU.EQ.0) GOTO 1700                                            E1000...
      NIQU=NIQU+1                                                        E1010...
      IQSOU(NIQU)=IQCU                                                   E1020...
      IF(IQCU.LT.0) IQSOUT=-1                                            E1030...
      IQU=IABS(IQCU)                                                     E1040...
      QUIN(IQU)=QUINC                                                    E1050...
      IF(IQCU.GT.0) GOTO 1450                                            E1060...
      WRITE(K3,1500) IQCU                                                E1070...
      GOTO 1600                                                          E1080...
 1450 WRITE(K3,1500) IQCU,QUINC                                          E1090...
 1500 FORMAT(11X,I10,13X,1PE14.7)                                        E1100...
 1600 GOTO 1300                                                          E1110...
 1700 IF(NIQU.EQ.NSOUI) GOTO 1890                                        E1120...
C.....END SIMULATION IF THERE NEED BE CORRECTIONS TO DATASET 18           E1130...
      IF(ME) 1740,1740,1760                                              E1140...
 1740 WRITE(K3,1750) NIQU,NSOUI                                          E1150...
 1750 FORMAT(////11X,'THE NUMBER OF SOLUTE SOURCE NODES READ, ',I5,      E1160...
     1    ' IS NOT EQUAL TO THE NUMBER SPECIFIED, ',I5////               E1170...
     2   11X,'PLEASE CORRECT DATA AND RERUN'////////                     E1180...
     3   22X,'S I M U L A T I O N   H A L T E D'/                        E1190...
     4   22X,'_______________________________________')                  E1200...
      ENDFILE(K3)                                                        E1210...
      STOP                                                               E1220...
 1760 WRITE(K3,1770) NIQU,NSOUI                                          E1230...
 1770 FORMAT(////11X,'THE NUMBER OF ENERGY SOURCE NODES READ, ',I5,      E1240...
     1      ' IS NOT EQUAL TO THE NUMBER SPECIFIED, ',I5////             E1250...
     2   11X,'PLEASE CORRECT DATA AND RERUN'////////                     E1260...
     3   22X,'S I M U L A T I O N   H A L T E D'/                        E1270...
     4   22X,'_______________________________________')                  E1280...
      ENDFILE(K3)                                                        E1290...
      STOP                                                               E1300...
 1890 IF(IQSOUT.EQ.-1) WRITE(K3,900)                                     E1310...
C                                                                        E1320...
 9000 RETURN                                                             E1330...
C                                                                        E1340...
      END                                                                E1350...
C     SUBROUTINE         B  O  U  N  D            SUTRA - VERSION 1284-2D F10.....
C                                                                        F20.....
C *** PURPOSE :                                                          F30.....
C ***  TO READ AND ORGANIZE SPECIFIED PRESSURE DATA AND                  F40.....
```

```
C ***  SPECIFIED TEMPERATURE OR CONCENTRATION DATA.                     F50.....
C                                                                       F60.....
      SUBROUTINE BOUND(IPBC,PBC,IUBC,UBC,IPBCT,IUBCT)                   F70.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               F80.....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6                           MODIFIED
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          F90NEW
     1   NSOP,NSOU,NBCN                                                 F100
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        F110....
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            F120NEW
      DIMENSION IPBC(NBCN),PBC(NBCN),IUBC(NBCN),UBC(NBCN)               F130....
C                                                                       F140....
C                                                                       F150....
      IPBCT=1                                                           F160....
      IUBCT=1                                                           F170....
      ISTOPP=0                                                          F180....
      ISTOPU=0                                                          F190....
      IPU=0                                                             F200....
      WRITE(K3,50)                                                      F210....
   50 FORMAT(1H1////11X,'B O U N D A R Y   C O N D I T I O N S')        F220....
      IF(NPBC.EQ.0) GOTO 400                                            F230....
      WRITE(K3,100)                                                     F240....
  100 FORMAT(//11X,'**** NODES AT WHICH PRESSURES ARE',                 F250....
     1    ' SPECIFIED ****'/)                                           F260....
      IF(ME) 107,107,114                                                F270....
  107 WRITE(K3,108)                                                     F280....
  108 FORMAT(11X,'     (AS WELL AS SOLUTE CONCENTRATION OF ANY'         F290....
     1    /16X,' FLUID INFLOW WHICH MAY OCCUR AT THE POINT'             F300....
     2    /16X,' OF SPECIFIED PRESSURE)'//12X,'NODE',18X,'PRESSURE',    F310....
     3    13X,'CONCENTRATION'//)                                        F320....
      GOTO 120                                                          F330....
  114 WRITE(K3,115)                                                     F340....
  115 FORMAT(11X,'     (AS WELL AS TEMPERATURE [DEGREES CELCIUS] OF ANY'F350....
     1    /16X,' FLUID INFLOW WHICH MAY OCCUR AT THE POINT'             F360....
     2    /16X,' OF SPECIFIED PRESSURE)'//12X,'NODE',18X,               F370....
     2    'PRESSURE',13X,'  TEMPERATURE'//)                             F380....
C                                                                       F390....
C.....INPUT DATASET 14                                                  F400....
  120 IPU=IPU+1                                                         F410....
      READ(K1,150) IPBC(IPU),PBC(IPU),UBC(IPU)                          F420....
  150 FORMAT(I5,2G20.0)                                                 F430....
      IF(IPBC(IPU).LT.0) IPBCT=-1                                       F440....
      IF(IPBC(IPU).EQ.0) GOTO 180                                       F450....
      IF(IPBC(IPU).GT.0) WRITE(K3,160) IPBC(IPU),PBC(IPU),UBC(IPU)      F460....
      IF(IPBC(IPU).LT.0) WRITE(K3,160) IPBC(IPU)                        F470....
  160 FORMAT(11X,I5,6X,1PD20.13,6X,1PD20.13)                            F480....
      GOTO 120                                                          F490....
  180 IPU=IPU-1                                                         F500....
      IP=IPU                                                            F510....
      IF(IP.EQ.NPBC) GOTO 200                                           F520....
      ISTOPP=1                                                          F530....
  200 IF(IPBCT.NE.-1) GOTO 400                                          F540....
      IF(ME) 205,205,215                                                F550....
  205 WRITE(K3,206)                                                     F560....
  206 FORMAT(//12X,'TIME-DEPENDENT SPECIFIED PRESSURE'/12X,'OR INFLOW ',F570....
     1    'CONCENTRATION INDICATED'/12X,'BY NEGATIVE NODE NUMBER')      F580....
      GOTO 400                                                          F590....
  215 WRITE(K3,216)                                                     F600....
  216 FORMAT(//11X,'TIME-DEPENDENT SPECIFIED PRESSURE'/12X,'OR INFLOW ',F610....
     1    'TEMPERATURE INDICATED'/12X,'BY NEGATIVE NODE NUMBER')        F620....
  400 IF(NUBC.EQ.0) GOTO 2000                                           F630....
C                                                                       F640....
      IF(ME) 500,500,550                                                F650....
```

```
  500 WRITE(K3,1000)
 1000 FORMAT(////11X,'**** NODES AT WHICH SOLUTE CONCENTRATIONS ARE ',
     1   'SPECIFIED TO BE INDEPENDENT OF LOCAL FLOWS AND FLUID SOURCES',
     2   ' ****'//12X,'NODE',13X,'CONCENTRATION'//)
      GOTO 1120
  550 WRITE(K3,1001)
 1001 FORMAT(////11X,'**** NODES AT WHICH TEMPERATURES ARE ',
     1   'SPECIFIED TO BE INDEPENDENT OF LOCAL FLOWS AND FLUID SOURCES',
     2   ' ****'//12X,'NODE',15X,'TEMPERATURE'//)
C
C.....INPUT DATASET 20
 1120 IPU=IPU+1
      READ(K1,150) IUBC(IPU),UBC(IPU)
      IF(IUBC(IPU).LT.0) IUBCT=-1
      IF(IUBC(IPU).EQ.0) GOTO 1180
      IF(IUBC(IPU).GT.0) WRITE(K3,1150) IUBC(IPU),UBC(IPU)
      IF(IUBC(IPU).LT.0) WRITE(K3,1150) IUBC(IPU)
 1150 FORMAT(11X,I5,6X,1PD20.13)
      GOTO 1120
 1180 IPU=IPU-1
      IU=IPU-IP
      IF(IU.EQ.NUBC) GOTO 1200
      ISTOPU=1
 1200 IF(IUBCT.NE.-1) GOTO 2000
      IF(ME) 1205,1205,1215
 1205 WRITE(K3,1206)
 1206 FORMAT(//12X,'TIME-DEPENDENT SPECIFIED CONCENTRATION'/12X,'IS ',
     1   'INDICATED BY NEGATIVE NODE NUMBER')
      GOTO 2000
 1215 WRITE(K3,1216)
 1216 FORMAT(//11X,'TIME-DEPENDENT SPECIFIED TEMPERATURE'/12X,'IS ',
     1   'INDICATED BY NEGATIVE NODE NUMBER')
C
C.....END SIMULATION IF THERE NEED BE CORRECTIONS TO DATASET 19 OR 20
 2000 IF(ISTOPP.EQ.0.AND.ISTOPU.EQ.0) GOTO 6000
      IF(ISTOPP.EQ.1) WRITE(K3,3000) IP,NPBC
 3000 FORMAT(////11X,'ACTUAL NUMBER OF SPECIFIED PRESSURE NODES',
     1   ' READ, ',I5,', IS NOT EQUAL TO NUMBER SPECIFIED IN',
     2   ' INPUT, ',I5)
      IF(ME) 3500,3500,4600
 3500 IF(ISTOPU.EQ.1) WRITE(K3,4000) IU,NUBC
 4000 FORMAT(////11X,'ACTUAL NUMBER OF SPECIFIED CONCENTRATION NODES',
     1   ' READ, ',I5,', IS NOT EQUAL TO NUMBER SPECIFIED IN',
     2   ' INPUT, ',I5)
      GOTO 4800
 4600 IF(ISTOPU.EQ.1) WRITE(K3,4700) IU,NUBC
 4700 FORMAT(////11X,'ACTUAL NUMBER OF SPECIFIED TEMPERATURE NODES',
     1   ' READ, ',I5,', IS NOT EQUAL TO NUMBER SPECIFIED IN',
     2   ' INPUT, ',I5)
 4800 WRITE(K3,5000)
  000 FORMAT(////11X,'PLEASE CORRECT DATA AND RERUN.'////////
     1   22X,'S I M U L A T I O N   H A L T E D'/
     2   22X,'____________________________________________')
      ENDFILE(K3)
      STOP

    0 IF(IPBCT.EQ.-1.OR.IUBCT.EQ.-1) WRITE(K3,7000)
    0 FORMAT(////11X,'THE SPECIFIED TIME VARIATIONS ARE ',
     1   'USER-PROGRAMMED IN SUBROUTINE  B C T I M E .')


      RETURN
      END
```

```
C     SUBROUTINE        O  B  S  E  R  V        SUTRA - VERSION 1284-2D G10.....
C                                                                       G20.....
C *** PURPOSE :                                                         G30.....
C ***  (1) TO READ AND ORGANIZE OBSERVATION NODE DATA                   G40.....
C ***  (2) TO MAKE OBSERVATIONS ON PARTICULAR TIME STEPS                G50.....
C ***  (3) TO OUTPUT OBSERVATIONS AFTER COMPLETION OF SIMULATION        G60.....
C                                                                       G70.....
      SUBROUTINE OBSERV(ICALL,IOBS,ITOBS,POBS,UOBS,OBSTIM,PVEC,UVEC,    G80.....
     1   ISTOP)                                                         G90.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               G100....
      CHARACTER*13 UNAME(2)                                             G110....
      CHARACTER*10 UNDERS                                               G120....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6                           MODIFIED
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          G130NEW
     1   NSOP,NSOU,NBCN                                                 G140....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        G150NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            G160NEW
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        G170....
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        G180....
      COMMON/OBS/ NOBSN,NTOBSN,NOBCYC,ITCNT                             G190....
      DIMENSION INOB(66)                                                G200....
      DIMENSION IOBS(NOBSN),POBS(NOBSN,NTOBSN),UOBS(NOBSN,NTOBSN),      G210....
     1   OBSTIM(NTOBSN),ITOBS(NTOBSN),PVEC(NN),UVEC(NN)                 G220....
      DATA UNAME(1)/'CONCENTRATION'/,UNAME(2)/'  TEMPERATURE'/,         G230....
     1   UNDERS/'__________'/,                                          G240....
     1   ITCNT/0000/                                                    G250....
C                                                                       G260....
C.....NOBS IS ACTUAL NUMBER OF OBSERVATION NODES                        G270....
C.....NTOBS IS MAXIMUM NUMBER OF TIME STEPS WITH OBSERVATIONS           G280....
      NOBS=NOBSN-1                                                      G290....
      NTOBS=NTOBSN-2                                                    G300....
      IOB=0                                                             G530NEW
      IF(ICALL-1) 50,500,5000                                           G310....
C                                                                       G320....
C.....INITIALIZATION CALL                                               G330....
C.....INPUT DATASET 21                                                  G340....
   50 CONTINUE                                                          G350....
      JSTOP=0                                                           G360....
      WRITE(K3,60)                                                      G370....
   60 FORMAT(////11X,'O B S E R V A T I O N   N O D E S')               G380....
      READ(K1,65) NOBCYC                                                G390....
   65 FORMAT(I10)                                                       G400....
      WRITE(K3,70) NOBCYC                                               G410....
   70 FORMAT(//11X,'**** NODES AT WHICH OBSERVATIONS WILL BE MADE',     G420....
     1   ' EVERY',I5,' TIME STEPS ****'//)                              G430....
      NTOBSP=ITMAX/NOBCYC                                               G440....
      IF(NTOBSP.GT.NTOBS) WRITE(K3,80) NTOBS,NTOBSP,ITMAX               G450....
   80 FORMAT(//11X,'-  W  A  R  N  I  N  G  -'/11X,                     G460....
     1   'NUMBER OF OBSERVATION STEPS SPECIFIED ',I5,                   G470....
     2   ', IS LESS THAN THE NUMBER POSSIBLE ',I5,','/                  G480....
     3   11X,'WITHIN THE MAXIMUM NUMBER OF ALLOWED TIME STEPS, ',I5,'.'/G490....
     4   11X,'PLEASE RECONFIRM THAT OBSERVATION COUNTS ARE CORRECT.'//) G500....
  100 READ(K1,150) INOB                                                 G510....
  150 FORMAT(16I5)                                                      G520....
      DO 200 JJ=1,16                                                    G540....
      IF(INOB(JJ).EQ.0) GOTO 250                                        G550....
      IOB=IOB+1                                                         G560....
      IOBS(IOB)=INOB(JJ)                                                G570....
  200 CONTINUE                                                          G580....
      IF(IOB.LT.NOBS) GOTO 100                                          G590....
  250 IF(IOB.NE.NOBS) JSTOP=1                                           G600....
      WRITE(K3,300) (IOBS(JJ),JJ=1,NOBS)                                G610....
```

```
  300 FORMAT((11X,16(3X,I6)))                                         G620....
      IF(ISTOP.EQ.0) GOTO 400                                         G630....
C.....END SIMULATION IF CORRECTIONS ARE NECESSARY IN DATASET 21       G640....
      WRITE(K3,350) IOB,NOBS                                          G650....
  350 FORMAT(////11X,'ACTUAL NUMBER OF OBSERVATION NODES',            G660....
     1   ' READ, ',I5,', IS NOT EQUAL TO NUMBER SPECIFIED IN',        G670....
     2   ' INPUT, ',I5///11X,'PLEASE CORRECT DATA AND RERUN.',        G680....
     3   ////////22X,'S I M U L A T I O N   H A L T E D'/             G690....
     4   22X,       '_____________________________________')          G700....
      STOP                                                            G710....
  400 RETURN                                                          G720....
C                                                                     G730....
C.....MAKE OBSERVATIONS EACH NOBCYC TIME STEPS                        G740....
  500 CONTINUE                                                        G750....
      IF(MOD(IT,NOBCYC).NE.0.AND.IT.GT.1.AND.ISTOP.EQ.0) RETURN       G760....
      IF(IT.EQ.0) RETURN                                              G770....
      ITCNT=ITCNT+1                                                   G780....
      ITOBS(ITCNT)=IT                                                 G790....
      OBSTIM(ITCNT)=TSEC                                              G800....
      DO 1000 JJ=1,NOBS                                               G810....
      I=IOBS(JJ)                                                      G820....
      POBS(JJ,ITCNT)=PVEC(I)                                          G830....
      UOBS(JJ,ITCNT)=UVEC(I)                                          G840....
 1000 CONTINUE                                                        G850....
      RETURN                                                          G860....
C                                                                     G870....
C.....OUTPUT OBSERVATIONS                                             G880....
 5000 CONTINUE                                                        G890....
      MN=2                                                            G900....
      IF(ME.EQ.-1) MN=1                                               G910....
      JJ2=0                                                           G920....
      MLOOP=(NOBS+3)/4                                                G930....
      DO 7000 LOOP=1,MLOOP                                            G940....
      JJ1=JJ2+1                                                       G950....
      JJ2=JJ2+4                                                       G960....
      IF(LOOP.EQ.MLOOP) JJ2=NOBS                                      G970....
      WRITE(K3,5999) (IOBS(JJ),JJ=JJ1,JJ2)                            G980....
 5999 FORMAT(1H1///5X,'O  B  S  E  R  V  A  T  I  O  N     ',         G990....
     1   'N  O  D  E     D  A  T  A'///23X,4(:8X,'NODE ',I5,8X))      G1000...
      WRITE(K3,6000) (UNDERS,JJ=JJ1,JJ2)                              G1010...
 6000 FORMAT(                            23X,4(:8X,  A10   ,  8X))    G1020...
      WRITE(K3,6001) (UNAME(MN),JJ=JJ1,JJ2)                           G1030...
 6001 FORMAT(/1X,'TIME STEP',4X,'TIME(SEC)',4(:2X,'PRESSURE',3X,A13)) G1040...
      DO 6500 ITT=1,ITCNT                                             G1050...
      WRITE(K3,6100) ITOBS(ITT),OBSTIM(ITT),                          G1060...
     1   (POBS(JJ,ITT),UOBS(JJ,ITT),JJ=JJ1,JJ2)                       G1070...
 6100 FORMAT(5X,I5,1X,1PD12.5,8(1X,1PD12.5))                          G1080...
 6500 CONTINUE                                                        G1090...
 7000 CONTINUE                                                        G1100...
      RETURN                                                          G1110...
                                                                      G1120...
                                                                      G1130...
      END                                                             G1140...
      SUBROUTINE       I  N  D  A  T  2       SUTRA - VERSION 1284-2D K10.....
                                                                      K20.....
    * PURPOSE :                                                       K30.....
       TO READ INITIAL CONDITIONS FROM UNIT-55, AND TO                K40.....
       INITIALIZE DATA FOR EITHER WARM OR COLD START OF               K50.....
       THE SIMULATION.                                                K60.....
                                                                      K70.....
      SUBROUTINE INDAT2(PVEC,UVEC,PM1,UM1,UM2,CS1,CS2,CS3,SL,SR,RCIT, K80.....
     1   SW,DSWDP,PBC,IPBC,IPBCT)                                     K90.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                             K100....
```

