Dissertations and Theses                                    Dissertations and Theses

1997

# A Telescopic Approach to Modeling Leachate Dynamics and Migration at the St. Johns Landfill, Portland, Oregon

Thomas Stephen Lowry
*Portland State University*

# THESIS APPROVAL

The abstract and thesis of Thomas Stephen Lowry for the Master of Science in Civil

Engineering were presented February $7^{th}$, 1997, and accepted by the thesis committee

and the department.

COMMITTEE APPROVALS:

_____

Dr. Shu-guang Li, Chair

_____

Dr. Scott Wells

_____

Dr. Ansel Johnson

_____

Dr. Marvin Beeson
Representative of the Office of Graduate Studies

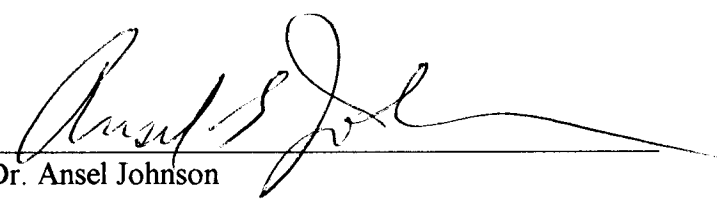DEPARTMENT APPROVAL:

_____

Dr. Franz Rad
Department of Civil Engineering

# ABSTRACT

An abstract of the thesis of Thomas Stephen Lowry for the Master of Science in Civil Engineering presented February 7th, 1997.

Title: A Telescopic Approach to Modeling Leachate Dynamics and Migration at the St. Johns Landfill, Portland, Oregon.

The St. Johns Landfill is a 225-acre site that lies near the confluence of the Columbia and Willamette Rivers, in the North Portland Industrial District of Portland, Oregon. A slough system surrounds the landfill on three sides with the fourth side adjacent to a shallow lake. The landfill first opened in 1930 when it was operated as an open dump. It operated as an open dump until 1969 when it was converted to a sanitary landfill, employing daily cover and compaction. The landfill was closed in 1991.

The landfill sits directly on top of a thick layer of silty overbank deposits. Directly below the overbank deposits lie a highly productive aquifer system. Concern has arisen about the landfill's impact on the underlying aquifers as well as the surface waters directly adjacent to the site. Past studies have hypothesized that the thick layer of overbank deposits provides a barrier to contaminant transport into the lower aquifers. The purpose of this project is to quantify the fate and transport of contamination exiting the landfill.

To properly assess the flow and transport dynamics, a telescopic approach is used which nests a series of six groundwater models; each designed to provide

uniquely specific information concerning the site. Contaminant mounding within the landfill due to rainfall provides the driving force for contaminant migration. The modeling system represents the quantitative inter-relationships between the landfill, the surface waters, and the underlying aquifer system.

The distribution of fluxes between the surface and groundwater is controlled by the anisotropy of the overbank deposits. Predictions show that the majority of the contaminants move laterally out of the landfill and into the surrounding slough system. The model also predicts that the underlying aquifer system has been impacted, but only in localized 'hot spots'. After the landfill is capped in 1991, contaminant migration is slowed due to the reduction of the contaminant mound. Future impact on the surrounding area at this point is due to regional flow dynamics controlled mainly by the regional recharge characteristics and the Columbia and Willamette Rivers.

A TELESCOPIC APPROACH TO MODELING

LEACHATE DYNAMICS AND MIGRATION

AT THE ST. JOHNS LANDFILL, PORTLAND, OREGON

by

THOMAS STEPHEN LOWRY

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
CIVIL ENGINEERING

Portland State University
1998

## Acknowledgements:

The St. Johns Landfill modeling project is a very large and complicated endeavor. By no means was it the work of just one person. This section is to acknowledge those who contributed and helped in making the project a reality.

The St. Johns Landfill modeling system was developed under the intergovernmental agreement between Portland State University and the Metropolitan Service District (METRO) of Portland, Oregon. Professor **Shu-Guang Li**, Ph.D., PE, of Portland State University was the principle investigator of the modeling study. **Joanna Karl**, PE, of the METRO solid waste department was the overall project manager. A separate final report in agreement of the contract exists outside of this thesis.

The water balance model, the particle tracking model, the two dimensional flow model (with the exception of the pre-processors), and the three dimensional contaminant transport model are all my own work. The local scale three dimensional flow model was developed by myself and **Christopher Berger**, who also produced the pre-processors for the two and three dimensional flow models. **Min Chen** and **Julie Peyton** originally developed the regional scale flow model. Extensive revisions to the regional flow model were conducted in the summer of 1997 by **Huang Quinsheng** and **Wenbin Xu,** whose results are presented here.

The advanced graphics and visualization system, was implemented by **Peter Gotseff** with **Juan-Juan Wang** providing assistance. **Mark Boyko** assisted with the

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

# 1. Introduction

## 1.1 Site Overview

The St. Johns Landfill is located in North Portland on the Columbia River floodplain near the confluence with the Willamette River. The landfill is immediately bounded by the Columbia Slough to the south and southwest, the North Slough and Bybee Lake to the north, and Smith Lake to the east (Figure 1). The landfill sits on top of, in sequence from young to old and top to bottom, overbank deposits (fine grained silt), the Columbia River sands, the Pleistocene sands and gravels, the Troutdale Formation, and the Columbia River basalts. The overbank deposits are mostly low permeability silty clays, and silts. These silty clays (referred to as the silts) serve as a natural bottom liner that separates the landfill from the deep aquifer system. The silts also comprise a natural surrounding dike that provides horizontal separation between the landfill and the surrounding surface waters.

Originally the 183 acre site was an un-named wetlands and seasonal lake. Since the early 1930's, the landfill was operated as an open municipal and industrial dump with no daily covering or compaction. By the mid 1950's, the lake/wetlands had been completely filled. In 1969, the landfill was converted to a sanitary landfill utilizing daily cover and periodic compaction, accepting mostly municipal waste (Figure 2). In 1980, an additional 55 acres, underlain by a contaminant collection system and separated from the surrounding surface waters by a man-made engineered dike, was added to the landfill along its northeastern edge. The landfill was finally closed in 1991 due to environmental concerns.

1
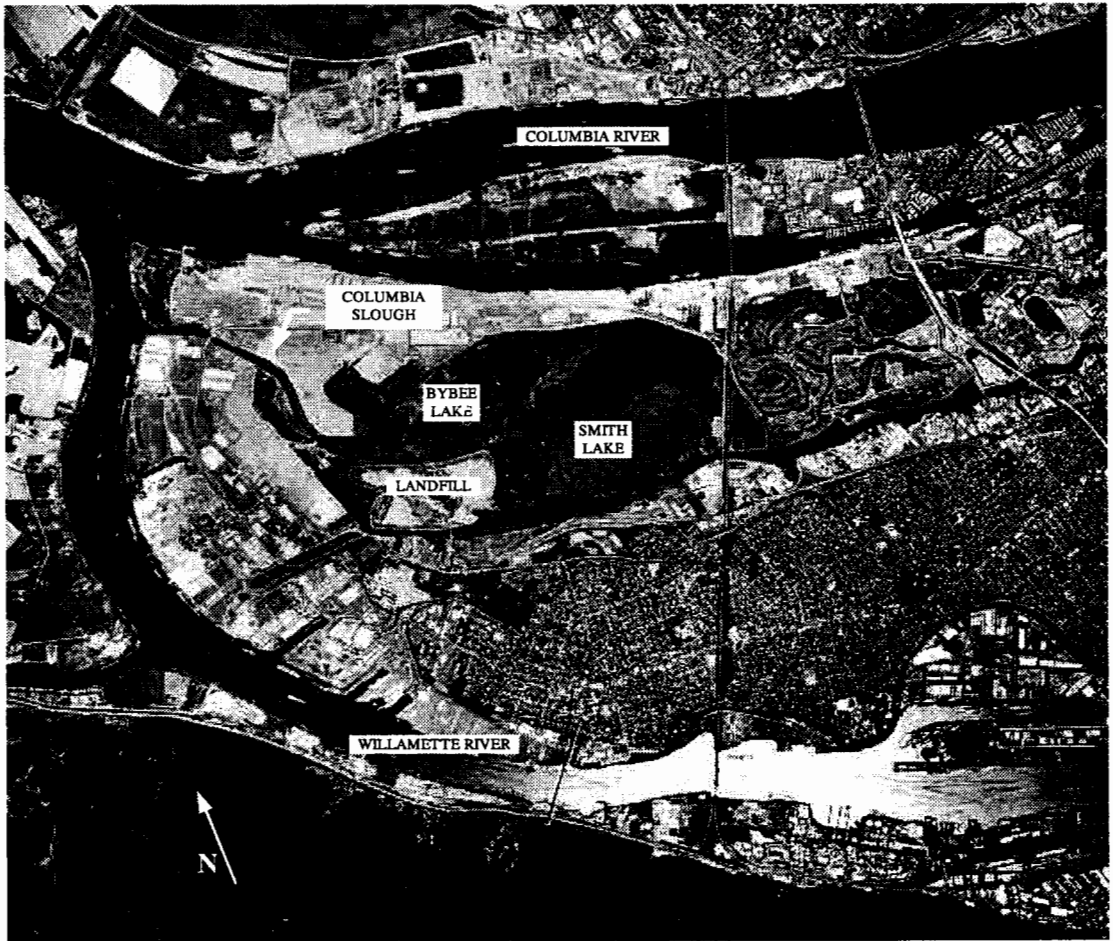
**Figure 1 - Plan view of St. Johns area with the Columbia River to the North and the Willamette River to the West and South. The Columbia Slough drains from the east, around the south side of the landfill, and into the lower Willamette.**
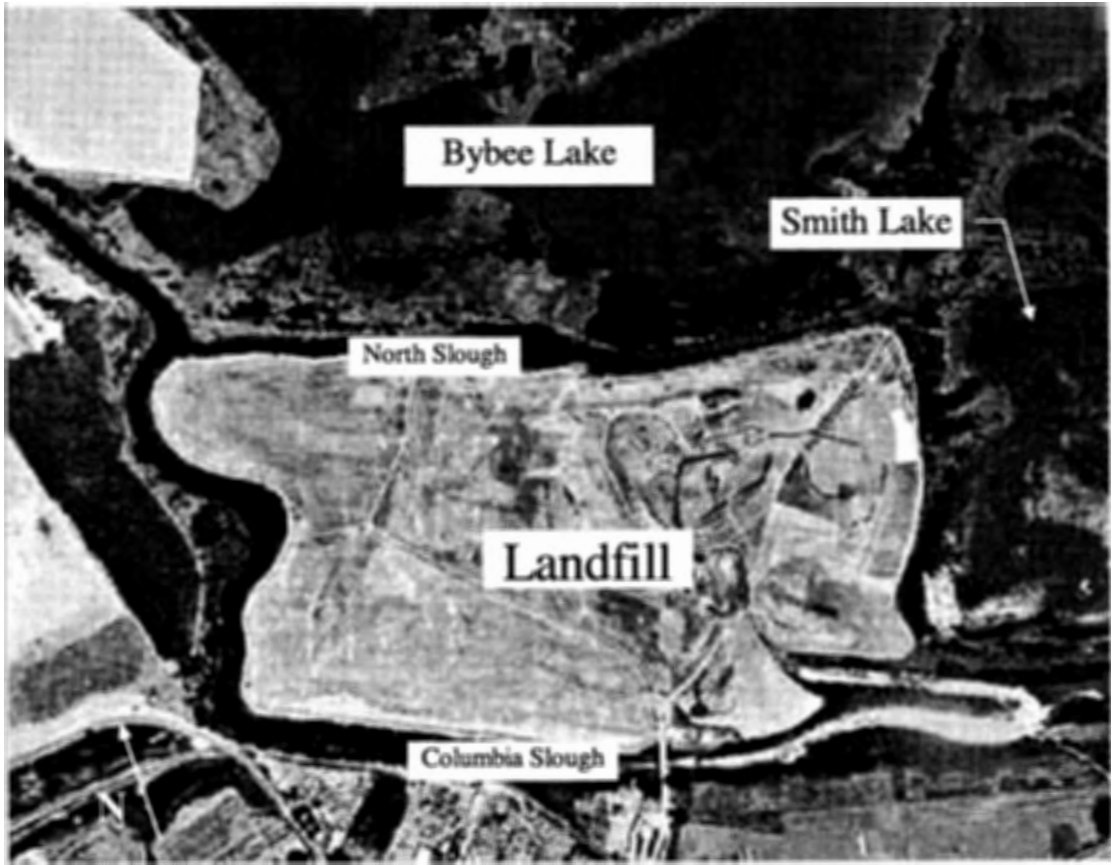
Figure 2 - Plan view of local St. Johns Landfill site. The Columbia Slough is to the west and south and the North Slough is to the north. Bybee Lake is connected to the North Slough by a one-way valve and weir (located just below the caption pointing to Smith Lake). Natural flow would allow for tidal fluctuations in Bybee Lake but the valve/weir system maintains an average 2-3 foot higher level in Bybee Lake than in the North Slough.

Figure 2 - Plan view of local St. Johns Landfill site. The Columbia Slough is to the west and south and the North Slough is to the north. Bybee Lake is connected to the North Slough by a one-way valve and weir (located just below the caption pointing to Smith Lake). Natural flow would allow for tidal fluctuations in Bybee Lake but the valve/weir system maintains an average 2-3 foot higher level in Bybee Lake than in the North Slough.

## 1.2    Questions and Issues

Threat to the surrounding surface waters and groundwater exists due to contaminant movement from the landfill.  As a landfill is built, rainfall accumulates in the refuse creating a mound of water which leaches contaminant from the refuse.  This mounding provides a driving force for advective flow of contaminant to enter the groundwater downward into the silts and towards the gravels, as well as laterally through the protective silt dikes towards the Sloughs.  As the mounding builds, the driving force increases until the flow out of the landfill equals the rainfall seepage into the landfill.  When inflow due to rainfall seepage equals the downward and vertical outward flux, the mounding stops growing and the landfill is in a 'quasi' steady state condition.  As the landfill is capped, and the rainfall seepage is reduced, the mounding dissipates until the outflow from the landfill is equivalent to the reduced, post-closure rate of rainfall seepage.  Diffusive flow will always continue assuming a concentration gradient exists between the landfill and the lower aquifers.

The purpose of this study is to quantify the environmental impact of the St. Johns Landfill on the adjacent surface and groundwater systems.  Principal questions of concern are:

- Is the landfill polluting the adjacent surface and groundwater systems?

- What is the contribution of the St. Johns Landfill to the already polluted Columbia and North Sloughs?

- How effective is the silt at protecting the deep aquifer system from contaminant migration?

- How long does it take for contaminant to penetrate the bottom silt layer?

- Has offsite migration occurred?

- How much contaminant goes into the groundwater?

- What is the extent of groundwater contamination in the gravel aquifer?

- How fast is the contaminant plume migrating? Where will it be 25 years from now?

- Is the landfill affecting Smith and Bybee Lakes?

- How will the proposed management plan of restoring Bybee Lake to intertidal conditions effect contaminant plume migration in the aquifer?

## 1.3   Project Approach

The St. Johns Landfill modeling system attempts to combine site specific field data with a physically based, systematic model in order to address the issues of the project. The field data allow for the use of site specific conditions within the model. Physical layout, stratigraphy, hydrology, chemistry, and long term weather patterns all effect plume migration. The project model effectively utilizes the field data which describe these parameters and allow us to construct a complete synoptic picture of a three-dimensional, time dependent, heterogeneous system at field scale.

In modeling the plume, a conservative contaminant (i.e., chloride) is used as the indicator. Modeling a conservative contaminant describes the worst case scenario for plume migration of any type of contaminant. This is due to the fact that sorptive plumes will only equal conservative plumes once the sorptive capacity of the soil is used up which effectively slows the migration of the plume, and chemically reactive

substances will decay to lesser concentrations. The representative conservative plume can then be used as a base for scaling the plume to meet other conditions.

## 1.4  Regional and Site Geology

The aquifer framework for the model was constructed based on stratigraphy and subsurface maps provided by the Metropolitan Service District, Portland, OR (METRO, 1994).  The geology framework accounts for 6 distinct layers, each with its own characteristics.  Those layers are the landfill refuse, the silt, the sand, the gravels, the Troutdale Formation (which includes the Troutdale aquifer and the undifferentiated sediments), and the Columbia River basalts.  Each is explained in detail below.  Two cross sections, one along a north to south line (A to A', Figure 3) and another along an east to west line (B to B', Figure 3), are shown below in Figure 4 and Figure 5.
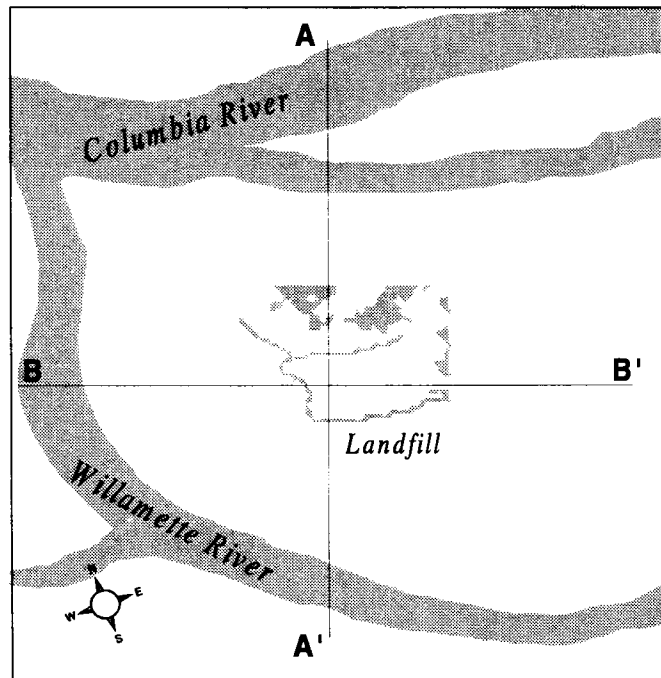


**Figure 3 - Plan view of landfill showing cut lines for north to south and east to west framework views in Figure 4 and Figure 5.**
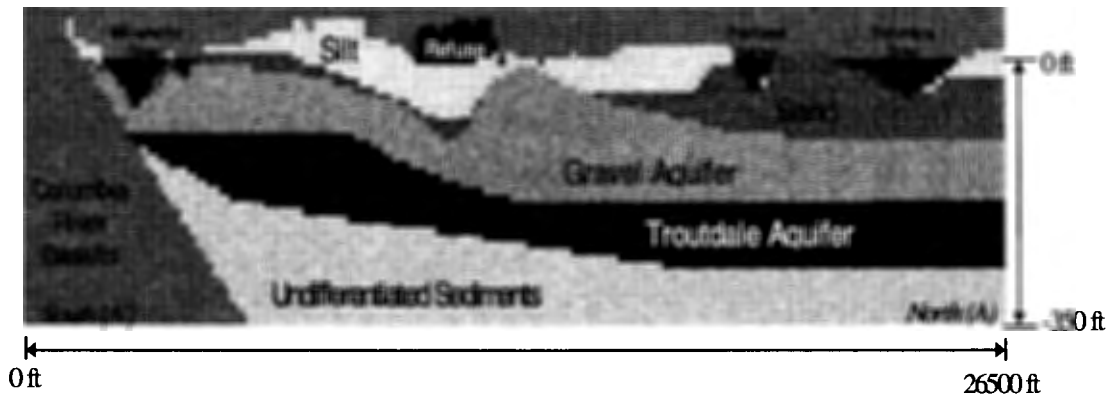
**Figure 4 - Aquifer framework cut on a north to south line through the center of the landfill.  The gravel ridge to the north of the landfill can be clearly seen here. Note the thickness of the sand formation underneath the Columbia River.  The undifferentiated sediments and Troutdale aquifer are both part of the Troutdale Formation.**
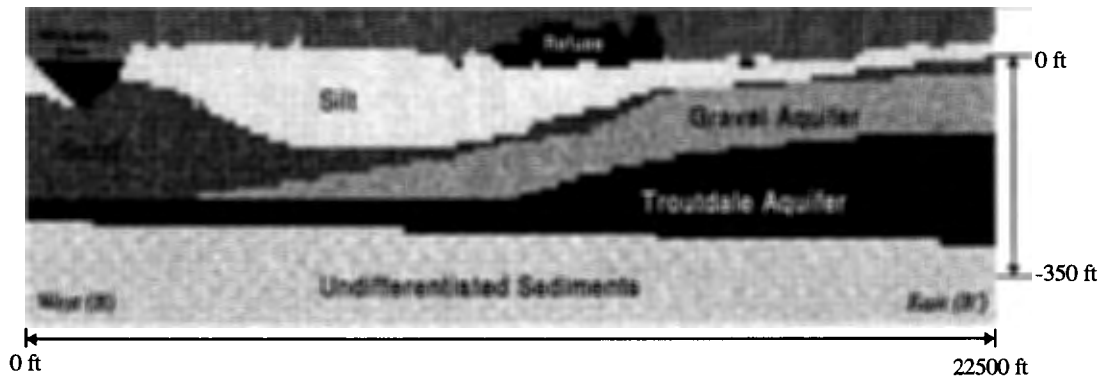


**Figure 5 - Aquifer framework cut on a east to west line through the center of the landfill.  The gravel aquifer pinches out towards the west and is replaced by a thick silt layer and eventually the thick sands underneath the Willamette River. The undifferentiated sediments and the Troutdale aquifer are both part of the Troutdale Formation.**
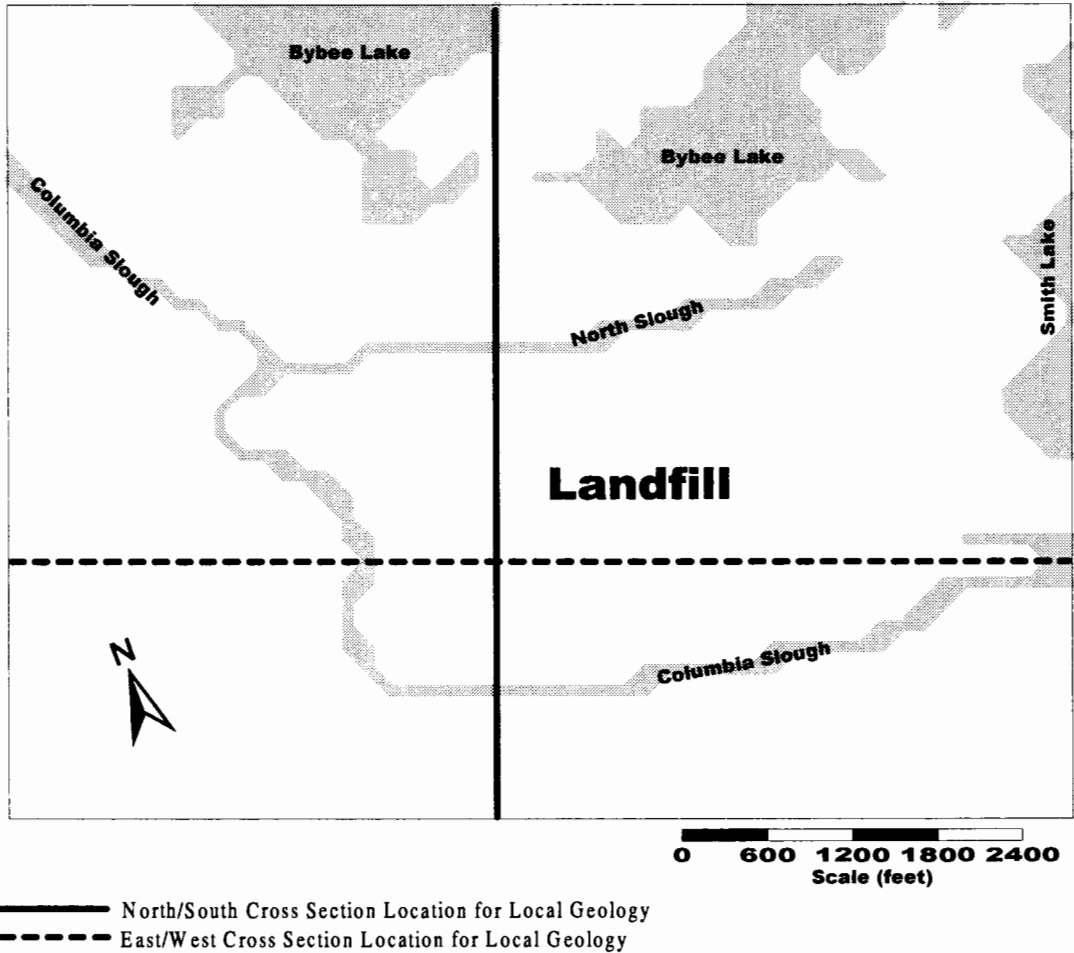
Figure 6 - Plan view of local landfill area showing the cross section locations for Figure 7 and Figure 8.
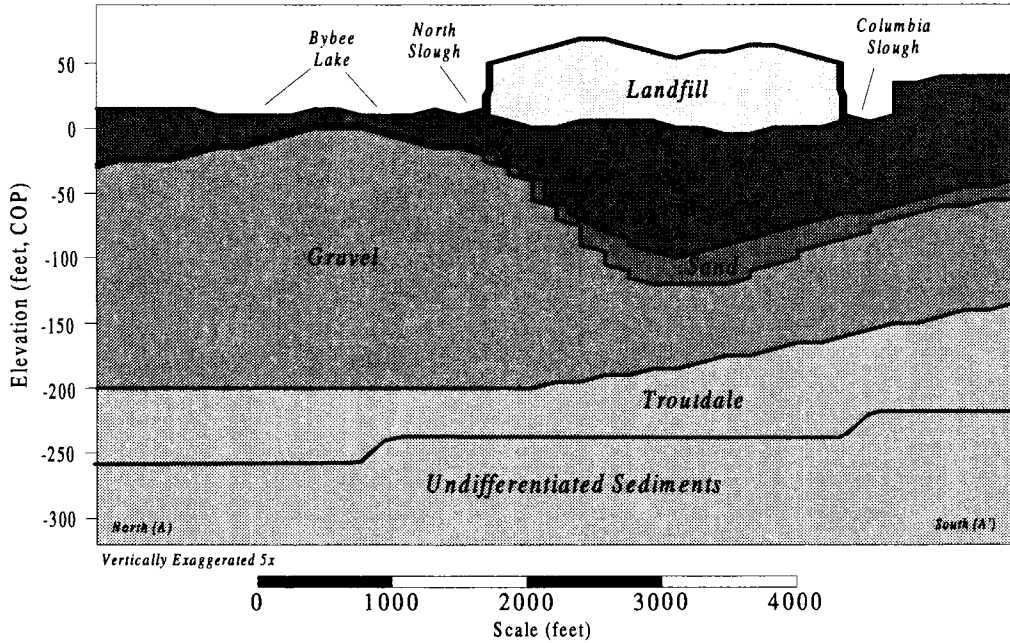
**Figure 7 - Geology in local landfill area along north/south cut. Note view is 180° from regional geology view. Gravel ridge is clearly seen in this cross section. An important feature of the geology is the rapidly varying thickness of the silt layer across the width of the landfill. Sand layer is very thin in this area compared to the silt and the gravel.**
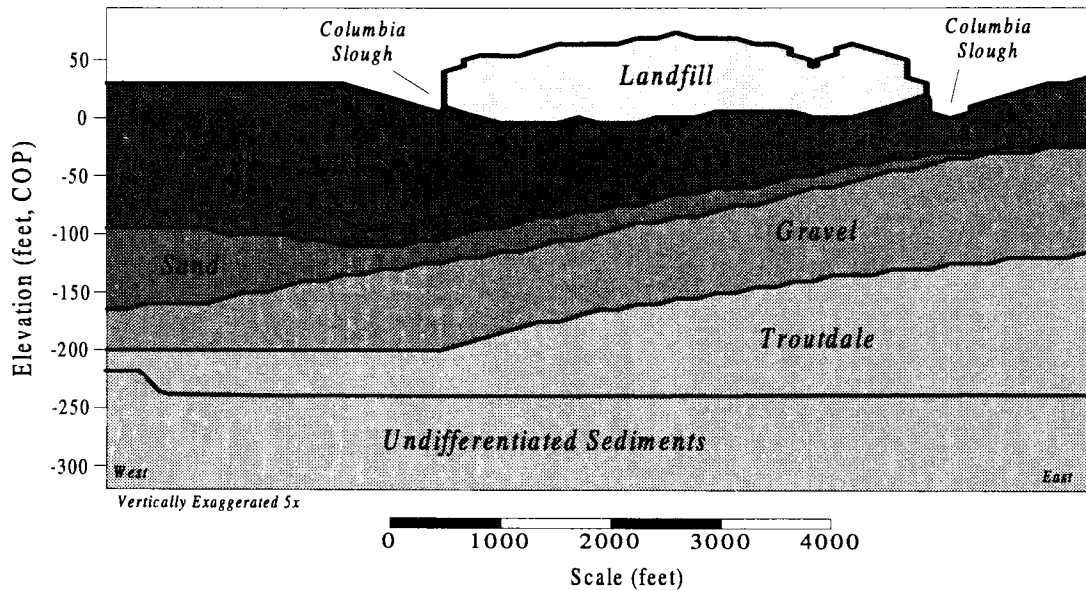


**Figure 8 - Geology in local landfill area along east/west cut. Note view is 180° from regional geology view.**

10

### 1.4.1 Refuse:

Prior to 1969, the St. Johns Landfill received both industrial and municipal waste. At its conversion to a sanitary landfill, the bulk of the dumping was from municipal waste. In addition, the operation of the landfill as a sanitary site has created many varying layers between the daily cover and the refuse. This has created a site which is highly heterogeneous and has a low to moderate permeability. Typical hydraulic conductivities (Oweis, et al, 1990) for refuse range from $K_{land} = 1.0 \times 10^{-2}$ to $1.0 \times 10^{-4}$ cm/sec, and the typical range for specific yield in landfills is $S_y = 0.1$ to $0.3$.

### 1.4.2 Floodplain Sediments:

The Columbia River floodplain sediments consist of clays and clayey silts, fine grained silt, and fine grained sands with low permeabilities. With the exception of the area to the extreme south of the landfill where the Portland West Hills rise approximately 1000 feet above the flood plain sediments, the silt defines the local topography of the region (Figure 9). Published values for the horizontal silt hydraulic conductivities at the site fall in the range of $K_{siltH} = 1.3 \times 10^{-6}$ to $3.1 \times 10^{-5}$ cm/sec(Sweet-Edwards/Emcon, 1989), with vertical values ranging from $K_{siltV} = 2.0 \times 10^{-7}$ to $5.0 \times 10^{-7}$ cm/sec (METRO, 1994). General agreement is shown in a report by CH$_2$M Hill in 1979 of $K_{siltV} = 6.0 \times 10^{-7}$ cm/sec. The silt thickness varies dramatically in the local landfill area ranging from 30 to 140 feet thick directly below the landfill (Figure 11). To the north, a large ridge in the underlying gravel aquifer pinches the silt to just a few feet thick directly below Bybee Lake (Figure 10). This gravel ridge and the corresponding thin layer of silt have implications on long-term

plume migration. The hydraulic properties and thickness of the silt layer are the defining properties which dictate the environmental impact of the St. Johns Landfill by acting as a confining layer to contaminant transport and controlling localized flow patterns.
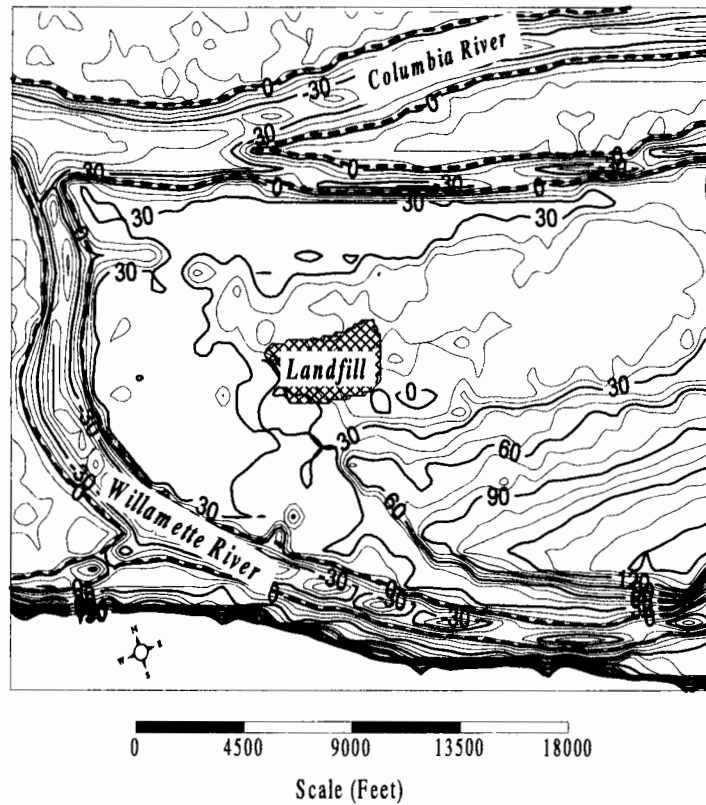
**Figure 9 - Top elevation of silt layer. The landfill is shown as the cross-hatched region in the center of the picture. The Columbia and Willamette Rivers are outlined by the dark dotted line. The top of the silt generally follows local topography except to the south east in the St. Johns heights area as well as to the extreme south of the study area where the Portland west hills rise above the lower lying floodplain.**
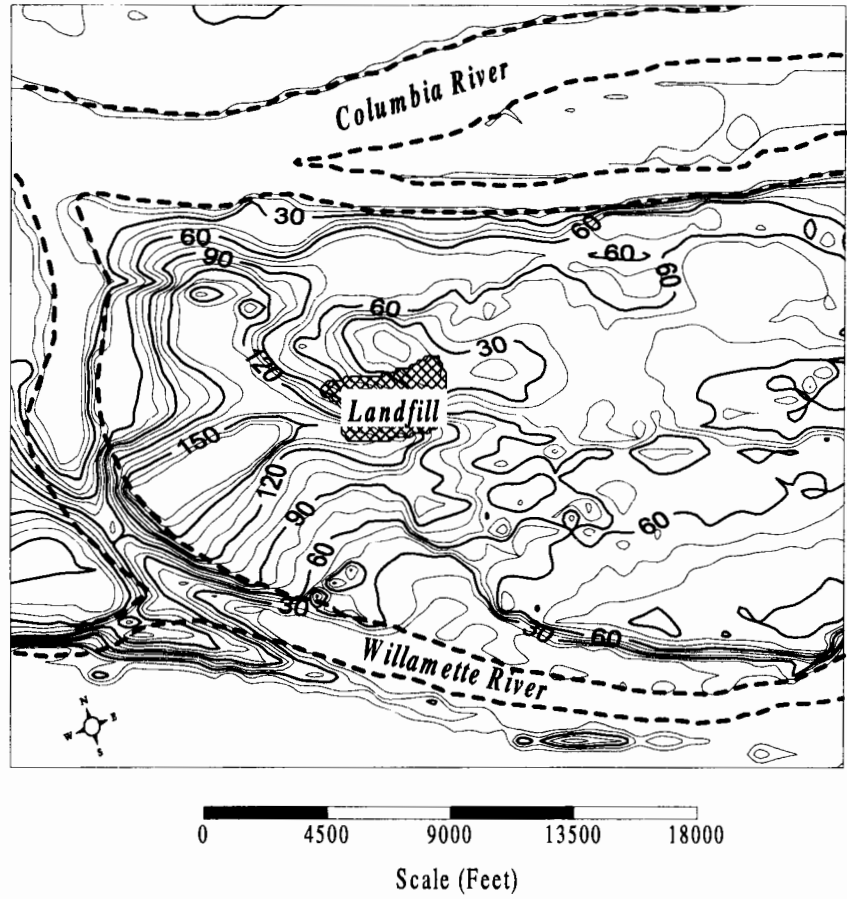
**Figure 10 - Thickness of silt layer.** The landfill is shown as the cross-hatched region in the center of the picture. The Columbia and Willamette Rivers are outlined by the dark dotted line. Note that the silt is thick to the west of the landfill and is thinnest in the Bybee Lake area (directly north of the landfill).
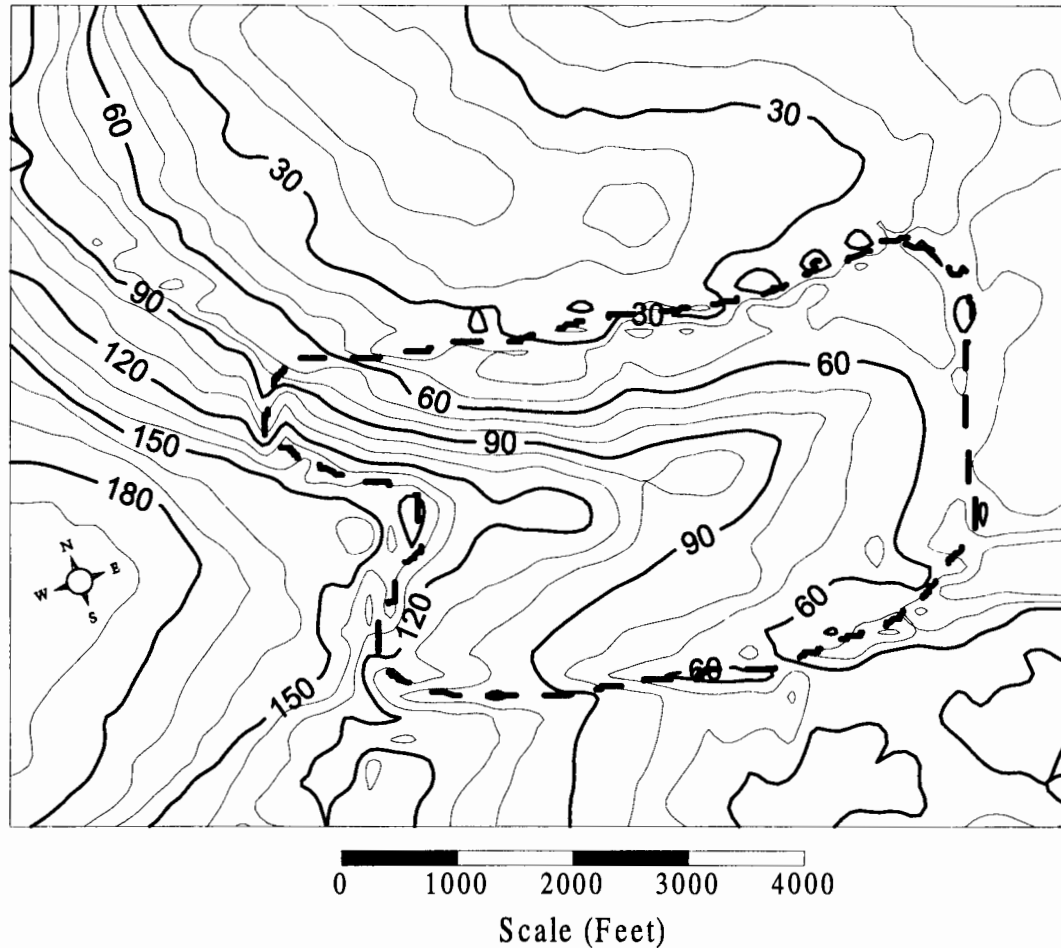
**Figure 11 - Thickness of silt layer in the local landfill area. The landfill (shown by the black dashed line) sits over a thick trough of low-conductivity silt. Along the north edge of the landfill, the silt is only 30-40 feet thick. The lack of silt underneath Bybee Lake (to the north of the landfill) is a concern due to lake-aquifer interactions.**

### 1.4.3 Columbia River Sands:

The Columbia River sands are deposited mostly beneath the Columbia River and the northern part of the Willamette River (Figure 12 and Figure 13). Little or no sand is present below the landfill and what is present only exists as small discontinuous lenses within the silt layer. The sands are of moderate conductivity as determined by a pump and slug test at the Columbia Steel site (1992) $K_{sand} = 1.2 \times 10^{-2}$ cm/sec, and the Sweet Edward's/Emcon (SE-E) rising head test which shows $K_{sand} = 5.1 \times 10^{-5}$ cm/sec (appears to be too small). The sands are one of the controlling factors in the regional flow because they lie in the path of the overall flow direction towards the north, creating a restrictive element for the communication between Columbia Rivers and the highly conductive gravel aquifer.
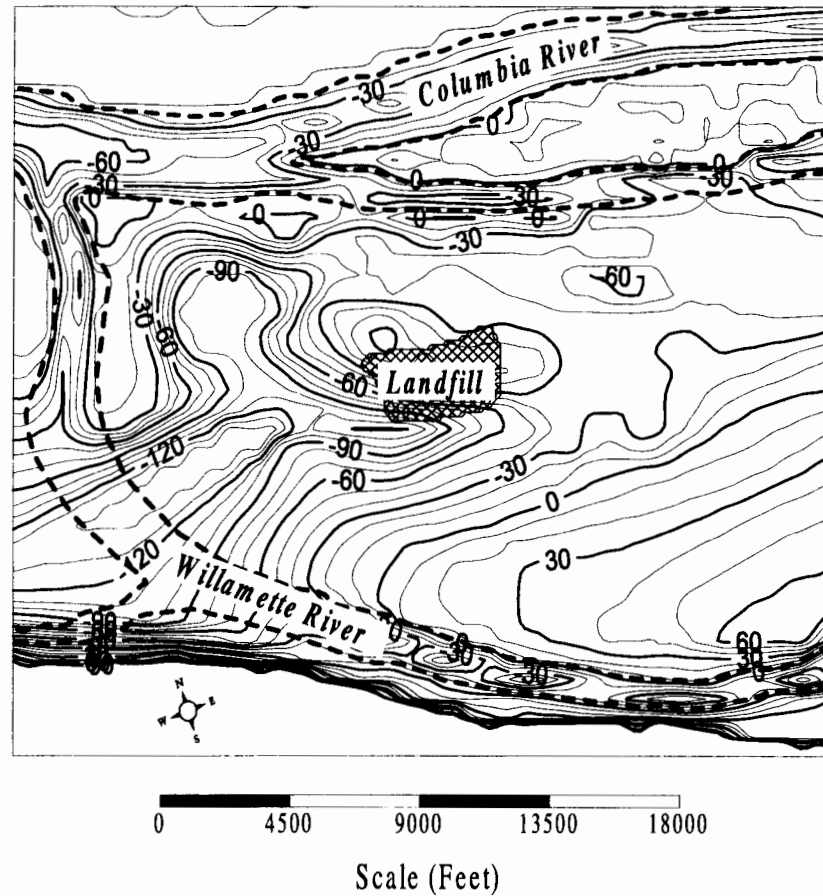
**Figure 12 - Top elevation of sand layer. The landfill is shown as the cross-hatched region in the center of the picture. The Columbia and Willamette Rivers are outlined by the dark dotted line. Note the deep trough to the west of the landfill, which is mainly filled with silt.**
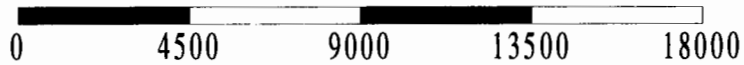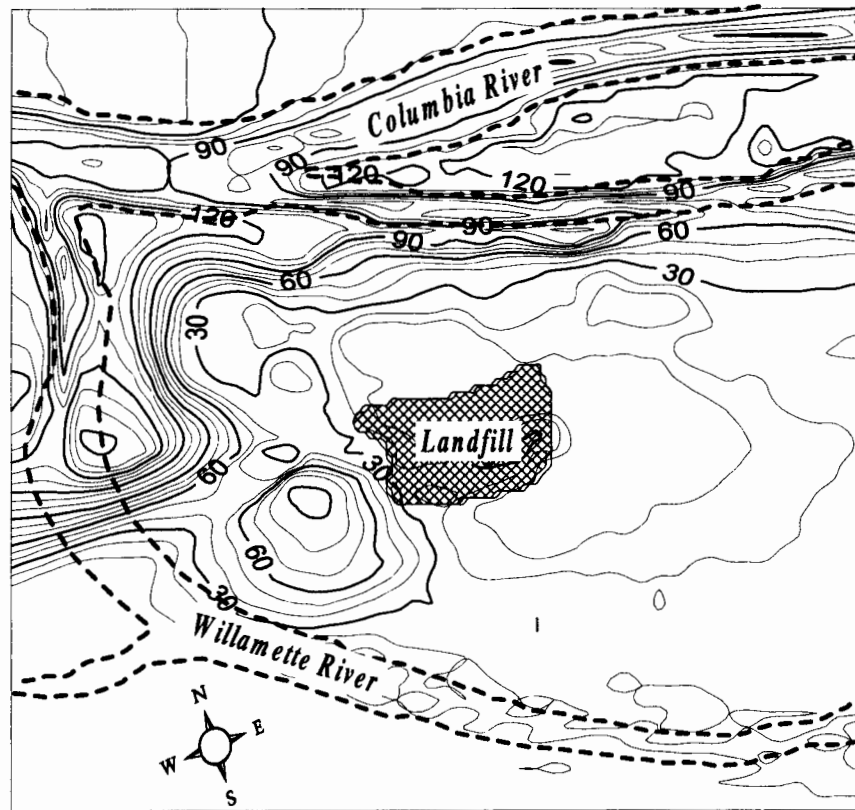
**Figure 13 - Thickness of sand layer. The landfill is shown as the cross-hatched region in the center of the picture. The Columbia and Willamette Rivers are outlined by the dark dotted line. The sands are deposited mostly in the Columbia River area with very little sand in the landfill and Bybee Lake area.**

18

### 1.4.4 Pleistocene Gravels:

The gravels are a highly conductive unconsolidated aquifer, hydrologically connecting the Columbia and Willamette Rivers (Figure 14 and Figure 15). The aquifer is used for water source purposes with about a dozen wells in the area (Sweet-Edwards/Emcon, 1989). Hydraulic conductivity for the gravels range from $K_{gravel}$ = $5.3 \times 10^{-2}$ to $1.4 \times 10^{-1}$ cm/sec (Sweet-Edwards/Emcon, 1989). The landfill sits directly over a trough in the gravels and is separated from the gravel by the silts. To the north of the landfill, the gravel forms a ridge running approximately east to west, which comes within a few feet of the surface directly below Bybee Lake. Because of the high conductivity within the gravels, the landfill's impact on the gravel is of great concern. If contaminant is able to move into the gravels, migration of the contaminant with the regional flow may provide a pathway into Bybee Lake through the thin part in the silts.
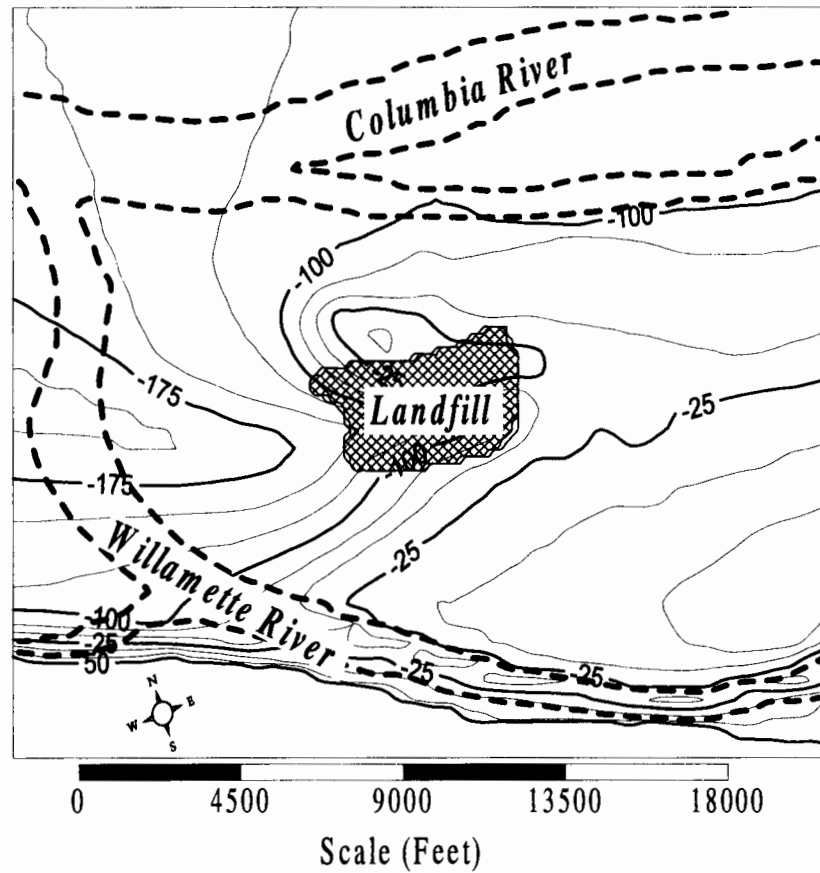
**Figure 14 - Top elevation of gravel aquifer. The landfill is shown as the cross-hatched region in the center of the picture. The Columbia and Willamette Rivers are outlined by the dark dotted line. Note the large trough to the west of the landfill site and the ridge that is to the north of the site. Both these features play large roles in the regional groundwater flow.**
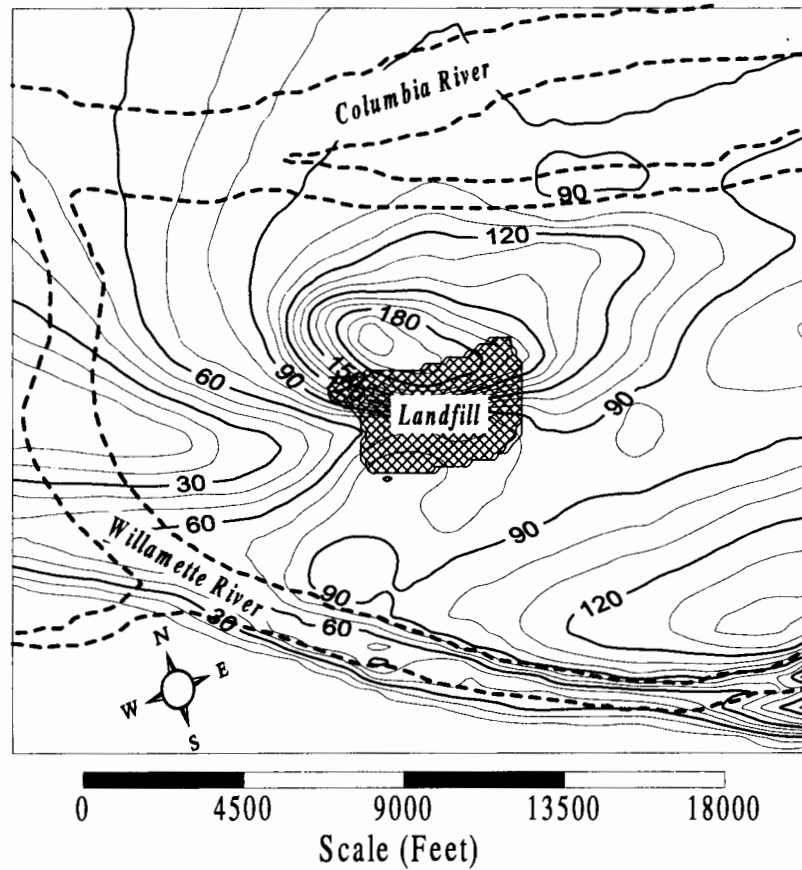
**Figure 15 - Thickness of the gravel aquifer. The landfill is shown as the crosshatched region in the center of the picture. The Columbia and Willamette Rivers are outlined by the dark dotted line. Note the trough to the west of the landfill where the gravel layer 'pinches' out. Also, a ridge north of the landfill where the gravel is the thickest comes within feet of the surface directly below Bybee Lake. Both these features have a large impact on the regional and local flow patterns.**

### 1.4.5 Troutdale Formation:

No data exist in the landfill area that characterizes the Troutdale Formation (Figure 16 and Figure 17). The formation, consists of consolidated gravels with moderate permeability (Fishman, 1986). Vertical communication between the Troutdale Formation and the gravel aquifer is slight (USGS, 1994).
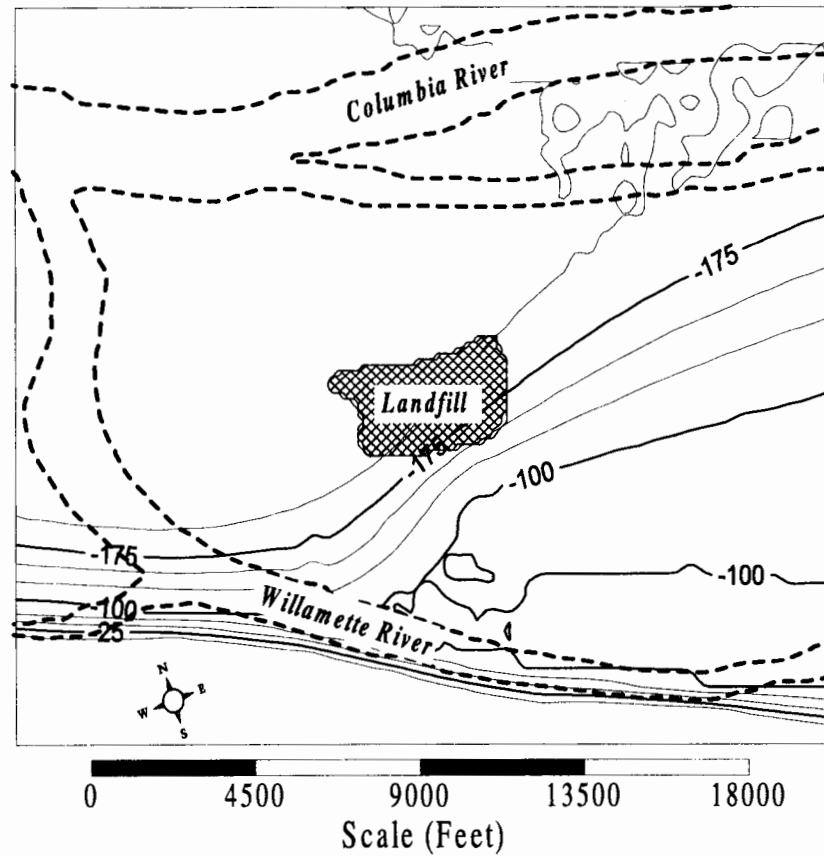
**Figure 16 - Top elevation of Troutdale Formation. The landfill is shown as the crosshatched region in the center of the picture. The Columbia and Willamette Rivers are outlined by the dark dotted line. The top of the formation is slanted upward from north to south until it meets the Columbia River basalts which rise from the ground to form the Portland West Hills in the extreme southern portion of the study region.**
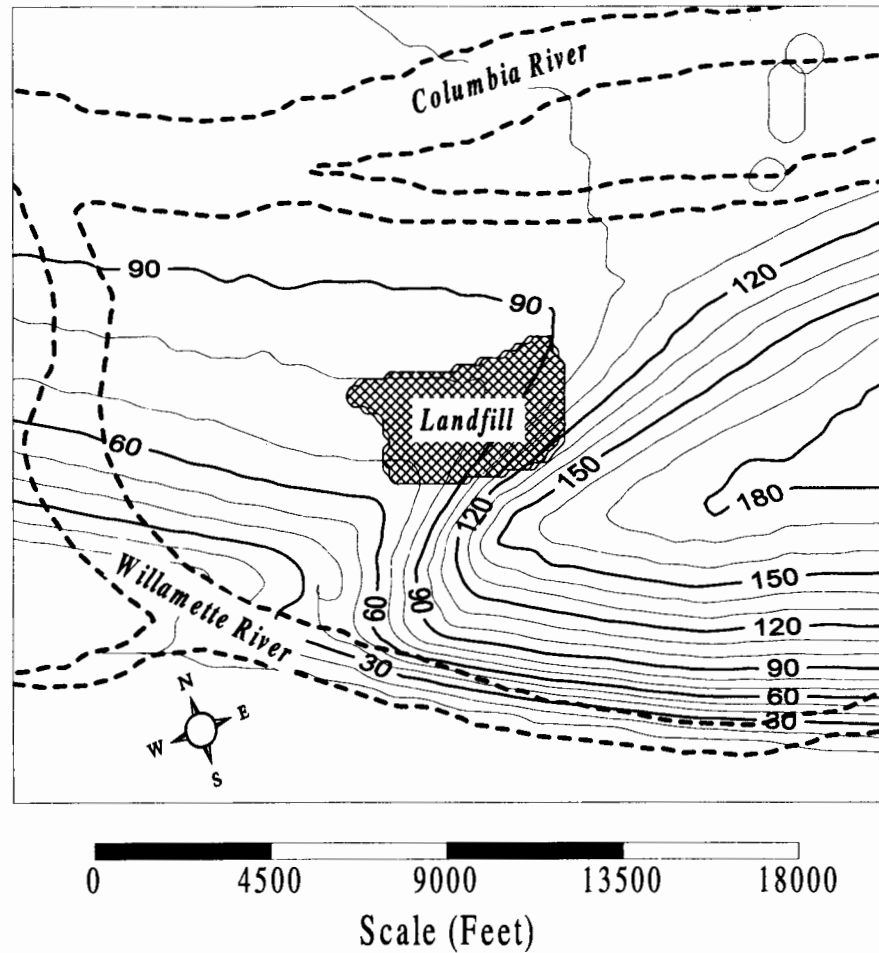
**Figure 17 - Thickness of Troutdale Formation. The landfill is shown as the cross hatched region in the center of the picture. The Columbia and Willamette Rivers are outlined by the dark dotted line.**

### 1.4.6 Columbia River Basalts:

The basalts underlie most of the lowland areas of northwestern Oregon. They are mainly defined by very low permeability rock, criss-crossed by fracturing and layering that enables movement of groundwater. Since the Columbia River basalts plunge very deeply near the landfill, they have little effect on the modeling domain nor does it appear they are threatened in any way by the landfill.

### *1.5* *Field Data Analysis:*

The St. Johns Landfill site has been studied for approximately the last 25 years. The majority of the monitoring wells lie along the perimeter of the landfill. This layout limits the ability to characterize plume movement directly below the landfill. The data allow for adequate characterization of the site as well as a look at many different aspects of the landfill dynamics and plume migration. This section will review a few of the prominent studies and outline their key points.

### 1.5.1 Stevens, Thompson, and Runyan, 1974:

Seventeen monitoring wells, grouped into three categories depending upon their screen depth, were installed in 1972. The A wells are screened in the refuse, the B wells are screened in the silts near the bottom of the landfill, and the C wells are shallow perimeter wells, again in the silt near the bottom of the landfill. Figure 18 shows the landfill and the position of the monitoring wells.

The monitoring in 1972 showed that the contaminant mound had already formed due to rain percolation with a mound height in the landfill of 22-28 feet above

the City of Portland datum (COP), or approximately 50% of the depth of the refuse. Contaminant indicator (chloride) concentration in the wells showed to be about 100-1200mg/l which was much higher compared to the background level of 5-20 mg/l (Sweet-Edwards/Emcon, 1983, Figure 19). The study recommended long term monitoring at the site.
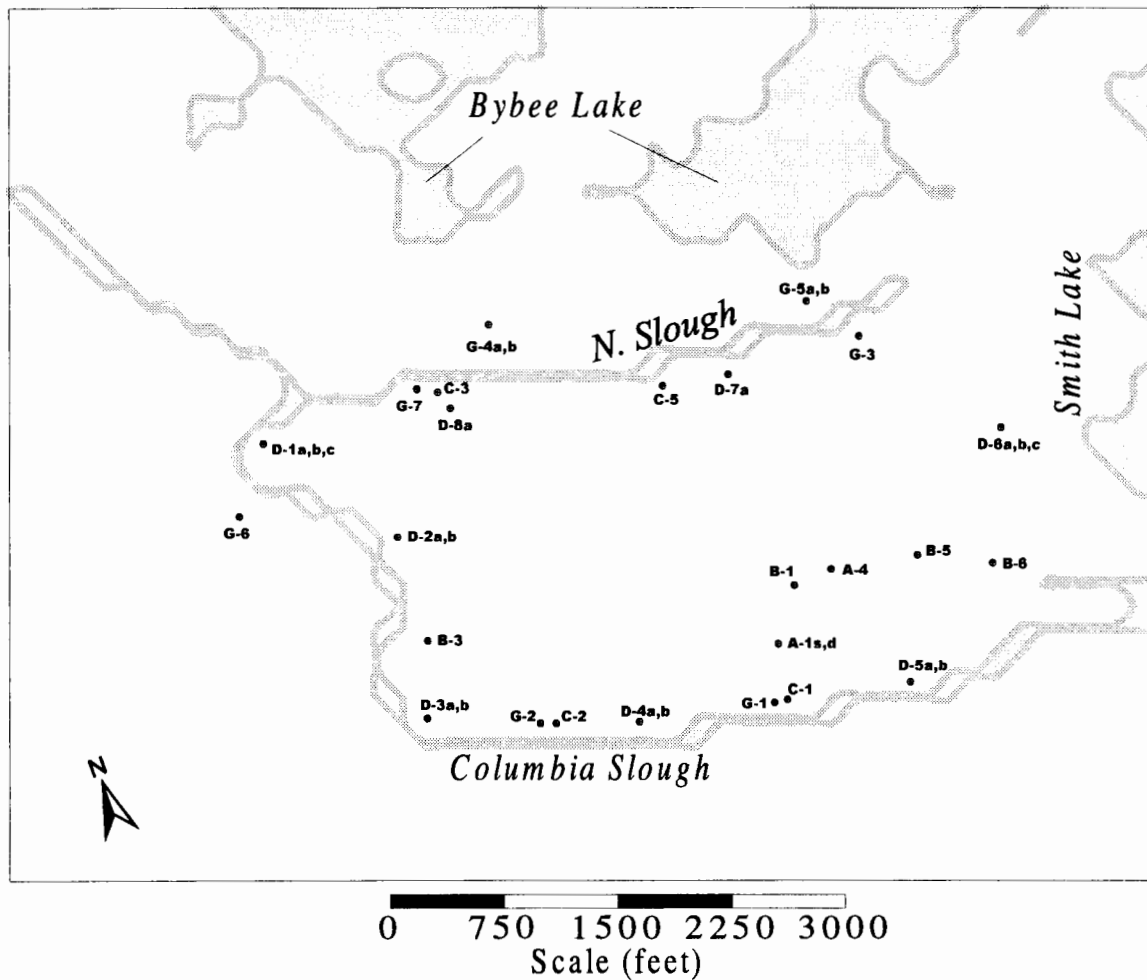
**Figure 18 - Monitoring well layout for the A, B, C, D, and G wells in the St. Johns Landfill area. Not all wells are currently being monitored.**

## 1.5.2  Sweet Edwards, 1983:

Sweet Edwards in 1983 presented a summary of all field data collected from 1972 to 1983. Field data showed that the contaminant mound appeared to have stabilized by 1972, at an elevation of approximately 20-30 ft (COP) depending on the location in the landfill, with variation between wet and dry seasons of 4-8 ft. The A wells screened in the interior of the refuse showed high concentrations of chloride with no obvious trend over time or depth. The B wells, screened in the silt near the bottom of the landfill in the landfills interior showed a strong positive chloride concentration trend over time which could indicate contaminant migration. The chloride concentration in the C wells, which are screened in the silt along the perimeter of the landfill varied considerably with location. Wells C1, C4, and C5 showed no obvious chloride concentration trend over time. Well C2 varied over the time period from 250-1500 mg/l chloride, showing a strong positive trend over time, and well C3 showed a range of 100-300 mg/l but with a slight negative trend over time. Figure 19 plots chloride concentration in each of the A, B, and C wells over time.

Sweet Edwards hypothesized in this report that the gravel aquifer was protected by groundwater upwelling from the Troutdale Formation.
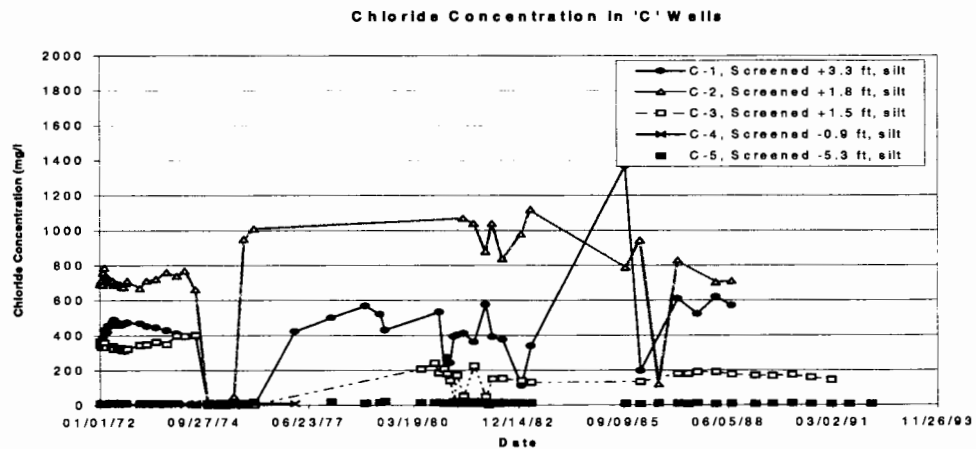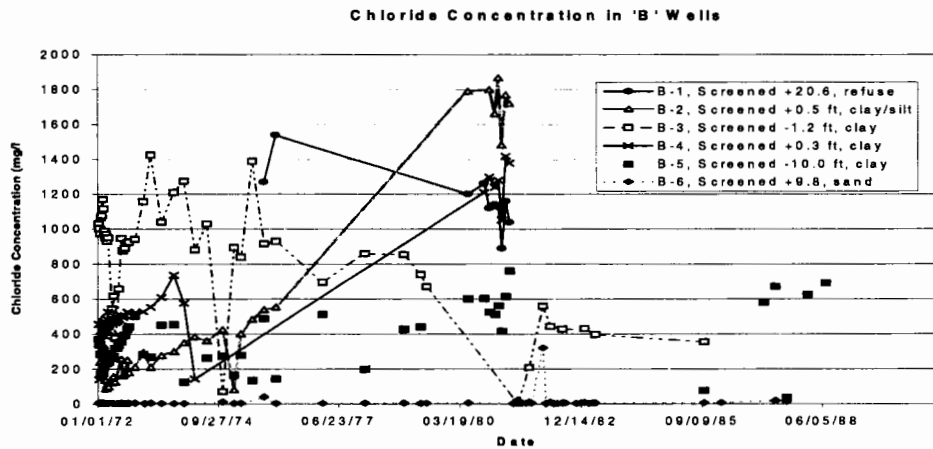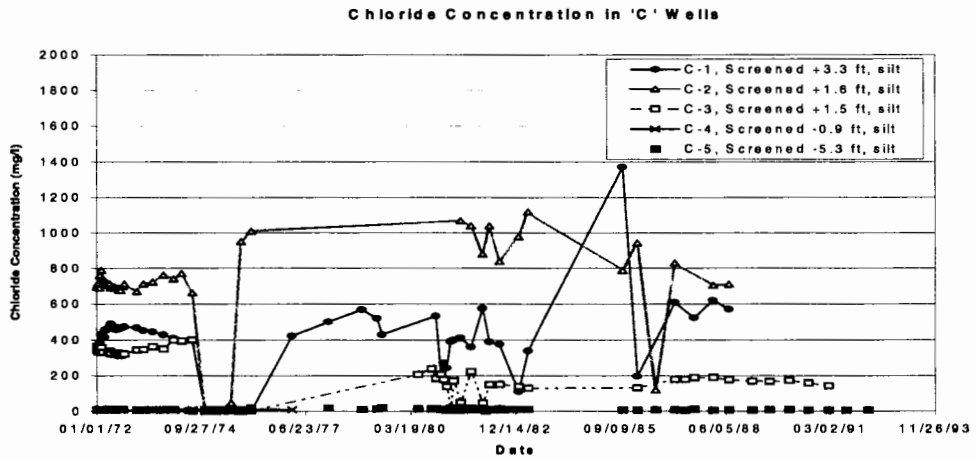
**Figure 19 - Monitoring results for the A, B, and C wells showing chloride concentrations over time. The A, B, and C wells are screened in the refuse or just below the refuse in the silt, and show high chloride concentrations over time with no distinct trend.**

29

### 1.5.3 Fishman, 1986:

Eight more monitoring wells (the D wells) were installed in 1985. These are multiple completion wells (several boreholes at the same spot finishing at varying depths) and go deeper into the silts and gravels than do the A, B, or C wells.

Chloride concentrations in the D wells (Figure 20) on the north side of the landfill seem to be generally higher than elsewhere in the landfill and showed a definite trend of decreasing concentration with depth. Contaminant had penetrated to more than a 40 foot depth in two of the six multiple completion perimeter wells (D8a and D3b) over the 30-50 years of the landfills existence. Both wells showed high chloride concentrations as compared to the rest, with D8a showing 425 mg/l and D3b showing 161 mg/l. D8a was later abandoned and backfilled because of concern that it may have been creating a preferential path for contaminant migration to the gravel aquifer. Other wells (D1c, D5b, D6c) showed near background levels on the landfill perimeter possibly indicating little or no offsite migration at these three points. A recommendation was made for further offsite monitoring.

**Chloride Concentration in 'D' Wells**

D-1A, Screened -12.0 ft, clay
D-1B, Screened -41.5 ft, clay
D-1C, Screened -83.0 ft, silt
D-2A, Screened -15.0 ft, clay
D-2B, Screened -44.0 ft, silt

Chloride Concentration (mg/l)

Date

**Chloride Concentration in 'D' Wells**

D-3A, Screened -12.5 ft, clay
D-3B, Screened -42.0 ft, clay/silt
D-4A, Screened -7.0 ft, clay/silt
D-4B, Screened -39.0 ft, clay/silt
D-5A, Screened -10.5 ft, clay/silt
D-5B, Screened -40.5 ft, clay/silt

Chloride Concentration (mg/l)

Date

**Chloride Concentration in 'D' Wells**

D-6A, Screened -2.0 ft, clay
D-6B, Screened -2.5 ft, clay
D-8C, Screened -80.5 ft, sand/gravel
D-7A, Screened -31.5 ft, sand/gravel

Chloride Concentration (mg/l)

Date

**Figure 20 - Chloride concentration in the D wells plotted over time. Wells D-3A and D-8A show increasing trends in concentration over time, while D-3B and D-4A show a decreasing trend. Most D wells indicate landfill contamination in the silt, decreasing with depth.**

31

### 1.5.4 Sweet Edwards/Emcon, 1989:

Deep perimeter wells (the G wells) were drilled off-site in the sands and/or gravels (G8bc, G4b, G5b, G6, and a Port of Portland well) with one on site (G7) next to the old D8a location (see Figure 18). High chloride concentrations (Figure 21) were found in well G7 (435-636 mg/l) indicating that the high concentrations found in well D8a might not have been due to contaminant seeping down the well casing.

Well G4b, which is located approximately 300 feet downgradient across the North Slough, screened in the sand and gravel, showed chloride concentrations at background levels. Near background levels were also observed in all other offsite wells (G8c - sand and gravel, G5b - sand and gravel, G6 - sand and gravel, D1c - silt and clay, D6c - sand, and the Port of Portland well - sand and gravel).

Work was done to establish the regional flow pattern in the area. Monthly water level measurements collected in 1988 and 1989 from the six off site wells showed a flat mean head gradient in the region of the landfill. Flow direction changes seasonally from northwesterly when the Columbia and Willamette River levels are low to southeasterly when the river levels are high (mainly Dec-Mar). The head distribution below the landfill is based on the six perimeter well head measurements. Sweet Edwards/Emcon also suggests that molecular diffusion dominates contaminant transport in the deep aquifer and that dispersion is unimportant due to the low velocities in the area.

Slough flux was estimated by a field trench test dug along the north side of the landfill. This resulted in a seepage flux of 1500 cubic feet/day (cfd - 0.017 cubic feet per second, cfs) representing 10% of the total recharge going into the sloughs. Flux to the gravel aquifer is implied by this to be 90% of total recharge.

**Chloride Concentration in 'G' Wells**
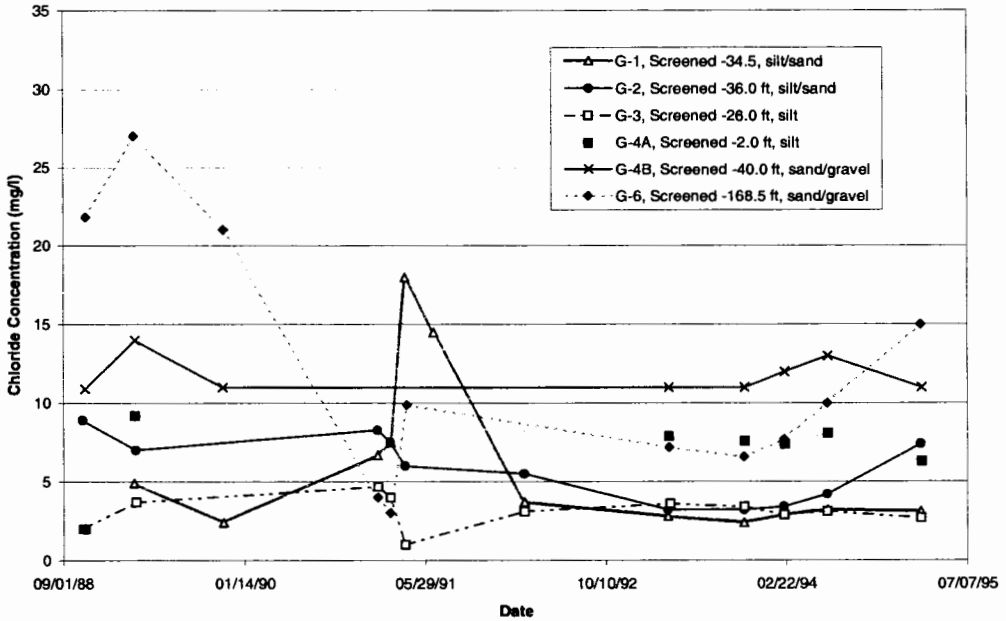
**Chloride Concentration in 'G' Wells**
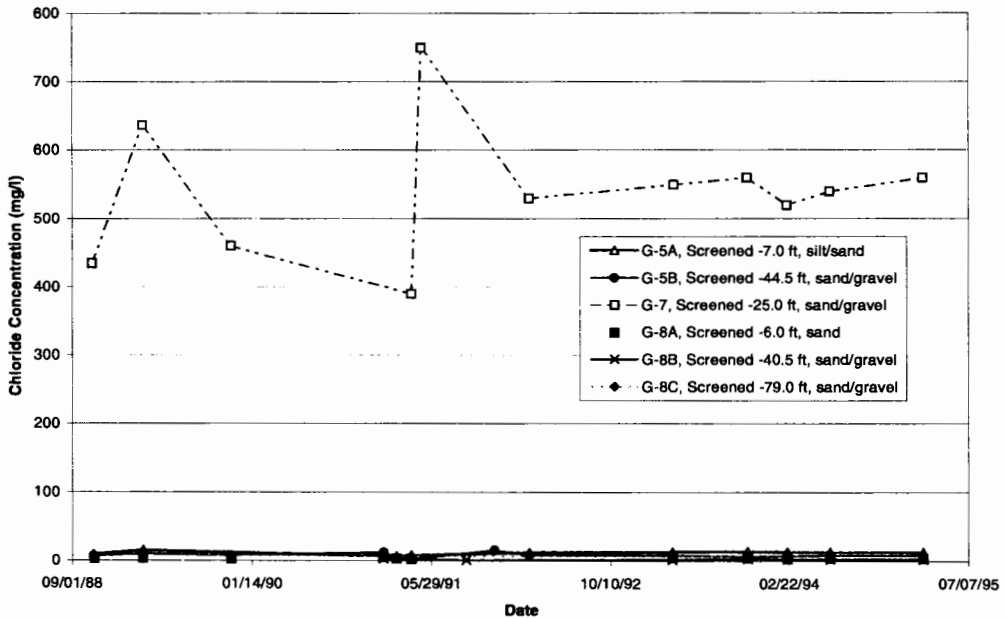
**Figure 21 - Chloride concentrations in the G wells plotted over time. With the exception of G-7, which is located in the hot spot area along the North Slough, the G-Wells do not show much more than background levels.**

### 1.5.5  Schock, 1993:

Kevin Schock presented a masters thesis in 1993 which provided for continuous water level monitoring from 1991-1992, field data analysis, and a 1-D computer model.

The study showed agreement in the flat head gradient in the regional flow below the landfill, with frequent flow reversals.  The water quality monitoring also showed a strong seasonal variation of chloride concentration but with a clear mild positive trend over time.  Concentration data reported by Schock are approximately one order of magnitude higher than those reported earlier and later in separate studies for the same wells, possibly indicating some kind of systematic sampling or recording error.  Thus the absolute value of the data collected may not be reliable.

Concentration data in well G7 were shown to be higher than in most other wells, confirming again that the G7 and D8a well area may indeed be a 'hot spot' for contaminant migration into the gravels.  There were also extremely high chloride concentrations in well D1a which when compared to data taken previously and since (which show concentrations from 150-250 mg/l), seems to be high.

The 1-D modeling provided an estimate of 52 cfd (0.0006 cfs) of contaminant flux into the sloughs, assuming a horizontal conductivity in the dike and refuse of $1.0 \times 10^{-7}$ to  $1.0 \times 10^{-8}$ cm/sec, respectively.  The vertical groundwater flux estimate is 26,000 cfd (0.30 cfs), and is based on the vertical gradient at one point.  It should be noted that the contaminant mound and the silt thickness vary greatly over the area of

the landfill and thus the vertical gradient varies from point to point also. This distribution of the gradient across the landfill has a large effect on the spatial and temporal distribution of the vertical contaminant flux.

## 1.5.6  Emcon, 1989:

In 1989, Emcon estimated the local recharge into the landfill using the HELP (Hydrologic Evaluation Landfill Performance) model based on the actual cover characteristics of the St. Johns Landfill. Recharge was estimated to be 7.29 in/yr (121,000 gallons/day) before landfill closure and near zero after landfill closure.