```
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6                           MODIFIED
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          K110NEW
     1   NSOP,NSOU,NBCN                                                 K120....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        K130NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            K140NEW
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        K150....
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        K160....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,K170....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2       K180....
      DIMENSION PVEC(NN),UVEC(NN),PM1(NN),UM1(NN),UM2(NN),SL(NN),SR(NN),K190....
     1   CS1(NN),CS2(NN),CS3(NN),RCIT(NN),SW(NN),DSWDP(NN),             K200....
     2   PBC(NBCN),IPBC(NBCN)                                           K210....
C                                                                       K220....
C                                                                       K230....
      IF(IREAD) 500,500,620                                             K240....
C.....INPUT INITIAL CONDITIONS FOR WARM START (UNIT-55 DATA)            K250....
  500 READ(K2,510) TSTART,DELTP,DELTU                                   K260....
  510 FORMAT(4G20.10)                                                   K270....
      READ(K2,510) (PVEC(I),I=1,NN)                                     K280....
      READ(K2,510) (UVEC(I),I=1,NN)                                     K290....
      READ(K2,510) (PM1(I),I=1,NN)                                      K300....
      READ(K2,510) (UM1(I),I=1,NN)                                      K310....
      READ(K2,510) (CS1(I),I=1,NN)                                      K320....
      READ(K2,510) (RCIT(I),I=1,NN)                                     K330....
      READ(K2,510) (SW(I),I=1,NN)                                       K340....
      READ(K2,510) (PBC(IPU),IPU=1,NBCN)                                K350....
C     CALL ZERO(CS2,NN,0.0D0)                                           K360....
C     CALL ZERO(CS3,NN,0.0D0)                                           K370....
      CALL ZERO(SL,NN,0.0D0)                                            K380....
      CALL ZERO(SR,NN,0.0D0)                                            K390....
      CALL ZERO(DSWDP,NN,0.0D0)                                         K400....
      DO 550 I=1,NN                                                     K410....
  550 UM2(I)=UM1(I)                                                     K420....
      GOTO 1000                                                         K430....
C                                                                       K440....
C.....INPUT INITIAL CONDITIONS FOR COLD START (UNIT-55 DATA)            K450....
  620 READ(K2,510) TSTART                                               K460....
      READ(K2,510) (PVEC(I),I=1,NN)                                     K470....
      READ(K2,510) (UVEC(I),I=1,NN)                                     K480....
C.....START-UP WITH NO PROJECTIONS BY SETTING BDELP=BDELU=1.D-16        K490....
C     IN PROJECTION FORMULAE FOUND IN SUBROUTINE SUTRA.                 K500....
      DELTP=DELT*1.D-16                                                 K510....
      DELTU=DELT*1.D-16                                                 K520....
C.....INITIALIZE SPECIFIED TIME-VARYING PRESSURES TO INITIAL PRESSURE   K530....
C     VALUES FOR START-UP CALCULATION OF INFLOWS OR OUTFLOWS            K540....
C     (SET QPLITR=0)                                                    K550....
      IF(IPBCT) 680,740,740                                             K560....
  680 DO 730 IP=1,NPBC                                                  K570....
      I=IPBC(IP)                                                        K580....
      IF(I) 700,700,730                                                 K590....
  700 PBC(IP)=PVEC(-I)                                                  K600....
  730 CONTINUE                                                          K610....
C.....INITIALIZE P, U, AND CONSISTENT DENSITY                           K620....
  740 DO 800 I=1,NN                                                     K630....
      PM1(I)=PVEC(I)                                                    K640....
      UM1(I)=UVEC(I)                                                    K650....
      UM2(I)=UVEC(I)                                                    K660....
      RCIT(I)=RHOW0+DRWDU*(UVEC(I)-URHOW0)                              K670....
  800 CONTINUE                                                          K680....
C.....INITIALIZE SATURATION, SW(I)                                      K690....
      CALL ZERO(SW,NN,1.0D0)                                            K700....
      CALL ZERO(DSWDP,NN,0.0D0)                                         K710....
```

```
      IF(IUNSAT.NE.1) GOTO 990                                          K720....
      IUNSAT=3                                                          K730....
      DO 900 I=1,NN                                                     K740....
  900 IF(PVEC(I).LT.0) CALL UNSAT(SW(I),DSWDP(I),RELK,PVEC(I))          K750....
  990 CONTINUE                                                          K760....
      CALL ZERO(CS1,NN,CS)                                              K770....
C     CALL ZERO(CS2,NN,0.0D0)                                           K780....
C     CALL ZERO(CS3,NN,0.0D0)                                           K790....
      CALL ZERO(SL,NN,0.0D0)                                            K800....
      CALL ZERO(SR,NN,0.0D0)                                            K810....
 1000 CONTINUE                                                          K820....
C                                                                       K830....
C.....SET STARTING TIME OF SIMULATION CLOCK, TSEC                       K840....
      TSEC=TSTART                                                       K850....
C                                                                       K860....
C                                                                       K870....
      RETURN                                                            K880....
      END                                                               K890....
C     SUBROUTINE        P  R  I  S  O  L        SUTRA - VERSION 1284-2D L10.....
C                                                                       L20.....
C *** PURPOSE :                                                         L30.....
C ***  TO PRINT PRESSURE AND TEMPERATURE OR CONCENTRATION               L40.....
C ***  SOLUTIONS AND TO OUTPUT INFORMATION ON TIME STEP, ITERATIONS,    L50.....
C ***  SATURATIONS, AND FLUID VELOCITIES.                               L60.....
C                                                                       L70.....
      SUBROUTINE PRISOL(ML,ISTOP,IGOI,PVEC,UVEC,VMAG,VANG,SW)           L80.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               L90.....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8                     MODIFIED
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          L100NEW
     1   NSOP,NSOU,NBCN                                                 L110....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        L120NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            L130NEW
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        L140....
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        L150....
      COMMON/ITERAT/ RPM,RPMAX,RUM,RUMAX,ITER,ITRMAX,IPWORS,IUWORS,     L160....
     1                 ICON,ITRMX2,OMEGA,RPMX2,RUMX2                    NEW
      COMMON/KPRINT/ KCOORD,KELINF,KINCID,KPLOTP,KPLOTU,KVEL,KBUDG       L170....
      DIMENSION PVEC(NN),UVEC(NN),VMAG(NE),VANG(NE),SW(NN)              L180....
C                                                                       L190....
C.....OUTPUT MAJOR HEADINGS FOR CURRENT TIME STEP                       L200....
      IF(IT.GT.0.OR.ISSFLO.EQ.2.OR.ISSTRA.EQ.1) GOTO 100                L210....
      WRITE(K3,60)                                                      L220....
   60 FORMAT(1H1////11X,'I N I T I A L   C O N D I T I O N S',          L230....
     1          /11X,'___________________________________')             L240....
      IF(IREAD.EQ.-1) WRITE(K3,65)                                      L250....
   65 FORMAT(//11X,'INITIAL CONDITIONS RETRIEVED FROM STORAGE ',        L260....
     1   'ON UNIT 55.')                                                 L270....
      GOTO 500                                                          L280....
C                                                                       L290....
  100 IF(IGOI.NE.0.AND.ISTOP.EQ.0) WRITE(K3,150) ITER,IT                L300....
  150 FORMAT(////////11X,'ITERATION ',I3,' SOLUTION FOR TIME STEP ',I4) L310....
C                                                                       L320....
      IF(ISTOP.EQ.-1) WRITE(K3,250) IT,ITER                             L330....
  250 FORMAT(1H1//11X,'SOLUTION FOR TIME STEP ',I4,                     L340....
     1   ' NOT CONVERGED AFTER ',I3,' ITERATIONS.')                     L350....
C                                                                       L360....
      IF(ISTOP.GE.0) WRITE(K3,350) IT                                   L370....
  350 FORMAT(1H1//11X,'RESULTS FOR TIME STEP ',I4/                      L380....
     1   11X,'_____________________________')                           L390....
      IF(ITRMAX.EQ.1) GOTO 500                                          L400....
      IF(ISTOP.GE.0.AND.IT.GT.0) WRITE(K3,355) ITER                     L410....
      IF(IT.EQ.0.AND.ISTOP.GE.0.AND.ISSFLO.EQ.2) WRITE(K3,355) ITER     L420....
```

```
  355 FORMAT(11X,'(AFTER ',I3,' ITERATIONS) :')
      WRITE(K3,450) RPM,IPWORS,RUM,IUWORS
  450 FORMAT(//11X,'MAXIMUM P CHANGE FROM PREVIOUS ITERATION ',
     1   1PD14.5,' AT NODE ',I5/11X,'MAXIMUM U CHANGE FROM PREVIOUS ',
     2   'ITERATION ',1PD14.5,' AT NODE ',I5)
C
  500 IF(IT.EQ.0.AND.ISSFLO.EQ.2) GOTO 680
      IF(ISSTRA.EQ.1) GOTO 800
      WRITE(K3,550) DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,
     1   TMONTH,TYEAR
  550 FORMAT(///11X,'TIME INCREMENT :',T27,1PD15.4,' SECONDS'//11X,
     1   'ELAPSED TIME :',T27,1PD15.4,' SECONDS',/T27,1PD15.4,' MINUTES'
     2   /T27,1PD15.4,' HOURS'/T27,1PD15.4,' DAYS'/T27,1PD15.4,' WEEKS'/
     3   T27,1PD15.4,' MONTHS'/T27,1PD15.4,' YEARS')
C
C.....OUTPUT PRESSURES FOR TRANSIENT FLOW SOLUTION (AND POSSIBLY,
C         SATURATION AND VELOCITY)
      IF(ML.EQ.2.AND.ISTOP.GE.0) GOTO 700
      IF(ISSFLO.GT.0) GOTO 700
      WRITE(K3,650) (I,PVEC(I),I=1,NN)
  650 FORMAT(///11X,'P   R   E   S   S   U   R   E'//8X,6('NODE',17X)/
     1   (7X,6(1X,I4,1X,1PD15.8)))
      IF(IUNSAT.NE.0) WRITE(K3,651) (I,SW(I),I=1,NN)
  651 FORMAT(///11X,'S   A   T   U   R   A   T   I   O   N'//8X,6('NODE',17X)/
     1   (7X,6(1X,I4,1X,1PD15.8)))
      IF(KVEL.EQ.1.AND.IT.GT.0) WRITE(K3,655) (L,VMAG(L),L=1,NE)
      IF(KVEL.EQ.1.AND.IT.GT.0) WRITE(K3,656) (L,VANG(L),L=1,NE)
  655 FORMAT(///11X,'F  L  U  I  D              V  E  L  O  C  I  T  Y'//
     1   11X,'M A G N I T U D E   AT CENTROID OF ELEMENT'//
     2   5X,6('ELEMENT',14X)/(7X,6(1X,I4,1X,1PD15.8)))
  656 FORMAT(///11X,'F  L  U  I  D              V  E  L  O  C  I  T  Y'//
     1   11X,'A N G L E   IN DEGREES FROM +X-AXIS TO FLOW DIRECTION ',
     2   'AT CENTROID OF ELEMENT'//
     3   5X,6('ELEMENT',14X)/(7X,6(1X,I4,1X,1PD15.8)))
      GOTO 700
C
C.....OUTPUT PRESSURES FOR STEADY-STATE FLOW SOLUTION
  680 WRITE(K3,690) (I,PVEC(I),I=1,NN)
  690 FORMAT(///11X,'S   T   E   A   D   Y   -   S   T   A   T   E       P   R   E   S',
     1   '   S   U   R   E'//8X,6('NODE',17X)/(7X,6(1X,I4,1X,1PD15.8)))
      IF(IUNSAT.NE.0) WRITE(K3,651) (I,SW(I),I=1,NN)
      GOTO 1000
C
C.....OUTPUT CONCENTRATIONS OR TEMPERATURES FOR
C         TRANSIENT TRANSPORT SOLUTION
  700 IF(ML.EQ.1.AND.ISTOP.GE.0) GOTO 1000
      IF(ME) 720,720,730
  720 WRITE(K3,725) (I,UVEC(I),I=1,NN)
  725 FORMAT(///11X,'C   O   N   C   E   N   T   R   A   T   I   O   N'//8X,
     1   6('NODE',17X)/(7X,6(1X,I4,1X,1PD15.8)))
      GOTO 900
  730 WRITE(K3,735) (I,UVEC(I),I=1,NN)
  735 FORMAT(///11X,'T   E   M   P   E   R   A   T   U   R   E'//8X,6('NODE',17X)/
     1   (7X,6(1X,I4,1X,F15.9)))
      GOTO 900
C
C.....OUTPUT CONCENTRATIONS OR TEMPERATURES FOR
C         STEADY-STATE TRANSPORT SOLUTION
  800 IF(ME) 820,820,830
  820 WRITE(K3,825) (I,UVEC(I),I=1,NN)
  825 FORMAT(///11X,'S   T   E   A   D   Y   -   S   T   A   T   E       C   O   N   C',
     1   '   E   N   T   R   A   T   I   O   N'//8X,6('NODE',17X)/
```

```
     2    (7X,6(1X,I4,1X,1PD15.8)))                                      L1050...
      GOTO 900                                                           L1060...
  830 WRITE(K3,835) (I,UVEC(I),I=1,NN)                                   L1070...
  835 FORMAT(///11X,'S  T  E  A  D  Y  -  S  T  A  T  E    T  E  M  P',  L1080...
     1    '  E  R  A  T  U  R  E'//8X,6('NODE',17X)/                     L1090...
     2    (7X,6(1X,I4,1X,F15.9)))                                        L1100...
C                                                                        L1110...
C.....OUTPUT VELOCITIES FOR STEADY-STATE FLOW SOLUTION                   L1120...
  900 IF(ISSFLO.NE.2.OR.IT.NE.1.OR.KVEL.NE.1) GOTO 1000                  L1130...
      WRITE(K3,925) (L,VMAG(L),L=1,NE)                                   L1140...
      WRITE(K3,950) (L,VANG(L),L=1,NE)                                   L1150...
  925 FORMAT(///11X,'S  T  E  A  D  Y  -  S  T  A  T  E      ',          L1160...
     1    'F  L  U  I  D     V  E  L  O  C  I  T  Y'//                   L1170...
     2    11X,'M A G N I T U D E   AT CENTROID OF ELEMENT'//             L1180...
     3    5X,6('ELEMENT',14X)/(7X,6(1X,I4,1X,1PD15.8)))                  L1190...
  950 FORMAT(///11X,'S  T  E  A  D  Y  -  S  T  A  T  E      ',          L1200...
     1    'F  L  U  I  D     V  E  L  O  C  I  T  Y'//                   L1210...
     2    11X,'A N G L E   IN DEGREES FROM +X-AXIS TO FLOW DIRECTION ',  L1220...
     3    'AT CENTROID OF ELEMENT'//                                     L1230...
     4    5X,6('ELEMENT',14X)/(7X,6(1X,I4,1X,1PD15.8)))                  L1240...
C                                                                        L1250...
 1000 RETURN                                                             L1260...
C                                                                        L1270...
      END                                                                L1280...
C     SUBROUTINE        Z  E  R  O             SUTRA - VERSION 1284-2D   M10.....
C                                                                        M20.....
C *** PURPOSE :                                                          M30.....
C ***  TO FILL AN ARRAY WITH A CONSTANT VALUE.                           M40.....
C                                                                        M50.....
      SUBROUTINE ZERO(A,IADIM,FILL)                                      M60.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                M70.....
      DIMENSION A(IADIM)                                                 M80.....
C                                                                        M90.....
C.....FILL ARRAY A WITH VALUE IN VARIABLE 'FILL'                         M100....
      DO 10 I=1,IADIM                                                    M110....
   10 A(I)=FILL                                                          M120....
C                                                                        M130....
C                                                                        M140....
      RETURN                                                             M150....
      END                                                                M160....
C     SUBROUTINE        B  C  T  I  M  E       SUTRA - VERSION 1284-2D   N10.....
C                                                                        N20.....
C *** PURPOSE :                                                          N30.....
C ***  USER-PROGRAMMED SUBROUTINE WHICH ALLOWS THE USER TO SPECIFY:      N40.....
C ***   (1) TIME-DEPENDENT SPECIFIED PRESSURES AND TIME-DEPENDENT        N50.....
C ***       CONCENTRATIONS OR TEMPERATURES OF INFLOWS AT THESE POINTS    N60.....
C ***   (2) TIME-DEPENDENT SPECIFIED CONCENTRATIONS OR TEMPERATURES      N70.....
C ***   (3) TIME-DEPENDENT FLUID SOURCES AND CONCENTRATIONS              N80.....
C ***       OR TEMPERATURES OF INFLOWS AT THESE POINTS                   N90.....
C ***   (4) TIME-DEPENDENT ENERGY OR SOLUTE MASS SOURCES                 N100....
C                                                                        N110....
      SUBROUTINE BCTIME(IPBC,PBC,IUBC,UBC,QIN,UIN,QUIN,IQSOP,IQSOU,      N120....
     1    IPBCT,IUBCT,IQSOPT,IQSOUT,UVEC)                                N130NEW
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                N140....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8                      MODIFIED
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,          NEW
     1    NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            NEW
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,           N150NEW
     1    NSOP,NSOU,NBCN                                                 N160....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,         N170....
     1    TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        N180....
      DIMENSION IPBC(NBCN),PBC(NBCN),IUBC(NBCN),UBC(NBCN),               N190....
```

```
     1   QIN(NN),UIN(NN),QUIN(NN),IQSOP(NSOP),IQSOU(NSOU)              N200....
      DIMENSION UVEC(NN)                                                NEW
C                                                                       N210....
C.....DEFINITION OF REQUIRED VARIABLES                                  N220....
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . N230....
C     NN = EXACT NUMBER OF NODES IN MESH                                N240....
C     NPBC = EXACT NUMBER OF SPECIFIED PRESSURE NODES                   N250....
C     NUBC = EXACT NUMBER OF SPECIFIED CONCENTRATION                    N260....
C            OR TEMPERATURE NODES                                       N270....
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . N280....
C     IT = NUMBER OF CURRENT TIME STEP                                  N290....
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . N300....
C     TSEC = TIME AT END OF CURRENT TIME STEP IN SECONDS                N310....
C     TMIN = TIME AT END OF CURRENT TIME STEP IN MINUTES                N320....
C     THOUR = TIME AT END OF CURRENT TIME STEP IN HOURS                 N330....
C     TDAY = TIME AT END OF CURRENT TIME STEP IN DAYS                   N340....
C     TWEEK = TIME AT END OF CURRENT TIME STEP IN WEEKS                 N350....
C     TMONTH = TIME AT END OF CURRENT TIME STEP IN MONTHS               N360....
C     TYEAR = TIME AT END OF CURRENT TIME STEP IN YEARS                 N370....
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . N380....
C     PBC(IP) = SPECIFIED PRESSURE VALUE AT IP(TH) SPECIFIED            N390....
C               PRESSURE NODE                                           N400....
C     UBC(IP) = SPECIFIED CONCENTRATION OR TEMPERATURE VALUE OF ANY     N410....
C               INFLOW OCCURRING AT IP(TH) SPECIFIED PRESSURE NODE      N420....
C     IPBC(IP) = ACTUAL NODE NUMBER OF IP(TH) SPECIFIED PRESSURE NODE   N430....
C                [WHEN NODE NUMBER I=IPBC(IP) IS NEGATIVE (I<0),        N440....
C                VALUES MUST BE SPECIFIED FOR PBC AND UBC.]             N450....
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . N460....
C     UBC(IUP) = SPECIFIED CONCENTRATION OR TEMPERATURE VALUE AT        N470....
C                IU(TH) SPECIFIED CONCENTRATION OR TEMPERATURE NODE     N480....
C                (WHERE IUP=IU+NPBC)                                    N490....
C     IUBC(IUP) = ACTUAL NODE NUMBER OF IU(TH) SPECIFIED CONCENTRATION  N500....
C                 OR TEMPERATURE NODE (WHERE IUP=IU+NPBC)               N510....
C                 [WHEN NODE NUMBER I=IUBC(IU) IS NEGATIVE (I<0),       N520....
C                 A VALUE MUST BE SPECIFIED FOR UBC.]                   N530....
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . N540....
C     IQSOP(IQP) = NODE NUMBER OF IQP(TH) FLUID SOURCE NODE.            N550....
C                  [WHEN NODE NUMBER I=IQSOP(IQP) IS NEGATIVE (I<0),    N560....
C                  VALUES MUST BE SPECIFIED FOR QIN AND UIN.]           N570....
C     QIN(-I) = SPECIFIED FLUID SOURCE VALUE AT NODE (-I)               N580....
C     UIN(-I) = SPECIFIED CONCENTRATION OR TEMPERATURE VALUE OF ANY     N590....
C               INFLOW OCCURRING AT FLUID SOURCE NODE (-I)              N600....
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . N610....
C     IQSOU(IQU) = NODE NUMBER OF IQU(TH) ENERGY OR                     N620....
C                  SOLUTE MASS SOURCE NODE                              N630....
C                  [WHEN NODE NUMBER I=IQSOU(IQU) IS NEGATIVE (I<0),    N640....
C                  A VALUE MUST BE SPECIFIED FOR QUIN.]                 N650....
C     QUIN(-I) = SPECIFIED ENERGY OR SOLUTE MASS SOURCE VALUE           N660....
C                AT NODE (-I)                                           N670....
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . N680....
C                                                                       N690....
C                                                                       N700....
C.....NSOPI IS ACTUAL NUMBER OF FLUID SOURCE NODES                      N710....
      NSOPI=NSOP-1                                                      N720....
C.....NSOUI IS ACTUAL NUMBER OF ENERGY OR SOLUTE MASS SOURCE NODES      N730....
      NSOUI=NSOU-1                                                      N740....
C                                                                       N750....
C                                                                       N760....
C                                                                       N770....
C                                                                       N780....
C                                                                       N790....
C                                                                       N800....
```

```
      IF(IPBCT) 50,240,240                                              N810....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N820....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N830....
C.....SECTION (1):  SET TIME-DEPENDENT SPECIFIED PRESSURES OR           N840....
C     CONCENTRATIONS (TEMPERATURES) OF INFLOWS AT SPECIFIED             N850....
C     PRESSURE NODES                                                    N860....
C                                                                       N870....
   50 CONTINUE                                                          N880....
      DO 200 IP=1,NPBC                                                  N890....
      I=IPBC(IP)                                                        N900....
      IF(I) 100,200,200                                                 N910....
  100 CONTINUE                                                          N920....
C     NOTE : A FLOW AND TRANSPORT SOLUTION MUST OCCUR FOR ANY           N930....
C            TIME STEP IN WHICH PBC( ) CHANGES.                         N940....
C     PBC(IP) =  ((          ))                                         N950....
C     UBC(IP) =  ((          ))                                         N960....
  200 CONTINUE                                                          N970....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N980....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N990....
C                                                                       N1000...
C                                                                       N1010...
C                                                                       N1020...
C                                                                       N1030...
C                                                                       N1040...
C                                                                       N1050...
  240 IF(IUBCT) 250,440,440                                             N1060...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1070...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1080...
C.....SECTION (2):  SET TIME-DEPENDENT SPECIFIED                        N1090...
C     CONCENTRATIONS (TEMPERATURES)                                     N1100...
C                                                                       N1110...
  250 CONTINUE                                                          N1120...
      DO 400 IU=1,NUBC                                                  N1130...
      IUP=IU+NPBC                                                       N1140...
      I=IUBC(IUP)                                                       N1150...
      IF(I) 300,400,400                                                 N1160...
  300 CONTINUE                                                          N1170...
C     NOTE : A TRANSPORT SOLUTION MUST OCCUR FOR ANY TIME STEP          N1180...
C            IN WHICH UBC( ) CHANGES.  IN ADDITION, IF FLUID PROPERTIES N1190...
C            ARE SENSITIVE TO 'U' THEN A FLOW SOLUTION MUST OCCUR AS WELN1200...
C     UBC(IUP) =    ((          ))                                      N1210...
  400 CONTINUE                                                          N1220...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1230...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1240...
C                                                                       N1250...
C                                                                       N1260...
C                                                                       N1270...
C                                                                       N1280...
C                                                                       N1290...
C                                                                       N1300...
  440 IF(IQSOPT) 450,640,640                                            N1310...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1320...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1330...
C.....SECTION (3):  SET TIME-DEPENDENT FLUID SOURCES/SINKS,             N1340...
C      OR CONCENTRATIONS (TEMPERATURES) OF SOURCE FLUID                 N1350...
C                                                                       N1360...
  450 CONTINUE                                                          N1370...
C                                                                       NEW
C                *** THE FOLLOWING MODIFICATION IS MADE TO              NEW
C                    TURN OF WITHDRAWL WELLS WHEN AVERAGE               NEW
C                    FLUID CONCENTRATION IS GREATER THAN CMAX           NEW
C                                                                       NEW
```

```
C                                                                       NEW
C                    FIRST CALCULATE VOLUME AVERAGED CONCENTRATION      NEW
      CMAX=0.9980                                                       NEW
      CBAR=0.0D0                                                        NEW
      QTOT=0.0D0                                                        NEW
      DO 605 IQP=1,NSOPI                                                NEW
      I=IQSOP(IQP)                                                      NEW
      I=IABS(I)                                                         NEW
      CBAR=CBAR+UVEC(I)*QIN(I)                                          NEW
      QTOT=QTOT+QIN(I)                                                  NEW
  605 CONTINUE                                                          NEW
      CBAR=CBAR/QTOT                                                    NEW
      WRITE (K6,606) TDAY,CBAR                                          NEW
  606 FORMAT(1H0,10X,'VOLUME AVERAGED SOLUTE CONCENTRATION',            NEW
     +' AT TIME STEP ',F10.2,' =',F10.4)                                NEW
      IF (CBAR.LE.CMAX) GO TO 610                                       NEW
C                                                                       NEW
C               CBAR EXCEEDS CMAX, TURN OFF THE WELLS AND               NEW
C               RESET IQSOPT SO PROGRAM DOES NOT RETURN HERE            NEW
      IQSOPT=+1                                                         NEW
      WRITE(K3,608) DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR        NEW
  608 FORMAT(///11X,'TIME INCREMENT :',T27,1PD15.4,' SECONDS'//11X,     NEW
     1   'ELAPSED TIME :',T27,1PD15.4,' SECONDS',/T27,1PD15.4,' MINUTES'NEW
     2   /T27,1PD15.4,' HOURS'/T27,1PD15.4,' DAYS'/T27,1PD15.4,' WEEKS'/NEW
     3   T27,1PD15.4,' MONTHS'/T27,1PD15.4,' YEARS')                    NEW
      CALL RUNDAT(TDAY)                                                 NEW
  607 FORMAT(1H0,10X,'CONCENTRATION EXCEEDS MAXIMUM VALUE (',F10.4,     NEW
     1 ')'/1H ,10X,'WELLS AT R=0 ARE TURNED OFF '/)                     NEW
      DO 600 IQP=1,NSOPI                                                N1380...
      I=IQSOP(IQP)                                                      N1390...
      I=IABS(I)                                                         NEW
      QIN(I) =   0.0D0                                                  N1440...
      UIN(I) =   0.0D0                                                  N1470...
  600 CONTINUE                                                          N1480...
C                                                                       NEW
C              RESET FLAG TO KEEP TIME STEP SIZE CONSTANT (MODIFIED     NEW
C              VERSION ONLY)                                            NEW
      ITCYC=0                                                           NEW
C                                                                       NEW
  610 CONTINUE                                                          NEW
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1490...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1500...
C                                                                       N1510...
C                                                                       N1520...
C                                                                       N1530...
C                                                                       N1540...
C                                                                       N1550...
C                                                                       N1560...
  640 IF(IQSOUT) 650,840,840                                            N1570...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1580...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1590...
C.....SECTION (4):  SET TIME-DEPENDENT SOURCES/SINKS                    N1600...
C     OF SOLUTE MASS OR ENERGY                                          N1610...
C                                                                       N1620...
  650 CONTINUE                                                          N1630...
      DO 800 IQU=1,NSOUI                                                N1640...
      I=IQSOU(IQU)                                                      N1650...
      IF(I) 700,800,800                                                 N1660...
  700 CONTINUE                                                          N1670...
```

```
C     NOTE : A TRANSPORT SOLUTION MUST OCCUR FOR ANY                    N1680...
C                 TIME STEP IN WHICH QUIN( ) CHANGES.                   N1690...
C     QUIN(-I) =    ((           ))                                     N1700...
  800 CONTINUE                                                          N1710...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1720...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - N1730...
C                                                                       N1740...
C                                                                       N1750...
C                                                                       N1760...
C                                                                       N1770...
C                                                                       N1780...
C                                                                       N1790...
  840 CONTINUE                                                          N1800...
C                                                                       N1810...
      RETURN                                                            N1820...
      END                                                               N1830...
C     SUBROUTINE        A  D  S  O  R  B        SUTRA - VERSION 1284-2D O10.....
C                                                                       O20.....
C *** PURPOSE :                                                         O30.....
C ***  TO CALCULATE VALUES OF EQUILIBRIUM SORPTION PARAMETERS FOR       O40.....
C ***  LINEAR, FREUNDLICH, AND LANGMUIR MODELS.                         O50.....
C                                                                       O60.....
      SUBROUTINE ADSORB(CS1,CS2,CS3,SL,SR,U)                            O70.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               O80.....
      CHARACTER*10 ADSMOD                                               O90.....
      COMMON/MODSOR/ ADSMOD                                             O100....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          O110NEW
     1   NSOP,NSOU,NBCN                                                 O120....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,O130....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2       O140....
      DIMENSION CS1(NN),CS2(NN),CS3(NN),SL(NN),SR(NN),U(NN)             O150....
C                                                                       O160....
C.....NOTE THAT THE CONCENTRATION OF ADSORBATE, CS(I), IS GIVEN BY:     O170....
C     CS(I) = SL(I)*U(I) + SR(I)                                        O180....
C                                                                       O190....
C.....NO SORPTION                                                       O200....
      IF(ADSMOD.NE.'NONE      ') GOTO 450                               O210....
      DO 250 I=1,NN                                                     O220....
      CS1(I)=0.D0                                                       O230....
      CS2(I)=0.D0                                                       O240....
      CS3(I)=0.D0                                                       O250....
      SL(I)=0.D0                                                        O260....
      SR(I)=0.D0                                                        O270....
  250 CONTINUE                                                          O280....
      GOTO 2000                                                         O290....
C                                                                       O300....
C.....LINEAR SORPTION MODEL                                             O310....
  450 IF(ADSMOD.NE.'LINEAR    ') GOTO 700                               O320....
      DO 500 I=1,NN                                                     O330....
      CS1(I)=CHI1*RHOW0                                                 O340....
      CS2(I)=0.D0                                                       O350....
      CS3(I)=0.D0                                                       O360....
      SL(I)=CHI1*RHOW0                                                  O370....
      SR(I)=0.D0                                                        O380....
      CONTINUE                                                          O390....
      GOTO 2000                                                         O400....
C                                                                       O410....
C.....FREUNDLICH SORPTION MODEL                                         O420....
      IF(ADSMOD.NE.'FREUNDLICH') GOTO 950                               O430....
      CHCH=CHI1/CHI2                                                    O440....
      DCHI2=1.D0/CHI2                                                   O450....
      RH2=RHOW0**DCHI2                                                  O460....
```

```
      CHI2F=((1.D0-CHI2)/CHI2)                                          0470....
      CH12=CHI1**DCHI2                                                  0480....
      DO 750 I=1,NN                                                     0490....
      IF(U(I)) 720,720,730                                              0500....
  720 UCH=1.0D0                                                         0510....
      GOTO 740                                                          0520....
  730 UCH=U(I)**CHI2F                                                   0530....
  740 RU=RH2*UCH                                                        0540....
      CS1(I)=CHCH*RU                                                    0550....
      CS2(I)=0.D0                                                       0560....
      CS3(I)=0.D0                                                       0570....
      SL(I)=CH12*RU                                                     0580....
      SR(I)=0.D0                                                        0590....
  750 CONTINUE                                                          0600....
      GOTO 2000                                                         0610....
C                                                                       0620....
C.....LANGMUIR SORPTION MODEL                                            0630....
  950 IF(ADSMOD.NE.'LANGMUIR  ') GOTO 2000                              0640....
      DO 1000 I=1,NN                                                    0650....
      DD=1.D0+CHI2*RHOW0*U(I)                                           0660....
      CS1(I)=(CHI1*RHOW0)/(DD*DD)                                       0670....
      CS2(I)=0.D0                                                       0680....
      CS3(I)=0.D0                                                       0690....
      SL(I)=CS1(I)                                                      0700....
      SR(I)=CS1(I)*CHI2*RHOW0*U(I)*U(I)                                 0710....
 1000 CONTINUE                                                          0720....
C                                                                       0730....
 2000 RETURN                                                            0740....
      END                                                               0750....
C     SUBROUTINE        E  L  E  M  E  N           SUTRA - VERSION 1284-2D P10.....
C                                                                       P20.....
C *** PURPOSE :                                                         P30.....
C ***  TO CONTROL AND CARRY OUT ALL CALCULATIONS FOR EACH ELEMENT BY    P40.....
C ***  OBTAINING ELEMENT INFORMATION FROM THE BASIS FUNCTION ROUTINE,   P50.....
C ***  CARRYING OUT GAUSSIAN INTEGRATION OF FINITE ELEMENT INTEGRALS,   P60.....
C ***  AND SENDING RESULTS OF ELEMENT INTEGRATIONS TO GLOBAL ASSEMBLY   P70.....
C ***  ROUTINE.  ALSO CALCULATES VELOCITY AT EACH ELEMENT CENTROID FOR  P80.....
C ***  PRINTED OUTPUT.                                                  P90.....
C                                                                       P100....
      SUBROUTINE ELEMEN(ML,IN,X,Y,THICK,PITER,UITER,RCIT,RCITM1,POR,    P110....
     1   ALMAX,ALMIN,ATAVG,PERMXX,PERMXY,PERMYX,PERMYY,PANGLE,          P120....
     2   VMAG,VANG,VOL,PMAT,PVEC,UMAT,UVEC,GXSI,GETA,PVEL,CWRK)          P130NEW
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               P140....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8                     MODIFIED
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          P150NEW
     1   NSOP,NSOU,NBCN                                                 P160....
      COMMON/TENSOR/ GRAVX,GRAVY                                        P170....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,P180....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2       P190....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        P200....
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        P210....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,         P220NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            P230NEW
      COMMON/KPRINT/ KNODAL,KELMNT,KINCID,KPLOTP,KPLOTU,KVEL,KBUDG      P240....
      DIMENSION IN(NIN),X(NN),Y(NN),THICK(NN),PITER(NN),                P250....
     1   UITER(NN),RCIT(NN),RCITM1(NN),POR(NN),PVEL(NN)                 P260....
      DIMENSION PERMXX(NE),PERMXY(NE),PERMYX(NE),PERMYY(NE),PANGLE(NE), P270....
     1   ALMAX(NE),ALMIN(NE),ATAVG(NE),VMAG(NE),VANG(NE),               P280....
     2   GXSI(NE,4),GETA(NE,4)                                          P290....
      DIMENSION VOL(NN),PMAT(NN,NBIP),PVEC(NN),UMAT(NN,NBIS),UVEC(NN)   P300NEW
      DIMENSION CWRK(NN)                                                NEW
      DIMENSION BFLOWE(4,4),DFLOWE(4),BTRANE(4,4),DTRANE(4,4),VOLE(4)   P310....
```

```
      DIMENSION F(4,4),W(4,4),DET(4),DFDXG(4,4),DFDYG(4,4),             P320....
     1   DWDXG(4,4),DWDYG(4,4)                                          P330....
      DIMENSION SWG(4),RHOG(4),VISCG(4),PORG(4),VXG(4),VYG(4),          P340....
     1   RELKG(4),RGXG(4),RGYG(4),VGMAG(4),THICKG(4)                    P350....
      DIMENSION RXXG(4),RXYG(4),RYXG(4),RYYG(4)                         P360....
      DIMENSION BXXG(4),BXYG(4),BYXG(4),BYYG(4),                        P370....
     1   EXG(4),EYG(4)                                                  P380....
      DIMENSION GXLOC(4),GYLOC(4)                                       P390....
      DATA GLOC/0.577350269189626D0/                                    P400....
      DATA INTIM/0/,ISTOP/0/,GXLOC/-1.D0,1.D0,1.D0,-1.D0/,              P410....
     1   GYLOC/-1.D0,-1.D0,1.D0,1.D0/                                   P420....
C                                                                       P430....
C.....DECIDE WHETHER TO CALCULATE CENTROID VELOCITIES ON THIS CALL      P440....
      IVPRNT=0                                                          P450....
      IF(MOD(IT,NPRINT).EQ.0.AND.ML.NE.2.AND.IT.NE.0) IVPRNT=1          P460....
      IF(IT.EQ.1) IVPRNT=1                                              P470....
      KVPRNT=IVPRNT+KVEL                                                P480....
C                                                                       P490....
C.....ON FIRST TIME STEP, PREPARE GRAVITY VECTOR COMPONENTS,            P500....
C          GXSI AND GETA, FOR CONSISTENT VELOCITIES,                    P510....
C          AND CHECK ELEMENT SHAPES                                     P520....
      IF(INTIM) 100,100,2000                                            P530....
  100 INTIM=1                                                           P540....
C.....LOOP THROUGH ALL ELEMENTS TO OBTAIN THE JACOBIAN                  P550....
C          AT EACH OF THE FOUR NODES IN EACH ELEMENT                    P560....
      DO 1000 L=1,NE                                                    P570....
       DO 500 IL=1,4                                                    P580....
        XLOC=GXLOC(IL)                                                  P590....
        YLOC=GYLOC(IL)                                                  P600....
        CALL BASIS2(0000,L,XLOC,YLOC,IN,X,Y,F(1,IL),W(1,IL),DET(IL),    P610....
     1     DFDXG(1,IL),DFDYG(1,IL),DWDXG(1,IL),DWDYG(1,IL),             P620....
     2     PITER,UITER,PVEL,POR,THICK,THICKG(IL),VXG(IL),VYG(IL),       P630....
     3     SWG(IL),RHOG(IL),VISCG(IL),PORG(IL),VGMAG(IL),RELKG(IL),     P640....
     4     PERMXX,PERMXY,PERMYX,PERMYY,CJ11,CJ12,CJ21,CJ22,             P650....
     5     GXSI,GETA,RCIT,RCITM1,RGXG(IL),RGYG(IL))                     P660....
        GXSI(L,IL)=CJ11*GRAVX+CJ12*GRAVY                                P670....
        GETA(L,IL)=CJ21*GRAVX+CJ22*GRAVY                                P680....
C.....CHECK FOR NEGATIVE- OR ZERO-AREA ERRORS IN ELEMENT SHAPES         P690....
        IF(DET(IL)) 200,200,500                                         P700....
  200   ISTOP=ISTOP+1                                                   P710....
        WRITE(K3,400) IN((L-1)*4+IL),L,DET(IL)                          P720....
  400   FORMAT(11X,'THE DETERMINANT OF THE JACOBIAN AT GAUSS POINT ',I4,P730....
     1      ' IN ELEMENT ',I4,' IS NEGATIVE OR ZERO, ',1PE15.7)         P740....
  500  CONTINUE                                                         P750....
 1000 CONTINUE                                                          P760....
C                                                                       P770....
      IF(ISTOP.EQ.0) GOTO 2000                                          P780....
      WRITE(K3,1500)                                                    P790....
 1500 FORMAT(//////11X,'SOME ELEMENTS HAVE INCORRECT GEOMETRY.'         P800....
     1   //11X,'PLEASE CHECK THE NODE COORDINATES AND ',                P810....
     2   'INCIDENCE LIST, MAKE CORRECTIONS, AND THEN RERUN.'////////    P820....
     3   11X,'S I M U L A T I O N   H A L T E D'/                       P830....
     4   11X,'___________________   ___________')                       P840....
      ENDFILE(K3)                                                       P850....
      STOP                                                              P860....
C                                                                       P870....
C.....LOOP THROUGH ALL ELEMENTS TO CARRY OUT SPATIAL INTEGRATION        P880....
C          OF FLUX TERMS IN P AND/OR U EQUATIONS                        P890....
 2000 IF(IUNSAT.NE.0) IUNSAT=2                                          P900....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P910....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P920....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P930....
      DO 9999 L=1,NE                                                    P940....
```