## 1.5.7  USGS, 1995:

The USGS has recently developed a Deep Percolation Model (DPM) for estimating recharge in the Portland basin at a resolution of 3000 by 3000 square feet. The recharge estimated depends on precipitation and land use characteristics (distribution of impervious areas). Estimated recharge in the St. Johns region ranged from 0-20 in/yr with recharge to the silt in the immediate landfill area estimated to be 15 in/yr. Higher recharge was estimated to the northwest of the landfill due to a low percentage of impermeable area. The model also showed a large seasonal variability with more recharge in the wet season to almost zero in the dry season.

## 2. St. Johns Landfill Modeling System: Telescopic Approach

The groundwater flow at the St. Johns Landfill site and its vicinity exhibits complex temporal and spatial dynamics. The flow pattern in the region is the result of the interaction between hydrologic forcing and geology at different spatial and temporal scales. Including large regional dynamics on the local landfill region is important to accurately address the issues of this project. The telescopic approach nests a series of models which allows for the regional flow characteristics, such as river-aquifer inter-relationships, and surface water features to be included into the localized landfill site flow dynamics. The St. Johns Landfill Modeling System is a series of six models that work together in an interrelated way to accomplish this task. This section will briefly outline each model, the intention of each model, its relationship with the other models, and the people mainly responsible for its implementation. Subsequent chapters will address each model one by one, specifically detailing the conceptual and mathematical representations, the boundary conditions and discretization, the parameter estimation and calibration, results, sensitivity analysis, and limitations. Each model is capable of running independently and providing insight into questions concerning the landfill, however the project issues are best addressed when the models work as a whole.

Water Balance Model:

Key concern in the modeling project is to quantify the landfill's impact on the groundwater and surrounding surface waters. A water balance model is used to

provide a 'lumped' view of the localized contaminant mounding dynamics. Its intent is to quantify the fluxes out of the landfill. The model outputs a yearly averaged and spatially averaged mound height within the landfill as well as total flux leaving the landfill laterally through the surrounding dike into the sloughs, and vertically through the silt into the gravel. In addition it aids in parameter estimation for use in the other models.

The water balance model is the original work of the author of this thesis, Thomas Lowry, consisting of the formulation of the governing equation, numerical integration of the governing equation, FORTRAN coding and debugging, model calibration and verification, sensitivity analysis, and documentation. Chris Berger wrote the preprocessors that convert the original geology data into raster format from which different values could be calculated for use in the water balance model.

Particle Tracking Model:

The particle tracking model uses the mounding results from the water balance model, to track particles across the dike and downward into the gravel. The model output provides distance traveled across the dike and downward through the silt as a function of time. 'Breakthrough' can be constructed from the results that show when the contaminant would completely penetrate the dike or the silt.

Like the water balance model, the particle tracking model is the complete and original work of the author of this thesis.

Regional Flow Model:

The regional scale flow model is designed to capture the large scale flow dynamics in the area between the Columbia and Willamette Rivers. The model produces a yearly averaged and vertically averaged (within each layer), two-dimensional head distribution for the silt layer, sand layer, the gravel layer, and the Troutdale aquifer. The regional flow model addresses the issue as to whether or not the contaminant plume is able to move offsite from the landfill area.

The regional flow model was constructed and run using MODFLOW by Min Chen, Huang Qinsheng, Julie Peyton, and Binhong Wu. Min provided the bulk of the early work on this model including calibration, verification, sensitivity analysis, visualization, and final report documentation. Huang provided the final sensitivity runs and calibration including the results presented here. Julie was instrumental in the initial setup and implementation of the model while Binhong assisted in sensitivity analysis and model simulation. Juan-Juan Wang and Peter Gotseff helped in the some of the model visualization for the final report (Li, et.al., 1994).

Two-Dimensional Mounding Model:

The two-dimensional mounding model can be thought of as a detailed version of the water balance model, capturing not only the temporally varying contaminant mound but also the spatially varying details of the mound. In reality, the contaminant mound is higher in the middle of the landfill and lower on the edges towards the sloughs. The two-dimensional mounding model resolves this spatial variation and outputs a two-dimensional contaminant mound distribution across the landfill on a

yearly averaged basis. The intent of the two-dimensional mounding model is to delineate the contaminant mound in detail for use as the upper boundary condition in the local three-dimensional flow model. The two-dimensional mounding model uses the two-dimensional head data from the regional flow model as part of its boundary conditions as well as the average contaminant mound results from the water balance model in its setup.

Utilizing the MODFLOW code, the two-dimensional site specific mounding model was implemented and run by the author of this thesis, Thomas Lowry. Tom adapted the preprocessor written by Chris Berger in FORTRAN for the three-dimensional local scale flow model for use in this model. In addition, he wrote other preprocessors for use in calculating the conductance for each landfill cell which outputs to the river module as input for MODFLOW. Post processing and visualization of the mounding model results were mainly conducted by Tom with hints and help provided by Peter Gotseff.

Three-Dimensional Flow Model:

The three-dimensional flow model captures the detailed three-dimensional flow field in the localized landfill area, encompassing the landfill, silt layer, sand layer, gravel layer, and Troutdale aquifer. Utilizing results from the regional model for its side and bottom boundary conditions as well as the results from the two-dimensional mounding model for part of its upper boundary conditions, the three-dimensional flow model provides the detailed site specific flow field needed for use in the contaminant transport model.

The model is run using the MODFLOW code and was originally setup and implemented by Chris Berger. Chris wrote the preprocessors for model setup, some initial MODFLOW code changes to account for growth of the landfill over time, and the initial calibration runs. He also provided extensive work for the original data conversion along with Mark Boyko. Approximately half way through the project, Chris moved on to other work and Thomas Lowry completed the model including sensitivity analysis, model documentation, and further code changes. Peter Gotseff and Thomas Lowry completed the post processing.

Contaminant Transport Model:

The contaminant transport model tracks contaminant migration from the landfill through the silts and into the underlying aquifers. The model addresses the issue of how effective the silt layer is at protecting the gravel aquifer. In addition, the contaminant transport model is able to predict the offsite migration of the plume. The models output is a spatially varying, yearly averaged concentration field which describes the plume in three dimensions for each year of the simulation.

The contaminant transport model uses the MT3D code and was setup and run by Thomas Lowry. Tom wrote the original preprocessor to complete this task. Calibrations, verification, sensitivity analysis, post processing, and model documentation are completed by Tom. Again, Peter Gotseff proved invaluable with his help in data visualization.

Dr. Shuguang Li, Ph.D., PE, was the principle investigator.

# 3.    Water Balance Model:

Two important unknowns regarding the St. Johns Landfill are the amount of contaminant flux laterally through the dikes to the North and Columbia Sloughs, and the amount of vertical flux of contaminant downward to the groundwater. The water balance model is a simple, effective tool for quantifying these fluxes and provides a 'big picture' look of the flux through the perimeter dike into the sloughs, and the flux downward through the silt into the gravel.

This section will outline the concepts behind the water balance model, its mathematical structure and setup, key input parameters, calibration, results and sensitivity analysis.

## 3.1    Conceptual Model:

Conceptually the landfill can be visualized as a large pool of water (Figure 22). Water is prevented from draining from the pool by the sides (the dikes) and the bottom (the silt). The main assumption is the conductivity of the landfill is much higher than the surrounding dikes or underlying silt, thus the controlling factor for the rate of drainage from the landfill is the hydraulic conductivity of the dikes and the hydraulic conductivity of the silt. Since the dikes (with the exception of the man-made engineered dike, which is considered nearly impermeable) are made of the same material as the underlying silts, the controlling factor is further reduced to one parameter, the anisotropy of the hydraulic conductivity of the silt. If the silt is highly permeable in the horizontal direction as compared to the vertical direction, the 'pool' (landfill) will drain through the dikes and into the sloughs while very little will travel

vertically through the silt and into the gravels. Conversely, if the silt is highly

permeable in the vertical direction as compared to the horizontal direction, the 'pool'

(landfill) will drain vertically through the silt into the gravels with very little moving

horizontally into the sloughs. The water balance model predicts the contaminant flux

into the sloughs and Smith Lake (along the eastern border) as well as contaminant flux

into the groundwater.
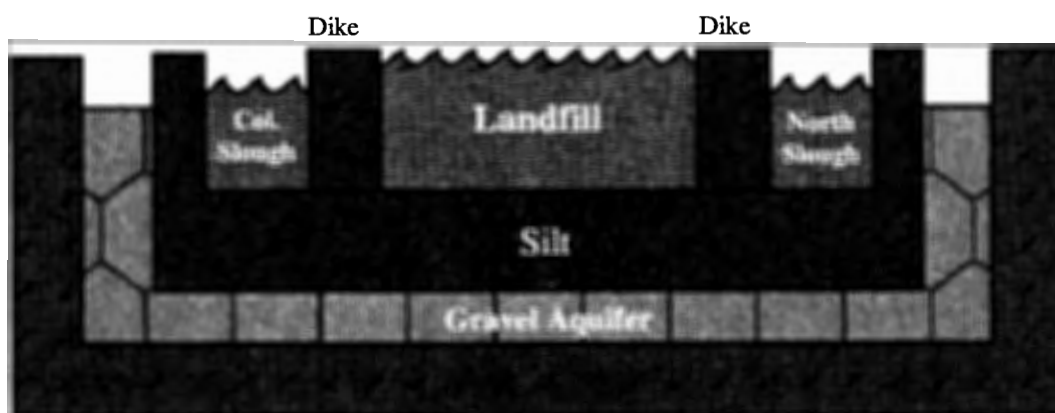


**Figure 22 - Conceptual layout of water balance model. The silt layer and the dikes provide the controlling medium for flow out of the landfill. The difference between the water level in the sloughs and the landfill, and the gravel aquifer and the landfill, as well as the horizontal and vertical conductivity components of the silt, determine the rate at which drainage occurs.**

## *3.2    Mathematical Model:*

A water balance model of this type is called a 'zero-dimensional' model

because it does not take into account x, y, or z directions. The model is a mass

balance statement of the sources and sinks balancing to equal the storage within the

landfill. Mathematically, this expression is given below in equation 1.

$$A(t)S_y \frac{d\phi}{dt} =$$ 
Storage

$$\varepsilon(t)A(t) -$$ 
Recharge

$$\frac{K_{siltH}}{b_{d1}}\left[\phi(t) - \phi_s(t)\right]l_{d1}\left[\frac{\phi(t) + \phi_s(t)}{2}\right] -$$ 
Flux through Natural Dike into Columbia, and North Sloughs

$$\frac{K_{d2}}{b_{d2}}\left[\phi(t) - \phi_s(t)\right]l_{d2}\left[\frac{\phi(t) + \phi_s(t)}{2}\right] -$$ 
Flux through Engineered Dike into North Slough

$$\frac{K_{d2}}{b_{d2}}\left[\phi(t) - \phi_l(t)\right]l_{d3}\left[\frac{\phi(t) + \phi_l(t)}{2}\right] -$$ 
Flux through Engineered Dike into Smith Lake

$$A(t)K_{siltV}\left[\phi(t) - \phi_g(t)\right]\left[\frac{1}{A(t)}\int\frac{dA}{b_s}\right] -$$ 
Flux Downward through Silt

$$\sum\frac{K_{pi}}{b_{pi}}\left[\phi(t) - \phi_{pi}(t)\right]A_{pi}(t) -$$ 
Flux due to Contaminant Collection System

$$\frac{K_c(t)}{b_c(t)}\left[\frac{(\phi(t) - Z_d)}{2}\right]\left[\phi(t) - Z_d\right](l_{d1} + l_{d2} + l_{d3})$$ 
Flux due to Surface Seepage

**(1)**

A(t) = Area of landfill.
$\phi(t)$ = Mean contaminant mound in the landfill.
$S_y$ = Specific yield in landfill.
$\varepsilon(t)$ = Recharge into landfill.
$K_{siltH}$ = Horizontal hydraulic conductivity of natural dike.
$b_{d1}$ = Thickness of natural dike
$\phi_s(t)$ = Water level in sloughs.
$l_{d1}$ = Length of natural dike surrounding landfill.
$K_{d2}$ = Horizontal hydraulic conductivity of engineered dike.
$b_{d2}$ = Thickness of engineered dike
$l_{d2}$ = Length of engineered dike along North Slough.
$\phi_l(t)$ = Water level in Smith Lake.
$l_{d3}$ = Length of engineered dike along Smith Lake.
$K_{siltV}$ = Vertical conductivity of the silt.
$\phi_g(t)$ = Head in gravels.
$b_s$ = Harmonic mean of thickness of silt below landfill.
$K_{pi}$ = Effective conductivity for inlet i to contaminant collection system.
$b_{pi}$ = Effective thickness for inlet i to contaminant collection system.
$\phi_{pi}(t)$ = Head in contaminant collection system.
$K_c$ = Horizontal hydraulic conductivity of daily cover on landfill.
$b_c$ = Thickness of daily cover on landfill.
$Z_d$ = Top elevation of dikes.

Equation (1) is a statement which balances the recharge into the landfill with the contaminant flux downward through the silt plus the contaminant flux laterally through the dikes. The landfill area, recharge, and water level parameters are all time dependent to allow for changes reflecting the growth of the landfill and yearly variability in water levels and recharge. The storage term is the time derivative with respect to the head (mean contaminant mound) times the specific yield. For an unconfined aquifer, which the landfill is, the specific yield represents the volume of water that an aquifer releases from storage per unit surface area of aquifer per unit decline in the water table (Freeze and Cherry, 1979).

Each of the fluxes are defined as the Darcy flux times the area along the appropriate boundary. The flux across the dike is non-linear since the head level in the landfill determines the flux area. As the head level increases in the landfill, the available area for contaminant to flow to the sloughs increases. This non-linearity prevents the equation from being solved analytically and thus it is solved numerically (source code and documentation are shown in Appendix A). Once the mounding is equal to or greater than the height of the dike, the dike area is constant, represented as the dike height times the length of the dike.

### 3.3    *Initial Conditions and Discretization:*

Since the water balance model is a zero-dimensional model, boundary conditions are not needed however initial conditions must be specified. The only parameters requiring initial conditions for the model to run are the mean head in the landfill, the water levels in the sloughs, and the head in the gravels. The assumption is

made that the contaminant mound in the landfill is the same at the start of the simulation as the surrounding surface waters. Dumping did occur before 1950 when the model simulation begins, but it is assumed the topography does not change enough prior to 1950 to produce any mounding above the surface water elevation. Head in the gravels is taken to be the average of all the cells underlying the landfill from the regional flow model results run with steady state conditions. Prior to calibration, all other parameters are initially set with text book or literature values.

Assuming a landfill mound equal to the surrounding surface water starting in 1950, the landfill mound gradually increases as recharge is added. Once the mound is high enough to produce offsetting fluxes equal to the recharge, the mound stabilizes. Only when the landfill is capped, and recharge is reduced, do we see the mound dissipate. The simulation runs for 70 years from 1950 to 2020.

### 3.4 *Parameter Estimation and Calibration:*

The parameters requiring calibration for the water balance model are recharge, $K_{siltH}$, $K_{siltV}$, and specific yield in the landfill. Head in the gravel layer is obtained from results of the regional flow model (Chapter 5). Water level in the sloughs is obtained from data supplied by the Army Corps of Engineers via Parametrix, Inc. Other parameters such as dike length, dike thickness, and silt thickness are derived from stratigraphic maps of the landfill region (METRO, 1994), and for this modeling study, are assumed accurate.

Hydraulic Conductivity:

Concern for determining the relative and absolute fluxes vertically into the deep aquifer as well as horizontally into the sloughs makes the calibration of $K_{siltH}$ and $K_{siltV}$ extremely important. Slug tests (Sweet-Edwards/Emcon, 1989) in the natural dike reported values for $K_{siltH}$ of $1.3 \times 10^{-6}$ cm/sec (0.0037 ft/day) to $3.1 \times 10^{-5}$ cm/sec (0.0878 ft/day) and $K_{siltV}$ has been reported as $2.0 \times 10^{-7}$ cm/sec (0.0006 ft/day) to $5.0 \times 10^{-7}$ cm/sec (0.0014 ft/day) in wells G1 and G2 respectively (METRO, 1994). The water balance model simulates the entire landfill as one unit thus the values for $K_{siltH}$ and $K_{siltV}$ are effective field scale values for the site. Slug tests, like those that have been performed in the past at the landfill site, allow for prediction of conductivity on a local scale. Care should be taken in applying local values to field scale problems. Overbank deposits, like the silts underlying the landfill, are by nature heterogeneous. The dike surrounding the landfill, being made up of the same overbank deposits as the silts, is also heterogeneous and contains discontinuities comprised of refuse (METRO, 1995). For this reason, the local scale values should not be blindly applied to the field scale model and do require calibration.

The quasi steady-state contaminant mound as determined from well data within the landfill allows us to back calculate $K_{siltH}$ and $K_{siltV}$. The dike mainly controls the lateral seepage as dictated by $K_{siltH}$ while the underlying silts mainly control the vertical seepage dictated by $K_{siltV}$. Since the natural dike and underlying silts are made of the same overbank deposits, we can reduce the task to finding just one parameter; anisotropy. Anisotropy is the ratio between the vertical hydraulic

conductivity and the horizontal conductivity and is mainly caused by heterogeneities in a geologic medium. By systematically adjusting $K_{siltH}$ and $K_{siltV}$ over likely ranges, a series of anisotropy's that reproduce the observed average landfill mound can be found. With a constant recharge, the water balance model establishes a relationship between the mean contaminant mound in the landfill and anisotropy.

Figure 23 shows the relationship between $K_{siltH}$, $K_{siltV}$, and the mean contaminant mound in the landfill. The figure distinguishes a range of $K_{siltH}$ and $K_{siltV}$ which allows the model to reproduce the mounding. The observed mean head of 25 feet along with the likely upper and lower limits of 30 and 20 feet (Sweet-Edwards/Emcon, 1989), are shown. Any values within the range could be thought to satisfy the mounding criteria, but clearly there are values which are more realistic than others.

Freeze and Cherry, (1979), suggest that in highly heterogeneous waste sites the horizontal conductivity can be much higher than the vertical conductivity. This ability for water to flow easier in the horizontal direction than in the vertical can occur for two reasons at sites like the St. Johns Landfill. First is that during deposition, particles tend to settle on their 'side' making for a matrix that is aligned with its longer dimensions in the horizontal direction. Second, and most important, is that floodplain deposits are seldom a homogeneous unit. Different flood events with different energies will deposit different sized particles. This creates a geologic unit that will be interspersed with horizontal 'lenses' of higher and lower conductivity material. Water tends to move along the path of least resistance and thus will move easier along the

48

horizontally oriented higher permeability zones. For perfectly stratified systems, the effective conductivity in the horizontal direction is the arithmetic mean of the horizontal conductivity of each layer while in the vertical direction, the effective conductivity is the harmonic mean of the vertical conductivities of each layer (Freeze and Cherry, 1979). For a system which is less the perfectly stratified, the effective conductivity will lie somewhere inbetween the arithmetic and the harmonic mean. The harmonic mean will tend to be much closer to the lowest value than the arithmetic mean (for example, the arithmetic mean of 1 and 99 is 50, while the harmonic mean of 1 and 99 is 1.98), meaning that groundwater movement in the vertical direction is controlled mainly by the lowest permeability layer, while movement in the horizontal direction is controlled mainly by the high permeability layers. It can be proven mathematically that the harmonic mean will always be less than the arithmetic mean indicating that flow across bedding layers is always more difficult than flow along bedding layers.

During calibration, anisotropy values of 10 to 20 are used. These values produce predicted contaminant plumes in the transport model (Chapter 8) which are much more extensive than well data show. Slug tests for wells G1 and G2 (Sweet-Edwards/Emcon, 1989) show anisotropy values in the natural dike to be 42 and 75 respectively. Due to settling, landfill weight, and preloads prior to closing, the vertical conductivity of the silt is reasoned to be greatly reduced and thus the values of anisotropy are probably increased beyond typical values given for silt overbank deposits. Since no direct data for anisotropy exist for the silt directly below the

landfill, and to be conservative in our estimates, calibration is run on anisotropy values from 10 to 200.
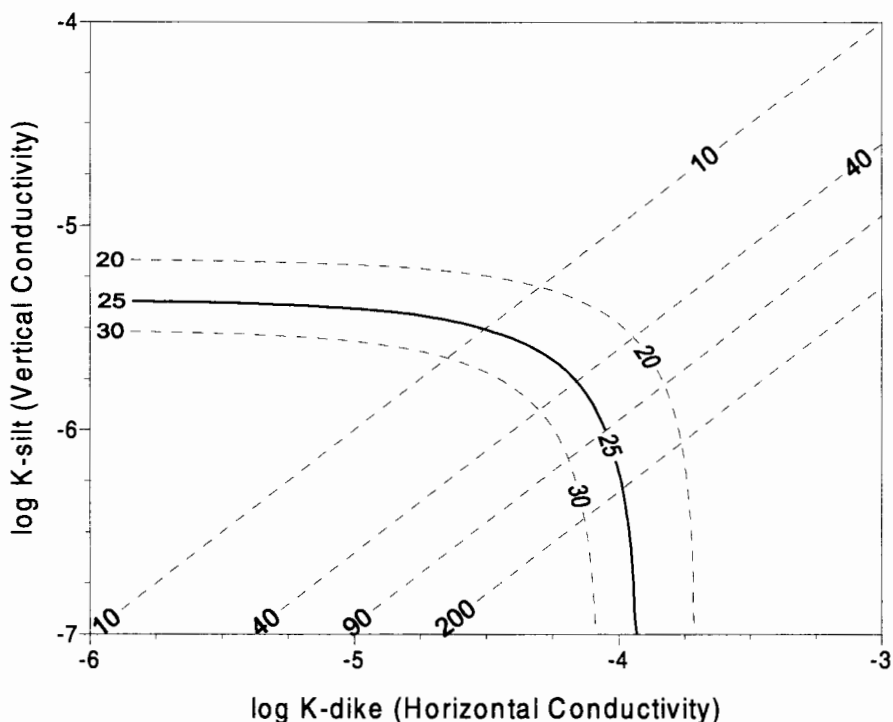


**Figure 23 - The 25 ft contour corresponds to the space-time average of the observed head in the landfill. The dashed 20 and 30 ft head contours correspond to the likely range of observed mean heads in the landfill. Anisotropy of 10, 40, 90, and 200 are shown by the solid, slanted lines. The band between the two limiting head contours represents the region where the horizontal dike and the vertical silt layer conductivity's are compatible with head observations. The region boxed by the limiting heads and the limiting anisotropy's represents the feasible region for the dike and silt conductivity.**

Calibration is performed on contaminant data (Sweet-Edwards/Emcon, 1989) which show an approximate vertical travel distance of contaminant through the silt to be 30 to 40 feet in 30 to 50 years in the area of well D8a (west end of the north slough). A value of 90 was adopted as base case to best reproduce these results. From

Figure 23, this corresponds to $K_{siltH}$ equal to $9.0 \times 10^{-5}$ cm/sec (0.2551 ft/day) and $K_{siltV}$ equal to $1.0 \times 10^{-6}$ cm/sec (0.0028 ft/day).

Recharge:

Recharge is the amount of water that seeps into the landfill from precipitation and controls the magnitude of total seepage into the sloughs and the deep aquifer. If recharge is high, then seepage into the sloughs and deep aquifer must be high to offset the inflow and still maintain the observed average contaminant mound.

The starting point for the recharge value is established from a study using the HELP model (Hydrological Evaluation Landfill Performance model, Sweet-Edwards/Emcon, 1989) which estimates the steady state landfill infiltration from the landfill cover characteristics and precipitation data. The cover characteristics have changed dramatically over the lifetime of the landfill as different management practices have changed thus recharge was higher when the landfill was not operated as a sanitary landfill (pre-1969). The EMCON HELP model predicts a pre-closure infiltration rate of 7.29 in/yr. The HELP model estimates infiltration rates much higher during the pre-1969 period than the post-1969 period, when it was managed as a sanitary landfill. Based on average rainfall of 43.1 in/yr (annual average from 1950 through 1994 for downtown Portland as supplied by George Taylor, Oregon State Climatologist) and to be conservative, a base case value of 10 inches per year (23,470 ft$^3$/day) was adopted. Once the landfill is capped, recharge is assumed to reduce to10% of the full non-closure rate (1 in/yr for the base case). Since recharge values are adopted results of the HELP model, recharge is not altered during calibration.

Specific Yield:

Specific yield effects the rate of contaminant mound growth or dissipation. Several studies address the issue of specific yield in landfills. Oweis et al. (1990) present a review of hydraulic properties of municipal landfills and recommend using a mean value of 0.1 for a typical compacted municipal landfill. The US EPA, from a study in 1971 on four different landfills, suggest that specific yield may range from 0.1 to 0.5 depending on the landfill's age, and the degree of compaction. Stevens et al. in 1972 formulated an empirical graphical relationship between the field capacity and the degree of compaction. Using an assumption of 500 lb. per cubic yard, they obtain an estimate of specific yield of 0.4 to 0.5 for the St. Johns Landfill. The earliest mounding data available are from the Stevens, Thompson, and Runyan study in 1974, well after the contaminant mound had stabilized, leaving no data showing the rate of growth of the contaminant mound. The latest data available at the time of this study show no significant decline in the mound, even though capping has begun. Since no data describing either the rate of increase or the rate of decrease of the contaminant mound are available, calibration for specific yield is impossible. Considering the present age of the St. Johns Landfill, the compaction, settlement, and preloading since 1970, and considering the typical values discussed in the literature, final model simulations were carried out for a range of specific yields from 0.1 to 0.5. The base case adopts a value of 0.2.

Head Values in Gravel Aquifer:

The head in the gravel is adopted from the results of the regional flow model. Head values for each cell from the regional model that corresponds to the area directly below the landfill are arithmetically averaged and used as the head in the gravel. Three different simulations of the regional model are performed, representing periods of different hydraulic stresses in the model to establish the head in the gravels. One simulation from 1950 to 1982 represents the time prior to Bybee Lake being closed off to the North Slough and returns a value of 11.8 feet (City of Portland datum - COP). The next simulation is from 1983 to 1997 during the time the North Sough is hydraulically disconnected from Bybee Lake, and returns a value of 11.44 feet (COP). The last simulation is for the future years of 1998 to 2020, and is based upon the calibration year for the regional model of 1989 and returns a value of 11.34 feet (COP). The lower value for 1989 is due to the widespread drought in the area during the late eighties and early nineties. It was originally thought that the full average value from 1950 to 1995 should be used for the 1998 to 2020 period but because the lower value (11.34 feet) is more conservative when it came to the question of contaminant migration into the gravel aquifer, 11.34 feet was chosen. As will be shown later, the model is fairly insensitive to the head in the gravel and either choice produces almost identical results.

Water Levels in Sloughs:

Water level values in the sloughs are the same as that used in the regional flow model (Chapter 5) as obtained from the US Army Corps of Engineers via Parametrix,

Inc.  Since no continuous slough level data exist for the historical portion of the modeling period (1950-1994), water levels for the Willamette River taken at the Hawthorne Bridge  are used.  There are slight differences between levels in the Willamette River and the Columbia and North Slough water levels, however, since the data are yearly averaged, and the fact that the model is very insensitive to water levels in the sloughs, no adjustments are made to accommodate the discrepancy.
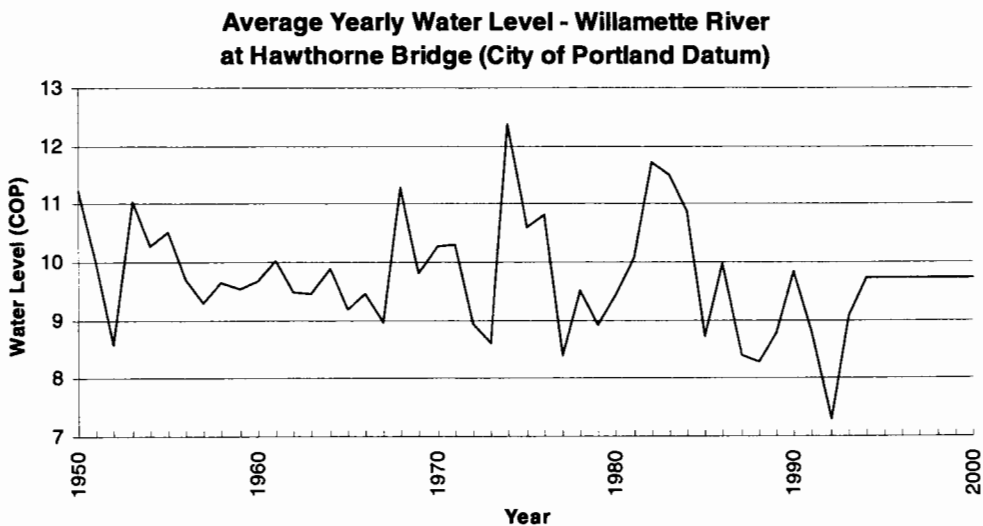


**Figure 24 - Yearly data from the Army Corps of Engineers for the Willamette River at the Hawthorne Bridge.  These data are used as the yearly mean slough level in the water balance model.  Continuous data for the Columbia Slough from 1950 to 1994 do not exist.  Because of this, and the fact that the model is insensitive to slough levels no adjustment is made in the data.**

## 3.5   Results:

To start the simulation, the contaminant mound in the landfill is equal to the water levels in the sloughs.  As recharge is added to the landfill, mounding increases (Figure 25).  As the mound increases, a driving force results which provides the mechanism for contaminant flux laterally into the sloughs and vertically into the

gravel. The mounding and the corresponding fluxes continue to grow until the

outflows from the various fluxes equals the inflow due to recharge. This growth

period lasts from 1950 to about 1960. At this point, active mounding stops and the

landfill is in a 'quasi' steady state, balanced between the inflows and outflows. The

quasi steady state mode remains until landfill closure begins around 1991. The

landfill closure reduces influx due to recharge and the mound and fluxes start to

dissipate. This phase continues for about 20 years until the mound decreases to the

point when it is equilibrated with the post-closure recharge rate and the outward

fluxes. The horizontal seepage flux into the sloughs and the vertical seepage flux into

the gravels are discussed below in detail.
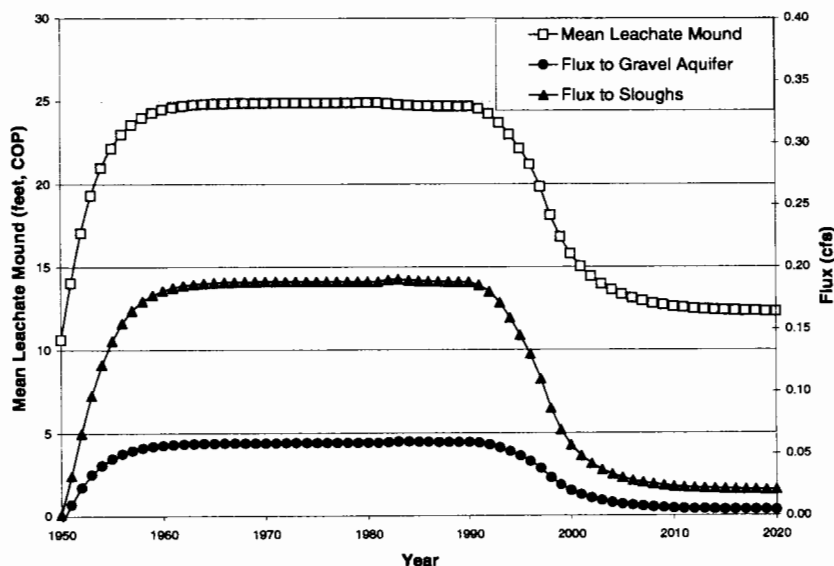


**Figure 25 - Time variation of predicted contaminant mound in the landfill, total seepage flux to the sloughs, and seepage flux through the silt to the gravel as predicted by the water balance model. The general shape of the curves reflects the process of contaminant mounding due to landfill infiltration and contaminant dissipation in response to landfill capping. When the mounding is neither increasing nor decreasing, the total flux rate out of the landfill is equivalent to the recharge rate.**

55

Lateral Flux through the Dike:

Anisotropy plays a large role in the distribution of the fluxes out of the landfill. For the base case of $K_{siltH}=9.0\times10^{-5}$cm/sec (0.2551 ft/day) and $K_{siltV}=1.0\times10^{-6}$cm/sec (0.0028 ft/day), and recharge of 10 inches per year, the model predicts the maximum lateral seepage flux into the sloughs for any year is 0.192 cfs (124,100 gallons per day). For anisotropy's of 10, 40, and 200, the predicted maximum lateral slough flux ranges from 0.069 to 0.150 to 0.219 cfs (44,600, 97,000, and 141,500 gallons per day, Table 1).

After landfill closure, the mounding starts to dissipate due to the reduced recharge. Assuming specific yields in the range of 0.2 to 0.4, and the base case anisotropy of 90, lateral predicted seepage flux to the sloughs is between 0.057 cfs (36,800 gallons per day, specific yield = 0.2) and 0.090 cfs (58,200 gallons per day, specific yield = 0.4) by the year 2000. These drop to 0.021 cfs (13,580 gallons per day, specific yield = 0.2) and 0.026 cfs (16,810 gallons per day, specific yield = 0.4) respectively by 2020.

Vertical Flux through Silt:

Although the effective area for seepage flux is much greater in the vertical direction than horizontally through the dikes, the flux to the gravel aquifer is much less due to the thickness of the silt as well as the strong horizontal to vertical silt conductivity. For anisotropy's of 10, 40, 90, and 200 the corresponding maximum seepage flux for any year into the gravel aquifer is predicted as 0.190 cfs (123,000 gallons per day), 0.104 cfs (67,200 gallons per day), 0.059 cfs (38,100 gallons per

56

day), and 0.031 cfs (20,000 gallons per day).  Note that the sum of the vertical flux

and the horizontal flux is approximately equal to the total recharge in all cases (about

.25 cfs - 161,600 gallons per day).  This relationship doesn't hold exactly since the

flux through the engineered dike to Smith Lake is neglected ($\sim 1 \times 10^{-5}$ cfs) and there is

some slight loss to surface seepage.  Assuming specific yields of 0.2 to 0.4, for the

base case of an anisotropy of 90, by the year 2000 the seepage flux into the gravel

aquifer reduces to 0.019 cfs (12,300 gallons per day) to 0.031 cfs (20,000 gallons per

day).

| Year | 1985 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Specific Yield** | **0.2** | | | | **0.4** | | | |
| **Anisotropy** | **10** | **40** | **90** | **200** | **10** | **40** | **90** | **200** |
| **Flux to Sloughs** | 0.067 | 0.147 | 0.189 | 0.216 | 0.067 | 0.147 | 0.190 | 0.217 |
| **Flux to Gravels** | 0.189 | 0.104 | 0.059 | 0.031 | 0.189 | 0.104 | 0.060 | 0.031 |
| **Mound Height** | 24.82 | 24.76 | 24.77 | 24.37 | 24.85 | 24.78 | 24.80 | 24.40 |
| **Year** | **1995** | | | | | | | |
| **Specific Yield** | **0.2** | | | | **0.4** | | | |
| **Anisotropy** | **10** | **40** | **90** | **200** | **10** | **40** | **90** | **200** |
| **Flux to Sloughs** | 0.050 | 0.112 | 0.145 | 0.166 | 0.056 | 0.125 | 0.161 | 0.184 |
| **Flux to Gravels** | 0.151 | 0.084 | 0.048 | 0.025 | 0.165 | 0.091 | 0.052 | 0.027 |
| **Mound Height** | 22.08 | 22.12 | 22.18 | 21.83 | 23.11 | 23.09 | 23.13 | 22.75 |
| **Year** | **2000** | | | | | | | |
| **Specific Yield** | **0.2** | | | | **0.4** | | | |
| **Anisotropy** | **10** | **40** | **90** | **200** | **10** | **40** | **90** | **200** |
| **Flux to Sloughs** | 0.019 | 0.043 | 0.057 | 0.065 | 0.031 | 0.069 | 0.090 | 0.102 |
| **Flux to Gravels** | 0.060 | 0.034 | 0.020 | 0.010 | 0.098 | 0.055 | 0.032 | 0.016 |
| **Mound Height** | 15.53 | 15.70 | 15.82 | 15.62 | 18.27 | 18.36 | 18.46 | 18.17 |
| **Year** | **2020** | | | | | | | |
| **Specific Yield** | **0.2** | | | | **0.4** | | | |
| **Anisotropy** | **10** | **40** | **90** | **200** | **10** | **40** | **90** | **200** |
| **Flux to Sloughs** | 0.008 | 0.017 | 0.021 | 0.023 | 0.009 | 0.020 | 0.026 | 0.029 |
| **Flux to Gravels** | 0.017 | 0.008 | 0.004 | 0.002 | 0.022 | 0.012 | 0.007 | 0.003 |
| **Mound Height** | 12.57 | 12.41 | 12.32 | 12.17 | 12.89 | 12.85 | 12.85 | 12.71 |

**Table 1 - Table showing results for varying specific yields and anisotropy. Recharge is 10 inches per year. Note that prior to closure (about 1990), the sum of the vertical seepage flux to the gravels and lateral seepage flux to the sloughs is equal to the recharge. An anisotropy of 10 corresponds to $K_{siltH}$=3.16×10$^{-5}$ cm/sec (0.090 ft/day) and $K_{siltv}$=3.16×10$^{-6}$ cm/sec (0.009 ft/day). An anisotropy of 40 corresponds to $K_{siltH}$=7.00×10$^{-5}$ cm/sec (0.198 ft/day) and $K_{siltv}$=1.75×10$^{-6}$ cm/sec (0.005 ft/day). An anisotropy of 90 corresponds to $K_{siltH}$=9.00×10$^{-5}$ cm/sec (0.255 ft/day) and $K_{siltv}$=1.00×10$^{-6}$ cm/sec (0.003 ft/day). An anisotropy of 200 corresponds to $K_{siltH}$=1.07×10$^{-4}$ cm/sec (0.303 ft/day) and $K_{siltv}$=5.34×10$^{-7}$ cm/sec (0.001 ft/day).**

### 3.6   Sensitivity Analysis:

Sensitivity analysis is performed on recharge, $K_{siltH}$, , $K_{siltv}$, specific yield within the landfill, head in the gravel layer, and slough level. This section reviews the extent that changes in these parameters effect results.

Sensitivity to Recharge:

With all other values held at base case, recharge is allowed to range from 5 to 15 in/yr in 2.5 in/yr increments (Figure 26). As recharge is increased over the base case of 10 in/yr, the mounding, flux to the sloughs, and flux to the gravel aquifer increase. The model accounts for surface seepage when the contaminant mound exceeds the dike height of 25 feet and since the base case assumes a mean contaminant mound of 25 feet, we see no surface seepage in the base case simulation. The change in recharge has a greater effect in terms of percent difference from base case, on the flux to the gravel aquifer because of the higher vertical conductance relative to the horizontal conductance. For the quasi steady state period from 1960-1990, flux to the gravel aquifer is predicted to increase from about 0.06 cfs (38,800 gallons per day) when recharge is at 10 in/yr to about 0.08 cfs (51,700 gallons per day) when recharge is at 15 in/yr. This represents an increase of 60%. For the flux to the slough, the increase is predicted to be 0.19 cfs (122,800 gallons per day) to 0.28 cfs (181,000 gallons per day). This represents an increase of only 48%. Predicted mounding increases are from 25 feet to about 31 feet or an increase of 24%.

Each 2.5 in/yr increase or decrease of recharge into the landfill within the likely range of 5 to 15 in/yr produces about the same magnitude increase or decrease in the mounding, flux to the sloughs, and flux to the gravels. It should be stressed that recharge is highly variable across the landfill due to variable composition, compaction, and management techniques while the landfill was being formed. The water balance model, being a lumped parameter zero dimensional model, does not account for this variability.
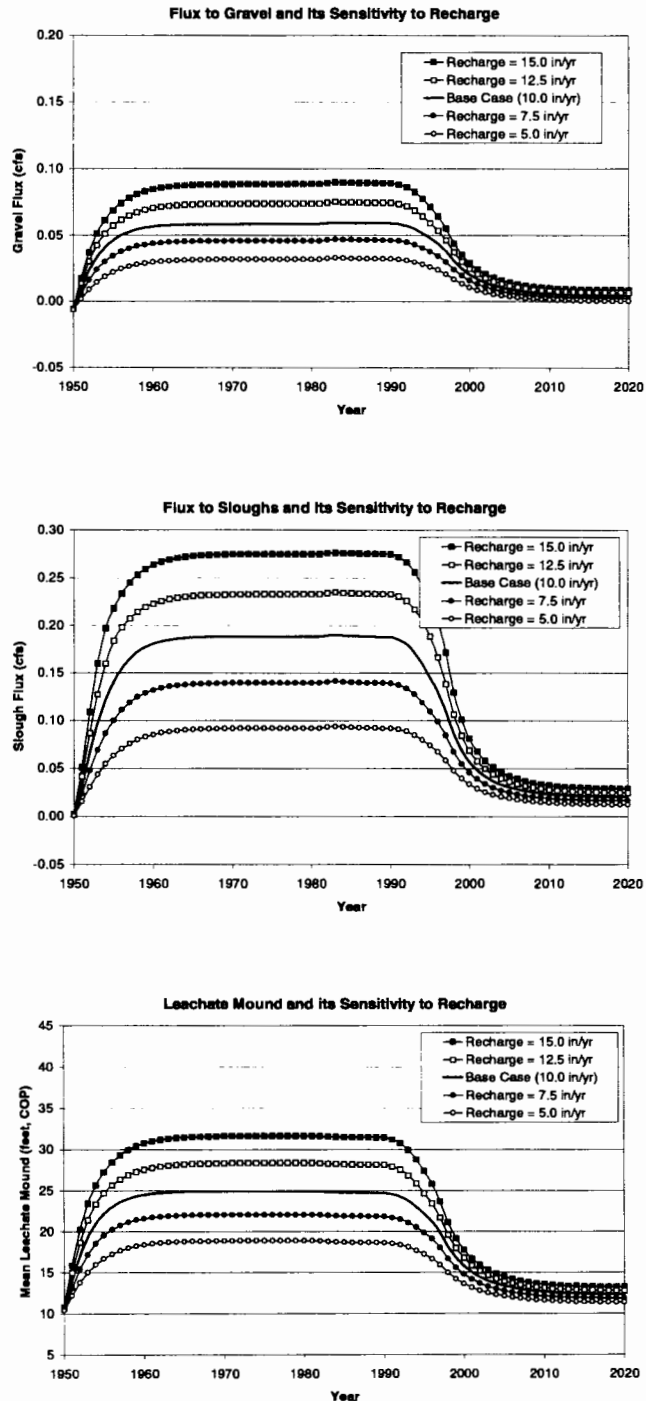
**Figure 26 - Sensitivity of fluxes and mounding to recharge. As recharge is increased, mounding as well as all flux components are predicted to increase. In the quasi steady-state period from 1960 through 1990, total flux out of the landfill is equivalent to the amount of recharge into the landfill.**

61

Sensitivity to Specific Yield:

Specific yield effects the rate of change of the contaminant mound in the landfill and has no effect on the steady state magnitude of the mound (Figure 27). Assuming all parameters on the right hand side of equation (1) remain constant, if specific yield is low, then the time derivative, or the rate of change of head in the landfill must be higher. The opposite holds true if specific yield is raised; the time derivative must be lower to compensate.

Sensitivity analysis is performed on specific yields of 0.1 to 0.5. As mentioned in the section on calibration, no data exist to calibrate specific yield. For proper calibration, data during the mounding or dissipation phase are needed. Monitoring of the contaminant mound does not start until after the mound stabilizes and currently, there is not enough of a drop from the capping procedure to produce significant dissipation of the mound. The sensitivity analysis shows that at a specific yield of 0.10, the contaminant mound stabilizes in about 4-5 years and dissipates in about the same time period. At specific yields of 0.50, the mound takes about 25 years to stabilize and about the same to dissipate. In all cases, the mounding, slough flux, and gravel flux are all within 20% of the 1950, pre-mound conditions by the year 2020.
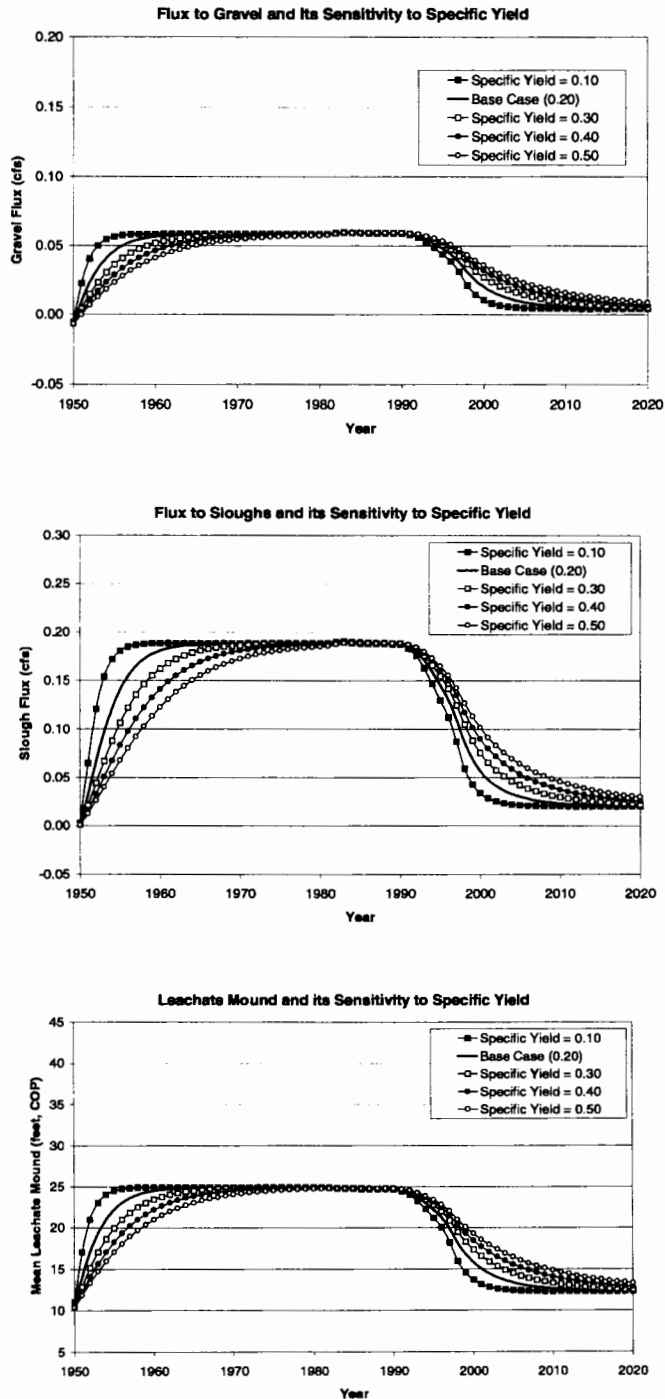
**Figure 27 - Sensitivity of fluxes and mounding to Specific Yield. The rate of mounding and dissipation is controlled by the specific yield. This can also be seen in the changes of the flux components. A low specific yield causes fast mounding and fast dissipation while a high value for specific yield causes mounding to increase slowly and dissipate slowly.**

Sensitivity to Slough Levels:

Water levels in the sloughs effect the head gradient across the dike and thus the flux into the slough. Sensitivity was performed on slough levels of 5 to 15 feet (base case is 9.74 feet, COP). Not surprisingly, when the slough level increases, the flux into the slough decreases (Figure 28). With the slough levels high, we see an increase in the mounding and a corresponding increase in the flux into the gravel aquifer.

While these results help validate the models ability to capture the physical processes at the site, the model does not exhibit high degrees of sensitivity to the water level in the sloughs. Over the 5 to 15 foot range in the slough, we see the predicted landfill mound range from 23 to 27 feet, the predicted seepage flux into the sloughs range from 0.195 cfs (126,000 gallons per day) to 0.18 cfs (116,300 gallons per day), and the predicted seepage flux into the gravel aquifer range from 0.045 cfs (29,100 gallons per day) to 0.065 cfs (42,000 gallons per day).
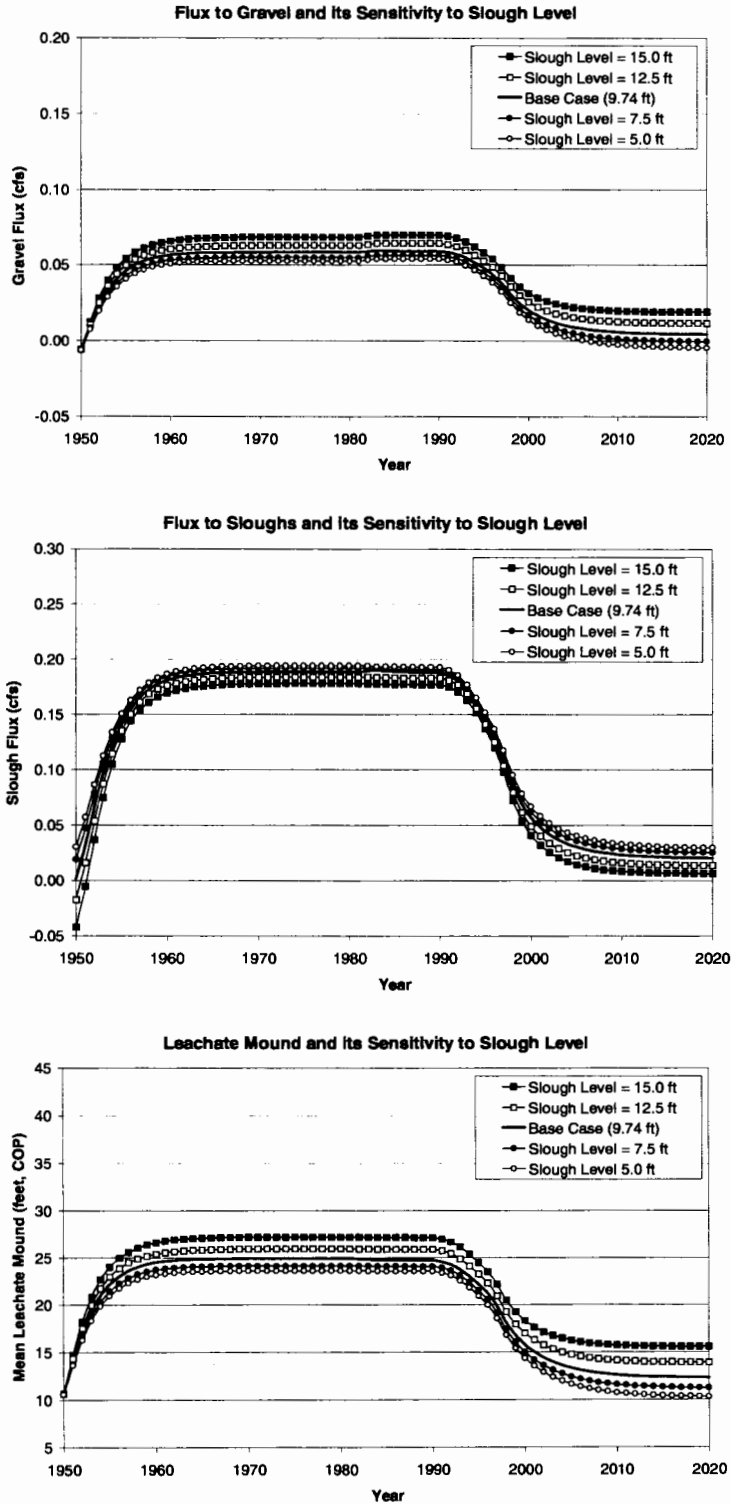
**Figure 28 - Sensitivity to water level in the sloughs. In all cases, model predictions are fairly insensitive to slough levels.**

Sensitivity to Head in Gravel Aquifer:

Gravel head is tested over a range of 8 to 16 feet (base case is 11.69 feet,

COP). The mechanics are very similar to the water level in the slough except that

there is no non-linearity due to changing area of flux with changing head in the landfill

(Figure 29). As the gravel head increases, flux to the gravel decreases, mounding in

the landfill increases, and flux to the sloughs increases. The opposite holds true if the

gravel head is decreased. Over the full range of 8 to 16 feet, mounding ranges from 24

to 26 feet, flux to the sloughs ranges from 0.175 cfs (113,100 gallons per day) to 0.205

cfs (132,500 gallons per day), and flux to the gravel aquifer ranges from 0.07 cfs

(45,239 gallons per day) to 0.04 cfs (25,900 gallons per day). The model is not

sensitive to changes in the head in the gravels.
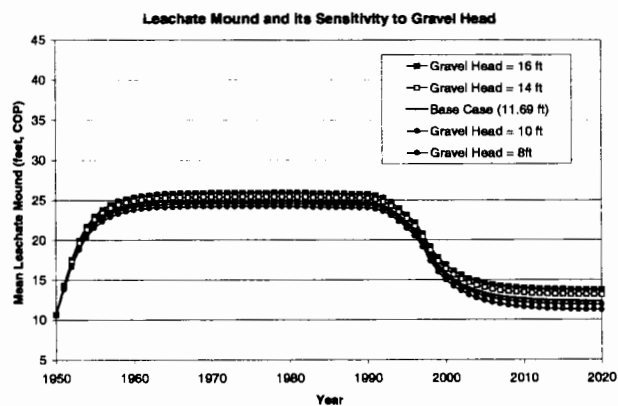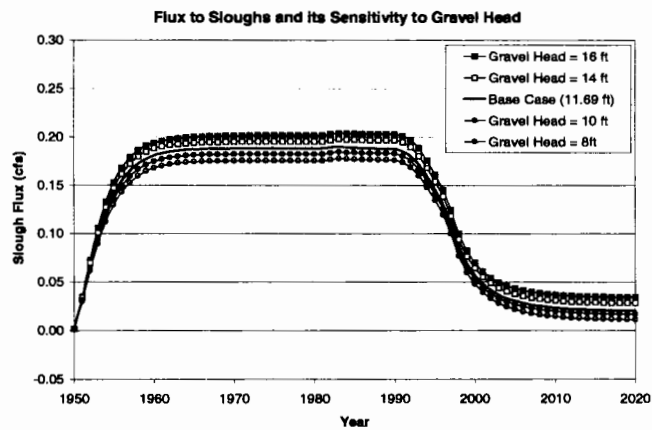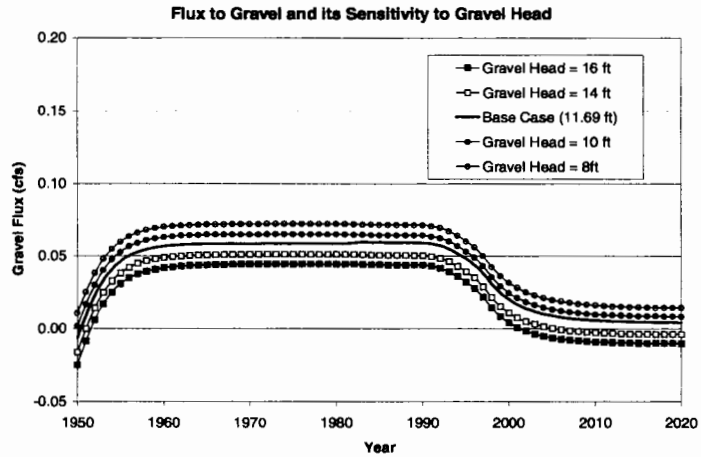
Figure 29 - Sensitivity to head level in the gravels. In all cases, model predictions are insensitive to head levels in the gravels.

Sensitivity to Horizontal Conductivity in the Silt ($K_{siltH}$):

In terms of contaminant mounding, the water balance model is most sensitive to the horizontal conductivity in the silt (which is equivalent to the horizontal conductivity in the dike). With all other parameters held constant at their base case values, $K_{siltH}$ is systematically changed from $1\times10^{-3}$ cm/sec to $1\times10^{-5}$ cm/sec (base case is $9\times10^{-5}$ cm/sec). As $K_{siltH}$ increases (the silt/dike becomes more conductive in the horizontal direction) the mounding decreases, the flux to the sloughs increase, and the flux to the gravels decrease (Figure 30). The opposite holds true when $K_{siltH}$ decreases from the base case.

Through the range of $1\times10^{-3}$ cm/sec to $1\times10^{-5}$ cm/sec, mounding is predicted to range from 12 to 45 feet, flux to the gravels ranges from 0.005 cfs to 0.15 cfs, and flux to the sloughs ranges from 0.25 cfs (161,500 gallons per day) to 0.05 cfs (32,300 gallons per day). When mounding is high, contaminant is also lost to surface seepage. For the mounding, the flux to the sloughs, and the flux to the gravels, the model becomes fairly insensitive to $K_{siltH}$ values higher than $5\times10^{-4}$ cm/sec. This is because the conductivity of the dike reaches a point where the limiting factor becomes the mounding, not $K_{siltH}$. If the conductivity of the dike is not low enough to produce sufficient mounding, no driving force is present for contaminant migration. As $K_{siltH}$ is increased beyond $5\times10^{-4}$ cm/sec, flux to the sloughs approaches recharge.
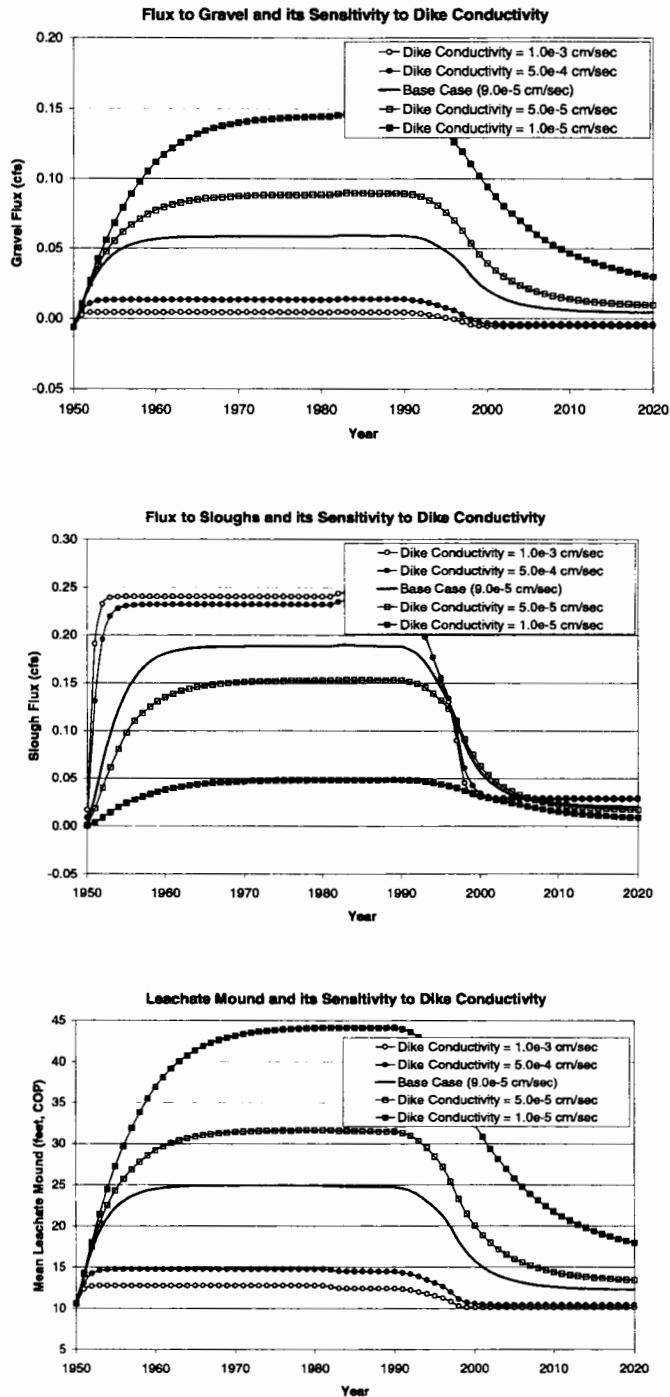
**Figure 30 - Sensitivity to conductivity in the dike. As the dike conductivity decreases, mounding increases, flux to the gravel increases, and flux to the slough decreases. In extreme cases with the conductivity very high, mounding is minimal and flux to the slough is very close to the amount of recharge into the landfill.**

Sensitivity to Vertical Conductivity in the Silt ($K_{siltV}$):

Second only to $K_{siltH}$, the water balance model is sesitive to changing values of $K_{siltV}$. Sensitivity analysis on $K_{siltV}$ ranged from $1 \times 10^{-5}$ cm/sec to $1 \times 10^{-7}$ cm/sec (base case is $1 \times 10^{-6}$ cm/sec). As $K_{siltV}$ increases from the base case value (the silt becomes more conductive in the vertical direction), the landfill mound decreases, the flux to the sloughs decrease, and the flux to the gravels increase (Figure 31). If the value of $K_{siltV}$ decreases from the base case, the opposite occurs.

As the values for $K_{siltV}$ are systematically changed from $1 \times 10^{-5}$ cm/sec $1 \times 10^{-7}$ cm/sec, the mounding goes from 12 to 25 feet, the flux to the sloughs go from 0.05 cfs (32,300 gallons per day) to 0.25 cfs (161,500 gallons per day), and the flux to the gravels go from 0.15 cfs (97,000 gallons per day) to 0.005 cfs (3,200 gallons per day).
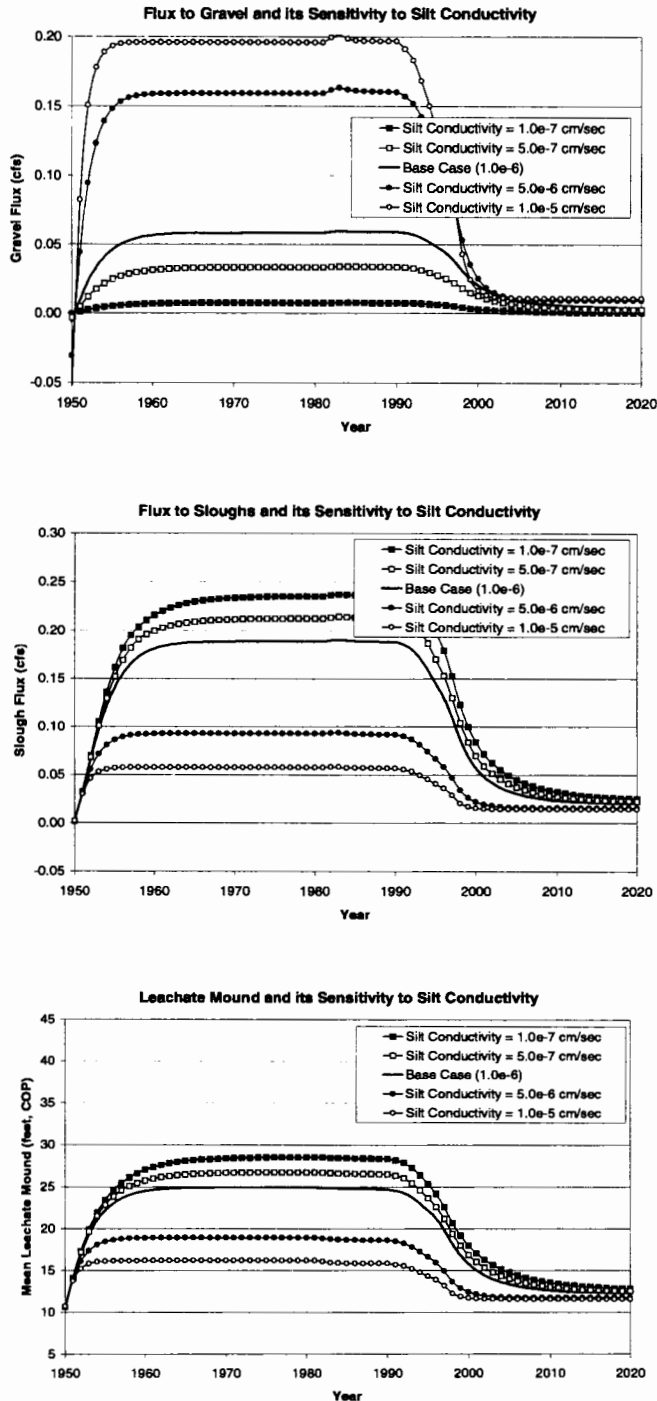
**Figure 31 - Sensitivity to conductivity in vertical direction in the silt. As the silt conductivity decreases, flow in the horizontal direction through the dikes becomes more pronounced and thus we see flux to the sloughs increase. As the silt conductivity increases, flow into the gravels increases and mounding decreases.**

Multi-Parameter Sensitivity Analysis:

Sensitivity analysis as described above, provides insight at the effect on the predicted contaminant mound as each parameter, one by one, is varied within a likely range. As each parameter is individually varied away from the base case, the predicted mound is always something other than the observed mound. There do however, exist many different parameter combinations which can reproduce the observed mean mound in the landfill. As an example, setting recharge high and conductivities high can produce the same predicted mound as setting recharge low and conductivities low. The magnitude and the distribution of the fluxes will change, but the predicted mound will stay the same (Figure 32). Multi-parameter sensitivity analysis looks at a range of parameters which are able to reproduce the observed mound.

Sensitivity analysis shows that the model is most sensitive to $K_{siltH}$, $K_{siltV}$, recharge, and to a lesser extent, specific yield. Multi-parameter analysis is done by perturbing all parameters within their likely ranges at the same time. For this analysis $K_{siltH}$ and $K_{siltV}$ are allowed to vary between $1 \times 10^{-3}$ cm/sec to $1 \times 10^{-7}$ cm/sec in 20 separate steps, recharge is varied from 7.5 in/yr to 12.5 in/yr in 2.5 in/yr increments, and specific yield is given values of 0.2 and 0.4. These combinations result in 2,400 model runs!

A contour is produced for each value of recharge and each value of specific yield, that shows the flux into the sloughs and the mound height given any value which lies in the ranges for $K_{siltH}$ and $K_{siltV}$. This analysis shows the likely ranges of

fluxes into the sloughs for all of the most likely non-unique solutions. Assuming a likely range of 20 - 30 feet for the contaminant mound, flux to the sloughs will range from 0.11 cfs (71,100 gallons per day, recharge = 7.5 in/yr, anisotropy = 40, mound = 20 ft.) to 0.27 cfs (174,500 gallons per day, recharge = 12.5 in/yr, anisotropy = 200, mound = 30 ft).

Figure 33 shows the results of the multi-parameter sensitivity analysis. The feasible range for $K_{siltH}$ and $K_{siltV}$ as determined by the landfill mound and the anisotropy are shown in the bold box on the contour plot. The flux contours passing through the box represent the upper and lower limit of flux into the sloughs and flux into the gravel aquifer. The base case is the point where the 25 foot mounding contour crosses the 90 anisotropy line.
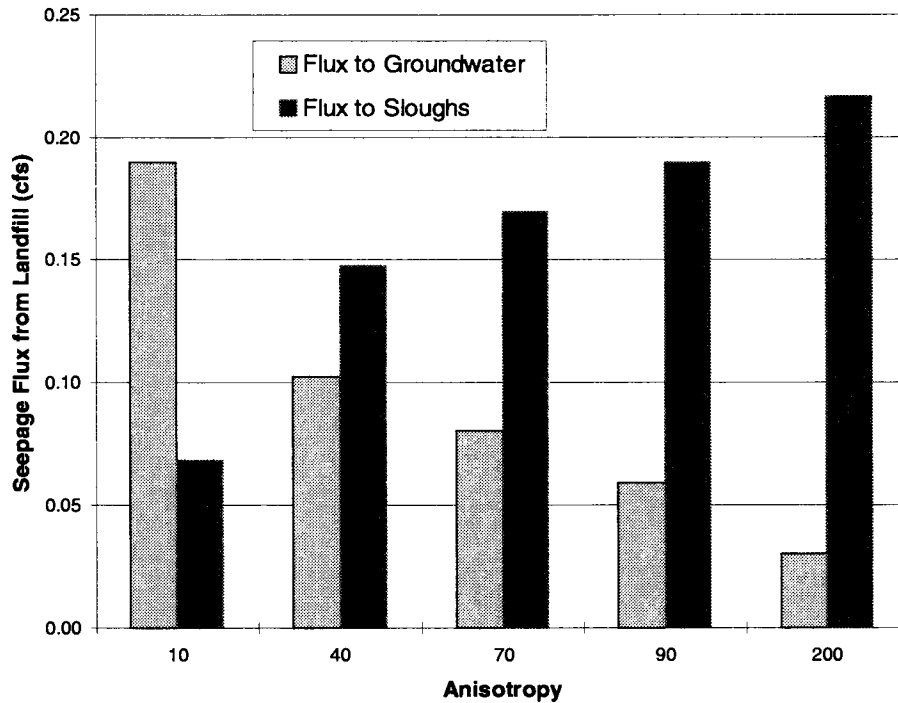
**Figure 32 - Effect of anisotropy and the trade off between lateral and vertical fluxes. As anisotropy increases, each component of the conductivity is adjusted to maintain a mean contaminant mound of 25 feet. The total flux out of the landfill is the sum of the flux to the sloughs and the flux to the groundwater. At higher anisotropy, flow is retarded in the vertical direction so flux to the groundwater is low, and flux to the sloughs is high. At lower anisotropy, the opposite is true.**
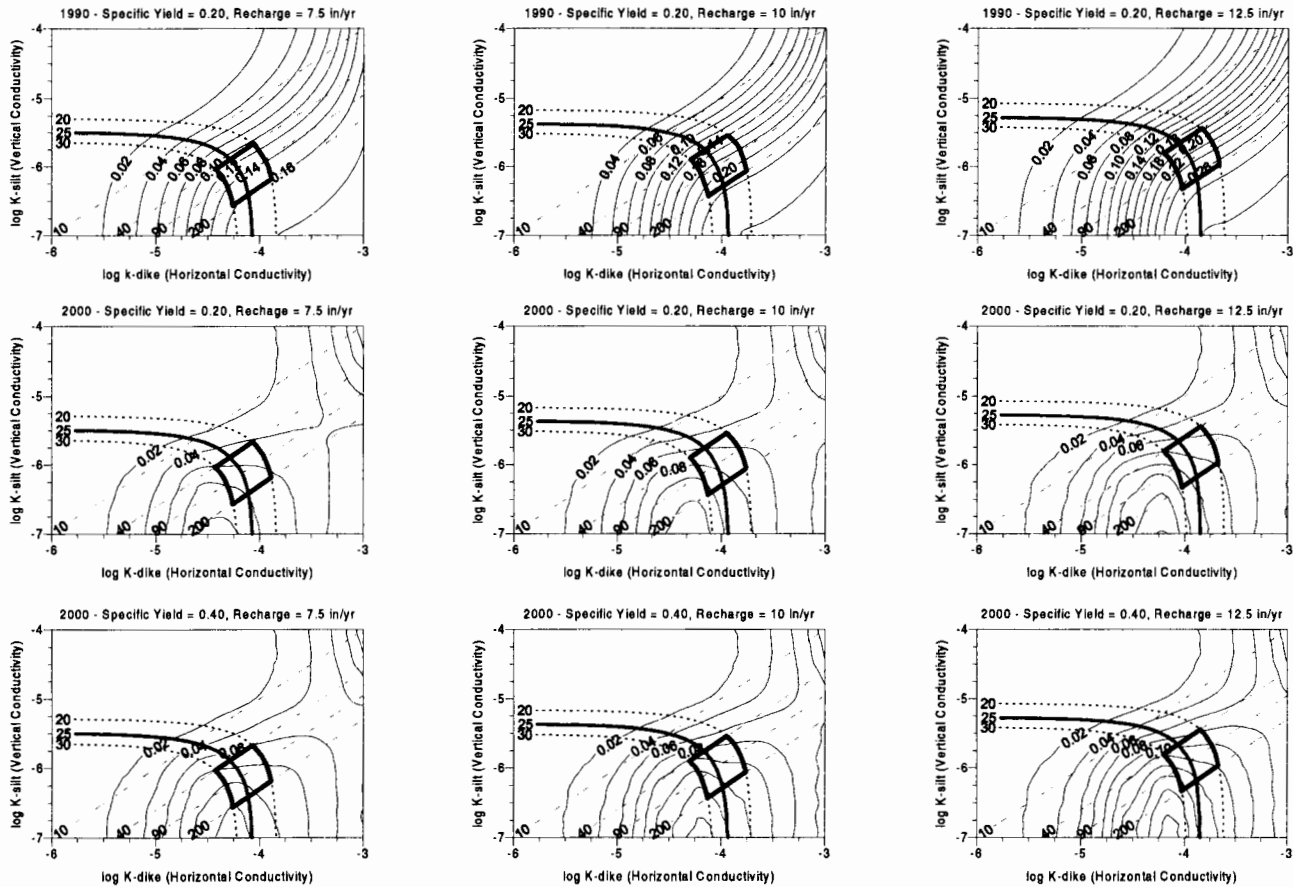
**Figure 33** - The flux contours passing through the 'feasible box' boundaries (dark, solid box in each plot) represent the likely limits of the predicted fluxes. The dotted lines represent the mean landfill mound, dashed lines are the lines of anisotropy, and the solid lines are the total flux into the sloughs. The flux values within the feasible conductivity region represent the likely flux variation due to parameter uncertainty.

## 3.7 Model Limitations:

The water balance model is accurate in regards to the conceptual model it is designed to predict. While the conceptual model is intended to be a realistic representation of the site, some assumptions and simplifications limit the accuracy of the predictions. The conceptual model assumes the landfill as a highly conductive medium that has no effect on the outflow of contaminant both laterally and vertically. If the landfill is assumed highly conductive, the assumption of a spatially averaged contaminant mound is a good one. Data show that the mound in the landfill is higher in the middle, and lower towards the edges (Sweet-Edwards/Emcon, 1989). In reality, these data are highly variable within the area of the landfill. Since the water balance model assumes an infinite conductivity in the landfill, this spatial variation of mounding and thus the spatial variation of the driving force for contaminant migration towards the gravels are not characterized. This is more likely a conservative assumption since head loss across the depth of the landfill is neglected and the full impact of the mound is 'felt' at the landfill/silt interface. Head loss across the depth of the landfill would reduce the mound, which is 'felt' at the landfill/silt interface thus reducing the gradient between the landfill and the gravels.

Dike and silt characteristics are also highly variable. In determining the horizontal and vertical conductivity in the silt for the water balance model, the mean landfill mound has to be maintained. This produces effective parameters for the entire landfill area. Since changes in the assumed value for the mean contaminant mound in the perimeter area of the dike, will correspond with an opposite change in horizontal

conductivity during calibration (for example if the mound decreases, the conductivity must increase), the flux values remain fairly constant and thus, the use of a mean contaminant mound is justified.

Because the area available for contaminant to migrate through the dikes is a function of the landfill mound, the governing equation is non-linear. Rather than simplify the non-linearity with the assumption of a constant area and solve the equation analytically, the governing equation is solved using numerical integration techniques. The model has the ability to solve the numerical equation explicitly, implicitly, or anywhere in between by adjusting a weighting factor in the input file. All model runs are conducted using a weighting factor of 0.5 indicating a balanced weighting between implicit and explicit. No comparison has been made between the numerical scheme used in the model and an analytical equivalent with an assumed constant dike area to test the accuracy of the model predictions.

Other parameters in the water balance model assume either site averages, time averages, or effective parameters. For instance, the horizontal dike conductivity is an effective parameter for the entire dike. It is known that the dike is heterogeneous, with areas that are actually made of refuse from the landfill (METRO, 1995). Areas of high conductivity within the dike may provide preferential paths for contaminant migration to the sloughs. An effective parameter accounts for this by calibrating to the data that is a reflection of real world conditions. The water balance model has no way of accounting for such heterogeneities nor would it be feasible to include that kind of detail in such a model. The water balance model is designed to provide a 'big picture'

77

view of the landfills impact on its surrounding environment answering the questions posed at the start of this project; how much contaminant is entering the sloughs, and how much contaminant is entering the deep aquifer. Answering these questions does not require the spatial distribution of each flux component.

# 4. Particle Tracking Model

The particle tracking model gives a preliminary, one-dimensional analysis to answer the question of timing for the contaminant flux to enter the sloughs and groundwater. Predictions of the gradient laterally across the dike and vertically across the silt thickness from the water balance model are used by the particle tracking model to track particles across the dike and downward through the silt. The particle tracking model predicts how long it takes a particle to cross the dike and enter the sloughs as well as penetrate the silt and enter the gravels by advection only. These results help to answer the question of how well the underlying silt protects the gravel aquifer, both now and in the future. There is no attempt to model the effects of dispersion in the particle tracking model. This section will outline the conceptual and mathematical models, parameter estimation and discretization, results and sensitivity analysis of the particle tracking model.

## 4.1 Conceptual Model:

As mounding in the landfill increases due to recharge into the landfill, the head difference between the mound in the landfill and the water level in the sloughs provides a driving force for contaminant to cross the dike and enter the sloughs. From Darcy's law, the larger the gradient, the faster the velocity across the dike. The water table level in the dike varies non-linearly as you move from the landfill through the dike and into the slough (Figure 34). This means that the velocity will be different depending upon where it is measured within the dike. In addition, the water table and

velocity changes over time as the contaminant mound increases and decreases. All these factors must be taken into account to properly track particles through the dike.
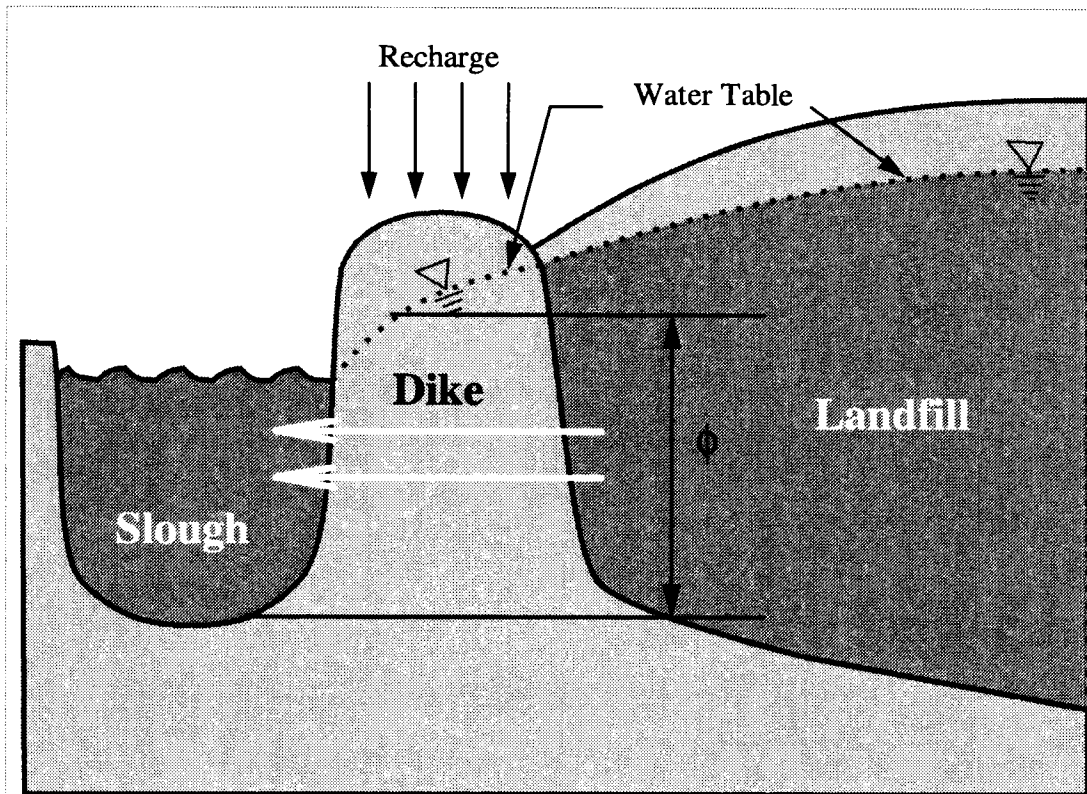


**Figure 34 - Conceptual representation of the praticle tracking model. Because the water level in the dike does not vary linearly between the landfill and the slough, particle tracking velocities will be different depending on the position of the particle within the dike. Recharge into the dike is the same as into the landfill except for after closure when recharge to the landfill goes to 10% of the non-closure rate (recharge to the dike stays the same). $\phi$ is the water level at any point in the dike.**

The vertical velocity in the silt is dependent upon the thickness of the silt below the landfill, the head difference between the landfill mound and the gravel aquifer, the vertical silt conductivity, and the effective porosity. With the assumption of a flat contaminant mound in the landfill, and the fact that the horizontal variability

of the head in the gravel is small, the main parameter controlling the head gradient between the landfill and the gravel is the thickness of the silt. The thickness of the silt varies considerably across the landfill and because of this the groundwater velocities also vary. In areas where the silt is thick, velocities will be slower than places where the silt is thin.

### *4.2    Mathematical Model:*

The particle tracking model is two separate models; one which tracks particles horizontally through the dike and into the sloughs, the other which tracks particles vertically through the silt and into the gravels.

Horizontal Particle Tracking Model:

The horizontal particle tracking model releases particles at the inside edge of the dike (landfill side) and tracks them through the dike in a variable velocity field until they reach the slough. This incorporates a two-step process. The first step determines the head at each point in the dike and the second step calculates the velocity at each point in the dike using Darcy's Law.

The water table height in the dike is represented mathematically by:

$$\frac{d}{dx} K\phi \frac{d\phi}{dx} + \varepsilon = 0$$

$\phi_s = f(t)$
$\phi_{lf} = f(t)$

(2)

81

where:

$d\phi$       = Difference in water levels across a distance $\partial x$ in the dike.

$\varepsilon$       = Rate of recharge to dike.

$K$       = Horizontal hydraulic conductivity within the dike.

$dx$       = Distance

$\phi_s$       = Water level in slough

$\phi_{lf}$       = leachate mound height in landfill

Since $K_{siltH}$ is assumed constant throughout the width of the dike, our governing

equation becomes:

$$K \frac{\partial}{\partial x} \phi \frac{\partial \phi}{\partial x} + \varepsilon = \frac{K}{2} \frac{\partial^2 \phi^2}{\partial^2 x} + \varepsilon = 0$$

where $K = K_{siltH}$.

      Integrating twice with respect to x, which is positive as you move from the

landfill to the slough, and solving for $\phi$, we get:

$$\phi = \sqrt{\frac{-\varepsilon}{K} x^2 + C_1 x + C_2}$$

(3)

where:

$C_1 = C_2 =$    constants of integration determined from boundary conditions.

Letting $\phi$ equal the contaminant mound height at x=0 (the edge of the dike next to the

landfill) and $\phi$ equal the slough height at x=L where L is the width of the dike, solving

for $C_1$ and $C_2$ gives us:

$$\phi = \sqrt{\frac{-\varepsilon}{K}x^2 + \left[\frac{\phi_s^2}{L} + \frac{\varepsilon L}{K} - \frac{\phi_{lf}^2}{L}\right]x + \phi_{lf}^2}$$

<div align="right">(4)</div>

where all like terms are defined as above.

Using Darcy's Law, we solve for the velocity at any point within the dike.

Darcy's Law in terms of pore velocity is stated as:

$$u = -\frac{K}{n}\frac{\partial \phi}{\partial x}$$

<div align="right">(5)</div>

where all like terms are defined as above and:

u   = Particle velocity in dike.
n   = Effective porosity.

Differentiating equation (4) with respect to x, and substituting back into

equation (5) gives the final equation for the velocity at any point in the dike. This is:

$$u = \frac{-\dfrac{\varepsilon}{n}x + \left[K\dfrac{\phi_s^2}{2nL} + \dfrac{\varepsilon L}{2n} - \dfrac{K\phi_{lf}^2}{2nL}\right]}{\sqrt{\dfrac{-\varepsilon}{K}x^2 + \left[\dfrac{\phi_s^2}{L} + \dfrac{\varepsilon L}{K} - \dfrac{\phi_{lf}^2}{L}\right]x + \phi_{lf}^2}}$$

<div align="right">(6)</div>

Equation (6) is an explicit formula for u at any point x within the dike, given

the contaminant mound height in the landfill, the hydraulic conductivity of the dike,

and the water level in the sloughs.

Vertical Particle Tracking Model:

The methodology is the same for particle tracking in the vertical direction as it is for particle tracking in the horizontal direction although the governing equation is a bit simpler. Velocity is calculated vertically through the silt by direct application of Darcy's Law. The head in the landfill, the head in the gravel layer, and the silt thickness determine the gradient. The gradient is multiplied by the vertical hydraulic conductivity in the silt and then divided by the effective porosity to give the velocity through the silt. Mathematically, this is:

$$u = \frac{K_{siltV}}{n} \frac{\partial \phi}{\partial z}$$

$$\phi_{lf} = f(t)$$
$$\phi_g = f(x, y, t)$$

(7)

where:

| | |
|---|---|
| u | = Particle velocity vertically in silt. |
| $K_{siltV}$ | = Vertical hydraulic conductivity in silt. |
| n | = Effective porosity. |
| $\phi_{lf}$ | = Leachate mound height in landfill |
| $\phi_g$ | = Head in gravel |
| $\partial z$ | = Silt thickness. |
| $\partial \phi$ | = Difference in head level between landfill and gravel aquifer. |

Substituting $\Delta z$ (the silt thickness) and $\Delta \phi$ (the difference in the landfill mound and the head in the gravel aquifer) into equation (7) for $\partial z$ and $\partial \phi$ for each point in the landfill, yields the final, explicit expression for the velocity in the vertical direction.

## *4.3    Boundary Conditions and Discretization:*

Both equations for the velocity in each direction are expressed explicitly and thus need no boundary or initial conditions. For the horizontal model, time discretization is important such that enough resolution is provided to capture the velocity variability across the dike. In the vertical model, spatial discretization is important so the variability in the silt thickness can be resolved to provide a 'smooth' distribution of velocities across the footprint of the landfill. The source code and further explanation are shown in Appendix A.

Horizontal Particle Tracking Model:

The length of the time step is very important in tracking particles across the dike. The governing equation is explicit so the time step has no bearing on the numerical stability of the solution, however, in order to capture the velocity variability across the dike, the time step needs to be small enough so that a smooth distribution of velocities can be reproduced. The model sets the time step to the amount of time necessary for a particle to travel 0.1 feet across the dike (the width of the dike is from stratigraphy data supplied by METRO and is averaged as 40 feet). If velocities are slow, then this time can be very large. To compensate, the model also sets an upper limit of 0.1 years on the time step. Thus, if a particle can move 0.1 feet in 0.1 years or less, then the total distance is incremented by 0.1 feet, and the total time is incremented by the distance traveled (0.1 feet) divided by the velocity as calculated from equation (6) across that span. If a particle cannot move 0.1 feet in 0.1 years or less, then the total time is incremented by 0.1 years and the total distance is

incremented by the time traveled (0.1 years) times the velocity as calculated from equation (6) across that span. This continues until the particle crosses the dike (40 feet) or the simulation time ends (the model simulates conditions for 70 years from 1950 to 2020).

Vertical Particle Tracking Model:

To track particles in the vertical direction, the landfill area is split into discrete areas, each with its own corresponding silt thickness. The landfill area is split into a uniform 130 by 80 node grid (75 foot spacing) with the silt thickness and the head in the gravel (from the two-dimensional head distribution of the regional flow model results) determined by the position of the cells within the grid. The model tracks a particle released at each cell downward through the silt by multiplying the time step by the appropriate velocity for each cell. To provide enough temporal resolution to assess contaminant penetration timing through the silt, a time step of 0.1 years is used. The model simulates conditions from 1950 to 2020.

## *4.4    Parameter Estimation and Calibration:*

Key parameters to the particle tracking models are the contaminant mounding heights, porosity, the horizontal hydraulic conductivity, the vertical hydraulic conductivity, head in the gravel aquifer, water level in the sloughs, and the retardation factor. The silt thickness and the dike thickness are established from stratigraphic profiles supplied by METRO. All parameters, except the retardation factor, are based on existing data or calibrated results from other models, so no calibration is needed for particle tracking.

Contaminant Mounding Heights:

The heights used for the contaminant mound are the results of the water balance model. To accommodate the time steps in the particle tracking model, the output for the water balance model is adjusted to 0.1 year time steps. If velocities are fast enough in the horizontal model such that the tracking limitation is the distance and not the time (see Section 4.3), then a linear interpolation between time steps is used to adjust the mound height. During the quasi-steady state period of constant mound height, this interpolation is not needed. The 0.1 year time step is small enough that the assumption of linear growth and dissipation of the landfill mound between time steps is valid.

Porosity:

Total porosity is defined as the ratio of the total volume of solids in a sample to the total volume in the sample (Freeze and Cherry, 1979). Effective porosity is the available porosity through which flow can occur. Not all pores spaces are connected in an aquifer, thus the effective porosity will always be less than or equal to (for a sample that has all pores interconnected) the total porosity. Effective porosity impacts the velocity of groundwater. By continuity, if the effective porosity is high, the velocity will be lower than if the effective porosity is small. A typical value for total porosity in silt is about 0.40 (Marsily, 1986). The effective porosity for the same medium will run from about 0.27 to 0.33 (Marsily, 1986), with some evidence pointing to even lower values. Since data do not exist for characterising effective

porosity in the silt, the total porosity of 0.40 is used. The limitations of this choice are discussed below.

Horizontal and Vertical Hydraulic Conductivities:

The base case for the conductivity components are the calibrated values from the water balance model. These are $K_{siltH} = 9 \times 10^{-5}$ cm/sec and $K_{siltv} = 1 \times 10^{-6}$ cm/sec (anisotropy = 90).

Head in the Gravel Aquifer and Water Levels in the Sloughs:

Head in the gravels are interpolated to the model grid from the steady state results of the regional flow model. Water levels in the sloughs are the same as those used for the water balance model (Section 3.4).

Retardation Factor:

The retardation factor is a contaminant specific parameter. It is only defined if a sorptive contaminant is to be modeled. The retardation factor has the effect of slowing a particle's migration through the chosen medium. The new velocity can simply be expressed as the groundwater velocity divided by the retardation factor. Thus, for a conservative contaminant, the retardation factor is 1. Changing this parameter to a value other than 1 shows the behavior of a sorptive contaminant moving horizontally through the dike or vertically through the silt.

The retardation factor modeled with the particle tracking model is an effective parameter. Preferential pathways, fracture zones, and conduit flow can offset the effect of even very high retardation factors by providing a direct path for contaminant

migration. This means that contaminants with very high retardation factors may still migrate out of the landfill through some type of direct path.

## 4.5    Results:

The particle tracking models address the questions of how long does it take for contaminant to penetrate the dike and how long does it take for contaminant to penetrate the silt. In a less quantitative way, it can also help determine the extent to which the underlying silt protects the gravel aquifer. Unlike the water balance model, which quantifies the various fluxes out of the landfill, the particle tracking models returns the time for a particle to cross the dike or silt (or the distance the particle travels if the time is not long enough for the particle to fully cross the medium).

Horizontal Particle Tracking Results:

Solute breakthrough time laterally through the dike is highly dependent upon the retardation factor. Simulations were made with retardation factors ranging from 1 to 200 (Figure 35). Results show that mobile solutes take anywhere from 2 years for a conservative solute (effective retardation factor equal to 1) to 30 years for solutes with an effective retardation factor of 50. A reactive solute with an effective retardation factor of 100 or greater may never penetrate the dike because the travel time necessary is greater than the time for the landfill mound to dissipate.

Vertical Particle Tracking Results:

The model predicts that contamination has broken through the silt below the landfill in three localized areas on the north side of the landfill near the North Slough

(Figure 36). For the base case of an anisotropy of 90, the time for contaminant to penetrate the silt is about 35 years. As anisotropy increase from 10 to 40 to 200, the breakthrough time goes from about 10 years, to 20 years to about 65 years. Figure 37 shows the effects of anisotropy on particle tracking.

Increasing the retardation factor simply increases the time by the same factor. Thus, the particle tracking model predicts that for all solutes with an effective retardation factor greater than about 2, silt penetration does not occur in the 70 year time frame which is modeled. At factors higher than 10, the contaminant is nearly immobile in the vertical direction (Figure 38).

Like the water balance model, the particle tracking model assumes the entire landfill, including the new expansion area is in operation since 1950. Since the new expansion area has about 8 years of history for contaminant migration to occur, the only place where solute is predicted to penetrate the silt is the western most 'hot spot'

## Conservative Contaminant
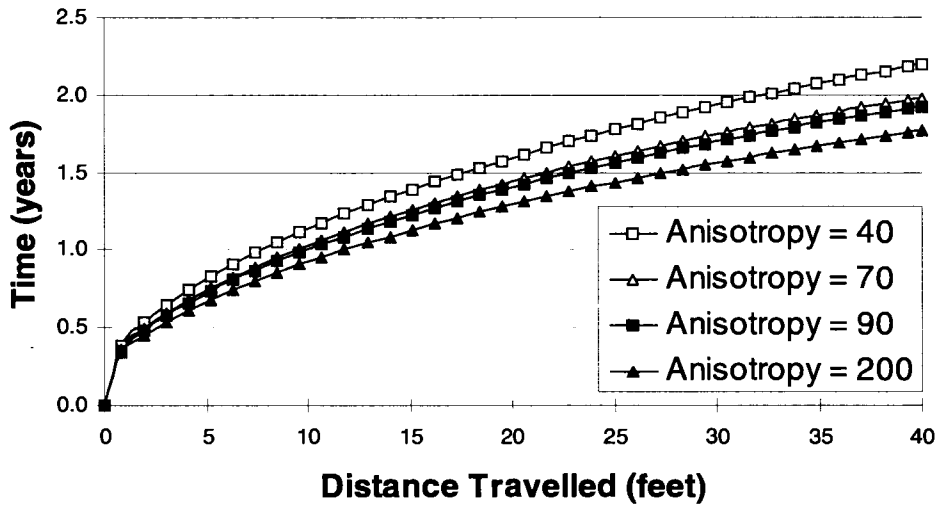


## Retardation Factor = 50
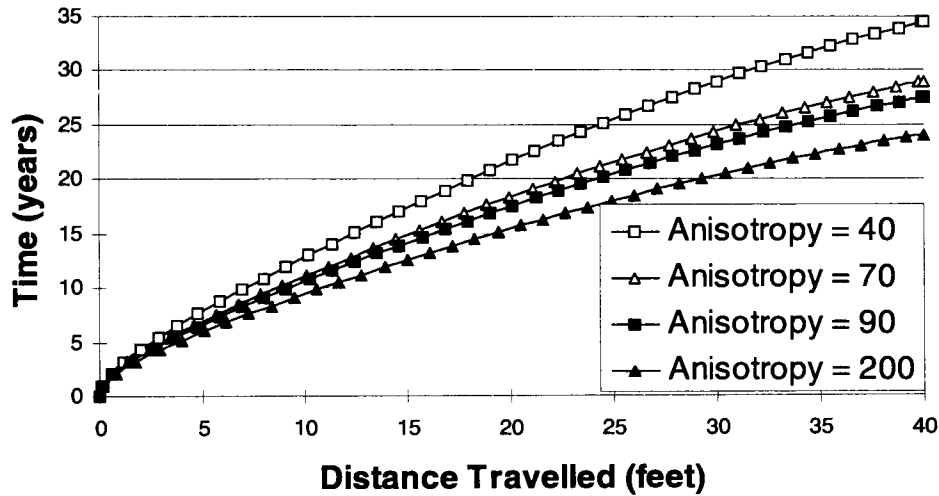


**Figure 35 - See caption and additional plots, next page.**

## Retardation Factor = 75
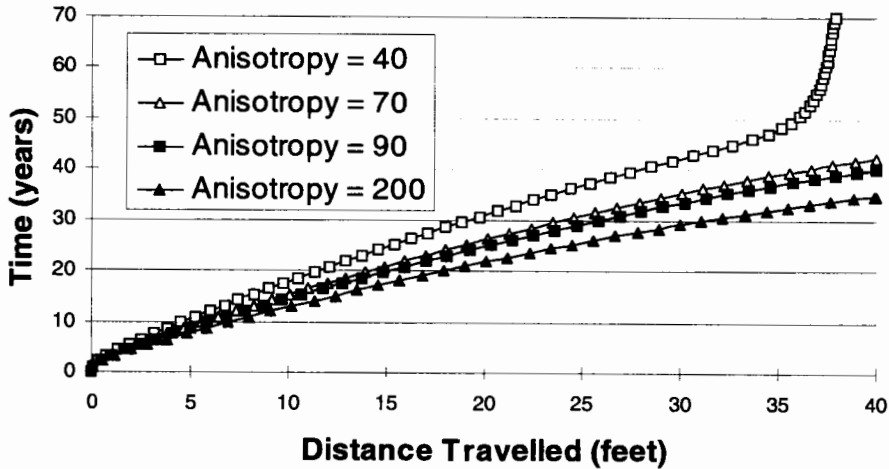


## Retardation Factor = 100
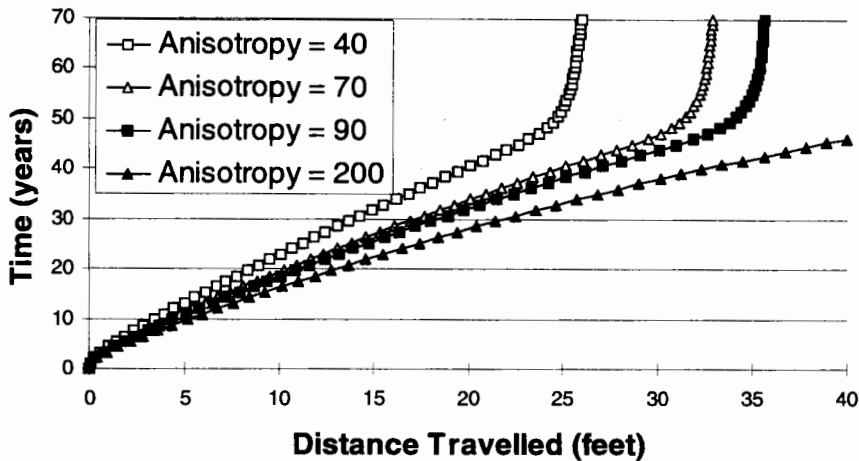


Figure 35 (See additional plots, previous page.) - Simple particle tracking for both conservative and reactive solutes in one-dimensional unsteady flow through the dike at varying anisotropy. As anisotropy increases, vertical movement declines in relation to the horizontal movement thus contaminant transport through the dike is possible at an effective retardation factor of 100 and an anisotropy of 200. For a conservative contaminant, penetration of the dike varies from 1.8 to 2.2 years for all reasonable values of anisotropy. Preferential pathways are not modeled here, thus contaminants with high retardation factors may be able to to migrate offsite via preferential pathways through the dike.
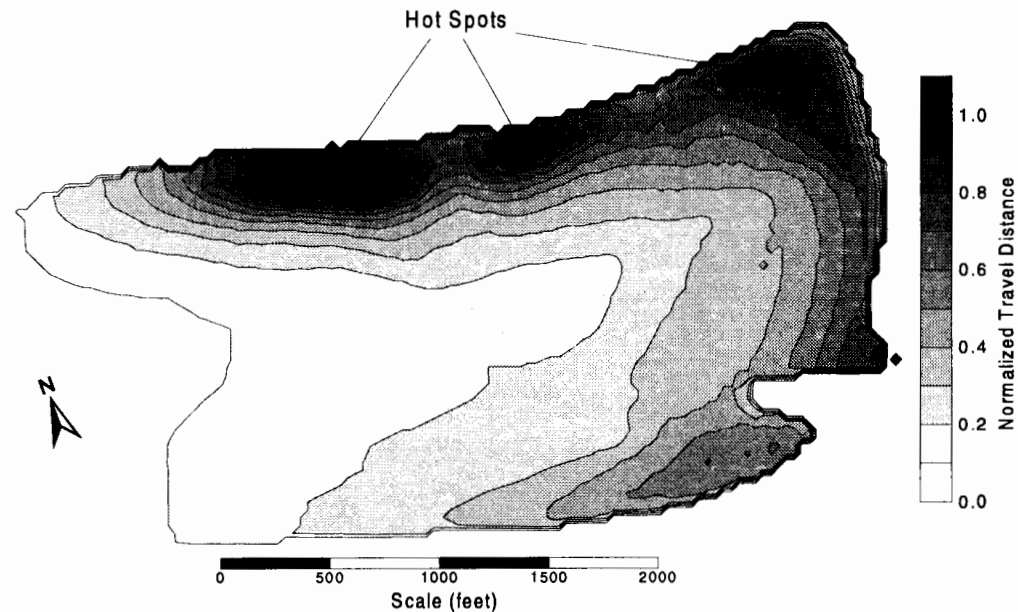
**Figure 36 - Spatial variation of normalized travel distance at the end of the model simulation (year 2020) for a particle moving vertically downward due to advection from the bottom of the landfill. A value of 1 means a particle has completely penetrated the silt layer at that point sometime during the simulation. Simulation assumes mean contaminant mounding, vertical flow, and the entire landfill being active at the start of the simulation. Three 'hot spots' are clearly shown, all along the north side of the landfill. The vertical penetration time in the hot spots is 30-40 years, making the western most hot spot the only place where contaminant can penetrate the silt since the other two hot spots occur in the new expansion area where only 8-10 years of movement is possible before the mounding dissipates. Solute penetration time in the other areas of the landfill exceeds 70 years thus penetration due to advection may never occur in these areas.**
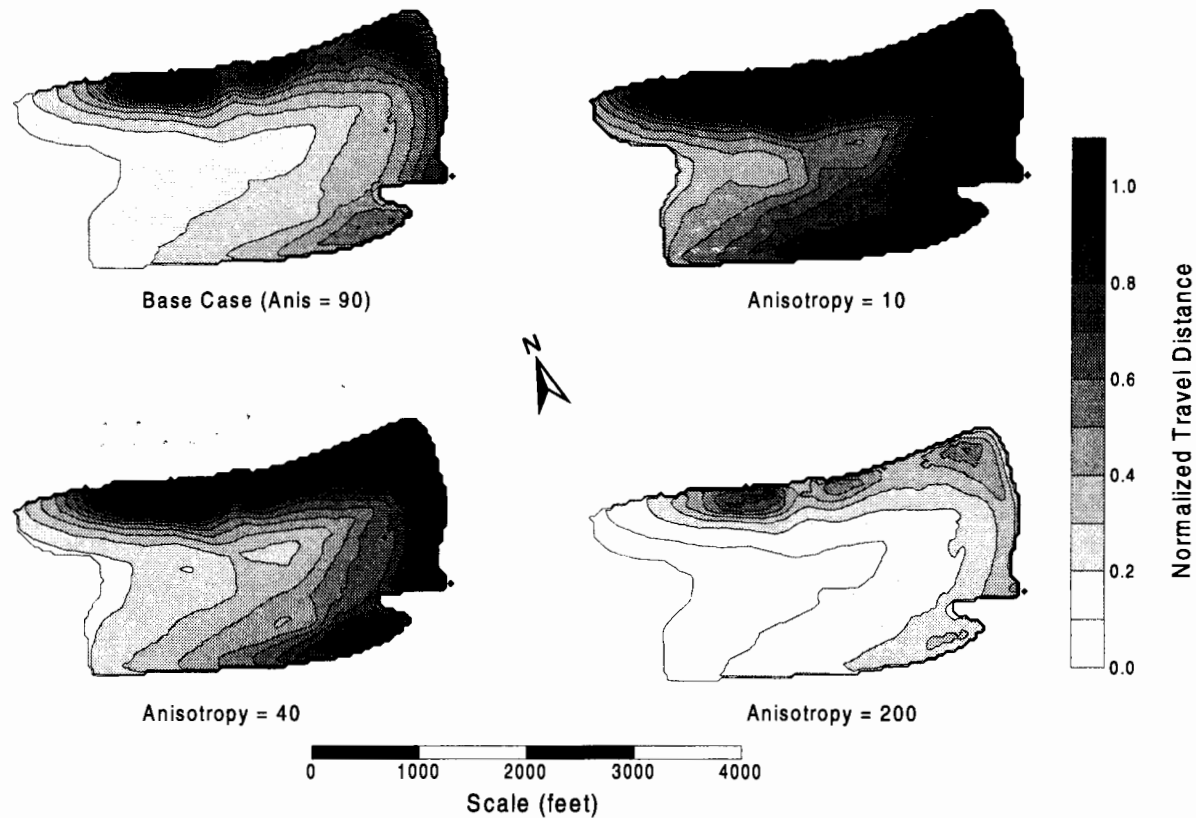
**Figure 37 - Silt penetration and the effect of anisotropy, year 2020.  A value of 1 means a particle has completely penetrated the silt layer at that point sometime during the simulation.  Base case is shown in upper left for comparison purposes.  At an anisotropy of 10, penetration is wide spread along the north and east sides of the landfill where the silt is the thinnest.  At an anisotropy of 200, the silt penetration in the hot spot is about 75% of the silt thickness.**

94
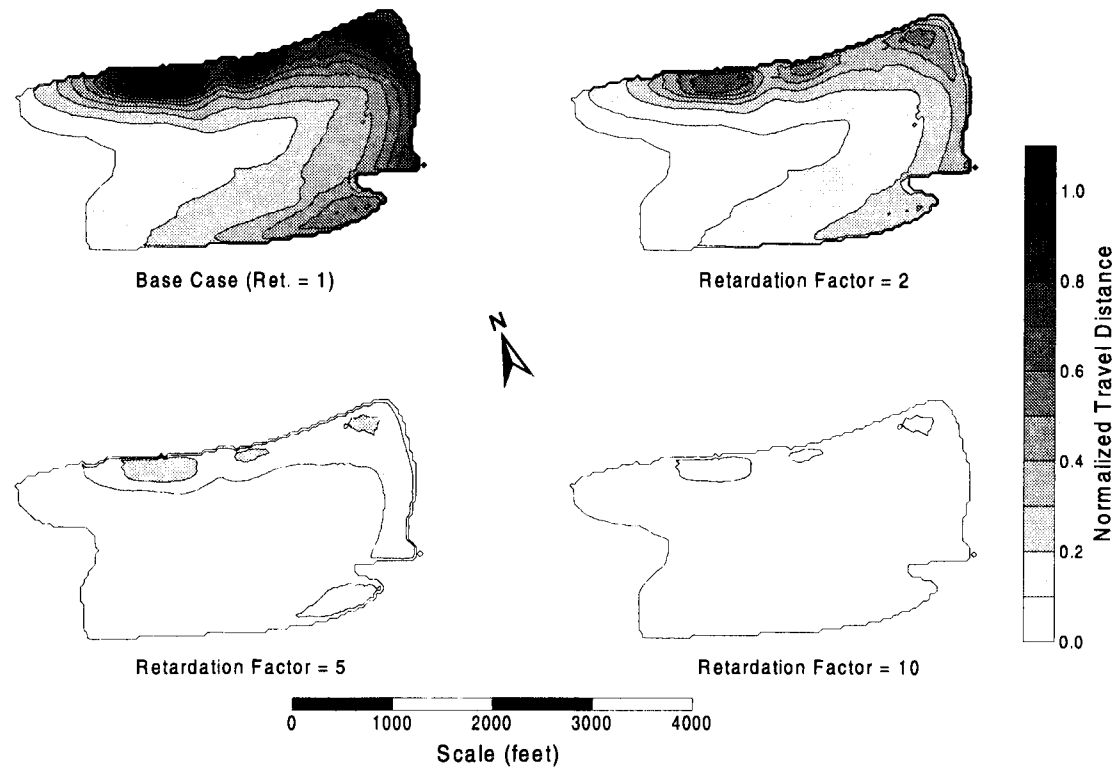
**Figure 38 - Silt penetration and the effect of retardation factor, year 2020, assuming constant anisotropy of 90. A value of 1 means a particle has completely penetrated the silt layer at that point sometime during the simulation. Base case is shown in upper left for comparison purposes. The transport of a sorptive contaminant vertically through the silt proves to be very sensitive to the retardation factory. At factors larger than 10, almost no penetration occurs. Even with a retardation factor of 2, contaminant penetrates a maximum of about 70% of the silt thickness in the hot spot area. Preferential pathways may allow for fast migration of contaminants with high retardation factors.**

## 4.6    Sensitivity Analysis:

The particle tracking model is an extension of the water balance model, using the same parameter values and results. The task of the particle tracking model is to return the time for particles to cross the dike or the silt, given a certain set of parameters as tested and calibrated from the water balance model. For this reason, time was not spent on sensitivity analysis. The effects of different anisotropy and retardation factors are discussed in the results section above.

## 4.7    Model Limitations:

Since the particle tracking model is an extension of the water balance model, the limitations are very similar. The assumption of a flat contaminant mound in the landfill effects particle velocities anywhere in the landfill where the mounding is different from the mean. Data show that the mound is higher in the middle of the landfill and lower along the edges. This may effect migration velocities across the landfill. In addition, the fact that the particle tracking model applies the effect of the entire contaminant mound to the landfill/silt interface, rather than factoring in head loss across the depth of the landfill, may produce velocities that are faster than reality would dictate.

The water balance and particle tracking models can be thought of as a series of steady state models. This means that for each year of the simulation, conditions in the landfill and the surrounding area do not change. Data show seasonal variation in the landfill mound, head gradients, and velocities. These seasonal variations may impact the migration of the contaminant plume.

Results from the particle tracking model show contaminant migration only by advection. In reality, dispersion plays a large role in the migration of the plume, essentially lowering the breakthrough times and increasing the landfills impact. Dispersion appears to be most important to contaminant migration before the mounding begins and after the mounding dissipates, or more simply, during times when transport due to advection is slow. Thus after closure, plume migration is probably faster than indicated by the particle tracking model.

## 5.    Regional-Scale Flow Model:

The regional scale flow model is used to simulate the large, regional-scale flow pattern, predict the interaction between the deep aquifer system and the Columbia and Willamette Rivers, and to provide boundary conditions for the local scale flow models. This section will address the conceptual and mathematical models, the boundary and initial conditions needed to run the model, key parameters, results, and sensitivity analysis.
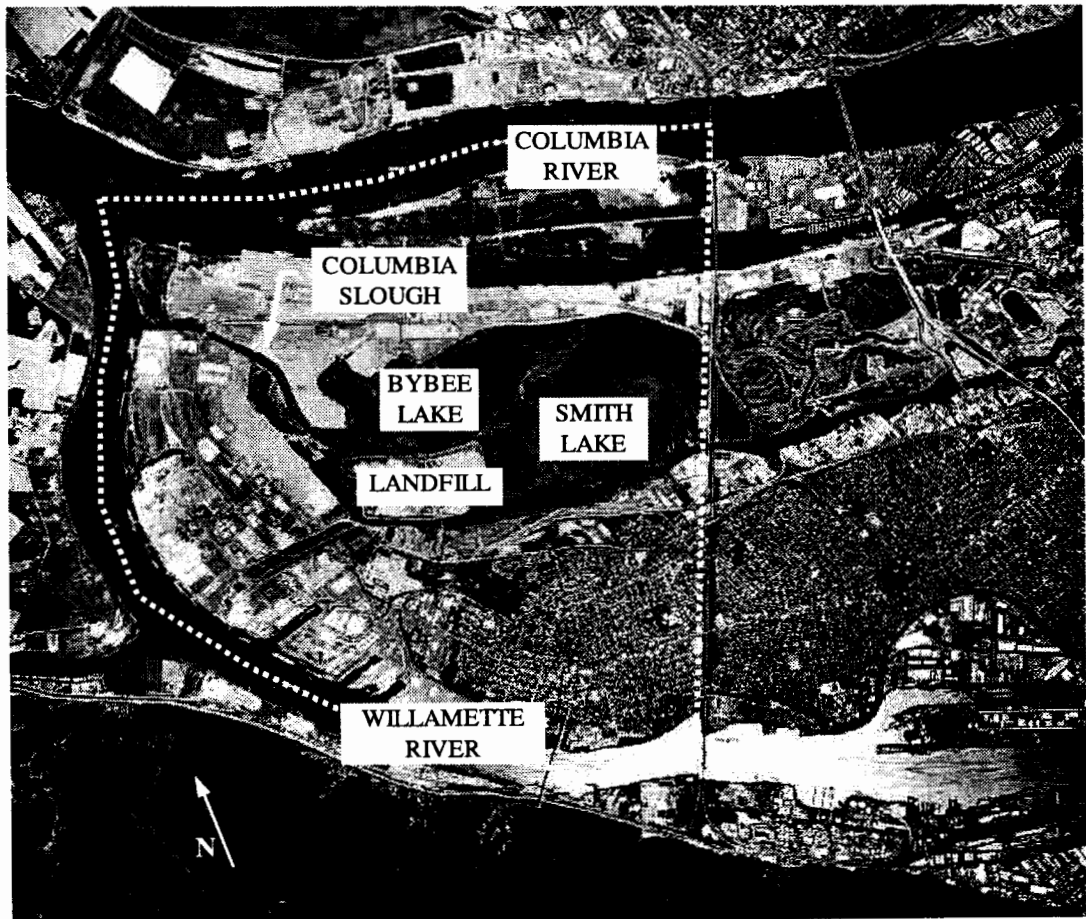


**Figure 39 - Model domain for the regional flow model. The white dotted line defines the model boundaries.**

## 5.1 *Conceptual Model:*

Conceptually, the regional model can be considered as a four layer system, bounded on three sides by rivers (north, south, and west sides) and the fourth side by a no flow boundary where flow is assumed to be parallel to the boundary (east side). The layers (Figure 40), defined from top to bottom are the floodplain sediments (the silts), the Columbia River sands (the sands), the Pleistocene gravels (the gravels), and the Troutdale formation. Recharge occurs regionally as rainfall that seeps through the silts and into the sands or into the gravels in places where sand is not present. Due to the high contrast of permeability's from the silt to the sand or gravel, flow is predominantly vertical in the silts. Source/sinks to the system include Smith and Bybee lakes, the Willamette and Columbia Rivers, and the sloughs, depending on the relative head in each unit. Ease of communication between the aquifers and Willamette and Columbia Rivers is determined by the hydraulic conductivity of the aquifers as well as the sediments in the rivers. Flow direction in the aquifers can change depending on the relative head values within the aquifers, the Willamette River, and the Columbia River. Temporally, the region is subject to tidal and seasonal changes. This has an effect on both magnitude and direction of flow. In addition, the temporal variability and flow direction changes have an effect on the dispersion characteristics of any contaminant plume that may be in the aquifers.
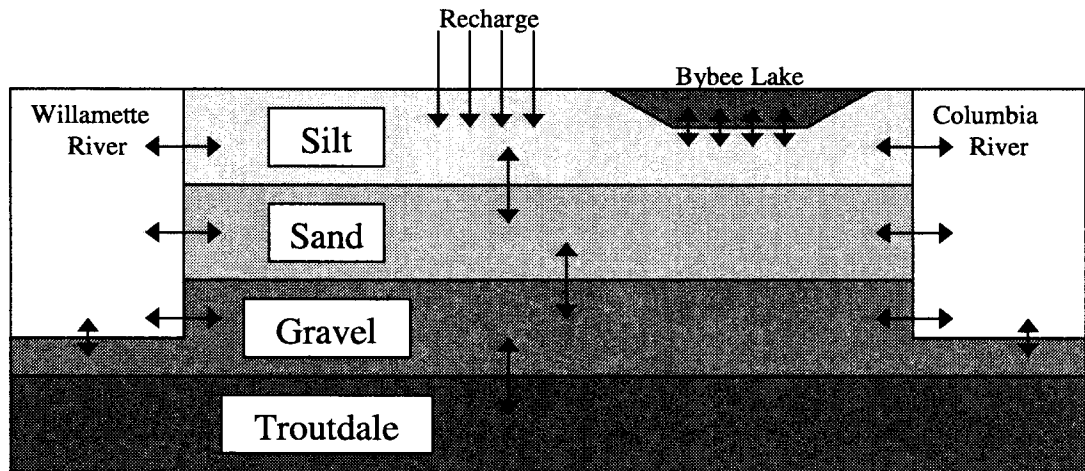
**Figure 40 - Conceptually, the regional flow model can be considered as a four layer system. The aquifer system (sand, gravel, and Troutdale formations) provides a conduit for hydraulic communication between the Columbia and Willamette Rivers. Due to the high contrast in permeability from the silt to the sand, flow in the silt is predominantly vertical. The underlying aquifer system can be considered as confined units with leakage between each other and the rivers. The arrows show possible flow directions to and from each unit.**

## 5.2    *Mathematical Model:*

The regional flow model is modeled as a 'quasi' three-dimensional system.

This means that each layer of the model is treated as a two-dimensional unit with a

leakance term to represent any communication between each unit. This makes each

layer vertically averaged.

The model utilizes MODFLOW (McDonald and Harbaugh, 1988).

MODFLOW is a block-centered, finite difference flow model developed by the USGS

which has been well tested and well verified. A commercial preprocessor and post

processor, The Department of Defense Groundwater Modeling System (GMS, version

2.1 beta) is used as an interface to the code and the user for the final version of the

model, allowing data to be input directly.

100

Governing Equations:

The governing equation for two-dimensional groundwater flow in each layer

is:

$$S \frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x} T_x \frac{\partial \phi}{\partial x} + \frac{\partial}{\partial y} T_y \frac{\partial \phi}{\partial y} + S/S$$

$\phi = f(x, y, t)$

(8)

where:

| | |
|---|---|
| S | = Confined storage coefficient (or specific yield for first layer). |
| $\phi$ | = Head in aquifer. |
| $T_x, T_y$ | = Directional component of transmissivity (aquifer thickness × hydraulic conductivity)...or hydraulic conductivity for first layer. |
| S/S | = Sources or sinks. |

For the regional model, the sources and sinks present are leakage between layers,

seepage to and from surface water bodies, and recharge.

The process of adding a leakage term to the source and sink term, and

vertically averaging the heads in each geologic unit are the defining qualities of a

quasi three-dimensional model. In the source sink term, leakage between layers is

given by:

$$q_{leak} = \frac{K_{ez}}{b} \Delta \phi$$

(9)

where:

| | |
|---|---|
| $q_{leak}$ | = Darcy velocity between layers. |
| $K_{ez}$ | = Effective vertical hydraulic conductivity. |
| b | = Distance between centers of each layer. |
| $\Delta \phi$ | = Difference in head between each layer. |

To obtain a flow volume, the area of flow is multiplied by the Darcy velocity. The leakance term ($K_{ez}$/b) may be modified to provide an effective leakance that describes any flow limiting aquitard between the two computational layers. $K_{ez}$ is the harmonic mean of the vertical hydraulic conductivities between each layer. Water tends to seek the path of least resistance thus flow in the low conductivity silt is predominantly vertical.

Recharge is added directly to the silt layer or the top most active cell in the case where there is no silt layer present or the water table has dropped below the bottom of the silt. In the general equation for the source and sink flow, recharge is represented as a constant rate value for each stress period represented by q such that:

$$q_{recharge} = q(x, y, t)$$

(10)

For interaction between the surface water bodies and the underlying aquifers, two conditions must be considered; first is when the head in the aquifer is higher than the river bottom, and secondly is the when the head in the aquifer is lower than the river bottom. In general, flow between surface water and groundwater is represented by:

$$q_{sw} = \frac{K'}{b'}(\phi_r - \phi)$$

(11)

where:

$q_{sw}$ = Darcy velocity between geologic layer and surface water.

K' = Effective hydraulic conductivity between the surface water body and the aquifer.

b' = Distance from top of river sediments to center of computational cell.

$\phi_r$ = Water level of surface water.

$\phi$ = Head level in geologic unit.

When the head in the aquifer is higher than the river bottom, $\phi$ is defined as above. When the head in the aquifer drops below the river bottom, $\phi$ is the elevation of the river bottom. b' is the thickness of the river sediments plus the distance from the river bottom to the center of the computational layer. $K'$ is the harmonic mean of the hydraulic conductivity of the river sediments and the computational layer.

## 5.3    *Boundary and Initial Conditions and Model Discretization:*

The model consists of four computational layers representing, the silt, the sands, the gravels, and the Troutdale formation. Each layer is treated as homogeneous in the vertical direction with no vertical flow being simulated within each layer. The model is discretized into a 50 by 59 grid with nodal spacing equal in all directions of 450 feet. The unsteady model is run with a time step of ten days for one year. The steady state model returns a layer averaged and yearly averaged head distribution while the unsteady model returns the layer averaged head distributions for each 10 day period. The regional model represents the landfill as a surface water body with a constant head throughout the simulation of 25 feet (COP).

Initial conditions are developed by assigning equal head values throughout the model domain and running the model until a steady state condition is achieved using

the 1950 data. This is then the initial condition for the yearly averaged version of the model. For the one year transient simulation, the initial conditions are developed by using 'best guess' values and then looping the simulation such that the initial conditions are far enough distant that they are 'forgotten'. This is known as a cyclic initial condition.

The Willamette River defines the south and western boundaries. Where the riverbed completely penetrates a layer, (this depends on the local thickness of each layer at each point along the river – see Figure 10 and Figure 13) the boundary is designated a head dependent flux boundary dependent upon the river level. To calculate flow between the river and the aquifer, the condition is modeled with a fully penetrating riverbed characterized by an effective conductance that is smaller than usual to compensate for the partial penetrating condition (Figure 41). The effective conductance represents the harmonically averaged conductance of the river sediments and the material in the aquifer.

The north end of the model is bounded by the Columbia River where the same conditions hold as for the Willamette River except that the silt layer is primarily non-existent underneath the Columbia. The eastern boundary was chosen such that flow between the Willamette and Columbia Rivers is assumed to flow parallel to the boundary and thus stipulate a no flow condition at the boundary. North Portland Road, which runs straight and perpendicular to both rivers, provides a convenient cut off line for the eastern model boundary. A model of the Portland Basin by the USGS (Morgan and McFarland, 1994) shows pumping wells along this boundary (Figure 42

and Figure 43). As modeled in the USGS model, the interaction of the pumping wells
along the North Portland Road boundary with Smith and Bybee Lakes is great enough
so that the flow along the boundary is not 'felt' at the landfill site; the lakes dampen
any drawdown in the area. The USGS model does not explicitly model the silt and
thus this interaction may be overstated. If this is the case then the pumping effects
may have a greater zone of influence than indicated in the USGS results and may
impact plume migration around the landfill. Further study is needed to determine if
this is true or not. Without pumping however, the North Portland Road boundary is
distant enough so as not to have noticeable influence in the landfill area. The no-flow
assumption without pumping is further justified through numerical experimentation
and sensitivity analysis.
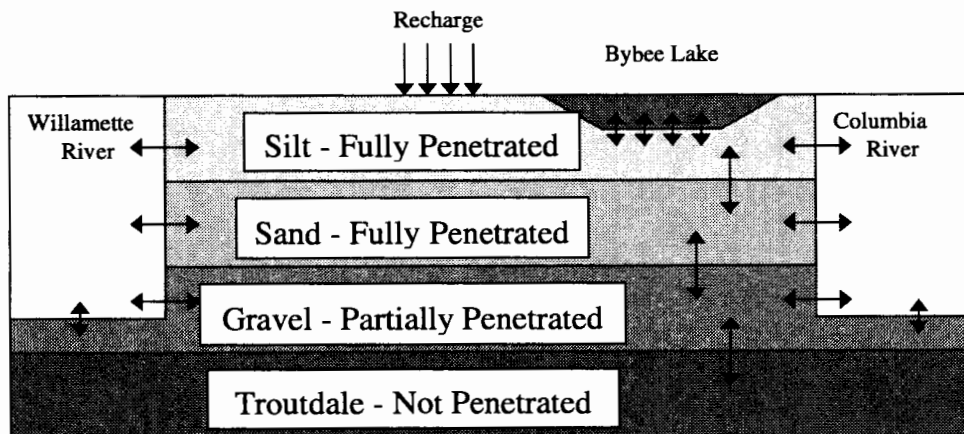


**Figure 41 - Layers 1 and 2 are fully penetrated by the river. In this case, the boundary condition is a head dependent flux boundary. For layer 3, which is partially penetrated, the boundary condition is a head dependent flux boundary which is reduced through the leakance term to account for the partial penetration. Hydraulic conductivity for the flux condition is determined by using the harmonic mean of the river sediments and the aquifer material. For a non-penetrated unit such as unit 4, the lateral boundary is a no-flow boundary.**

# USGS Portland Basin Model Results
## Head Contours in US (ft, MSL)
## St. Johns Landfill Region
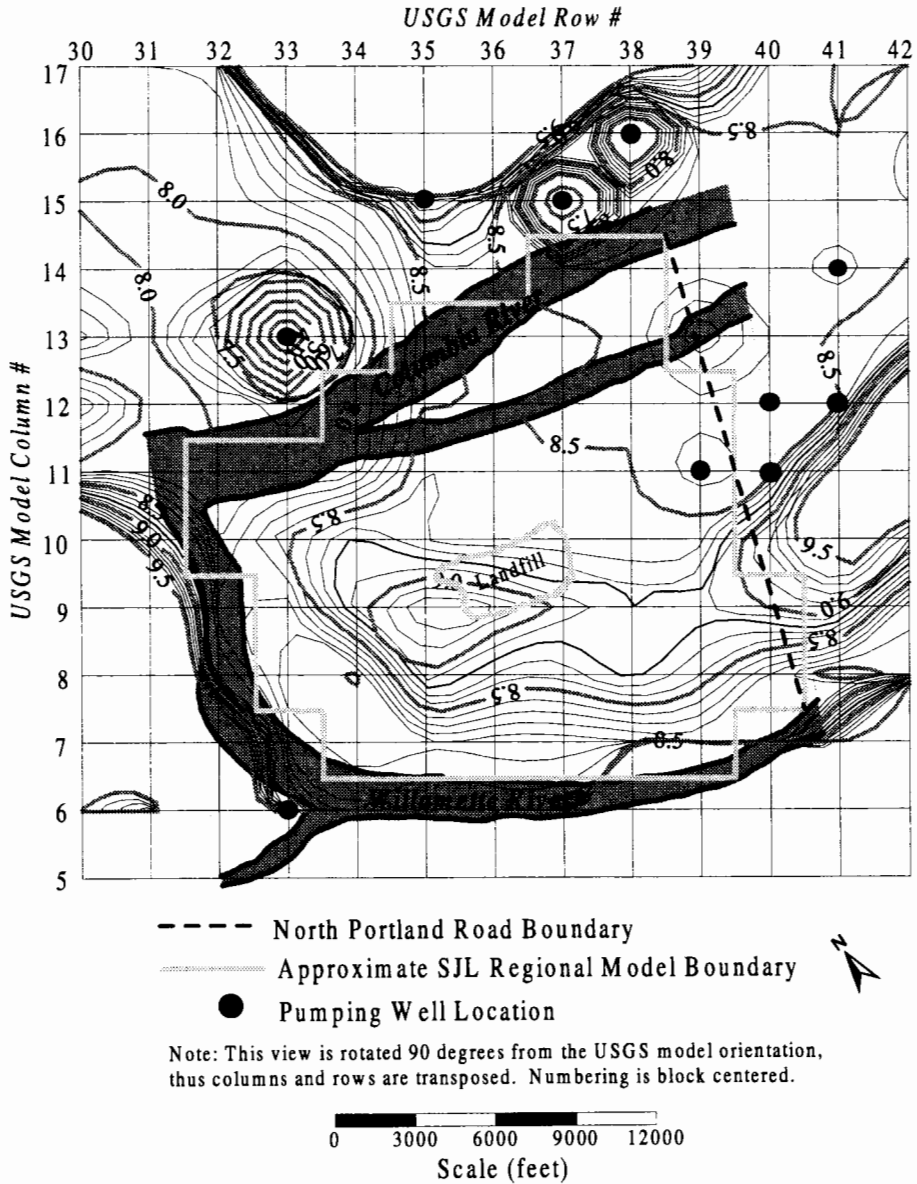


**Figure 42 - Head contours in the unconsolidated sediments (US) as predicted by the USGS Portland Basin Regional Model. The US corresponds to the silt, sand, and gravel layers in the St. Johns Landfill regional model. The approximate outline of the SJL regional model domain is shown in light gray with the North Portland Road no-flow boundary shown by the dotted line.**

106

# USGS Portland Basin Model Results
# Head Contours in TG (ft, MSL)
# St. Johns Landfill Region



**Figure 43 - Head contours in the Troutdale Aquifer (TG) as predicted by the USGS Portland Basin Regional Model. The approximate outline of the SJL regional model domain is shown in light gray with the North Portland Road no-flow boundary shown by the dotted line.**
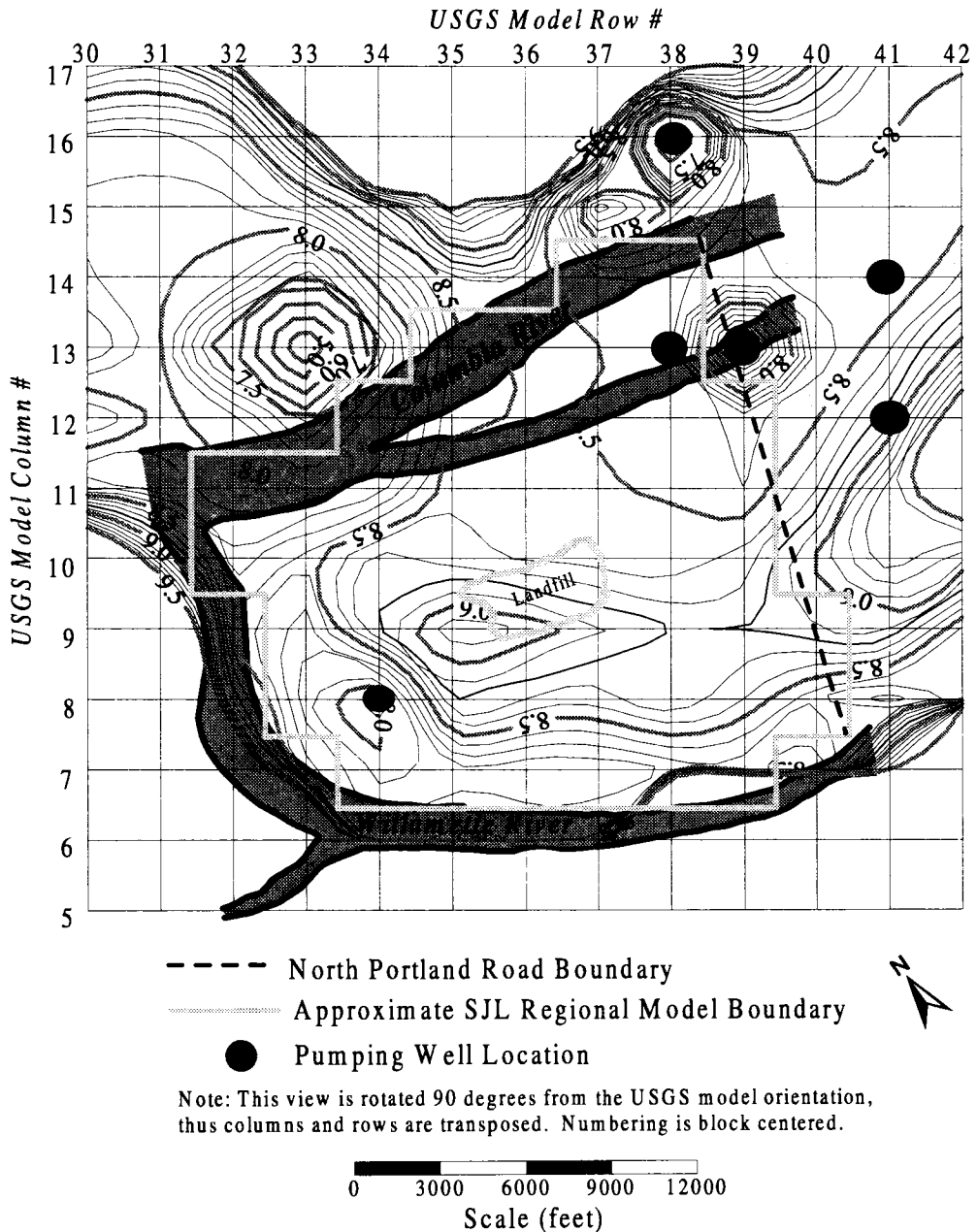
On top, the boundary is the water table in the silt (or the next active layer where the upper layer is not present or goes dry). The undifferentiated sediments are the bottom boundary, which is considered a prescribed flux boundary as determined by the flow predictions of the USGS Portland Basin model. Upwelling from the undifferentiated sediments to the Troutdale Aquifer in the local landfill area is slight (Figure 44), representing less than 4% of the total inflows into that area (mainly from recharge). The bulk of the upwelling in the regional area occurs near the Willamette and Columbia Rivers.

# USGS Portland Basin Model Results
## Vertical Flow* (cfd/cell area) from TG to UF
## St. Johns Landifll Region
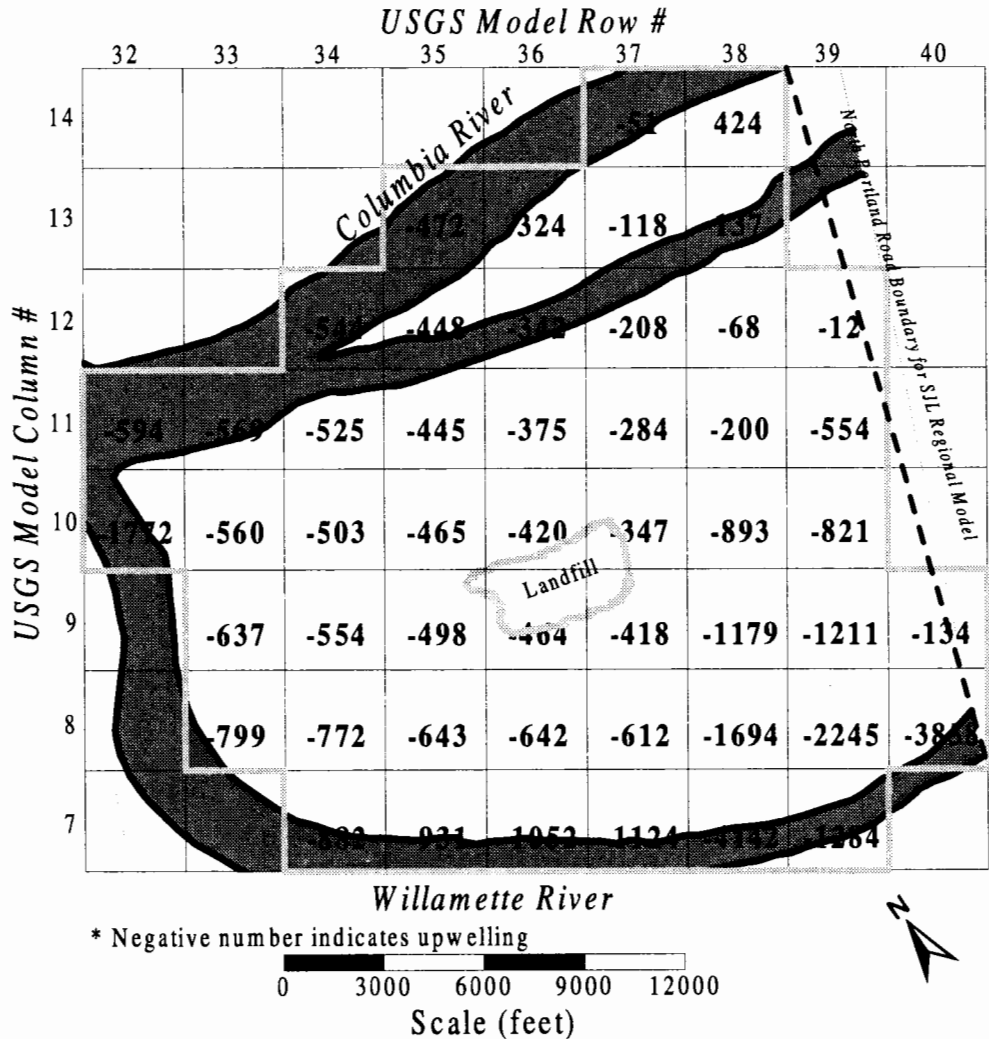
### USGS Model Row #



* Negative number indicates upwelling

Scale (feet)

Approximate SJL Regional Model Boundary

Note: This view is rotated 90 degrees from the USGS model orientation, thus columns and rows are transposed.

**Figure 44 - Flux from the Troutdale Aquifer (TG) to the undifferentiated sediments (UF) for use as the bottom boundary condition in the regional flow model as predicted by the USGS Portland Basin Model. Negative numbers equal upwelling from the UF to the TG. Flux values are total for each 3000 by 3000-foot cell. Upwelling in the local landfill area is slight, averaging about .25 in/yr or less than 4% of the recharge flux to the area.**

### 5.4    *Parameter Estimation and Calibration:*

Key inputs for the model are the stratigraphy which describes the geology

within the model domain, the hydraulic properties of each geologic unit, river levels in

the Columbia and Willamette Rivers, Smith and Bybee Lakes, and the North and

Columbia Sloughs, and the space-time variation of the recharge rate. The calibration

and basis for each parameter are described below.

Recharge:

Recharge is added directly to the silt layer (or the next wet layer in the case

where the silt layer goes dry or silt is not present). Annual mean recharge is based on

a function of precipitation given by:

$$\varepsilon(x, y, t) \approx 0.48\, P(t) - 0.354\, \alpha(x, y) - 4.79$$

(12)

where:

$\varepsilon$    = Rate of recharge.
P    = Yearly precipitation.
$\alpha$    = Ratio of impervious to total area.

This formula is based on a regression analysis (Bauer and Vaccaro, 1989)

developed from the results of the deep percolation model simulation for the Portland

basin (Snyder et al, 1994, page 16). Generally, the area to the north and northwest of

the landfill show lower percentages of impervious area and thus higher levels of

recharge. To the south, the impervious area is higher due to more development.

Within the model domain, recharge ranges from about 4 inches per year to 12 inches

per year. The overall spatial distribution of the rate of recharge is shown Figure 45.

Precipitation data for the model is obtained from the Oregon State Climatologist at

Oregon State University.
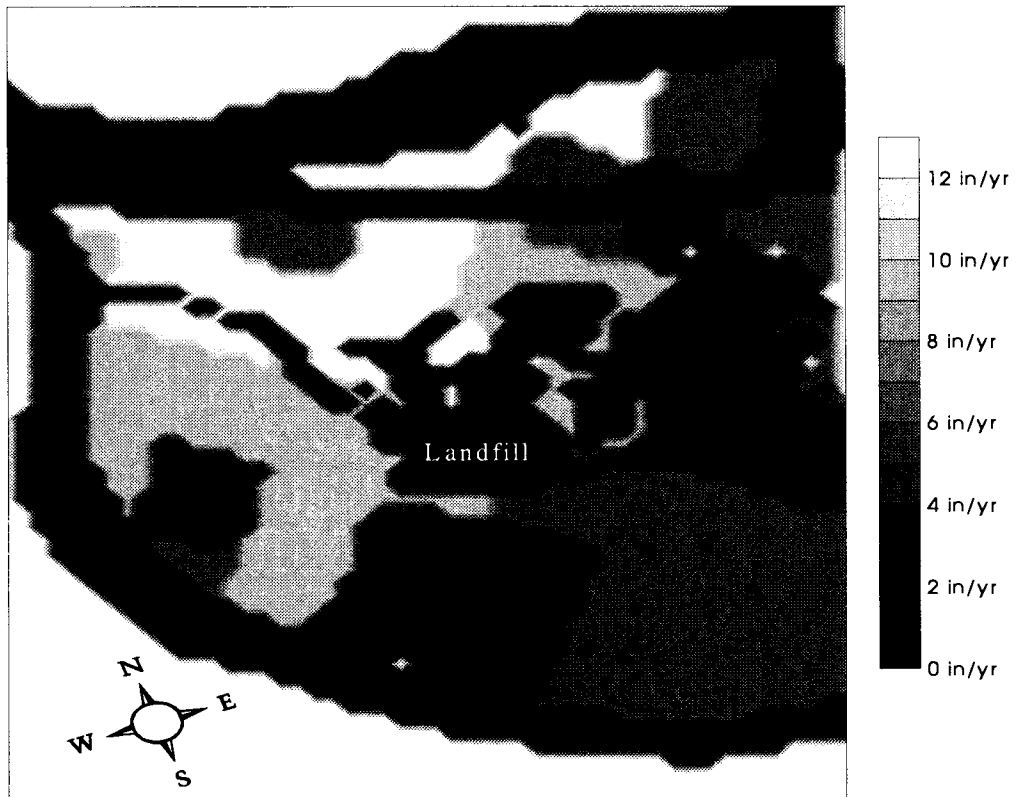


**Figure 45 - Spatial distribution of recharge for the calibration year of 1989. Recharge ranges from 2 to 12 inches per year depending upon the amount of impervious area that is present. Recharge values are derived from a regression formula based on the results of the deep percolation model simulations for the Portland basin done by the USGS.**

Surface Water Levels:

River levels for the steady state model are averaged out for a 44-year period

from 1950 to 1994 and are obtained through Parametrix, Inc. from the US Army Corps

of Engineers.

Water levels in the Columbia and Willamette River after 1995 use the average of the last 10 years of data (1986-1995). Measured water levels in the Columbia Slough and Willamette River averaged out over a simulation time step show little difference from that of the Columbia River and are always assumed the same. For Smith and Bybee Lakes, prior to 1983, the water levels are considered the same as that in the Columbia River since the lakes were open to free flow from the North Slough. In 1983, a restrictive dam was built between the North Slough and the outlet of Bybee Lake that resulted in water levels which are 2-3 feet higher than that in the Columbia River. The larger differences in water levels occur during times of dry periods so a three-foot difference is used during the relatively dry calibration year of 1989. The long-term mean uses a two-foot difference. A maximum of 12 feet is used for the water level in the lake due to water escapement over the top of the dam at levels higher than that. Using 12 feet assumes a dam height of 11.775 (COP) plus the level of the water over the top of the dam.

Conductivity:

Vertical and horizontal conductivities are determined differently depending upon the geologic layer being addressed. For the silt layer in the immediate vicinity of the landfill, the calibrared conductivities from the water balance model are used. Outside the immediate area, conductivity values in the silt are calibrated to provide the necessary recharge to the sands and gravels to meet head level data in selected wells. An assumption is made that the influence of the landfill on the silts by compaction is great enough such that two conductivity zones are needed. Conductivity in the sand is

assumed to be one-half that of the gravels based on limited pump test data in the area (ERM Southwest, 1992). The hydraulic conductivity in the Troutdale formation is considered to be one-fifth that of the gravel conductivity and has very little effect on heads in either the gravels or the sands due to little vertical flow between the gravels and the Troutdale formation (USGS, 1994). The ratios of the conductivities are not changed during calibration, only the conductivity for the gravel is changed. Base case values are shown in Table 2.

| Conductivity (cm/sec) | Silt - By Landfill | Silt - Outside Landfill | Sand | Gravel | Troutdale |
|---|---|---|---|---|---|
| Horizontal | $9.0\times10^{-5}$ | $9.0\times10^{-5}$ | $3.18\times10^{-2}$ | $6.35\times10^{-2}$ | $1.27\times10^{-3}$ |
| Vertical | $1.0\times10^{-6}$ | $9.0\times10^{-6}$ | $3.18\times10^{-3}$ | $6.35\times10^{-3}$ | $1.27\times10^{-3}$ |

**Table 2 - Conductivity values for the silt, sand, gravel, and Troutdale aquifer as used in the regional flow model. Vertical conductivity in the silt under the landfill is reduced to account for local data and compaction due to the landfill.**

Conductance in the river sediments is representative of the effective conductance determined by the vertical hydraulic conductivity of the river sediments and the vertical hydraulic conductivity of the silts. This effective conductance is the limiting factor for recharge/discharge from surface water to the underlying aquifers. Since the effective conductance is a pure mathematical representation, there are no means of measuring the equivalent conductance factor and thus it is determined exclusively through calibration. The base case value is 50,000 ft$^2$/day (18,300,000 cm$^2$/day).

Storage Coefficient:

The storage coefficient for a confined aquifer is defined as the volume of water that a unit volume of aquifer releases from storage under a single unit decline in hydraulic head. The storage coefficient is determined via calibration and is evaluated as the product of the specific storage and the corresponding layer thickness (Li, et. al., 1995). The silt, sand, and gravel layers have a calibrated value for the storage coefficient of $1.0 \times 10^{-4}$ times the unit thickness while the Troutdale aquifer is calibrated to $1.0 \times 10^{-5}$ times the unit thickness.

Geology:

All stratigraphy and geologic layout is provided directly from METRO.

Calibration:

The model is calibrated on a set of head from perimeter wells in the gravels around the landfill gathered from Nov 2, 1988 to Nov 1, 1989 (Sweet-Edwards/EMCON, 1989). 'Known' parameters (physical geology, boundary conditions, and recharge) are not altered during calibration while the others are altered on a trial and error basis until a 'fit' to the data is reached. Parameters are started with a best guess or a typical published value and are altered within a likely range of variation.

The calibrated model appears to capture the regional flow fairly well in the area adjacent to the landfill. Predictions describe the general shape, magnitude, and timing of each of the hydrographs with the exception of well G8c, which appears to be

more dissipated than the actual observed response. This is probably due to local heterogeneities in the region of well G8c which are unaccounted for in the model. The general agreement indicates the model is capturing the dominant flow processes in the landfill area.

## 5.5    Results:

The annual mean heads as used in the water balance model for each computational layer for the time prior to 1982 are shown in Figure 46. With the exception of the silt layer, the head contours generally follow the shape of the Willamette and Columbia rivers. The silt layer is strongly influenced by large surface features such as Smith and Bybee Lakes, the landfill, and the Columbia Slough. An elongated mound running generally east-west, is formed in the center area of the model domain between the rivers. The head gradient at the top of the mound is small and increases both to the north and south as you move towards the rivers. The landfill site is located almost directly over the flat area of the mound providing a small gradient in the local landfill area.
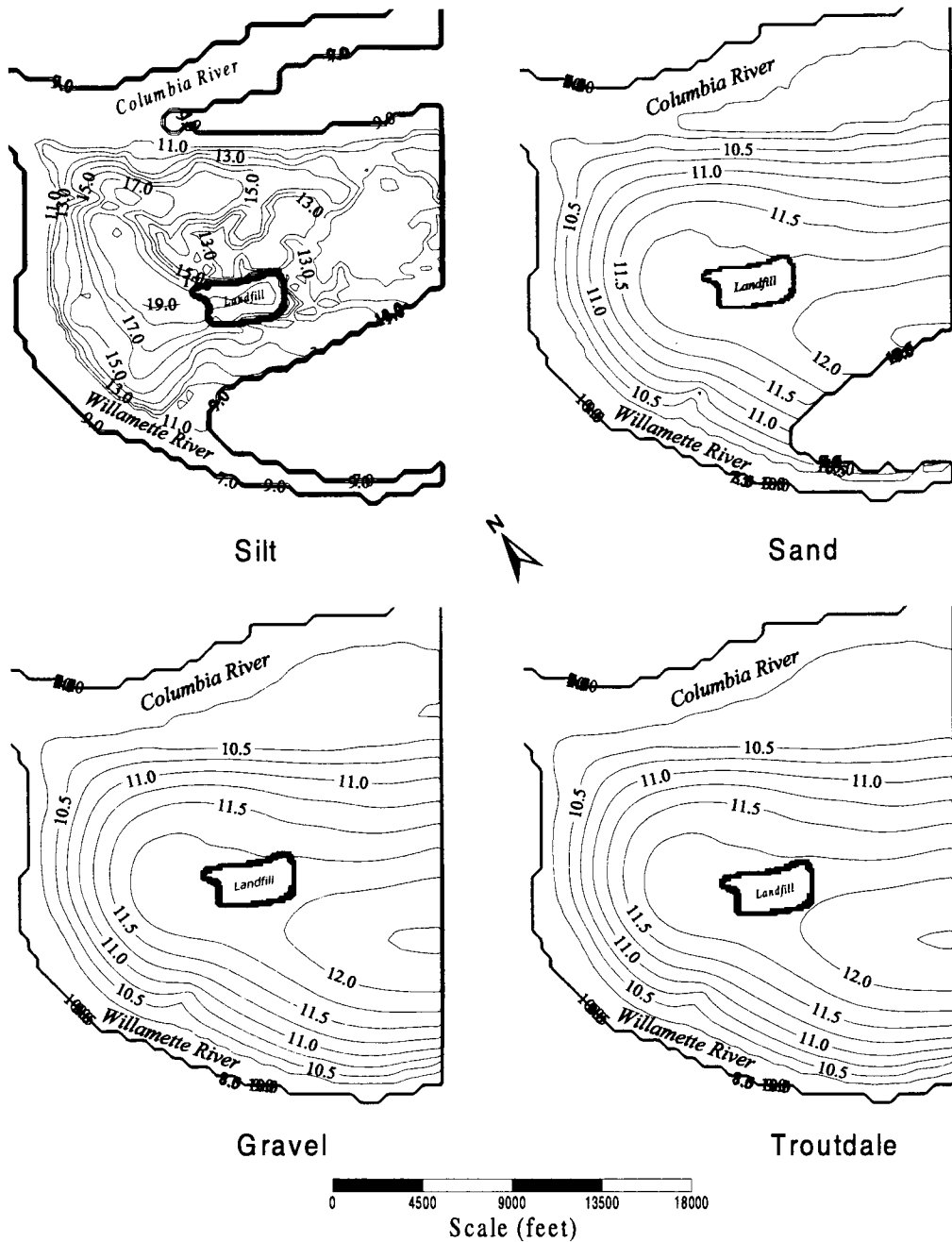
**Figure 46 - Steady state head contours as used in the three dimensional flow model for each computational layer for the time prior to 1982. With the exception of the silt layer, the head contours generally follow the shape of the boundaries outlined by the Columbia and Willamette Rivers. The silt layer is strongly influenced by surface effects such as Smith and Bybee Lakes, the landfill, the Columbia Slough, and recharge variability. The 'cut-out' areas in the silt and sand plots are where the layers are dry.**

116

Seasonal Variability:

Using the transient flow model, 'snap shots' were taken of the flow pattern every 20 days throughout the year based on the calibrated values and data spanning November 1, 1988 to October 31, 1989. Actual simulation is carried out at a time step of ten days. As can be seen in Figure 47, the flow pattern is highly variable from season to season. During the dry season, flow is characterized by a groundwater divide to the south of the landfill area. To the north of the divide, flow is generally from the southeast to the northwest. To the south of the divide, flow is generally from the northeast to the southwest. During the wet season, the regional flow pattern is highly variable. Direction and magnitude is dependent on recharge as well as water levels in the Columbia and Willamette rivers.

The complexity of the regional flow pattern has a large impact on contaminant transport in the gravel aquifer. Frequent direction changes and velocity variations have the effect of dispersing any contaminant plume that may have penetrated the protective silt.
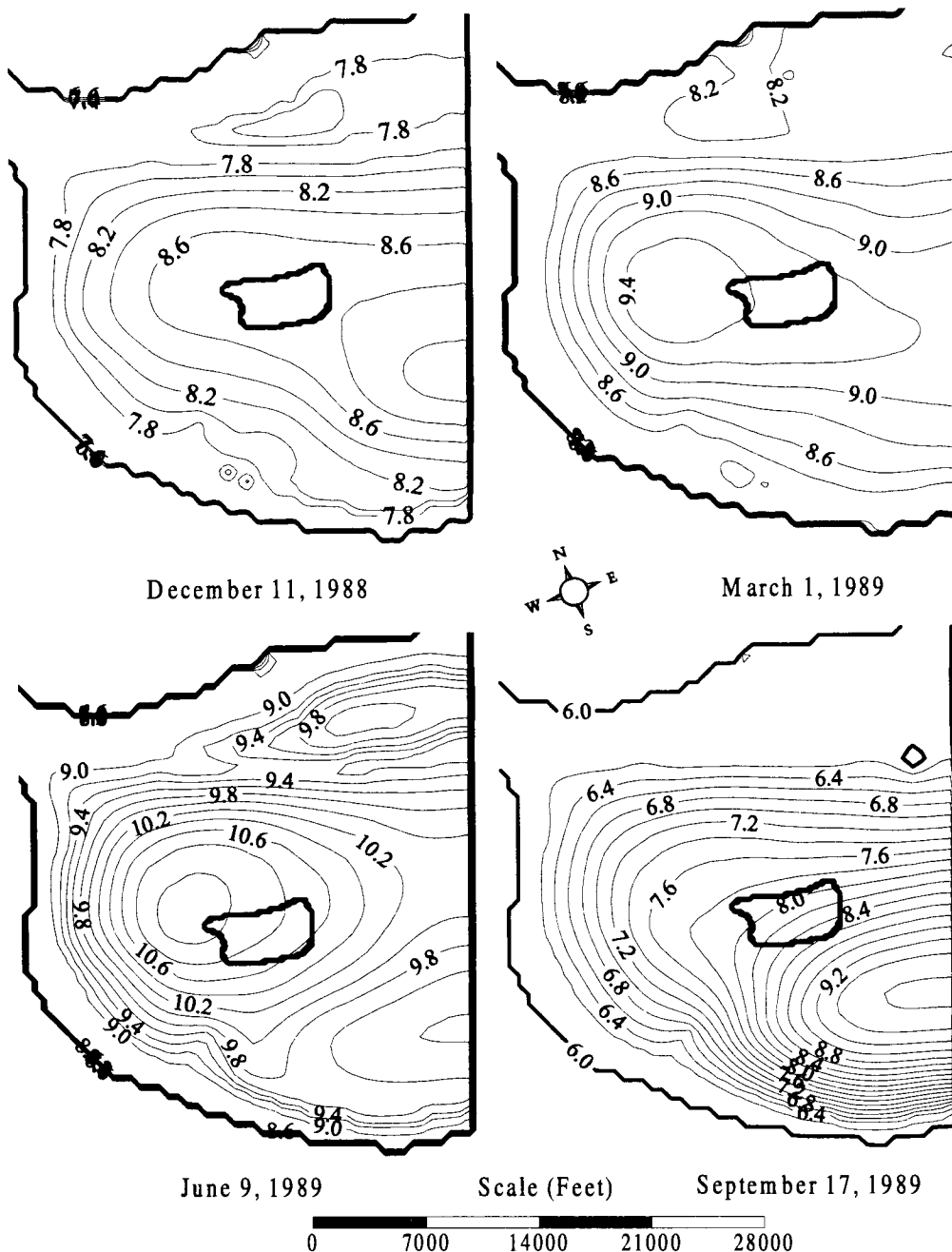
**Figure 47 - Selected head contours for the regional area, in the gravel aquifer, in winter, spring, summer, and fall. Highest head levels occur in the spring, when the Willamette and Columbia rivers are at their peak due to snow runoff from the mountains. Notice that with the exception of the fall period (September), the gradient across the landfill, which is located directly in the center of each figure, is slight, on the order of 0.5 ft per mile. The outermost contour is the outer shore of the Willamette and Columbia Rivers. The outline of the landfill is shown approximately in the center of each contour plot.**

118

Water Balance:

For the case where Smith and Bybee Lakes are isolated from the North Slough, the lakes recharge the gravel aquifer. Recharge via deep percolation into the gravel aquifer in the areas outside the lakes and the landfill represent about 97% of the total recharge into the system (influx) which is approximately balanced by the system outflow of 99.2% into the Columbia and Willamette Rivers. The landfill and the lakes supply about 1% each total recharge into the deep aquifer. Upwelling from the undifferentiated sediments supplies about 4% of the total recharge into the system.