```
      XIX=-1.D0                                                         P950....
      YIY=-1.D0                                                         P960....
      KG=0                                                              P970....
C.....OBTAIN BASIS FUNCTION AND RELATED INFORMATION AT EACH OF          P980....
C       FOUR GAUSS POINTS IN THE ELEMENT                                P990....
      DO 2200 IYL=1,2                                                   P1000...
       DO 2100 IXL=1,2                                                  P1010...
        KG=KG+1                                                         P1020...
        XLOC=XIX*GLOC                                                   P1030...
        YLOC=YIY*GLOC                                                   P1040...
      CALL BASIS2(0001,L,XLOC,YLOC,IN,X,Y,F(1,KG),W(1,KG),DET(KG),      P1050...
     1    DFDXG(1,KG),DFDYG(1,KG),DWDXG(1,KG),DWDYG(1,KG),              P1060...
     2    PITER,UITER,PVEL,POR,THICK,THICKG(KG),VXG(KG),VYG(KG),        P1070...
     3    SWG(KG),RHOG(KG),VISCG(KG),PORG(KG),VGMAG(KG),RELKG(KG),      P1080...
     4    PERMXX,PERMXY,PERMYX,PERMYY,CJ11,CJ12,CJ21,CJ22,              P1090...
     5    GXSI,GETA,RCIT,RCITM1,RGXG(KG),RGYG(KG))                      P1100...
 2100   XIX=-XIX                                                        P1110...
 2200  YIY=-YIY                                                         P1120...
C                                                                       P1130...
C.....CALCULATE VELOCITY AT ELEMENT CENTROID WHEN REQUIRED              P1140...
      IF(KVPRNT-2) 3000,2300,3000                                       P1150...
 2300 AXSUM=0.0D0                                                       P1160...
      AYSUM=0.0D0                                                       P1170...
      DO 2400 KG=1,4                                                    P1180...
       AXSUM=AXSUM+VXG(KG)                                              P1190...
 2400  AYSUM=AYSUM+VYG(KG)                                              P1200...
      VMAG(L)=DSQRT(AXSUM*AXSUM+AYSUM*AYSUM)/4.0D0                      P1210...
      IF(AXSUM) 2500,2700,2800                                          P1220...
 2500 AYX=AYSUM/AXSUM                                                   P1230...
      VANG(L)=DATAN(AYX)/1.745329D-02                                   P1240...
      IF(AYSUM.LT.0.0D0) GOTO 2600                                      P1250...
      VANG(L)=VANG(L)+180.0D0                                           P1260...
      GOTO 3000                                                         P1270...
 2600 VANG(L)=VANG(L)-180.0D0                                           P1280...
      GOTO 3000                                                         P1290...
 2700 VANG(L)=90.0D0                                                    P1300...
      IF(AYSUM.LT.0.0D0) VANG(L)=-90.0D0                                P1310...
      GOTO 3000                                                         P1320...
 2800 AYX=AYSUM/AXSUM                                                   P1330...
      VANG(L)=DATAN(AYX)/1.745329D-02                                   P1340...
C                                                                       P1350...
C.....INCLUDE MESH THICKNESS IN NUMERICAL INTEGRATION                   P1360...
 3000 DO 3300 KG=1,4                                                    P1370...
 3300  DET(KG)=THICKG(KG)*DET(KG)                                       P1380...
C                                                                       P1390...
C.....CALCULATE PARAMETERS FOR FLUID MASS BALANCE AT GAUSS POINTS       P1400...
      IF(ML-1) 3400,3400,6100                                           P1410...
 3400 SWTEST=0.D0                                                       P1420...
      DO 4000 KG=1,4                                                    P1430...
       SWTEST=SWTEST+SWG(KG)                                            P1440...
       ROMG=RHOG(KG)*RELKG(KG)/VISCG(KG)                                P1450...
       RXXG(KG)=PERMXX(L)*ROMG                                          P1460...
       RXYG(KG)=PERMXY(L)*ROMG                                          P1470...
       RYXG(KG)=PERMYX(L)*ROMG                                          P1480...
       RYYG(KG)=PERMYY(L)*ROMG                                          P1490...
 4000  CONTINUE                                                         P1500...
C                                                                       P1510...
C.....INTEGRATE FLUID MASS BALANCE IN AN UNSATURATED ELEMENT            P1520...
C       USING ASYMMETRIC WEIGHTING FUNCTIONS                            P1530...
      IF(UP.LE.1.0D-06) GOTO 5200                                       P1540...
      IF(SWTEST-3.999D0) 4200,5200,5200                                 P1550...
 4200 DO 5000 I=1,4                                                     P1560...
```

```
      DF=0.D0                                                           P1570...
      VO=0.D0                                                           P1580...
      DO 4400 KG=1,4                                                    P1590...
       VO=VO+F(I,KG)*DET(KG)                                            P1600...
 4400  DF=DF+((RXXG(KG)*RGXG(KG)+RXYG(KG)*RGYG(KG))                     P1610...
     1         *DWDXG(I,KG)                                             P1620...
     2       +  (RYXG(KG)*RGXG(KG)+RYYG(KG)*RGYG(KG))                   P1630...
     3         *DWDYG(I,KG))*DET(KG)                                    P1640...
      DO 4800 J=1,4                                                     P1650...
       BF=0.D0                                                          P1660...
       DO 4600 KG=1,4                                                   P1670...
 4600   BF=BF+((RXXG(KG)*DFDXG(J,KG)+RXYG(KG)*DFDYG(J,KG))*DWDXG(I,KG)P1680...
     2        +(RYXG(KG)*DFDXG(J,KG)+RYYG(KG)*DFDYG(J,KG))*DWDYG(I,KG))P1690...
     3         *DET(KG)                                                 P1700...
 4800  BFLOWE(I,J)=BF                                                   P1710...
      VOLE(I)=VO                                                        P1720...
 5000 DFLOWE(I)=DF                                                      P1730...
     GOTO 6200                                                          P1740...
C                                                                       P1750...
C.....INTEGRATE FLUID MASS BALANCE IN A SATURATED OR UNSATURATED         P1760...
C        ELEMENT USING SYMMETRIC WEIGHTING FUNCTIONS                     P1770...
 5200 DO 6000 I=1,4                                                     P1780...
      DF=0.D0                                                           P1790...
      VO=0.D0                                                           P1800...
      DO 5400 KG=1,4                                                    P1810...
       VO=VO+F(I,KG)*DET(KG)                                            P1820...
 5400  DF=DF+((RXXG(KG)*RGXG(KG)+RXYG(KG)*RGYG(KG))*DFDXG(I,KG)        P1830...
     2     +  (RYXG(KG)*RGXG(KG)+RYYG(KG)*RGYG(KG))*DFDYG(I,KG))        P1840...
     3        *DET(KG)                                                  P1850...
      DO 5800 J=1,4                                                     P1860...
       BF=0.D0                                                          P1870...
       DO 5600 KG=1,4                                                   P1880...
 5600   BF=BF+((RXXG(KG)*DFDXG(J,KG)+RXYG(KG)*DFDYG(J,KG))*DFDXG(I,KG)P1890...
     1       +  (RYXG(KG)*DFDXG(J,KG)+RYYG(KG)*DFDYG(J,KG))*DFDYG(I,KG))P1900...
     2         *DET(KG)                                                 P1910...
 5800  BFLOWE(I,J)=BF                                                   P1920...
      VOLE(I)=VO                                                        P1930...
 6000 DFLOWE(I)=DF                                                      P1940...
 6200 CONTINUE                                                          P1950...
      IF(ML-1) 6100,9000,6100                                           P1960...
 6100 IF(NOUMAT.EQ.1) GOTO 9000                                         P1970...
C                                                                       P1980...
C                                                                       P1990...
C.....CALCULATE PARAMETERS FOR ENERGY BALANCE OR SOLUTE MASS BALANCE     P2000...
C        AT GAUSS POINTS                                                P2010...
      DO 7000 KG=1,4                                                    P2020...
       ESWG=PORG(KG)*SWG(KG)                                            P2030...
       RHOCWG=RHOG(KG)*CW                                               P2040...
       ESRCG=ESWG*RHOCWG                                                P2050...
       IF(VGMAG(KG)) 6300,6300,6600                                     P2060...
 6300  EXG(KG)=0.0D0                                                    P2070...
       EYG(KG)=0.0D0                                                    P2080...
       DXXG=0.0D0                                                       P2090...
       DXYG=0.0D0                                                       P2100...
       DYXG=0.0D0                                                       P2110...
       DYYG=0.0D0                                                       P2120...
       GOTO 6900                                                        P2130...
 6600  EXG(KG)=ESRCG*VXG(KG)                                            P2140...
       EYG(KG)=ESRCG*VYG(KG)                                            P2150...
C                                                                       P2160...
C.....DISPERSIVITY MODEL FOR ANISOTROPIC MEDIA                           P2170...
C        WITH PRINCIPAL DISPERSIVITIES:  ALMAX,ALMIN, AND ATAVG          P2180...
       VANGG=1.570796327D0                                              P2190...
```

```
        IF(VXG(KG)*VXG(KG).GT.0.D0) VANGG=DATAN(VYG(KG)/VXG(KG))        P2200...
        VKANGG=VANGG-PANGLE(L)                                          P2210...
        DCO=DCOS(VKANGG)                                                P2220...
        DSI=DSIN(VKANGG)                                                P2230...
C.....EFFECTIVE LONGITUDINAL DISPERSIVITY IN FLOW DIRECTION, ALEFF      P2240...
        ALEFF=0.0D0                                                     P2250...
        IF(ALMAX(L)+ALMIN(L)) 6800,6800,6700                            P2260...
 6700   ALEFF=ALMAX(L)*ALMIN(L)/(ALMIN(L)*DCO*DCO+ALMAX(L)*DSI*DSI)     P2270...
 6800   DLG=ALEFF*VGMAG(KG)                                             P2280...
        DTG=ATAVG(L)*VGMAG(KG)                                          P2290...
C                                                                       P2300...
        V2GMI=1.D0/(VGMAG(KG)*VGMAG(KG))                                P2310...
        V2ILTG=V2GMI*(DLG-DTG)                                          P2320...
        VX2G=VXG(KG)*VXG(KG)                                            P2330...
        VY2G=VYG(KG)*VYG(KG)                                            P2340...
C.....DISPERSION TENSOR                                                 P2350...
        DXXG=V2GMI*(DLG*VX2G+DTG*VY2G)                                  P2360...
        DYYG=V2GMI*(DTG*VX2G+DLG*VY2G)                                  P2370...
        DXYG=V2ILTG*VXG(KG)*VYG(KG)                                     P2380...
        DYXG=DXYG                                                       P2390...
C                                                                       P2400...
C.....IN-PARALLEL CONDUCTIVITIES (DIFFUSIVITIES) FORMULA                P2410...
 6900   ESE=ESRCG*SIGMAW+(1.D0-PORG(KG))*RHOCWG*SIGMAS                  P2420...
C.....ADD DIFFUSION AND DISPERSION TERMS TO TOTAL DISPERSION TENSOR     P2430...
        BXXG(KG)=ESRCG*DXXG+ESE                                         P2440...
        BXYG(KG)=ESRCG*DXYG                                             P2450...
        BYXG(KG)=ESRCG*DYXG                                             P2460...
 7000   BYYG(KG)=ESRCG*DYYG+ESE                                         P2470...
C                                                                       P2480...
C.....INTEGRATE SOLUTE MASS BALANCE OR ENERGY BALANCE                   P2490...
C        USING SYMMETRIC WEIGHTING FUNCTIONS FOR DISPERSION TERM AND    P2500...
C        USING EITHER SYMMETRIC OR ASYMMETRIC WEIGHTING FUNCTIONS       P2510...
C        FOR ADVECTION TERM                                             P2520...
        DO 8000 I=1,4                                                   P2530...
         DO 8000 J=1,4                                                  P2540...
         BT=0.D0                                                        P2550...
         DT=0.D0                                                        P2560...
         DO 7500 KG=1,4                                                 P2570...
          BT=BT+((BXXG(KG)*DFDXG(J,KG)+BXYG(KG)*DFDYG(J,KG))*DFDXG(I,KG)P2580...
     1         +(BYXG(KG)*DFDXG(J,KG)+BYYG(KG)*DFDYG(J,KG))*DFDYG(I,KG))P2590...
     2          *DET(KG)                                                P2600...
 7500     DT=DT+(EXG(KG)*DFDXG(J,KG)+EYG(KG)*DFDYG(J,KG))               P2610...
     1          *W(I,KG)*DET(KG)                                        P2620...
         BTRANE(I,J)=BT                                                 P2630...
 8000    DTRANE(I,J)=DT                                                 P2640...
 9000  CONTINUE                                                         P2650...
C                                                                       P2660...
C                                                                       P2670...
C.....SEND RESULTS OF INTEGRATIONS FOR THIS ELEMENT TO                  P2680...
C        GLOBAL ASSEMBLY ROUTINE                                        P2690...
 9999  CALL GLOBAN(L,ML,VOLE,BFLOWE,DFLOWE,BTRANE,DTRANE,               P2700...
     1    IN,VOL,PMAT,PVEC,UMAT,UVEC,CWRK)                              P2710NEW
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P2720...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P2730...
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P2740...
C                                                                       P2750...
C                                                                       P2760...
      RETURN                                                            P2770...
      END                                                               P2780...
C     SUBROUTINE        B  A  S  I  S  2         SUTRA - VERSION 1284-2D Q10.....
C                                                                       Q20.....
C *** PURPOSE :                                                         Q30.....
```

```
C ***  TO CALCULATE VALUES OF BASIS AND WEIGHTING FUNCTIONS AND THEIR   Q40.....
C ***  DERIVATIVES, TRANSFORMATION MATRICES BETWEEN LOCAL AND GLOBAL    Q50.....
C ***  COORDINATES AND PARAMETER VALUES AT A SPECIFIED POINT IN A       Q60.....
C ***  QUADRILATERAL FINITE ELEMENT.                                    Q70.....
C                                                                       Q80.....
      SUBROUTINE BASIS2(ICALL,L,XLOC,YLOC,IN,X,Y,F,W,DET,               Q90.....
     1   DFDXG,DFDYG,DWDXG,DWDYG,PITER,UITER,PVEL,POR,THICK,THICKG,     Q100....
     2   VXG,VYG,SWG,RHOG,VISCG,PORG,VGMAG,RELKG,                       Q110....
     3   PERMXX,PERMXY,PERMYX,PERMYY,CJ11,CJ12,CJ21,CJ22,               Q120....
     4   GXSI,GETA,RCIT,RCITM1,RGXG,RGYG)                               Q130....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               Q140....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          Q150NEW
     1   NSOP,NSOU,NBCN                                                 Q160....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,         Q170NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            Q180NEW
      COMMON/SATPAR/ PCENT,SWRES,PCRES,SSLOPE,SINCPT                    Q190....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,Q200....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2       Q210....
      COMMON/TENSOR/ GRAVX,GRAVY                                        Q220....
      DOUBLE PRECISION XLOC,YLOC                                        Q230....
      DIMENSION IN(NIN),X(NN),Y(NN),UITER(NN),PITER(NN),PVEL(NN),       Q240....
     1   POR(NN),PERMXX(NE),PERMXY(NE),PERMYX(NE),PERMYY(NE),THICK(NN)  Q250....
      DIMENSION GXSI(NE,4),GETA(NE,4),RCIT(NN),RCITM1(NN)               Q260....
      DIMENSION F(4),W(4),DFDXG(4),DFDYG(4),DWDXG(4),DWDYG(4)           Q270....
      DIMENSION FX(4),FY(4),AFX(4),AFY(4),                              Q280....
     1   DFDXL(4),DFDYL(4),DWDXL(4),DWDYL(4),                           Q290....
     2   XDW(4),YDW(4),XIIX(4),YIIY(4)                                  Q300....
      DATA XIIX/-1.D0,+1.D0,+1.D0,-1.D0/,                               Q310....
     1   YIIY/-1.D0,-1.D0,+1.D0,+1.D0/                                  Q320....
C                                                                       Q330....
C                                                                       Q340....
C.....AT THIS LOCATION IN LOCAL COORDINATES, (XLOC,YLOC),               Q350....
C        CALCULATE SYMMETRIC WEIGHTING FUNCTIONS, F(I),                 Q360....
C        SPACE DERIVATIVES, DFDXG(I) AND DFDYG(I), AND                  Q370....
C        DETERMINANT OF JACOBIAN, DET.                                  Q380....
C                                                                       Q390....
      XF1=1.D0-XLOC                                                     Q400....
      XF2=1.D0+XLOC                                                     Q410....
      YF1=1.D0-YLOC                                                     Q420....
      YF2=1.D0+YLOC                                                     Q430....
C                                                                       Q440....
C.....CALCULATE BASIS FUNCTION, F.                                      Q450....
      FX(1)=XF1                                                         Q460....
      FX(2)=XF2                                                         Q470....
      FX(3)=XF2                                                         Q480....
      FX(4)=XF1                                                         Q490....
      FY(1)=YF1                                                         Q500....
      FY(2)=YF1                                                         Q510....
      FY(3)=YF2                                                         Q520....
      FY(4)=YF2                                                         Q530....
      DO 10 I=1,4                                                       Q540....
   10 F(I)=0.250D0*FX(I)*FY(I)                                          Q550....
C                                                                       Q560....
C.....CALCULATE DERIVATIVES WITH RESPECT TO LOCAL COORDINATES.          Q570....
      DO 20 I=1,4                                                       Q580....
      DFDXL(I)=XIIX(I)*0.250D0*FY(I)                                    Q590....
   20 DFDYL(I)=YIIY(I)*0.250D0*FX(I)                                    Q600....
C                                                                       Q610....
C.....CALCULATE ELEMENTS OF JACOBIAN MATRIX, CJ.                        Q620....
      CJ11=0.D0                                                         Q630....
      CJ12=0.D0                                                         Q640....
      CJ21=0.D0                                                         Q650....
```

```
      CJ22=0.D0
      DO 100 IL=1,4
      II=(L-1)*4+IL
      I=IN(II)
      CJ11=CJ11+DFDXL(IL)*X(I)
      CJ12=CJ12+DFDXL(IL)*Y(I)
      CJ21=CJ21+DFDYL(IL)*X(I)
  100 CJ22=CJ22+DFDYL(IL)*Y(I)
C
C.....CALCULATE DETERMINANT OF JACOBIAN MATRIX.
      DET=CJ11*CJ22-CJ21*CJ12
C
C.....RETURN TO ELEMEN WITH JACOBIAN MATRIX ON FIRST TIME STEP.
      IF(ICALL.EQ.0) RETURN
C
C.....CALCULATE ELEMENTS OF INVERSE JACOBIAN MATRIX, CIJ.
      ODET=1.D0/DET
      CIJ11=+ODET*CJ22
      CIJ12=-ODET*CJ12
      CIJ21=-ODET*CJ21
      CIJ22=+ODET*CJ11
C
C.....CALCULATE DERIVATIVES WITH RESPECT TO GLOBAL COORDINATES
      DO 200 I=1,4
      DFDXG(I)=CIJ11*DFDXL(I)+CIJ12*DFDYL(I)
  200 DFDYG(I)=CIJ21*DFDXL(I)+CIJ22*DFDYL(I)
C
C.....CALCULATE CONSISTENT COMPONENTS OF (RHO*GRAV) TERM IN LOCAL
C         COORDINATES AT THIS LOCATION, (XLOC,YLOC)
      RGXL=0.D0
      RGYL=0.D0
      RGXLM1=0.D0
      RGYLM1=0.D0
      DO 800 IL=1,4
      II=(L-1)*4+IL
      I=IN(II)
      ADFDXL=DABS(DFDXL(IL))
      ADFDYL=DABS(DFDYL(IL))
      RGXL=RGXL+RCIT(I)*GXSI(L,IL)*ADFDXL
      RGYL=RGYL+RCIT(I)*GETA(L,IL)*ADFDYL
      RGXLM1=RGXLM1+RCITM1(I)*GXSI(L,IL)*ADFDXL
      RGYLM1=RGYLM1+RCITM1(I)*GETA(L,IL)*ADFDYL
  800 CONTINUE
C
C.....TRANSFORM CONSISTENT COMPONENTS OF (RHO*GRAV) TERM TO
C         GLOBAL COORDINATES
      RGXG=CIJ11*RGXL+CIJ12*RGYL
      RGYG=CIJ21*RGXL+CIJ22*RGYL
      RGXGM1=CIJ11*RGXLM1+CIJ12*RGYLM1
      RGYGM1=CIJ21*RGXLM1+CIJ22*RGYLM1
C
C.....CALCULATE PARAMETER VALUES AT THIS LOCATION, (XLOC,YLOC)
C
      PITERG=0.D0
      UITERG=0.D0
      DPDXG=0.D0
      DPDYG=0.D0
      PORG=0.D0
      THICKG=0.0D0
      DO 1000 IL=1,4
      II=(L-1)*4 +IL
      I=IN(II)
```

```
      DPDXG=DPDXG+PVEL(I)*DFDXG(IL)                                     Q1280...
      DPDYG=DPDYG+PVEL(I)*DFDYG(IL)                                     Q1290...
      PORG=PORG+POR(I)*F(IL)                                            Q1300...
      THICKG=THICKG+THICK(I)*F(IL)                                      Q1310...
      PITERG=PITERG+PITER(I)*F(IL)                                      Q1320...
      UITERG=UITERG+UITER(I)*F(IL)                                      Q1330...
 1000 CONTINUE                                                          Q1340...
C                                                                       Q1350...
C.....SET VALUES FOR DENSITY AND VISCOSITY                              Q1360...
C.....RHOG = FUNCTION(UITER)                                            Q1370...
      RHOG=RHOW0+DRWDU*(UITERG-URHOW0)                                  Q1380...
C.....VISCG = FUNCTION(UITER)                                           Q1390...
C        VISCOSITY IN UNITS OF VISC0*(KG/(M*SEC))                       Q1400...
      IF(ME) 1300,1300,1200                                             Q1410...
 1200 VISCG=VISC0*239.4D-07*(10.D0**(248.37D0/(UITERG+133.15D0)))       Q1420...
      GOTO 1400                                                         Q1430...
C.....FOR SOLUTE TRANSPORT... VISCG IS TAKEN TO BE CONSTANT             Q1440...
 1300 VISCG=VISC0                                                       Q1450...
 1400 CONTINUE                                                          Q1460...
C                                                                       Q1470...
C.....SET UNSATURATED FLOW PARAMETERS SWG AND RELKG                     Q1480...
      IF(IUNSAT-2) 1600,1500,1600                                       Q1490...
 1500 IF(PITERG) 1550,1600,1600                                         Q1500...
 1550 CALL UNSAT(SWG,DSWDPG,RELKG,PITERG)                               Q1510...
      GOTO 1700                                                         Q1520...
 1600 SWG=1.0D0                                                         Q1530...
      RELKG=1.0D0                                                       Q1540...
 1700 CONTINUE                                                          Q1550...
C                                                                       Q1560...
C.....CALCULATE CONSISTENT FLUID VELOCITIES WITH RESPECT TO GLOBAL      Q1570...
C        COORDINATES, VXG, VYG, AND VGMAG, AT THIS LOCATION, (XLOC,YLOC)Q1580...
      DENOM=1.D0/(PORG*SWG*VISCG)                                       Q1590...
      PGX=DPDXG-RGXGM1                                                  Q1600...
      PGY=DPDYG-RGYGM1                                                  Q1610...
C.....ZERO OUT RANDOM BOUYANT DRIVING FORCES DUE TO DIFFERENCING        Q1620...
C..... NUMBERS PAST PRECISION LIMIT                                     Q1630...
C..... MINIMUM DRIVING FORCE IS 1.D-10 OF PRESSURE GRADIENT             Q1640...
C..... (THIS VALUE MAY BE CHANGED DEPENDING ON MACHINE PRECISION)       Q1650...
      IF(DPDXG) 1720,1730,1720                                          Q1660...
 1720 IF(DABS(PGX/DPDXG)-1.0D-10) 1725,1725,1730                        Q1670...
 1725 PGX=0.0D0                                                         Q1680...
 1730 IF(DPDYG) 1750,1760,1750                                          Q1690...
 1750 IF(DABS(PGY/DPDYG)-1.0D-10) 1755,1755,1760                        Q1700...
 1755 PGY=0.0D0                                                         Q1710...
 1760 VXG=-DENOM*(PERMXX(L)*PGX+PERMXY(L)*PGY)*RELKG                    Q1720...
      VYG=-DENOM*(PERMYX(L)*PGX+PERMYY(L)*PGY)*RELKG                    Q1730...
      VXG2=VXG*VXG                                                      Q1740...
      VYG2=VYG*VYG                                                      Q1750...
      VGMAG=DSQRT(VXG2+VYG2)                                            Q1760...
C                                                                       Q1770...
C.....AT THIS POINT IN LOCAL COORDINATES, (XLOC,YLOC),                  Q1780...
C        CALCULATE ASYMMETRIC WEIGHTING FUNCTIONS, W(I),                Q1790...
C        AND SPACE DERIVATIVES, DWDXG(I) AND DWDYG(I).                  Q1800...
C                                                                       Q1810...
C.....ASYMMETRIC FUNCTIONS SIMPLIFY WHEN  UP=0.0                        Q1820...
      IF(UP.GT.1.0D-06.AND.NOUMAT.EQ.0) GOTO 1790                       Q1830...
      DO 1780 I=1,4                                                     Q1840...
      W(I)=F(I)                                                         Q1850...
      DWDXG(I)=DFDXG(I)                                                 Q1860...
      DWDYG(I)=DFDYG(I)                                                 Q1870...
 1780 CONTINUE                                                          Q1880...
C.....RETURN WHEN ONLY SYMMETRIC WEIGHTING FUNCTIONS ARE USED           Q1890...
      RETURN                                                            Q1900...
```

```
C                                                                       Q1910...
C.....CALCULATE FLUID VELOCITIES WITH RESPECT TO LOCAL COORDINATES,     Q1920...
C..... VXL, VYL, AND VLMAG, AT THIS LOCATION, (XLOC,YLOC).              Q1930...
 1790 VXL=CIJ11*VXG+CIJ21*VYG                                           Q1940...
      VYL=CIJ12*VXG+CIJ22*VYG                                           Q1950...
      VLMAG=DSQRT(VXL*VXL+VYL*VYL)                                      Q1960...
C                                                                       Q1970...
      AA=0.0D0                                                          Q1980...
      BB=0.0D0                                                          Q1990...
      IF(VLMAG) 1900,1900,1800                                          Q2000...
 1800 AA=UP*VXL/VLMAG                                                   Q2010...
      BB=UP*VYL/VLMAG                                                   Q2020...
C                                                                       Q2030...
 1900 XIXI=.750D0*AA*XF1*XF2                                            Q2040...
      YIYI=.750D0*BB*YF1*YF2                                            Q2050...
      DO 2000 I=1,4                                                     Q2060...
      AFX(I)=.50D0*FX(I)+XIIX(I)*XIXI                                   Q2070...
 2000 AFY(I)=.50D0*FY(I)+YIIY(I)*YIYI                                   Q2080...
C                                                                       Q2090...
C.....CALCULATE ASYMMETRIC WEIGHTING FUNCTION, W.                       Q2100...
      DO 3000 I=1,4                                                     Q2110...
 3000 W(I)=AFX(I)*AFY(I)                                                Q2120...
C                                                                       Q2130...
      THAAX=0.50D0-1.50D0*AA*XLOC                                       Q2140...
      THBBY=0.50D0-1.50D0*BB*YLOC                                       Q2150...
      DO 4000 I=1,4                                                     Q2160...
      XDW(I)=XIIX(I)*THAAX                                              Q2170...
 4000 YDW(I)=YIIY(I)*THBBY                                              Q2180...
C                                                                       Q2190...
C.....CALCULATE DERIVATIVES WITH RESPECT TO LOCAL COORDINATES.           Q2200...
      DO 5000 I=1,4                                                     Q2210...
      DWDXL(I)=XDW(I)*AFY(I)                                            Q2220...
 5000 DWDYL(I)=YDW(I)*AFX(I)                                            Q2230...
C                                                                       Q2240...
C.....CALCULATE DERIVATIVES WITH RESPECT TO GLOBAL COORDINATES.          Q2250...
      DO 6000 I=1,4                                                     Q2260...
      DWDXG(I)=CIJ11*DWDXL(I)+CIJ12*DWDYL(I)                            Q2270...
 6000 DWDYG(I)=CIJ21*DWDXL(I)+CIJ22*DWDYL(I)                            Q2280...
C                                                                       Q2290...
C                                                                       Q2300...
      RETURN                                                            Q2310...
      END                                                               Q2320...
C     SUBROUTINE        U  N  S  A  T          SUTRA - VERSION 1284-2D R10.....
C                                                                       R20.....
C *** PURPOSE :                                                         R30.....
C ***  USER-PROGRAMMED SUBROUTINE GIVING:                               R40.....
C ***  (1)  SATURATION AS A FUNCTION OF PRESSURE ( SW(PRES) )           R50.....
C ***  (2)  DERIVATIVE OF SATURATION WITH RESPECT TO PRESSURE           R60.....
C ***       AS A FUNCTION OF EITHER PRESSURE OR SATURATION              R70.....
C ***       ( DSWDP(PRES), OR DSWDP(SW) )                               R80.....
C ***  (3)  RELATIVE PERMEABILITY AS A FUNCTION OF EITHER               R90.....
C ***       PRESSURE OR SATURATION ( REL(PRES) OR RELK(SW) )            R100....
C ***                                                                   R110....
C ***  CODE BETWEEN DASHED LINES MUST BE REPLACED TO GIVE THE           R120....
C ***  PARTICULAR UNSATURATED RELATIONSHIPS DESIRED.                    R130....
C                                                                       R140....
      SUBROUTINE UNSAT(SW,DSWDP,RELK,PRES)                              R150....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               R160....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,         R170NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT   ,ITIME         R180NEW
C                                                                       R190....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - --R200....
```

```
C     THREE PARAMETERS FOR UNSATURATED FLOW RELATIONSHIPS OF            R210....
C     VAN GENUCHTEN(1980)                                               R220....
C        RESIDUAL SATURATION, SWRES, GIVEN IN UNITS [L**0]              R230....
C        PARAMETER, AA, GIVEN IN INVERSE PRESSURE UNITS [m*(s**2)/kg]   R240....
C        PARAMETER, VN, GIVEN IN UNITS [L**0]                           R250....
      DATA   SWRES/0.30D0/,   AA/5.0D-05/,   VN/2.0D0/                  R260....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - --R270....
C                                                                       R280....
C                                                                       R290....
C                                                                       R300....
C                                                                       R310....
C                                                                       R320....
C                                                                       R330....
C***********************************************************************R340....
C***********************************************************************R350....
C.....SECTION (1):                                                      R360....
C     SW VS. PRES   (VALUE CALCULATED ON EACH CALL TO UNSAT)            R370....
C     CODING MUST GIVE A VALUE TO SATURATION, SW.                       R380....
C                                                                       R390....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - R400....
C     THREE PARAMETER MODEL OF VAN GENUCHTEN(1980)                      R410....
      SWRM1=1.D0-SWRES                                                  R420....
      AAPVN=1.D0+(AA*(-PRES))**VN                                       R430....
      VNF=(VN-1.D0)/VN                                                  R440....
      AAPVNN=AAPVN**VNF                                                 R450....
      S W   =   SWRES+SWRM1/AAPVNN                                      R460....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - R470....
C***********************************************************************R480....
C***********************************************************************R490....
C                                                                       R500....
C                                                                       R510....
C                                                                       R520....
C                                                                       R530....
C                                                                       R540....
C                                                                       R550....
      IF(IUNSAT-2) 600,1200,1800                                        R560....
C***********************************************************************R570....
C***********************************************************************R580....
C.....SECTION (2):                                                      R590....
C     DSWDP VS. PRES, OR DSWDP VS. SW   (CALCULATED ONLY WHEN IUNSAT=1) R600....
C     CODING MUST GIVE A VALUE TO DERIVATIVE OF SATURATION WITH         R610....
C     RESPECT TO PRESSURE, DSWDP.                                       R620....
C                                                                       R630....
  600 CONTINUE                                                          R640....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - R650....
      DNUM=AA*(VN-1.D0)*SWRM1*(AA*(-PRES))**(VN-1.D0)                   R660....
      DNOM=AAPVN*AAPVNN                                                 R670....
      D S W D P   =   DNUM/DNOM                                         R680....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - R690....
      GOTO 1800                                                         R700....
C***********************************************************************R710....
C***********************************************************************R720....
C                                                                       R730....
C                                                                       R740....
C                                                                       R750....
C                                                                       R760....
C                                                                       R770....
C                                                                       R780....
C***********************************************************************R790....
C***********************************************************************R800....
C.....SECTION (3):                                                      R810....
C     RELK VS. P, OR RELK VS. SW   (CALCULATED ONLY WHEN IUNSAT=2)      R820....
C     CODING MUST GIVE A VALUE TO RELATIVE PERMEABILITY, RELK.          R830....
```

```
C                                                                       R840....
 1200 CONTINUE                                                          R850....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - R860....
C     GENERAL RELATIVE PERMEABILITY MODEL FROM VAN GENUCHTEN(1980)      R870....
      SWSTAR=(SW-SWRES)/SWRM1                                           R880....
      R E L K   =   DSQRT(SWSTAR)*                                      R890....
     1                 (1.D0-(1.D0-SWSTAR**(1.D0/VNF))**(VNF))**2.D0    R900....
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - R910....
C                                                                       R920....
C***********************************************************************R930....
C***********************************************************************R940....
C                                                                       R950....
C                                                                       R960....
C                                                                       R970....
C                                                                       R980....
C                                                                       R990....
C                                                                       R1000...
 1800 RETURN                                                            R1010...
C                                                                       R1020...
      END                                                               R1030...
C     SUBROUTINE        G  L  O  B  A  N        SUTRA - VERSION 1284-2D S10.....
C                                                                       S20.....
C *** PURPOSE :                                                         S30.....
C ***  TO ASSEMBLE RESULTS OF ELEMENTWISE INTEGRATIONS INTO             S40.....
C ***  A GLOBAL BANDED MATRIX AND GLOBAL VECTOR FOR BOTH                S50.....
C ***  FLOW AND TRANSPORT EQUATIONS.                                    S60.....
C                                                                       S70.....
      SUBROUTINE GLOBAN(L,ML,VOLE,BFLOWE,DFLOWE,BTRANE,DTRANE,          S80.....
     1   IN,VOL,PMAT,PVEC,UMAT,UVEC,CWRK)                               S90.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               S100....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          S110NEW
     1   NSOP,NSOU,NBCN                                                 S120....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        S130NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            S140NEW
      DIMENSION BFLOWE(4,4),DFLOWE(4),BTRANE(4,4),DTRANE(4,4),VOLE(4)   S150....
      DIMENSION VOL(NN),PMAT(NN,NBIP),PVEC(NN),UMAT(NN,NBIS),UVEC(NN)   S160NEW
      DIMENSION CWRK(NN)                                                NEW
      DIMENSION IN(NIN)                                                 S170....
C                                                                       S180....
      N1=(L-1)*4+1                                                      S190....
      N4=N1+3                                                           S200....
C                                                                       S210....
C.....ADD RESULTS OF INTEGRATIONS OVER ELEMENT L TO GLOBAL              S220....
C        P-MATRIX AND P-VECTOR                                          S230....
      IF(ML-1) 50,50,150                                                S240....
   50 IE=0                                                              S250....
C        (NBHALF=1 FOR PRESSURE EQN)                                    NEW
         NBHALF=1                                                       NEW
         NBWR=JT+2                                                      NEW
      DO 100 II=N1,N4                                                   S260....
C        ZERO OUT WORK ARRAY                                            NEW
         DO 60 IR=1,NBWR                                                NEW
   60    CWRK(IR)=0.0                                                   NEW
      IE=IE+1                                                           S270....
      IB=IN(II)                                                         S280....
      VOL(IB)=VOL(IB)+VOLE(IE)                                          S290....
      PVEC(IB)=PVEC(IB)+DFLOWE(IE)                                      S300....
      JE=0                                                              S310....
      DO 110 JJ=N1,N4                                                   S320....
      JE=JE+1                                                           S330....
C        SAVE ONLY SYMMETRIC HALF IN CONDENSED FORM                     NEW
```