For the case with Smith and Bybee Lakes open to free flow from the sloughs, the lakes are a discharge area for the gravel aquifer. The landfill represents 1% of total recharge into the aquifer, with the balance being supplied by non-landfill and non-lake areas. Outflow into the Columbia and Willamette Rivers drops to about 98% with 1% outflow into the lakes. In all cases, the Columbia and Willamette Rivers are the main outlet for the gravel aquifer receiving the vast majority of recharge into the system.

## 5.6 Sensitivity Analysis:

Sensitivity analysis is performed on storage coefficient, specific yield, conductivity of the silt, conductivity of the sand, conductivity of the gravels, the river sediment conductance, spatial variability in regional recharge, water levels in Smith and Bybee Lakes, and boundary conditions. Each parameter is changed from its calibrated value within a reasonable range while the other parameters are held constant in order to determine the models sensitivity to that particular parameter.

Sensitivity to Storage Coefficient:

As the value for storage coefficient (storativity × unit thickness) is increased, the predicted head in the wells drop in magnitude and react slower to changes in river level. Physically this is correct because as the ability for an aquifer to hold water increases, its sensitivity to outside stresses is reduced. Conversely, as storage coefficient is decreased, we see the magnitude increase and the reaction time decrease in the predicted values.

Sensitivity to Conductivity in the Silt:

The surface water features, the spatial distribution of recharge, and the vertical conductivity in the silt impact the head values in the silt layer. Consequently, the conductivity values in the silt are extremely important for determining the underlying aquifer and surface water interaction. The conductivity in the silt is broken into two different areas, one in the local landfill area and the other outside the local landfill area. This is to better reflect local data in the landfill region and to account for compaction of the silt layer due to the landfill itself.

The silt conductivity values in the immediate vicinity of the landfill used as base case for the regional flow model are obtained from the water balance model calibration. Changes in the silt conductivity have only minor impacts on the heads in the sand, gravel, and Troutdale aquifers along the local landfill model boundaries. Impact in the silt layer is large but overall the conductivity value is conservative. Calibrated vertical conductivity values for the base case use larger than the observed

values at seven out of nine sample locations and more than an order of magnitude higher than the harmonic mean of the observed values.

Sensitivity to Conductivity in the Sand:

The model, in the area of the landfill, is not sensitive to the hydraulic conductivity in the sand. As the value for the conductivity in the sand is increased to equal that in the gravel, very little change is noted. This is because changes in sand parameter values are mostly attenuated in the local landfill area (where calibration occurs) where little or no sand is present. Low sensitivity of the model to conductivity in the sand makes calibration of this parameter impossible. Because the sand is very thick near the Columbia River, changes in sand conductivity greatly change the heads in that region as well as the regional flow patterns. There is, however, no data in the sands from the Columbia River area to calibrate the model with. To uniquely calibrate the sand conductivity, monitoring wells need to be drilled in the sands in the vicinity of the Columbia River.

Sensitivity to River Sediment Conductance:

The base case for the river sediment conductance is 50,000 $ft^2$/day (18,300,000 $cm^2$/day). This seems to be a break point in the models sensitivity to this parameter because as the value is increased (to 5,000,000 $ft^2$/day for example) there is little reaction by the model, but decreasing the value to 500 $ft^2$/day exhibits large changes in the model results, especially in times when the aquifer is discharging to the rivers.

Sensitivity to Conductivity in the Gravel:

The models most sensitive parameter in regards to regional flow pattern is the hydraulic conductivity in the gravel. This is no surprise due to the high permeability of the gravels as well as its continuous nature throughout the model domain. For the region around the landfill however, the gravels are not sensitive to changes in conductivity. The gradient across the landfill stays fairly constant and very small at about 0.0001 ft/ft.

Sensitivity to Recharge:

Changing recharge to a uniform distribution in the model domain increases the recharge in the southeast region and decreases it in the northwest region. The resulting mound in the regional flow pattern is higher and more pronounced with steeper gradients to the north and south. The increased recharge to the southeast combined with the higher topography, shift the peak of the mound to the southeast as well. This variability due to changes in recharge distribution demonstrates the importance of this parameter on regional flow. The base case values with higher recharge to the northwest and the lower recharge to the southeast help keep the gradient flat underneath the landfill. Changes in the regional flow underneath the landfill may have large effects on offsite migration and it is for this reason that any future studies should include a refinement of the distribution of recharge.

Sensitivity to Lake Levels:

The model is not effected by changes in the Smith and Bybee lakes management scheme. With the lakes allowed to fluctuate with the sloughs at the same

level as if the dam on the North Slough was not in place, we see a slight lowering (~0.2 ft) in the magnitude of the head underneath the landfill but little change in the gradient across the landfill. This insensitivity is due to the low permeability silts which separates the gravel aquifer from the surface waters.

Sensitivity to Boundary Conditions:

As mentioned earlier, the North Portland Road boundary condition to the east is treated as a no flow boundary. Additional studies should be conducted to evaluate the effect of pumping wells along the North Portland Well boundary. For the bottom boundary, upwelling from the undifferentiated sediments was tested to find its effect on the regional flow pattern. Because the magnitude of the upwelling is slight compared to other inputs into the model (upwelling represents less than 4% of total recharge into the system as predicted by the USGS Portland Basin model), changes in the bottom boundary condition produce almost no change in the regional flow pattern.

In all cases, the area around the landfill is insensitive to boundary condition changes in the regional model.

# 6.    Two-dimensional Leachate-mounding Model

Data show that the contaminant mound within the landfill is higher in the middle and lower along the edges.  This variability in the mounding cannot be modeled using the zero dimensional water balance model.  The two-dimensional leachate-mounding model (Figure 48) predicts the spatial variability across the landfill of the contaminant mound for later use in the three-dimensional flow model.  This section looks at all aspects of the leachate-mounding model including the conceptual and mathematical models, important parameters to the model, results, and sensitivity analysis.



**Figure 48 - Model domain of two-dimensional leachate-mounding model.  The white dotted line defines the model boundary.**

## 6.1    Conceptual Model:

The conceptual model for the two-dimensional site specific mounding model is very similar to that of the water balance model (Chapter 3). The landfill mound is fed by recharge, which can escape either laterally through the dike into the sloughs, vertically through the silts into the gravels, or through the surface cover of the landfill as surface seepage. The landfill has the ability to store and transmit water just like any ground medium. The landfills hydraulic conductivity will determine the steepness of the surface contours of the contaminant mound across the landfill area. Unlike the water balance model, which assumes an infinite hydraulic conductivity in the landfill (and thus a flat contaminant mound surface), the two-dimensional site specific mounding model accounts for restricted flow through the landfill. The perimeter of the landfill can drain contaminant laterally through the dikes and vertically through the silts while the only drain in the interior of the landfill is downward vertically through the silts. Because of this it is expected that the mounding along the edges will be less than in the middle of the landfill.

The landfill is bordered on all sides by either natural overbank deposits (the dike) or a man-made engineered dike which borders about a quarter of the landfill from the southeast corner, along the east side, and about a quarter of the north side of the landfill. The engineered dike is considered very low permeability and may also effect the shape of the contaminant mound in the landfill.
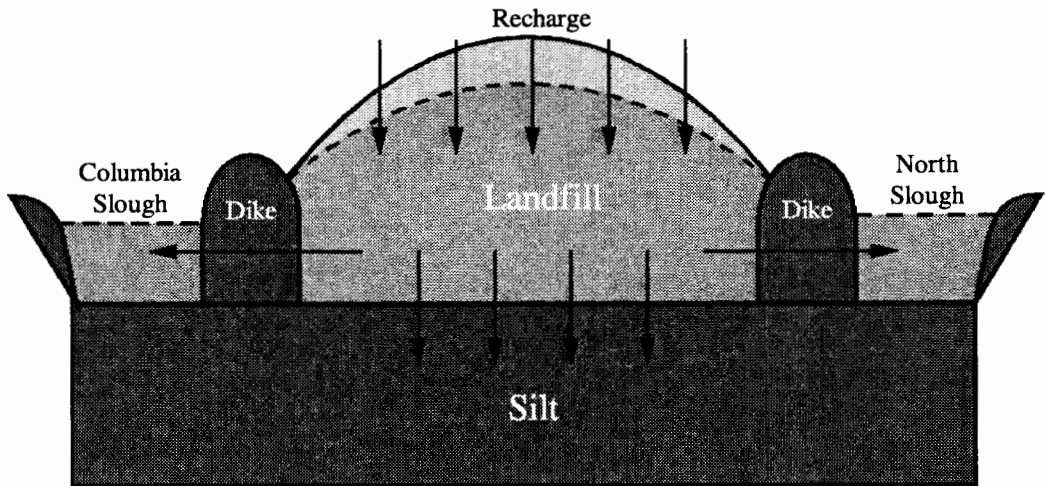
**Figure 49 - Conceptual schematic of two-dimensional site specific mounding model. As recharge enters the top of the landfill, flow can move either laterally out to the sloughs or vertically down through the silt layer towards the gravel aquifer. The sum of the fluxes out of the landfill equals the recharge into the landfill.**

### *6.2    Mathematical Model:*

The two-dimensional site specific mounding model employs MODFLOW

(McDonald and Harbaugh, 1989) and is run on Sparc workstations using the UNIX

platform.

Governing Equations:

The governing equation for two-dimensional groundwater flow is:

$$S_y \frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x} K_x \frac{\partial \phi}{\partial x} + \frac{\partial}{\partial y} K_y \frac{\partial \phi}{\partial y} + S/S$$

$\phi = f(x, y, t)$

(13)

126

where:

$S_y$ = Specific Yield.

$\phi$ = Head in aquifer.

$K_i$ = Directional component of the hydraulic conductivity.

$S/S$ = Sources or sinks.

For the two-dimensional leachate-mounding model, the only sources or sinks

are the seepage to and from the sloughs, surface seepage through the landfill cover,

and recharge.

Recharge is added directly to the landfill and is represented as a rate (L/t) value

by q such that:

$$q_{recharge} = q(t)$$

**(14)**

Seepage flux from the landfill to the sloughs is based on Darcy's law using the

horizontal conductivity of the dike. This is represented mathematically by:

$$q_s = \frac{K'}{b'}(\phi - \phi_s)$$

**(15)**

where:

$q_s$ = Darcy velocity between landfill and slough.

$K'$ = Horizontal hydraulic conductivity of the dike.

$b'$ = Thickness of the dike.

$\phi_s$ = Water level of slough.

$\phi$ = Contaminant mound height at the dike/landfill interface.

Surface seepage is similar to Equation (15) but the elevation of the landfill at

the point of seepage is substituted for the water level in the slough. In addition,

because of sanitary landfill management practices, a semi-permeable daily cover,

usually fill dirt, is used to cover the each days added refuse. Thus there is a restrictive

element to the surface seepage that is represented by the hydraulic conductivity of the

daily cover. The equation for surface seepage as modeled here is:

$$q_{surf} = \frac{K_c{'}}{b_c{'}}(\phi - E)$$

where:

| | |
|---|---|
| $q_{surf}$ | = Darcy velocity from surface seepage from landfill. |
| $K_c{'}$ | = Hydraulic conductivity of daily cover. |
| $b_c{'}$ | = Thickness of daily cover. |
| $\phi$ | = Contaminant mound height at seepage point. |
| E | = Elevation of seepage point. |

If $\phi$ is less than E, $q_{surf}$ is given as zero. The only time surface seepage is non zero is

when the contaminant mound in the landfill is greater than the elevation of the seepage

point.


## 6.3    *Boundary Conditions and Discretization:*

The two-dimensional leachate-mounding model is a refinement of the water

balance model in that it splits the landfill into a uniform grid where each cell has

lateral flux to a neighboring cell (or through the dike and into the slough if the cell is

on the perimeter of the landfill), vertical flux to the silt, and surface seepage. Thus,

each cell has the potential of flow out of all 6 sides, laterally through four sides,

vertically downward through the bottom, and vertically upward through the top.

The model domain is the landfill itself discretized into 153 (east/west) by 117 (north/south) uniform grid blocks with a grid spacing of 50 feet. One layer in the vertical direction is modeled. Each cell in the grid is of different height depending upon the thickness of the refuse at that point. The expansion of the landfill over time is taken into account by activating cells at the appropriate times and locations within the simulation based on data supplied by METRO. The model simulates the annual mean condition with a time step of one year.

The landfill is treated as an unconfined unit with all cells in the landfill modeled as MODFLOW river cells. Each cell is assigned an effective conductance depending upon its position within the landfill. If a cell is located in the interior of the landfill, then contaminant migration can occur either downward through the silt or laterally into the neighboring cell. The flow downward into the silt is determined by the head in the gravel (given as a boundary condition), the thickness of the silt below the cell, the vertical hydraulic conductivity of the silt ($K_{siltV}$), and the height of the contaminant mound in the cell (predicted by the model). The lateral flow into the neighboring cells is determined by the head gradient between the cells and the hydraulic conductivity of the refuse. For cells along the edge of the landfill the flow into the sloughs is determined by the horizontal hydraulic conductivity of the silt ($K_{siltH}$), the height of the contaminant mound in the cell, the thickness of the dike, and the water level in the slough. Recharge is added directly to each cell.

Each cell is assigned an effective conductance depending on its location in the landfill. The horizontal effective conductance that accounts for the faces of a cell that borders the slough is:

$$C_k = \sum_{n=1}^{N} \frac{\Delta i K'}{2b'} (\phi_l + \phi_s)$$

where:
$C_k$ = Effective horizontal conductance
$N$ = Number of cell sides bordering slough
$\Delta i$ = Appropriate cell dimension (either $\Delta x$ or $\Delta y$)
$K'$ = Horizontal conductivity of silt ($K_{siltH}$)
$b'$ = Dike thickness
$\phi_l$ = Contaminant mound height at dike
$\phi_s$ = Water level in slough

The contaminant mound and the water level in the sloughs are needed to calculate the area available for contaminant migration through the dike. If the mound is higher than or equal to the height of the dike, than the area available for contaminant migration is the dike height (assumed at 25 feet) times the width of the cell. If the contaminant mound is less than the height of the dike the area is approximated as the average between the contaminant mound and the water level in the slough times the cell width. The mean contaminant mound as determined by the water balance model is used as the contaminant mound height for calculating the effective conductance for each cell. The vertical effective conductance is the vertical hydraulic conductivity of the silt times the footprint area of the cell, divided by the silt thickness under the cell. Mathematically the vertical effective conductance is represented by:

130

$$C_v = \frac{AK_v}{b}$$

<div align="right">(18)</div>

where:

$C_v$ = Effective vertical conductance for cell.

A = Cell area ($\Delta x \times \Delta y$)

K = Vertical hydraulic conductivity of the silt ($K_{silt}v$)

b = Silt thickness

The total conductance for each cell is the effective horizontal conductance plus the effective vertical conductance. If a cell is bordered on more than one side by the sloughs, its total conductance is higher than if it is bordered on one or no sides by the sloughs. All cells will have some vertical component based on the thickness of the silt below that cell while only the cells along the edges will have any horizontal component.

The horizontal flow out of the perimeter cells is the horizontal effective conductance times the difference between the water level in the slough and the leachate mound height in the perimeter cell. The vertical flow out of all cells is the vertical effective conductance times the difference between the head in the gravel aquifer (as determined by the regional flow model) and the leachate mound height in the cell. Slough water levels and the head distribution in the gravel aquifer are model inputs.

The simulation runs from 1950 to 2020, with output being the spatially distributed, yearly averaged mound heights for each cell for each year simulated. The 70 year simulation is broken into 48 separate stress periods, with the first 47 stress periods consisting of one time step of one year each. The 48th stress period consists of

24 time steps of one year each. Modifications are made to the original MODFLOW code to allow for the expansion of the landfill area over time. The source code for each preprocessor, the changes made to the MODFLOW code, and a flow chart showing the structure for entire two-dimensional leachate-mounding model are detailed in 9.3.2Appendix B.

## *6.4    Parameter Estimation and Calibration:*

The only new parameter requiring calibration in the two-dimensional leachate-mounding model is the hydraulic conductivity in the refuse ($K_{siltL}$). Stratigraphy and recharge are parameterized as described in section 5.4 of the regional flow model description. All other parameters are passed from the water balance model and assumed to apply directly to the two-dimensional site specific mounding model.

The hydraulic conductivity of the landfill determines the amount of head loss across the landfill and thus controls the spatial shape of the contaminant mound. The model is calibrated to the time-averaged maximum head difference across the landfill (Sweet-Edwards/Emcon, 1989). As in most landfills, the refuse at the St. Johns Landfill is extremely heterogeneous. Hydraulic conductivity inferred from local refuse composition can be totally unrepresentative of the overall field scale hydraulic properties. Typical conductivities in landfills have been found to be anywhere from $10^{-2}$ to $10^{-4}$ cm/sec (Oweis et al., 1990). Calibration of the two-dimensional site specific contaminant mounding model leads to an effective field-scale conductivity of $5.0 \times 10^{-4}$ cm/sec.

Unlike the water balance model, the two-dimensional model is able to distinguish between dike material and the underlying silt material. While both formations are made from the same overbank deposits, an assumption is made that the vertical conductivity in the dike is twice that of the underlying silt due to compaction under the landfill from the weight of the refuse and the contaminant mound. Since flow is primarily horizontal through the dikes, the vertical conductivity in the dike is very insignificant to the models results.

## 6.5  *Results:*

Like the water balance model, the two-dimensional site specific mounding model predicts the rising period, quasi steady state period, and the dissipation period for the landfill mound. The time for each period agrees with the predictions of the water balance model although there are some important differences (Figure 50). The two-dimensional site specific mounding model predicts the growth period to last from 1950 to about 1965. This is a little longer than the water balance model due to the higher mounding in the center of the landfill. The center of the landfill mound reaches a maximum height of about 34 feet by 1970. The shape and height of the mound maintains very consistently until the landfill expansion begins in 1989. Due to the added area, the mound spreads itself out and fills the new area. When closure begins in 1992 the landfill mound begins to dissipate. As the landfill dissipates, a slight east to west gradient is developed due to the impervious engineered dike that borders the eastern side of the landfill and the inability for water to drain laterally from that side.

By the year 2010, the mounding has dissipated and equilibrium with the surrounding slough levels and the underlying gravel aquifer is reached.

No attempt is made to distinguish the distribution of the outflows. Data on the spatially varying composition of the natural dike do not exist in enough detail to adequately model this. Without this data, it is impossible to match observed dike seepage points to model results.
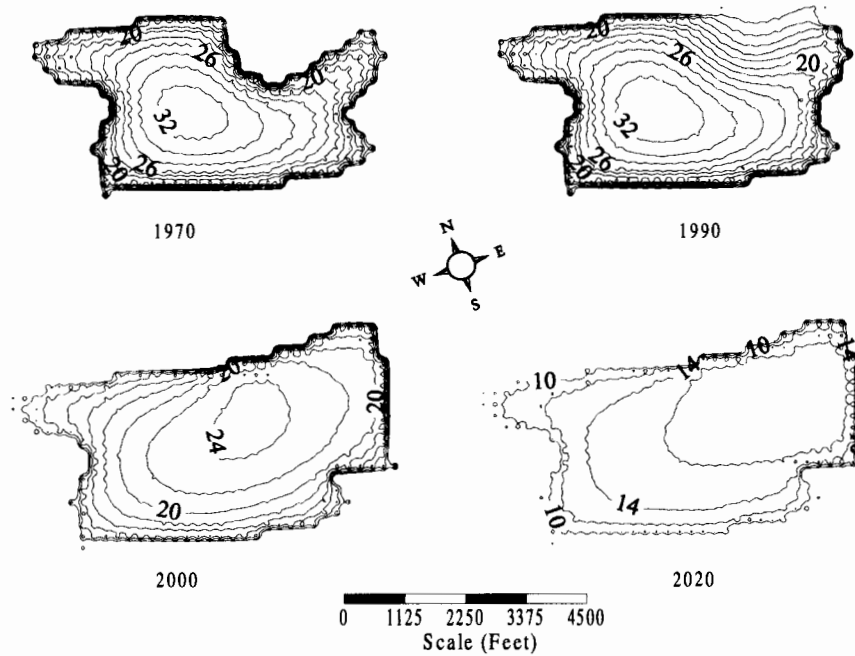
Figure 50 - Spatial distribution of contaminant mound height at selected times for the base case. By 1970 the mound has reached steady state and remains virtually unchanged until the new expansion area is added in 1989 and 1990. The mound has a slight east to west slope in the 2010 plot due to the impenetrable engineered dike which surrounds the landfill on the east and northeast side.

### 6.6 Sensitivity Analysis:

The two-dimensional site specific mounding model is simply an extension of the water balance model so large scale sensitivity analysis is not performed with respect to most parameters although all parameters are tested to insure the model responds correctly to the stress changes. Since $K_{siltL}$ is not used in the water balance model, sensitivity analysis is performed to determine the models reactions to changing values in the landfill conductivity. As expected, as $K_{siltL}$ is increased from the base case of $5.0 \times 10^{-4}$ cm/sec (increasing the conductivity within the landfill) the shape of the contaminant mound becomes flatter and the maximum height is less. When $K_{siltL}$ is lowered to $5.0 \times 10^{-6}$ cm/sec, the contours of the contaminant mound become more pronounced and the maximum height increases. This is shown in Figure 51.

While the shape of the mounding changes with different values of $K_{siltL}$, the changes in outward flux and the impact on the results of the three-dimensional flow model and contaminant transport model are minimal, as will be discussed later.
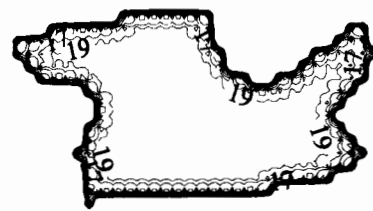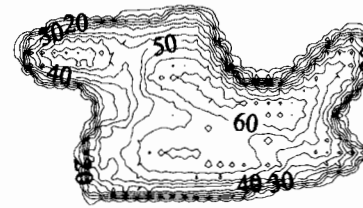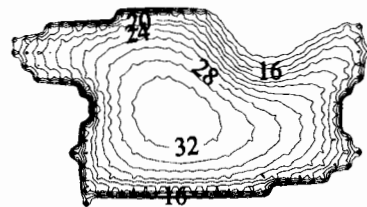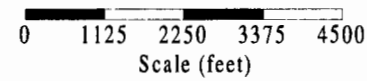
Figure 51 - Three contours showing $K_{siltL}$ with values of $5.0 \times 10^{-2}$ cm/sec, $5.0 \times 10^{-4}$ cm/sec, and $5.0 \times 10^{-6}$ cm/sec. Notice that as the conductivity decreases mounding and spatial variability increase. The case showing $K_{siltH} = 5.0 \times 10^{-4}$ cm/sec, is the base case result (Figure 50).

# 7.    Three-dimensional Flow Model

A key concern for the St. Johns Landfill is the possibility of offsite

contaminant migration.  The intent of the three-dimensional flow model is to produce

a vertically detailed, three-dimensional flow field in the silt, sand, gravel, and

Troutdale aquifer and to establish the refined flow field around the local landfill area.

Capturing the three-dimensional details within each geologic unit allows for better

prediction of plume mixing and migration.  This section will look specifically at the

three-dimensional flow model, its conceptual and mathematical models, results and

sensitivity analysis.

## *7.1    Conceptual Model:*

The groundwater flow patterns in the immediate region of the landfill exhibit

complex spatial and temporal dynamics due to differing hydrologic stresses (tidal,

seasonal water level fluctuations, and the landfill mound) and heterogeneous geologic

conditions on both macro and micro scales.  Within the area directly below the landfill

and adjacent to the sloughs, the flow is dominated by the contaminant mounding and

dissipation, and the interaction between the landfill mound and the slough (see Figure

52).  Since the contrast between the silt conductivity and the gravel conductivity is

large, flow in the silt can be assumed vertical, except in the area directly adjacent to

the sloughs where a contaminant mound-induced horizontal gradient exists.
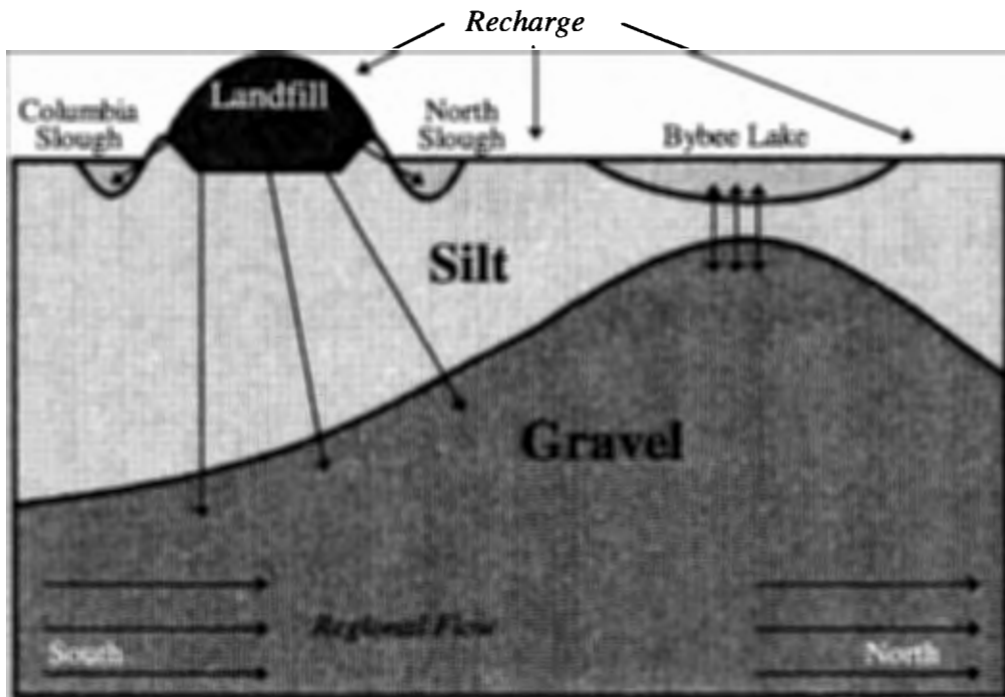
**Figure 52 - Three-dimensional flow model conceptual layout. Recharge enters the landfill and the silt. The sand layer is very thin in the immediate region of the landfill and is not included in the conceptual model. Regional flow from the gravel aquifer interaction with the Willamette and Columbia Rivers produces a yearly mean, steady, offsite migration in the gravels. Due to the high contrast in conductivity's between the silt and gravel, flow through the silts is mainly vertical into the gravels except in the area around the sloughs where horizontal seepage into the sloughs is prevalent. Concern exists for contaminant seepage moving vertically from the landfill into the silts, northward with the regional flow, and then discharging into Bybee Lake at the area where the silts are very thin.**

The landfill sits directly atop a westward-dipping trough in the gravel aquifer which is overlain by the silt. To the south, north, and east, the gravel slopes upward with the steepest slope to the north where the gravel comes within several feet of the bottom sediments of Bybee Lake (Figure 7). Concern exists of the potential of contaminant migration moving downward through the silts into the gravel, northward in the gravel with the regional flow and then upward into Bybee Lake. Flow in both

the gravel aquifer and the Troutdale aquifer is predominantly horizontal in the region

of the landfill as predicted by the regional flow model and the USGS Portland Basin

model. The horizontal flow in the aquifers could be a mechanism for offsite plume

migration once contamination penetrates the silt layer.

### 7.2    *Mathematical Model:*

The local scale three-dimensional flow model is used to reproduce the three-

dimensional flow pattern within the critical study area surrounding the landfill. The

model utilizes the MODFLOW groundwater code (McDonald and Harbaugh, 1989)

for its simulation, compiled and run on a UNIX workstation.

Governing Equations:

The three-dimensional flow model is a fully three-dimensional, temporally

varying model meaning it is designed to produce a three-dimensional flow field in

both time and space. The governing equation for groundwater flow in three

dimensions is:

$$S_s \frac{\partial \phi}{\partial t} - R = \frac{\partial}{\partial x} K_x \frac{\partial \phi}{\partial x} + \frac{\partial}{\partial y} K_y \frac{\partial \phi}{\partial y} + \frac{\partial}{\partial z} K_z \frac{\partial \phi}{\partial z}$$

$\phi = f(x, y, z, t)$

(19)

where:

| | |
|---|---|
| $\phi$ | = Head level. |
| t | = Time |
| $S_s$ | = Specific storage. |
| $K_x, K_y, K_z$ | = Components of hydraulic conductivity in x, y, and z directions. |
| R | = General source/sink term. |

R defines the volume of inflow to the system per unit volume of aquifer per unit time, and is always considered positive. For outflow terms, R is negative. Each one of the source/sink terms is handled identically to the two-dimensional leachate-mounding model (Chapter 6).

## 7.3 Boundary Conditions and Discretization:

The domain of the three-dimensional local flow model is such that it is able to capture the local flow characteristics with in the probable area of plume migration (Figure 53). The model consists of a 7650 foot (east/west direction) by 5850 foot (north/south direction) rectangular area around the landfill encompassing the area of potential contamination for the complete simulation time span. Vertically, the model includes the landfill, the silt layer, the sand layer, the gravel aquifer, and the Troutdale aquifer.

The rectangular model domain is discretized into a 51 by 39, 150 foot, uniform, computational grid (Figure 54). Vertically, the model is discretized into 39 layers from elevation 95 feet down to -350 feet (City of Portland Datum) with a vertical grid spacing of 5 feet each from elevation 95 to -30 feet, 10 feet each from elevation -30 to -50 feet, and 25 feet each from elevation -50 feet to -350 feet. The high resolution 5 foot spacing encompasses most of the silt layer where flow is predominantly vertical and comparatively slow, the 10 foot spacing is in the transition sand zone, and the 25 foot spacing is in the gravel aquifer and Troutdale Aquifer where flow is primarily horizontal. The grid is designed to allow sufficient detail of the contaminant plume, and to obey the numerical criteria required to maintain model

accuracy and minimize numerical diffusion. The model simulates annual mean flow

dynamics in three-dimensional space from 1950 to 2020 using a time step of one year.
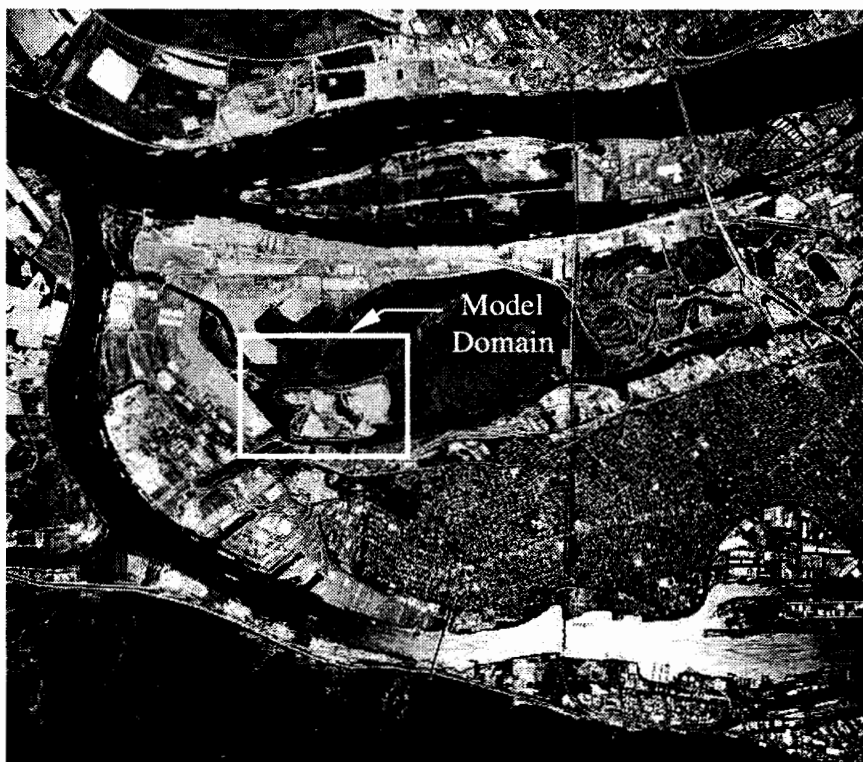


**Figure 53 - Large regional view of landfill area showing the three-dimensional flow model, modeling domain. Boundary conditions for the sand, gravel, and Troutdale are determined from the results of the regional flow model and allows the local scale three-dimensional model to include large regional stresses in its predictions. The two-dimensional leachate mounding model results are used as the upper boundary condition in the landfill. Flow in the silt is assumed vertical along the lateral boundaires, thus a no flow boundary in the silt is used.**

# Cross Section Locations and
# Horizontal Local Model Discretization



················· North/South Cross Section Location on Column 25

━ ━ ━ ━ ━ ━ East/West Cross Section Location on Row 29
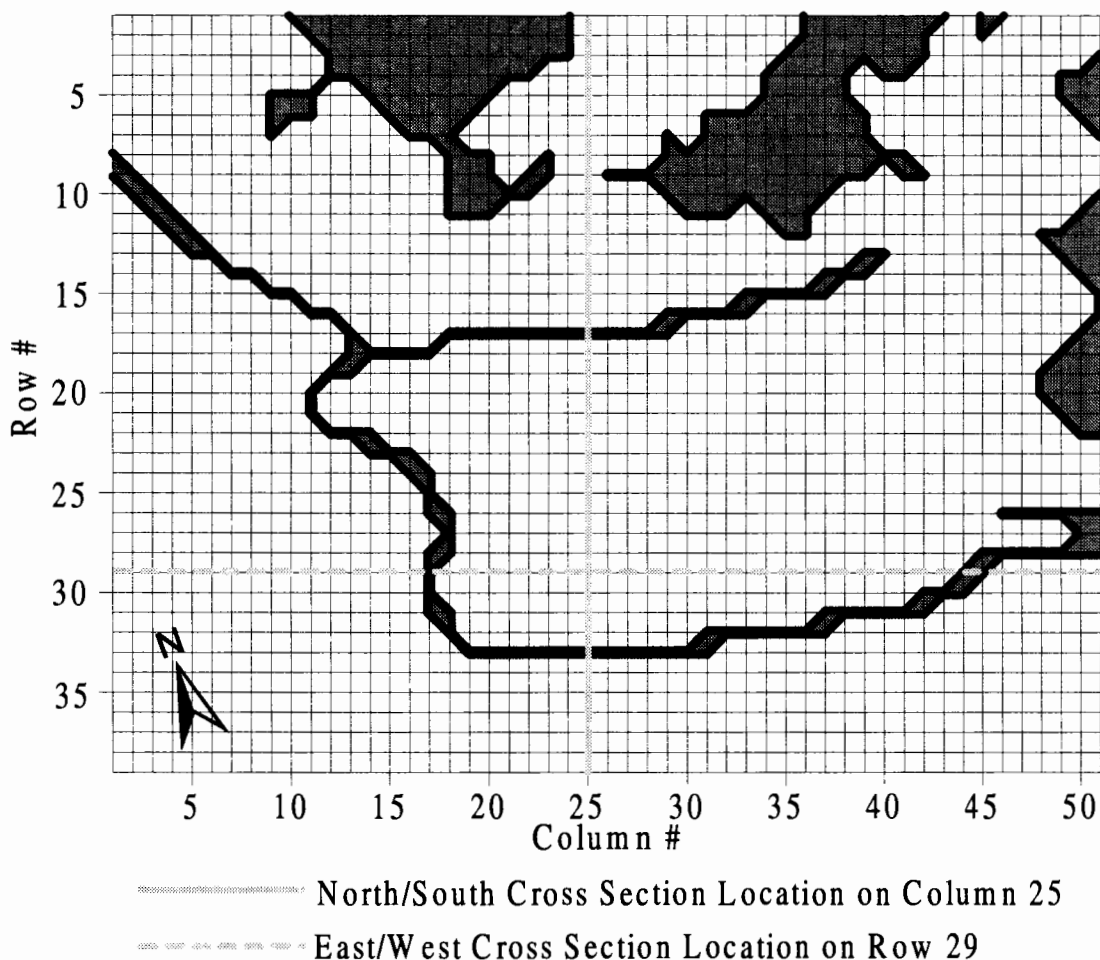
**Figure 54 - Local model discretization showing cross section locations for vertical contour plots (Figure 55, Figure 56, and Figure 57).**

Initial conditions for the sands, gravels, and Troutdale Aquifer are obtained

from the regional flow model results. Values for each cell are linearly interpolated

from the larger regional grid. Initial head values in the silts are 10.0 feet (COP).

The boundary conditions are derived from the regional flow model, the water

balance model, and the two-dimensional leachate-mounding model. The lateral

boundary conditions in the sand, gravel, and Troutdale are determined from the results of the regional flow model. Head values in the coarse grid regional flow model are linearly interpolated to fit the finer grid local model. The regional flow model results are split into three broad time spans and run as steady state for each of the time spans. The first is pre-1982 before the diversion dam on the North Slough is built and Bybee Lake is allowed to fluctuate naturally with seasonal and other temporal changes. The second time span simulates the period from 1982 to 1997 with the dam in place and Bybee Lake held at 3 feet above the surrounding surface waters. The last period is the post 1997 time span which, for the base case, is the same as the pre 1982 results. The lateral boundary conditions for the silt are considered no flow boundaries because of the predominantly vertical flow in the silt layer. Existing data can be interpreted to show lateral flow along the silt boundary however, due to the tightness of the silt and the distance of the boundary to the landfill, the stresses in the plume migration area are not effected by the silt boundary condition. This application of a no flow boundary condition is supported by Anderson and Woesner (1992, p. 102).

The bottom boundary condition is chosen as no flow since flow in the Troutdale Aquifer is mainly horizontal. The USGS Portland Basin model indicates slight upwelling from the underlying undifferentiated sediments to the Troutdale aquifer, however the upwelling is slight (less than 4% total recharge) and thus is not modeled. (Figure 44). Flow dynamics are dominated locally by the landfill mounding and changes on the bottom boundary within realistic values, show no effect on the local flow conditions. At the top, the lakes and sloughs are represented as

144

MODFLOW river boundaries, with heads equal to the slough and lake levels. The mound height in the landfill is determined from the results of the two-dimensional leachate-mounding model and sets the upper boundary in the landfill as a prescribed head boundary. The remaining top boundary that excludes the landfill and surface water is a water table boundary receiving recharge due to precipitation.

Subroutines are added to the MODFLOW code which allows for reading in the yearly results from the two-dimensional model for the upper boundary conditions as described above, the lateral boundary conditions as simulated by the regional flow model, and the changing geology due to the expansion over time in the landfill. As is the case in the two-dimensional leachate mounding model, the growth of the landfill is simulated subarea by subarea, with a particular subarea becoming completely active the year it was first put into use with an initial head of 10 feet. The timing for each stress period as well as the timing of the landfill growth is identical to the two-dimensional leachate-mounding model (section 6.3). Appendix C shows the source code and documentation for the three-dimensional flow model.

## 7.4    *Results and Sensitivity Analysis:*

The driving force for the three-dimensional flow system in the immediate vicinity of the landfill is dominated by the contaminant mound. As the contaminant mound builds, a vertical gradient between the landfill and the gravel aquifer is created. Streamlines along the head contours show flow moving vertically through the silt to the gravel except in the immediate areas directly adjacent to the sloughs where flow is horizontal from the landfill into the sloughs (Li & Lowry, 1997). Below the silt, in the

145

sand, gravel, and Troutdale aquifers, flow is split, moving to the north on the northern side of the landfill and to the south on the southern side of the landfill. The mound-induced driving force from the landfill causes this split. The landfills influence on the regional flow extends about 300 to 500 feet beyond the edge of the landfill. Head contours of the head values along column 25 and row 29 for the base case values are shown in Figure 55.

Sensitivity analysis is performed on hydraulic conductivity, recharge and silt boundary condition, and boundary conditions in the sand, gravel, and Troutdale aquifers.

Sensitivity to Hydraulic Conductivity:

Sensitivity analysis on hydraulic conductivity is performed in the 'upstream' models and is carried through all models in line. For example, if $K_{siltH}$ is changed in the water balance model, the change is made in every model to determine the modeling systems sensitivity to that parameter. For the three-dimensional flow model, sensitivity to hydraulic conductivity is determined by its effect on the contaminant plume migration as predicted by the contaminant transport model. For this reason, a complete description of the three-dimensional flow models sensitivity to hydraulic conductivity changes will be discussed with the contaminant transport model (section 7.4).
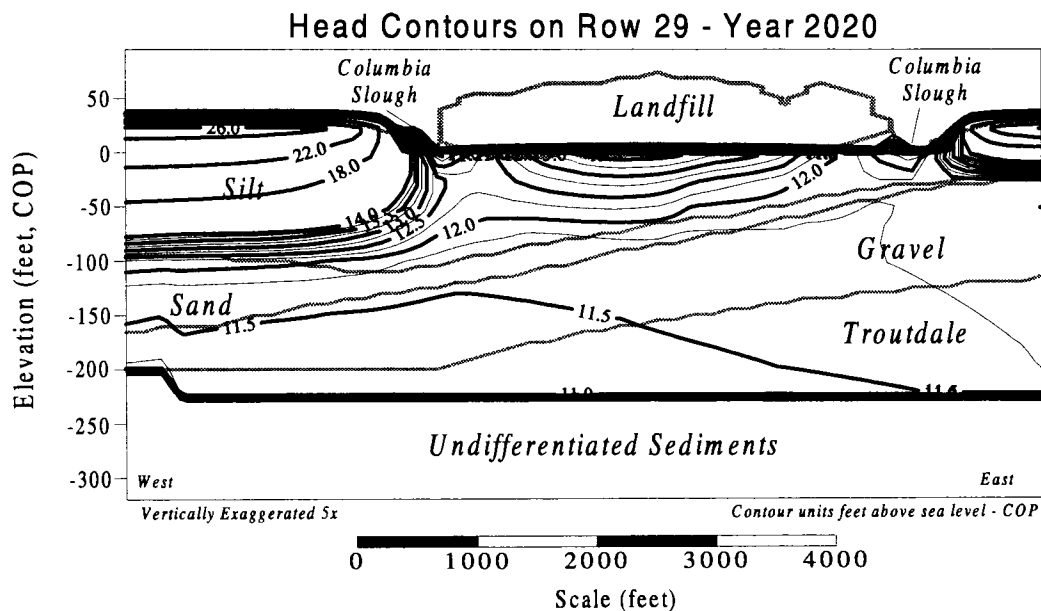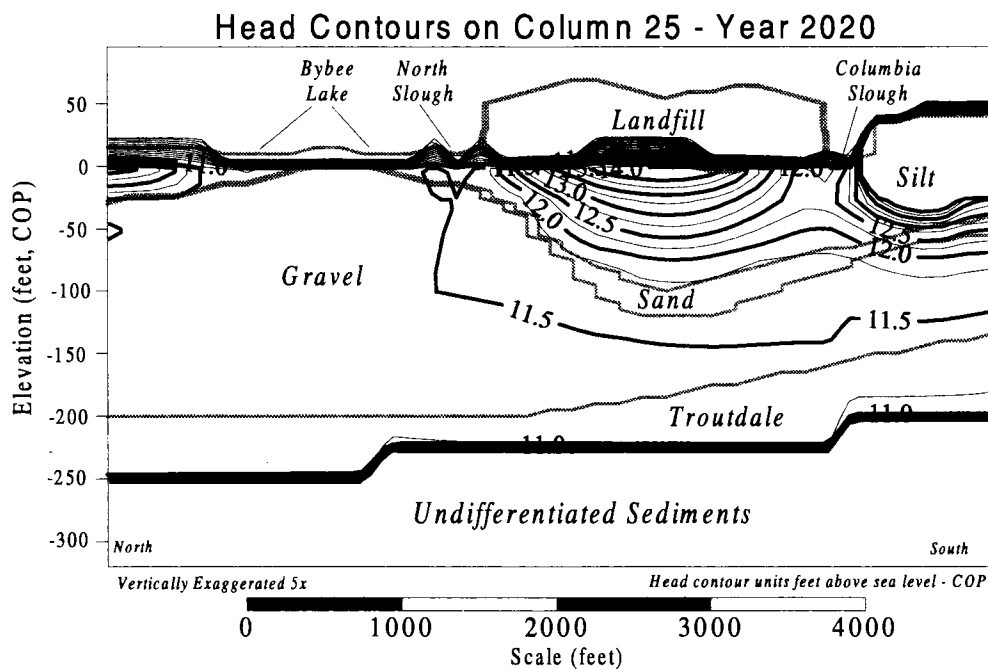
**Figure 55 - Head contours for base case values along column 25 and row 29 (see Figure 54). The landfill mound and influence dominate the local flow dynamics. Generally, flow is vertically downward through the silt except in the immediate vicinity of the sloughs. Flow in the sand, gravel, and Troutdale aquifers follow the regional flow.**

Sensitivity to Recharge and Silt Boundary Condition:

The intent of setting the modeling domain for the three-dimensional flow model is to capture the flow dynamics in the area of potential plume migration, extend the boundaries far enough from the landfill such that the boundary effects cannot be 'felt' in the area of plume migration, and to maintain a computationally efficient model. The no-flow boundary condition in the silt brings about concern of pressure build up along the boundary when recharge is applied to the silt. This pressure buildup could have a large impact on the local plume dynamics.

As an extreme test, recharge to the silt is dropped to zero and the no-flow boundary in the silt is changed to a constant head of 8 feet. This reduces the pressure build up due to recharge as well as releases any pressure build up due to any horizontal flow in the silt through the low constant head boundary.

As can be seen in Figure 56, the effects of the silt boundary are extremely localized. The landfill mound dominates the local flow dynamics and the changes in the silt boundary condition are not 'felt' more than a few cell widths from the boundary. This is due primarily to the low conductivity of the silt. The three-dimensional local flow model is not sensitive to silt boundary conditions.

## Head Contours on Column 25 - Year 2020

## Head Contours on Row 29 - Year 2020
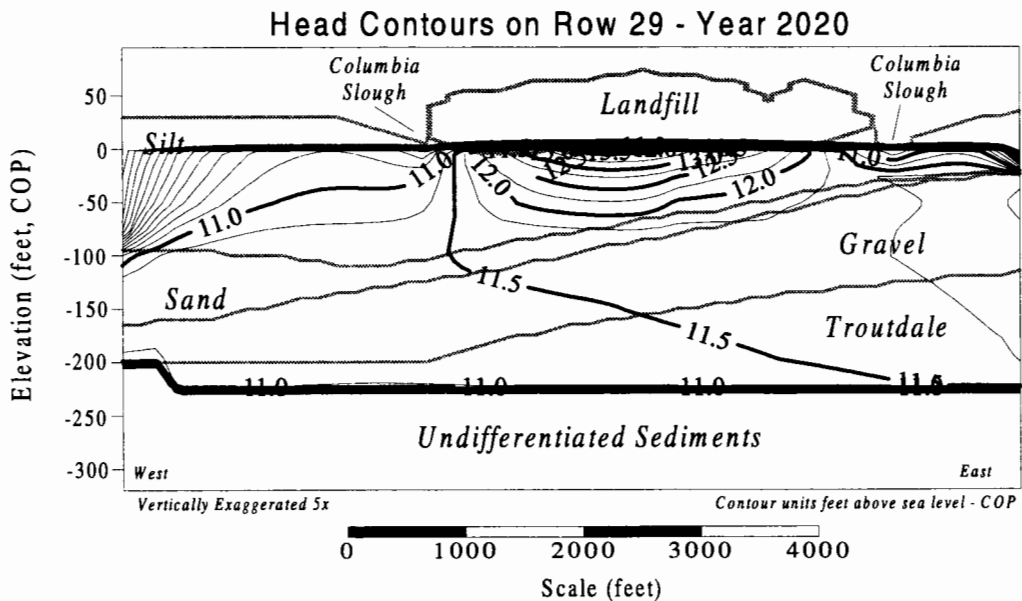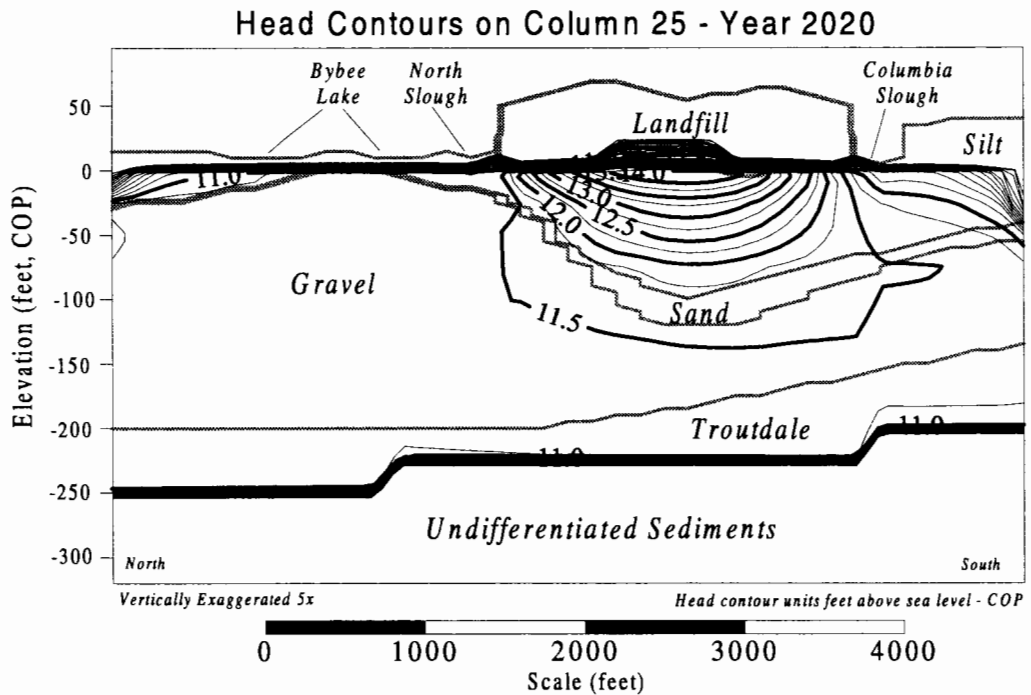
**Figure 56 - Head contours along column 25 and row 29 (see Figure 54) with zero recharge everywhere but the landfill and a constant head boundary of 8 feet in the silt. Due to the low permeability of the silt, the boundary conditions are not 'felt' on the interior of the modeling domain. Comparison of the head contours in the landfill area to that of the base case (Figure 55) shows little or no change.**

Sensitivity to Sand, Gravel, and Troutdale Aquifer Boundary Condition:

The boundary condition along the sand, gravel, and Troutdale aquifer is tested to examine the boundary's influence on the local flow dynamics in the landfill area. Since the flow in the aquifers is mainly horizontal, changes along the boundaries are 'felt' throughout the entire model, although the impact is small. Sensitivity of the boundary effect is tested by inducing an 'artificial' regional flow across the model domain. To do this, the southern boundary of the model is fixed with a head 0.5 feet higher than the fixed head of the northern boundary (11.8 feet south versus 11.3 feet north). The east and west boundaries are maintained with base case values as predicted by the regional flow model. The same condition is then reproduced for a second test in the east/west direction with the northern and southern boundaries set at base case. The first scenario produces a gradient of $8.5 \times 10^{-5}$ ft/ft in the north/south direction (flow from south to north), and the second scenario produces a gradient of $6.5 \times 10^{-5}$ ft/ft in the east/west direction (flow from east to west). As a comparison, the gradient from south to north as predicted by the regional flow model is $1.7 \times 10^{-5}$ ft/ft and the east/west gradient is about $3.1 \times 10^{-5}$ ft/ft.

As can be seen in Figure 57, the changes in flow dynamics are minimal. Comparing to the base case (Figure 55), the groundwater divide beneath the Columbia Slough in the north/south plot can be seen to shift slightly in the direction of flow. For the 'artificial' gradient in the east west direction, the impact is even less. Because of the slight impact for reasonable gradients across the modeling domain, the three-

dimensional flow model is determined to be insensitive to boundary conditions in the

sand, gravel, and Troutdale aquifer.
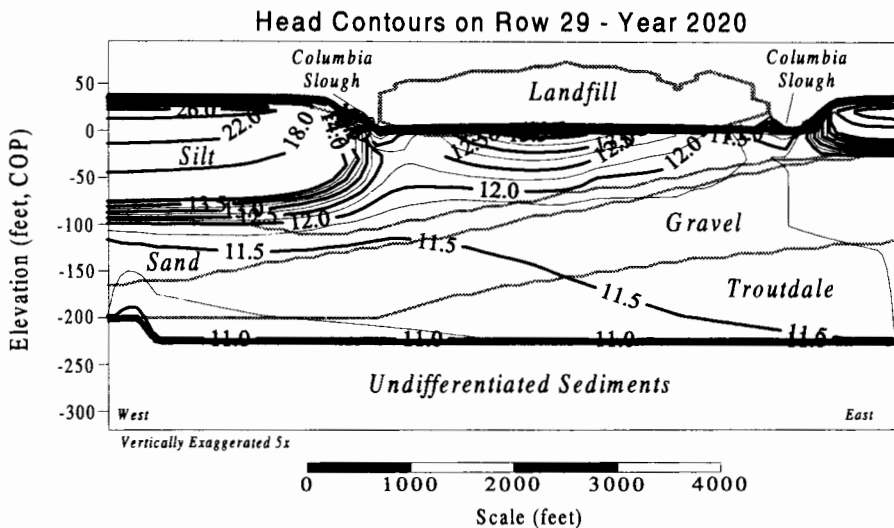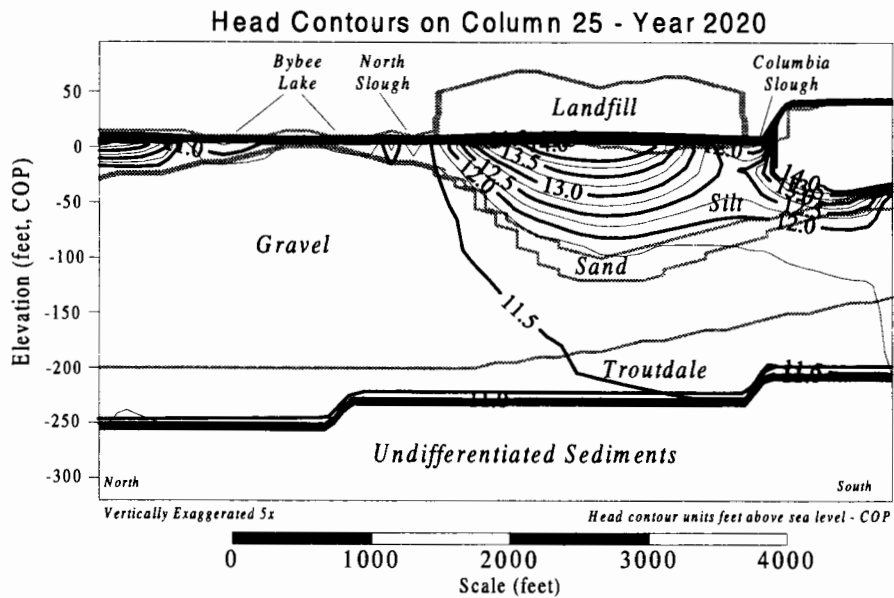
Figure 57 - Head contours along column 25 and row 29 showing the effect of different boundary conditions in the sand, gravel, and Troutdale aquifers. In both plots, an 'artificial' regional flow is set by assigning a fixed boundary condition in the sand, gravel, and Troutdale aquifers of 11.8 feet along the right boundary (south and east boundaries) and 11.3 feet along the left boundary (north and west boundaries). Each scenario is run independently of each other. Comparing the flow dynamics in the local landfill area to the base case results (Figure 55), it can be seen that the three-dimensional flow model is insensitive to changes in the boundary conditions.

## 7.5   *Calibration:*

Calibration of most parameters is done in the 'upstream' models for use in the three-dimensional flow model; $K_{siltH}$, $K_{siltv}$, specific yield, and landfill infiltration are determined from calibration of the water balance model; $K_{siltL}$ is determined by calibration of the two-dimensional leachate mounding model; river and lake levels, non-landfill infiltration, confined storage coefficient, and conductivity's in the sand, gravel aquifer, and Troutdale Aquifer are determined from the regional flow model.

The telescopic approach of utilizing larger scale models to initialize boundary conditions and model parameters can introduce parameter uncertainty due to changing field scales between each model. The smallest heterogeneities that effect the local flow model are large enough to effect the regional model too. Because the regional model is calibrated to the six gravel wells around the landfill, we see that the base case parameter values simulates well data closely in the three-dimensional flow model, simply because of the boundary conditions set by the 'upstream' models. Sufficient data detailing the flow field directly below the landfill in the lower silt and gravel aquifer do not exist for detailed calibration of the three-dimensional flow model. Any parameter uncertainty that may arise due to changes in scale or local conditions is addressed through sensitivity analysis (see previous section).

# 8.    Contaminant Transport Model

The contaminant transport model is an extension of the three-dimensional local scale flow model.  The three-dimensional flow field produced by the flow model is utilized by the contaminant transport model to predict fate and transport of the contaminant plume.  This in turn produces a three-dimensional concentration field showing the shape and extent of the contaminant plume for each time step.  Knowing the extent of the plume will help us determine how effective the silt is at protecting the gravel aquifer; whether or not offsite migration has occurred; what is the extent of the groundwater contamination in the gravel aquifer; how fast is the plume moving; where will the plume be in the future; whether the landfill is affecting Smith and Bybee Lakes; and how will the proposed management plan of restoring Bybee Lake to intertidal conditions affect the contaminant plume migration in the gravel aquifer.  This section looks at the contaminant transport model, its concepts and mathematical application, parameter estimation and calibration, results, and sensitivity analysis.

## 8.1    Conceptual Model:

The contaminant mound dominates plume migration in the local landfill region.  Early on in the simulation, as the contaminant mound builds, a head differential is produced between the contaminant mound and the gravel aquifer and the contaminant mound and the sloughs.  This differential creates outward flux from the landfill laterally into the sloughs and vertically into the silt.

The rate of downward migration will depend upon the head gradient between the landfill mound and the gravel aquifer. Since the horizontal head gradient in the gravel is essentially flat, the controlling parameter for the head gradient between the landfill and the gravel aquifer is the silt thickness. The water balance model predicts that contaminant has penetrated the thinner parts of the silt under the north side of the landfill. Once in the gravel aquifer, contaminant migration becomes predominantly horizontal as defined by the regional flow, moving to the north away from the landfill. Interaction of the gravel aquifer and Bybee Lake due to the aquifer rising to within a few feet of the surface below Bybee Lake, could present a window for contamination to enter Bybee Lake and pollute the lake as well as the adjacent, connected surface waters.

Temporal fluctuations in the flow within the gravel aquifer due to seasonal precipitation changes, tidal influences, and differing management decisions for the water levels in Smith and Bybee Lake all contribute to plume migration due to increased dispersion. The temporal flow variations aid in 'sloshing' the plume back and forth within the gravel aquifer, possibly spreading the plume much farther than the mean flow would indicate. Depending upon the water level in the lakes, the lakes can act either as a discharge point for the gravel aquifer or a recharge point.

### 8.2 Mathematical Model:

To model the contaminant transport portion of the St. Johns Landfill, a commercial package, MT3D is used. MT3D is a modular, fully three-dimensional transport model capable of simulating contaminant transport due to advection and

dispersion, in addition to handling chemical reactions and source/sink mixing. The model uses a modular structure that links it directly to the previous flow model results produced by MODFLOW. MT3D uses a mixed Eulerian-Langrangian approach to solve the three-dimensional advective-dispersive-reactive equation (all references to MT3D, the transport equations, and model characteristics are summarized from the MT3D manual; Zheng, 1992). To simulate advection, MT3D uses a Lagrangian approach meaning the frame of reference is moving with the flow field. Particle tracking options include the method of characteristics technique (MOC), modified method of characteristics (MMOC) technique, or a hybrid of the two (HMOC). For modeling the St. Johns landfill, MMOC is used. The fixed Eulerian reference is used for solving the dispersion, source/sink, and reaction terms of the governing equation using a fully explicit finite-difference scheme.

The model assumes a conservative contaminant so only advection, dispersion, and source/sink mixing are modeled.

Governing Equations:

The governing equation for contaminant transport in groundwater is complex. This section looks at the governing equation solved by MT3D, the individual components of the equation (advection, dispersion, and source and sink mixing), and some of the issues in solving the equation.

The general partial differential equation describing contaminant transport in groundwater due to advection, dispersion, chemical reactions, and source/sink mixing as applied in this project is:

$$R \frac{\partial C}{\partial t} = \frac{\partial}{\partial x_i}\left[ D_{ij} \frac{\partial C}{\partial x_j} \right] - \frac{\partial}{\partial x_i}\left[ v_i C \right] + S/S$$

$$C = f(x, y, z, t)$$
$$v_i = f(x, y, z, t)$$

**(20)**

where:

| | |
|---|---|
| $C(x,y,z,t)$ | = Concentration of contaminants dissolved in groundwater. |
| $t$ | = Time |
| $x_i$ | = Distance along the respective Cartesian Coordinate axis. |
| $D_{ij}$ | = Hydrodynamic dispersion coefficient. |
| $v_i$ | = Seepage or linear pore water velocity. |
| S/S | = Inputs and outputs due to sources and sinks. |
| R | = Retardation factor. |

The equation describes the rate of change of concentration of contaminant over time and space. The first term on the right hand side describes the change of concentration due to dispersion, the second term due to advection, and the third term due to inputs and outputs from and sources and sinks.

Dividing through by the retardation factor, R, the 'velocity' is expressed in terms of a retarded velocity, $\frac{v_i}{R}$. This retarded velocity is due to the 'slowing' of contaminant movement due to sorption onto the soil particles. For the landfill site, the contaminant is considered to be conservative, predicting the 'worst' case scenario since any reactive contaminant will be contained within the bounds of the conservative

contaminant's plume. Since we are modeling a conservative contaminant in the St.

Johns Landfill, the retardation factor is one (no sorption).

MT3D uses the operator splitting concept in solving the governing equation.

Operator splitting assumes that each component of plume migration (advection,

dispersion, and source and sink mixing) act independently of one another and thus can

be carried out in a step wise fashion. The plume is moved forward in the velocity field

by advection, then the effects of dispersion are applied, and then the effects of any

sources or sinks are applied. In reality, each component has a continuous effect on

plume migration. For the operator splitting method to be valid, the time step must be

kept small. MT3D internally calculates the necessary time step to maintain accuracy.

Advection:

The advective term, $\dfrac{\partial}{\partial x_i}[v_i C]$ describes the transport of dissolved constituents

in groundwater that move with the groundwater flow. It is the only term which is

linked to the three-dimensional flow model and is linked by Darcy's law.

Mathematically the link is:

$$v_i = -\sum_{j=1}^{3} \frac{K_{ij}}{n} \frac{\partial \phi}{\partial x_j}$$

(21)

Where all like terms are defined as above and:

$K_{ij}$ = Direction component of the hydraulic conductivity.
$\phi$ = Hydraulic head as predicted by the 3d flow model.
N = Effective porosity.

158

MT3D has three choices in solving for advective transport. The first is MOC which tracks particles forward in time based on the three-dimensional velocity field. The second choice is MMOC which tracks particles backwards in time from each nodal point. This is the particle tracking method used in this study. The third choice, HMOC, is a hybrid of the first two choices, utilizing the MOC routine in areas where the concentration gradient is large, and the MMOC routine in areas where the concentration gradient is small. The strength of the MOC routine is its lack of numerical dispersion making it ideal in the case of large concentration gradients, while the strength of the MMOC routine is its fast computing speed and low memory requirements, making it ideal for large models. The third choice tries to take advantage of the MOC and the MMOC's strengths while downplaying each of their weaknesses.

Since no interpolation is required in the MOC technique it is virtually free from numerical dispersion. The disadvantage of the MOC technique is the computer power requirements it needs to track each particle through each time step. This is why the St. Johns Landfill modeling system uses the MMOC scheme as opposed to any of the others.

The MMOC Scheme:

The first step in particle tracking is to evaluate the groundwater velocity field. This requires velocities evaluated between nodal points so a velocity interpolation scheme must be used. MT3D uses a simple piecewise linear interpolation that assumes that the velocity varies linearly between each nodal point with respect to the

159

direction of the velocity component. The known velocity field allows for the tracking

of particles through that field. MT3D provides three algorithms for use in particle

tracking; a first-order Euler algorithm, a fourth-order Runge-Kutta algorithm, and a

combination of the two. Due to the high computational demand for the fourth-order

Runge-Kutta method, and because the velocity field below the landfill is fairly

uniform, the first-order Euler algorithm is used. This is given in the MT3D manual as:

$$
\begin{cases}
x^{n+1} = x^n + \dfrac{\Delta t}{R} V_x(x^n, y^n, z^n) \\[2mm]
y^{n+1} = y^n + \dfrac{\Delta t}{R} V_y(x^n, y^n, z^n) \\[2mm]
z^{n+1} = z^n + \dfrac{\Delta t}{R} V_z(x^n, y^n, z^n)
\end{cases}
$$

$$(22)$$

where

| | |
|---|---|
| $x^{n+1}, y^{n+1}, z^{n+1}$ | = Particle position at new time step. |
| $x^n, y^n, z^n$ | = Particle position at old time step. |
| $V_x, V_y, V_z$ | = Velocity components evaluated at $x^n$, $y^n$, $z^n$. |
| R | = Retardation factor (1 for conservative contaminant). |
| $\Delta t$ | = Length of time between n and n+1. |

The second step in particle tracking is to track the particle through the velocity

field. In the MMOC scheme, particles are placed at each node, and then tracked

backwards in time to find where the particle needed to originate from at the beginning

of the time step (designated as coordinates $\hat{x}, \hat{y}, \hat{z}$) in order to end up at a nodal point

at the end of the time step. The concentration at the nodal point at the end of the time

step is then the concentration associated with the position $\hat{x}, \hat{y}, \hat{z}$. Usually the position

$\hat{x}, \hat{y}, \hat{z}$ does not coincide with a nodal point making it necessary to interpolate the

concentration at $\hat{x}, \hat{y}, \hat{z}$ from neighboring nodal points. MT3D uses a first-order polynomial interpolation scheme also called trilinear interpolation in three dimensions. If $\hat{x}$ is located between nodes $x_{j-1}$ and $x_j$, and $\hat{y}$ is located between points $y_{i-1}$ and $y_i$, and $\hat{z}$ is located between $z_{k-1}$ and $z_k$, then the concentration at $\hat{x}, \hat{y}, \hat{z}$ is:

$$
\begin{aligned}
C(\hat{x}, \hat{y}, \hat{z}) = &(1-\omega_x)(1-\omega_y)(1-\omega_z)C_{i-1,j-1,k-1} + (1-\omega_x)\omega_y(1-\omega_z)C_{i-1,j,k-1} + \\
&\omega_x(1-\omega_y)(1-\omega_z)C_{i,j-1,k-1} + (1-\omega_x)(1-\omega_y)\omega_z C_{i-1,j-1,k} + \omega_x\omega_y(1-\omega_z)C_{i,j,k-1} + \\
&(1-\omega_x)\omega_y\omega_z C_{i-1,j,k} + \omega_x(1-\omega_y)\omega_z C_{i,j-1,k} + \omega_x\omega_y\omega_z C_{i,j,k}
\end{aligned}
$$

(23)

where $\omega_x, \omega_y, \omega_z$ are interpolation factors given by:

$$
\begin{cases}
\omega_x = \dfrac{\hat{x} - x_{i-1}}{0.5\Delta x_i + 0.5\Delta x_{i-1}} \\[2mm]
\omega_y = \dfrac{\hat{y} - y_{j-1}}{0.5\Delta y_j + 0.5\Delta y_{j-1}} \\[2mm]
\omega_z = \dfrac{\hat{z} - z_{k-1}}{0.5\Delta z_k + 0.5\Delta z_{k-1}}
\end{cases}
$$

(24)

If the point $\hat{x}, \hat{y}, \hat{z}$ is at a nodal point, $C(\hat{x}, \hat{y}, \hat{z})$ is just $C(i,j,k)$, where i,j,k is the x,y,z coordinate of the nodal point. The benefit of using a low order interpolation scheme such as this is its computational efficiency and small mass balance error. The major drawback is that it is susceptible to numerical dispersion, especially in the cases of sharp concentration fronts. For concentration fields that are relatively smooth, the numerical dispersion due to the MMOC technique is insignificant. For the St. Johns Landfill, the front of the contaminant plume is fairly sharp. Data and sensitivity analysis suggest that field dispersion is greater than the numerical dispersion caused by the MMOC technique, so the use of the MMOC technique can be justified.

Dispersion:

Once the concentration due to pure advection for each cell is determined, a weighted average of the concentration at the beginning of the time step and the end of the time step is used to calculate the concentration change due to dispersion as discussed above. The governing equation describing dispersion solved by MT3D is:

$$\frac{DC_{dsp}}{Dt} = \frac{1}{R}\frac{\partial}{\partial x_i}\left[D_{ij}\frac{\partial C}{\partial x_j}\right]$$

**(25)**

The term $D_{ij}$ is the 9 components of the hydrodynamic dispersion coefficient, evaluated at the cell interfaces. The components are calculated from the velocity components and the dispersivities by:

$$D_{xx} = \alpha_L\frac{v_x^2}{|v|} + \alpha_T\frac{v_y^2}{|v|} + \alpha_T\frac{v_z^2}{|v|} + D^* \qquad D_{xy} = D_{yx} = (\alpha_L - \alpha_T)\frac{v_x v_y}{|v|}$$

$$D_{yy} = \alpha_L\frac{v_y^2}{|v|} + \alpha_T\frac{v_x^2}{|v|} + \alpha_T\frac{v_z^2}{|v|} + D^* \qquad D_{xz} = D_{zx} = (\alpha_L - \alpha_T)\frac{v_x v_z}{|v|}$$

$$D_{zz} = \alpha_L\frac{v_z^2}{|v|} + \alpha_T\frac{v_x^2}{|v|} + \alpha_T\frac{v_y^2}{|v|} + D^* \qquad D_{yz} = D_{zy} = (\alpha_L - \alpha_T)\frac{v_y v_z}{|v|}$$

where

| | |
|---|---|
| $\alpha_L$ | = Longitudinal dispersivity. |
| $\alpha_T$ | = Transverse dispersivity. |
| $v_x, v_y, v_z$ | = Velocity components. |
| $D^*$ | = Effective molecular diffusion coefficient. |
| $|v| = (v_x^2 + v_y^2 + v_z^2)^{\frac{1}{2}}$ | = Magnitude of the velocity vector. |

The longitudinal and transverse dispersivities are input from the user with one of each value entered for each cell.

The dispersion term is solved with the fully explicit, central finite-difference scheme that creates a limitation on the time step in order to maintain stability.

Source and Sink:

Source and sink mixing is the changing of aquifer concentration due to the input or exit of mass solute through a source or sink. The change in concentration due to source and sink mixing is given in our governing equation as:

$$\frac{DC_{ssm}}{Dt} = -\frac{q_s}{Rn}\left(C - C_s\right)$$

(26)

where:

$C_{ssm}$ = Concentration due to source and sink mixing.
$q_s$ = Flux of sink or source per unit volume.
R = Retardation factor.
n = Porosity.
C = Aquifer concentration.
$C_s$ = Source or sink concentration.

Source and sink conditions must match that of the flow model. For the St. Johns Landfill, we have two sinks, and one source; the surface drainage from the landfill and dikes, and the flux into the sloughs and lakes make up the sinks, with recharge being the only source. River cells can act as a source in the case where the river level exceeds the head level in the aquifer below it. For water entering the domain from either river cells or recharge, the concentration is given as zero. For water leaving the domain from drainage or river cells, the concentration is given as the concentration in the aquifer.

163

### 8.3    *Boundary Conditions and Discretization:*

Spatial and temporal discretization are the same as the three-dimensional flow model (section 7.3) with output being a three-dimensional yearly averaged concentration field for each year of the simulation.

An original pre-processor was written and used for the transport model to sort and organize the large amount of data needed for the simulation. The pre-processor applies user input in the correct format and distributes those data that need to be distributed across the model domain on a cell by cell basis. The pre-processor source code and data requirements are given in Appendix D.

When the three-dimensional flow model is discretized, the domain is chosen such that the boundaries are outside the area of potential contaminant migration, thus a constant concentration of zero is used on all boundaries of the model except for the landfill, where a positive, constant, source concentration is used. Source concentration is considered as an input parameter and is addressed in detail below. Initial conditions assume no contamination in the silt prior to 1950, (concentration of zero in all cells except for the landfill cells).

### 8.4    *Parameter Estimation and Calibration:*

Dispersivity, source concentration, and porosity are the three parameters that require calibration in the transport model. Site stratigraphy and water level data are from existing data and are identical to that used in the three-dimensional flow model.

The transport model needs no new flow parameters since the flow field is predicted by the three-dimensional flow model.

Dispersivity:

Dispersivity is a physical property of the medium. Experiments have demonstrated that, in an isotropic medium, dispersion varies linearly with the average speed of groundwater flow (Wang and Anderson, 1982), thus the dispersion coefficient is defined as the dispersivity times the groundwater velocity. Dispersivity can be thought of as a medium's ability to disperse a plume; the higher the dispersivity constant, the higher the dispersion. Dispersivity can have a major effect on the extent, size, shape, and concentration of the plume. Total dispersion in the transport model represents a combined 'effective' parameter accounting for dispersion caused by geologic heterogeneities, dispersion caused by temporal fluctuations, and numerical dispersion caused by model interpolations.

Since the model is a yearly averaged model, the dispersivity must account for temporal fluctuations. On a yearly averaged basis, regional groundwater velocities are slow in the St. Johns Landfill region (~1-2 ft/yr in the gravel across the local modeling domain), however the regional flow is known to exhibit temporal fluctuations and even flow direction changes over time. In addition, localized effects (surface water features, landfill mounding, etc.) create smaller zones of higher velocity areas. These temporal dynamics and spatial variations have the effect of dispersing the plume at a greater rate than would occur with yearly averaged conditions. Dispersivity values must account for these dynamics.

To calibrate dispersivity, the front edge of the plume needs to be known. At present, there exist no data that describe the extent of the plume so dispersivity cannot accurately be calibrated. A maximum value of dispersivity can be resolved by increasing the value until the predicted migration shows up in wells where data show no plume. The maximum value of dispersivity corresponds to about 50 feet in the silt and sand, and 250 feet in the gravel and Troutdale. Base case values of 1 ft. in the silt and 5 ft in the sand and gravel are adopted.

Source Concentration:

Source concentration for all landfill cells is 1000 mg/l of chloride, modeled as a conservative contaminant. As the expansion of the landfill is modeled over time, any new landfill cells added to the model domain, are added with an instantaneous concentration of 1000 mg/l. This figure is used for two reasons; one is a 1000 mg/l roughly represents an overall average of concentration data in the landfill, and two; it is an easy factor for scaling. No attempt is made to represent the source concentration as spatially varying because concentration data within the landfill are too sparse and the model refinement too large. As the average source concentration is refined with better data, the results can be adjusted using a scaling factor.

Effective Porosity:

Effective porosity is the porosity that is available for groundwater flow and effects the extent of the plume by controlling the advective speed with which it moves. Assuming the hydraulic conductivity is constant, contaminant moving through media with a large effective porosity will move slower than if it were moving through media

166

with a small effective porosity. Published values (Marsily, 1986) for the various porosity's are used for the different geologic units within the model domain. Concentration data over time in the silts suggest a downward migration rate of approximately 1 ft/yr. Based on this the base case values for porosity are 0.40 for silt, and 0.25 for the sand, gravel, and Troutdale. Like dispersivity, porosity cannot be fully calibrated until the front of the plume is located and an average migration speed is determined.

Parameter uncertainty caused by lack of calibration or inadequate calibration as a result of limited or unrepresentative data is addressed through a systematic sensitivity analysis as described below.

## 8.5 Results:

The results from the water balance model show three hot spots on the north side of the landfill where the simulated chloride has penetrated the silt into the gravels. The three-dimensional flow model and contaminant transport model show only the hot spot which is farthest west, about two thirds along the length of the North Slough. This is due to the transport model's ability to simulate the growth of the landfill in the new expansion area. The contaminant mounding in the expansion area on the east side of the landfill has not been present long enough for contamination to penetrate the silt. The model simulations show primarily vertical flow in the silt and a slight northerly migration of the plume once it reaches the gravel. The driving force for vertical migration through the silt is the contaminant mound, while the driving force for the

northerly migration through the gravel aquifer is the Willamette and Columbia River water levels.

Figure 58 through Figure 61 show contours of the plume concentration at the top of the gravel aquifer for the base case values in the years 1980, 1990, 2000, and 2010. Breakthrough into the gravel aquifer is clearly seen in the hot spot directly below the North Slough. Maximum concentration in the center of the hot spot is approximately 65% of the source concentration in 2010. Northward migration of the plume once it enters the gravel can be seen over the 40 years represented in the figures. Due to anisotropy and a small northward gradient, the center of the hot spot is located directly below the North Slough.

Figure 63 through Figure 66 show the corresponding cross sections of the plume at the hot spot area. Mounding and dissipation can clearly be seen as can the northward migration of the plume after closure. Plume dynamics in the silt are dominated by the silt characteristics and the leachate mounding process and are virtually unaffected by conditions in the gravel.

Most of the plume is contained within the silt layer for the entire simulation and appears it may only completely penetrate the silt in areas other than the 'hot spot' in the southeast portion of the landfill (Figure 61). As the plume enters the gravel aquifer, the flow changes to mostly horizontal in the northward direction. By the year 2010 the plume has moved approximately 250 feet north of the North Slough.

Extrapolating results into the future using an average transport rate (including dispersion) between the landfill and Bybee lake in the gravels of 10 feet per year as predicted by the transport model, shows that the plume should not be approaching the gravel 'window' below Bybee Lake for another 50 years after the end of the simulation (2070). The effects of the plume on Bybee Lake at that point are unknown, however the plume will continue to dissipate with time due the dilution effects caused by dispersion.

## Concentration at Top of Gravel Aquifer
## Base Case - 1980
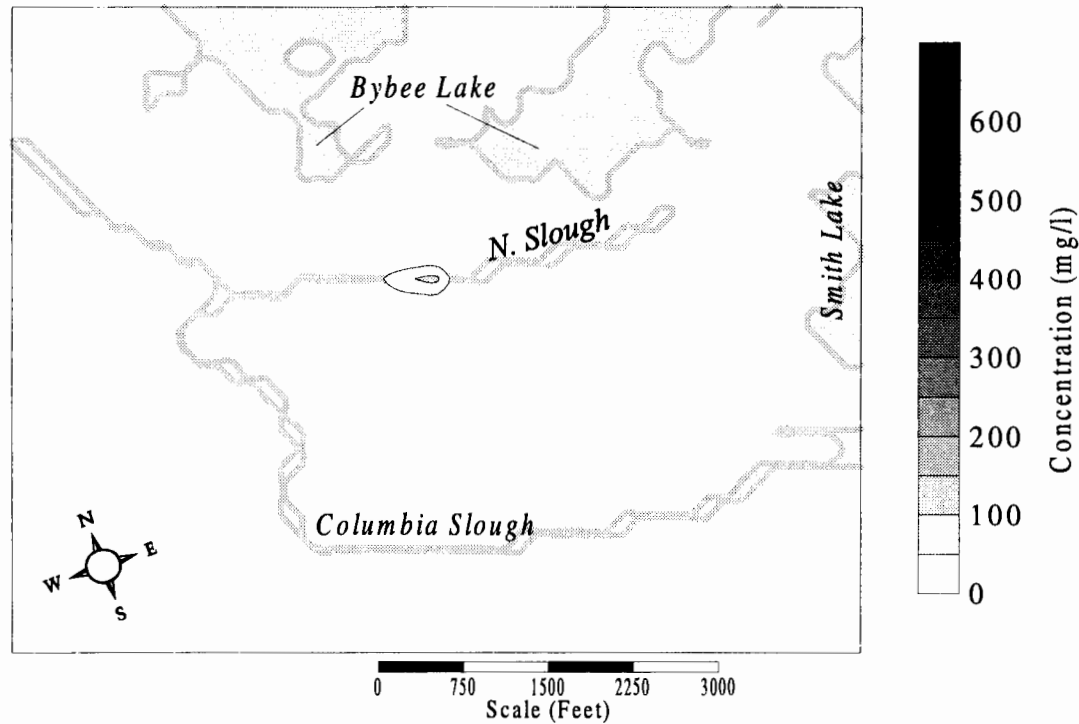


**Figure 58 - Concentration of chloride plume at top of gravel in 1980 for base case values. Slight breakthrough can be seen in the mid-North Slough area with a maximum concentration in the gravel of 112.37 mg/l (~11% of source concentration). The silt thickness is about 30 feet thick at the hot spot, which corresponds to a travel time of approximately 1 ft/year for contaminant to first reach the gravel.**

Concentration at Top of Gravel Aquifer
Base Case - 1990
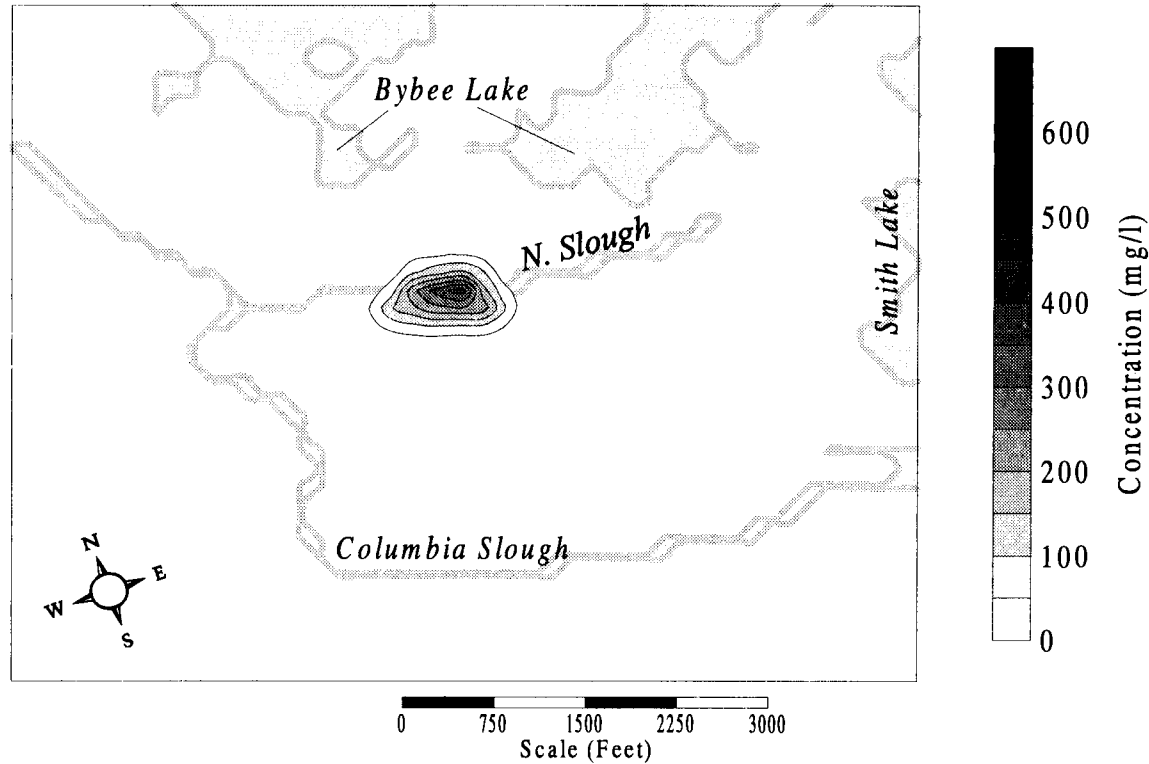
**Figure 59 - Concentration of chloride plume at top of gravel in 1990 for base case values. Lateral (east/west) spreading of the hot spot can be seen when compared to the 1980 plot as the plume continues to migrate downward into the gravel. Maximum concentration in the gravel is predicted to be 361.404 mg/l (~36% of source concentration).**

Concentration at Top of Gravel Aquifer
Base Case - 2000
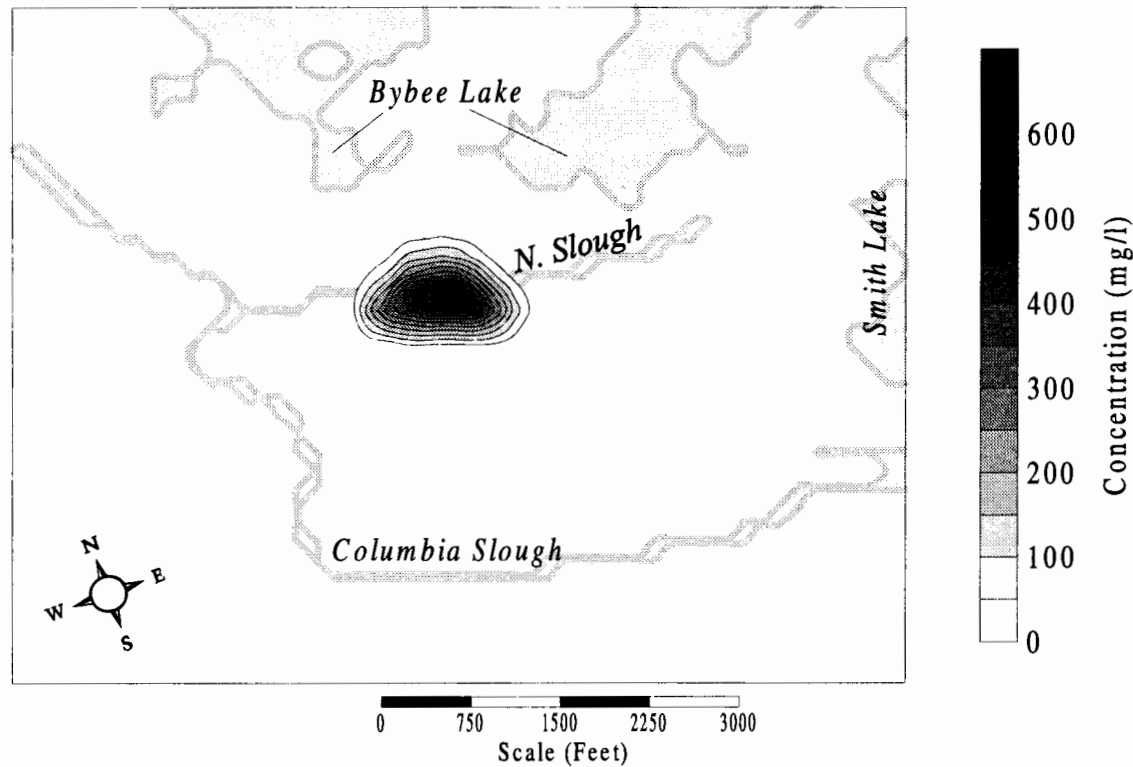
**Figure 60 - Concentration of chloride plume at top of gravel in 2000 for base case values. Northward migration of the plume towards Bybee Lake can be seen due to the regional flow in the gravel. The lateral (east/west) extent of the plume is at about its maximum as downward migration has slowed due to the dissipation of the leachate mound in the landfill. Maximum concentration in the gravel is predicted to be 573.661 mg/l (~57% of source concentration).**

Concentration at Top of Gravel Aquifer
Base Case - 2010
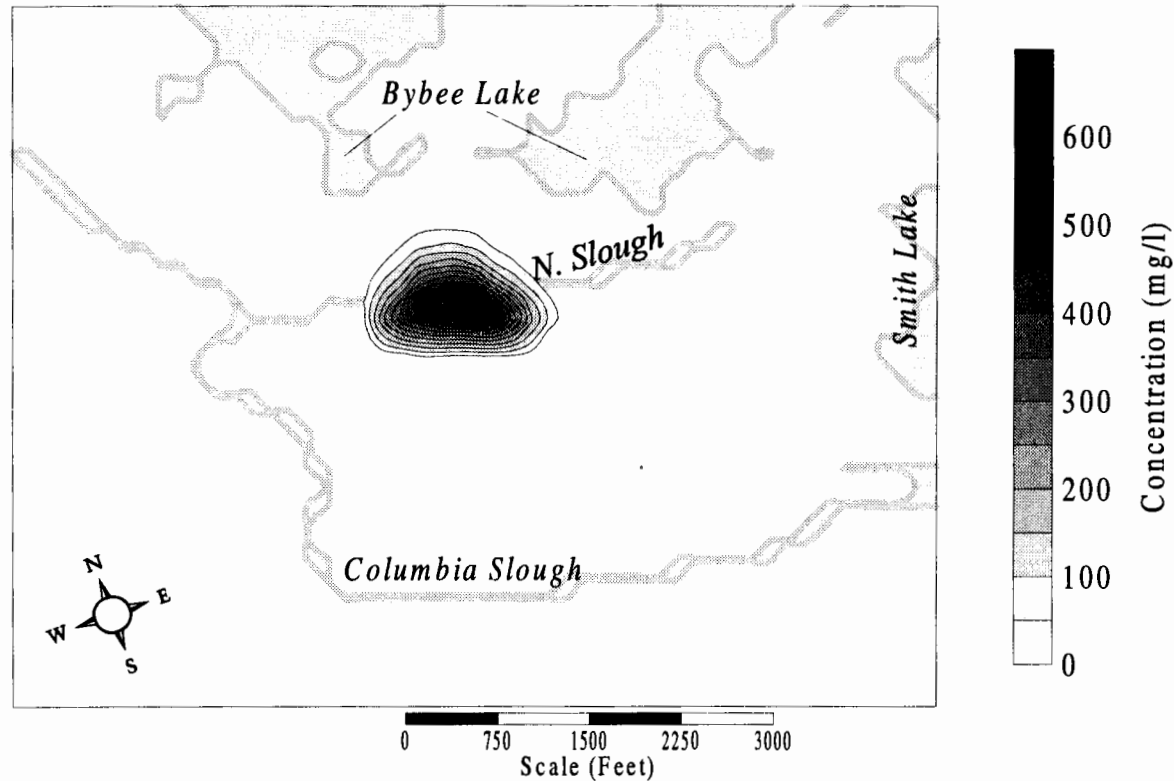
**Figure 61 - Concentration of chloride plume at top of gravel in 2010 for base case values. The plume has moved approximately 250 feet north of the North Slough at this point due to the regional flow in the gravel. Slight breakthrough can be seen on the south side of the landfill underneath subarea 5a, which is the oldest part of the landfill. Maximum concentration in the gravel is predicted to be 651.063 mg/l (~65% of source concentration).**

Location of Column 25 Cross Section
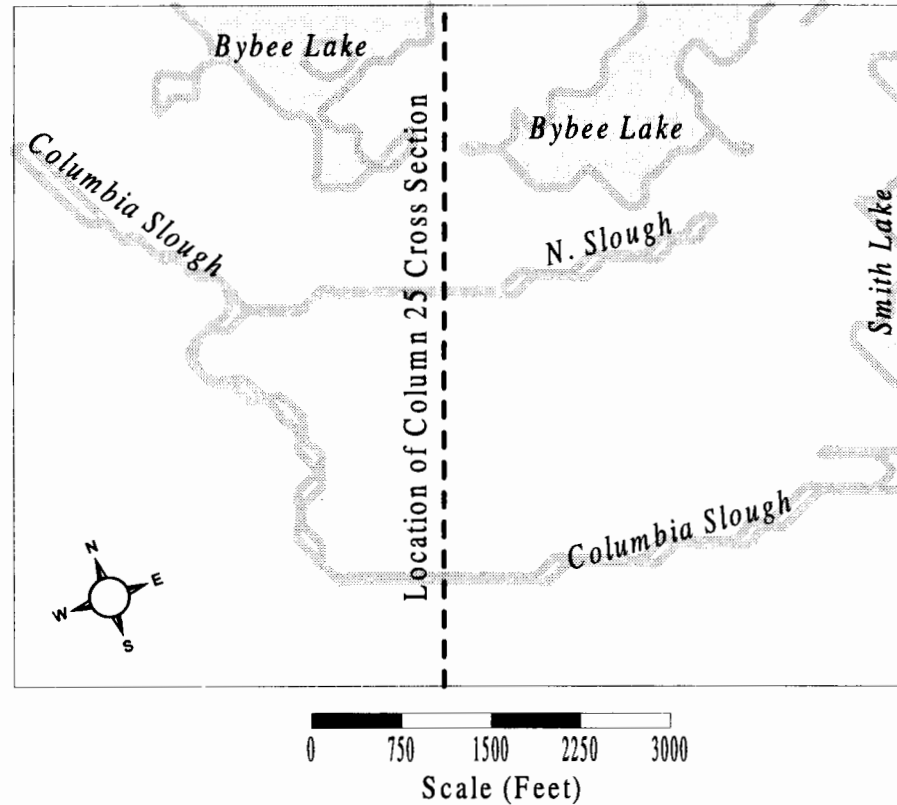
**Figure 62 - Location of cross section used for plume visualization through the hot spot area on computational grid column 25.**

Plume Cross Section on Column 25
Base Case - 1980
Vertically Exaggerated 8x
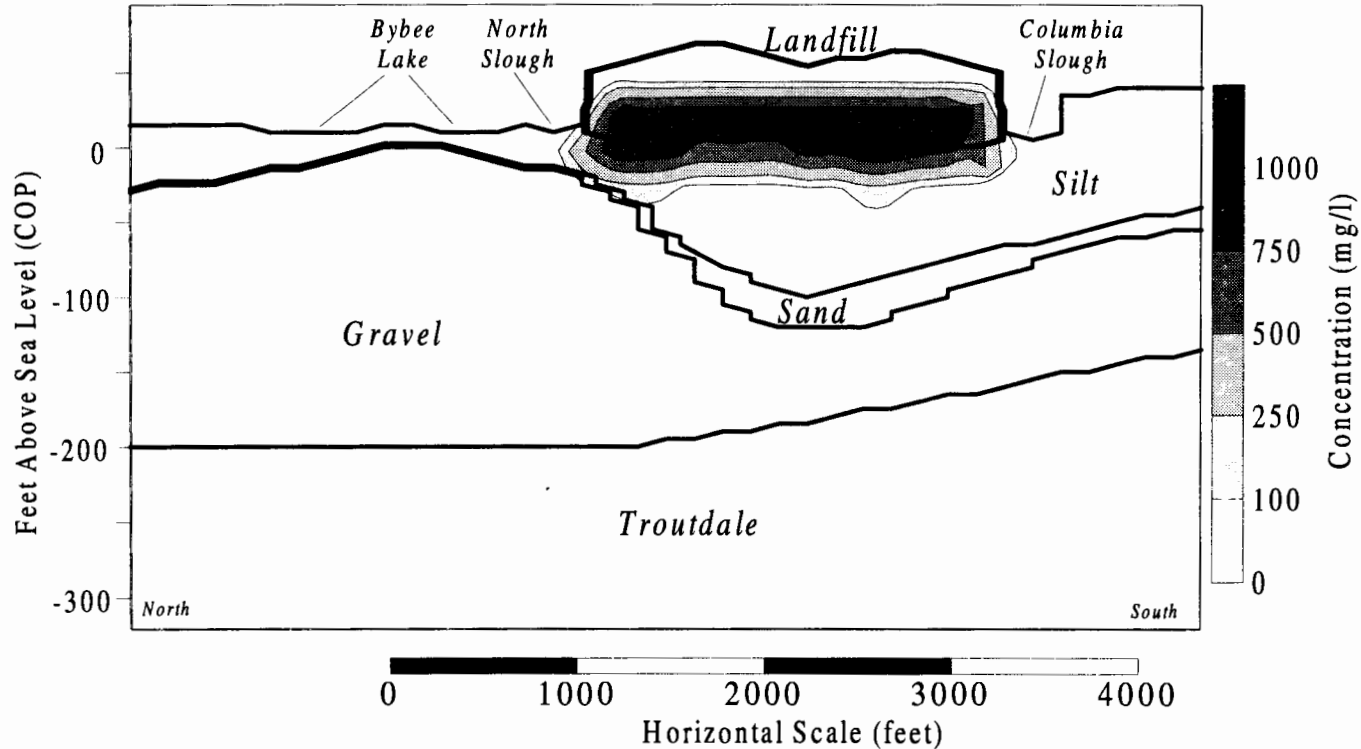
**Figure 63 - Cross section of chloride plume on column 25 through the hot spot in 1980 using base case values. Breakthrough of the plume can be seen where the 100 mg/l contour contacts the gravel layer. Downward migration of the plume is dominated by the silt characteristics and the leachate mound dynamics. See Figure 62 for location of cross section.**

Plume Cross Section on Column 25
Base Case - 1990
Vertically Exaggerated 8x
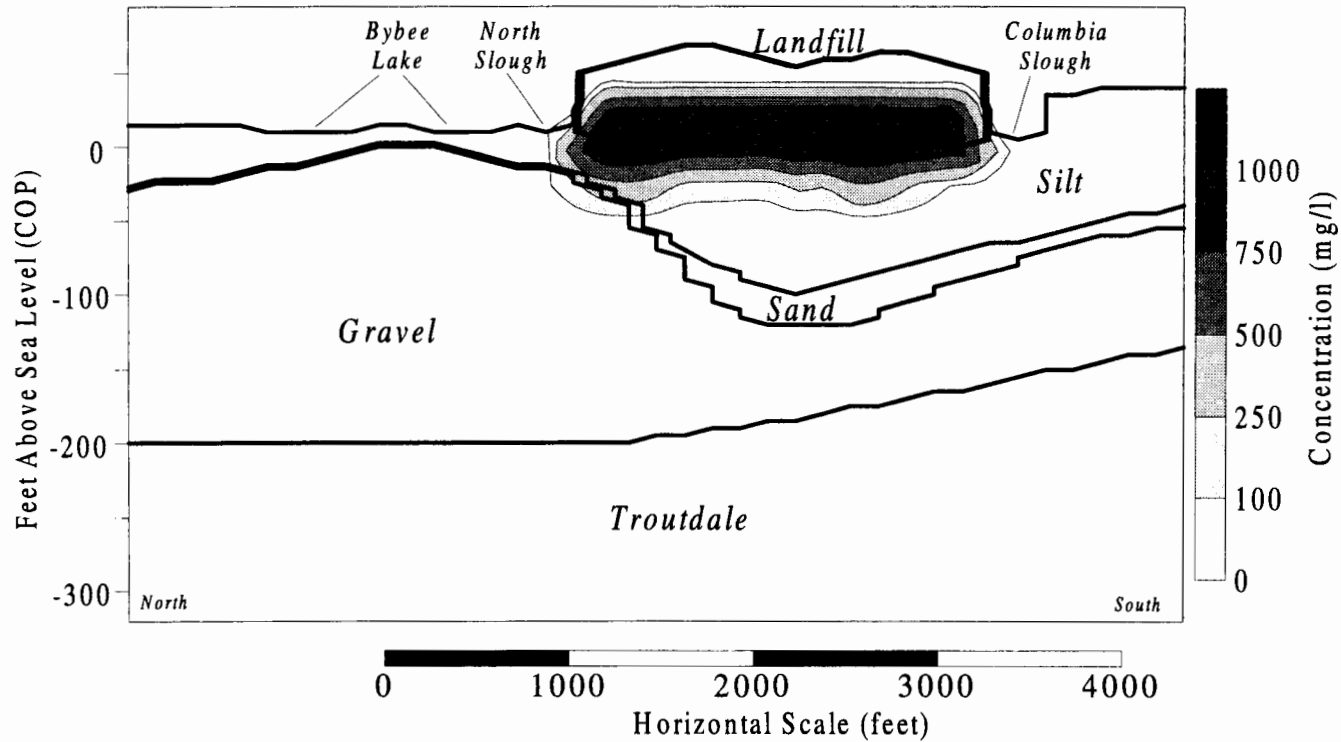
**Figure 64 - Cross section of chloride plume on column 25 through the hot spot in 1990 using base case values. Downward migration of the plume continues to move contaminant into the gravel layer. Due to the thickness of the silt along this cross section, downward migration is slower in areas away from the hot spot. See Figure 62 for location of cross section.**

176
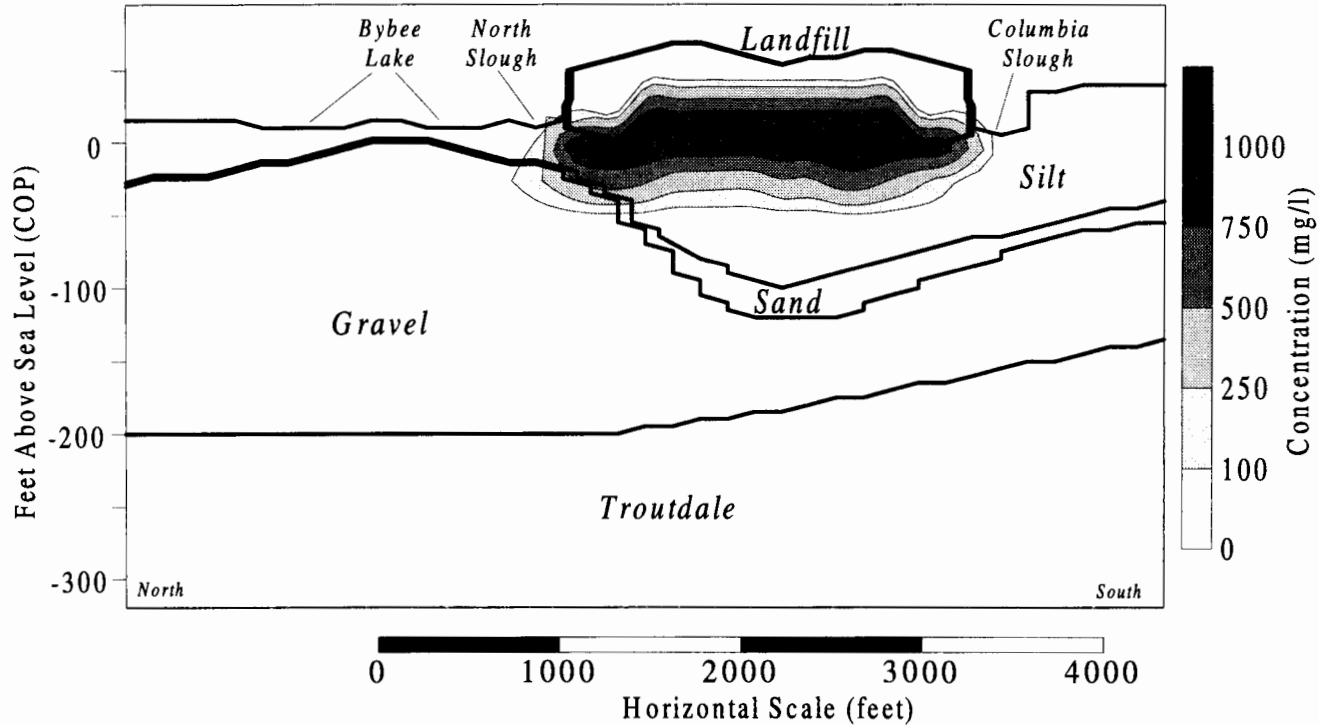
**Figure 65 - Cross section of chloride plume on column 25 through the hot spot in 2000 using base case values. Leachate mound dissipation can be seen by the irregular shape of the top of the leachate mound. As the mound dissipates, the downward migration of the plume due to advection decreases. Some northward migration of the plume can be seen in the gravel when compared to the 1990 plot. See Figure 62 for location of cross section.**

Plume Cross Section on Column 25
Base Case - 2010
Vertically Exaggerated 8x
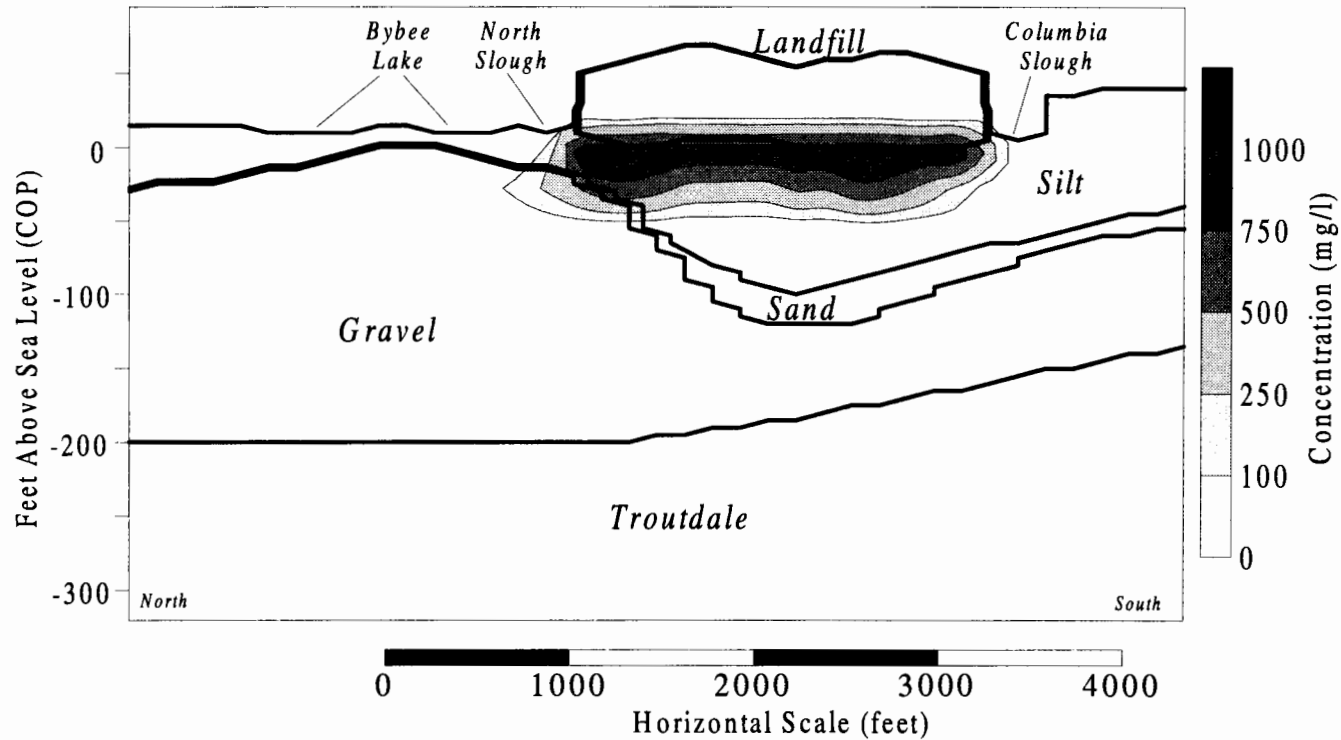
**Figure 66 - Cross section of chloride plume on column 25 through the hot spot in 2010 using base case values. The leachate mound has completely dissipated at this point. A high concentration area is present in the silt just below the landfill. Migration northward in the gravel has progressed approximately 250 feet north of the North Slough. See Figure 62 for location of cross section.**

Sensitivity Analysis:

Sensitivity analysis (see below) indicates that the silt conductivity and silt thickness are the key factors in containing the plume within the horizontal bounds of the landfill and the vertical bounds of the silt/gravel interface. Sensitivity analysis is performed in regards to silt conductivity, dispersivity, and lake levels as controlled by the dam on the North Slough. The plume's migration is most effected by the values of $K_{siltH}$, and $K_{siltV}$. These values control the migration of the plume through the silt and effect the extent of the plume within the gravel aquifer.

Silt Conductivity:

Figure 67 through Figure 74 show various views of the concentration distribution of the plume with anisotropy equal to 40 ($K_{siltH} = 7.0 \times 10^{-5}$ cm/sec, $K_{siltV} = 1.75 \times 10^{-6}$ cm/sec) and 200 ($K_{siltH} = 1.07 \times 10^{-4}$ cm/sec, $K_{siltV} = 5.34 \times 10^{-7}$ cm/sec). Conductivity values are changed to maintain the mean leachate mound of 25 feet at each anisotropy value. The results agree very well with the water balance model showing that at the low value of 40, contaminant has clearly penetrated the silt on the northern side of the landfill where the silt is thinnest and has been present in the gravels long enough to show northward migration. In addition, slight penetration through the silt is predicted in the southeast corner of the landfill due to the additional effect of dispersion on the plume that is not modeled in the water balance model. With an anisotropy value of 200, penetration is barely perceptible in the hot spot area and the center of the contour has moved north due to an increased rate of lateral plume

migration  caused by the high anisotropy in the silt.  As the contaminant mound dissipates, contaminant may never penetrate the other relatively thick portions of the silt.

Concentration at Top of Gravel Aquifer, 1990
Kxsilt = 7.0e-5 cm/sec, Kzsilt = 1.75e-6 cm/sec
Anisotropy = 40
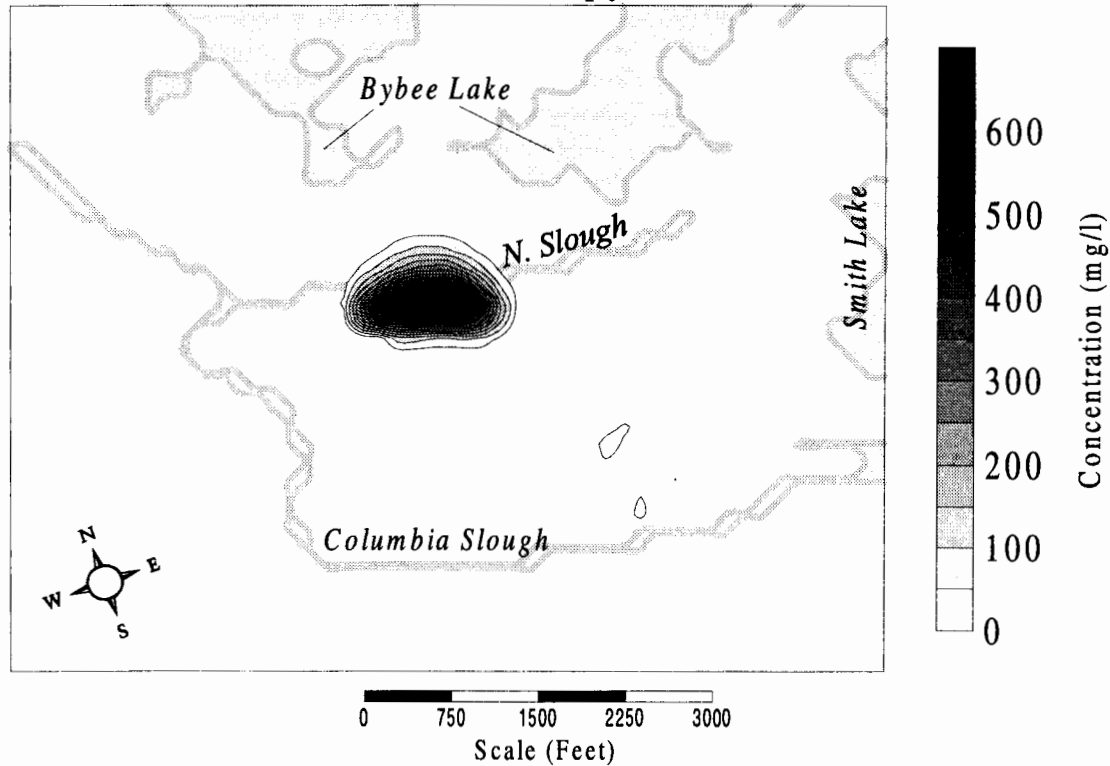


**Figure 67 - Sensitivity to silt conductivity. Concentration of chloride at top of gravel aquifer in 1990 with an anisotropy of 40. Slight penetration can be seen underneath the landfill in subarea 5a (southeast portion of landfill). Maximum concentration in the gravel is predicted to be 699.201 mg/l (~70% of source concentration) for this case.**

Concentration at Top of Gravel Aquifer, 1990
Kxsilt = 1.07e-4 cm/sec, Kzsilt = 5.34e-7 cm/sec
Anisotropy = 200
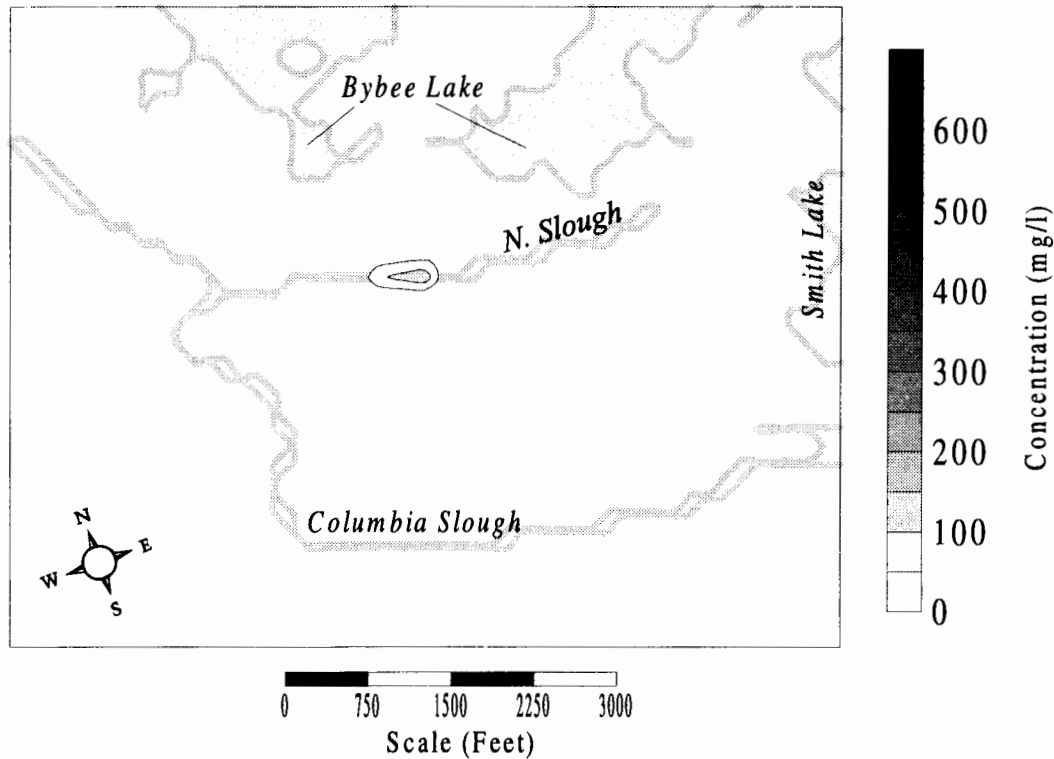
**Figure 68 - Sensitivity to silt conductivity. Concentration of chloride at top of gravel aquifer in 1990 with an anisotropy of 200. Penetration at the hot spot area is slight due to the high horizontal conductivity as compared to the vertical conductivity. Maximum concentration in the gravel is predicted to be 133.033 mg/l (~13% of source concentration) for this case.**

Concentration at Top of Gravel Aquifer, 2010
K x silt = 7.0e-5 cm/sec, K z silt = 1.75e-6 cm/sec
Anisotropy = 40
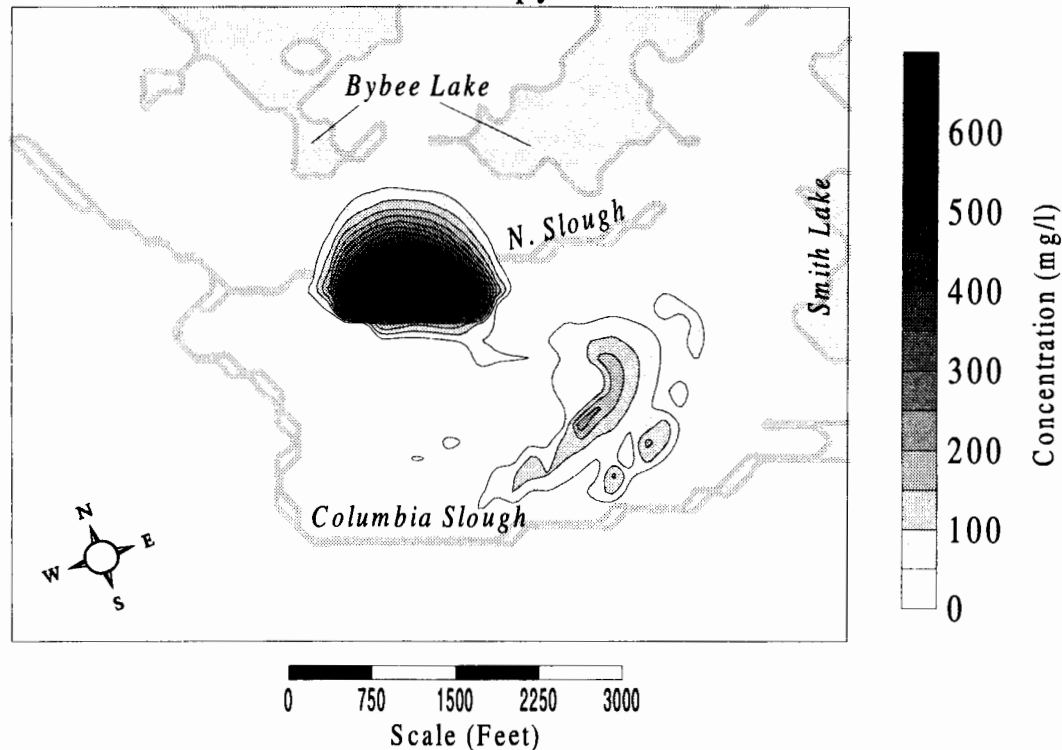
**Figure 69 - Sensitivity to silt conductivity. Concentration of chloride at top of gravel aquifer in 2010 with an anisotropy of 40. Penetration in the hot spot area is severe with a maximum concentration in the gravel predicted to be 885.672 mg/l (~86% of source concentration) . For this case, all areas of silt penetration coincide where non-expansion areas of the landfill (older areas) sit atop thinner parts of the silt, following the contours of the top of the gravel.**

Concentration at Top of Gravel Aquifer, 2010
Kxsilt = 1.07e-4 cm/sec, Kzsilt = 5.34e-7 cm/sec
Anisotropy = 200
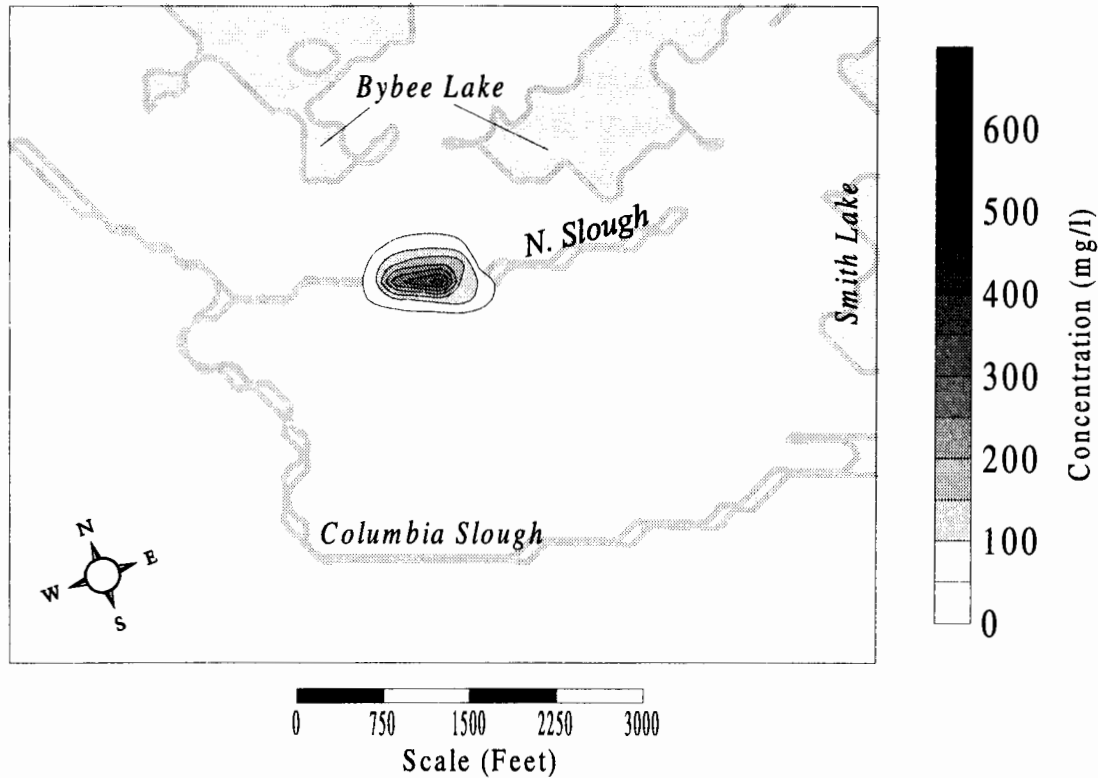
**Figure 70 - Sensitivity to silt conductivity. Concentration of chloride at top of gravel aquifer in 2010 with an anisotropy of 200. After 60 years, penetration into the gravel aquifer is slight as compared to other cases. Maximum concentration in the gravel is predicted to be 357.492 mg/l (~36% of source concentration) for this case.**

Plume Cross Section on Column 25, 1990
Kxsilt = 7.0e-5 cm/sec, Kzsilt = 1.75e-6 cm/sec
Anisotropy = 40, Vertically Exaggerated 8x
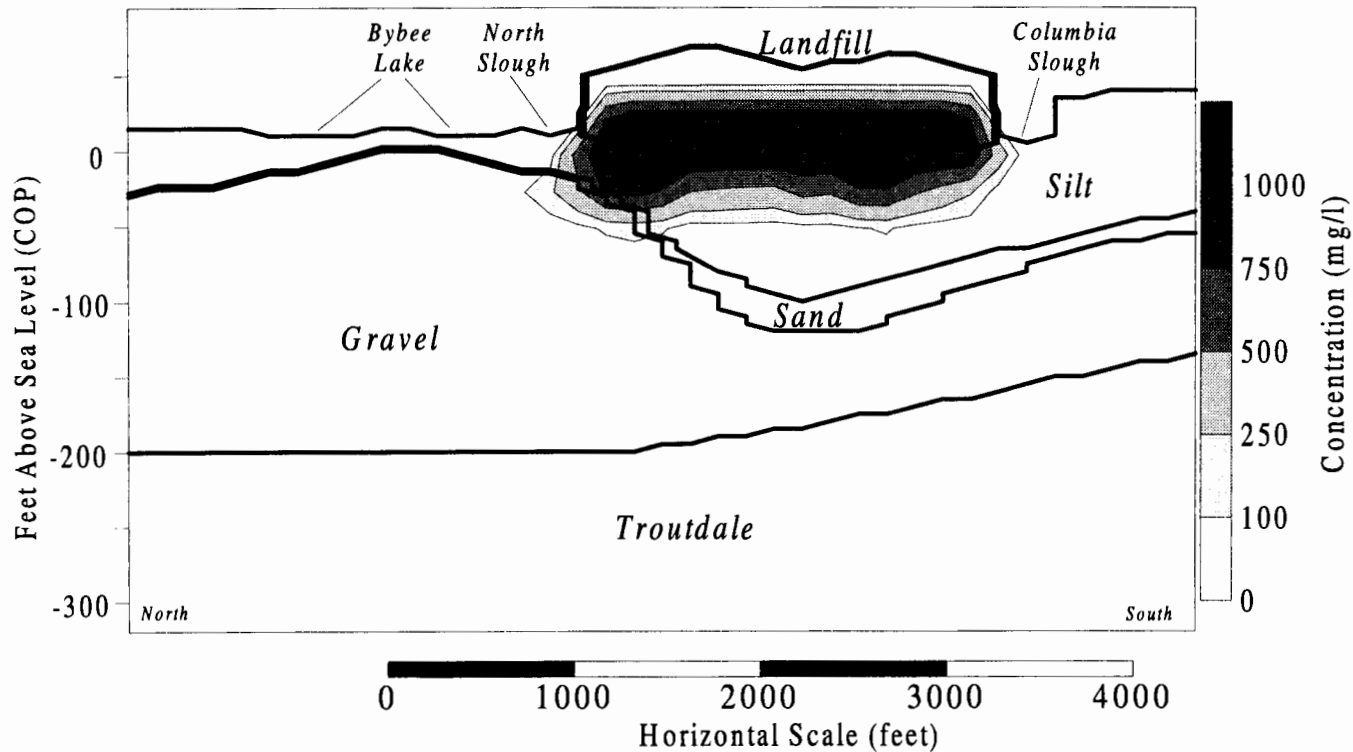
**Figure 71 - Sensitivity to silt conductivity. Cross section of chloride plume on column 25 through the hot spot in 1990 with an anisotropy of 40. Penetration is greater than the base case due to the higher vertical conductivity in the silt. Due to the thickness of the silt along this cross section, downward migration is slower in areas away from the hot spot. See Figure 62 for location of cross section.**

185
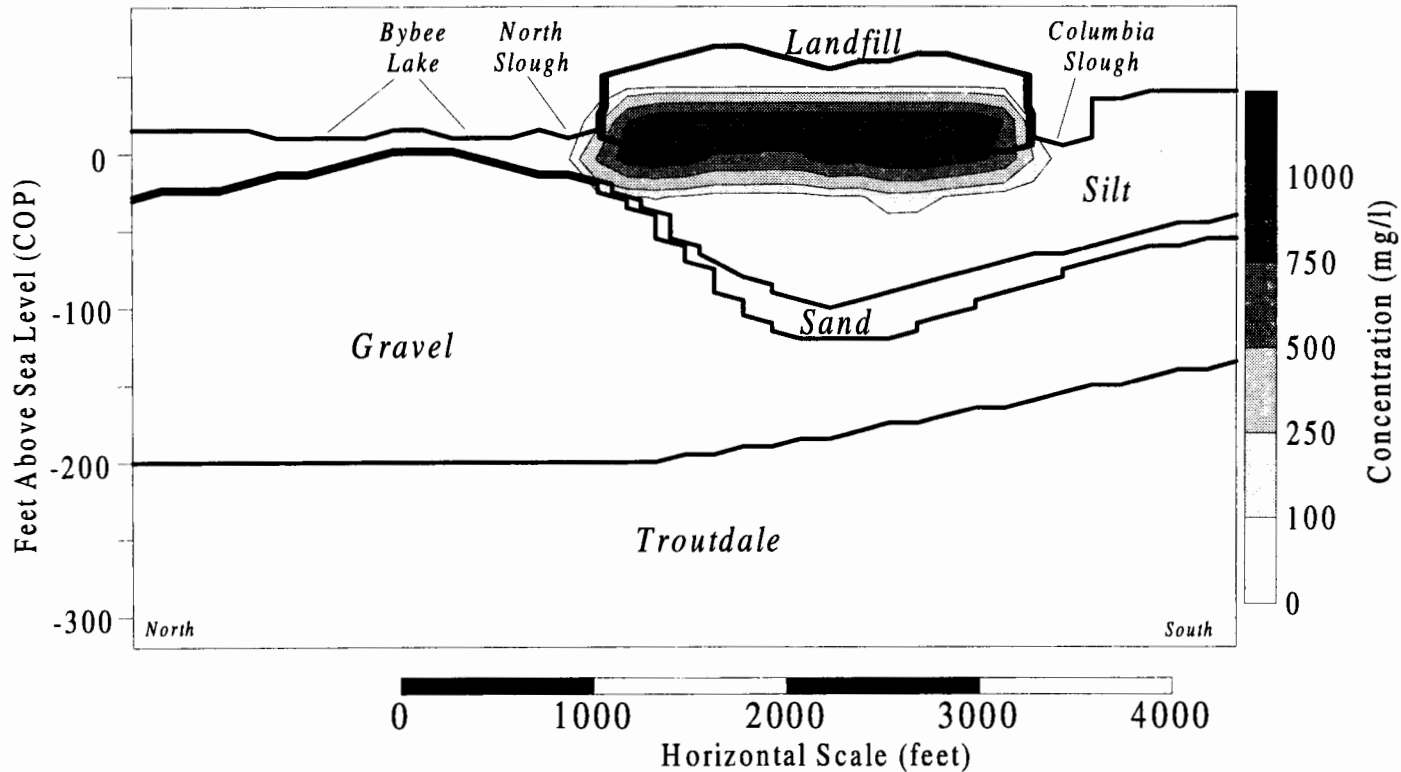
Figure 72 - Sensitivity to silt conductivity.  Cross section of chloride plume on column 25 through the hot spot in 1990 with an anisotropy of 200.  Vertical migration is very slow due to the small vertical conductivity and high horizontal conductivity in the silt.  See Figure 62 for location of cross section.

**Plume Cross Section on Column 25, 2010**
**Kxsilt = 7.0e-5 cm/sec, Kzsilt = 1.75e-6 cm/sec**
**Anisotropy = 40, Vertically Exaggerated 8x**
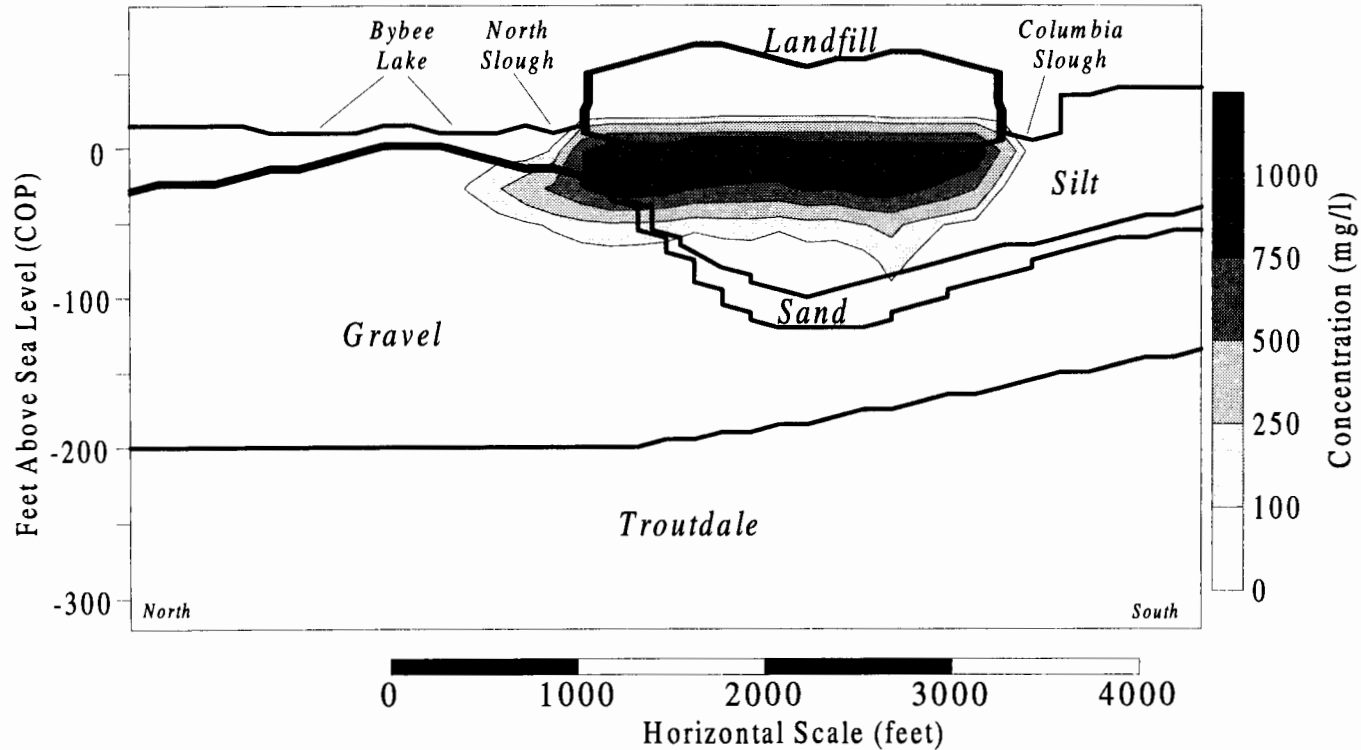
**Figure 73 - Sensitivity to silt conductivity. Cross section of chloride plume on column 25 through the hot spot in 2010 with an anisotropy of 40. Penetration from the south side of the landfill can be seen. Northward migration is greater than the base case example since contaminant enters the gravel aquifer earlier and has longer to travel. Mounding at this point as dissipated. See Figure 62 for location of cross section.**
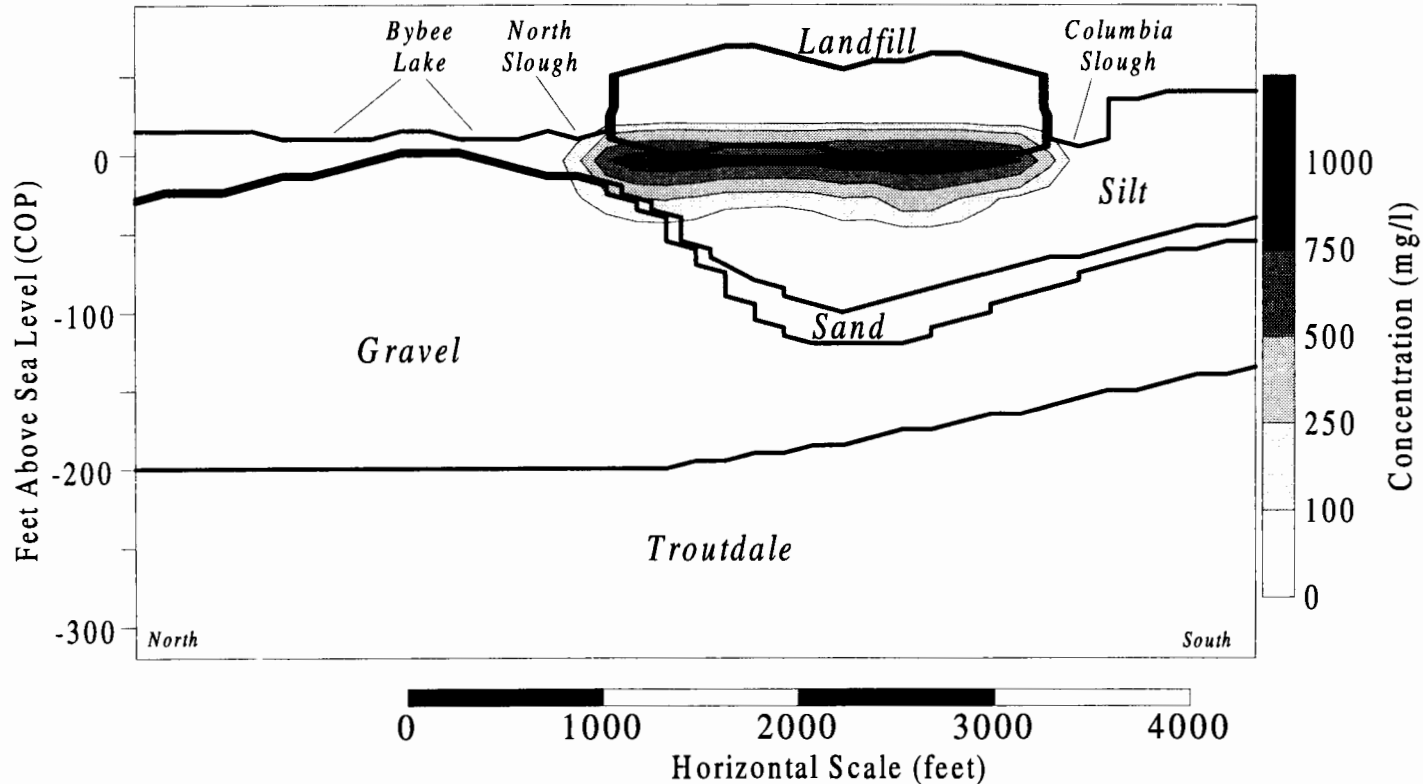
Figure 74 - Sensitivity to silt conductivity. Cross section of chloride plume on column 25 through the hot spot in 2010 with an anisotropy of 200. The north edge of the plume is just able to enter the gravel in this case. Penetration elsewhere is slight. See Figure 62 for location of cross section.

Porosity:

Porosity is split into two catagories; total porosity and effective porosity. Total porosity is defined as the volume of the voids divided by the total volume. Effective porosity is defined as the volume of the voids that can conduct water flow divided by the total volume. Not all pore spaces are connected to one another, rendering them unavailable to conduct water flow. In addition, if a large precent of the a pore area is occupied by the zero velocity shear layer of water molecules against the soil particle, then the ability for water to flow through that pore is severely limited. Both these conditions and any other condition which may limit flow through a pore will reduce the effective porosity of an aquifer.

For the silt layer, the total porosity value is used. This is an unconservative assumption and may predict plume migration that is slower than reality would dictate. Sensitivity was performed for effective porosity's of 0.20 and 0.30 (Figure 75 to Figure 80). As can be seen in the figures, the plume migration is greatly effected by the lower porosity value (0.20).

In defense of the base case value, groundwater velocities in the silt are also effected by hydraulic conductivity and the head gradient across the silt, thus in calibration, the porosity parameter may be self-correcting in relation to plume migration. In addition, since the assumption of the full effect of the leachate mound being applied to the landfill/silt interface (no head loss across the depth of the landfill) is conservative, the use of the total porosity value can be justified.

Concentration at Top of Gravel Aquifer, 2020
Silt Porosity = 0.30
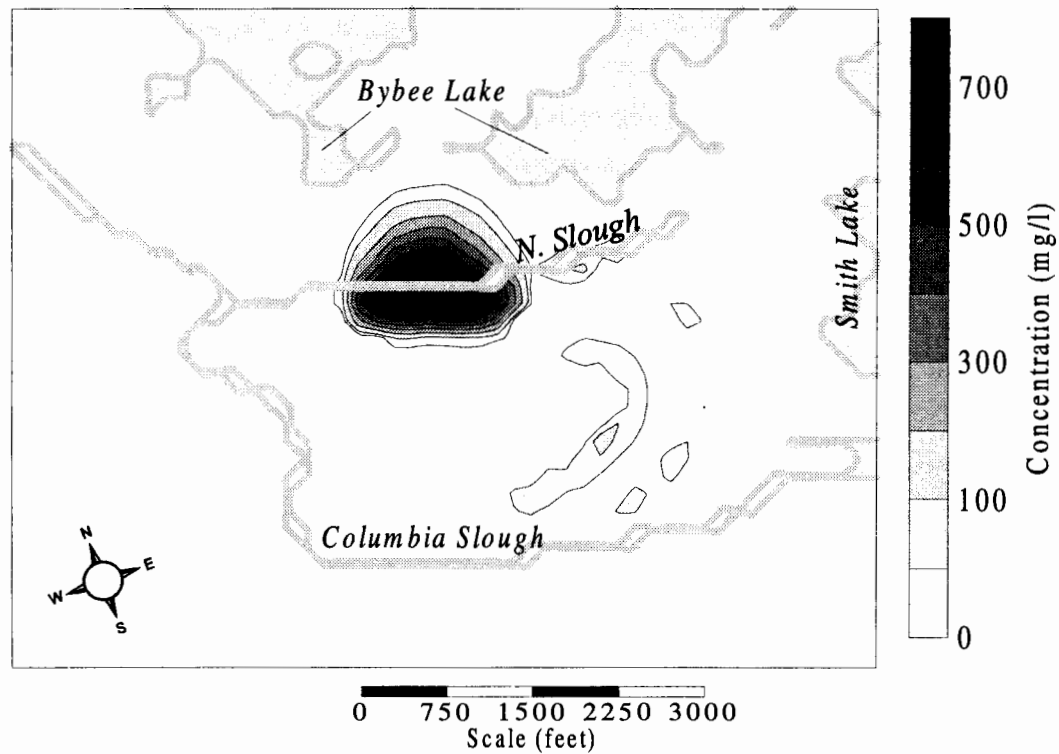
Figure 75 - Sensitivity to porosity. Plan view showing concentration at the top of the gravel aquifer in the year 2020. As porosity decreases, groundwater velocities increase and thus vertical migration of the plume is accellerated. Breakthrough of the plume can be seen in the southeast portion of the landfill.

190

Concentration at Top of Gravel Aquifer, 2020
Silt Porosity = 0.20
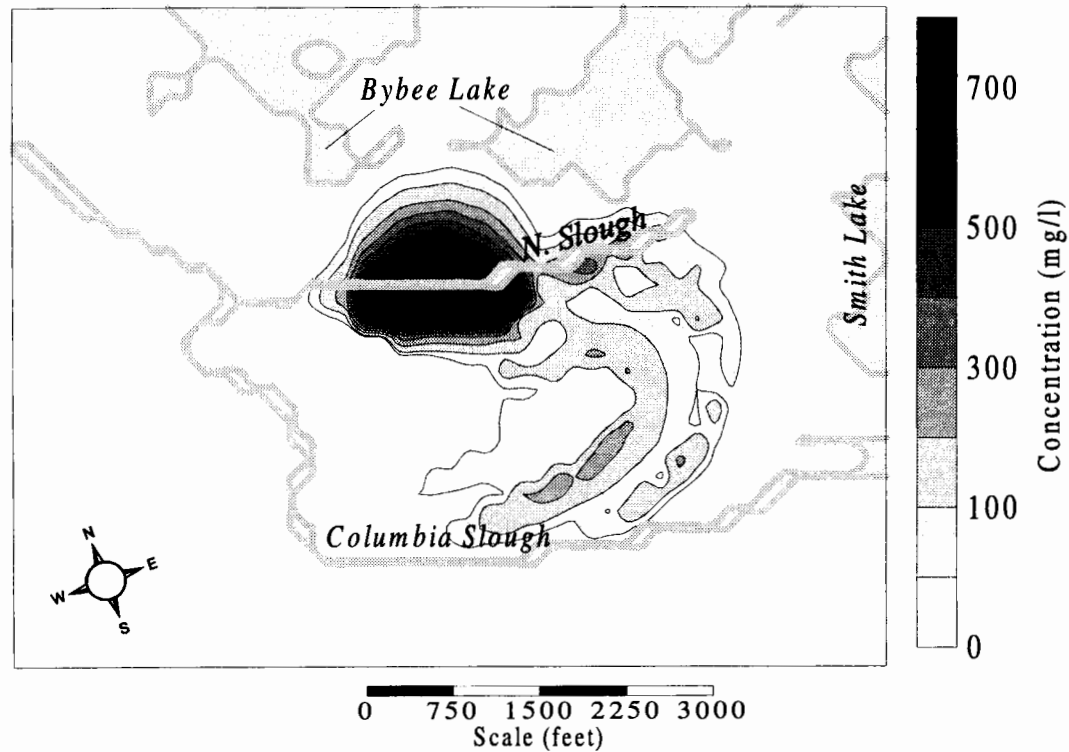
**Figure 76 - Sensitivity to porosity. Plan view showing concentration at the top of the gravel aquifer in the year 2020. As porosity decreases, groundwater velocities increase and thus vertical migration of the plume is accellerated. Breakthrough of the plume is major across large portions of the landfill for lower values of porosity.**

**Figure 77 - Sensitivity to porosity. Cross section of plume showing vertical penetration across the geologic layers in the year 2020. Changes in porostiy are for the silt layer only and do not directly effect migration once the plume enters the sand or gravel layers. See Figure 62 for location of cross section.**

**Figure 78 - Sensitivity to porosity. Cross section of plume showing vertical penetration across the geologic layers in the year 2020. For this low value of porosity in the silt, breakthrough to the sands and gravels can be seen in the southern portion of the landfill. See Figure 62 for location of cross section.**

## Plume Cross Section on Row 29, 2020
## Silt Porosity = 0.30
## Vertically Exaggerated 8x
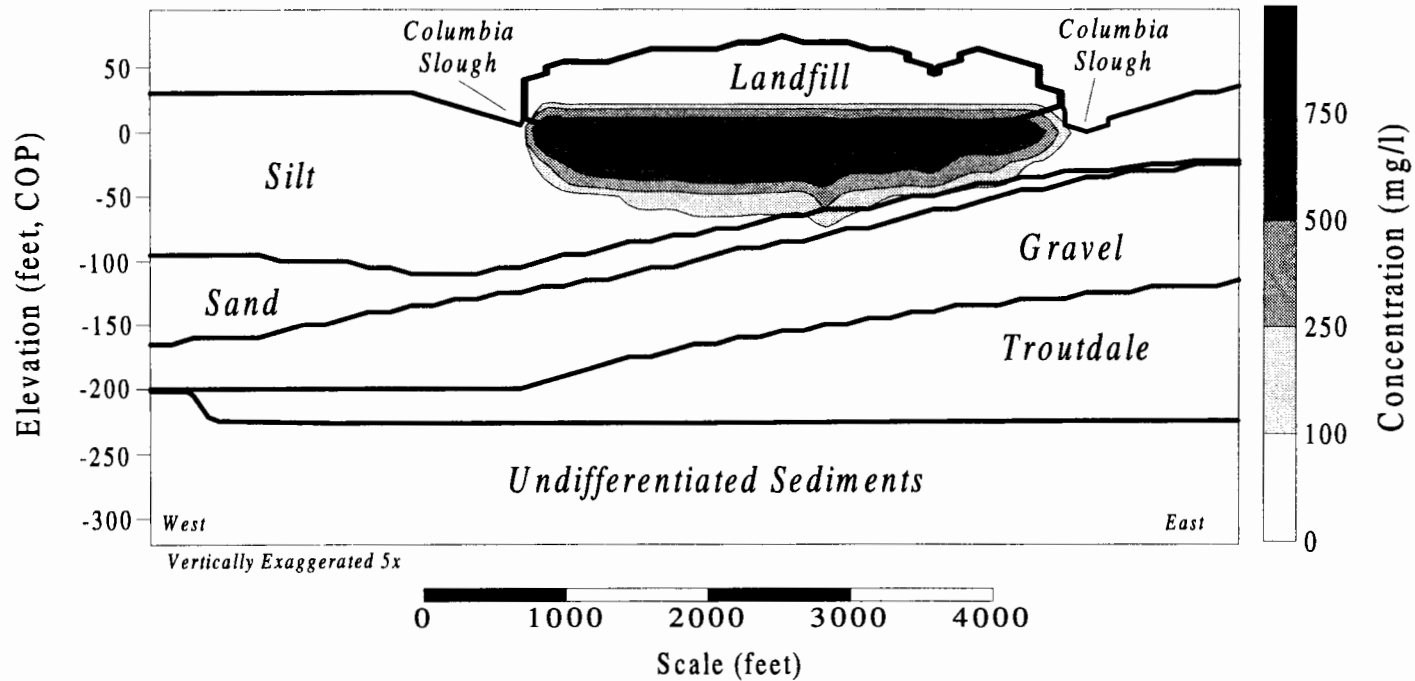
**Figure 79 - Sensitivity to porosity. Cross section of plume showing vertical penetration across the geologic layers in the year 2020. Changes in porostiy are for the silt layer only and do not directly effect migration once the plume enters the sand or gravel layers. Some breakthrough to the sands can be seen in the eastern part of the landfill. See Figure 62 for location of cross section.**

**Plume Cross Section on Row 29, 2020**
**Silt Porosity = 0.20**
**Vertically Exaggerated 8x**
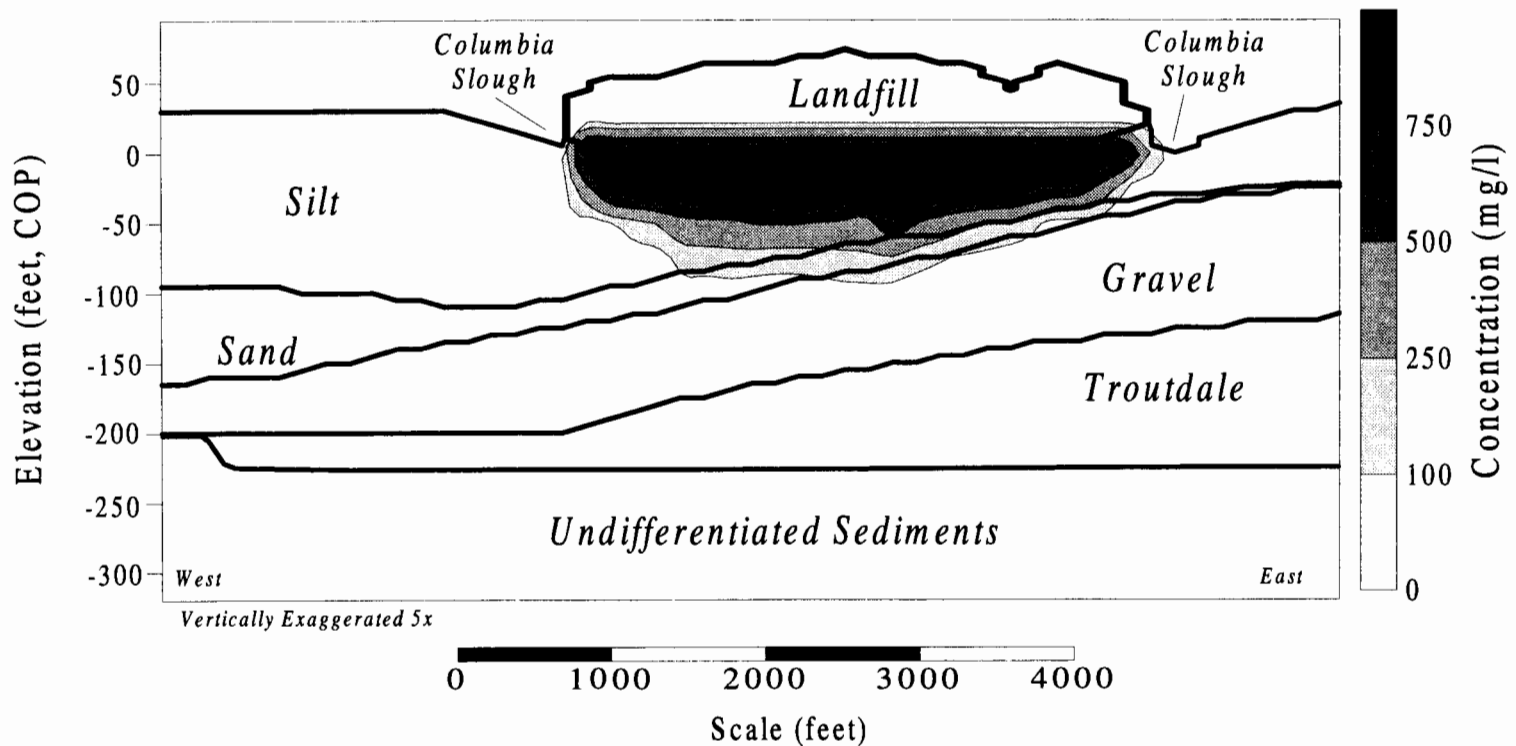
**Figure 80 - Sensitivity to porosity.** Cross section of plume showing vertical penetration across the geologic layers in the year 2020. Almost uniform penetration to the gravels can be seen in this cross section. See Figure 62 for location of cross section.

Dispersivity:

Dispersivity, as explained in the previous section, affects the rate of plume spreading. Figure 81 through Figure 84 show the concentration of the chloride plume at the top of the gravel aquifer with dispersivities that are 5 and 50 times greater than the base case values (base case values are 1 foot in the silt and sand, 5 feet in the gravel and troutdale) in 1990 and 2010. Figure 85 through Figure 88 show a north south cross section of the plume using the same conditions.

Sensitivity analysis shows that as the dispersivity is increased to 50 times the base case values, the extent of the plume is similar to that of lowering the anisotropy to 40, but the maximum concentration in the gravel aquifer is much less due to the dilution effects of dispersion. Increasing dispersivity five times over the base case has a minimal effect on the plume migration. Data do not support the case with higher dispersivities.