```
      JB=IN(JJ)-IB+NBHALF                                               S340....
        IF(JB.LT.1) GO TO 110                                           NEW
      CWRK(JB)=CWRK(JB)+BFLOWE(IE,JE)                                   NEW
  110 CONTINUE                                                          NEW
C       ADD TERMS FROM WORK ARRAY TO GLOBAL MATRIX                      NEW
      DO 120 IR=1,NBIP                                                  NEW
      NR=NPT(IR+4)                                                      NEW
  120 PMAT(IB,IR)=PMAT(IB,IR)+CWRK(NR)                                  NEW
  100 CONTINUE                                                          NEW
      IF(ML-1) 150,300,150                                              S360....
C                                                                       S370....
C.....ADD RESULTS OF INTEGRATIONS OVER ELEMENT L TO GLOBAL               S380....
C        U-MATRIX                                                       S390....
  150 IF(NOUMAT.EQ.1) GOTO 300                                          S400....
      IE=0                                                              S410....
C       (NBHALF=JT+2 FOR TRANSPORT EQN)                                 NEW
        NBHALF=JT+2                                                     NEW
        NBWR=2*JT+3                                                     NEW
      DO 200 II=N1,N4                                                   S420....
C       ZERO OUT WORK ARRAY                                             NEW
        DO 70 IR=1,NBWR                                                 NEW
   70   CWRK(IR)=0.0                                                    NEW
      IE=IE+1                                                           S430....
      IB=IN(II)                                                         S440....
C.....POSITION FOR ADDITION TO U-VECTOR                                 S450....
C     UVEC(IB)=UVEC(IB)+ ((    ))                                       S460....
      JE=0                                                              S470....
      DO 210 JJ=N1,N4                                                   S480....
      JE=JE+1                                                           S490....
      JB=IN(JJ)-IB+NBHALF                                               S500....
C       SAVE FULL ROW IN CONDENSED FORM                                 NEW
      JB=IN(JJ)-IB+NBHALF                                               NEW
      CWRK(JB)=CWRK(JB)+DTRANE(IE,JE)+BTRANE(IE,JE)                     NEW
  210 CONTINUE                                                          NEW
C       ADD TERMS FROM WORK ARRAY TO GLOBAL MATRIX                      NEW
      DO 220 IR=1,NBIS                                                  NEW
      NR=NBHALF+NPT(IR)-1                                               NEW
  220 UMAT(IB,IR)=UMAT(IB,IR)+CWRK(NR)                                  NEW
  200 CONTINUE                                                          NEW
C                                                                       S520....
  300 CONTINUE                                                          S530....
C                                                                       S540....
C                                                                       S550....
      RETURN                                                            S560....
      END                                                               S570....
C     SUBROUTINE        N  O  D  A  L  B        SUTRA - VERSION 1284-2D T10.....
C                                                                       T20.....
C *** PURPOSE :                                                         T30.....
C ***  (1) TO CARRY OUT ALL CELLWISE CALCULATIONS AND TO ADD CELLWISE   T40.....
C ***      TERMS TO THE GLOBAL BANDED MATRIX AND GLOBAL VECTOR FOR      T50.....
C ***      BOTH FLOW AND TRANSPORT EQUATIONS.                           T60.....
C ***  (2) TO ADD FLUID SOURCE AND SOLUTE MASS OR ENERGY SOURCE TERMS   T70.....
C ***      TO THE MATRIX EQUATIONS.                                     T80.....
C                                                                       T90.....
      SUBROUTINE NODALB(ML,VOL,PMAT,PVEC,UMAT,UVEC,PITER,UITER,PM1,UM1, T100....
     1   UM2,POR,QIN,UIN,QUIN,CS1,CS2,CS3,SL,SR,SW,DSWDP,RHO,SOP)       T110....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               T120....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          T130NEW
     1   NSOP,NSOU,NBCN                                                 T140....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        T150....
```

```
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        T160....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,T170....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2       T180....
      COMMON/SATPAR/ PCENT,SWRES,PCRES,SSLOPE,SINCPT                    T190....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        T200NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            T210NEW
      DIMENSION VOL(NN),PMAT(NN,NBIP),PVEC(NN),UMAT(NN,NBIS),UVEC(NN)   T220NEW
      DIMENSION PITER(NN),UITER(NN),PM1(NN),UM1(NN),UM2(NN),            T230....
     1   POR(NN),QIN(NN),UIN(NN),QUIN(NN),CS1(NN),CS2(NN),CS3(NN),      T240....
     2   SL(NN),SR(NN),SW(NN),RHO(NN),DSWDP(NN),SOP(NN)                 T250....
C                                                                       T260....
C                                                                       T270....
      IF(IUNSAT.NE.0) IUNSAT=1                                          T280....
C                                                                       T290....
C.....DO NOT UPDATE NODAL PARAMETERS ON A TIME STEP WHEN ONLY U IS      T300....
C     SOLVED FOR BY BACK SUBSTITUTION (IE: WHEN NOUMAT=1)               T310....
      IF(NOUMAT) 50,50,200                                              T320....
C.....SET UNSATURATED FLOW PARAMETERS AT NODES, SW(I) AND DSWDP(I)      T330....
   50 DO 120 I=1,NN                                                     T340....
      IF(IUNSAT-1) 120,100,120                                          T350....
  100 IF(PITER(I)) 110,120,120                                          T360....
  110 CALL UNSAT(SW(I),DSWDP(I),RELK,PITER(I))                          T370....
  120 CONTINUE                                                          T380....
C.....SET FLUID DENSITY AT NODES, RHO(I)                                T390....
C     RHO = F (UITER(I))                                                T400....
      DO 150 I=1,NN                                                     T410....
  150 RHO(I)=RHOW0+DRWDU*(UITER(I)-URHOW0)                              T420....
  200 CONTINUE                                                          T430....
C                                                                       T440....
      DO 1000 I=1,NN                                                    T450....
      SWRHON=SW(I)*RHO(I)                                               T460....
C                                                                       T470....
      IF(ML-1) 220,220,230                                              T480....
C                                                                       T490....
C.....CALCULATE CELLWISE TERMS FOR P EQUATION                           T500....
C.....FOR STEADY-STATE FLOW, ISSFLO=2; FOR TRANSIENT FLOW, ISSFLO=0     T510....
  220 AFLN=(1-ISSFLO/2)*                                                T520....
     1    (SWRHON*SOP(I)+POR(I)*RHO(I)*DSWDP(I))*VOL(I)/DELTP           T530....
      CFLN=POR(I)*SW(I)*DRWDU*VOL(I)                                    T540....
      DUDT=(1-ISSFLO/2)*(UM1(I)-UM2(I))/DLTUM1                          T550....
      CFLN=CFLN*DUDT                                                    T560....
C.....ADD CELLWISE TERMS AND FLUID SOURCES OR FLUXES TO P EQUATION      T570....
C         LOAD TERMS ON DIAGONAL (NBHALF=1 FOR PRESSURE EQN)            NEW
          NBHALF=1                                                      NEW
      PMAT(I,NBHALF) = PMAT(I,NBHALF) + AFLN                            T580....
      PVEC(I) = PVEC(I) - CFLN + AFLN*PM1(I) + QIN(I)                   T590....
C                                                                       T600....
      IF(ML-1) 230,1000,230                                             T610....
C                                                                       T620....
C.....CALCULATE CELLWISE TERMS FOR U-EQUATION                           T630....
  230 EPRS=(1.D0-POR(I))*RHOS                                           T640....
      ATRN=(1-ISSTRA)*(POR(I)*SWRHON*CW+EPRS*CS1(I))*VOL(I)/DELTU       T650....
      GTRN=POR(I)*SWRHON*PRODF1*VOL(I)                                  T660....
      GSV=EPRS*PRODS1*VOL(I)                                            T670....
      GSLTRN=GSV*SL(I)                                                  T680....
      GSRTRN=GSV*SR(I)                                                  T690....
      ETRN=(POR(I)*SWRHON*PRODF0+EPRS*PRODS0)*VOL(I)                    T700....
C.....CALCULATE SOURCES OF SOLUTE OR ENERGY CONTAINED IN                T710....
C     SOURCES OF FLUID (ZERO CONTRIBUTION FOR OUTFLOWING FLUID)         T720....
      QUR=0.0D0                                                         T730....
      QUL=0.0D0                                                         T740....
      IF(QIN(I)) 360,360,340                                            T750....
```

```
  340 QUL=-CW*QIN(I)                                                    T760....
      QUR=-QUL*UIN(I)                                                   T770....
C.....ADD CELLWISE TERMS, SOURCES OF SOLUTE OR ENERGY IN FLUID INFLOWS, T780....
C        AND PURE SOURCES OR FLUXES OF SOLUTE OR ENERGY TO U-EQUATION   T790....
  360 IF(NOUMAT) 370,370,380                                            T800....
C       LOAD TERMS ON DIAGONAL (NBHALF=5 FOR TRANSPORT EQN)             NEW
  370   NBHALF=5                                                        NEW
      UMAT(I,NBHALF) = UMAT(I,NBHALF) + ATRN - GTRN - GSLTRN - QUL      T810....
  380 UVEC(I) = UVEC(I) + ATRN*UM1(I) + ETRN + GSRTRN + QUR + QUIN(I)   T820....
C                                                                       T830....
 1000 CONTINUE                                                          T840....
C                                                                       T850....
      RETURN                                                            T860....
      END                                                               T870....
C     SUBROUTINE        B  C  B                 SUTRA - VERSION 1284-2D U10.....
C                                                                       U20.....
C *** PURPOSE :                                                         U30.....
C ***  TO IMPLEMENT SPECIFIED PRESSURE AND SPECIFIED TEMPERATURE OR     U40.....
C ***  CONCENTRATION CONDITIONS BY MODIFYING THE GLOBAL FLOW AND       U50.....
C ***  TRANSPORT MATRIX EQUATIONS.                                      U60.....
C                                                                       U70.....
      SUBROUTINE BCB(ML,PMAT,PVEC,UMAT,UVEC,IPBC,PBC,IUBC,UBC,QPLITR)   U80.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               U90.....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          U100NEW
     1   NSOP,NSOU,NBCN                                                 U110....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        U120....
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        U130....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,U140....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2       U150....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,        U160NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT,ITIME            U170NEW
      DIMENSION PMAT(NN,NBIP),PVEC(NN),UMAT(NN,NBIS),UVEC(NN),          U180NEW
     1   IPBC(NBCN),PBC(NBCN),IUBC(NBCN),UBC(NBCN),QPLITR(NBCN)         U190....
C                                                                       U200....
C                                                                       U210....
      IF(NPBC.EQ.0) GOTO 1050                                           U220....
C.....SPECIFIED P BOUNDARY CONDITIONS                                   U230....
      DO 1000 IP=1,NPBC                                                 U240....
      I=IABS(IPBC(IP))                                                  U250....
C                                                                       U260....
      IF(ML-1) 100,100,200                                              U270....
C.....MODIFY EQUATION FOR P BY ADDING FLUID SOURCE AT SPECIFIED         U280....
C        PRESSURE NODE                                                  U290....
  100 GINL=-GNU                                                         U300....
      GINR=GNU*PBC(IP)                                                  U310....
C       LOAD TERMS ON DIAGONAL (NBHALF=1 FOR PRESSURE EQN)              NEW
        NBHALF=1                                                        NEW
      PMAT(I,NBHALF)=PMAT(I,NBHALF)-GINL                                U320....
      PVEC(I)=PVEC(I)+GINR                                              U330....
C                                                                       U340....
      IF(ML-1) 200,1000,200                                             U350....
C.....MODIFY EQUATION FOR U BY ADDING U SOURCE WHEN FLUID FLOWS IN      U360....
C        AT SPECIFIED PRESSURE NODE                                     U370....
  200 GUR=0.0D0                                                         U380....
      GUL=0.0D0                                                         U390....
      IF(QPLITR(IP)) 360,360,340                                        U400....
  340 GUL=-CW*QPLITR(IP)                                                U410....
      GUR=-GUL*UBC(IP)                                                  U420....
  360 IF(NOUMAT) 370,370,380                                            U430....
C       LOAD TERMS ON DIAGONAL (NBHALF=5 FOR TRANSPORT EQN)             NEW
  370   NBHALF=5                                                        NEW
      UMAT(I,NBHALF)=UMAT(I,NBHALF)-GUL                                 U440....
```

```
  380 UVEC(I)=UVEC(I)+GUR                                               U450....
 1000 CONTINUE                                                          U460....
C                                                                       U470....
C                                                                       U480....
 1050 IF(ML-1) 1100,3000,1100                                           U490....
C.....SPECIFIED U BOUNDARY CONDITIONS                                   U500....
C        MODIFY U EQUATION AT SPECIFIED U NODE TO READ: U = UBC         U510....
 1100 IF(NUBC.EQ.0) GOTO 3000                                           U520....
      DO 2000 IU=1,NUBC                                                 U530....
      IUP=IU+NPBC                                                       U540....
      I=IABS(IUBC(IUP))                                                 U550....
      IF(NOUMAT) 1200,1200,2000                                         U560....
 1200 DO 1500 JB=1,NBIS                                                 U570....
 1500 UMAT(I,JB)=0.0D0                                                  U580....
C       LOAD TERMS ON DIAGONAL (NBHALF=5 FOR TRANSPORT EQN)             NEW
        NBHALF=5                                                        NEW
      UMAT(I,NBHALF)=1.0D0                                              U590....
 2000 UVEC(I)=UBC(IUP)                                                  U600....
C                                                                       U610....
 3000 CONTINUE                                                          U620....
C                                                                       U630....
C                                                                       U640....
      RETURN                                                            U650....
      END                                                               U660....
C     SUBROUTINE        B  U  D  G  E  T        SUTRA - VERSION 1284-2D X10.....
C                                                                       X20.....
C *** PURPOSE :                                                         X30.....
C ***  TO CALCULATE AND OUTPUT FLUID MASS AND SOLUTE MASS OR            X40.....
C ***  ENERGY BUDGETS.                                                  X50.....
C                                                                       X60.....
      SUBROUTINE BUDGET(ML,IBCT,VOL,SW,DSWDP,RHO,SOP,QIN,PVEC,PM1,      X70.....
     1   PBC,QPLITR,IPBC,IQSOP,POR,UVEC,UM1,UM2,UIN,QUIN,IQSOU,UBC,     X80.....
     2   CS1,CS2,CS3,SL,SR)                                             X90.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               X100....
      CHARACTER*10 ADSMOD                                               X110....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8                     MODIFIED
      COMMON/MODSOR/ ADSMOD                                             X120....
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          X130NEW
     1   NSOP,NSOU,NBCN                                                 X140....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        X150....
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        X160....
      COMMON/PARAMS/ COMPFL,COMPMA,DRWDU,CW,CS,RHOS,DECAY,SIGMAW,SIGMAS,X170....
     1   RHOW0,URHOW0,VISC0,PRODF1,PRODS1,PRODF0,PRODS0,CHI1,CHI2       X180....
      COMMON/CNTRL1/ GNU,UP,DTMULT,DTMAX,ME,ISSFLO,ISSTRA,ITCYC,         X190NEW
     1   NPCYC,NUCYC,NPRINT,IREAD,ISTORE,NOUMAT,IUNSAT   ,ITIME         X200NEW
      CHARACTER*13 UNAME(2)                                             X210....
      DIMENSION QIN(NN),UIN(NN),IQSOP(NSOP),QUIN(NN),IQSOU(NSOU)        X220....
      DIMENSION IPBC(NBCN),UBC(NBCN),QPLITR(NBCN),PBC(NBCN)             X230....
      DIMENSION POR(NN),VOL(NN),PVEC(NN),UVEC(NN),SW(NN),DSWDP(NN),     X240....
     1   RHO(NN),SOP(NN),PM1(NN),UM1(NN),UM2(NN),                       X250....
     2   CS1(NN),CS2(NN),CS3(NN),SL(NN),SR(NN)                          X260....
      DATA UNAME(1)/'CONCENTRATION'/,UNAME(2)/' TEMPERATURE '/          X270....
C                                                                       X280....
C                                                                       X290....
      MN=2                                                              X300....
      IF(IUNSAT.NE.0) IUNSAT=1                                          X310....
      IF(ME.EQ.-1) MN=1                                                 X320....
      WRITE(K3,10)                                                      X330....
   10 FORMAT(1H1)                                                       X340....
C.....SET UNSATURATED FLOW PARAMETERS, SW(I) AND DSWDP(I)               X350....
      IF(IUNSAT-1) 40,20,40                                             X360....
   20 DO 30 I=1,NN                                                      X370....
```

```
      IF(PVEC(I)) 25,27,27                                              X380....
   25 CALL UNSAT(SW(I),DSWDP(I),RELK,PVEC(I))                           X390....
      GOTO 30                                                           X400....
   27 SW(I)=1.0D0                                                       X410....
      DSWDP(I)=0.0D0                                                    X420....
   30 CONTINUE                                                          X430....
C                                                                       X440....
C.....CALCULATE COMPONENTS OF FLUID MASS BUDGET                         X450....
   40 IF(ML-1) 50,50,1000                                               X460....
   50 CONTINUE                                                          X470....
      STPTOT=0.D0                                                       X480....
      STUTOT=0.D0                                                       X490....
      QINTOT=0.D0                                                       X500....
      DO 100 I=1,NN                                                     X510....
      STPTOT=STPTOT+(1-ISSFLO/2)*RHO(I)*VOL(I)*                         X520....
     1   (SW(I)*SOP(I)+POR(I)*DSWDP(I))*(PVEC(I)-PM1(I))/DELTP          X530....
      STUTOT=STUTOT+(1-ISSFLO/2)*POR(I)*SW(I)*DRWDU*VOL(I)*             X540....
     1   (UM1(I)-UM2(I))/DLTUM1                                         X550....
      QINTOT=QINTOT+QIN(I)                                              X560....
  100 CONTINUE                                                          X570....
C                                                                       X580....
      QPLTOT=0.D0                                                       X590....
      DO 200 IP=1,NPBC                                                  X600....
      I=IABS(IPBC(IP))                                                  X610....
      QPLITR(IP)=GNU*(PBC(IP)-PVEC(I))                                  X620....
      QPLTOT=QPLTOT+QPLITR(IP)                                          X630....
  200 CONTINUE                                                          X640....
C                                                                       X650....
C.....OUTPUT FLUID MASS BUDGET                                          X660....
      WRITE(K3,300) IT,STPTOT,STUTOT,UNAME(MN),QINTOT,QPLTOT            X670....
  300 FORMAT(//11X,'F L U I D   M A S S   B U D G E T      AFTER TIME', X680....
     1    ' STEP ',I5,',      IN (MASS/SECOND)'///11X,1PD15.7,5X,       X690....
     2    'RATE OF CHANGE IN TOTAL STORED FLUID DUE TO PRESSURE CHANGE', X700....
     3    ', INCREASE(+)/DECREASE(-)',/11X,1PD15.7,5X,                  X710....
     2    'RATE OF CHANGE IN TOTAL STORED FLUID DUE TO ',A13,' CHANGE', X720....
     3    ', INCREASE(+)/DECREASE(-)',                                  X730....
     3    /11X,1PD15.7,5X,'TOTAL OF FLUID SOURCES AND SINKS, ',         X740....
     4    'NET INFLOW(+)/NET OUTFLOW(-)'/11X,1PD15.7,5X,                X750....
     5    'TOTAL OF FLUID FLOWS AT POINTS OF SPECIFIED PRESSURE, ',     X760....
     6    'NET INFLOW(+)/NET OUTFLOW(-)')                               X770....
C                                                                       X780....
      IF(IBCT.EQ.4) GOTO 600                                            X790....
      NSOPI=NSOP-1                                                      X800....
      INEGCT=0                                                          X810....
      DO 500 IQP=1,NSOPI                                                X820....
      I=IQSOP(IQP)                                                      X830....
      IF(I) 325,500,500                                                 X840....
  325 INEGCT=INEGCT+1                                                   X850....
      IF(INEGCT.EQ.1) WRITE(K3,350)                                     X860....
  350 FORMAT(///22X,'TIME-DEPENDENT FLUID SOURCES OR SINKS'//22X,       X870....
     1    ' NODE',5X,'INFLOW(+)/OUTFLOW(-)'/37X,'   (MASS/SECOND)'//)   X880....
      WRITE(K3,450) -I,QIN(-I)                                          X890....
  450 FORMAT(22X,I5,10X,1PD15.7)                                        X900....
  500 CONTINUE                                                          X910....
C                                                                       X920....
  600 IF(NPBC.EQ.0) GOTO 800                                            X930....
      WRITE(K3,650)                                                     X940....
  650 FORMAT(///22X,'FLUID SOURCES OR SINKS DUE TO SPECIFIED PRESSURES',X950....
     1  //22X,' NODE',5X,'INFLOW(+)/OUTFLOW(-)'/37X,'   (MASS/SECOND)'/)X960....
      DO 700 IP=1,NPBC                                                  X970....
      I=IABS(IPBC(IP))                                                  X980....
      WRITE(K3,450) I,QPLITR(IP)                                        X990....
  700 CONTINUE                                                          X1000...
```