Concentration at Top of Gravel Aquifer, 1990
Sensitivity to Dispersivity
$D_{silt} = D_{sand} = 5ft$, $D_{gravel} = D_{trout} = 50ft$
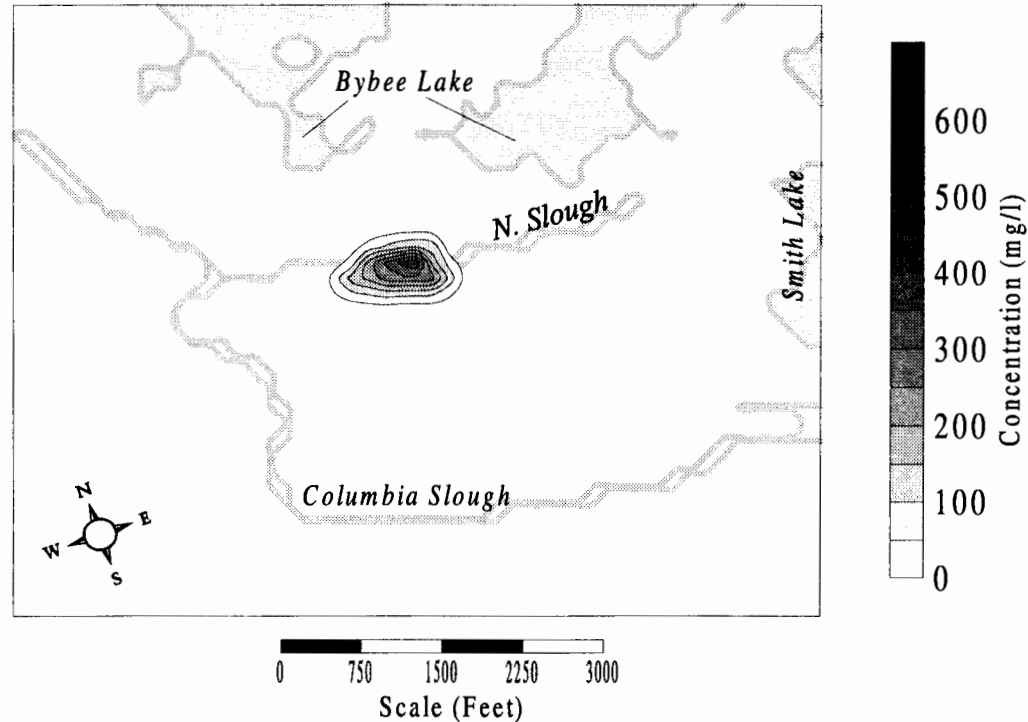
**Figure 81 - Sensitivity to Dispersivity. Concentration of chloride plume at top of gravel in 1990 and its sensitivity to dispersivity in the silt and gravel. Maximum concentration in the gravel is predicted to be 382.087 mg/l (~38% source concentration). Penetration through the silt at this point is limited to the hot spot area on the North Slough.**

Concentration at Top of Gravel Aquifer, 1990
Sensitivity to Dispersivity
$D_{silt} = D_{sand} = 50ft, D_{gravel} = D_{trout} = 250ft$
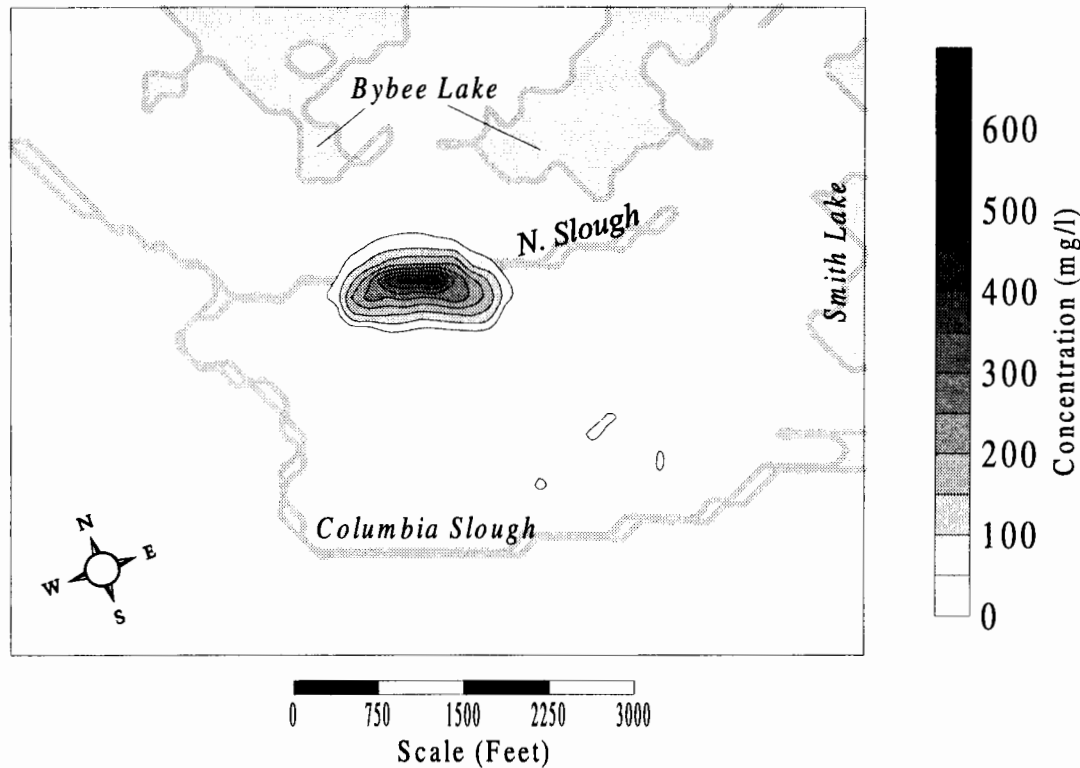
**Figure 82 - Sensitivity to Dispersivity. Concentration of chloride plume at top of gravel in 1990 and its sensitivity to dispersivity in the silt and gravel. Maximum concentration at the top of the gravel is predicted to be 433.998 mg/l (~43% of source concentration). Slight penetration on the south side of the landfill under subarea 5a can be seen in this case. Subarea 5a is the oldest part of the landfill.**

Concentration at Top of Gravel Aquifer, 2010
Sensitivity to Dispersivity
$D\,silt = D\,sand = 5\,ft,\ D\,gravel = D\,trout = 50\,ft$
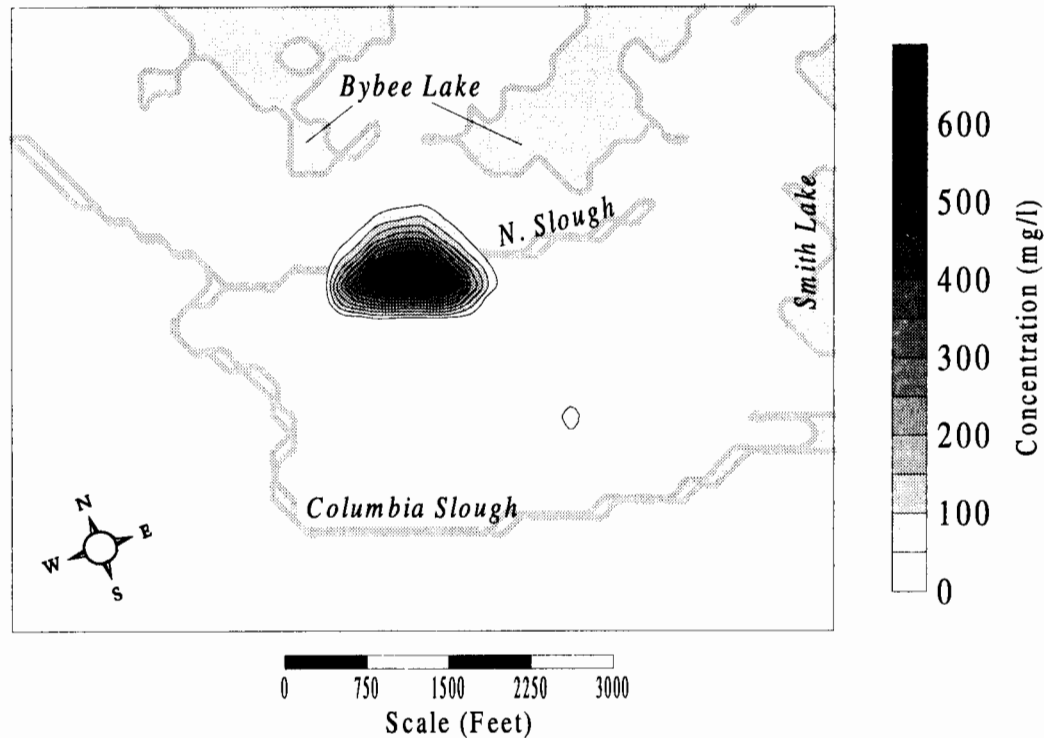
**Figure 83 - Sensitivity to Dispersivity.** Concentration of chloride plume at top of gravel in 2010 and its sensitivity to dispersivity in the silt and gravel. Maximum concentration in the gravel is predicted to be 694.539 mg/l (~69% source concentration). Slight penetration can be seen under subarea 5a although the difference as compared to the base case is slight. Subarea 5a is the oldest part of the landfill.

Concentration at Top of Gravel Aquifer, 2010
Sensitivity to Dispersivity
D silt = D sand = 50ft, D gravel = D trout = 250ft
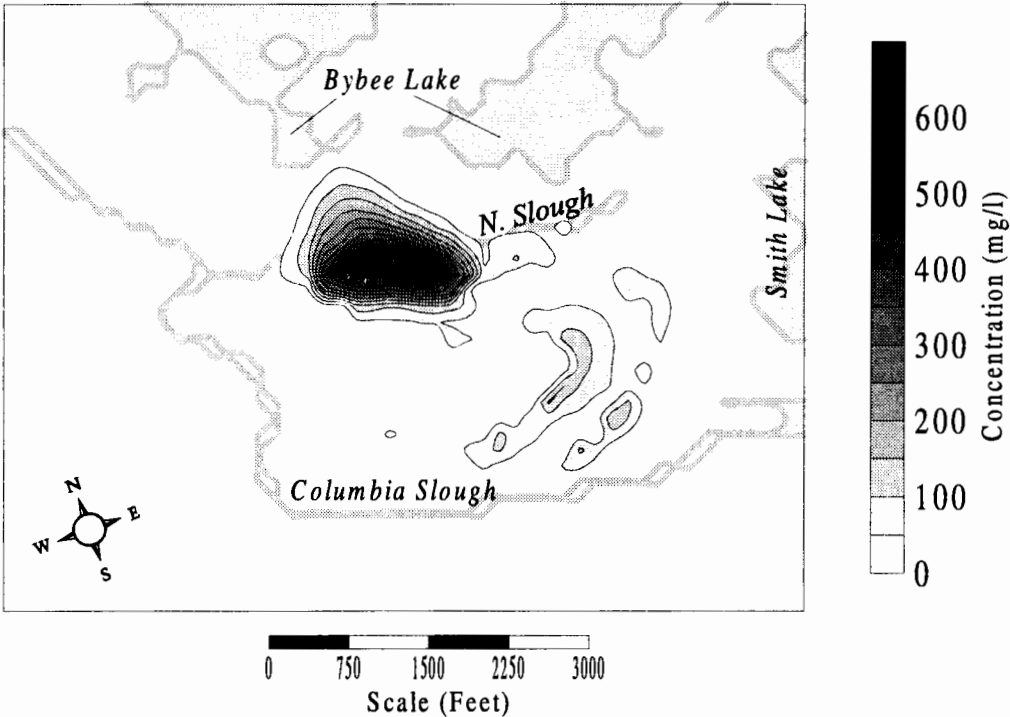
**Figure 84 - Sensitivity to Dispersivity. Concentration of chloride plume at top of gravel in 2010 and its sensitivity to dispersivity in the silt and gravel. Maximum concentration in gravel is predicted to be 643.735 mg/l (~64% source concentration). Extent of plume migration is similar to sensitivity analysis of lower anisotropy however the maximum concentration in the gravel is much less (643.735 mg/l compared to 885.672 mg/l) due to the dilution effects of the higher dispersivity.**

Plume Cross Section on Column 25, 1990
Sensitivity to Dispersivity
Dsilt = Dsand = 5 ft, Dgravel = Dtrout = 50 ft
Vertically Exaggerated 8x
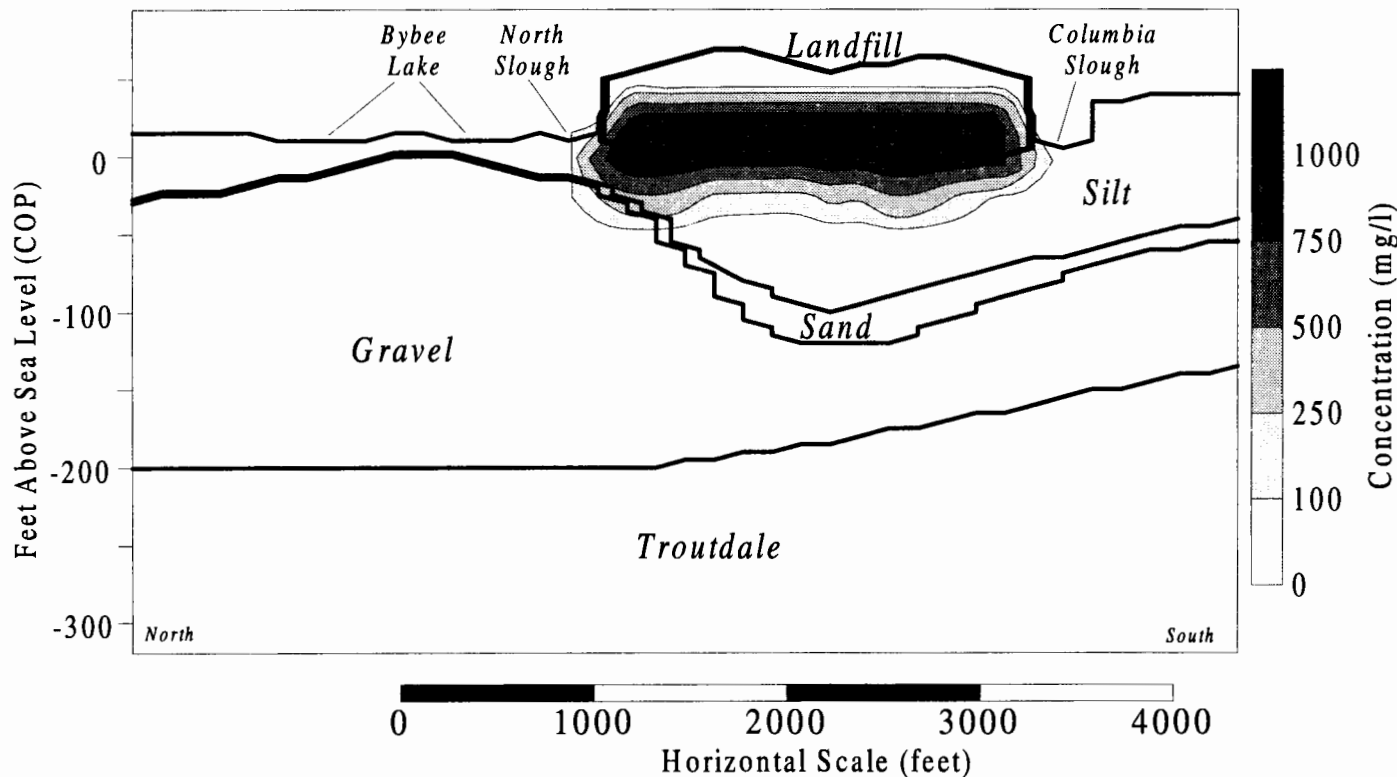
**Figure 85 - Sensitivity to Dispersivity. Plume cross section and its sensitivity to dispersivity. Vertical plume migration is similar to the base case conditions. See Figure 62 for location of cross section.**

**Plume Cross Section on Column 25, 1990**
**Sensitivity to Dispersivity**
**D silt = D sand = 50 ft, D gravel = D trout = 250 ft**
**Vertically Exaggerated 8x**
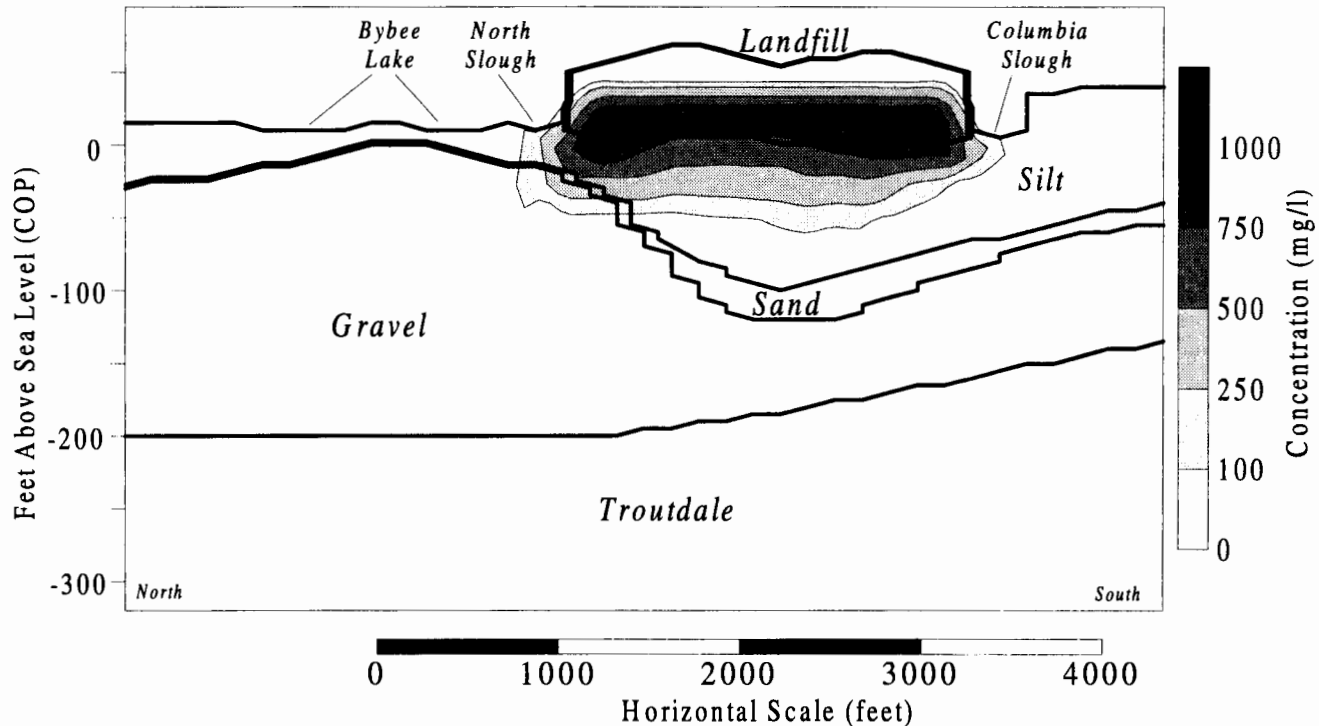
**Figure 86 - Sensitivity to Dispersivity. Plume cross section and its sensitivity to dispersivity. Extensive horizontal plume migration in the silt can be seen when compared to base case conditions. This could have an implication on contaminant migration into the sloughs and lake for long term simulations. See Figure 62 for location of cross section.**

Figure 87 - Sensitivity to Dispersivity. Plume cross section and its sensitivity to dispersivity. Vertical plume migration is similar while horizontal migration in the gravel is slightly greater than the base case scenario. See Figure 62 for location of cross section.

203

Plume Cross Section on Column 25, 2010
Sensitivity to Dispersivity
D silt = D sand = 50 ft, D gravel = D trout = 250 ft
Vertically Exaggerated 8x
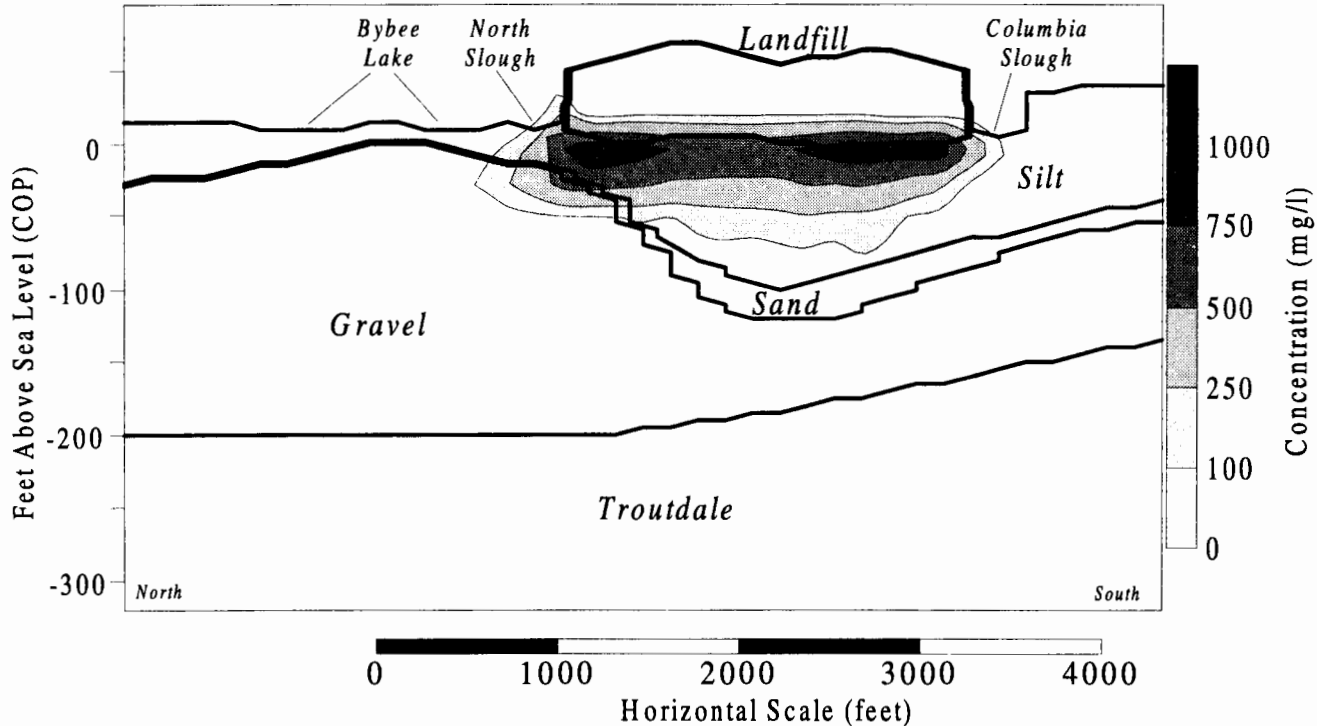
**Figure 88 - Plume cross section and its sensitivity to dispersivity. Extensive horizontal migration in the silt can be seen in this case, extending beyond the center of the sloughs. Areas of high concentration values which are close to the source concentration are much less as compared to lower dispersivities due to the dilution effect dispersion has on the plume over time. See Figure 62 for location of cross section.**

Lake Management Schemes:

Plume migration is insensitive to different management schemes for lake levels. Simulations are done with Bybee Lake closed to the North Slough from 1982 to 2020, the lake being closed from 1982 to 1996, and a hypothetical case of the lake always being open to the slough from 1950 to 2020. In the hypothetical case with Bybee Lake never being closed to the North Slough, there is a very slight difference when compared to the lake staying closed from 1982 on. In all cases, plume migration is virtually insensitive to lake levels.

# 9.    Summary, Conclusions, and Recommendations

The St. Johns Landfill modeling system is specifically designed to address the issues of the project. The modeling system simulates three-dimensional groundwater flow and contaminant transport at the landfill site and its vicinity. The system consists of a regional flow model which predicts the large area regional flow pattern, the water balance model which simulates average contaminant mounding and lateral seepage flux to the sloughs and vertical seepage flux to the gravel aquifer, the particle tracking model which details contaminant movement due to advection close to the landfill, the two-dimensional site specific mounding model which predicts the spatial and temporal distribution of the contaminant mound, the three-dimensional flow model which simulates the three-dimensional flow field in the local landfill area as a function of time, and the contaminant transport model which predicts the three-dimensional concentration field for the contaminant being modeled (chloride).

The groundwater modeling system represents the quantitative inter-relationships among the geology, hydrology, hydrodynamics, physics, and chemistry in the general St. Johns Landfill area. Using a telescopic approach the modeling system is an organized framework for applying large regional effects on the more detailed local scale models. Most importantly, the modeling system provides a tool for the prediction and simulation of both real and hypothetical situations, enabling the direction of limited resources to areas of critical concern for management purposes.

The water balance model is calibrated to reproduce the long term space-time average of the observed mound height. The two-dimensional site specific mounding model is calibrated to the time-averaged measurements of head difference across the landfill. The regional scale flow model is calibrated to reproduce the seasonal fluctuation as determined by the measured hydrographs in the six wells penetrating the gravel during Nov 2, 1988 to Nov 1, 1989. The calibrated flow system parameters are used in the local scale three-dimensional flow model and the contaminant transport model.

Sensitivity analysis is also performed on all parameters, with the more sensitive parameters receiving an extensive systematic analysis, to quantify model uncertainty which provides information for prioritizing further sampling needs.

## 9.1    Answers to Project Questions and Issues

Addressing each issue individually, specific conclusions emerge from this modeling investigation which are summarized below.

*Is the landfill leaching contaminant to the adjacent surface and groundwater systems?*

Based on the analysis provided from the regional flow model and the water balance model, groundwater and surface water interaction between the landfill, the sloughs, the silt layer, the gravel aquifer, and Smith and Bybee Lakes takes place. Contamination first enters the gravel aquifer around 1980 and is subject to the regional flow within the gravel layer migrating northward towards Bybee Lake. Laterally, the dike is predicted to be penetrated in less than three years (for

207

conservative contaminants) after mounding begins due to the higher horizontal permeability in the dike. Reactive chemicals with effective retardation factors of 100 or more may only be able to penetrate the dike through preferential pathways. Conversely, they might not ever penetrate the dike elsewhere because the travel time necessary for penetration is greater than the time for the landfill mound to dissipate. Likewise, reactive chemicals with effective retardation factors of 10 or more may only penetrate to the gravel aquifer via preferential pathways. As long as the contaminant mound is present within the landfill, downward migration into the gravels and lateral migration into the sloughs will continue.

*What is the contribution of the St. Johns Landfill to the already polluted Columbia and North Sloughs?*

The amount of recharge will determine the total amount of contaminant flux out of the landfill while anisotropy controls the distribution of the flux. The calibrated value for the St. Johns Landfill site for anisotropy is 90, meaning that the permeability in the horizontal direction is 90 times greater than the permeability in the vertical direction. This amounts to a total flux from the landfill into the sloughs that is approximately three times greater than the total flux from the landfill into the silts. Assuming a recharge rate of 10 inches per year, anisotropy's of 40, 90 (base case), and 200, and an average contaminant mound of 25 feet (COP) prior to closure in 1985, the flux to the sloughs is 0.147 cfs (95,000 gallons per day), 0.189 cfs (122,145 gallons per day), and 0.216 cfs (140,000 gallons per day), respectively. After closure, the rate of decrease in the flux is

dependent upon the specific yield, which controls the rate of mound dissipation. Assuming a range for the specific yield of 0.2 to 0.4, and the base case for anisotropy of 90, the flux into the slough at partial closure in 1995 is 0.145 cfs (93,700 gallons per day) to 0.161 cfs (104,000 gallons per day). At total closure in 2000, the flux is reduced to a range of 0.057 cfs (36,800 gallons per day) to 0.090 (58,200 gallons per day) cfs and by the year 2020, the flux is 0.021 cfs (13,600 gallons per day) to 0.026 cfs (16,800 gallons per day).

As a single source of contamination into the sloughs, the landfill is significant, providing about 45 million gallons (base case prior to closing) of contaminated water into the sloughs annually. Compared to other contaminant sources, such as combined sewer overflows (CSO's) during storm events, the landfills' contribution is fairly minor. The lower Columbia Slough maintains an average net flow (with severe tidal fluctuations) of about 100 cfs - 200 cfs (64,600,000 - 129,200,000 gallons per day; S. Wells, 1995) meaning that the landfills contribution represents about one tenth of one percent of the total flow in the slough. The significance of this contribution on the water quality is beyond the scope of this project and has not been assessed. The North Slough flow is much less than the Columbia Slough meaning the impact on water quality could be much greater.

*How effective is the silt at protecting the deep aquifer system from contaminant migration?*

The silts underlying the landfill provide good protection to the deep aquifer system. As mentioned above, anisotropy controls the distribution of fluxes out of the landfill. Approximately one-third of the amount of contaminant flux that enters the sloughs enters the silts, with a small portion of that reaching the gravels. The thickness of the silt layer as well as the high horizontal permeability in the silt as compared to the vertical permeability in the silt are the reasons for the significance of the protection. For the base case through 2020, except for the isolated hot spot underneath the western third of the North Slough and a slight penetration in the southeastern corner of the landfill, the silt appears to protect the gravel aquifer from receiving contamination from the landfill; the silt acting as storage for the contaminant. The landfill site fortunately is placed over a deep trough in the gravel aquifer where the silt layer varies in thickness from 30 to 170 feet. Contaminant penetration through the silt only shows up in areas underneath the original landfill (non-expansion area) where the silt is less than 40 feet thick. Like the flux into the sloughs, once the contaminant mound dissipates to post closure levels, the driving force for the contaminant to migrate into the gravel aquifer is greatly reduced.

*How long does it take contaminant to penetrate the bottom silt layer?*

Based on the conceptual model, penetration through the silt depends upon the head differential between the landfill and the gravel aquifer, the vertical hydraulic conductivity in the silt, the porosity of the silt, and the thickness of the silt. The combination of all these factors leads to slow contaminant migration. In only the thinnest areas of the silt that underlie the older parts of the landfill do we see contaminant penetration through the silt. For the simulation time from 1950 to 2020, about 35 years is needed (1985) for contaminant to be first seen at the top of the gravel aquifer.

*Has offsite migration occurred?*

Once the plume enters the gravel aquifer, regional flow moves the plume northward. The simulation predicts that by the year 2020, the plume will have migrated approximately 300 - 400 feet north of the north edge of the landfill. It should be noted that dispersion in the gravel aquifer is uncalibrated because of a lack of data that defines the front edge of the plume. Dispersivity values which are much higher than the 5 foot base case value for the gravel could significantly add to the dispersion and spreading of the plume.

*How much contaminant goes to the groundwater?*

Assuming recharge of 10 inches per year, anisotropy's of 40, 90 (base case), and 200, and an average contaminant mound of 25 feet, the seepage flux

downward to the silts prior to closure (1985) based on the model predictions, is

0.104 cfs (67,212 gallons per day), 0.059 cfs (38,130 gallons per day), and 0.031

cfs (20,000 gallons per day), respectively.  Assuming a contaminant concentration

of 1000 mg/l, the seepage flux values correspond to a contaminant flux of 560,

318, and 166 lbs/day, respectively.  In 1995 at partial closure, the seepage flux

drops to 0.084 cfs (54,290 gallons per day), 0.048 cfs (31,000 gallons per day),

and to 0.025 cfs (16,150 gallons per day), corresponding to a contaminant flux of

453, 258, and 134 lbs/day under the same assumption as above (1000 mg/l

concentration).  After closure seepage and contaminant flux decrease over time as

the contaminant mound dissipates.  The rate of the flux decline is dependent on the

value of the specific yield in the landfill.  Assuming specific yields of 0.2 and 0.4,

and an anisotropy of 90, by the year 2000 the vertical seepage flux to the gravel

aquifer will reduce to 0.020 cfs (12,900 gallons per day) to 0.032 cfs (20,700

gallons per day) and by the year 2020, the seepage flux is 0.004 cfs (2,600 gallons

per day) to 0.007 cfs (4,500 gallons per day).  Again, assuming a contaminant

concentration of 1000 mg/l, the contaminant flux is 107, 172, 21, and 37 lbs/day

respectively.

*Is the landfill affecting Smith and Bybee Lakes?*

Based on model predictions, the contaminant plume has not been present

long enough in the gravel aquifer to migrate under and into the lakes.  Smith Lake

is separated from the landfill at high water by the nearly impermeable engineered

dike, so any lateral contaminant migration from the landfill to Smith Lake is from

212

underneath the engineered dike. Due to the relatively high elevation of the bottom of the engineered dike next to Smith Lake (approximately 11 to 17 ft COP), there may be a significant connection between the landfill and Smith Lake through the silt below the engineered dike (Li and Lowry, 1997). Consideration should be given to the long-term potential of contaminant migration through the gravel aquifer and into Bybee Lake through the 'window' area underneath the lake where the silt is only a few feet thick. Within the time boundaries of this modeling system, and under current management practices concerning the Bybee Lake/North Slough connection, the landfill has not impacted Smith or Bybee Lakes.

*How will the proposed management plan of restoring Bybee Lake to intertidal conditions effect contaminant plume migration in the aquifer?*

The models predict interaction between the Smith and Bybee Lakes and the gravel aquifer to be weak due to the small vertical conductivity associated with the fine-grained silt separating the two systems. Under present conditions, Bybee Lake is isolated from the North Slough by a weir. The weir removes daily tidal fluctuations in lake levels. In the calibration year of 1989, the lakes recharge the aquifer at an annual mean rate of 0.08 cfs (51,700 gallons per day) representing only about 1% of the total regional system inflow. The lake flux is of the same order of magnitude as the flux to the gravel from the landfill. All inflows into the gravel aquifer system are eventually drained to the Columbia or Willamette Rivers.

If Bybee Lake is restored to its intertidal condition, the head gradient is reversed, but very small. The annual mean upwelling flux for the year 1989 is predicted as 0.04 cfs (25,800 gallons per day). Seasonally, flux rates may be much greater than described above, as river levels, aquifer levels, and lake levels rise and fall. Regardless, lake dynamics are predicted to have little or no influence on the plume migration within the gravel aquifer.

## 9.2 *Summary*

The St. Johns Landfill modeling system allows for a detailed look at the interactions and dynamics between the landfill, the sloughs, the gravel aquifer, Smith and Bybee Lake, and the Columbia and Willamette Rivers. The system is not without limitations although the predictions are able to reasonably and defensibly reproduce the area dynamics.

At the start of the model simulation (1950), the contaminant mound in the landfill begins to grow due to infiltration into the landfill from rain water (recharge). This mounding continues until about 1965, when the mound is sufficiently high to produce a driving force which results in outward fluxes that are equal to the recharge. This state continues until about 1990 when recharge is systematically reduced due to landfill capping and the mound begins to dissipate. Contaminant migration out of the landfill is controlled by the hydraulic conductivity in the silt. Mounding dynamics and site geology limit the range of horizontal and vertical conductivities in the silt, which are able to reproduce field conditions at the landfill. Anisotropy controls the distribution of horizontal flux to vertical flux within the bounds of suitable

conductivities. For base case conditions in 1985, the model predicts that seepage flux laterally out towards the sloughs is about three times the seepage flux vertically into the silts (0.189 cfs through the dikes compared to 0.059 cfs through the silt).

The model predicts that contaminant flux migration through the silts and into the gravels takes approximately 35 years, with the first contaminant showing in the gravel in about 1985. The breakthrough area is a localized spot below the north slough about one-third along its length from the mouth. This is due to a thin area in the protective silt layer in that area. Reactive contaminants with an effective retardation factor greater than 10, may never penetrate the silt since the time for penetration is longer than the time for the contaminant mound to dissipate. Likewise, for reactive contaminants with an effective retardation factor greater than 100, penetration laterally through the dike may be impossible. Preferential pathways may allow contaminants with high retardation factors to penetrate both the dike and silt in comparable short time periods. Once contaminant has entered the gravel, the influence of the landfill is negligible. Regional flow in the gravel aquifer, caused by the aquifers interaction with the Willamette and Columbia Rivers, carry the contamination off site. Regional flow in the vicinity of the landfill is in the northerly direction towards Bybee Lake. Within the time span of the modeling project (1950-2020) the lake is not predicted to interact with the contaminant plume. Regional flow moves the contaminant approximately 300 feet north by the year 2020, or about one-third of the way towards the lake. Currently the landfill closure is nearing completion and the contaminant mound is dissipating. Complete dissipation to post-closure levels

should take place in 10-20 years (2010) at which point any mound induced migration of the plume will be greatly reduced.

### *9.3    Recommendations*

Future study should concentrate on the hot spot area on the north side of the landfill. This is clearly of great concern because is appears to be the only place where contaminant has penetrated the silt and reached the gravel aquifer. Two types of studies should occur. First is focused monitoring around the hot spot area. Added data from this area will help verify the results of the current model and allow for refined calibration to provide more accurate predictions. Secondly, focused modeling should occur that would refine the resolution of the model, especially around the hot spot area. Part of this additional work should include a comprehensive analysis of the existing data based on the new understanding of the system brought about by this project.

### 9.3.1  Focused Monitoring:

The current monitoring network at the site is too sparse in the area of critical concern around the hot spot area. The plume has been predicted to be localized and fairly narrow when compared to the existing well placements and in order to refine the results of this modeling investigation, closer study of the current state of the plume is needed. To do this, we recommend:

Conduct focused monitoring to the north of the North Slough immediately downgradient from the hot spot. Preferable treatment would be to

use a series of nested wells that would return head level and water quality data from several different points at several different elevations below ground. Drilling and sampling should be carried out sequentially so that feedback from the earlier installed wells can be used to place the later wells.

Conduct monitoring of water levels in the North Slough, the silt layer, the sand layer, and the gravel layer in the area immediately downgradient from the hot spot to refine the head gradient and rate of contaminant migration offsite.

Conduct a thorough survey and review of the site conditions, especially in the area of the hot spot and downgradient. This would include surveying the dike condition, dike width, local topography with position of surface water bodies, top of silt, sand, and gravel layer elevations, and concentration data of both conservative and reactive contaminants in the hot spot area.

## 9.3.2  Focused Modeling:

While the current model is robust and defensible, focused modeling coupled with the focused monitoring recommendations outlined above, will allow for better site characterization as well as more accurate future predictions. Recommendations for focused modeling are:

Need to improve model resolution.

The current model predicts a plume that is mostly localized within the silts and gravels on the north side of the landfill. This suggests that we should

dramatically reduce the model domain and conduct focused modeling in the vicinity of the hot spot, where the contamination penetrated the silt into the gravel. One possible way to address this is to model only the silt layer, sand layer, and gravel aquifer with only the portion of the landfill which contributes to the contaminant penetration through the silt as the source of contamination and simulate flow and transport in an area approximately 1000 feet wide (east/west) by 2500 long (north/south). This would greatly improve the model resolution and simulation accuracy both in time and space, reduce numerical smearing of the predicted plume fronts due to numerical dispersion, and provide a detailed description of the flow paths and plume structure in the silt, sand, and gravel aquifer. A higher resolution model such as this could be added to the current modeling system, further refining the telescopic approach while still taking advantage of the regional scale influences.

Need to simulate density effect.

The minimal contaminant penetration through the silt into the gravel suggests that a significant amount of the contamination remains within the silt layer below the landfill. As this 'stored' contaminant enters the gravel due to dispersion and molecular diffustion, density effects may play a large role in plume migration within the higher conductivity gravel layer. Total dissolved solids (TDS) concentrations in wells G7 and D8a, located close to the hot spot, are 1600-2300 mg/l and 1300-2000 mg/l respectively. This indicates a

contaminant density that is 0.10% to 0.15% higher than that of the unpolluted

groundwater.

Density differences can amount to a significant increase in groundwater

velocity in the vertical direction. Equation (27) (Bear, 1972) shows a form of the

Darcy equation which describes the additional effects of density induced vertical

velocity of groundwater. Using this equation, it is possible to calculate an order

of magnitude estimation of the impact a 0.10% to 0.15% density difference

would have on the vertical velocity of contaminant in the gravels below the

landfill.

$$v = \frac{1}{n} K_v \frac{d\phi}{dz} + \frac{1}{n} K_v \frac{\Delta\rho}{\rho_o}$$

(27)

where:

| | |
|---|---|
| v | = Vertical velocity of the groundwater. |
| $K_v$ | = Vertical hydraulic conductivity. |
| n | = Effective porosity. |
| $d\phi/dz$ | = Vertical hydraulic gradient over distance z. |
| $\rho_o$ | = Background density of the unpolluted water. |
| $\Delta\rho$ | = Density difference between the contaminated groundwater and the unpolluted water. |

Assuming $K_v$ in the gravel is 18 ft/day ($6.35\times10^{-3}$ cm/sec - calibrated

value from the regional flow model assuming an anisotropy of 10, which is also

an order of magnitude estimate based on the local flow model), n=.25 (typical

value), $\frac{d\phi}{dz}$ =0.0001 (order of magnitude estimate based on the local three-

dimensional model and data), and $\dfrac{\Delta\rho}{\rho_o}$ =0.001 to 0.0015 (based on the well G7

and D8a data) we get a vertical velocity due to the density effect of 0.072 to 0.11

ft/day. This is about 100 times faster than the vertical flow due to head

differences within the gravel and is comparable in magnitude to the horizontal

velocity. A vertical downward velocity equal to the horizontal velocity would

indicate a contaminant plume which is angling downward at a $45^{o}$ angle away

from the landfill. This indicates that the impact of the density effect may

actually be controlling the movement of the contamination and causing the

plume to sink. This sinking of the plume may be why well G4b (approximately

300 ft downgradient of the hot spot screened in the top of the gravel) shows no

indication of contamination since the contamination may be moving below the

screen depth of the well.

The density effect should be coupled with the refined resolution model

mentioned above. Multiple contaminant indicators that effect density, such as

TDS (electrical conductivity) and temperature, need to be modeled to help define

the extent of the plume. The improved flow and transport model would enable

us to quantify the plumes sinking effect due to density differences, define more

accurately vertical plume positioning and structure, and provide information for

accurate vertical sampling. Using multiple indicators also allows the use of

available data measurements of temperature, total dissolved solids, and electrical

conductivity in addition to the chloride data for calibrating the flow and contaminant plumes as well as validates the flow and transport models.

Need to improve the estimate of silt conductivities and silt anisotropy.

Sensitivity analysis clearly shows that silt conductivity is the most important parameter when it comes to assessing the environmental impact of the landfill. The anisotropy controls the flux dynamics of contaminant migration out of the landfill dictating the distribution of fluxes between the sloughs and the groundwater. Unfortunately, silt conductivities and anisotropy are the most uncertain parameter to accurately estimate on a field scale. Local measurements are rarely an indicator of field scale conductivities with the anisotropy and conductivity values varying over a wide range across the entire site. Uncertainty in the conductivity values and the anisotropy translates into uncertainty in the models ability to predict the impact of the landfill on the adjacent sloughs and the underlying groundwater system.

Calibration of the silt conductivity and anisotropy has been difficult with the current model because there exist very few data from the interior of the landfill that describes the extent of the plume. Calibration to the measured concentrations observed along the perimeter of the landfill is inaccurate since they are mostly affected by source concentration near the landfill boundary as well as being insensitive to anisotropy. Meaningful model-data comparisons are not possible unless we know the actual geometry of the irregular dike and how close to the dike the solid wastes were deposited.

Silt anisotropy can be calibrated to the observed hydrographs in the shallow silt near the sloughs and dike. The two components of the silt conductivities will dictate the dynamic slough and shallow groundwater interaction and controls the timing (phase lag) and amplitude of the head fluctuation in response to water level changes in the slough. Calibration of this sort would require the construction of a very fine resolution model to simulate the unsteady, small-scale flow patterns and dynamics in the shallow groundwater area. Due to the relatively small width of the sloughs (50-100 feet for the half width), the small width of the dike (20-50 feet), and the dynamic nature of the slough condition, (the dominant periods of slough level fluctuations are 12 hours, 24 hours, and most importantly, two weeks), to accurately resolve the complex space-time head distribution the model resolution would need to be on the order of 10-20 feet in the horizontal direction and 3-5 feet in the vertical direction, with the time step being a fraction of the smallest dynamic period. Modeling of this detail is unrealistic computationally unless the modeling domain is greatly reduced. Spatial details (site conditions and geology) would need to be described at a compatible spatial resolution to allow for a meaningful model-data comparison.

The simplest way to achieve further calibration of silt anisotropy is to use the proposed three-dimensional, high resolution flow model (recommendations 1 and 2) based on a refined site description at a consistent spatial scale (see recommendations 2 and 3 on focused monitoring), in conjunction with a few

high-resolution two-dimensional vertical profile flow models at selected locations where there are concentrations of data. This approach would allow for the full use of the available data in the calibration process. Specifically the data that can be used for calibration is observed heads, TDS, chloride, and temperature in the perimeter wells as a function of space and time. This multi-parameter calibration will greatly reduce the model uncertainty and allow for more accurate predictions of the landfills' impact on the surrounding surface and groundwater systems.

Need to refine the spatial distribution of regional recharge.

Sensitivity analysis shows that the regional flow pattern is highly effected by the spatial distribution of the regional recharge. Recharge values are based on percentage of impervious area based on the crude estimate reported in Snyder et al. (1994) for the overall Portland basin at a grid resolution of 3000 by 3000 feet.

A refinement of the regional recharge distribution in the St. Johns Landfill region based upon the percentage of impervious area is needed to more accurately capture the overall regional flow. Reviewing aerial photographs followed by a confirming field survey in the regional study area would accomplish this. The boundaries for the refinement would include the area bounded by the Willamette and Columbia Rivers to the north, west and south, and the North Portland Road to the east (same boundaries as the regional flow model) at a resolution of 450 by 450 feet (same as regional flow model). The

223

improved estimate of impervious area can then be used to re-calculate the regional recharge using the regression formula developed by Bauer and Vaccaro (section 5.4). The improved spatial distribution of the regional recharge can then be used to re-run the regional flow model to provide a more accurate prediction of the regional flow pattern and the offsite migration of the contaminant plume in the St. Johns Landfill area.

Need to improve the estimate of the sand conductivity.

Changes in the sand conductivity within its realistic range of values, shows large changes in heads in the gravel aquifer and sand layer to the northwest of the landfill near the confluence of the Willamette and Columbia Rivers. Due to the lack of sand in the local landfill area, the same range of sand conductivity's produce very little change in head values in the region close to the landfill where the calibration data are available. This makes calibration of the hydraulic conductivity in the sand very difficult. Data need to be collected in the region to the northwest of the landfill where more sands are deposited to better calibrate this value. The possible changes in head values to the northwest of the landfill helps dictate the overall regional flow direction and may effect long-term plume migration in the gravels.

Need to confirm the geologic characterization to the west of the landfill.

As mentioned above, site characterization and regional geology has a large impact in the prediction of the regional flow pattern. The landfill is sitting on top of a steeply sloping trough in the gravel (Figure 4) that is filled with low

permeability silt. To the west of the landfill, towards the deepening end of the trough, the silt becomes very thick (200-300 feet). This large, low permeability formation causes a splitting of the regional flow and a flat gradient in the area near the landfill. The recent review by EMCON (1995) of the regional stratigraphy supplied by METRO (1994) indicates that, although the shallow portions of the overbank deposits to the west of the landfill contain silts similar to those found in the landfill region, the deeper portions of these deposits are described as sandy-silts and contain higher portions of fine and medium grained sand. It may not be appropriate to lump this lower portion of the formation together with the finer overbank deposits laid on top. The effect of this would be to characterize the trough area to the west of the landfill as being of lower permeability than is actually the case, creating a 'road block' to regional flow when there should be no road block at all (or at least less of a road block). Future work should include careful characterization and delineation of the regional geology to the west of the landfill, as well as direct groundwater monitoring in the sands and gravels in this area. Water level measurements to the west of the landfill can be used to back-calculate the hydraulic conductivities in the silts and sands.

# References:

M.P. Anderson and H.F.Wang, 1982, Introduction to Groundwater Modeling, Finite Difference and Finite Element Methods, W.H. Freeman and Company, N.Y.

M.P.Anderson and W.W.Woessner, 1991, Applied Groundwater Modeling, Academic Press.

H.H. Bauer and J. J. Vaccaro, 1989, Documentation of a Deep Percolation Model for Estimating Ground-Water Recharge: USGS Open-File Report 86-536.

Bear, Dynamics of Fluids in Porous Media. Academic Press, New York, American Elsevier.

Ecology and Environment, Inc., 1986, St. Johns Landfill National Dioxin Study, Portland, Oregon. Final Report prepared for US Environmental Protection Agency.

EMCON, 1995, A Letter Review of METRO Report "Stratigraphy of Rivergate and Industrial Districts of North Portland, Oregon"

J.E. Luzier and D.G. Livermore, 1986, Smith and Bybee Lakes Environmental Studies, Technical Appendix B - Groundwater Hydrology, 1986. Report prey pared by J.E. Luzier and D.G. Livermore for Port of Portland, Planning and Development Department, and City of Portland, Bureau of Environmental Services, Portland, Oregon. Project managed by Fishman Environmental Services, Portland, Oregon

W. Fish, 1994, Characterization of Subsoil Affecting the Transport of Toxic Metals, Ammonia and Hydrophobic Organic Compounds at the St. Johns Landfill Site. Final report prepared for METRO, Portland, Oregon.

A. Freeze and J. Cherry, 1979, Groundwater, Prentice-Hall, N.J.

L.W. Gelhar, 1993, Stochastic Subsurface Hydrology, Prentice-Hall.

M.G. McDonald and A.W. Harbaugh, 1988, A Modular Three Dimensional Finite-Difference Groundwater Flow Model. USGS Water Resources Investigations, book 6, Chapter Al.

METRO, 1994, Stratigraphy of Rivergate and Industrial Districts of North Portland, Oregon, prepared by Solid Waste Department.

METRO, 1995, Controlling Seepage from St. Johns Landfill to Surrounding Surface Water. Report prepared by Solid Waste Department for the Oregon Department of Environmental Quality.

I.S. Oweis, D.A. Smith, R.B. Ellwood, and D.S. Greene, 1990, Hydraulic Characteristics of Municipal Refuse, Journal of Geotechnical Engineering, Vol. 116, No. 4, April.

L. Peyton and P.R. Schroeder, 1988. Field Verification of HELP model for landfills. Journal of Environmental Engineering Division, ASCE, 114(EE-2), 247-269.

P.R. Schroeder and A.C. Gibson, and M.D. Smolen, 1984. The Hydrologic Evaluation of Landfill Performance (HELP) Model, Vol II, Doc. for Version 1, EPA/530-SW-84-101, US EPA, Cincinnati, Ohio.

K. Schock, 1993, Predicting Lateral and Vertical Seepage from St. Johns Landfill into Columbia Slough, Master's Thesis, Portland State University.

S.G. Li, T.S. Lowry, M. Chen, 1995, St. Johns Landfill Groundwater Modeling System: Predicting Leachate Mounding, Fluxes, and Offiste Migration, Portland State University technical report EWR-9-95. Final report prepared for METRO.

S.G. Li, T.S. Lowry, 1997, 2D Vertical Profile Model Using SEEP2D, St. Johns Landfill, Portland, OR, Portland State University technical report EWR-0-00. Final report prepared for METRO.

S.G. Li, T.S. Lowry, Q. Huang, W. Xu, 1997, St. Johns Landfill Modeling System: Sensitivity Simulations and Response to ECMON Review Comments, Portland State University. Final report prepared for METRO.

G. de Marsily, 1986, Quantitative Hydrogeology - Groundwater Hydrology for Engineers, Academic Press

D.T. Snyder, D.S. Morgan, and T.S. McGrath, 1994, Estimation of GroundWater Recharge From Precipitation, Runoff into Drywells, and On-Site Waste Disposal Systems in the Portland Basin, Oregon and Washington. USGS Water Resources Investigations Report 92-4010.

R.D. Swanson, W.D. McFarland, J.B. Gonthier, and J.M. Wilkinson, 1993, A Description of Hydrogeologic Units in the Portland Basin, Oregon and Washington. USGS Water Resources Investigations Report 92-4196.

Stevens, Thompson and Runyan, Inc., 1972, Portland Sanitary Landfill Hydrogeological Studies. Final report prepared for Bureau of Refuse Disposal, Office of City Engineer.

Stevens, Thompson and Runyan, Inc., 1974, Leachate Monitoring for the St. Johns Landfill. Final report prepared for Bureau of Refuse Disposal, Office of City Engineer.

Sweet-Edwards and Associates, 1983, St. Johns Sanitary Landfill - Water Quality Study. Report prepared for METRO, Portland, Oregon.

Sweet-Edwards/EMCON, Inc., 1989, St. Johns Landfill Water Quality Impact Investigation and Environmental Management Options, Volume I-IV and letter report. Report prepared for METRO. Portland, Oregon.

C.Zheng, 1990, MT3D, a Modular Three Dimensional Transport Model, S.S. Papadopulos & Assoc., Rockville, MD.

## Appendix A - Water Balance and Particle Tracking Model Source Code and Documentation

The water balance model and the particle tracking model are, each designed to answer specific questions about the site. Because the code setup and structure are very similar, they will both be addressed as one unit. The water balance model performs the flux calculations. Written in FORTRAN 77 and compiled for use on UNIX machines, it answers the question of how much contamination seeps vertically downward into the gravels, laterally into the sloughs, and outward through the top of the landfill as surface seepage. It also produces output showing the height of the contaminant mound at each time step. The particle tracking model, also written in FORTRAN 77 and compiled for use on UNIX machines, performs particle tracking, answering the question of how long it takes for contaminate to migrate horizontally through the dike and into the sloughs, as well as downward through the silt and into the gravel aquifer.

### *Water Balance Model:*

The water balance model is made up of several programs designed to be used in succession, with the last program producing the useful output. It consists of a preprocessor (called PTIME.FOR), a flux calculation program (called TDIM.FOR), and a version of the flux calculation program (called TDIMDS.FOR) for use in the sensitivity analysis. PTIME.FOR reads a user defined data file (TIME.NPT) and outputs the data to two separate files, one for constant parameters that don't change over time (such as dike length, silt thickness, etc.), and the other for temporally

varying parameters like recharge, slough level, etc. TDIM.FOR reads the two output

files from PTIME.FOR and calculates the various fluxes out of the landfill, outputting

each flux at the end of each computational time step. TDIMDS.FOR is used to

perform sensitivity analysis on the anisotropy. It allows for sequential steps through

various values of $K_{siltH}$ and $K_{siltV}$, the ranges which are defined by the user in the input

file, and produces a continuous output showing the fluxes and contaminant mounding

at all possible combinations of $K_{siltH}$ and $K_{siltV}$. The output from TDIMDS.FOR is

how the anisotropy contours are created. See Table 5 for a complete listing of the

input and output files for each program. The flow chart for the flux calculation group

is as follows:
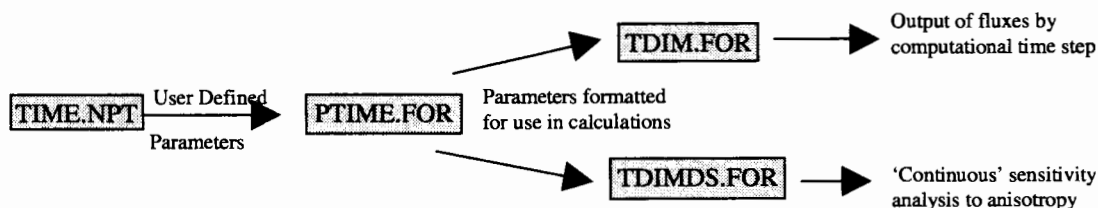


**Figure 89 - Components and flow chart for water balance model. The use of a data file and preprocessor allows for easy changing of parameters without having to recompile the source code.**

TDIM.FOR outputs data for every computational time step, which is specified

by the user in the input file. The St. Johns Landfill simulation uses a 0.1 year

computational time step. For the sensitivity analysis, the user must specify how many

times the ranges for $K_{siltH}$ and $K_{siltV}$ should be split. The more times the range is split,

the more 'continuous' the sensitivity analysis will be. This simulation uses a range of

$1\times10^{-3}$ cm/sec to $1\times10^{-7}$ cm/sec for both $K_{siltH}$ and $K_{siltV}$. The range of likely hydraulic conductivities is broken into 20 steps meaning that for every simulation, 400 model runs are performed.

To run the model, fill in the appropriate parameter values into TIME.NPT as described below, run the executable of PTIME.FOR, then depending on the results desired, run either the executable for TDIM.FOR or the executable for TDIMDS.FOR.

The source code for each program is shown below. The data file TIME.NPT contains base case parameter values.

| Program Name | Input Files Needed | Output Files Produced | Description of Output |
|---|---|---|---|
| TIME.NPT | -None- | NA | User input file |
| PTIME.FOR | TIME.NPT | TIME.IN | Time varying parameters |
| | | NONTIME.IN | Constant Parameters |
| TDIM.FOR | TIME.IN | CSFLUX.OUT | Flux to Columbia Slough |
| | NONTIME.IN | DRAINFLUX.OUT | Flux through collection system (inactive) |
| | | HEAD.OUT | Contaminant mound height at each time step |
| | | LAKEFLUX.OUT | Flux to Smith Lake |
| | | NSFLUX.OUT | Flux to North Slough |
| | | NSCSFLUX.OUT | Flux to North and Columbia Sloughs (total) |
| | | PUMP.OUT | Flux due to pumping (inactive) |
| | | SILTFLUX.OUT | Flux through silt into gravel aquifer |
| | | SURFACEFLUX.OUT | Surface seepage flux |
| | | TDIM.OUT | Summary file of all fluxes and mounding |
| | | TIME.OUT | List of time for each output |
| | | LF.NPT | Contaminant mound height at end of each year |
| TDIMDS.FOR | TIME.IN | CSFLUX.OUT | Flux to Columbia Slough |
| | NONTIME.IN | DRAINFLUX.OUT | Flux through collection system (inactive) |
| | | HEAD.OUT | Contaminant mound height at each time step |
| | | LAKEFLUX.OUT | Flux to Smith Lake |
| | | NSFLUX.OUT | Flux to North Slough |
| | | NSCSFLUX.OUT | Flux to North and Columbia Sloughs (total) |
| | | PUMP.OUT | Flux due to pumping (inactive) |
| | | SILTFLUX.OUT | Flux through silt into gravel aquifer |
| | | SURFACEFLUX.OUT | Surface seepage flux |
| | | TDIM.OUT | Summary file of all fluxes and mounding |
| | | TIME.OUT | List of time for each output |
| | | LF.NPT | Contaminant mound height at end of each year |
| | | XLOGK.OUT | List of $K_{siltH}$ values at each time step |
| | | YLOGK.OUT | List of $K_{siltV}$ values at each time step |

**Table 3 - TIME.NPT is a user defined data file that contains all the model parameters. PTIME.FOR reads the data in TIME.NPT and places it into two different files for use by the flux calculation program, TDIM.FOR, and the sensitivity analysis program, TDIMDS.FOR.**

## TIME.NPT:

```
nkd  nks  ipr
 20   20   10
xkmin xkmax
 -7    -3
nper
 6
npmin npmax
 1    32
 33   41
 42   43
 44   45
 46   47
 48   70
```

| Dike1cond | Dike2cond | Dike1length | Dike2length | Dike3length | Dike1thick | Dike2thick | Siltthick |
|-----------|-----------|-------------|-------------|-------------|------------|------------|-----------|
| 9.0e-5 | 1e-7 | 10370 | 2180 | 2375 | 40 | 40 | 71.42 |

| SpecificYld | DT | Dikeheigth | Theta | Time | Numpipe | Hold |
|-------------|------|-----------|-------|------|---------|------|
| 0.20 | 0.10 | 25 | 0.5 | 70 | 1 | 10.2 |

| Covthick | Hriv | Siltcond | Covcond | Hlake | Hgrav | Rech | Area | Pump |
|----------|-------|----------|---------|-------|-------|------|-----------|------|
| 1.5 | 10.20 | 1.0e-6 | 1e-6 | 10.20 | 11.80 | 10.0 | 9713880.0 | 0.0 |

```
0.0  pxoverb
5.00  hpipe
```

| | | | | | | | | |
|----------|-------|----------|---------|-------|-------|------|-----------|------|
| 1.5 | 9.74 | 1.0e-6 | 1e-6 | 11.74 | 11.44 | 10.0 | 9713880.0 | 0.0 |

```
0.0  pxoverb
5.00  hpipe
```

| | | | | | | | | |
|----------|-------|----------|---------|-------|-------|------|-----------|------|
| 1.5 | 9.74 | 1.0e-6 | 1e-6 | 11.74 | 11.44 | 9.0 | 9713880.0 | 0.0 |

```
0.0  pxoverb
5.00  hpipe
```

| | | | | | | | | |
|----------|-------|----------|---------|-------|-------|------|-----------|------|
| 1.5 | 9.74 | 1.0e-6 | 1e-6 | 11.74 | 11.44 | 7.0 | 9713880.0 | 0.0 |

```
0.0  pxoverb
5.00  hpipe
```

| | | | | | | | | |
|----------|-------|----------|---------|-------|-------|------|-----------|------|
| 1.5 | 9.74 | 1.0e-6 | 1e-6 | 11.74 | 11.44 | 5.0 | 9713880.0 | 0.0 |

```
0.0  pxoverb
5.00  hpipe
```

| | | | | | | | | |
|----------|-------|----------|---------|-------|-------|------|-----------|------|
| 1.5 | 9.74 | 1.0e-6 | 1e-8 | 9.74 | 11.34 | 1.0 | 9713880.0 | 0.0 |

```
0.0  pxoverb
5.00  hpipe
```

*Siltcond is vertical silt conductivity under the landfill. To find anisotropy, divide Dike1cond by Siltcond.*
*Head in gravel is taken from average of regional flow model for cells under the landfill.*

TIME.NPT is the user defined parameter file. All input for the model is contained here and should be in this format to be properly read by PTIME.FOR. Spacing is not important, but line order, parameter order, and numerical format (scientific notation, integer, floating point, etc.) is important.

233

Parameter definitions are as follows:

| nkd & nks | Determines the number of splits between the high and low ranges of $K_{siltH}$ and $K_{siltV}$ for use in the sensitivity analysis program TDIMDS.FOR |
|---|---|
| ipr | Determines the frequency of the output for the sensitivity analysis in years. Used only in TDIMDS.FOR |
| xkmin & xkmax | Designates the exponent of the high and low ranges of $K_{siltH}$ and $K_{siltV}$ for use in the sensitivity analysis program TDIMDS.FOR. For example, if xkmin is equal to -7 and xkmax is equal to -3, then sensitivity analysis will be performed for conductivity values of $1 \times 10^{-7}$ to $1 \times 10^{-3}$ cm/sec. |
| nper | Number of stress periods. A stress period is a block of time during the simulation in which no parameters change. Used by both TDIM.FOR and TDIMDS.FOR. |
| npmin & npmax | The time periods encompassed by each time step. There should be the as many values of npmin and npmax as designated by nper. In this example, the first stress period is from year 1 to year 32 of the 70 year simulation, stress period two is from year 33 to year 41, and so on. Used by both TDIM.FOR and TDIMDS.FOR. |
| Dike1cond | The horizontal conductivity of the natural dike. Same as $K_{siltH}$. Used only by TDIM.FOR. (cm/sec) |
| Dike2cond | The horizontal conductivity of the engineered dike. Used by both TDIM.FOR and TDIMDS.FOR. (cm/sec) |
| Dike1length | The total length of the natural dike. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Dike2length | The length of the engineered dike which borders the sloughs. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Dike3length | The length of the engineered dike which borders Smith Lake. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Dike1thick & Dike2thick | The thickness of the natural and engineered dikes. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Siltthick | The harmonic mean of the silt thickness directly below the landfill. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| SpecificYld | The specific yield of the landfill. Used by both TDIM.FOR and TDIMDS.FOR. (unit-less) |
| DT | The length of each computational time step. Used by both TDIM.FOR and TDIMDS.FOR. (years) |
| Dikeheight | Height of both the engineered and natural dikes. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |

| | |
|---|---|
| Theta | Time weighting factor between 0 and 1 which determines the degree to which the model solves the governing equation either fully explicit (theta=0) or fully implicit (theta=1). Used by both TDIM.FOR and TDIMDS.FOR. (unit-less) |
| Numpipe | Dummy variable. Use default value. |
| Hold | The initial head condition in the landfill. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Covthick | The thickness of the daily landfill covering. This is not the same as the final cap put on the landfill. One value for each stress period. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Hriv | Water level height in the sloughs. One value for each stress period. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Siltcond | The vertical hydraulic conductivity of the silt. This is the same as $K_{siltv}$. One value for each stress period. Used by both TDIM.FOR and TDIMDS.FOR. (cm/sec) |
| Covcond | Conductivity of the daily landfill covering. One value for each stress period. Used by both TDIM.FOR and TDIMDS.FOR. (cm/sec) |
| Hlake | Water level in Smith Lake. One value for each stress period. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Hgrav | Head level in the gravel aquifer below the landfill. One value for each stress period. Used by both TDIM.FOR and TDIMDS.FOR. (feet) |
| Rech | Average amount of recharge across the entire landfill area. One value for each stress period. Used by both TDIM.FOR and TDIMDS.FOR. (inches/year) |
| Area | The total area of the landfill not including the dikes. One value for each stress period. Used by both TDIM.FOR and TDIMDS.FOR. (feet squared) |
| Pump | Dummy variable. Use default value. One value for each stress period. |
| pxoverb | Dummy variable. Use default value. One value for each stress period. |
| hpipe | Dummy variable. Use default value. One value for each stress period. |

**Table 4 - Variables and descriptions for the water balance model input file. Applies to both the flux calculation group and the particle tracking group of programs.**

## PTIME.FOR:

```fortran
c
c     Program to take water balance input file and process to two
c     seperate files.  One time dependent, the other not.  These are read
c     by tdim.for, and tdimds.for
c
      real covthick,hriv,siltcond,covcond,hlake,hgrav,rech,area,
     &    pkoverb,hpipe,d1c,d2c,dike1length,dike2length,
     &    dike3length,dike1thick,dike2thick,siltthick,specyld,
     &    dt,dikeheight,theta,time,hold,pump
      integer numpers,npmin,npmax,numpipes
      dimension npmin(100),npmax(100),pkoverb(100,100),hpipe(100,100),
     &    covthick(100),hriv(100),siltcond(100),pump(100),
     &    covcond(100),hlake(100),hgrav(100),rech(100),area(100)

      open(100,file='time.npt',status='old')
      open(200,file='time.in',status='unknown')
      open(300,file='nontime.in',status='unknown')

      read(100,*)
      read(100,*)
      read(100,*)
      read(100,*)
      read(100,*)

      read(100,*)numpers
      write(300,*)numpers
      read(100,*)
      do 5, i=1,numpers
        read(100,*)npmin(i),npmax(i)
      write(300,*)npmin(i),npmax(i)
5     continue
      read(100,*)
      read(100,*)d1c,d2c,dike1length,dike2length,
     &    dike3length,dike1thick,dike2thick,
     &    siltthick,specyld,dt,dikeheight,theta,
     &    time,numpipes,hold
      write(300,*)d1c,d2c,dike1length,dike2length,
     &    dike3length,dike1thick,dike2thick,
     &    siltthick,specyld,dt,dikeheight,theta,
     &    time,numpipes,hold
      read(100,*)
      do 15, i=1,numpers
        read(100,*)covthick(i),hriv(i),siltcond(i),covcond(i),
     &      hlake(i),hgrav(i),rech(i),area(i),pump(i)
        do 25, j=1,numpipes
          read(100,*)pkoverb(j,i)
          read(100,*)hpipe(j,i)
25    continue
15    continue
      do 35, i=1,numpers
        do 45, j=npmin(i),npmax(i)
          write(200,*)covthick(i),hriv(i),siltcond(i),covcond(i),
     &        hlake(i),hgrav(i),rech(i),area(i),pump(i),j
45      continue
        do 55, k=1,numpipes
          write(200,*)pkoverb(k,i)
          write(200,*)hpipe(k,i)
55      continue
35    continue
      stop
      end
```

236

## TDIM.FOR:

```
c Program for zero dimensional St. Johns Landfill Model
c Thomas Lowry - Portland State University - 4/95
c
c List of Variables:
c theta - weighting factor between successive time steps
c hold - head in landfill in current time step
c hnew - head in landfill in next time step
c hriv - head in sloughs
c hlake - head in Smith and Bybee Lakes
c hgrav - head in gravel aquifer
c hpipe - array of head levels in perimeter drains
c dike1cond - conductivity in natural dike
c dike2cond - conductivity in engineered dike
c siltcond - conductivity in silt
c covcond - conductivity of daily cover
c pipecond - array of effective conductivities for perimeter
drains
c dike1length - length of natural dike
c dike2length - length of engineered dike bordering slough
c dike3length - length of engineered dike bordering Smith
Lake
c dike1thick - thickness of natural dike
c dike2thick - thickness of engineered dike
c covthick - thickness of daily cover
c siltthick - harmonic mean of silt thickness data
c pipethick - array of effective thicknesses for perimeter
drains
c area - landfill area
c pipearea - array of effective areas for perimeter drains
c rech - recharge
c specyld - specific yield of landfill
c dt - time step size
c dikeheight - elevation of top of dikes
c time - total simulation time in years
c numpipe - number of perimeter drain segements
c conv - seconds per year
c conv2 - cm/sec to ft/day
c conv3 - gal/min to cfs

      parameter(pipes=10,times=100,conv=365*24*60*60,
     &      conv2=24*60*60/30.48,conv3=.002228)
      real pkoverb,hpipe,bp,cp,hp,pa,
     &      covthick,hriv,siltcond,covcond,hlake,hgrav,rech,
     &
df,theta,hold,hnew,dike1cond,dike2cond,dike1length,
     &      dike2length,dike3length,dike1thick,dike2thick,
     &      siltthick,specyld,dt,area,pump,pmp,pp,pkb,
     &      dikeheight,bt,ct,dd1,ad1,dd2,ad2,ad3,ds,d7,ce,cpu,
     &      cd1,cd2,bs,bbp,ccp,a7,b7,c7,cd3,afactor,bfactor,
     &      cfactor,ndf,esf,sf,dfl,ssf,elf,d1c,d2c,rch
      integer time,numpipe,ii,jj,nper,npmin,npmax
      dimension pkoverb(pipes,times),hpipe(pipes,times),
     &      bp(pipes),cp(pipes),hp(pipes),pa(pipes),
     &      covthick(times),hriv(times),
     &      siltcond(times),covcond(times),hlake(times),
     &      hgrav(times),rech(times),df(pipes),area(times),
     &      npmin(times),npmax(times),pump(times)

      open (46,file='time.out',status='unknown')
      open(100,file='nontime.in',status='old')
      open(200,file='time.in',status='old')
      open(300,file='tdim.out',status='unknown')
      open(301,file='lf.rpt',status='unknown')
c
c Read in initial and constant parameters
c
      read(100,*)nper
      do 3, i=1,nper
        read(100,*)npmin(i),npmax(i)
3     continue
      read(100,*)d1c,d2c,dike1length,dike2length,
     &      dike3length,dike1thick,dike2thick,
     &      siltthick,specyld,dt,dikeheight,theta,
     &      time,numpipe,hold
      dike1cond=d1c*365*conv2
      dike2cond=d2c*365*conv2
c
c Calculate factors which are not time dependent.
c
c Storage
c
      dst=specyld/dt
c
c Silt
c
      ds=1/(siltthick)
c
c Surface Seepage
c
      d7=(dike1length+dike2length+dike3length)/2
c
c Read in time dependent variables
c
      do 23, kk=1,nper
        do 25, j=npmin(kk),npmax(kk)
          read(200,*)covthick(j),hriv(j),siltc,covc,
     &        hlake(j),hgrav(j),rch,area(j),pmp
          rech(j)=rch/12.0
          siltcond(j)=siltc*365*conv2
          covcond(j)=covc*365*conv2
          pump(j)=pmp*conv3*conv
25      continue
        if(numpipe.eq.0)goto 36
        do 35, i=1,numpipe
          read(200,*)pkb
          read(200,*)hhpp
          do 33, ll=npmin(kk),npmax(kk)
            pkoverb(i,ll)=pkb*conv
            hpipe(i,ll)=hhpp
33        continue
35      continue
36      continue
23    continue
c
c Start time loop
c
      ii=1
      jj=1
      ddt=0.0
      numsteps=time/dt
      do 45, i=1,numsteps
        write (46,6000) ddt
        if(ddt.gt.ii)then
          ctt=covthick(ii+1)
          hr=hriv(ii+1)
          sc=siltcond(ii+1)
          cc=covcond(ii+1)
          hl=hlake(ii+1)
          hg=hgrav(ii+1)
          rc=rech(ii+1)
          ar=area(ii+1)
          pp=pump(ii+1)
        if(numpipe.eq.0)goto 56
```

```
      do 55, k=1,numpipe
      pa(k)=pkoverb(k,ii+1)
      hp(k)=hpipe(k,ii+1)
 55   continue
 56   continue
      ii=ii+1
      else
      ctt=covthick(ii)
      hr=hriv(ii)
      sc=siltcond(ii)
      cc=covcond(ii)
      hl=hlake(ii)
      hg=hgrav(ii)
      rc=rech(ii)
      ar=area(ii)
      pp=pump(ii)
      if(numpipe.eq.0)goto 66
      do 65, k=1,numpipe
      pa(k)=pkoverb(k,ii)
      hp(k)=hpipe(k,ii)
 65   continue
 66   continue
      endif
      if(ddt.gt.(time-1))then
      ctt=covthick(ii)
      hr=hriv(ii)
      sc=siltcond(ii)
      cc=covcond(ii)
      hl=hlake(ii)
      hg=hgrav(ii)
      rc=rech(ii)
      ar=area(ii)
      pp=pump(ii)
      if(numpipe.eq.0)goto 68
      do 67, k=1,numpipe
      pa(k)=pkoverb(k,ii)
      hp(k)=hpipe(k,ii)
 67   continue
 68   continue
      ii=ii-1
      endif
c
c Calculate time dependent factors
c
c Storage factors
c
      bt=dst*(area(ii+1)*theta+ar*(1-theta))
      ct=-bt*hold
c
c Recharge
c
      ce=(rech(ii+1)*area(ii+1)*theta+
     & rc*ar*(1-theta))
c
c Natural Dike
c
      dd1=(dike1cond*dike1length)/(2*(dike1thick))
      if(hold.gt.dikeheight)then
      ad1=0
      bd1=(dikeheight+hriv(ii+1))*theta*dd1
      cd1=(-hriv(ii+1)*dikeheight*theta-
hriv(ii+1)*hriv(ii+1)*theta
     &       +hold*dikeheight*(1-theta)-hr*dikeheight*(1-
theta)
     &       +hold*hr*(1-theta)-hr*hr*(1-theta))*dd1
      else
      ad1=(theta*dd1)
```

```
      bd1=0
      cd1=dd1*(-
hriv(ii+1)*hriv(ii+1)*theta+hold*hold*(1-theta)
     &       -hr*hr*(1-theta))
      endif
c
c Engineered Dike to Slough
c
      dd2=(dike2cond*dike2length)/(2*(dike2thick))
      if(hold.gt.dikeheight)then
      ad2=0
      bd2=(dikeheight+hriv(ii+1))*theta*dd2
      cd2=(-hriv(ii+1)*dikeheight*theta-
hriv(ii+1)*hriv(ii+1)*theta
     &       +hold*dikeheight*(1-theta)-hr*dikeheight*(1-
theta)
     &       +hold*hr*(1-theta)-hr*hr*(1-theta))*dd2
      else
      ad2=(theta*dd2)
      bd2=0
      cd2=dd2*(-
hriv(ii+1)*hriv(ii+1)*theta+hold*hold*(1-theta)
     &       -hr*hr*(1-theta))
      endif
c
c Silt
c
      bs=ds*(siltcond(ii+1)*area(ii+1)*theta)
      cs=ds*(-
siltcond(ii+1)*hgrav(ii+1)*area(ii+1)*theta+sc*hold*
     & ar*(1-theta)-sc*hg*ar*(1-theta))
c
c Perimeter Drains
c
      bbp=0.0
      ccp=0.0
      if(numpipe.eq.0)goto 76
      do 75, k=1,numpipe
      if(hold.le.hp(k))then
      bp(k)=0
      cp(k)=0
      else
      bp(k)=pkoverb(k,ii+1)*theta
      cp(k)=-pkoverb(k,ii+1)*hpipe(k,ii+1)*theta
     &       +pa(k)*hold*(1-theta)-pa(k)*hp(k)*(1-theta)
      bbp=bbp+bp(k)
      ccp=ccp+cp(k)
      endif
 75   continue
 76   continue
c
c Surface Seepage
c
      if(hold.lt.dikeheight)then
      a7=0
      b7=0
      c7=0
      else
      a7=(covcond(ii+1)*d7*theta)/(covthick(ii+1))
      b7=(-
2*covcond(ii+1)*dikeheight*theta*d7)/(covthick(ii+1))
      c7=((covcond(ii+1)*dikeheight*dikeheight*theta)/
     &    (covthick(ii+1))
     &    +((cc*(1-theta)*hold*hold)/(ctt))
     &    -((2*cc*dikeheight*(1-theta)*hold)/(ctt))
     &    +((cc*dikeheight*dikeheight*(1-theta))/(ctt)))*d7
      endif
```

```fortran
c
c Engineered Dike to Lake
c
      dd3=(dike2cond*dike3length)/(2*(dike2thick))
      if(hold.gt.dikeheight)then
        ad3=0
        bd3=(dikeheight+hriv(ii+1))*theta*dd3
        cd3=(-hriv(ii+1)*dikeheight*theta-
     hriv(ii+1)*hriv(ii+1)*theta
     &      +hold*dikeheight*(1-theta)-hr*dikeheight*(1-
     theta)
     &      +hold*hr*(1-theta)-hr*hr*(1-theta))*dd3
      else
        ad3=(theta*dd3)
        bd3=0
        cd3=dd3*(-
     hlake(ii+1)*hlake(ii+1)*theta+hold*hold*(1-theta)
     &      -hl*hl*(1-theta))
      endif
c
c Pumps
c
      cpu=pp*theta+pump(ii+1)*(1-theta)
c
c Consolidate factors and solve for hnew
c
      afactor=-ad1-ad2-a7-ad3
      bfactor=-bt-bs-bbp-b7-bd1-bd2-bd3
      cfactor=-ct+ce-cd1-cd2-cs-ccp-c7-cd3-cpu
c
c Calculate new head using binomial theorem
c
      hnew=(-bfactor-sqrt((bfactor*bfactor)-
     4*afactor*cfactor))
     &    /(2*afactor)
c
c Calculate fluxes for each component
c
c Natural Dike
c
      ndf=hnew*hnew*ad1+hnew*bd1+cd1
c
c Engineered Dike to Slough
c
      esf=hnew*hnew*ad2+hnew*bd2+cd2
c
c Silt
c
      sf=hnew*bs+cs
c
c Perimeter Drain
c
      dfl=0.0
      if(numpipe.eq.0)goto 96
      do 95, k=1,numpipe
      if(hnew.lt.hpipe(k,ii+1))then
        df(k)=0.0
        goto 95
      endif
      if(hold.lt.hp(k))then
        df(k)=0.0
        goto 95
      endif
      df(k)=hnew*bp(k)+cp(k)
      dfl=dfl+df(k)
  95  continue
  96  continue
c
c Surface Seepage
c
      ssf=hnew*hnew*a7+hnew*b7+c7

 2000 continue
c
c Engineered Dike to Lake
c
      elf=hnew*hnew*ad3+hnew*bd3+cd3
c
c Pump
c
      puf=cpu
c
c Calculate fluxes for each part
c
        dike1flux=ndf/conv
        dike2flux=esf/conv
        dike3flux=elf/conv
        siltflux=sf/conv
        drainflux=dfl/conv
        surflux=ssf/conv
        pumpflux=puf/conv
        head=hnew
        xnsflux=((ndf/dike1length)*2645.+
     &      (esf/dike2length)*2180.)/conv
        csflux=((ndf/dike1length)*7225.+
     &      (esf/dike2length)*500.)/conv
c
c Write out results for each time period
c
      open (35,file='nsflux.out',status='unknown')
      open (36,file='csflux.out',status='unknown')
      open (37,file='nscsflux.out',status='unknown')
      open (38,file='lakeflux.out',status='unknown')
      open (39,file='siltflux.out',status='unknown')
      open (40,file='drainflux.out',status='unknown')
      open (41,file='surfaceflux.out',status='unknown')
      open (42,file='head.out',status='unknown')
      open (47,file='pump.out',status='unknown')

      write (35,*) xnsflux
      write (36,*) csflux
      write (37,*) xnsflux+csflux
      write (38,*) dike3flux
      write (39,*) siltflux
      write (40,*) drainflux
      write (41,*) surflux
      write (42,*) head
      write (47,*) pumpflux

      if(i.eq.1)then
      write(300,'(8a10)')'Nat.Dike','Eng.DikeS','Silt',
     &      'Drain',
     &      'Surface','Eng.DikeL','Time','Head'
      endif
      if(ddt.ge.jj.or.ddt.ge.(time-dt))then
       write(300,'(6f10.5,i10,f10.5)')ndf/conv,esf/conv,
     &              sf/conv,
     &              dfl/conv,ssf/conv,elf/conv,
     &              jj,hnew
       write(301,'(f10.4)')hnew
      jj=jj+1
      endif
c
c Step time by dt
```

239

```
c
      ddt=ddt+dt
c
c Move new heads to old heads
c
      hold=hnew

 6000   format (30(1x,f9.6))
   45 continue
      stop
      end
```

## TDIMDS.FOR:

```
c Program for zero dimensional St. Johns Landfill Model
c Anisotropy sensitivity analysis program
c Thomas Lowry - Portland State University - 4/95
c
c List of Variables:
c theta - weighting factor between successive time steps
c hold - head in landfill in current time step
c hnew - head in landfill in next time step
c hriv - head in sloughs
c hlake - head in Smith and Bybee Lakes
c hgrav - head in gravel aquifer
c hpipe - array of head levels in perimeter drains
c dike1cond - conductivity in natural dike
c dike2cond - conductivity in engineered dike
c siltcond - conductivity in silt
c covcond - conductivity of daily cover
c pipecond - array of effective conductivities for perimeter
drains
c dike1length - length of natural dike
c dike2length - length of engineered dike bordering slough
c dike3length - length of engineered dike bordering Smith
Lake
c dike1thick - thickness of natural dike
c dike2thick - thickness of engineered dike
c covthick - thickness of daily cover
c siltthick - harmonic mean of silt thickness data
c pipethick - array of effective thicknesses for perimeter
drains
c area - landfill area
c pipearea - array of effective areas for perimeter drains
c rech - recharge
c specyld - specific yield of landfill
c dt - time step size
c dikeheight - elevation of top of dikes
c time - total simulation time in years
c numpipe - number of perimeter drain segements
c ipr - frequency of output (in years)

      parameter(pipes=10,times=100,conv=365*24*60*60,
     &    conv2=24*60*60/30.48,conv3=.002228)
      real pkoverb,hpipe,bp,cp,hp,pa,
     &    covthick,hriv,siltcond,covcond,hlake,hgrav,rech,
     &
df,theta,hold,hnew,dike1cond,dike2cond,dike1length,
     &    dike2length,dike3length,dike1thick,dike2thick,
     &
siltthick,specyld,dt,area,pump,pmp,pp,pkb,pumpflux,
     &    dikeheight,bt,ct,dd1,ad1,dd2,ad2,ad3,ds,d7,ce,
     &    cd1,cd2,bs,bbp,ccp,a7,b7,c7,cd3,afactor,bfactor,
     &    cfactor,ndf,esf,sf,dfl,ssf,elf,d1c,d2c,rch
      integer time,numpipe,ii,jj,nper,npmin,npmax
      dimension pkoverb(pipes,times),hpipe(pipes,times),
     &    bp(pipes),cp(pipes),hp(pipes),pa(pipes),
     &    covthick(times),hriv(times),
     &    siltcond(times),covcond(times),hlake(times),
     &    hgrav(times),rech(times),df(pipes),area(times),
     &    npmin(times),npmax(times),pump(times)

      dimension dike1flux(100,100),dike2flux(100,100),
     &
dike3flux(100,100),csflux(100,100),xnsflux(100,100),
     &
siltflux(100,100),drainflux(100,100),surflux(100,100),
     &    head(100,100),xkd(100),xks(100),hnew(100,100),
     &    hold(100,100),pumpflux(100,100)
```

240

```fortran
      open (46,file='time.out',status='unknown')
      open(100,file='nontime.in',status='old')
      open(200,file='time.in',status='old')
      open(300,file='tdim.out',status='unknown')
      open(400,file='time.npt',status='unknown')
c
c Read in initial and constant parameters
c
      read(100,*)nper
      do 3, i=1,nper
        read(100,*)npmin(i),npmax(i)
  3   continue

      read(100,*)d1c,d2c,dike1length,dike2length,
     &    dike3length,dike1thick,dike2thick,
     &    siltthick,specyld,dt,dikeheight,theta,
     &    time,numpipe,hold(1,1)
      dike1cond=d1c*365*conv2
      dike2cond=d2c*365*conv2
c
c Read in timing information
c
      read(400,*)
      read(400,*)nkd,nks,ipr
      read(400,*)
      read(400,*)xkmin,xkmax
c
c Calculate factors which are not time dependent.
c
c Storage
c
      dst=specyld/dt
c
c Silt
c
      ds=1/(siltthick)
c
c Surface Seepage
c
      d7=(dike1length+dike2length+dike3length)/2
c
c Read in time dependent variables
c
      do 23, kk=1,nper
        do 25, j=npmin(kk),npmax(kk)
          read(200,*)covthick(j),hriv(j),siltc,covc,
     &        hlake(j),hgrav(j),rch,area(j),pmp
          rech(j)=rch/12.0
          siltcond(j)=siltc*365*conv2
          covcond(j)=covc*365*conv2
          pump(j)=pmp*conv3*conv
 25     continue
        if(numpipe.eq.0)goto 36
        do 35, i=1,numpipe
          read(200,*)pkb
          read(200,*)hhpp
          do 33, ll=npmin(kk),npmax(kk)
            pkoverb(i,ll)=pkb*conv
            hpipe(i,ll)=hhpp
 33       continue
 35     continue
 36     continue
 23   continue
c
c Start time loop
c
      ii=1
```

```fortran
      jj=1
      ddt=0.0
      numsteps=time/dt
      do kd=1,nkd
        do ks=1,nks
          hold(kd,ks)=hold(1,1)
        end do
      end do

      do 45, i=1,numsteps
        write (46,6000) ddt
        do kd=1,nkd
          xkd(kd)=xkmin+(kd-1)*(xkmax-xkmin)/(nkd-1)
          do ks=1,nks
            xks(ks)=xkmin+(ks-1)*(xkmax-xkmin)/(nkd-1)
c
c Convert Units to ft/year
c
            dike1cond=10**(xkd(kd))*365*conv2
c
c Calculate Constants
c Natural Dike
c
            dd1=(dike1cond*dike1length)/(2*(dike1thick))
            ad1=(theta*dd1)
c
c Engineered Dike to Slough
c
            dd2=(dike2cond*dike2length)/(2*(dike2thick))
            ad2=(theta*dd2)
c
c Engineered Dike to Lake
c
            dd3=(dike2cond*dike3length)/(2*(dike2thick))
            ad3=(theta*dd3)

            do j=1,time
              siltcond(j)=10**(xks(ks))*365*conv2
            end do

            if(ddt.gt.ii)then
              ctt=covthick(ii+1)
              hr=hriv(ii+1)
              sc=siltcond(ii+1)
              cc=covcond(ii+1)
              hl=hlake(ii+1)
              hg=hgrav(ii+1)
              rc=rech(ii+1)
              ar=area(ii+1)
              pp=pump(ii+1)
              if(numpipe.eq.0)goto 56
              do 55, k=1,numpipe
                pa(k)=pkoverb(k,ii+1)
                hp(k)=hpipe(k,ii+1)
 55           continue
 56           continue
              ii=ii+1
            else
              ctt=covthick(ii)
              hr=hriv(ii)
              sc=siltcond(ii)
              cc=covcond(ii)
              hl=hlake(ii)
              hg=hgrav(ii)
              rc=rech(ii)
              ar=area(ii)
              pp=pump(ii)
```

241

```fortran
              if(numpipe.eq.0)goto 66
              do 65, k=1,numpipe
                pa(k)=pkoverb(k,ii)
                hp(k)=hpipe(k,ii)
65            continue
66            continue
            endif
            if(ddt.gt.(time-1))then
              ctt=covthick(ii)
              hr=hriv(ii)
              sc=siltcond(ii)
              cc=covcond(ii)
              hl=hlake(ii)
              hg=hgrav(ii)
              rc=rech(ii)
              ar=area(ii)
              pp=pump(ii)
              if(numpipe.eq.0)goto 68
              do 67, k=1,numpipe
                pa(k)=pkoverb(k,ii)
                hp(k)=hpipe(k,ii)
67            continue
68            continue
            ii=ii-1
            endif
c
c Calculate time dependent factors
c
c
c Storage factors
c
            bt=dst*(area(ii+1)*theta+ar*(1-theta))
            ct=-bt*hold(kd,ks)
c
c Recharge
c
            ce=(rech(ii+1)*area(ii+1)*theta+
     &          rc*ar*(1-theta))
c
c Natural Dike
c
       dd1=(dike1cond*dike1length)/(2*(dike1thick))
       if(hold(kd,ks).gt.dikeheight)then
         ad1=0
         bd1=(dikeheight+hriv(ii+1))*theta*dd1
         cd1=(-hriv(ii+1)*dikeheight*theta-
hriv(ii+1)*hriv(ii+1)*theta
     &       +hold(kd,ks)*dikeheight*(1-theta)-
hr*dikeheight*(1-theta)
     &       +hold(kd,ks)*hr*(1-theta)-hr*hr*(1-theta))*dd1
       else
         ad1=(theta*dd1)
         bd1=0
         cd1=dd1*(-
hriv(ii+1)*hriv(ii+1)*theta+hold(kd,ks)*hold(kd,ks)
     &       *(1-theta)-hr*hr*(1-theta))
       endif
c
c Engineered Dike to Slough
c
       dd2=(dike2cond*dike2length)/(2*(dike2thick))
       if(hold(kd,ks).gt.dikeheight)then
         ad2=0
         bd2=(dikeheight+hriv(ii+1))*theta*dd2
         cd2=(-hriv(ii+1)*dikeheight*theta-
hriv(ii+1)*hriv(ii+1)*theta
```

```fortran
     &       +hold(kd,ks)*dikeheight*(1-theta)-
hr*dikeheight*(1-theta)
     &       +hold(kd,ks)*hr*(1-theta)-hr*hr*(1-theta))*dd2
       else
         ad2=(theta*dd2)
         bd2=0
         cd2=dd2*(-
hriv(ii+1)*hriv(ii+1)*theta+hold(kd,ks)*hold(kd,ks)
     &       *(1-theta)-hr*hr*(1-theta))
       endif
c
c Silt
c
            bs=ds*(siltcond(ii+1)*area(ii+1)*theta)
            cs=ds*(-
siltcond(ii+1)*hgrav(ii+1)*area(ii+1)*theta+sc*
     &         hold(kd,ks)*
     &         ar*(1-theta)-sc*hg*ar*(1-theta))
c
c Perimeter Drains
c
            bbp=0.0
            ccp=0.0
            if(numpipe.eq.0)goto 76
            do 75, k=1,numpipe
              if(hold(kd,ks).le.hp(k))then
                bp(k)=0
                cp(k)=0
              else
                bp(k)=pkoverb(k,ii+1)*theta
                cp(k)=-pkoverb(k,ii+1)*hpipe(k,ii+1)*theta
     &              +pa(k)*hold(kd,ks)*(1-theta)-
pa(k)*hp(k)*
     &              (1-theta)
                bbp=bbp+bp(k)
                ccp=ccp+cp(k)
              endif
75          continue
76          continue
c
c    Surface Seepage
c
       if(hold(kd,ks).lt.dikeheight)then
       a7=0
       b7=0
       c7=0
       else
       a7=(covcond(ii+1)*d7*theta)/(covthick(ii+1))
       b7=(-
2*covcond(ii+1)*dikeheight*theta*d7)/(covthick(ii+1))
       c7=((covcond(ii+1)*dikeheight*dikeheight*theta)/
     &    (covthick(ii+1))
     &    +((cc*(1-theta)*hold(kd,ks)*hold(kd,ks))/(ctt))
     &    -((2*cc*dikeheight*(1-theta)*hold(kd,ks))/(ctt))
     &    +((cc*dikeheight*dikeheight*(1-theta))/(ctt)))*d7
       endif
c
c    Engineered Dike to Lake
c
       dd3=(dike2cond*dike3length)/(2*(dike2thick))
       if(hold(kd,ks).gt.dikeheight)then
         ad3=0
         bd3=(dikeheight+hriv(ii+1))*theta*dd3
         cd3=(-hriv(ii+1)*dikeheight*theta-
hriv(ii+1)*hriv(ii+1)*theta
     &       +hold(kd,ks)*dikeheight*(1-theta)-
hr*dikeheight*(1-theta)
```

242

```
      &      +hold(kd,ks)*hr*(1-theta)-hr*hr*(1-theta))*dd3
     else
       ad3=(theta*dd3)
       bd3=0
       cd3=dd3*(-
hlake(ii+1)*hlake(ii+1)*theta+hold(kd,ks)*hold(kd,ks)*(1
-theta)
     &      -hl*hl*(1-theta))
     endif
c
c Pumps
c
     cpu=pp*theta+pump(ii+1)*(1-theta)


c
c    Consolidate factors and solve for hnew
c
     afactor=-ad1-ad2-a7-ad3
     bfactor=-bt-bs-bbp-b7-bd1-bd2-bd3
     cfactor=-ct+ce-cd1-cd2-cs-ccp-c7-cd3-cpu

         hnew(kd,ks)=(-bfactor-sqrt((bfactor*bfactor)-
     &       4*afactor*cfactor))
     &       /(2*afactor)


c
c Calculate fluxes for each component
c
c Natural Dike
c

ndf=hnew(kd,ks)*hnew(kd,ks)*ad1+hnew(kd,ks)*bd1+cd1
c
c Engineered Dike to Slough
c

esf=hnew(kd,ks)*hnew(kd,ks)*ad2+hnew(kd,ks)*bd2+cd2
c
c Silt
c
         sf=hnew(kd,ks)*bs+cs
c
c Perimeter Drain
c
         dfl=0.0
          if(numpipe.eq.0)goto 96
          do 95, k=1,numpipe
           if(hnew(kd,ks).lt.hpipe(k,ii+1))then
             df(k)=0.0
             goto 95
           endif
           if(hold(kd,ks).lt.hp(k))then
             df(k)=0.0
             goto 95
           endif
           df(k)=hnew(kd,ks)*bp(k)+cp(k)
           dfl=dfl+df(k)
 95        continue
 96        continue
c
c Surface Seepage
c
     ssf=hnew(kd,ks)*hnew(kd,ks)*a7+hnew(kd,ks)*b7+c7
c
c Engineered Dike to Lake
c
```

```
elf=hnew(kd,ks)*hnew(kd,ks)*ad3+hnew(kd,ks)*ad3+cd3
c
c Pump
c
     puf=cpu
c
c
c Write our results for each time period
c
        if(i.eq.1)then
          write(300,'(8a10)')'Nat.Dike','Eng.DikeS','Silt',
     &        'Drain',
     &        'Surface','Eng.DikeL','Time','Head'
        endif
        if(ddt.ge.jj.or.ddt.ge.(time-dt))then
          write(300,'(6f10.5,i10,f10.5)')ndf/conv,esf/conv,
     &        sf/conv,
     &        dfl/conv,ssf/conv,elf/conv,
     &        jj,hnew(kd,ks)
          jj=jj+1
        endif

        dike1flux(kd,ks)=ndf/conv
        dike2flux(kd,ks)=esf/conv
        dike3flux(kd,ks)=elf/conv
        siltflux(kd,ks)=sf/conv
        drainflux(kd,ks)=dfl/conv
        surflux(kd,ks)=ssf/conv
        pumpflux(kd,ks)=puf/conv
        head(kd,ks)=hnew(kd,ks)

        xnsflux(kd,ks)=((ndf/dike1length)*2645.+
     *       (esf/dike2length)*2180.)/conv
        csflux(kd,ks)=((ndf/dike1length)*7225.+
     *       (esf/dike2length)*500.)/conv
        hold(kd,ks)=hnew(kd,ks)
      end do
     end do

     ddt=ddt+dt


     open (35,file='nsflux.out',status='unknown')
     open (36,file='csflux.out',status='unknown')
     open (37,file='nscsflux.out',status='unknown')
     open (38,file='lakeflux.out',status='unknown')
     open (39,file='siltflux.out',status='unknown')
     open (40,file='drainflux.out',status='unknown')
     open (41,file='surfaceflux.out',status='unknown')
     open (42,file='head.out',status='unknown')
     open (43,file='xlogk.out',status='unknown')
     open (44,file='ylogk.out',status='unknown')
     open (47,file='pump.out',status='unknown')

     if (ii.eq.(ii/ipr)*ipr.or.ii.eq.1) then
       write (35,6000)
((xnsflux(kd,ks),ks=1,nks),kd=1,nkd)
       write (36,6000) ((csflux(kd,ks),ks=1,nks),kd=1,nkd)
       write (37,6000)
((xnsflux(kd,ks)+csflux(kd,ks),ks=1,nks),
     *       kd=1,nkd)
       write (38,6000)
((dike3flux(kd,ks),ks=1,nks),kd=1,nkd)
       write (39,6000)
((siltflux(kd,ks),ks=1,nks),kd=1,nkd)
```

243

```
      write (40,6000)
((drainflux(kd,ks),ks=1,nks),kd=1,nkd)
      write (41,6000)
((surflux(kd,ks),ks=1,nks),kd=1,nkd)
      write (42,6000) ((head(kd,ks),ks=1,nks),kd=1,nkd)
      write (47,6000)
((pumpflux(kd,ks),ks=1,nks),kd=1,nkd)
    endif


    if (ii.eq.1) then
      write (43,6001) ((xkd(kd),ks=1,nks),kd=1,nkd)
      write (44,6001) ((xks(ks),ks=1,nks),kd=1,nkd)
    endif

6000  format (30(1x,f16.10))
6001  format (30(1x,f9.2))

45  continue
    stop
    end
```

## Particle Tracking Model:

The particle tracking model is very similar to the flux calculation model. There is a preprocessor (PTIMEPT.FOR) which forms the user input (TIMEPT.NPT) in such a way that can be easily used by the main programs, a version of the water balance model (TDIMPT.FOR) which outputs the contaminant mound values in a form which can be used by the particle tracking programs, and two particle tracking programs; one for tracking particles vertically through the silt (SILTTRANS.FOR), and the other for tracking particles horizontally through the dike (ZDIMTRANS.FOR).
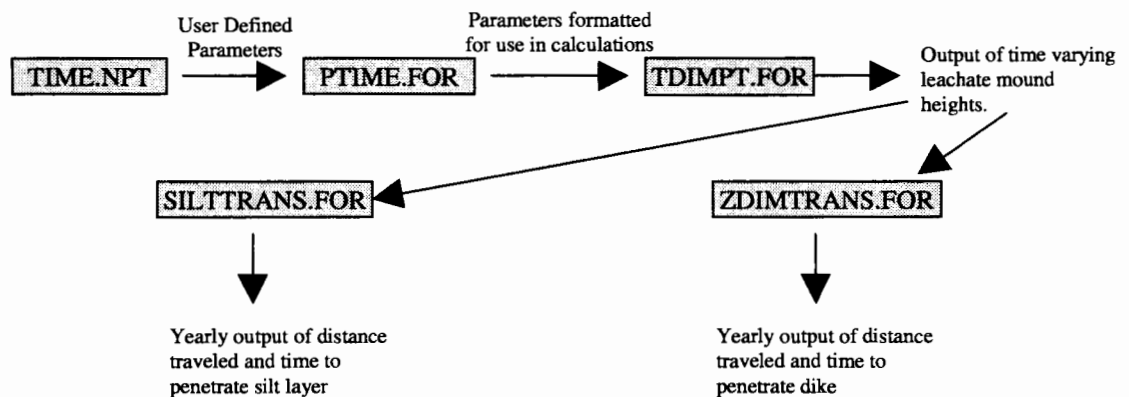


**Figure 90 - Shows program structure of the water balance particle tracking model. TIME.NPT is a user-input file which defines all model parameters.**

TIMEPT.NPT, PTIMEPT.FOR, and TDIMPT.FOR are the same as their counterparts in the water balance model, with slight variations on the output format. SILTTRANS.FOR splits the landfill domain into a 130 by 80 equally spaced grid with each grid point assigned a silt thickness determined from the geology files supplied by

245

METRO, and tracks a particle released at the bottom of the landfill in each cell. Output for SILTTRANS.FOR (called SILTTIMES.OUT and SILTDIST.OUT, see Table 5) is in a stacked format with each yearly output stacked one on top of the other. Cell addresses are tracked in two separate output files, X.OUT and Y.OUT. The first entry in either SILTTIMES.OUT or SILTDIST.OUT correspond to a node address of the first entry in X.OUT for the x coordinate, and the first entry in Y.OUT for the y coordinate; the second entry matches to the second entry in each file and so on. These files are for use in post-processing with PV-WAVE (a commercial graphing and visualization program).

ZDIMTRANS.FOR only tracks one particle across the dike since the dike is assumed uniform thickness along its entire length. Output for ZDIMTRANS.FOR is straight forward with output for each time step showing time and distance traveled in two separate files. The time step is not uniform because the model calculates the time for the particle to move 0.1 feet, incrementally moving the particle across the dike until the particle has penetrated the dike or the simulation comes to an end. Velocities across the dike will vary depending upon the mound height, the water level in the sloughs, and the position in the dike.

SILTTHICK.OUT is the output from an interpolation program which applies a linear interpolation to the stratigraphy data supplied by METRO to 'fit' the smaller resolution stratigraphy data to the larger resolution SILTRANS.FOR. Other than being described in Table 5 below, neither the interpolation program nor its output is shown.

| Program Name | Input Files Needed | Output Files Produced | Description of Output |
|---|---|---|---|
| SILTTHICK.OUT | -None- | NA | Data file describing the silt thickness at each node address for SILTTRANS.FOR |
| TIMEPT.NPT | -None- | NA | User input file |
| PTIMEPT.FOR | TIME.NPT | TIME.IN<br>NONTIME.IN | Time varying parameters<br>Constant Parameters |
| TDIMPT.FOR | TIME.IN<br>NONTIME.IN<br>TRANS.NPT | TRANS.NPT | Contaminant mound height at each time step |
| ZDIMTRANS.FOR | TIME.IN<br>NONTIME.IN<br>TRANS.NPT<br>SILTTHICK.OUT | TRANS.OUT<br><br><br>TIME.OUT<br><br>DISTANCE.OUT<br>NUM.OUT | Summary output for each time step of distance traveled, time step, head, and velocity of particle<br>Time traveled for particle. Stops when particle penetrates dike<br>Distance traveled at each time step<br>Number of time steps for particle to penetrate dike at various retardation factors |
| SILTTRANS.FOR | TIME.IN<br>NONTIME.IN<br>TRANS.NPT | SILTTIMES.OUT<br><br>SILTDIST.OUT<br><br>X.OUT<br><br>Y.OUT<br><br>YESNO.OUT | Time traveled for each particle. Stops when particle penetrates silt<br>Distance traveled for each particle at each time step<br>Node address in x direction corresponding to time and distance output<br>Node address in y direction corresponding to time and distance output<br>End of simulation output, producing a 1 if the particle has penetrated the silt, a 0 if not |

**Table 5 - Input and output files needed for the particle tracking group of programs for the water balance model. TIME.NPT is a user defined parameter file that is read by PTIMEPT.FOR and formatted for use in the other programs.**

To run the particle tracking model, fill in the appropriate parameter values into TIMEPT.NPT as described in the water balance model above, run the executable of PTIMEPT.FOR, then depending on the results desired, run either the executable for ZDIMTRANS.FOR or the executable for SILTTRANS.FOR.

The source code for each program is shown below. The data file TIMEPT.NPT contains base case parameter values.

**TIMEPT.NPT:**

```
nkd  nks  ipr
 20   20   10
xkmin  xkmax
 -7    -3
nper
 6
npmin  npmax
 1    32
 33   41
 42   43
 44   45
 46   47
 48   70
Dike1cond  Dike2cond  Dike1length  Dike2length  Dike3length  Dike1thick  Dike2thick  Siltthick
 9.0e-5       1e-7       10370         2180         2375          40          40         71.42
SpecificYld  DT  Dikeheigth  Theta  Time  Numpipe  Hold  Ret
 0.20       0.10    25        0.5    70      1      10.2   1
Covthick  Hriv  Siltcond  Covcond  Hlake  Hgrav   Rech  Area     Pump
 1.5      10.20  1.0e-6     1e-6    10.20  11.80   10.0 9713880.0  0.0
 0.0  pxoverb
 5.00  hpipe
 1.5       9.74  1.0e-6     1e-6    11.74  11.44   10.0 9713880.0  0.0
 0.0  pxoverb
 5.00  hpipe
 1.5       9.74  1.0e-6     1e-6    11.74  11.44    9.0 9713880.0  0.0
 0.0  pxoverb
 5.00  hpipe
 1.5       9.74  1.0e-6     1e-6    11.74  11.44    7.0 9713880.0  0.0
 0.0  pxoverb
 5.00  hpipe
 1.5       9.74  1.0e-6     1e-6    11.74  11.44    5.0 9713880.0  0.0
 0.0  pxoverb
 5.00  hpipe
 1.5       9.74  1.0e-6     1e-8     9.74  11.34    1.0 9713880.0  0.0
 0.0  pxoverb
 5.00  hpipe
```

*Siltcond is vertical silt conductivity under the landfill. To find anisotropy, divide Dike1cond by Siltcond.*
*Head in gravel is taken from average of regional flow model for cells under the landfill.*

All parameter descriptions and formats are identical to the water balance model described above except for the variable *Ret* which is the unit-less retardation factor for chemicals being modeled which may sorb to soil particles. A complete description on the use of this file is presented in the flux calculation portion above.

## PTIMEPT.FOR:

```
real covthick,hriv,siltcond,covcond,hlake,hgrav,rech,area,
&    pkoverb,hpipe,d1c,d2c,dike1length,dike2length,
&    dike3length,dike1thick,dike2thick,siltthick,specyld,
&    dt,dikeheight,theta,time,hold,pump,retard
integer numpers,npmin,npmax,numpipes
dimension npmin(100),npmax(100),pkoverb(100,100),hpipe(100,100),
&    covthick(100),hriv(100),siltcond(100),pump(100),
&    covcond(100),hlake(100),hgrav(100),rech(100),area(100)

open(100,file='time.npt',status='old')
open(200,file='time.in',status='unknown')
open(300,file='nontime.in',status='unknown')

read(100,*)
read(100,*)
read(100,*)
read(100,*)
read(100,*)

read(100,*)numpers
write(300,*)numpers
read(100,*)
do 5, i=1,numpers
  read(100,*)npmin(i),npmax(i)
write(300,*)npmin(i),npmax(i)
5   continue
  read(100,*)
  read(100,*)d1c,d2c,dike1length,dike2length,
&    dike3length,dike1thick,dike2thick,
&    siltthick,specyld,dt,dikeheight,theta,
&    time,numpipes,hold,retard
  write(300,*)d1c,d2c,dike1length,dike2length,
&    dike3length,dike1thick,dike2thick,
&    siltthick,specyld,dt,dikeheight,theta,
&    time,numpipes,hold,retard
  read(100,*)
  do 15, i=1,numpers
    read(100,*)covthick(i),hriv(i),siltcond(i),covcond(i),
&       hlake(i),hgrav(i),rech(i),area(i),pump(i)
    do 25, j=1,numpipes
      read(100,*)pkoverb(j,i)
      read(100,*)hpipe(j,i)
25   continue
15   continue
  do 35, i=1,numpers
    do 45, j=npmin(i),npmax(i)
      write(200,*)covthick(i),hriv(i),siltcond(i),covcond(i),
&         hlake(i),hgrav(i),rech(i),area(i),pump(i),j
45     continue
    do 55, k=1,numpipes
      write(200,*)pkoverb(k,i)
      write(200,*)hpipe(k,i)
55     continue
35   continue
  stop
  end
```

## TDIMPT.FOR:

```
c Program for zero dimensional St. Johns Landfill Model
c Particle Tracking Group
c Thomas Lowry - Portland State University - 4/95
c
c List of Variables:
c theta - weighting factor between successive time steps
c hold - head in landfill in current time step
c hnew - head in landfill in next time step
c hriv - head in sloughs
c hlake - head in Smith and Bybee Lakes
c hgrav - head in gravel aquifer
c hpipe - array of head levels in perimeter drains
c dike1cond - conductivity in natural dike
c dike2cond - conductivity in engineered dike
c siltcond - conductivity in silt
c covcond - conductivity of daily cover
c pipecond - array of effective conductivities for perimeter
drains
c dike1length - length of natural dike
c dike2length - length of engineered dike bordering slough
c dike3length - length of engineered dike bordering Smith
Lake
c dike1thick - thickness of natural dike
c dike2thick - thickness of engineered dike
c covthick - thickness of daily cover
c siltthick - harmonic mean of silt thickness data
c pipethick - array of effective thicknesses for perimeter
drains
c area - landfill area
c pipearea - array of effective areas for perimeter drains
c rech - recharge
c specyld - specific yield of landfill
c dt - time step size
c dikeheight - elevation of top of dikes
c time - total simulation time in years
c numpipe - number of perimeter drain segements
c conv - seconds per year
c conv2 - cm/sec to ft/day
c conv3 - gal/min to cfs

      parameter(pipes=10,times=100,conv=365*24*60*60,
     &     conv2=24*60*60/30.48,conv3=.002228)
      real pkoverb,hpipe,bp,cp,hp,pa,
     &  covthick,hriv,siltcond,covcond,hlake,hgrav,rech,
     &  theta,hold,hnew,dike1cond,dike2cond,dike1length,
     &  dike2length,dike3length,dike1thick,dike2thick,
     &  siltthick,specyld,dt,area,pump,pmp,pp,pkb,
     &  dikeheight,bt,ct,dd1,ad1,dd2,ad2,ad3,ds,d7,ce,cpu,
     &  cd1,cd2,bs,bbp,ccp,a7,b7,c7,cd3,afactor,bfactor,
     &  cfactor,d1c,d2c,rch
      integer time,numpipe,ii,jj,nper,npmin,npmax
      dimension pkoverb(pipes,times),hpipe(pipes,times),
     &  bp(pipes),cp(pipes),hp(pipes),pa(pipes),
     &  covthick(times),hriv(times),
     &  siltcond(times),covcond(times),hlake(times),
     &  hgrav(times),rech(times),area(times),
     &  npmin(times),npmax(times),pump(times)

      open(100,file='nontime.in',status='old')
      open(200,file='time.in',status='old')
c
c Read in initial and constant parameters
c
      read(100,*)nper
      do 3, i=1,nper
        read(100,*)npmin(i),npmax(i)
3     continue
      read(100,*)d1c,d2c,dike1length,dike2length,
     &     dike3length,dike1thick,dike2thick,
     &     siltthick,specyld,dt,dikeheight,theta,
     &     time,numpipe,hold
      dike1cond=d1c*365*conv2
      dike2cond=d2c*365*conv2
c
c Calculate factors which are not time dependent.
c
c Storage
c
      dst=specyld/dt
c
c Silt
c
      ds=1/(siltthick)
c
c Surface Seepage
c
      d7=(dike1length+dike2length+dike3length)/2
c
c Read in time dependent variables
c
      do 23, kk=1,nper
        do 25, j=npmin(kk),npmax(kk)
          read(200,*)covthick(j),hriv(j),siltc,covc,
     &         hlake(j),hgrav(j),rch,area(j),pmp
          rech(j)=rch/12.0
          siltcond(j)=siltc*365*conv2
          covcond(j)=covc*365*conv2
          pump(j)=pmp*conv3*conv
25      continue
        if(numpipe.eq.0)goto 36
        do 35, i=1,numpipe
          read(200,*)pkb
          read(200,*)hhpp
          do 33, ll=npmin(kk),npmax(kk)
            pkoverb(i,ll)=pkb*conv
            hpipe(i,ll)=hhpp
33        continue
35      continue
36      continue
23    continue
c
c Start time loop
c
      ii=1
      jj=1
      ddt=0.0
      numsteps=time/dt
      do 45, i=1,numsteps
        if(ddt.gt.ii)then
          ctt=covthick(ii+1)
          hr=hriv(ii+1)
          sc=siltcond(ii+1)
          cc=covcond(ii+1)
          hl=hlake(ii+1)
          hg=hgrav(ii+1)
          rc=rech(ii+1)
          ar=area(ii+1)
          pp=pump(ii+1)
        if(numpipe.eq.0)goto 56
          do 55, k=1,numpipe
          pa(k)=pkoverb(k,ii+1)
          hp(k)=hpipe(k,ii+1)
55      continue
```

250

```
56  continue
      ii=ii+1
      else
      ctt=covthick(ii)
      hr=hriv(ii)
      sc=siltcond(ii)
      cc=covcond(ii)
      hl=hlake(ii)
      hg=hgrav(ii)
      rc=rech(ii)
      ar=area(ii)
      pp=pump(ii)
      if(numpipe.eq.0)goto 66
      do 65, k=1,numpipe
        pa(k)=pkoverb(k,ii)
        hp(k)=hpipe(k,ii)
65    continue
66    continue
      endif
      if(ddt.gt.(time-1))then
      ctt=covthick(ii)
      hr=hriv(ii)
      sc=siltcond(ii)
      cc=covcond(ii)
      hl=hlake(ii)
      hg=hgrav(ii)
      rc=rech(ii)
      ar=area(ii)
      pp=pump(ii)
      if(numpipe.eq.0)goto 68
      do 67, k=1,numpipe
        pa(k)=pkoverb(k,ii)
        hp(k)=hpipe(k,ii)
67    continue
68    continue
      ii=ii-1
      endif
c
c Calculate time dependent factors
c
c Storage factors
c
      bt=dst*(area(ii+1)*theta+ar*(1-theta))
      ct=-bt*hold
c
c Recharge
c
      ce=(rech(ii+1)*area(ii+1)*theta+
     &   rc*ar*(1-theta))
c
c Natural Dike
c
      dd1=(dike1cond*dike1length)/(2*(dike1thick))
      if(hold.gt.dikeheight)then
       ad1=0
       bd1=(dikeheight+hriv(ii+1))*theta*dd1
       cd1=(-hriv(ii+1)*dikeheight*theta-
hriv(ii+1)*hriv(ii+1)*theta
     &       +hold*dikeheight*(1-theta)-hr*dikeheight*(1-
theta)
     &       +hold*hr*(1-theta)-hr*hr*(1-theta))*dd1
      else
       ad1=(theta*dd1)
       bd1=0
       cd1=dd1*(-
hriv(ii+1)*hriv(ii+1)*theta+hold*hold*(1-theta)
     &       -hr*hr*(1-theta))
```

```
      endif
c
c Engineered Dike to Slough
c
      dd2=(dike2cond*dike2length)/(2*(dike2thick))
      if(hold.gt.dikeheight)then
       ad2=0
       bd2=(dikeheight+hriv(ii+1))*theta*dd2
       cd2=(-hriv(ii+1)*dikeheight*theta-
hriv(ii+1)*hriv(ii+1)*theta
     &       +hold*dikeheight*(1-theta)-hr*dikeheight*(1-
theta)
     &       +hold*hr*(1-theta)-hr*hr*(1-theta))*dd2
      else
       ad2=(theta*dd2)
       bd2=0
       cd2=dd2*(-
hriv(ii+1)*hriv(ii+1)*theta+hold*hold*(1-theta)
     &       -hr*hr*(1-theta))
      endif
c
c Silt
c
      bs=ds*(siltcond(ii+1)*area(ii+1)*theta)
      cs=ds*(-
siltcond(ii+1)*hgrav(ii+1)*area(ii+1)*theta+sc*hold*
     &   ar*(1-theta)-sc*hg*ar*(1-theta))
c
c Perimeter Drains
c
      bbp=0.0
      ccp=0.0
      if(numpipe.eq.0)goto 76
      do 75, k=1,numpipe
       if(hold.le.hp(k))then
        bp(k)=0
        cp(k)=0
       else
        bp(k)=pkoverb(k,ii+1)*theta
        cp(k)=-pkoverb(k,ii+1)*hpipe(k,ii+1)*theta
     &        +pa(k)*hold*(1-theta)-pa(k)*hp(k)*(1-theta)
        bbp=bbp+bp(k)
        ccp=ccp+cp(k)
       endif
75   continue
76   continue
c
c Surface Seepage
c
      if(hold.lt.dikeheight)then
      a7=0
      b7=0
      c7=0
      else
      a7=(covcond(ii+1)*d7*theta)/(covthick(ii+1))
      b7=(-
2*covcond(ii+1)*dikeheight*theta*d7)/(covthick(ii+1))
      c7=((covcond(ii+1)*dikeheight*dikeheight*theta)/
     &   (covthick(ii+1))
     &   +((cc*(1-theta)*hold*hold)/(ctt))
     &   -((2*cc*dikeheight*(1-theta)*hold)/(ctt))
     &   +((cc*dikeheight*dikeheight*(1-theta))/(ctt)))*d7
      endif
c
c Engineered Dike to Lake
c
      dd3=(dike2cond*dike3length)/(2*(dike2thick))
```

251

```fortran
      if(hold.gt.dikeheight)then
        ad3=0
        bd3=(dikeheight+hriv(ii+1))*theta*dd3
        cd3=(-hriv(ii+1)*dikeheight*theta-
hriv(ii+1)*hriv(ii+1)*theta
     &      +hold*dikeheight*(1-theta)-hr*dikeheight*(1-
theta)
     &      +hold*hr*(1-theta)-hr*hr*(1-theta))*dd3
      else
        ad3=(theta*dd3)
        bd3=0
        cd3=dd3*(-
hlake(ii+1)*hlake(ii+1)*theta+hold*hold*(1-theta)
     &      -hl*hl*(1-theta))
      endif
c
c Pumps
c
      cpu=pp*theta+pump(ii+1)*(1-theta)
c
c Consolidate factors and solve for hnew
c
      afactor=-ad1-ad2-a7-ad3
      bfactor=-bt-bs-bbp-b7-bd1-bd2-bd3
      cfactor=-ct+ce-cd1-cd2-cs-ccp-c7-cd3-cpu

      hnew=(-bfactor-sqrt((bfactor*bfactor)-
4*afactor*cfactor))
     &     /(2*afactor)
c
c Write results for use in transport model
c
      open (48,file='trans.npt',status='unknown')
      write (48,*) hold,ddt

      ddt=ddt+dt
      hold=hnew
 6000    format (30(1x,f9.6))
   45 continue
      stop
      end
```

## SILTTRANS.FOR:
```fortran
c
c File to produce solute breakthrough times for silt below
landfill.
c Must run ptimept, tdimpt and then silttrans.
c Particle Tracking Group
c Thomas Lowry - Portland State University 4/94
c
      parameter(nmax=70,maxsteps=70000,conv=365*24*60*6
0,
     &
conv2=24*60*60/30.48,nrow=80,ncol=130,nlay=32)
      real recharge,porosity,velocity,ddt,siltc,covc,pmp,rch
      dimension covthick(nmax),hriv(nmax),
     &    hlake(nmax),hgrav(nmax),area(nmax),
     &    npmin(nmax),npmax(nmax),recharge(nmax),
     &    hnew(maxsteps),siltthick(ncol,nrow),
     &    itype(ncol,nrow,nlay),ktype(ncol,nrow),
     &    tottime(ncol,nrow),siltcond(nmax),totx(ncol,nrow),
     &    yesno(ncol,nrow)
c
c   Open input files
c
      open(100,file='nontime.in',status='old')
      open(200,file='time.in',status='old')
      open(300,file='trans.npt',status='old')
      open(400,file='silttimes.out',status='unknown')
      open(700,file='siltdist.out',status='unknown')
      open(800,file='x.out',status='unknown')
      open(900,file='y.out',status='unknown')

open(950,file='~/zdimtrans/silt10/siltthick.out',status='old'
)
      open(960,file='yesno.out',status='unknown')
c
c Read siltthickness under the landfill
c
      do j=1,nrow
        do i=1,ncol
          read(950,*)siltthick(i,j)
        end do
      end do
      close(950)
c
c   Read input
c
      read(100,*)nper
      do 5, i=1,nper
        read(100,*)npmin(i),npmax(i)
    5   continue
      read(100,*)d1c,d2c,dike1length,dike2length,
     &    dike3length,dike1thick,dike2thick,
     &    st,specyld,dtime,dikeheight,theta,
     &    alltime,numpipe,hold,retard
      dike1cond=d1c*365*conv2
      dike2cond=d2c*365*conv2
      close(100)

      do 15, kk=1,nper
        do 25, j=npmin(kk),npmax(kk)
          read(200,*)covthick(j),hriv(j),siltc,covc,
     &        hlake(j),hgrav(j),rch,area(j),pmp
          recharge(kk)=rch/12.0
          siltcond(kk)=siltc*365*conv2
   25     continue
        if(numpipe.eq.0)goto 36
```

```fortran
      do 35, i=1,numpipe
        read(200,*)pkb
        read(200,*)hhpp
35      continue
36      continue
15    continue
      close(200)

      do 45,i=1,alltime/dtime
        read(300,*)hnew(i),ddt
45    continue
      close(300)
c
c     Run across every landfill cell
c
      do 85, i=1,ncol
        do 95, j=1,nrow
c
c     Initialize parameters
c
          porosity=0.40
          totx(i,j)=0.0
          tottime(i,j)=0.0
          k=0
          kk=1
          ncount=0
          velocity=0.0
c
c     If cell is not part of landfill, skip it
c
          if(siltthick(i,j).eq.0)goto 1005
c
c     If distance is less than dike thickness
c
55        k=k+1
          if(totx(i,j).lt.siltthick(i,j).and.tottime(i,j).le.
     &        alltime)then
c
c     Step time dependent parameters one step
c
            if(k*dtime.gt.npmax(kk))kk=kk+1
            if(kk.gt.nper)kk=nper
            recharge(kk)=0
c
c     Calculate velocity at each node
c
            velocity=(((hnew(k)-hgrav(kk))/siltthick(i,j))*
     &          (siltcond(kk)/porosity))/retard
            totx(i,j)=totx(i,j)+velocity*dtime
            if(totx(i,j).lt.0)totx(i,j)=0.
            tottime(i,j)=tottime(i,j)+dtime
            goto 55
          else
            if(totx(i,j).ge.siltthick(i,j))yesno(i,j)=1
            if(tottime(i,j).gt.alltime)yesno(i,j)=0
            goto 1005
          endif
1005      continue
95      end do
85    end do
c
c     Produce Output
c
      do 105, j=1,nrow
        do 115 i=1,ncol
          write(400,403)tottime(i,j)
          write(700,403)totx(i,j)
          write(960,403)yesno(i,j)
115     end do
105   end do
403   format(30f10.6)

      do 125, j=nrow,1,-1
        do 135, i=1,ncol
          write(900,801)j
          write(800,801)i
135     end do
125   end do
801   format(30i3)

      stop
      end
```

## ZDIMTRANS.FOR:

```fortran
c
c File to produce Dike breakthrough curves in 1d.  Need to
run ptimept, tdimpt
c and then zdimtrans in that order. Output will be a
stacked file of retardation
c ranging from 1 to 100 by 10's.
c

      parameter(nmax=70,maxsteps=70000,conv=365*24*60*6
     0,
     &    conv2=24*60*60/30.48)
      real recharge,porosity,head,velocity,a,b,c,siltc,covc,
     &    rch,pmp
      dimension covthick(nmax),hriv(nmax),
     &    hlake(nmax),hgrav(nmax),area(nmax),
     &    npmin(nmax),npmax(nmax),recharge(nmax),
     &    hnew(maxsteps),ddt(maxsteps)
c
c    Open input files
c
      retard=1.0
      open(400,file='trans.out',status='unknown')
      open(500,file='time.out',status='unknown')
      open(600,file='distance.out',status='unknown')
      open(700,file='num.out',status='unknown')
655   continue
      open(100,file='nontime.in',status='old')
      open(200,file='time.in',status='old')
      open(300,file='trans.npt',status='old')
c
c    Initialize variables
c
      porosity=0.4
      dt=0.1
      dx=0.1
c
c    Read input
c
      read(100,*)nper
      do 5, i=1,nper
        read(100,*)npmin(i),npmax(i)
5     continue
      read(100,*)d1c,d2c,dike1length,dike2length,
     &    dike3length,dike1thick,dike2thick,
     &    siltthick,specyld,dtime,dikeheight,theta,
     &    alltime,numpipe,hold,buff
      dike1cond=d1c*365*conv2
      dike2cond=d2c*365*conv2

      do 15, kk=1,nper
        do 25, j=npmin(kk),npmax(kk)
          read(200,*)covthick(j),hriv(j),siltc,covc,
     &        hlake(j),hgrav(j),rch,area(j),pmp
          recharge(j)=rch/12.0
25      continue
        if(numpipe.eq.0)goto 36
        do 35, i=1,numpipe
          read(200,*)pkb
          read(200,*)hhpp
35      continue
36      continue
15    continue

      do 45,i=1,alltime/dtime
        read(300,*)hnew(i),ddt(i)
45    continue
```

```fortran
      close(100)
      close(200)
      close(300)
c
c    Begin loop until total thickness of dike is covered
c
      totx=0.0
      tottime=0.0
      kk=1
      ncount=0
      velocity=0.0
      time=0.0
      num=0
c
c    If distance is less than dike thickness
c
55    if(totx.lt.dike1thick.and.tottime.lt.alltime)then
        if(velocity.eq.0.)then
          totx=0.0
        endif
        if(ncount.eq.0)goto 56
        write(400,401)totx,tottime,head,velocity,hhnew
        num=num+1
        write(500,501)tottime
        write(600,501)totx-1
401     format(5f10.4)
501     format(f10.4)
56      continue
c
c    Step time dependent parameters one step
c
        if(tottime.gt.npmax(kk))kk=kk+1
        if(kk.gt.nper)kk=nper
        recharge(kk)=0
c
c    Match heads to correspond with total elapsed time
c
        do jj=1,alltime/dtime
          if(tottime.lt.ddt(jj+1).and.tottime.ge.ddt(jj))then
            hhnew=(hnew(jj)*(ddt(jj+1)-
tottime)+hnew(jj+1)*
     &          (tottime-ddt(jj)))/(ddt(jj+1)-ddt(jj))
            goto 65
          else
            continue
          endif
        end do
65      continue

        if(ncount.eq.0)then
          time=0.0
          goto 75
        endif

        if(velocity.eq.0.0)then
          time=time+dtime
        else
          time=dx/velocity
        endif

75      continue
        ncount=ncount+1
c
c    If travel distance is less than dx
c
        if(time.ge.dt)then
          x=velocity*dt
```

```
            totx=totx+x
            x=totx
            tottime=tottime+dt
            a=-2*recharge(kk)/dike1cond

b=(hriv(kk)*hriv(kk)/dike1thick)+(2*recharge(kk)*dike1th
ick)
      &        /dike1cond-(hhnew*hhnew)/dike1thick
            c=hhnew*hhnew
            head=sqrt(a*x*x+b*x+c)
            velocity=(-
(a*x+b/2)*(dike1cond/(head*porosity)))/retard
            goto 55
          endif
c
c     If travel distance is more than dx
c
          if(time.lt.dt)then
            x=totx
            a=-2*recharge(kk)/dike1cond

b=(hriv(kk)*hriv(kk)/dike1thick)+(2*recharge(kk)*dike1th
ick)
      &        /dike1cond-(hhnew*hhnew)/dike1thick
            c=hhnew*hhnew
            head=sqrt(a*x*x+b*x+c)
            velocity=(-
(a*x+b/2)*(dike1cond/(head*porosity)))/retard
            if(hriv(kk).eq.hhnew)velocity=0.0
            totx=totx+dx
            tottime=tottime+time
            goto 55
          endif
        endif

      write(700,'(i6,f10.2)')num,retard

      if(retard.ge.10.)retard=retard+10.
      if(retard.eq.1.)retard=retard+9.
      if(retard.gt.100.)goto 555
      goto 655
 555  continue
      stop
      end
```

**Appendix B    - Two-dimensional Site specific mounding model Source Code and Documentation**

The intent of the two-dimensional site specific mounding model is to create a time varying and spatially varying contaminant mound within the landfill. MODFLOW is used to perform the flow calculations but there are several preprocessors that are needed to set up the model properly. The preprocessors are used to input the stratigraphy and geologic layout provide conductance values for each cell in the grid (see Section 6.3 for a description of the setup of the two-dimensional flow model), provide initial and boundary conditions, and input other hydrologic data such as recharge and slough levels. The setup is very similar to the 3d flow model as described in 7.3, but is modified to provide for the two-dimensional layout.

MODFLOW is broken into a series of subprograms each designed to perform a specific task such as handling river cells, general head boundary conditions, drain cells, and so on. The detailed structure of the MODFLOW code will only be discussed here as it pertains to this project and the revisions that were needed for this project. The MODFLOW main program is version #1638, dated July 24, 1987. Each subroutine and subprogram have their own revision dates and version numbers, with the form of each being compatible with the above stated main program. Some changes were made to the MODFLOW code to incorporate the expansion of the landfill over time as well as to provide an output form that is more useful for our post-processing. In the BCF2.FOR subprogram, a subroutine, which activates cells in the landfill as the landfill grows over time, was added. It reads an input file called

ADDCELL.NPT which tells the model when cells become activated depending upon the growth of the landfill. The BAS1.FOR subprogram also has an addition that formats output so that it is easier to use for post-processing. The changes to the MODFLOW code are shown below with the line number at the start of the change in the subprogram where they are added. Complete documentation of the MODFLOW code can be found in the MODFLOW users manual titled A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model (McDonald and Harbaugh, 1988).

Six separate preprocessors must be run before the MODFLOW executable can be run. Four of the six set up the stratigraphy and the geology to fit the model domain (GRID2D.FOR, SILT_THICK.FOR, SILT_TOP.FOR, and R1ADD.FOR). Another program (RIV2D.FOR) is used to convert the landfill cells to river cells for each stress period in the simulation and determine the conductance for each cell. The last preprocessor (MF2D.FOR) uses the output from the previous five programs and rearranges them into a form which is then useable by MODFLOW. The program structure is shown below in Figure 91.

SILT_THICK.FOR and SILT_TOP.FOR are used to define the thickness and the top of the silt layer respectively. SILT_THICK.FOR produces a file with one value for each cell that describes the thickness of the silt at that cell. SILT_TOP.FOR describes the top of the silt layer at that cell. GRID2D.FOR produces a three-dimensional file which describes the geology in the landfill region based on the grid layout of the three-dimensional flow model. The geology layout is later converted to a

standard two-dimensional format by RIV2D.FOR. It is done this way because it is easier to share data files between the two-dimensional flow model and the three-dimensional flow model if the grid structure is kept the same. SILT_THICK.FOR, SILT_TOP.FOR and GRID2D.FOR all rely on the data files supplied to us by METRO that describes the layout of each geologic layer. These three files read the smaller scale data files and using a linear interpolation scheme to fit them to the larger scale model grid. For the SILT_THICK.FOR file, it uses the difference between the top of the silt layer and the top of the gravel layer to calculate the silt thickness under the landfill.
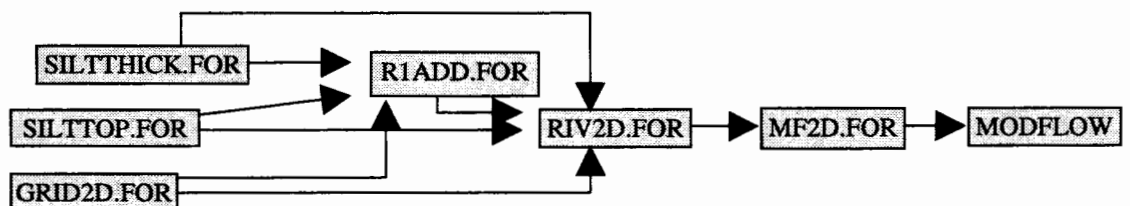


**Figure 91 - Program structure for two-dimensional site specific mounding model. The first 5 programs setup the geology, hydrology, and model structure for the entire simulation. MF2D.FOR places the data in a form which can then be used in MODFLOW.**

R1ADD.FOR uses the re-formatted data files produced by SILT_THICK.FOR, SILT_TOP.FOR, and GRID2D.FOR to produce a file describing all cells in the model domain which coincide with surface water bodies, assigning a grid address, a conductance, and a typing code (i.e. - Columbia Slough, North Slough, etc.) for each river cell. RIV2D.FOR then uses all the output from the previous four programs to produce a new geology file which has the landfill as the only active cells in the model domain and designates all those cells as river cells, each with its own conductance

based on its position in the landfill. Finally, MF2D.FOR utilizes the output from RIV2D.FOR and places the data in a form that is then used by the MODFLOW code to perform the simulation.

In addition to the above programs, an additional program (BCHEAD.FOR) is needed which converts the output from the regional flow model into the grid layout for either the two-dimensional flow model or the three-dimensional flow model. It also computes the average head of all the cells in the gravel layer below the landfill for the use as the boundary condition in the water balance model. It is shown in Appendix C in the form used to produce the needed data files for the three-dimensional flow model and is not shown here.

An input file called FLOWFILE is used to direct MODFLOW to the correct files and output schemes. The easiest way to run the two-dimensional site specific mounding model is to run a batch file which contains the above files in succession. The batch file should look like:

```
SILT_THICK
SILT_TOP
GRID2D
R1ADD
RIV2D
MF2D
MF < FLOWFILE > MF.OUT
```

All files listed above are the executable files of the *.FOR files of the same name. The exception is the MF file that is the name of the MODFLOW executable. The MF.OUT is an output file that will contain screen output produced by MODFLOW.

Total file space needed is about 125Mb of disk space including all the input files needed as well as the output files produced. The source code, with explanations of input, execution methods, and key variables are shown below.

| Program Name | Input Files | Output Files | Description of Output |
|---|---|---|---|
| SILT_TOP.FOR | SILT | CHECK.DAT | Verification file to make sure grid is set correctly |
| | | SILTTOP.DAT | Data file of top of silt elevation at each cell address |
| SILT_THICK.FOR | SILT<br>GRAV | CHECK.DAT | Verification file to make sure grid is set correctly |
| | | SILTTHICK.DAT | Data file of silt thickness at each cell address |
| GRID2D.FOR | WATER<br>LANDFILL<br>SILT<br>SAND<br>GRAV<br>TROUT<br>UNDIFF<br>SUBAREA<br>ROAD<br>DIKE | CHECK.DAT<br><br>GTYPE2D.OUT | Verification file to make sure grid is set correctly<br>Data file delineating the geology within the model domain, assigning a 'type' code to each cell address depending upon where it lies in the physical domain. |
| R1ADD.FOR | GTYPE2D.OUT<br>SILTTOP.OUT<br>SILTTHICK.OUT<br>RIVERND2.CJB | CHECK.DAT<br><br>RIV.NPT<br><br>STAGE.DAT | Verification file to make sure grid is set correctly<br>Cell addresses describing all surface water bodies<br>Water level data over time. |
| RIV2D.FOR | GTYPE2D.OUT<br>SILTTHICK.OUT<br>SILTTOP.OUT<br>LF.NPT<br>RIV.NPT<br>RIVLEV.DAT<br>BCPRE82.NPT<br>BC82_97.NPT<br>BCPOST97.NPT | RIVSPA.NPT<br><br><br>THICK.OUT<br>LFBOT.OUT<br><br>2DGTYPE.OUT<br><br><br><br>LFTOP.OUT | Data file describing each landfill cell as a river cell with its own conductance and water elevation.<br>Landfill thickness at each cell address<br>Data file of bottom of landfill elevations<br>2D description of each cell in the model domain. Delineates subareas of landfill with all other cells designated as inactive.<br>Top of landfill elevations |
| MF2D.FOR | 2DGTYPE.OUT<br>BCPRE82.NPT<br>THICK.OUT<br>LFBOT.OUT<br>LFTOP.OUT | CHECK.DAT<br>BAS.NPT<br><br>BCF2.NPT<br>BALANCE.NPT<br><br>OC.NPT<br>PCG2.NPT<br>RCH.NPT<br>DRN.NPT<br>ADDCELL.NPT<br><br>RIV2.NPT | Verification file.<br>Data for basic input used in MODFLOW<br>Data for block centered flow package<br>File used in post-processing for assessing water budget<br>Output control file<br>Data for conjugate gradient package<br>Data for recharge package<br>Data for drain package<br>Data describing growth of landfill over time<br>Same as rivspa.npt |

**Table 6 - Listing of programs, needed files to run programs, and output files with descriptions for the 2d site specific mounding model. The needed data files listed without an extension (i.e. GRAV) are files delinieating the geology of the area and are supplied by METRO for our use in the project. The BC\*.NPT files are generated from results of the regional flow model by a post-processor for that model called BCHEAD.FOR. It is explained in Appendix C.**

## SILTTOP.FOR

```
c  Silt_top.for - Program determines top elevation of silt
layer
c  Portland State University - 1994 - Chris Berger
c  As modified - June, 1995 - Thomas Lowry

     integer hgu
     parameter(ix=200,jy=200,lay=60,hgu=7,dltx=15.0)
     parameter(ncol=1513,nrow=1783)
     dimension xd(ix),yd(jy),zd(lay),xs(ix),ys(jy),zs(lay)
     dimension xint(ix),yint(jy),zint(lay)
     dimension ztemp(hgu),siltz(ix,jy)
     integer siltrow(ncol)

     kk=hgu

c  initializing 'type' array to zero which will be
c  used as a flag later to help determine if grid pt. has
been missed

     do i=1,ix
       do j=1,jy
         siltz(i,j)=0.0
       end do
     end do

c  Assinging # of boundaries of constant cell length along
x-axis (ic),
c  along y-axis (ir), , along z-axis (il)

     ici=2
     iri=2
     ili=4

c  (xm,ym) - Assigning location of origin in terms of
model coordinate
c  system

     xm=8100
     ym=15300

c  Assigning distances of intervals of constant cell width
along x-axis from
c  point defined by xm

     xs(1)=0
     xs(2)=7650
     xint(1)=50

c  Assigning distances of intervals of constant cell width
along y-axis from
c  point defined by ym

     ys(1)=0
     ys(2)=5850
     yint(1)=50

c  Assigning elevations of intevals of constant cell width
along z-axis

     zs(1)=95
     zs(2)=-30
     zs(3)=-50
     zs(4)=-350
     zint(1)=5
     zint(2)=10
     zint(3)=25
```

```
c  counting number of cell boundaries

     ic=1
     ir=1
     il=1
     do i=2,ici
       ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
     end do
     do j=2,iri
       ir=ir+nint((ys(j)-ys(j-1))/yint(j-1))
     end do
     do k=2,ili
       il=il+nint((zs(k-1)-zs(k))/zint(k-1))
     end do

c calculating location of cell boundary along axis

     xd(1)=xs(1)
     yd(1)=ys(1)
     zd(1)=zs(1)
     icnt=1
     jcnt=1
     kcnt=1
     do i=2,ic
       xd(i)=xd(i-1)+xint(icnt)
       if(nint(xd(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
     end do
     do j=2,ir
       yd(j)=yd(j-1)+yint(jcnt)
       if(nint(yd(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
     end do
     do k=2,il
       zd(k)=zd(k-1)-zint(kcnt)
       if(nint(zd(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
     end do

     open(91,file='check.dat',status='unknown')

     write(91,*)ic,ir,il
     write(91,*)
     do i=1,ic
       write(91,*)xd(i)
     end do
     write(91,*)
     do i=1,ir
       write(91,*)yd(i)
     end do
     write(91,*)
     do i=1,il
       write(91,*)zd(i)
     end do


     open(3,file='~/modflow/largegrid/landdat/silt',status='old'
     )

c  skipping first six lines of silt data file

     do l=1,6
       read(3,*)
     end do

     do 300 ii=nrow,1,-1

       read(3,*)(siltrow(jj), jj=1,ncol)
```

262

```
      do 305 jj=1,ncol
       ztemp(3)=real(siltrow(jj))

       xx=real(jj-1)*dltx
       yy=real(ii-1)*dltx

c   translating from model grid coordinates to origin
(upper-left corner)
c    defined by computational grid

       x=xx-xm
       y=ym-yy

c   determining if data grid point has corresponding cell
center
c   scanning through the coordinates of all the model cell
boundaries
c   and determing if any cells are centered about current
read data coordinates

       do i=1,ic-1
        xtemp=(xd(i)+xd(i+1))/2.0
        ixdist=int(xtemp/dltx)*int(dltx)
        if(ixdist.eq.nint(x))then
          do j=1,ir-1
            ytemp=(yd(j)+yd(j+1))/2.0
            iydist=int(ytemp/dltx)*int(dltx)
            if(iydist.eq.nint(y))then
              siltz(i,j)=ztemp(3)
            end if
          end do
        end if
       end do
20     continue
305    continue
300    continue

       open(31,file='silttop.dat',status='unknown')
       do j=1,ir-1
        write(31,'(30f7.1)')(siltz(i,j), i=1,ic-1)
       end do

       stop
       end
```

SILTTOP.FOR defines the top of the silt layer for each cell in the grid. The top of silt elevations are later used as a reference for determining the thickness of the silt as well as the thickness of the landfill. The first part of the program sets up the grid geometry that determines the number of rows, columns, and layers as well as the thickness, width, and height of each cell. This file only needs to be run once and may be skipped for subsequent model sensitivity runs assuming its output file, SILTTOP.DAT, is saved.

# SILTTHICK.FOR

```
c  Silt_thick.for - Program determines distance between
top of gravel and top
c  of silt layers
c  Portland State University - 1994 - Chris Berger
c  As modified - June, 1995 - Thomas Lowry

   integer hgu
   parameter(ix=200,jy=200,lay=60,hgu=7,dltx=15.0)
   parameter(ncol=1513,nrow=1783)
   dimension xd(ix),yd(jy),zd(lay),xs(ix),ys(jy),zs(lay)
   dimension xint(ix),yint(jy),zint(lay)
   dimension ztemp(hgu),dist(ix,jy)
   integer siltrow(ncol),
   -        gravrow(ncol)

   kk=hgu

c  initializing 'type' array to zero which will be
c  used as a flag later to help determine if grid pt. has
been missed

   do i=1,ix
     do j=1,jy
       dist(i,j)=0.0
     end do
   end do

c  Assinging # of boundaries of constant cell length along
x-axis (ic),
c  along y-axis (ir), , along z-axis (il)

   ici=2
   iri=2
   ili=4

c  (xm,ym) - Assigning location of origin in terms of
model coordinate
c  system

   xm=8100
   ym=15300

c  Assigning distances of intervals of constant cell width
along x-axis from
c  point defined by xm

   xs(1)=0
   xs(2)=7650
   xint(1)=50

c  Assigning distances of intervals of constant cell width
along y-axis from
c  point defined by ym

   ys(1)=0
   ys(2)=5850
   yint(1)=50

c  Assigning elevations of intevals of constant cell width
along z-axis

   zs(1)=95
   zs(2)=-30
   zs(3)=-50
   zs(4)=-350
   zint(1)=5
```

```
   zint(2)=10
   zint(3)=25

c  counting number of cell boundaries

   ic=1
   ir=1
   il=1
   do i=2,ici
     ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
   end do
   do j=2,iri
     ir=ir+nint((ys(j)-ys(j-1))/yint(j-1))
   end do
   do k=2,ili
     il=il+nint((zs(k-1)-zs(k))/zint(k-1))
   end do

c calculating location of cell boundary along axis

   xd(1)=xs(1)
   yd(1)=ys(1)
   zd(1)=zs(1)
   icnt=1
   jcnt=1
   kcnt=1
   do i=2,ic
     xd(i)=xd(i-1)+xint(icnt)
     if(nint(xd(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
   end do
   do j=2,ir
     yd(j)=yd(j-1)+yint(jcnt)
     if(nint(yd(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
   end do
   do k=2,il
     zd(k)=zd(k-1)-zint(kcnt)
     if(nint(zd(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
   end do

   open(91,file='check.dat',status='unknown')
   write(91,*)ic,ir,il
   write(91,*)
   do i=1,ic
     write(91,*)xd(i)
   end do
   write(91,*)
   do i=1,ir
     write(91,*)yd(i)
   end do
   write(91,*)
   do i=1,il
     write(91,*)zd(i)
   end do


open(3,file='~/modflow/largegrid/landdat/silt',status='old'
)

open(5,file='~/modflow/largegrid/landdat/grav',status='ol
d')

c  skipping first six lines of each file

   do l=1,6
     read(3,*)
   end do
   do l=1,6
```

```
          read(5,*)
        end do


      do 300 ii=nrow,1,-1

        read(3,*)(siltrow(jj), jj=1,ncol)
        read(5,*)(gravrow(jj), jj=1,ncol)

        do 305 jj=1,ncol
        ztemp(3)=real(siltrow(jj))
        ztemp(5)=real(gravrow(jj))

        xx=real(jj-1)*dltx
        yy=real(ii-1)*dltx

c   translating from model grid coordinates to origin
(upper-left corner)
c   defined by computational grid

        x=xx-xm
        y=ym-yy

c   determining if data grid point has corresponding cell
center
c   scanning through the coordinates of all the model cell
boundaries
c   and determing if any cells are centered about current
read data coordinates

        do i=1,ic-1
        xtemp=(xd(i)+xd(i+1))/2.0
        ixdist=int(xtemp/dltx)*int(dltx)
        if(ixdist.eq.nint(x))then
          do j=1,ir-1
            ytemp=(yd(j)+yd(j+1))/2.0
            iydist=int(ytemp/dltx)*int(dltx)
            if(iydist.eq.nint(y))then
             dist(i,j)=ztemp(3)-ztemp(5)
            end if
          end do
        end if
        end do
20     continue
305    continue
300    continue

      open(31,file='siltthick.dat',status='unknown')
      do j=1,ir-1
        write(31,'(30f7.1)')(dist(i,j), i=1,ic-1)
      end do

      stop
      end
```

SILTTHICK.FOR operates the same as SILTTOP.FOR except it also reads the top of the gravel layer and then calculates the thickness of the silt layer by subtracting the elevation of the top of the silt layer from the elevation of the top of the gravel layer. It too, only needs to run once assuming its output (SILTTHICK.DAT) is saved for subsequent model sensitivity runs.

## GRID2D.FOR

```fortran
c    Portland State University - Chris Berger 1993
c    As modified February, 1996 by Thomas Lowry
c    St. Johns Landfill modeling project

c    This program uses stratigraphy files delineating upper
and lower
c    layer boundaries and assigns each model cell a
number corresponding
c    to the type of geological unit within which the
computation
c    point lies

c    hgu - # of geological units
c    ix - maximum number of computational grid pts. in x-
direction
c    jy - maximum number of computational grid pts. in y-
direction
c    lay - maximum number of layers

     integer hgu
     parameter(ix=200,jy=200,lay=60,hgu=7,dltx=15.0)

     integer type(ix,jy,lay)

     dimension xd(ix),yd(jy),zd(lay),xs(ix),ys(jy),zs(lay)
     dimension xint(ix),yint(jy),zint(lay)

c    initializing 'type' array to zero which will be
c    used as a flag later to help determine if grid pt. has
been missed

     do i=1,ix
       do j=1,jy
         do k=1,lay
           type(i,j,k)=0
         end do
       end do
     end do

c    Assinging # of boundaries of constant cell length along
x-axis (ic),
c    along y-axis (ir), , along z-axis (il)

     ici=2
     iri=2
     ili=4

c    (xm,ym) - Assigning location of origin in terms of
model coordinate
c    system

     xm=8100
     ym=15300

c    Assigning distances of intervals of constant cell width
along x-axis from
c    point defined by xm

     xs(1)=0
     xs(2)=7650
     xint(1)=50

c    Assigning distances of intervals of constant cell width
along y-axis from
c    point defined by ym

     ys(1)=0
     ys(2)=5850
     yint(1)=50

c    Assigning elevations of intevals of constant cell width
along z-axis

     zs(1)=95
     zs(2)=-30
     zs(3)=-50
     zs(4)=-350
     zint(1)=5
     zint(2)=10
     zint(3)=25

c    counting number of cell boundaries

     ic=1
     ir=1
     il=1
     do i=2,ici
       ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
     end do
     do j=2,iri
       ir=ir+nint((ys(j)-ys(j-1))/yint(j-1))
     end do
     do k=2,ili
       il=il+nint((zs(k-1)-zs(k))/zint(k-1))
     end do

c    calculating location of cell boundary along axis

     xd(1)=xs(1)
     yd(1)=ys(1)
     zd(1)=zs(1)
     icnt=1
     jcnt=1
     kcnt=1
     do i=2,ic
       xd(i)=xd(i-1)+xint(icnt)
       if(nint(xd(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
     end do
     do j=2,ir
       yd(j)=yd(j-1)+yint(jcnt)
       if(nint(yd(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
     end do
     do k=2,il
       zd(k)=zd(k-1)-zint(kcnt)
       if(nint(zd(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
     end do

c    Check.dat is a file that prints the coordinates of each
cell boundary
c    It is not used in any calculations but is useful in
checking the
c    accuracy of the grid setup.

     open(91,file='check.dat',status='unknown')
     write(91,*)ic,ir,il
     write(91,*)
     do i=1,ic
       write(91,*)xd(i)
     end do
     write(91,*)
     do i=1,ir
       write(91,*)yd(i)
     end do
```

268

```fortran
      write(91,*)
      do i=1,il
        write(91,*)zd(i)
      end do

c     reading files containing top and bottom of layer info
and assigning
c     layer type # to computational grid cells

      call assign(type,xd,yd,zd,xm,ym,il,ic,ir)

      stop
      end


c     ***************************************************
c     *************************
      subroutine assign(type,xd,yd,zd,xm,ym,il,ic,ir)

      integer hgu
      parameter(ix=200,jy=200,lay=60,hgu=7,dltx=15.0)
      parameter(ncol=1513,nrow=1783)
      integer type(ix,jy,lay)
      dimension xd(ix),yd(jy),zd(lay),ztemp(hgu)
      integer
lfrow(ncol),watrow(ncol),siltrow(ncol),sandrow(ncol),
     -    gravrow(ncol),troutrow(ncol),unconrow(ncol),
     -    subarea(ncol),road(ncol),dike(ncol)
      kk=hgu

      open(1,file='~/modflow/largegrid/landdat/water',
     &    status='old')
      open(2,file='~/modflow/largegrid/landdat/landfill',
     &    status='old')
      open(3,file='~/modflow/largegrid/landdat/silt',
     &    status='old')
      open(4,file='~/modflow/largegrid/landdat/sand',
     &    status='old')
      open(5,file='~/modflow/largegrid/landdat/grav',
     &    status='old')
      open(6,file='~/modflow/largegrid/landdat/trout',
     &    status='old')
      open(7,file='~/modflow/largegrid/landdat/undiff',
     &    status='old')

c     skipping first six lines of each file

      do i=1,hgu
        do l=1,6
          read(i,*)
        end do
      end do

c     initializing layer type number to zero

      do i=1,ic
        do j=1,ir
          do l=1,il
            type(i,j,l)=0
          end do
        end do
      end do

      do 300 ii=nrow,1,-1

        read(1,*)(watrow(jj), jj=1,ncol)
        read(2,*)(lfrow(jj), jj=1,ncol)
        read(3,*)(siltrow(jj), jj=1,ncol)
        read(4,*)(sandrow(jj), jj=1,ncol)
        read(5,*)(gravrow(jj), jj=1,ncol)
        read(6,*)(troutrow(jj), jj=1,ncol)
        read(7,*)(unconrow(jj), jj=1,ncol)


        do 305 jj=1,ncol
          ztemp(1)=real(watrow(jj))
          ztemp(2)=real(lfrow(jj))
          ztemp(3)=real(siltrow(jj))
          ztemp(4)=real(sandrow(jj))
          ztemp(5)=real(gravrow(jj))
          ztemp(6)=real(troutrow(jj))
          ztemp(7)=real(unconrow(jj))

          xx=real(jj-1)*dltx
          yy=real(ii-1)*dltx

c     translating from model grid coordinates to origin
(upper-left corner)
c     defined by computational grid

          x=xx-xm
          y=ym-yy

c     determining if data grid point has corresponding cell
center
c     scanning through the coordinates of all the model cell
boundaries
c     and determing if any cells are centered about current
read data
c     coordinates

          do i=1,ic-1
            xtemp=(xd(i)+xd(i+1))/2.0
            ixdist=int(xtemp/dltx)*int(dltx)
            if(ixdist.eq.nint(x))then
              do j=1,ir-1
                ytemp=(yd(j)+yd(j+1))/2.0
                iydist=int(ytemp/dltx)*int(dltx)
                if(iydist.eq.nint(y))then
                  do k=1,il-1
                    zelev=(zd(k)+zd(k+1))/2.0
                    do l=2,kk
                      if(zelev.le.ztemp(l-1))then
                        type(i,j,k)=l-1
                      end if
                      if(zelev.lt.ztemp(kk))then
                        type(i,j,k)=kk
                      end if
                    end do
15                  continue
                  end do
                  go to 20
                end if
              end do
            end if
          end do
20        continue
305     continue
300   continue

c     reading file defining subareas
```

```
open(27,file='~/modflow/largegrid/landdat/subarea',status
='old')

    do ik=1,6
      read(27,*)
    end do

    do 404 ii=nrow,1,-1
      read(27,*)(subarea(jj), jj=1,ncol)
      do 405 jj=1,ncol
        if(subarea(jj).eq.0)go to 405
        xx=real(jj-1)*dltx
        yy=real(ii-1)*dltx

c    translating from model grid coordinates to origin
(upper-left corner)
c    defined by computational grid

        x=xx-xm
        y=ym-yy

c    determining in which model cell subarea gridpoint is
located

        do i=1,ic-1
          if(x.ge.xd(i).and.x.lt.xd(i+1))then
            do j=1,ir-1
              if(y.ge.yd(j).and.y.lt.yd(j+1))then
                do k=1,il-1
                  if(type(i,j,k).eq.2.or.type(i,j,k).ge.10)then
                    type(i,j,k)=20+subarea(jj)
                  end if
                end do
                go to 405
              end if
            end do
          end if
        end do

405    end do
404  end do

    close(27)

c    reading file defining road


open(28,file='~/modflow/largegrid/landdat/road',status='o
ld')

    do ik=1,6
      read(28,*)
    end do

    do 504 ii=nrow,1,-1
      read(28,*)(road(jj), jj=1,ncol)
      do 505 jj=1,ncol
        if(road(jj).eq.0)go to 505
        xx=real(jj-1)*dltx
        yy=real(ii-1)*dltx

c    translating from model grid coordinates to origin
(upper-left corner)
c    defined by computational grid

        x=xx-xm
```

```
        y=ym-yy

c    determining in which model cell the road gridpoint is
located

        do i=1,ic-1
          if(x.ge.xd(i).and.x.lt.xd(i+1))then
            do j=1,ir-1
              if(y.ge.yd(j).and.y.lt.yd(j+1))then
                do k=1,il-1
                  if(type(i,j,k).eq.2.or.
  &                   type(i,j,k).ge.10)then
                    if(type(i,j,k).eq.24.or.
  &                     type(i,j,k).eq.25)then
                      type(i,j,k)=31
                    else
                      type(i,j,k)=30
                    end if
                  end if
                end do
                go to 505
              end if
            end do
          end if
        end do

505    end do
504  end do

    close(28)

c    reading file defining dike


open(29,file='~/modflow/largegrid/landdat/dike',status='ol
d')

    do ik=1,6
      read(29,*)
    end do

    do 604 ii=nrow,1,-1
      read(29,*)(dike(jj), jj=1,ncol)
      do 605 jj=1,ncol
        if(dike(jj).eq.0)go to 605
        xx=real(jj-1)*dltx
        yy=real(ii-1)*dltx

c    translating from model grid coordinates to origin
(upper-left corner)
c    defined by computational grid

        x=xx-xm
        y=ym-yy

c    determining in which model cell the dike gridpoint is
located

        do i=1,ic-1
          if(x.ge.xd(i).and.x.lt.xd(i+1))then
            do j=1,ir-1
              if(y.ge.yd(j).and.y.lt.yd(j+1))then
                do k=1,il-1
                  if(type(i,j,k).eq.2.or.type(i,j,k).ge.10)then
                    if(zd(k+1).ge.15.0)then
                      type(i,j,k)=40
                    end if
```

270

```
              end if
            end do
          go to 605
         end if
       end do
     end if
   end do

605   end do
604  end do

   close(29)

   open(89,file='test22d.out',status='unknown')
   do i=1,ic-1
     do j=1,ir-1
       do k=1,il-1
         write(89,122)i,j,k,type(i,j,k)
122        format(4(i4))
       end do
     end do
   end do
   close(89)


   return
   end
```

GRID2D defines the geology for each grid in the computational domain. It reads all geology files supplied by METRO and based on the grid layout, assigns a code to each cell which tells the model which geologic type that cell should be. Just like SILT_THICK.FOR, and SILT_TOP.FOR, this program only needs to be run once assuming its output, GTYPE2D.FOR, is saved for subsequent model sensitivity runs.