```
C                                                                       X1010...
C.....CALCULATE COMPONENTS OF ENERGY OR SOLUTE MASS BUDGET              X1020...
  800 IF(ML-1) 1000,4500,1000                                           X1030...
 1000 CONTINUE                                                          X1040...
      FLDTOT=0.D0                                                       X1050...
      SLDTOT=0.D0                                                       X1060...
      P1FTOT=0.D0                                                       X1070...
      P1STOT=0.D0                                                       X1080...
      P0FTOT=0.D0                                                       X1090...
      P0STOT=0.D0                                                       X1100...
      QQUTOT=0.D0                                                       X1110...
      QIUTOT=0.D0                                                       X1120...
C.....SET ADSORPTION PARAMETERS                                         X1130...
      IF(ME.EQ.-1.AND.ADSMOD.NE.'NONE      ')                           X1140...
     1   CALL ADSORB(CS1,CS2,CS3,SL,SR,UVEC)                            X1150...
      DO 1300 I=1,NN                                                    X1160...
      ESRV=POR(I)*SW(I)*RHO(I)*VOL(I)                                   X1170...
      EPRSV=(1.D0-POR(I))*RHOS*VOL(I)                                   X1180...
      DUDT=(1-ISSTRA)*(UVEC(I)-UM1(I))/DELTU                            X1190...
      FLDTOT=FLDTOT+ESRV*CW*DUDT                                        X1200...
      SLDTOT=SLDTOT+EPRSV*CS1(I)*DUDT                                   X1210...
      P1FTOT=P1FTOT+ESRV*PRODF1                                         X1220...
      P1STOT=P1STOT+EPRSV*PRODS1*(SL(I)*UVEC(I)+SR(I))                  X1230...
      P0FTOT=P0FTOT+ESRV*PRODF0                                         X1240...
      P0STOT=P0STOT+EPRSV*PRODS0                                        X1250...
      QQUTOT=QQUTOT+QUIN(I)                                             X1260...
      IF(QIN(I)) 1200,1200,1250                                         X1270...
 1200 QIUTOT=QIUTOT+QIN(I)*CW*UVEC(I)                                   X1280...
      GOTO 1300                                                         X1290...
 1250 QIUTOT=QIUTOT+QIN(I)*CW*UIN(I)                                    X1300...
 1300 CONTINUE                                                          X1310...
C                                                                       X1320...
      QPUTOT=0.D0                                                       X1330...
      DO 1500 IP=1,NPBC                                                 X1340...
      IF(QPLITR(IP)) 1400,1400,1450                                     X1350...
 1400 I=IABS(IPBC(IP))                                                  X1360...
      QPUTOT=QPUTOT+QPLITR(IP)*CW*UVEC(I)                               X1370...
      GOTO 1500                                                         X1380...
 1450 QPUTOT=QPUTOT+QPLITR(IP)*CW*UBC(IP)                               X1390...
 1500 CONTINUE                                                          X1400...
C                                                                       X1410...
      IF(ME) 1550,1550,1615                                             X1420...
C                                                                       X1430...
C.....OUTPUT SOLUTE MASS BUDGET                                         X1440...
 1550 WRITE(K3,1600) IT,FLDTOT,SLDTOT,P1FTOT,P1STOT,P0FTOT,P0STOT,      X1450...
     1   QIUTOT,QPUTOT,QQUTOT                                           X1460...
 1600 FORMAT(//11X,'S O L U T E   B U D G E T      AFTER TIME STEP ',I5,X1470...
     1   ',   IN (SOLUTE MASS/SECOND)'///11X,1PD15.7,5X,'NET RATE OF ', X1480...
     2   'INCREASE(+)/DECREASE(-) OF SOLUTE'/11X,1PD15.7,5X,            X1490...
     3   'NET RATE OF INCREASE(+)/DECREASE(-) OF ADSORBATE'/11X,1PD15.7,X1500...
     4   5X,'NET FIRST-ORDER PRODUCTION(+)/DECAY(-) OF SOLUTE'/11X,     X1510...
     5   1PD15.7,5X,'NET FIRST-ORDER PRODUCTION(+)/DECAY(-) OF ',       X1520...
     6   'ADSORBATE'/11X,1PD15.7,5X,'NET ZERO-ORDER PRODUCTION(+)/',    X1530...
     7   'DECAY(-) OF SOLUTE'/11X,1PD15.7,5X,'NET ZERO-ORDER ',         X1540...
     8   'PRODUCTION(+)/DECAY(-) OF ADSORBATE'/11X,1PD15.7,5X,          X1550...
     9   'NET GAIN(+)/LOSS(-) OF SOLUTE THROUGH FLUID SOURCES AND SINKS'X1560...
     *   /11X,1PD15.7,5X,'NET GAIN(+)/LOSS(-) OF SOLUTE THROUGH ',      X1570...
     1   'INFLOWS OR OUTFLOWS AT POINTS OF SPECIFIED PRESSURE'          X1580...
     2   /11X,1PD15.7,5X,'NET GAIN(+)/LOSS(-) OF SOLUTE THROUGH ',      X1590...
     3   'SOLUTE SOURCES AND SINKS')                                    X1600...
      GOTO 1645                                                         X1610...
C                                                                       X1620...
```

```
C.....OUTPUT ENERGY BUDGET                                              X1630...
 1615 WRITE(K3,1635) IT,FLDTOT,SLDTOT,P0FTOT,P0STOT,QIUTOT,QPUTOT,QQUTOTX1640...
 1635 FORMAT(//11X,'E N E R G Y   B U D G E T      AFTER TIME STEP ',I5,X1650...
     1   ',    IN (ENERGY/SECOND)'///11X,1PD15.7,5X,'NET RATE OF ',     X1660...
     2   'INCREASE(+)/DECREASE(-) OF ENERGY IN FLUID'/11X,1PD15.7,5X,   X1670...
     3   'NET RATE OF INCREASE(+)/DECREASE(-) OF ENERGY IN SOLID GRAINS'X1680...
     4   /11X,1PD15.7,5X,'NET ZERO-ORDER PRODUCTION(+)/LOSS(-) OF ',    X1690...
     5   'ENERGY IN FLUID'/11X,1PD15.7,5X,'NET ZERO-ORDER ',            X1700...
     6   'PRODUCTION(+)/LOSS(-) OF ENERGY IN SOLID GRAINS'              X1710...
     7   /11X,1PD15.7,5X,'NET GAIN(+)/LOSS(-) OF ENERGY THROUGH FLUID ',X1720...
     8   'SOURCES AND SINKS'/11X,1PD15.7,5X,'NET GAIN(+)/LOSS(-) OF ',  X1730...
     9   'ENERGY THROUGH INFLOWS OR OUTFLOWS AT POINTS OF SPECIFIED ',  X1740...
     *   'PRESSURE'/11X,1PD15.7,5X,'NET GAIN(+)/LOSS(-) OF ENERGY ',    X1750...
     1   'THROUGH ENERGY SOURCES AND SINKS')                            X1760...
C                                                                       X1770...
 1645 NSOPI=NSOP-1                                                      X1780...
      IF(NSOPI.EQ.0) GOTO 2000                                          X1790...
      IF(ME) 1649,1649,1659                                             X1800...
 1649 WRITE(K3,1650)                                                    X1810...
 1650 FORMAT(///22X,'SOLUTE SOURCES OR SINKS AT FLUID SOURCES AND ',    X1820...
     1   'SINKS'//22X,' NODE',8X,'SOURCE(+)/SINK(-)'/32X,               X1830...
     2   '(SOLUTE MASS/SECOND)'/)                                       X1840...
      GOTO 1680                                                         X1850...
 1659 WRITE(K3,1660)                                                    X1860...
 1660 FORMAT(///22X,'ENERGY SOURCES OR SINKS AT FLUID SOURCES AND ',    X1870...
     1   'SINKS'//22X,' NODE',8X,'SOURCE(+)/SINK(-)'/37X,               X1880...
     2   '(ENERGY/SECOND)'/)                                            X1890...
 1680 DO 1900 IQP=1,NSOPI                                               X1900...
      I=IABS(IQSOP(IQP))                                                X1910...
      IF(QIN(I)) 1700,1700,1750                                         X1920...
 1700 QU=QIN(I)*CW*UVEC(I)                                              X1930...
      GOTO 1800                                                         X1940...
 1750 QU=QIN(I)*CW*UIN(I)                                               X1950...
 1800 WRITE(K3,450) I,QU                                                X1960...
 1900 CONTINUE                                                          X1970...
C                                                                       X1980...
 2000 IF(NPBC.EQ.0) GOTO 4500                                           X1990...
      IF(ME) 2090,2090,2150                                             X2000...
 2090 WRITE(K3,2100)                                                    X2010...
 2100 FORMAT(///22X,'SOLUTE SOURCES OR SINKS DUE TO FLUID INFLOWS OR ', X2020...
     1   'OUTFLOWS AT POINTS OF SPECIFIED PRESSURE'//22X,' NODE',8X,    X2030...
     2   'SOURCE(+)/SINK(-)'/32X,'(SOLUTE MASS/SECOND)'/)               X2040...
      GOTO 2190                                                         X2050...
 2150 WRITE(K3,2160)                                                    X2060...
 2160 FORMAT(///22X,'ENERGY SOURCES OR SINKS DUE TO FLUID INFLOWS OR ', X2070...
     1   'OUTFLOWS AT POINTS OF SPECIFIED PRESSURE'//22X,' NODE',8X,    X2080...
     2   'SOURCE(+)/SINK(-)'/37X,'(ENERGY/SECOND)'/)                    X2090...
 2190 DO 2400 IP=1,NPBC                                                 X2100...
      I=IABS(IPBC(IP))                                                  X2110...
      IF(QPLITR(IP)) 2200,2200,2250                                     X2120...
 2200 QPU=QPLITR(IP)*CW*UVEC(I)                                         X2130...
      GOTO 2300                                                         X2140...
 2250 QPU=QPLITR(IP)*CW*UBC(IP)                                         X2150...
   00 WRITE(K3,450) I,QPU                                               X2160...
    0 CONTINUE                                                          X2170...
                                                                        X2180...
      IF(IBCT.EQ.4) GOTO 4500                                           X2190...
      NSOUI=NSOU-1                                                      X2200...
      INEGCT=0                                                          X2210...
      DO 3500 IQU=1,NSOUI                                               X2220...
      I=IQSOU(IQU)                                                      X2230...
      IF(I) 3400,3500,3500                                              X2240...
 3400 INEGCT=INEGCT+1                                                   X2250...
```

```
      IF(ME) 3450,3450,3460                                             X2260...
 3450 IF(INEGCT.EQ.1) WRITE(K3,3455)                                    X2270...
 3455 FORMAT(///22X,'TIME-DEPENDENT SOLUTE SOURCES AND SINKS'//22X,     X2280...
     1   ' NODE',10X,'GAIN(+)/LOSS(-)'/30X,'  (SOLUTE MASS/SECOND)'//)  X2290...
      GOTO 3475                                                         X2300...
 3460 IF(INEGCT.EQ.1) WRITE(K3,3465)                                    X2310...
 3465 FORMAT(///22X,'TIME-DEPENDENT ENERGY SOURCES AND SINKS'//22X,     X2320...
     1   ' NODE',10X,'GAIN(+)/LOSS(-)'/35X,'  (ENERGY/SECOND)'//)       X2330...
 3475 CONTINUE                                                          X2340...
      WRITE(K3,3490) -I,QUIN(-I)                                        X2350...
 3490 FORMAT(22X,I5,10X,1PD15.7)                                        X2360...
 3500 CONTINUE                                                          X2370...
C                                                                       X2380...
C                                                                       X2390...
 4500 CONTINUE                                                          X2400...
C                                                                       X2410...
      RETURN                                                            X2420...
      END                                                               X2430...
C     SUBROUTINE        S  T  O  R  E           SUTRA - VERSION 1284-2D Y10.....
C                                                                       Y20.....
C *** PURPOSE :                                                         Y30.....
C ***  TO STORE RESULTS THAT MAY LATER BE USED TO RE-START              Y40.....
C ***  THE SIMULATION.                                                  Y50.....
C                                                                       Y60.....
      SUBROUTINE STORE(PVEC,UVEC,PM1,UM1,CS1,RCIT,SW,PBC)               Y70.....
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               Y80.....
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8                     MODIFIED
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,          Y90NEW
     1   NSOP,NSOU,NBCN                                                 Y100....
      COMMON/TIME/ DELT,TSEC,TMIN,THOUR,TDAY,TWEEK,TMONTH,TYEAR,        Y110....
     1   TMAX,DELTP,DELTU,DLTPM1,DLTUM1,IT,ITMAX                        Y120....
      DIMENSION PVEC(NN),UVEC(NN),PM1(NN),UM1(NN),CS1(NN),RCIT(NN),     Y130....
     1   SW(NN),PBC(NBCN)                                               Y140....
C                                                                       Y150....
C.....REWIND UNIT-66 FOR WRITING RESULTS OF CURRENT TIME STEP           Y160....
      REWIND(66)                                                        Y170....
C                                                                       Y180....
C.....STORE TIME INFORMATION                                            Y190....
      WRITE(K4,100) TSEC,DELTP,DELTU                                    Y200....
  100 FORMAT(4D20.10)                                                   Y210....
C                                                                       Y220....
C.....STORE SOLUTION                                                    Y230....
      WRITE(K4,110) (PVEC(I),I=1,NN)                                    Y240....
      WRITE(K4,110) (UVEC(I),I=1,NN)                                    Y250....
      WRITE(K4,110) (PM1(I),I=1,NN)                                     Y260....
      WRITE(K4,110) (UM1(I),I=1,NN)                                     Y270....
      WRITE(K4,110) (CS1(I),I=1,NN)                                     Y280....
      WRITE(K4,110) (RCIT(I),I=1,NN)                                    Y290....
      WRITE(K4,110) (SW(I),I=1,NN)                                      Y300....
      WRITE(K4,110) (PBC(IP),IP=1,NBCN)                                 Y310....
  110 FORMAT(4(1PD20.13))                                               Y320....
C                                                                       Y330....
      ENDFILE(K4)                                                       Y340....
C                                                                       Y350....
      RETURN                                                            Y360....
      END                                                               Y370....
C     SUBROUTINE        F  O  P  E  N           SUTRA - VERSION 0690-2D Z10....MODIFIED
C                                                                       Z20....MODIFIED
C *** PURPOSE :                                                         Z30....MODIFIED
C ***  OPENS FILES FOR SUTRA SIMULATION.                                Z40....MODIFIED
C ***  OPENS ERROR OUTPUT FILE, READS FILE NUMBERS AND NAMES,           Z50....MODIFIED
C ***  CHECKS FOR EXISTENCE OF INPUT FILES, AND WRITES ERROR MESSAGES.  Z60....MODIFIED
```

```
C                                                                       Z70....MODIFIED
      SUBROUTINE FOPEN(UNAME,ENAME,FNAME,IUNIT,NFILE)                   Z90....MODIFIED
      CHARACTER*80 FN,UNAME,ENAME,FNAME                                 Z100...MODIFIED
      LOGICAL IS                                                        Z110...MODIFIED
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8                     Z120...MODIFIED
      DIMENSION FNAME(8),IUNIT(8)                                       Z130...MODIFIED
C                                                                       Z140...MODIFIED
C.....OPEN FILE UNIT CONTAINING UNIT NUMBERS AND FILE ASSIGNMENTS       Z150...MODIFIED
      IU=K0                                                             Z160...MODIFIED
      FN=UNAME                                                          Z170...MODIFIED
      INQUIRE(FILE=UNAME,EXIST=IS)                                      Z180...MODIFIED
      IF(IS) THEN                                                       Z190...MODIFIED
       OPEN(UNIT=IU,FILE=UNAME,STATUS='OLD',FORM='FORMATTED',           Z200...MODIFIED
     1   IOSTAT=KERR)                                                   Z210...MODIFIED
      ELSE                                                              Z220...MODIFIED
       GOTO 8000                                                        Z230...MODIFIED
      ENDIF                                                             Z240...MODIFIED
      IF(KERR.GT.0) GOTO 9000                                           Z250...MODIFIED
C                                                                       Z260...MODIFIED
C.....READ FILE CONTAINING UNIT NUMBERS AND FILE ASSIGNMENTS            Z270...MODIFIED
      NFILE=0                                                           Z280...MODIFIED
  100 READ(K0,*,END=200) IU                                             Z290...MODIFIED
      READ(K0,150,END=200) FN                                           Z300...MODIFIED
  150 FORMAT(A80)                                                       Z310...MODIFIED
      NFILE=NFILE+1                                                     Z320...MODIFIED
      IUNIT(NFILE)=IU                                                   Z330...MODIFIED
      FNAME(NFILE)=FN                                                   Z340...MODIFIED
      GOTO 100                                                          Z350...MODIFIED
  200 CONTINUE                                                          Z360...MODIFIED
C.....CHECK FOR EXISTENCE OF INPUT FILES                                Z370...MODIFIED
C          AND OPEN BOTH INPUT AND OUTPUT FILES                         Z380...MODIFIED
      DO 300 NF=1,NFILE                                                 Z390...MODIFIED
      IU=IUNIT(NF)                                                      Z400...MODIFIED
      FN=FNAME(NF)                                                      Z410...MODIFIED
      IF(NF.LE.2) THEN                                                  Z420...MODIFIED
       INQUIRE(FILE=FN,EXIST=IS)                                        Z430...MODIFIED
       IF(IS) THEN                                                      Z440...MODIFIED
        OPEN(UNIT=IU,FILE=FN,STATUS='OLD',FORM='FORMATTED',IOSTAT=KERR)
Z450...MODIFIED
       ELSE                                                             Z460...MODIFIED
        GOTO 8000                                                       Z470...MODIFIED
       ENDIF                                                            Z480...MODIFIED
      ELSE                                                              Z490...MODIFIED
       OPEN(UNIT=IU,FILE=FN,STATUS='UNKNOWN',FORM='FORMATTED',          Z500...MODIFIED
     1    IOSTAT=KERR)                                                  Z510...MODIFIED
      ENDIF                                                             Z520...MODIFIED
      IF(KERR.GT.0) GOTO 9000                                           Z530...MODIFIED
  300 CONTINUE                                                          Z540...MODIFIED
      K1=IUNIT(1)                                                       Z550...MODIFIED
      K2=IUNIT(2)                                                       Z560...MODIFIED
      K3=IUNIT(3)                                                       Z570...MODIFIED
      K4=IUNIT(4)                                                       Z580...MODIFIED
      K5=IUNIT(5)                                                       Z581...MODIFIED
      K6=IUNIT(6)                                                       Z582...MODIFIED
      K7=IUNIT(7)                                                       Z583...MODIFIED
      K8=IUNIT(8)                                                       Z584...MODIFIED
      RETURN                                                            Z590...MODIFIED
```

```
C                                                                      Z600...MODIFIED
C.....OPEN FILE UNIT FOR ERROR MESSAGES                                 Z610...MODIFIED
 8000 OPEN(UNIT=K00,FILE=ENAME,STATUS='UNKNOWN',FORM='FORMATTED')
Z620...MODIFIED
C.....WRITE ERROR MESSAGE AND STOP                                     Z630...MODIFIED
      WRITE(K00,8888) FN                                               Z640...MODIFIED
 8888 FORMAT('* E R R O R *'/'THE FILE:'/A80/'DOES NOT EXIST!')        Z650...MODIFIED
      ENDFILE(K00)                                                     Z660...MODIFIED
      STOP                                                             Z670...MODIFIED
C                                                                      Z680...MODIFIED
C.....OPEN FILE UNIT FOR ERROR MESSAGES                                 Z690...MODIFIED
 9000 OPEN(UNIT=K00,FILE=ENAME,STATUS='UNKNOWN',FORM='FORMATTED') Z700...MODIFIED
C.....WRITE ERROR MESSAGE AND STOP                                     Z710...MODIFIED
      WRITE(K00,9999) IU,FN                                            Z720...MODIFIED
 9999 FORMAT('* E R R O R *'/'UNIT ',I3/'ASSIGNED TO FILE:'/A80/       Z730...MODIFIED
     1   'CANNOT BE OPENED!')                                          Z740...MODIFIED
      ENDFILE(K00)                                                     Z750...MODIFIED
      STOP                                                             Z760...MODIFIED
C                                                                      Z770...MODIFIED
      END                                                              Z780...MODIFIED
```

Appendix III

Subprograms SOLVEC and LSORA Used to Solve System of Equations

```
C      SUBROUTINE   N E W   S O L V E
C.....SUBROUTINE   N E W   S O L V E
C
C.....PURPOSE: SOLVE FLOW EQUATIONS USING THE INCOMPLETE
C                CHOLESKY-CONJUGATE GRADIENT TECHNIQUE
C
C.....SOLVE SYSTEM OF EQUATIONS FOR FLOW
C
      SUBROUTINE SOLVEC(NBW,A,OLDH,RHS,P,R,AP,XK1,AB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,
     1   NSOP,NSOU,NBCN
      COMMON/ITERAT/ RPM,RPMAX,RUM,RUMAX,ITER,ITRMAX,IPWORS,IUWORS,
     1                ICON,ITRMX2,OMEGA,RPMX2,RUMX2
      DIMENSION A(NN,NBW),OLDH(NN),RHS(NN)
      DIMENSION P(NN),R(NN),AP(NN),XK1(NN),AB(NN,5)
C
      EPS1=RPMX2
C
C.....INITIALIZE R1 AND P1, AND STORE OLDH IN RHS FOR ITERATIVE SOLUTION
      CALL MATMLP(A,OLDH,R,NBW)
      DO 20 I=1,NN
      R(I)=RHS(I)-R(I)
      RHS(I)=OLDH(I)
20    CONTINUE
        IDC=0
        CALL SDCOMP(A,AB,R,P,NBW,IDC)
        IDC=1
        CALL SDCOMP(A,AB,R,P,NBW,IDC)
C
C.....BEGIN ITERATIVE LOOP -- SOLUTION MUST CONVERGE IN NN ITERATIONS
      NN1=NN+1
      DO 30 ITR=1,NN1
      CALL MATMLP(A,P,AP,NBW)
C
C.....FORM DOT PRODUCT OF P AND AP AND STORE IT AS LAMDA
      XLAM=0.0
      DO 110 K=1,NN
110   XLAM=XLAM+P(K)*AP(K)
        IDC=1
        CALL SDCOMP(A,AB,R,XK1,NBW,IDC)
C
C.....FORM DOT PRODUCT OF R AND XK1 AND STORE IT AS RR1
      RR1=0.0
      DO 120 K=1,NN
120   RR1=RR1+R(K)*XK1(K)
C
C.....UPDATE H (BUT STORE IT IN RHS)
C.....UPDATE R AND XKI AND CHECK MAXIMUM ERROR
      ALPHA=RR1/XLAM
      RMAX=0.0
      DO 40 J=1,NN
      RHS(J)=RHS(J)+ALPHA*P(J)
      R(J)=R(J)-ALPHA*AP(J)
      RABS=DABS(R(J))
40    IF(RABS.GT.RMAX) RMAX=RABS
```

```
C
C.....CHECK IF METHOD HAS CONVERGED
      IF(RMAX.LT.EPS1) GOTO 70
C
C.....CHECK IF USER SPECIFIED ITERATION LIMIT IS EXCEEDED
      IF(ITR.GE.ITRMX2) GOTO 50
C
      IF(MOD(ITR,10).EQ.0) WRITE(6,533) ITR,RMAX
C
C.....UPDATE P AND GO ON TO NEXT ITERATION
        IDC=1
        CALL SDCOMP(A,AB,R,XK1,NBW,IDC)
C       FORM DOT PRODUCT OF R AND XK1 AND STORE IT AS RR2
      RR2=0.0
      DO 130 K=1,NN
130   RR2=RR2+R(K)*XK1(K)
      BETA= RR2/RR1
        DO 35 J=1,NN
35      P(J)=XK1(J)+BETA*P(J)
30    CONTINUE
70    CONTINUE
      WRITE(K3,99) ITR
99    FORMAT(/10X,'ICCG METHOD CONVERGED IN',I5,' ITERATIONS')
      GO TO 60
50    WRITE(K3,98) ITRMX2
98    FORMAT(//,5X,'FAILED TO CONVERGE AFTER ',I6,' ITERATIONS'/
     1        /,5X,'PROGRAM WILL STOP')
533   FORMAT(1H ,3X,'RMAX AT ITERATION',I5,' =',1P1E15.5)
      STOP 151
60    RETURN
      END

C
C     SUBROUTINE MATMLP-- WRITTEN BY E.J. WEXLER
C     PURPOSE: TO MULTIPLY A VECTOR B BY A NN X NN BANDED MATRIX
C     WITH ONLY THE UPPER NON-ZERO BANDS OF A STORED.
C
C       LOOP THROUGH ALL ROWS OF MATRIX A

      SUBROUTINE MATMLP (A,B,C,NBW)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,
     1   NSOP,NSOU,NBCN
      DIMENSION A(NN,NBW),B(NN),C(NN)
      DO 100 K=1,NN
      SUM=0.0
      DO 300 J=1,NBW
      NPTJ=NPT(J+4)
      IC1=NPTJ+K-1
      IC2=K-NPTJ+1
      IF(IC1.LE.NN) SUM=SUM+A(K,J)*B(IC1)
      IF (J.LT.2) GOTO 300
      IF(IC2.GT.0) SUM=SUM+A(IC2,J)*B(IC2)
300   CONTINUE
      C(K)=SUM
100   CONTINUE
      RETURN
```

```
      END

C
C          SDCOMP--MODIFIED BY E.J. WEXLER TO DO AN INCOMPLETE
C     C          CHOLESKY DECOMPOSITION OF A SYMMETRIC BANDED MATRIX
C                  SSOLVE DOES THE FOWARD AND BACKWARDS SUBSTITUTION

      SUBROUTINE SDCOMP (A,AB,R,XK1,NBW,IDC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,
     1   NSOP,NSOU,NBCN
      DIMENSION A(NN,NBW),AB(NN,5),R(NN),XK1(NN)
C
      IF(IDC.GT.0) GOTO 300
C
C.....DECOMPOSE SYM. MATRIX A AND STORE IN AB
      DO 100 K=1,NN
      DO 100 J=1,NBW
      NPTJ=NPT(J+4)
      IC1=NPTJ+K-1
      IF (IC1.GT.NN) GO TO 100
      SUM=A(K,J)
      DO 10 L=2,NBW
      NPTL=NPT(L+4)
      IC2=K-NPTL+1
      IF (IC2.LT.1) GO TO 10
      IC3=NPTJ+NPTL-1
      M=J+L-1
      IF (M.GT.NBW) GO TO 10
      NPTM=NPT(M+4)
      IF(NPTM.NE.IC3) GO TO 10
      SUM=SUM-AB(IC2,L)*AB(IC2,M)
   10 CONTINUE
      IF (NPTJ.EQ.1) THEN
C          STOP IF DIVIDING BY ZERO.
        IF (SUM.LE.0.0) THEN
C           WRITE (*,120) K,SUM
C           WRITE (K3,120) K,SUM
          STOP
        END IF
        ADIAGN=1./DSQRT(SUM)
        AB(K,J)=ADIAGN
      END IF
      IF (NPTJ.GT.1) AB(K,J)=SUM*ADIAGN
  100 CONTINUE
      RETURN
C
C     ************
C     ENTRY SSOLVE
C
C.....FORWARD SUBSTITUTE FOR LOWER TRIANGLE
300   DO 80 K=1,NN
      SUM=R(K)
      DO 60 J=2,NBW
      NPTJ=NPT(J+4)
      IC2=K-NPTJ+1
```

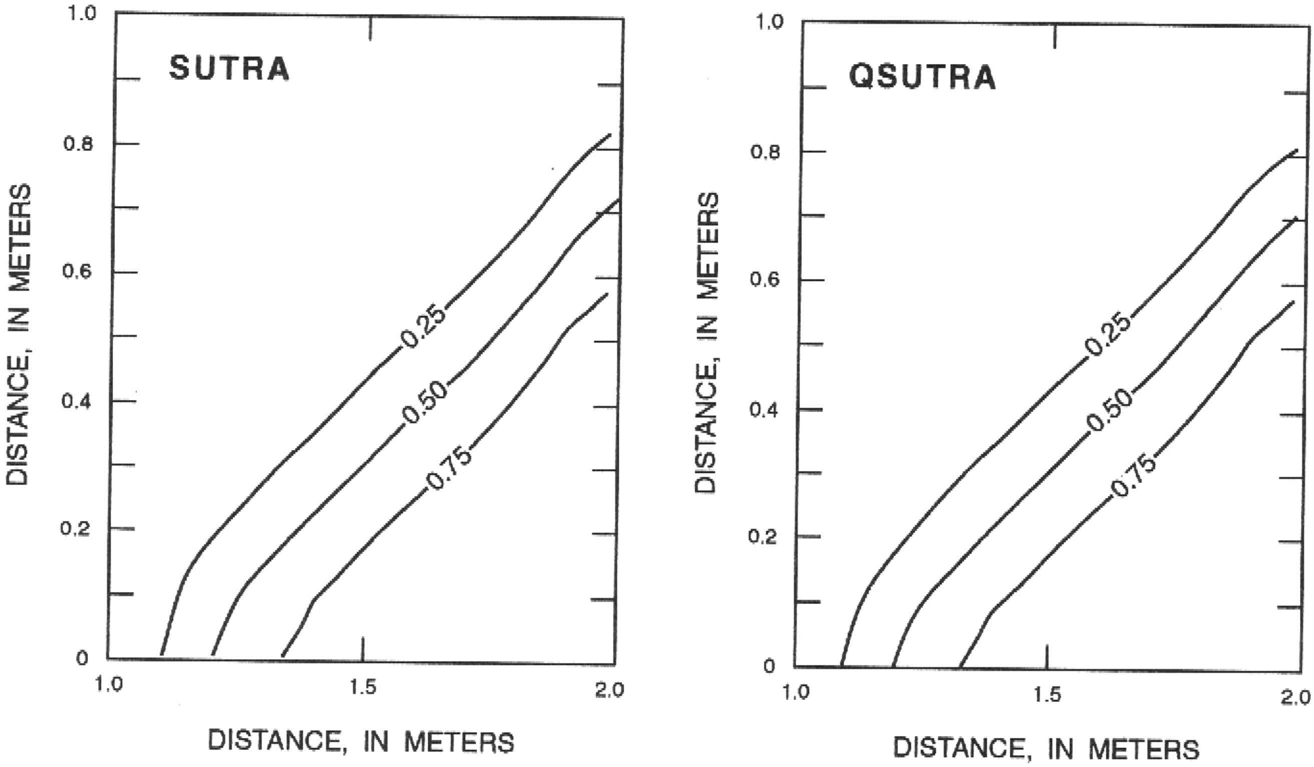
```
      IF (IC2.LT.1) GO TO 80
      SUM=SUM-AB(IC2,J)*XK1(IC2)
   60 CONTINUE
   80 XK1(K)=SUM*AB(K,1)
C
C.....BACKWARD SUBSTITUTE FOR UPPER TRIANGLE
      DO 110 K=1,NN
      IJ=NN-K+1
      SUM=XK1(IJ)
      DO 90 J=2,NBW
      NPTJ=NPT(J+4)
      IC1=NPTJ+IJ-1
      IF (IC1.GT.NN) GO TO 110
   90 SUM=SUM-AB(IJ,J)*XK1(IC1)
  110 XK1(IJ)=SUM*AB(IJ,1)
      RETURN
C
C
  120 FORMAT (1H1,5X,'**ERROR**',5X,'DIVIDE BY ZERO AT LINE ',I4,' IN DE
     1COMPOSITION ROUTINE',3X,'SUM =',1P1E13.5)
      END

C
C     SUBROUTINE LSORA
C
C     SOLVE SYSTEM OF EQUATIONS FOR TRANSPORT
C
C       LINE-SUCCESSIVE OVER-RELAXATION TECHNIQUE (LSOR)
C         A = FULL ASYMETRIC MATRIX
C         B = RHS (SOLUTION IS LOADED INTO B AT END)
C         X0 = INITIAL GUESS
C         X = SOLUTION VECTOR
C         AA = WORK ARRAYS FOR LSOR SOLUTION
C       LOAD X0 INTO X AS INITIAL GUESS
      SUBROUTINE LSORA (NBW,A,B,X0,X,XP,AA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/FUNITS/ K00,K0,K1,K2,K3,K4,K5,K6,K7,K8
      COMMON/DIMS/ NN,NE,NIN,IS,JT,NBIP,NBIS,NPT(9),NPBC,NUBC,
     1   NSOP,NSOU,NBCN
      COMMON/ITERAT/ RPM,RPMAX,RUM,RUMAX,ITER,ITRMAX,IPWORS,IUWORS,
     1               ICON,ITRMX2,OMEGA,RPMX2,RUMX2
      DIMENSION A(NN,NBW),B(NN),X0(NN),X(NN),AA(NN,5),XP(NN)
      EPS=RUMX2
      DO 5 I=1,NN
      XP(I)=X0(I)
5     X(I)=XP(I)
C
C.....BEGIN ITERATION LOOP
      ITER1 = 0
10    ITER1 = ITER1 + 1
C
C.....LOOP THROUGH ALL I
      DO 50 I=1,IS
      II=(I-1)*JT
C
C.....LOAD COEFFICIENTS FOR LINE INTO AA
      DO 20 J=1,JT
```

```
      JJ=II+J
      AA(J,1)=A(JJ,4)
      AA(J,2)=A(JJ,5)
      AA(J,3)=A(JJ,6)
      DD=B(JJ)
      DO 30 K=1,3
      NPTK=NPT(K+6)
      IC1=JJ+NPTK-1
      IF(IC1.LE.NN) DD=DD-A(JJ,K+6)*X(IC1)
      NPTK=NPT(K)
      IC2=JJ+NPTK-1
      IF(IC2.GE.1) DD=DD-A(JJ,K)*X(IC2)
30    CONTINUE
      AA(J,4)=DD
20    CONTINUE
C
C.....SOLVE ROW EQUATIONS USING THOMAS ALGORITHM
      CALL THOMAS (AA,JT,NN)
C
C.....LOAD NEW BLOCK VALUES INTO X ARRAY
      DO 45 J=1,JT
      JJ=II+J
      X(JJ)=XP(JJ) + OMEGA*(AA(J,5)-XP(JJ))
45    CONTINUE
50    CONTINUE
C
C.....FIND LARGEST CHANGE AND STORE NEW VALUE FOR X(I) IN XP(I)
      DIFMAX=0.0
      DO 40 I=1,NN
      DIF = DABS(X(I)-XP(I))
      IF(DIF.GT.DIFMAX) DIFMAX=DIF
      XP(I)=X(I)
40    CONTINUE
C
C.....CHECK FOR MAXIMUM NUMBER OF ITERATIONS
      IF (ITER1.GT.ITRMX2) THEN
        WRITE (K3,901)
901     FORMAT (5X,'MAXIMUM ITERATIONS EXCEEDED, PROGRAM WILL STOP')
        STOP
      END IF
C
C.....CHECK FOR CONVERGENCE
      IF (MOD(ITER1,10).EQ.0) WRITE (K3,105) ITER1,DIFMAX
105   FORMAT (5X,'MAXIMUM DIFFERENCE AT ITERATION NUMBER',I5,' = ',
     1 1P1E12.5)
      IF (DIFMAX.GT.EPS) GO TO 10

C.....CONVERGENCE ACHIEVED
      WRITE (K3,101) ITER1
101   FORMAT (10X,'LSOR METHOD CONVERGED IN',I5,' ITERATIONS'/)
C       LOAD SOLUTION INTO B
      DO 70 I=1,NN
70    B(I)=X(I)
      RETURN
      END
C
C       SUBROUTINE THOMAS ALGORITHIM
```

```
C
C       THOMAS ALGORITHIM FOR A TRIDIAGONAL MATRIX
C       A(I,1),A(I,2),A(I,3) ARE THE DIAGONALS OF THE MATRIX
C       A(I,4) IS THE RHS, A(I,5) IS THE SOLUTION VECTOR

      SUBROUTINE THOMAS (A,N,NN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NN,5)
      DO 10 I=2,N
      A(I,2) = A(I,2)-A(I,1)*A(I-1,3)/A(I-1,2)
      A(I,4) = A(I,4)-A(I,1)*A(I-1,4)/A(I-1,2)
10    CONTINUE
C
C.....BACK SUBSTITUTE
      A(N,5) = A(N,4)/A(N,2)
      N1=N-1
      DO 20 I=1,N1
      NI=N-I
      A(NI,5) = (A(NI,4)-A(NI,3)*A(NI+1,5))/A(NI,2)
20    CONTINUE
      RETURN
      END
```

Appendix IV

Comparison of Results from SUTRA and QSUTRA for Henry's (1964) Seawater Intrustion Problem [See Voss (1984, p. 196-203) for details on problem]



Comparison of concentration profiles for Henry's (1964) problem using QSUTRA and the original SUTRA codes.

Comparison of mass flux across the model boundary for Henry's (1964) problem using QSUTRA and the original SUTRA codes

[Fluid sources or sinks due to specified pressures]

QSUTRA		SUTRA	
Node	Inflow(+)/outflow(-) (mass per second)	Node	Inflow(+)/outflow(-) (mass per second)
221	2.0505180D-03	221	2.0445683D-03
222	3.9052976D-03	222	3.8945305D-03
223	3.6464678D-03	223	3.6372692D-03
224	3.1973050D-03	224	3.1903238D-03
225	2.4313601D-03	225	2.4270948D-03
226	1.0648645D-03	226	1.0642539D-03
227	-1.8302268D-03	227	-1.8264702D-03
228	-7.0298556D-03	228	-7.0211951D-03
229	-1.5196096D-02	229	-1.5180209D-02
230	-2.8955106D-02	230	-2.8933956D-02
231	-2.9209879D-02	231	-2.9197070D-02