## R1ADD2D.FOR

```
c R1ADD2D.FOR  - Produces river cell output
distinguishing between
c North, Columbia, and Blind sloughs, Smith and Bybee
Lakes.
c Portland State University - 1994, Chris Berger
c As modified June, 1995 - Thomas Lowry

    parameter
(ncolm=250,nrowm=200,nlaym=60,nperm=20)
    dimension x(ncolm+1),y(nrowm+1),z(nlaym+1),
    *       zz(nrowm,ncolm)
    dimension perlen(nperm)
    dimension itype(ncolm*nrowm)
    real  kcond(10)
    dimension stage(ncolm*nrowm)
    dimension noder(ncolm*nrowm)
    dimension ir(ncolm*nrowm),jr(ncolm*nrowm)
    dimension cond(ncolm*nrowm)
    integer type(ncolm,nrowm,nlaym)
    dimension iunit(24)
    dimension siltz(ncolm,nrowm),dist(ncolm,nrowm)
    dimension
xint(ncolm+1),yint(nrowm+1),zint(nlaym+1)
    dimension xs(ncolm+1),ys(nrowm+1),zs(nlaym+1)
    nper=48
    iunit(13)=14
    open(98,file='~/modflow/2d/landdat/gtype2d.out',
    &   status='old')

c  reading file with grid type #'s

23   read(98,122,end=33)i,j,k,itemp
     type(i,j,k)=itemp
122   format(4(i4))
     go to 23
33   continue

c    Assinging # of boundaries of constant cell length along
x-axis (ic),
c    along y-axis (ir), , along z-axis (il)

     ici=2
     iri=2
     ili=4

c    (xm,ym) - Assigning location of origin in terms of
model coordinate
c    system

     xm=8100
     ym=15300

c    Assigning distances of intervals of constant cell width
along x-axis from
c    point defined by xm

     xs(1)=0
     xs(2)=7650
     xint(1)=50

c    Assigning distances of intervals of constant cell width
along y-axis from
c    point defined by ym

     ys(1)=0
     ys(2)=5850
```

```
     yint(1)=50

c    Assigning elevations of intevals of constant cell width
along z-axis

     zs(1)=95
     zs(2)=-30
     zs(3)=-50
     zs(4)=-350
     zint(1)=5
     zint(2)=10
     zint(3)=25

c    counting number of cell boundaries

     ic=1
     irr=1
     il=1
     do i=2,ici
       ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
     end do
     do j=2,iri
       irr=irr+nint((ys(j)-ys(j-1))/yint(j-1))
     end do
     do k=2,ili
       il=il+nint((zs(k-1)-zs(k))/zint(k-1))
     end do

c calculating location of cell boundary along axis

     x(1)=xs(1)
     y(1)=ys(1)
     z(1)=zs(1)
     icnt=1
     jcnt=1
     kcnt=1
     do i=2,ic
       x(i)=x(i-1)+xint(icnt)
       if(nint(x(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
     end do
     do j=2,irr
       y(j)=y(j-1)+yint(jcnt)
       if(nint(y(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
     end do
     do k=2,il
       z(k)=z(k-1)-zint(kcnt)
       if(nint(z(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
     end do

     open(91,file='check.dat',status='unknown')
     write(91,*)ic,irr,il
     write(91,*)
     do i=1,ic
       write(91,*)x(i)
     end do
     write(91,*)
     do i=1,irr
       write(91,*)y(i)
     end do
     write(91,*)
     do i=1,il
       write(91,*)z(i)
     end do
     close(91)


     ncol=ic-1
```

```fortran
      nrow=irr-1
      nlay=il-1

c initializing bottom of cell elevations to an unreasonable #

      do i=1,nrowm
       do j=1,ncolm
        zz(i,j)=999.0
       end do
      end do

      open(31,file='~/modflow/2d/landdat/silttop.dat',
     &   status='old')
      do j=1,nrow
       read(31,'(30f7.1)')(siltz(i,j), i=1,ncol)
      end do
      close(31)

      open(32,file='~/modflow/2d/landdat/siltthick.dat',
     &   status='unknown')
      do j=1,nrow
       read(32,'(30f7.1)')(dist(i,j), i=1,ncol)
      end do
      close(32)


      do 30 iper=1,nper
30     perlen(iper)=30.4
      do 40 i=1,10
40     kcond(i)=0.1


      x00=xm
      y00=ym
      istagefile=0
      call  river (ncol,nrow,nlay,ncolm,nrowm,nlaym,nperm,
     *    x,y,z,zz,perlen,kcond,cond,itype,stage,noder,ir,jr,
     *    x00,y00,istagefile,nper,iunit,type,siltz,dist)
      stop
      end
cccccccccccccRiver package
c
c    Documentation
c
c    ncol,nrow,nlay = number of computational cells in
column, row, and
c    vertical directions
c    ncolm,nrowm,nlaym = max limits of ncol,nrow,nlay
c    nperm = max number of stress periods
c    (x(i),i=1,ncol+10, computational discretization in x
c    (y(i),i=1,nrow+10, computational discretization in y
c    (z(i),i=1,nlay+10, computational discretization in z
c    zz(i,j), minimum river bottom elevation in cell (i,j)
c    k=1 refuse layer
c    k=2 surface water bodies
c    k=3 silt
c    k=4 sand
c    k=5 grvel
c    k=6 trout
c    k=7 undiff
c    k=8 crb
c    (xleak(itype),itype=1,5), leakance of different
rivers/lakes
c    itype=1, Smith/Bybee Lake
c    itype=2, Ramsey Lake
c    itype=3, Columbia River
c    itype=4, Willamette
c    itype=5, Multnomah Channel
c    itype=6, North Portland Harbor
c    itype=7, Columbia Slough
c    itype=8, north Slough
c    itype=9, Blind Slough
c    itype=10,  small pond near blind slough at SE corner
of landfill
c    ir(ico),ico=1,maxnumber river cell, incidence list of
river cell
c    = i index of the (ico)th river cell
c    jr(ico),ico=1,maxnumber river cell, incidence list of
river cell
c    = j index of the (ico)th river cell
c    itype(ico) =  type of river for the (ico)th river cell
c    noder(ico) = number of  15x15 river
c    in the (ico)th computational river cell
c    stage(ico) = riverstage at the (ico)th river cell
c    cond(ico) = river conductance of the (ico)th river cell
c    cond = xleak* (active river area in one computational
cell)
c    x00,y00 = coordinates of lower left corner of model
area in state system
c    istagefile = 1, river stage input from a data file
c    istagefile = 0, river stages generated in this routine
c    nper = number of stress periods
c
      subroutine river
(ncol,nrow,nlay,ncolm,nrowm,nlaym,nperm,
     *    x,y,z,zz,perlen,kcond,cond,itype,stage,noder,ir,jr,
     *    x00,y00,istagefile,nper,iunit,type,siltz,dist)
      dimension x(ncolm+1),y(nrowm+1),z(nlaym+1),
     *    zz(nrowm,ncolm),iunit(24)
      dimension perlen(nperm)
      dimension itype(ncolm*nrowm)
      dimension stage(ncolm*nrowm)
      dimension noder(ncolm*nrowm)
      dimension ir(ncolm*nrowm),jr(ncolm*nrowm)
      dimension cond(ncolm*nrowm)
      dimension siltz(ncolm,nrowm),dist(ncolm,nrowm)
      integer type(ncolm,nrowm,nlaym)
      real kcond(10)
      dxs=15.
      dys=15.
      open
(11,file="~/modflow/largegrid/landdat/rivernd2.cjb",
     -    status="old")
      open
(12,file="~/modflow/largegrid/landdat/stage.dat",
     &    status="unknown")
      open (13,file="riv.npt",status="unknown")

      mxrivr=ncolm*nrowm
      irivcb=0
      write (13,"(2I10)")mxrivr,irivcb

      do 510 i=1,ncol*nrow
       noder(i)=0
510   continue
      nriv=0
167   read(11,'(2f8.0,i2,f5.0)',end=168) xss,yss,nr,bot
      call xtoi (xss-x00,i,ncol,x,ncolm+1)
      call xtoi (y00-yss,j,nrow,y,nrowm+1)
      if(i.eq.0.or.j.eq.0)go to 167
      zz(i,j)=min(zz(i,j),bot)
      nnn=(i-1)*nrow+j
      noder(nnn) = noder(nnn)+1
```

274

```fortran
      if(noder(nnn).eq.1) goto 169
      do 310 ii=1,nriv
        if(nnn.eq.(ir(ii)-1)*nrow+jr(ii)) goto 167
310   continue
169   nriv=nriv+1
      ir(nriv)=i
      jr(nriv)=j
      itype(nriv)=nr
      goto 167

c     skip the first two record of x and y corrdinates

168   time=0.0

      do 540 iper=1,nper


c     for first 38 time period iper=1-8, getting rid of small
pond (itype=10)

      if(iper.le.38)then
        itmp=0
        do i=1,nriv
          if(itype(i).ne.10)itmp=itmp+1
        end do
      end if
      if(iper.ge.39)itmp=nriv

      write(13,"(I10)")itmp

      time=time+(perlen(iper)+perlen(iper+1))*0.5
      w=2.*3.14159/365.

      if(istagefile.eq.0)then
        do 191 i=1,nriv
c
c     Reset river and lake levels
c
          if(itype(i).ge.7.and.itype(i).le.9)then
            if(iper.le.31)rivlev=10.2
            if(iper.ge.32)rivlev=9.74
          endif
          if(itype(i).eq.1)then
            if(iper.le.31)rivlev=10.2
            if(iper.ge.32.and.iper.le.46)rivlev=11.74
            if(iper.ge.47)rivlev=9.74
          endif

          if(itype(i).eq.10)stage(i)=15.0
          if(itype(i).ne.10)stage(i)=rivlev
          if(itype(i).eq.1)stage(i)=rivlev
191       continue

        if(iper.eq.1)then
          write (12,*) nriv
          write (12,"(5x,20I7)") (ico,ico=1,nriv)
          write (12,"(5x,20I7)") (ir(ico),ico=1,nriv)
          write (12,"(5x,20I7)") (jr(ico),ico=1,nriv)
        endif
        write(12,"(I5,20(1x,f6.1))") iper,(stage(i),i=1,nriv)
      else
        read(12,*)
        read(12,*)
        read(12,*) (ir(i),i=1,nriv)
        read(12,*) (jr(i),i=1,nriv)
        read(12,*) (stage(i),i=1,nriv)
      endif

      do 530 ico=1,nriv

c     ii - col #

      ii=ir(ico)

c     jj - row #

      jj=jr(ico)

c     scanning gtype2d.out file to ensure that rivercell is
c     top most active one

      do l=1,nlay
        if(type(ii,jj,l).ge.3.and.type(ii,jj,l).le.7)then
          layer=l
          go to 137
        end if
      end do
137     continue

c     check to make sure that river cell is near water

      wmp=stage(ico)
423     if(z(layer+1).gt.wmp)then
        layer=layer+1
        go to 423
      end if

      nnn=(ii-1)*nrow+jj
      thick=1.0

      if(type(ii,jj,layer).ge.3)then
        thick=siltz(ii,jj)-(z(layer)-z(layer+1))/2.0
      end if
      if(thick.lt.1.0)thick=1.0
      cond(ico) =
(kcond(itype(ico))/thick)*noder(nnn)*dxs*dys

c     skipping over landfill pond for 38 stress periods

      if(itype(ico).eq.10.and.iper.le.38)go to 530
      if(itype(ico).eq.10)zz(ii,jj)=10.0
      if(itmp.ge.1)
write(13,"(3I10,F10.2,g10.2,f10.0,i10)")
*       layer,jj,ii,stage(ico),cond(ico),zz(ii,jj),itype(ico)
530   continue
540 continue
      return
      end

****************************************************
************************

      subroutine xtoi (x0,i0,nx,x,nm)
      dimension x(nm)
      i0=0
      do 100 i=1,nx+1
        if(x(i).ge.x0) then
          i0=i-1
          return
        endif
100   continue
      return
      end
```

R1ADD2D.FOR produces a file that tells the model which cells are river cells and what the conductance of each river cell is. It reads the geology files created by GRID2D.FOR as well as SILTTHICK.DAT and SILTTOP.DAT created by SILT_THICK.FOR and SILT_TOP.FOR respectively, and calculates a conductance for each river cell based on the thickness of the underlying silt and the effective conductivity of the silt and river sediments.

Possible changes that could be made for sensitivity analysis would be to the water levels in the sloughs and Smith and Bybee Lakes. The river levels are designated in the RIVER subroutine by the variable RIVLEV. RIVLEV has three broad time periods for its designation; prior to 1982, from 1982 to 1996, and post 1996. For the sloughs (types 7, 8, and 9 as designated by GTYPE2D.FOR) the levels prior to 1982 and from 1982 to 1996 are the same. The post 1996 level is based on the average of the 1982 to 1994 data (the last year of data at the time of the modeling project is 1994) and reflects a slightly drier period then prior to 1982. The water level in Smith Lake is held 2 feet higher from 1982 to 1996 due to the regulating dam placed between the North Slough and Bybee Lake (which is connected to Smith Lake). If sensitivity to river levels is to be simulated, RIVLEV must be changed appropriately.

The effective conductivity is designated on line 155 in the main part of the program by an array called KCOND. KCOND is in feet per day and represents the effective, vertical hydraulic conductivity between the silt layer and the river cell.

Once the grid is converted to a true two-dimensional grid in RIV2D.FOR, this parameter becomes ineffective since all cells outside of the landfill are deemed inactive, it does become important in the three-dimensional flow model discussed in Chapter 7.

## RIV2D.FOR

```fortran
c    riv2d.for - Program to convert landfill cells next to
dikes and
c    along bottom to river cells for use in the 2d model.
c    Portland State University - June,1995 - Thomas Lowry
c
     parameter(ncol=153,nrow=117,nnlay=1,nlay=39,
     &   nper=48,datum=3.195,hconv=2834.6278,
     &   dx=50,dy=50,bdike=40,hydd=1e-7*hconv,
     &   numcell=ncol*nrow,dike=25)
     integer lftype,type,numriv,type,rtype,ncell
     real rivlev,cnd,dist,grav,lfh,lfbot,lftop,hyd,hhyd
     dimension dist(ncol,nrow),type(ncol,nrow,nlay),
     &   lftype(ncol,nrow),cnd(ncol,nrow),grav(ncol,nrow),
     &   silttop(ncol,nrow),bot(nlay),hnew(ncol,nrow),
     &   thick(ncol,nrow),ntype(ncol,nrow),lfh(nper+1),
     &   cond(10),lfbot(ncol,nrow),lftop(ncol,nrow),
     &
rcond(5000),irtype(5000),rtype(ncol,nrow),stage(5000),
     &   condv(10),condh(10),zz(ncol,nrow)
     open(1,file='../../../0d/data/sjlin.npt',status='old')
     open(2,file='../../data/gtype2d.out',status='old')
     open(3,file='../../data/siltthick.dat',status='old')
     open(4,file='../../data/rivspa.npt',status='unknown')

open(5,file='../../../../commondata/rivlev.dat',status='old')
     open(6,file='../../data/bcpre82.npt',status='old')
     open(7,file='../../data/silttop.dat',status='old')
     open(8,file='../../data/thick.out',status='unknown')
     open(9,file='../../data/lfbot.out',status='unknown')
     open(10,file='../../data/2dgtype.out',status='unknown')
     open(11,file='../../data/lf.npt',status='old')
     open(12,file='../../data/lftop.out',status='unknown')
     open(14,file='../../data/riv.npt',status='old')
     open(15,file='../../data/bc82_97.npt',status='old')
     open(16,file='../../data/bcpost97.npt',status='old')
c
c    Read in Horizontal Silt Conductivity (cm/sec), and
anisotropy
c
     read(1,*)hhyd
     hyd=hhyd*hconv
     hyd2=hyd/2
     read(1,*)trpy
c
c    Read in first line of riv.npt file
c
     read(14,"(2I10)")mxrivr,irivcb
c
c    Calculate anisotropy for use in calculation
c
     anis=(1/trpy)*2
c
c    Read in silt thicknesses for use in conductance
calculation
c
     do j=1,nrow
       read(3,'(30f7.1)')(dist(i,j), i=1,ncol)
     end do
     close(3)
c
c    Read in silt top elevation to determine silt bottom
elevation
c
     do j=1,nrow
       read(7,'(30f7.1)')(silttop(i,j), i=1,ncol)
     end do
```

```fortran
c
c    Calculate bottom of layer elevation
c
     do k=1,nlay
       if(k.ge.1.and.k.le.25)bot(k)=95-k*5.0
       if(k.ge.26.and.k.le.27)bot(k)=(95-25*5.0)-(k-
25)*10.0
       if(k.gt.27)bot(k)=(95-25*5.0)-(2*10.0)-(k-27)*25
     end do
c
c    reading file with grid type #'s
c
 23  read(2,122,end=33)i,j,k,itemp
     type(i,j,k)=itemp
     if(type(i,j,k).ge.20.and.type(i,j,k).le.40)then
       type(i,j,k)=type(i,j,k)+100
     endif
 122 format(4(i4))
     go to 23
 33  continue
     close(2)
c
c    Make new gtype.out array for 2d model
c
     do i=1,ncol
       do j=1,nrow
         do k=1,nlay
           if(type(i,j,k).gt.100)then
             ntype(i,j)=type(i,j,k)-100
             goto 266
           endif
           ntype(i,j)=0
         end do
 266     continue
       end do
     end do
c
c    Flatten out bottom of landfill
c
     do i=1,ncol
       do j=1,nrow
         lfbot(i,j)=silttop(i,j)
         if(ntype(i,j).gt.0)lfbot(i,j)=-5.0
       end do
     end do
c
c    Correct for resolution problem for 3 cells at end of
engineered dike
c    and write 2dgtype.out file
c
     ntype(90,48)=0
     ntype(91,48)=0
     ntype(89,49)=0
c
c    Write out new geology file and add unique value to
landfill type numbers
c
     do i=1,ncol
       write(10,'(3i4)')(i,j,ntype(i,j),j=1,nrow)
       do j=1,nrow
         if(ntype(i,j).gt.20)ntype(i,j)=ntype(i,j)+100
       end do
     end do

c
c    Loop through cells to find top of landfill to define layer
thickness
```

278

```fortran
c     at that point
c
      do i=1,ncol
        do j=1,nrow
          do k=1,nlay
            thick(i,j)=0.0
            if(type(i,j,k).ge.20)then
              thick(i,j)=(bot(k)+5)-silttop(i,j)
              goto 265
            endif
            if(k.eq.nlay)then
              thick(i,j)=0.0
            endif
          end do
265       continue
          lftop(i,j)=silttop(i,j)+thick(i,j)
        end do
      end do
c
c     Output bottom of landfill, and cell thickness to
seperate files
c
      do i=1,ncol
        write(8,'(39f10.4)')(thick(i,j),j=1,nrow)
        write(9,'(39f10.4)')(lfbot(i,j),j=1,nrow)
        write(12,'(39f10.4)')(lftop(i,j),j=1,nrow)
      end do
c
c     Begin loop for each stress period
c
      do n=1,nper
c
c     Read in gravel heads
c
        if(n.eq.1)then
          do j=1,nrow
            read(6,'(30f9.3)')(grav(i,j),i=1,ncol)
          end do
          close(6)
        endif
        if(n.eq.32)then
          do j=1,nrow
            read(15,'(30f9.3)')(grav(i,j),i=1,ncol)
          end do
        endif
        if(n.eq.47)then
          do j=1,nrow
            read(16,'(30f9.3)')(grav(i,j),i=1,ncol)
          end do
          close(6)
        endif

c
c     Read in river types for each cell and convert to 2d
c     irtype=1, Smith/Bybee Lake
c     irtype=2, Ramsey Lake
c     irtype=3, Columbia River
c     irtype=4, Willamette
c     irtype=5, Multnomah Channel
c     irtype=6, North Portland Harbor
c     irtype=7, Columbia Slough
c     irtype=8, north Slough
c     irtype=9, Blind Slough
c     irtype=10, Small pond near blind slough at SE corner
of landfill
c
          read(14,"(I10)")itmp
```

```fortran
      do i=1,ncol
        do j=1,nlay
          rtype(i,j)=0
        end do
      end do
      do ir=1,itmp
        read(14,"(3I10,F10.2,g10.2,f10.0,i10)")
     &      k,j,i,stage(ir),rcond(ir),zz(i,j),irtype(ir)
        if(irtype(ir).ge.7.and.irtype(ir).le.9)then
          rtype(i,j)=irtype(ir)
        endif
      end do
117   format(4i4)
c
c     Read in Landfill head from water balance model for
use in calculating
c     conductance for dike cells
c
      read(11,'(f10.4)')lfh(n)
c
c     Reset type array to reflect growth of landfill
c
      do i=1,ncol
        do j=1,nrow
          if(n.eq.1)then
            if(ntype(i,j).eq.126)
     &            ntype(i,j)=ntype(i,j)-100
          endif
          if(n.eq.3)then
            if(ntype(i,j).eq.127)
     &            ntype(i,j)=ntype(i,j)-100
            if(ntype(i,j).eq.122)
     &            ntype(i,j)=ntype(i,j)-100
            if(ntype(i,j).eq.128)
     &            ntype(i,j)=ntype(i,j)-100
          endif
          if(n.eq.4)then
            if(ntype(i,j).eq.121)
     &            ntype(i,j)=ntype(i,j)-100
          endif
          if(n.eq.6)then
            if(ntype(i,j).eq.123)
     &            ntype(i,j)=ntype(i,j)-100
            if(ntype(i,j).eq.129)
     &            ntype(i,j)=ntype(i,j)-100
            if(ntype(i,j).eq.130)
     &            ntype(i,j)=ntype(i,j)-100
          endif
          if(n.eq.39)then
            if(ntype(i,j).eq.124)
     &            ntype(i,j)=ntype(i,j)-100
            if(ntype(i,j).eq.140)
     &            ntype(i,j)=ntype(i,j)-100
          endif
          if(n.eq.40)then
            if(ntype(i,j).eq.125)
     &            ntype(i,j)=ntype(i,j)-100
          endif
        end do
      end do


c
c     Assign river levels
c
      open(17,file='./.././0d/data/sjlin.npt',status='old')
      do i=1,13
```

```
        read(17,*)
      end do
      read(17,*)watlev1
      read(17,*)watlev2
      read(17,*)watlev3
      read(17,*)watlev4
      read(17,*)watlev5
      read(17,*)watlev6
      close(17)

      if(n.le.32)rivlev=watlev1
      if(n.ge.33.and.n.le.41)rivlev=watlev2
      if(n.ge.42.and.n.le.43)rivlev=watlev3
      if(n.ge.44.and.n.le.45)rivlev=watlev4
      if(n.ge.46.and.n.le.47)rivlev=watlev5
      if(n.ge.47)rivlev=watlev6
c
c     Add engineered dike to top of silttop
c
      do i=1,ncol
        do j=1,nrow
          do k=1,nlay
            if(type(i,j,k).eq.40)then
              silttop(i,j)=silttop(i,j)+bot(k)+5.0
              goto 444
            endif
          end do
444       continue
        end do
      end do
c
c     Initialize lftype array
c
      do i=1,ncol
        do j=1,nrow
          do k=1,nlay
            lftype(i,j)=0
          end do
        end do
      end do
c
c     Check to the sides of all cells to see if they are next to
dike
c
      numriv=0
      do i=1,ncol
        do j=1,nrow
          do ii=1,10
            cond(ii)=0.0
          end do
          cnd(i,j)=0.0
          nlf=0
          ncell=0
c
c     If it is a landfill cell then:
c
          if(ntype(i,j).gt.20.and.ntype(i,j).lt.100.and.
     &       ntype(i,j).ne.40)then
c
c     Check cell to the east of the landfill cell
c
            if(ntype(i+1,j).lt.20.or.ntype(i+1,j).eq.40.or.
     &         ntype(i+1,j).gt.100)then
c
c     Find which river cell the landfill cell is close to, no
more than
c     6 cells away.
```

```
c
            do ii=i+1,ncol
              if(ncell.ge.7)goto 540
              if(rtype(ii,j).gt.0)then
                lftype(i,j)=rtype(ii,j)
                goto 540
              end if
              ncell=ncell+1
            end do
540         continue
            ncell=0
c
c     Find if cell is bordering old landfill lake
c
            if(rtype(i+1,j).eq.0.and.
     &         lftype(i,j).eq.0)lftype(i,j)=6
c
c     If cell is not Blind, North or Columbia Slough, skip
calculations
c
            if(lftype(i,j).lt.6.or.lftype(i,j).gt.9)goto 545
            nlf=nlf+1
            if(lfh(n).ge.rivlev.and.lfh(n).lt.dike)then
              ave=(lfh(n)+rivlev)/2
              dz=ave
            endif
            if(lfh(n).ge.dike)then
              ave=(dike+rivlev)/2
              dz=ave
            endif
            if(lfh(n).lt.rivlev)then
              ave=(dike+rivlev)/2
              dz=ave
            endif
            condv(lftype(i,j))=hyd2*anis*dx*dy/dist(i,j)
            condh(lftype(i,j))=hyd*dz*dy/bdike
            if(ntype(i+1,j).eq.40)then
              condh(lftype(i,j))=hydd*dz*dy/bdike
            endif
            hnew(i,j)=((condv(lftype(i,j))*grav(i,j)+
     &        condh(lftype(i,j))*rivlev)/
     &        (condv(lftype(i,j))+condh(lftype(i,j))))
            cond(lftype(i,j))=condv(lftype(i,j))+
     &        condh(lftype(i,j))
          endif
545       continue
c
c     Check cell to the west of the landfill cell
c
          if(ntype(i-1,j).lt.20.or.ntype(i-1,j).eq.40.or.
     &       ntype(i-1,j).gt.100)then
c
c     Find which river cell the landfill cell is close to
c
            do ii=i-1,1,-1
              if(ncell.ge.7)goto 550
              if(rtype(ii,j).gt.0)then
                lftype(i,j)=rtype(ii,j)
                goto 550
              end if
              ncell=ncell+1
            end do
550         continue
            ncell=0
c
c     Find if cell is bordering old landfill lake
c
```

280

```
              if(rtype(i-1,j).eq.0.and.
     &             lftype(i,j).eq.0)lftype(i,j)=6
c
c    If cell is not Blind, North or Columbia Slough, skip
calculations
c
              if(lftype(i,j).lt.6.or.lftype(i,j).gt.9)goto 555
              nlf=nlf+1
              if(lfh(n).ge.rivlev.and.lfh(n).lt.dike)then
                ave=(lfh(n)+rivlev)/2
                dz=ave
              endif
              if(lfh(n).ge.dike)then
                ave=(dike+rivlev)/2
                dz=ave
              endif
              if(lfh(n).lt.rivlev)then
                ave=(dike+rivlev)/2
                dz=ave
              endif
              condv(lftype(i,j))=hyd2*anis*dx*dy/dist(i,j)
              condh(lftype(i,j))=hyd*dz*dy/bdike
              if(ntype(i-1,j).eq.40)then
                condh(lftype(i,j))=hydd*dz*dy/bdike
              endif
              hnew(i,j)=((condv(lftype(i,j))*grav(i,j)+
     &             condh(lftype(i,j)))*rivlev)/
     &             (condv(lftype(i,j))+condh(lftype(i,j))))
              cond(lftype(i,j))=condv(lftype(i,j))+
     &             condh(lftype(i,j))
            endif
555         continue
c
c    Check cell to the south of the landfill cell
c
              if(ntype(i,j+1).lt.20.or.ntype(i,j+1).eq.40.or.
     &             ntype(i,j+1).gt.100)then
c
c    Find which river cell the landfill cell is close to
c
              do jj=j+1,nrow
                if(ncell.ge.7)goto 560
                if(rtype(i,jj).gt.0)then
                   lftype(i,j)=rtype(i,jj)
                   goto 560
                end if
                ncell=ncell+1
              end do
560             continue
              ncell=0
c
c    Find if cell is bordering old landfill lake
c
              if(rtype(i,j+1).eq.0.and.
     &             lftype(i,j).eq.0)lftype(i,j)=6
c
c    If cell is not Blind, North or Columbia Slough, skip
calculations
c
              if(lftype(i,j).lt.6.or.lftype(i,j).gt.9)goto 565
              nlf=nlf+1
              if(lfh(n).ge.rivlev.and.lfh(n).lt.dike)then
                ave=(lfh(n)+rivlev)/2
                dz=ave
              endif
              if(lfh(n).ge.dike)then
                ave=(dike+rivlev)/2
                dz=ave
              endif
              if(lfh(n).lt.rivlev)then
                ave=(dike+rivlev)/2
```

```
                dz=ave
              endif
              if(lfh(n).lt.rivlev)then
                ave=(dike+rivlev)/2
                dz=ave
              endif
              condv(lftype(i,j))=hyd2*anis*dx*dy/dist(i,j)
              condh(lftype(i,j))=hyd*dz*dy/bdike
              if(ntype(i,j+1).eq.40)then
                condh(lftype(i,j))=hydd*dz*dy/bdike
              endif
              hnew(i,j)=((condv(lftype(i,j))*grav(i,j)+
     &             condh(lftype(i,j))*rivlev)/
     &             (condv(lftype(i,j))+condh(lftype(i,j))))
              cond(lftype(i,j))=condv(lftype(i,j))+
     &             condh(lftype(i,j))
            endif
565         continue
c
c    Check cell to the north of the landfill cell to see if it is
natural dike
c
              if(ntype(i,j-1).lt.20.or.ntype(i,j-1).eq.40.or.
     &             ntype(i,j-1).gt.100)then
c
c    Find which river cell the landfill cell is close to
c
              do jj=j-1,1,-1
                if(ncell.ge.7)goto 570
                if(rtype(i,jj).gt.0)then
                   lftype(i,j)=rtype(i,jj)
                   goto 570
                end if
                ncell=ncell+1
              end do
570             continue
              ncell=0
c
c    Find if cell is bordering old landfill lake
c
              if(rtype(i,j-1).eq.0.and.
     &             lftype(i,j).eq.0)lftype(i,j)=6
c
c    If cell is not Blind, North or Columbia Slough, skip
calculations
c
              if(lftype(i,j).lt.6.or.lftype(i,j).gt.9)goto 575
              nlf=nlf+1
              if(lfh(n).ge.rivlev.and.lfh(n).lt.dike)then
                ave=(lfh(n)+rivlev)/2
                dz=ave
              endif
              if(lfh(n).ge.dike)then
                ave=(dike+rivlev)/2
                dz=ave
              endif
              if(lfh(n).lt.rivlev)then
                ave=(dike+rivlev)/2
                dz=ave
              endif
              condv(lftype(i,j))=hyd2*anis*dx*dy/dist(i,j)
              condh(lftype(i,j))=hyd*dz*dy/bdike
              if(ntype(i,j-1).eq.40)then
                condh(lftype(i,j))=hydd*dz*dy/bdike
              endif
              hnew(i,j)=((condv(lftype(i,j))*grav(i,j)+
     &             condh(lftype(i,j))*rivlev)/
```

281

```
     &                  (condv(lftype(i,j))+condh(lftype(i,j))))
               cond(lftype(i,j))=condv(lftype(i,j))+
     &              condh(lftype(i,j))
            endif
 575        continue
c
c     Sum up conductance from all sides to get effective
conductance
c
            if(nlf.gt.0)then
              do ii=6,9
                 cnd(i,j)=cnd(i,j)+cond(ii)
              end do
            endif
c
c     If cell is surrounded by landfill calculate downward
conductance
c
            if(nlf.eq.0)then
              lftype(i,j)=99
              cnd(i,j)=hyd2*anis*dx*dy/dist(i,j)
              hnew(i,j)=grav(i,j)
            endif
          endif
        end do
      end do
c
c     Find number of river cells for each stress period
c
     do i=1,ncol
       do j=1,nrow
         if(lftype(i,j).ge.6.and.lftype(i,j).le.9.or.
     &        lftype(i,j).eq.99)numriv=numriv+1
       end do
     end do


c
c     Write out to new rivspa.npt file in current directory
c
     if(n.eq.1)write(4,"(2I10)")mxrivr,irivcb
     write(4,"(i10)")numriv
     k=1
     do i=1,ncol
       do j=1,nrow
         if(lftype(i,j).lt.6.or.lftype(i,j).gt.9.and.
     &        lftype(i,j).ne.99)goto 585
         write(4,"(3I10,F10.2,g10.2,f10.0,2i10)")k,j,i,
     &        hnew(i,j),cnd(i,j),lfbot(i,j),
     &        lftype(i,j),n
 585        continue
       end do
     end do
     end do
     stop
     end
```

RIV2D.FOR is the program which converts the three-dimensional grid to the two-dimensional grid for use in the 2D mounding model. It uses output from the water balance model and the regional flow model to establish boundary conditions and calculate the conductance for all the landfill cells. Horizontal conductance to the sloughs through the dike and vertical conductance downward through the silt to the gravels is summed up to provide an effective conductance for each landfill cell. Horizontal conductance only pertains to those cells that are adjacent to the sloughs or Smith Lake.

Parameters, which may be changed for sensitivity analysis, are water levels in the sloughs, heads in the gravels, and conductivity and anisotropy in the silt. The water levels in the sloughs and lakes are defined just like in R1ADD2D.FOR and are designated by the same variable name, RIVLEV. Heads in the gravels are the results of simulations from the regional flow model and can only be changed by re-running the regional flow model. The horizontal silt conductivity is designated at the start of the program by a PARAMETER statement for the variable HYD. The anisotropy is designated in the same PARAMETER statement by the variable name TRPY. The vertical hydraulic conductivity is calculated internally by multiplying HYD times TRPY.

It is important when running the entire St. Johns Landfill modeling system that all parameters in all programs match up with each other in order to insure consistent results based on the same parameter values. RIV2D.FOR should be run anytime there

are changes to the hydraulic conductivity in the silt, the anisotropy in the silt, or the water levels in the sloughs and lakes.

## MF2DPRE.FOR

```
c    mf2dpre.for    modflow preprocessor
c    St. Johns Landfill modeling study
c    Portland State University - Chris Berger - 1994
c    As modified June, 1995 & February, 1996 by Thomas
Lowry
c
c    reads input file "gtype2d.out" defining type of layer
each
c    computational node is associated with and creates
input files
c    for modflow model

     integer hgu
     parameter(ix=160,jy=120,lay=40,hgu=8)
     integer type(ix,jy)
     dimension xd(ix),yd(jy),zd(lay),iunit(24)
     dimension xint(ix),yint(jy),zint(lay)
     dimension xs(ix),ys(jy),zs(lay)
     dimension nstp(80)

     common /info/xd,yd,zd,xm,ym,il,ic,ir,type
     common /time/nper,nstp
     common /files/iunit


     open(98,file='.././data/2dgtype.out',status='old')

c    Assinging # of boundaries of constant cell length along
x-axis (ic),
c    along y-axis (ir), , along z-axis (il)

     ici=2
     iri=2
     ili=4

c    (xm,ym) - Assigning location of origin in terms of
model coordinate
c    system

     xm=8100
     ym=15300

c    Assigning distances of intervals of constant cell width
along x-axis from
c    point defined by xm

     xs(1)=0
     xs(2)=7650
     xint(1)=50

c    Assigning distances of intervals of constant cell width
along y-axis from
c    point defined by ym

     ys(1)=0
     ys(2)=5850
     yint(1)=50

c    Assigning elevations of intevals of constant cell width
along z-axis

     zs(1)=95
     zs(2)=-30
     zs(3)=-50
     zs(4)=-350
     zint(1)=5
```

```
     zint(2)=10
     zint(3)=25

c    counting number of cell boundaries

     ic=1
     ir=1
     il=1
     do i=2,ici
       ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
     end do
     do j=2,iri
       ir=ir+nint((ys(j)-ys(j-1))/yint(j-1))
     end do
     do k=2,ili
       il=il+nint((zs(k-1)-zs(k))/zint(k-1))
     end do

c    calculating location of cell boundary along axis

     xd(1)=xs(1)
     yd(1)=ys(1)
     zd(1)=zs(1)
     icnt=1
     jcnt=1
     kcnt=1
     do i=2,ic
       xd(i)=xd(i-1)+xint(icnt)
       if(nint(xd(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
     end do
     do j=2,ir
       yd(j)=yd(j-1)+yint(jcnt)
       if(nint(yd(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
     end do
     do k=2,il
       zd(k)=zd(k-1)-zint(kcnt)
       if(nint(zd(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
     end do

     open(91,file='.././data/check.dat',status='unknown')
     write(91,*)ic,ir,il
     write(91,*)
     do i=1,ic
       write(91,*)xd(i)
     end do
     write(91,*)
     do i=1,ir
       write(91,*)yd(i)
     end do
     write(91,*)
     do i=1,il
       write(91,*)zd(i)
     end do
     close(91)

c    reading file with grid type #'s

 23  read(98,*,end=33)i,j,itemp
     type(i,j)=itemp
     go to 23
 33  continue

     call addcell

c    calling subroutine which creates basic package input

     call basic
```

285

```
c    calling subroutine which creates body centered flow          c    iunit(11) - SSOR Package
     package input                                                c    iunit(12) - Output control option  OC
                                                                  c    iunit(13) - Pre-conditioned conjugate gradient
     call bcf2                                                          package PCG2
                                                                  c    iunit(14) - Stream Package report STR1
c    calling subroutine which creates output control file         c    iunit(22) - mt3d linking module LKMT18
                                                                  c    initializing all values in IUNIT array to zero
     call output
                                                                       do i=1,24
c    calling subroutine which creates pcg2 control file                iunit(i)=0
                                                                       end do
     call pcg2
                                                                  c    assigning packages which are used nonzero unit
c    calling subrouting which creates recharge input file              numbers

     call recharge                                                     iunit(1)=11
                                                                       iunit(3)=13
c    calling subrouting which creates drain input file                 iunit(4)=14
                                                                       iunit(8)=18
     call drain                                                        iunit(12)=22
                                                                       iunit(13)=23

     stop                                                         c    IAPART - indicates whether array BUFF is separate
     end                                                               from array RHS
                                                                  c    IAPART =0 same space,       IAPART=1 separate
************************************************
***********************                                                iapart=0
c    creates basic package input
                                                                  c    istrt -indicates whether starting heads are saved (.eq.0
     subroutine basic                                                  no, .ne.0 yes)
     parameter(ix=160,jy=120,lay=40,hgu=8)
     integer type,ibound(ix,jy,lay),acell                              istrt=1
     dimension xd(ix),yd(jy),zd(lay),shead(ix,jy,lay)
     dimension perlen(80),grav(ix,jy),buff(ix,jy)                c    locati - location of data to be read into integer arrays
     character*80 headng1,headng2                                c    (unit number=1) for basic package
     character*20 fmtini,fmtinsh
                                                                       locati=1
     common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy)                  iconsti=1
     common /time/nper,nstp(80)
     common /files/iunit(24)                                     c    fmtini - format for integer input arrays
     common /landf/ acell(ix,jy)
                                                                       fmtini='(30i3)'
c    defining variables
c    headng1, headng2 - first two lines of basic input file      c    iprni - flag determining how input of integer arrays is
     (file headings)                                                  written
     headng1='Modflow Basic Package Input File'                  c    to output files
     headng2='St. Johns Landfill Project - PSU 1994'
     nlay=1                                                            iprni=-1
     nrow=ir-1
     ncol=ic-1                                                   c    defining boundary array IBOUND(i,j,k) using type #
c    NPER - number of stress periods                             c    IBOUND<0  constant head cell
     nper=48                                                     c    IBOUND=0  inactive cell
c    itmuni - time unit code, days=4                             C    IBOUND>0 variable head cell
     itmuni=4                                                    c    if type#=0, ibound=0 (undefined)
c    iunit - array which contains unit numbers for package       c    if type#=1, ibound=0 (water)
     inputs                                                      c    if type#=2, ibound>1  (landfill)
c    iunit(1) - block-centered flow package BCF2                 c    if type#=3, ibound>1  (silt layer)
c    iunit(2) - well package WEL                                 c    if type#=4, ibound>1 (sand layer)
c    iunit(3) - drain package DRN                                c    if type#=5, ibound>1 (gravel layer)
c    iunit(4) - river package RIV                                c    if type#=6, ibound=0 (troutdale aquifer)
c    iunit(5) - evapotranspiration package EVT                   c    if type#=7, ibound=0 (unconsolidated)
c    iunit(6) -reserved for transient leakage package            c    if type#=8, ibound=0 (columbia river basalt)
c    iunit(7) - general head boundary package GHB                c    if type#>10, ibound>1 (some subdivision of landfill)
c    iunit(8) - recharge package RCH
c    iunit(9) - SIP package                                           do j=1,nrow
c    iunit(10) - reserved for additional solver                       do i=1,ncol
                                                                           ibound(i,j,1)=0
```

```
      if(type(i,j).gt.0)ibound(i,j,1)=5
      if(acell(i,j).ne.0)then
        if(acell(i,j).gt.0)ibound(i,j,1)=0
      end if
    end do
  end do

c  hnoflo - head value assigned to inactive cells

  hnoflo=0.0
c
c Read in initial heads for gravel
c
  open(6,file='.././data/bcpre82.npt',
  &  status='old')
    do j=1,nrow
    read(6,'(30f9.3)')(grav(i,j),i=1,ncol)
    end do
    do j=1,nrow
    read(6,'(30f9.3)')(buff(i,j),i=1,ncol)
    end do
    close(6)

c assigning initial heads

    do j=1,nrow
      do i=1,ncol
        shead(i,j,1)=grav(i,j)
      end do
    end do

c   locat - location of data to be read into initial head
arrays.
c   (unit number=0) means head is equal to const

    locatsh=1
    constsh=1.0
    fmtinsh='(10g11.4)'
    iprnsh=0

c   perlen - length of stress period
    perlen(1)=365.0
    perlen(2)=365.0
    perlen(3)=365.0
    perlen(4)=365.0
    perlen(5)=365.0
    perlen(6)=365.0
    perlen(7)=365.0
    perlen(8)=365.0
    perlen(9)=365.0
    perlen(10)=365.0
    perlen(11)=365.0
    perlen(12)=365.0
    perlen(13)=365.0
    perlen(14)=365.0
    perlen(15)=365.0
    perlen(16)=365.0
    perlen(17)=365.0
    perlen(18)=365.0
    perlen(19)=365.0
    perlen(20)=365.0
    perlen(21)=365.0
    perlen(22)=365.0
    perlen(23)=365.0
    perlen(24)=365.0
    perlen(25)=365.0
    perlen(26)=365.0

    perlen(27)=365.0
    perlen(28)=365.0
    perlen(29)=365.0
    perlen(30)=365.0
    perlen(31)=365.0
    perlen(32)=365.0
    perlen(33)=365.0
    perlen(34)=365.0
    perlen(35)=365.0
    perlen(36)=365.0
    perlen(37)=365.0
    perlen(38)=365.0
    perlen(39)=365.0
    perlen(40)=365.0
    perlen(41)=365.0
    perlen(42)=365.0
    perlen(43)=365.0
    perlen(44)=365.0
    perlen(45)=365.0
    perlen(46)=365.0
    perlen(47)=365.0
    perlen(48)=8760.0

c   nstp - number of time steps in stress period

    nstp(1)=1
    nstp(2)=1
    nstp(3)=1
    nstp(4)=1
    nstp(5)=1
    nstp(6)=1
    nstp(7)=1
    nstp(8)=1
    nstp(9)=1
    nstp(10)=1
    nstp(11)=1
    nstp(12)=1
    nstp(13)=1
    nstp(14)=1
    nstp(15)=1
    nstp(16)=1
    nstp(17)=1
    nstp(18)=1
    nstp(19)=1
    nstp(20)=1
    nstp(21)=1
    nstp(22)=1
    nstp(23)=1
    nstp(24)=1
    nstp(25)=1
    nstp(26)=1
    nstp(27)=1
    nstp(28)=1
    nstp(29)=1
    nstp(30)=1
    nstp(31)=1
    nstp(32)=1
    nstp(33)=1
    nstp(34)=1
    nstp(35)=1
    nstp(36)=1
    nstp(37)=1
    nstp(38)=1
    nstp(39)=1
    nstp(40)=1
    nstp(41)=1
    nstp(42)=1
```

```
      nstp(43)=1
      nstp(44)=1
      nstp(45)=1
      nstp(46)=1
      nstp(47)=1
      nstp(48)=24

c     tsmult - multiplier for the length of successive time
steps

      tsmult=1.0

      open(1,file='../../data/bas.npt',status='unknown')

c     writing to modflow input file for basic package

      write(1,'(a80)')headng1
      write(1,'(a80)')headng2
      write(1,'(5i10)')nlay,nrow,ncol,nper,itmuni
      write(1,'(24i3)')iunit
      write(1,'(2i10)')iapart,istrt

      do l=1,nlay
        write(1,'(i10,i10,a20,i10)')locati,iconsti,fmtini,iprni
        do j=1,nrow
          write(1,fmtini)(ibound(i,j,l),i=1,ncol)
        end do
      end do

      write(1,'(f10.0)')hnoflo

      do l=1,nlay

write(1,'(i10,f10.1,a20,i10)')locatsh,constsh,fmtinsh,iprnsh
        do j=1,nrow
          write(1,fmtinsh)(shead(i,j,l),i=1,ncol)
        end do
      end do

      do i=1,nper
        write(1,'(f10.1,i10,f10.1)') perlen(i),nstp(i),tsmult
      end do

      close(1)

      return
      end

c**************************************************
c*************************
c     creates bcf2 package input
      subroutine bcf2

c     hconv converts cm/sec to feet/day

parameter(ix=160,jy=120,lay=40,hgu=8,hconv=2834.627
8)
      integer type,acell
      real anis,hhyd,sy
      dimension xd(ix),yd(jy),zd(lay),laycon(lay),
    .   delr(ix),delc(jy),sf1(ix,jy,lay),bot(ix,jy),
    .   hy(ix,jy,lay),vcont(ix,jy,lay),sf2(ix,jy,lay),
    .   thresh(ix,jy,lay),tran(ix,jy,lay)
      dimension thick(ix,jy)
      character*20 fmtinr,fmtinc,fmtin1,fmtin2,fmtinh,
    .   fmtinv,fmtina,fmtinw,fmtinb
```

```
      common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy)
      common /files/iunit(24)
      common /landf/acell(ix,jy)

c     defining variables

      nlay=1
      nrow=ir-1
      ncol=ic-1
c     ISS - steady-state flag, if ISS=0 transient

      iss=0

c     ibcfcb - flag and unit number, if=0 cell by cell flow
terms
c     will not be printed or recorded

      ibcfcb=0

c     hdry - head assigned to cells which are converted to
dry

      hdry=888.0

c     iwdflg - flag which determines if wetting capacity
active(=1)

      iwdflg=0

c     wetfct - wetting factor

      wetfct=1.0

c     iwetit - iteration interval for attempting to wet cells

      iwetit=1

c     ihdwet - flag to determine which equation to use to
c     determine if cell is wet

      ihdwet=1
c     laycon - array that contains layer type, if layer is
above minus 5 ft,
c     laycon=3; below minus 5 ft, laycon=0

      laycon(1)=1

c     trpy - array defining anisotropic factor for each layer,
=1 for isotropic
c     control card for trpy: locats=0,constnts=1.0

      locats=0
      constnts=1.0

c     delr - vector defining cell widths along rows

      do i=1,ncol
        delr(i)=xd(i+1)-xd(i)
      end do
      locatr=iunit(1)
      cnstntr=1.0
      fmtinr='(15f7.1)'
      iprnr=0

c     delc - vector defining cell widths along colummns

      do j=1,nrow
```

```
      delc(j)=yd(j+1)-yd(j)
      end do
      locatc=iunit(1)
      cnstntc=1.0
      fmtinc='(15f7.1)'
      iprnc=0

c    sf1 - primary storage coefficient. If laycon=1, sf1 -
specfic yield
c    If laycon=3,0; sf1 - confined storage coefficient

      open(335,file='../../../0d/data/sjlin.npt',status='old')
      read(335,*)hhyd
      read(335,*)anis
      read(335,*)sy
      close(335)

      do j=1,nrow
        do i=1,ncol
          if(laycon(1).eq.1)then
            sf1(i,j,1)=0.1
            if(type(i,j).eq.1)sf1(i,j,1)=0.10
            if(type(i,j).eq.2)sf1(i,j,1)=0.15
            if(type(i,j).ge.10)sf1(i,j,1)=sy
            if(type(i,j).eq.3)sf1(i,j,1)=0.10
            if(type(i,j).eq.4)sf1(i,j,1)=0.20
            if(type(i,j).eq.5)sf1(i,j,1)=0.25
            if(type(i,j).eq.6)sf1(i,j,1)=0.15
            if(type(i,j).eq.7)sf1(i,j,1)=0.15
            if(type(i,j).eq.8)sf1(i,j,1)=0.15
          else if(laycon(1).eq.3.or.laycon(1).eq.0)then
            sf1(i,j,1)=1.0e-6
            if(type(i,j).eq.2)sf1(i,j,1)=1.0e-6
            if(type(i,j).ge.10)sf1(i,j,1)=1.0e-6
            if(type(i,j).eq.3)sf1(i,j,1)=1.0e-6
            if(type(i,j).eq.4)sf1(i,j,1)=1.0e-6
            if(type(i,j).eq.5)sf1(i,j,1)=1.0e-6
            if(type(i,j).eq.6)sf1(i,j,1)=1.0e-6
            if(type(i,j).eq.7)sf1(i,j,1)=1.0e-6
            if(type(i,j).eq.8)sf1(i,j,1)=1.0e-6
            if(type(i,j).eq.1)sf1(i,j,1)=1.0e-6
          end if
        end do
      end do
      locat1=iunit(1)
      cnstnt1=1.0
      fmtin1='(10g11.4)'
      iprn1=-1

c    tran - transmissivity
c    assigning conductivities to layer types

      open(333,file='../../data/thick.out',status='old')
      do i=1,ncol
        read(333,'(39f10.4)')(thick(i,j),j=1,nrow)
      end do
      close(333)


      do j=1,nrow
        do i=1,ncol
          if(type(i,j).eq.2)hydc=1.0e-3*hconv
          if(type(i,j).ge.10)hydc=5.0e-4*hconv
          if(type(i,j).eq.3)hydc=hhyd*hconv
          if(type(i,j).eq.4)hydc=((6.35e-2)/2)*hconv
          if(type(i,j).eq.5)hydc=6.35e-2*hconv
          if(type(i,j).eq.6)hydc=1.0e-4*hconv
          if(type(i,j).eq.7)hydc=1.0e-6*hconv
          if(type(i,j).eq.8)hydc=1.0e-6*hconv
          if(type(i,j).le.0)then
            hyd=1.0
          else
            hyd=hydc
          end if
          tran(i,j,1)=hyd*thick(i,j)
        end do
      end do
      locata=iunit(1)
      cnstnta=1.0
      fmtina='(10g11.4)'
      iprna=-1

c    hy - hydraulic conductivity

      do j=1,nrow
        do i=1,ncol
          if(type(i,j).le.0)then
            hy(i,j,1)=1.0
          else
            hy(i,j,1)=hydc
          end if
        end do
      end do
      locath=iunit(1)
      cnstnth=1.0
      fmtinh='(10g11.4)'
      iprnh=-1

c    defining layer bottoms

      open(334,file='../../data/lfbot.out',status='old')
      do i=1,ncol
        read(334,'(39f10.4)')(bot(i,j),j=1,nrow)
      end do
      close(334)
      locatb=iunit(1)
      cnstntb=1.0
      fmtinb='(10g11.4)'
      iprnb=-1
c
c Vanis is vertical anisotropy under landfill compared to
horizontal
c conductivity under landfill (.5 of hdike to vsilt)
c
      vanis=1/anis


c    sf2 -secondary storage coefficient, write only for
layers where
c    laycon is 2 or 3 - always specific yield

      do j=1,nrow
        do i=1,ncol
          sf2(i,j,1)=0.1
          if(type(i,j).eq.1)sf2(i,j,1)=0.10
          if(type(i,j).eq.2)sf2(i,j,1)=0.15
          if(type(i,j).ge.10)sf2(i,j,1)=sy
          if(type(i,j).eq.3)sf2(i,j,1)=0.10
          if(type(i,j).eq.4)sf2(i,j,1)=0.20
          if(type(i,j).eq.5)sf2(i,j,1)=0.25
          if(type(i,j).eq.6)sf2(i,j,1)=0.15
          if(type(i,j).eq.7)sf2(i,j,1)=0.15
          if(type(i,j).eq.8)sf2(i,j,1)=0.15
        end do
```

289

```
          end do
          locat2=iunit(1)
          cnstnt2=1.0
          fmtin2='(10g11.4)'
          iprn2=-1
          locatt=0

c     wetdry - combination of wetting threshold and a flag to
indicate
c     which neighboring cells can cause a cell to become
wet.
c     write if laycon is 1 or 3 and wetting capacity active
(iwdflg.ne.0)

          do j=1,nrow
            do i=1,ncol
              thresh(i,j,1)=-thick(i,j)/15.0

c     assigning thresholds to area of landfill which was
c     occupied by old lake

              if(acell(i,j).gt.0)then
                thresh(i,j,1)=0.0
              end if
            end do
          end do
          locatw=0
          cnstntw=1.0
          fmtinw='(10g11.4)'
          iprnw=0
          iprnw=-1

          open(1,file='../../data/bcf2.npt',status='unknown')

          write(1,104)iss,ibcfcb,hdry,iwdflg,wetfct,iwetit,ihdwet
104  format(2i10,f10.0,i10,f10.2,2i10)

          write(1,'(40i2)')(laycon(i), i=1,nlay)
          write(1,'(i10,f10.1)')locats,constnts
          write(1,'(i10,f10.1,a20,i10)')locatr,cnstntr,fmtinr,iprnr
          write(1,fmtinr)(delr(i),i=1,ncol)
          write(1,'(i10,f10.1,a20,i10)')locatc,cnstntc,fmtinc,iprnc
          write(1,fmtinc)(delc(j),j=1,nrow)

          do l=1,nlay

write(1,'(i10,f10.1,a20,i10)')locat1,cnstnt1,fmtin1,iprn1
          do j=1,nrow
            write(1,fmtin1)(sf1(i,j,l), i=1,ncol)
          end do

          if(laycon(l).eq.0.or.laycon(l).eq.2)then

write(1,'(i10,f10.1,a20,i10)')locata,cnstnta,fmtina,iprna
          do j=1,nrow
            write(1,fmtina)(tran(i,j,l), i=1,ncol)
          end do
          end if


          if(laycon(l).eq.1.or.laycon(l).eq.3)then

write(1,'(i10,f10.1,a20,i10)')locath,cnstnth,fmtinh,iprnh
          do j=1,nrow
            write(1,fmtinh)(hy(i,j,l), i=1,ncol)
          end do
          end if
```

```
          if(laycon(l).eq.1.or.laycon(l).eq.3)then

write(1,'(i10,f10.1,a20,i10)')locatb,cnstntb,fmtinb,iprnb
          do j=1,nrow
            write(1,fmtinb)(bot(i,j), i=1,ncol)
          end do
          end if


          if(laycon(l).eq.3.or.laycon(l).eq.2)then

write(1,'(i10,f10.1,a20,i10)')locat2,cnstnt2,fmtin2,iprn2
          do j=1,nrow
            write(1,fmtin2)(sf2(i,j,l), i=1,ncol)
          end do
          end if


if(laycon(l).eq.3.or.laycon(l).eq.1.and.iwdflg.ne.0)then

write(1,'(i10,f10.1,a20,i10)')locatw,cnstntw,fmtinw,iprnw
          do j=1,nrow
            write(1,fmtinw)(thresh(i,j,l),i=1,ncol)
          end do
          end if
          end do
          close(1)

          open(33,file='../../data/balance.npt',status='unknown')
          do l=1,nlay
            if(l.ne.nlay)then

write(33,'(i10,f10.1,a20,i10)')locatv,cnstntv,fmtinv,iprnv
          do j=1,nrow
            write(33,fmtinv)(vcont(i,j,l), i=1,ncol)
          end do
          end if
          end do
          close(33)

          return
          end

c*************************************************
c************************
c     creates output control input
          subroutine output
          parameter(ix=160,jy=120,lay=40,hgu=8)
          integer hdpr,ddpr,hdsv,ddsv
          common /time/nper,nstp(80)
          common /files/iunit(24)

c     ihedfm - code for the format which heads will be
written

          ihedfm=0

c     iddnfm - code for the format which drawdowns will be
printed

          iddnfm=0

c     ihedun - unit # to which heads will be written if
c     they are saved on disk

          ihedun=0
```

290

```
c    iddnun - unit # to which drawdowns will be written if
c    they are saved on disk

     iddnun=0

c    incode - is the head/drawdown output code (=0, all
layers
c    treated the same)

     incode=0

c    ihddfl- is a head and drawdown output flag

     ihddfl=1

c    ibudfl - budget print flag (if.ne.0 will be printed)

     ibudfl=1

c    icbcfl - cell-by-cell flow term flag

     icbcfl=1

c    hdpr - output flag for head printout

     hdpr=0

c    ddpr - output flag for drawdown printout

     ddpr=0

c    hdsv  - output flag for head save

     hdsv=0

c    ddsv - output flag for drawdown save

     ddsv=0

     open(51,file='../../data/oc.npt',status='unknown')

c    writing to output control input file

     write(51,'(4i10)')ihedfm,iddnfm,ihedun,iddnun
     do ip=1,nper
       do i=1,nstp(ip)
         write(51,'(4i10)')incode,ihddfl,ibudfl,icbcfl
         write(51,'(4i10)')hdpr,ddpr,hdsv,ddsv
       end do
     end do

     close(51)

     return
     end


*************************************************
************************
c    creates pcg2 input
     subroutine pcg2

c    mxiter- maximum  outer iterations  (=1 for linear
problems,
c    >1 for nonlinear problems)
```

```
     mxiter=90

c    iter1 - is the maximum number of inner iterations

     iter1=15

c    npcond - flag used to select the matrix preconditioning
method

     npcond=1

c    hclose - is the head change criterion for convergence,
in units of length

     hclose=0.05

c    rclose residual criterion for convergence, in unist of
cubic
c    length per time

     rclose=0.05

c    relaxation parameter used with npcond=1

     relax=1.0

c    used when npcond=2

     nbpol=2

c    iprpcg- printout interval for pcb

     iprpcg=0

c    multpcg - is a flag which controls printing from solver
c    (=0, everthing is printed)

     multpcg=0

c    ipcgcd - flag which is used when npcond=1

     ipcgcd=0

     open(51,file='../../data/pcg2.npt',status='unknown')

c    writing to pcg2 input file

     write(51,'(3i10)')mxiter,iter1,npcond

write(51,231)hclose,rclose,relax,nbpol,iprpcg,multpcg,ipc
gcd
 231  format(3f10.3,4i10)

     close(51)
     end

*************************************************
************************
c    creates recharge input file
     subroutine recharge

c conv is to convert in/yr to ft/day.


parameter(ix=160,jy=120,lay=40,conv=0.00022831*1.0)
     integer type
     dimension xd(ix),yd(jy),zd(lay)
```

291

```
      dimension locat(80),cnstnt(80),recha(ix,jy,50)
      character*20 fmtin

      common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy)
      common /time/nper,nstp(80)
      common /files/iunit(24)

c     defining variables

      nlay=1
      nrow=ir-1
      ncol=ic-1

c     nrchop - recharge option code (=3), recharge is
      applied to highest
c     active cell in each vertical column

      nrchop=3

c     irchcb - is a flag and unit number

      irchcb=0

c     inrech - recharge read flag

      inrech=0

c     inirch - only need if nrchop=2

      inirch=0

c     read in recharge values

      open(34,file='../../../0d/data/sjlin.npt',status='old')
      do i=1,4
        read(34,*)
      end do
      read(34,*)rech1
      read(34,*)rech2
      read(34,*)rech3
      read(34,*)rech4
      read(34,*)rech5
      read(34,*)rech6
      close(34)

c     initializing recharge array

      do it=1,nper
        do i=1,ncol
          do j=1,nrow
            recha(i,j,it)=0.0
          end do
        end do
      end do

c     assigning appropriate recharge to each subarea

      do it=1,nper
        if(it.le.32)fact=1.0
        if(it.eq.33.or.it.eq.41)fact=rech2/rech1
        if(it.eq.42.or.it.eq.43)fact=rech3/rech1
        if(it.eq.44.or.it.eq.45)fact=rech4/rech1
        if(it.eq.46.or.it.eq.47)fact=rech5/rech1
        if(it.ge.48)fact=rech6/rech1

        do i=1,ncol
          do j=1,nrow
```

```
            if(type(i,j).eq.3)then
              recha(i,j,it)=rech1*conv*0.8
              goto 45
            endif
            if(type(i,j).eq.21)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.43)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.22)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.44)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.23)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.45)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.24)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.47)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.25)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.46)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.26)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.46)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.27)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.46)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.28)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.47)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.29)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.45)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.30)then
              recha(i,j,it)=rech1*conv*fact

      if(it.gt.45)recha(i,j,it)=rech1*conv*(rech6/rech1)
              goto 45
            endif
            if(type(i,j).eq.31)then
              recha(i,j,it)=rech1*conv*fact
```

292

```fortran
      if(it.gt.45)recha(i,j,it)=rech1*conv*(rech6/rech1)
         goto 45
         endif
         if(type(i,j).eq.40)then
            recha(i,j,it)=rech1*conv*.6
         goto 45
         endif
45       end do
      end do
      end do

      do i=1,nper
         cnstnt(i)=1.0
      end do
      fmtin='(10g11.4)'
      iprn=0

      do i=1,nper
         locat(i)=iunit(8)
      end do

      open(1,file='.././data/rch.npt',status='unknown')
c     open(133,file='.././data/rech.txt',status='unknown')

      write(1,'(2i10)')nrchop,irchcb

      do it=1,nper
         write(1,'(2i10)')inrech,inirch

      write(1,'(i10,f10.1,a20,i10)')locat(it),cnstnt(it),fmtin,iprn
         do j=1,nrow
            write(1,fmtin)(recha(i,j,it), i=1,ncol)
         end do
c        if(it.eq.40)then
c        do i=1,ncol,3
c        do j=nrow,1,-3
c        write(133,'(2i5,2x,f12.10)')i,nrow+1-j,recha(i,j,it)
c        end do
c        end do
c        endif
      end do

      close(1)

      return
      end


      ***********************************************
      ***********************
c     creates drain input file
      subroutine drain

      parameter(ix=160,jy=120,lay=40)
      integer type
      dimension xd(ix),yd(jy),zd(lay),ncnt(50)
      dimension
      nldrn(ix*jy,50),nrdrn(ix*jy,50),ncdrn(ix*jy,50)
      dimension elev(ix*jy,50),cond(ix*jy,50),top(ix,jy),
     &   thick(ix,jy)

      common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy)
      common /time/nper,nstp(80)
      common /files/iunit(24)

c     defining variables
```

```fortran
      nlay=1
      nrow=ir-1
      ncol=ic-1

c     idrncb - is a flag and unit number on which cell-by-
cell flow terms will
c     be recorded

      idrncb=0
      do it=1,nper
         ncnt(it)=0
      end do
      open(334,file='.././data/lftop.out',status='old')
      do i=1,ncol
         read(334,'(39f10.4)')(top(i,j),j=1,nrow)
      end do
      close(334)
      open(333,file='.././data/thick.out',status='old')
      do i=1,ncol
         read(333,'(39f10.4)')(thick(i,j),j=1,nrow)
      end do
      close(333)

      do it=1,nper
         do j=1,nrow
            do i=1,ncol
               if(type(i,j).gt.0)then
               ncnt(it)=ncnt(it)+1
               nrdrn(ncnt(it),it)=j
               ncdrn(ncnt(it),it)=i
               nldrn(ncnt(it),it)=1
               elev(ncnt(it),it)=top(i,j)
               cond(ncnt(it),it)=1000.0
               go to 123
               end if
123            end do
         end do
      end do

c     mxdrn - max number of drain cells active at one time

      mxdrn=ncol*nrow

      do itt=1,nper
         mxdrn=max(ncnt(itt),mxdrn)
      end do

      open(1,file='.././data/drn.npt',status='unknown')

      write(1,'(2i10)')mxdrn,idrncb

      do itt=1,nper
         write(1,'(i10)')ncnt(itt)
         do i=1,ncnt(itt)
            write(1,'(3i10,f10.1,f10.3)')nldrn(i,itt),nrdrn(i,itt),
     .         ncdrn(i,itt),elev(i,itt),cond(i,itt)
         end do
      end do

      close(1)

      return
      end
      ***********************************************
      ***********************
c     creates input file delineating areas of landfill added
```

```fortran
      subroutine addcell

      parameter(ix=160,jy=120,lay=40)
      integer type
      dimension xd(ix),yd(jy),zd(lay),time(50)
      dimension thick(ix,jy)
      integer acell(ix,jy),izcnt(50)

      common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy)
      common /files/iunit(24)
      common /landf/ acell
      common /time/nper,nstp(80)
      nper=48

c     defining variables

      nlay=1
      nrow=ir-1
      ncol=ic-1

c     nzone- # of subarea groups added

      nzone=5

c     time - time when subarea group added
c     subarea 2 added

      time(1)=2.0*365.0

c     subarea 1 added

      time(2)=3.0*365.0

c     subarea 3&PLC added

      time(3)=5.0*365.0

c     subarea 4 added and dike added

      time(4)=38.0*365.0

c     subarea 5 added

      time(5)=40.0*365.0


c     intializing variable 'izcnt' which counts the number of
c     cells added in each subarea

      do ii=1,6
        izcnt(ii)=0
      end do

        do i=1,ncol
          do j=1,nrow
            acell(i,j)=0.
          end do
        end do

      do j=1,nrow
        do i=1,ncol
            if(type(i,j).eq.22.or.type(i,j).eq.27.
     .        or.type(i,j).eq.28)then
              izone=1
              acell(i,j)=izone
              izcnt(izone)=izcnt(izone)+1

            end if
            if(type(i,j).eq.21)then
              izone=2
              acell(i,j)=izone
              izcnt(izone)=izcnt(izone)+1
            end if
            if(type(i,j).eq.23.or.type(i,j).eq.29.or.
     .        type(i,j).eq.30)then
              izone=3
              acell(i,j)=izone
              izcnt(izone)=izcnt(izone)+1
            end if
            if(type(i,j).eq.24.or.type(i,j).eq.40.or.
     .        type(i,j).eq.31)then
              izone=4
              acell(i,j)=izone
              izcnt(izone)=izcnt(izone)+1
            end if
            if(type(i,j).eq.25)then
              izone=5
              acell(i,j)=izone
              izcnt(izone)=izcnt(izone)+1
            end if
        end do
      end do

      open(333,file='../../data/thick.out',status='old')
      do i=1,ncol
        read(333,'(39f10.4)')(thick(i,j),j=1,nrow)
      end do
      close(333)
      open(1,file='../../data/addcell.npt',status='unknown')

      write(1,'(i10)')nzone

      do iz=1,nzone
        write(1,'(i10,f10.1)')izcnt(iz),time(iz)
        l=1
          do i=1,ncol
            do j=1,nrow
              if(abs(acell(i,j)).eq.iz)then
                thr=-thick(i,j)/15.0
                write(1,'(4i4,f10.1,i5)')i,j,l,acell(i,j),thr,
     .            type(i,j)
              end if
            end do
          end do
      end do

      close(1)

      return
      end
```

The main purpose of MF2D.FOR is to place the reformed data for the two-dimensional flow model into a form which is useable in MODFLOW. It is also where the timing information, the specific yields, the conductivities, and the anisotropy are assigned to each cell. In addition, there is a subroutine (ADDCELL) which tells the model when to activate cells in the model domain to simulate the expansion of the landfill over time.

The grid set up and model discretization are designated in the main part of the program, with a copy of the same code used in the earlier preprocessors. A series of subroutines then assimilates and outputs data based on the unit numbers appropriate for each part of MODFLOW. Detailed discussion of the MODFLOW parameters and model control flags can be found in the MODFLOW users manual. Parameters pertinent to the hydrological conditions of the model area that may want to be changed for sensitivity analysis are specific yield, hydraulic conductivity, and anisotropy. Timing information can also be changed but won't have any impact on the simulation results unless input is changed with it (i.e. refining timing to include seasonal water fluctuations). Since the changing of the timing constitutes the building of a new model, it is not discussed here.

Specific yield, hydraulic conductivity, and anisotropy are all changed within the BCF2 subroutine that produces the data used by the block-centered flow package of MODFLOW. Specific yield is controlled by the array SF1 and for the purposes of the two-dimensional flow model will only be applied to the landfill cells since all

295

other cells are considered inactive. SF1 is a three-dimensional array, with one value

for each cell in the grid. The value of SF1 is dependent upon the geologic type of each

cell. The geologic type is set by the GRID2D.FOR and the RIV2D.FOR programs as

explained above. Since all of the cells other than the landfill cells are considered

inactive in the two-dimensional flow model, the only statement which needs to be

changed to adjust the specific yield is the statement,

$$\text{if(type(i,j).ge.10)sf1(i,j,1)=0.20}$$

which says that if the cell is a type 10 or greater (indicating a landfill cell), then the

value of SF1 is to be 0.20.

The hydraulic conductivity, which is input into the model, is the horizontal

component. Anisotropy is also input and the vertical component is calculated

internally. The changes to these parameters are also made in the BCF2 subroutine. A

value for the horizontal hydraulic conductivity for each cell in the model domain is

given with that value being multiplied by the thickness of the cell to give the

transmissivity of that cell. The transmissivity is stored in a three-dimensional array

for output. HYDC is the variable for the horizontal hydraulic conductivity and just

like the specific yield, will only apply to the landfill cells since all other cells are

deemed inactive in the two-dimensional flow model. The only line that will effect the

model simulation as it pertains to hydraulic conductivity is,

$$\text{if(type(i,j).ge.10)hydc=5.0e-4*hconv}$$

which says that if the cell is of type 10 or greater (indicating a landfill cell), then the value of hydc is to be 5.0e-4*hconv (hconv is a parameter statement which converts units of cm/sec to ft/day in the program).

Anisotropy is adjusted by the variable TRPY in the BCF2 subroutine. There is only one value per layer and is indicated as one over the anisotropy value (i.e. - 1/90).

If the horizontal hydraulic conductivity and the anisotropy were only changed within MF2D.FOR and not in the previous preprocessors, the simulation results would change very little if at all. Because of the structure of the two-dimensional flow model (which is a three-dimensional flow model with all cells inactive except the top layer) and the fact that all active cells are designated as river cells, the model relies on the effective conductance for each cell as its most sensitive parameter. The effective conductance is calculated in the RIV2D.FOR program and should be adjusted from there.

## Modifications to MODFLOW code:

The following has been added to the BCF2.FOR subprogram, inserted at line 228. All text in lower case represents the additions and all text in upper case part of the original code. This addition reads the file ADDCELL.NPT to determine when cells should be activated to simulate the growth of the landfill over time. This modification was written and added to the source code by Chris Berger.

```
SUBROUTINE BCF2AD(IBOUND,HOLD,BOT,WETDRY,IWDFLG,ISS,
   1          NCOL,NROW,NLAY,totim)
C
C-----VERSION 1434 14MAY1991 BCF2AD
C    *********************************************************************
C    SET HOLD TO BOT WHENEVER A WETTABLE CELL IS DRY
C    *********************************************************************
C
C    SPECIFICATIONS:
C    ---------------------------------------------------------------
C
   DIMENSION IBOUND(NCOL,NROW,NLAY),HOLD(NCOL,NROW,NLAY),
   1      BOT(NCOL,NROW,NLAY),WETDRY(NCOL,NROW,NLAY)
C
   COMMON /FLWCOM/LAYCON(80)
C    ---------------------------------------------------------------
C
c    checking to see if cell addition file needs to be opened
c
   if(iacflag.eq.0)then
    open(71,file='.././data/addcell.npt',status='old')
    read(71,'(i10)')nzone
    iacflag=1
    icurr=1
   end if
   if(icurr.le.nzone)then
    read(71,'(i10,f10.1)')icnt,time
    if(totim.gt.time)then
     do ki=1,icnt
       read(71,'(4i4,f10.1)')ii,jj,ll,iadd,thresh
       if(iadd.lt.0)then
         ibound(ii,jj,ll)=ll
       end if
       if(iadd.gt.0)then
         ibound(ii,jj,ll)=ll
              hold(ii,jj,ll)=-2.0
       end if
     end do
     icurr=icurr+1
    else
     backspace(71)
    end if
   end if
```

The following has been added to the BAS1.FOR subroutine, inserted at line 307. It is used to produce an output file that read by the three-dimensional flow model

to establish the fixed head boundary condition in the landfill. This modification was

written and added to the source code by Chris Berger.

```
c writing out heads at each time steps at particular cross-sections
c
    if(ichris.eq.0)then
      ichris=1
      open(89,file='../../output/head.dat',status='unknown')
    endif
    do l=1,nlay
      write(89,'(30f6.1)')((HNEW(i,j,l),i=ncol,1,-1),j=nrow,1,-1)
    end do
```

## Appendix C  - Three-dimensional Flow Model Source Code and Documentation

The three-dimensional flow model is designed to simulate groundwater flow in a fully three-dimensional flow field within the local landfill area. The setup for the three-dimensional flow model is simpler than the two-dimensional flow model discussed in Appendix B because there is no need for the extra programs which establish the landfill cells as river cells with a unique conductance value and to convert the data to the two-dimensional grid. The three-dimensional flow model requires four preprocessors; one to set up and establish the geology of the area within the grid framework of the model (GRIDTYPE.FOR); the second to establish which cells are treated as river cells and to calculate the conductance for each river cell (R1ADD.FOR); the third to convert the output from the regional flow model into the grid coordinates of the three-dimensional flow model (BCHEAD.FOR); and the fourth is to place the input data into a form which can be used by MODFLOW and to establish boundary conditions (MFPRE.FOR). The three-dimensional flow model is structured as follows:
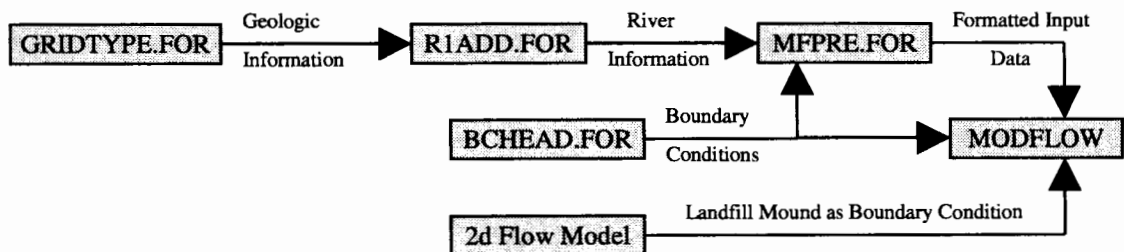


**Figure 92 - 3d flow model program flow chart. GRIDTYPE.FOR and R1ADD.FOR set up the geology and river cell information. BCHEAD.FOR uses input from the regional flow model and adapts it for use as side and lower boundary conditions for the 3d flow model.**

The MODFLOW code used in the three-dimensional flow model is the same code version which is used in the two-dimensional flow model. There were changes made to the original MODFLOW source code to account for varying boundary conditions in the landfill and the Troutdale Formation. These changes were made to the BCF2.FOR subprogram where two subroutines were added to account for the changing fixed-head boundary conditions over time (the landfill is treated as a fixed head boundary in the three-dimensional flow model which means that as the landfill mound grows over time, as determined by the two-dimensional flow model simulation, the fixed head boundary must be updated to reflect this growth), as well as account for the growth of the landfill area over time. CALL statements were added to the main MODFLOW program to go to the added subroutines in BCF2.FOR at the appropriate times. In addition to the above, a small change was made to the subprogram which handles recharge (RCH1.FOR) to bypass recharge from being added to 'perched' cells and instead be added to the next active cell below it. The additions to each subprogram along with explanations and the line number where they appear in each subprogram are shown at the end of this appendix.

The GRIDTYPE.FOR file reads in the stratigraphy from files supplied to us by METRO. This file is almost identical to the GRID2D.FOR file used in the two-dimensional flow model except the grid coordinates are input through a read statement from a file called GRIDDEF.IN which describes the origin of the grid (the upper, northwest corner of the modeling domain) in terms of the coordinate system used in the METRO files, as well as describes the cell height, width, and depth for each cell in

301

the grid. The GRIDDEF.IN file is also used in the R1ADD.FOR program as well as the MFPRE.FOR program and allows for easy changing of the grid layout without having to edit each separate program. R1ADD.FOR uses the SILTTOP.DAT and the SILTTHICK.DAT files as discussed in Appendix B. These are the output files from SILT_TOP.FOR and SILT_THICK.FOR which have been changed to reflect the larger grid resolution in the three-dimensional flow model. See Appendix B for an example of the source code for these programs.

In order to input the recharge over time, a separate input file called RECH.IN is used. This file allows for varying recharge over time as well as adding the ability to 'phase out' each subarea from receiving recharge as the landfill goes through the closing process. The source code for each preprocessor is shown below.

| Program Name | Input Files Needed | Output Files Produced | Description of Output |
|---|---|---|---|
| BCHEAD.FOR | GRIDDEF.IN<br>GRIDDEF.REG<br>HEAD2.5094<br>HEAD3.5094 | CHECK.DAT<br><br>BCHEAD.NPT | Verification file to make sure grid is set correctly<br>Heads in Troutdale and Gravel formations in 3d grid layout |
| GRIDTYPE.FOR | GRIDDEF.IN<br>WATER<br>LANDFILL<br>SILT<br>SAND<br>GRAV<br>TROUT<br>UNDIFF<br>SUBAREA<br>ROAD<br>DIKE | CHECK.DAT<br><br>GTYPE.OUT | Verification file to make sure grid is set correctly<br>Data file delineating the geology within the model domain, assigning a 'type' code to each cell address depending upon where it lies in the physical domain. |
| R1ADD.FOR | GTYPE.OUT<br>SILTTOP.OUT<br>SILTTHICK.OUT<br>RIVERND2.CJB | CHECK.DAT<br><br>RIV.NPT<br><br>STAGE.DAT | Verification file to make sure grid is set correctly<br>Cell addresses describing all surface water bodies<br>Water level data over time. |
| MFPRE.FOR | GRIDDEF.IN<br>GRIDDEF.REG<br>GTYPE.OUT<br>BCPRE82.NPT<br>SILTTHICK.OUT<br>RECH.IN<br>HEAD2D.DAT | CHECK.DAT<br><br>BAS.NPT<br>BCF2.NPT<br>BALANCE.NPT<br><br>OC.NPT<br>PCG2.NPT<br>RCH.NPT<br>DRN.NPT<br>ADDCELL.NPT<br>RIV2.NPT | Verification file to make sure grid is set correctly<br>Data for basic input used in MODFLOW<br>Data for block centered flow package<br>File used in post-processing for assessing water budget<br>Output control file<br>Data for conjugate gradient package<br>Data for recharge package<br>Data for drain package<br>Data describing growth of landfill over time<br>Final version of RIV.NPT |

**Table 7 - Listing of programs, files needed to run programs, and output for each program necessary to run the 3d flow model. The GRIDDEF.\* files are user input files designating the grid layout and spacing. The HEAD\*.5094 files are output files from the regional flow model describing the heads in the gravel aquifer (HEAD2.5094) and the Troutdale formation (HEAD3.5094) from 1950 to 1994. The needed data files listed without an extension (i.e. GRAV) are files delineating the geology of the area and are supplied by METRO for our use in the project. The RIVERND2.CJB is also supplied by METRO and delineates the surface water bodies in the landfill region. The HEAD2D.DAT is the output from the 2d site specific mounding model.**

## BCHEAD.FOR:

```
c    Reads output from regional flow model and converts to
c    3d flow model grid layout for use as boundary
condition.
     parameter(ix=250,jy=200,lay=60,itst=44)
     dimension  grav(ix,jy,itst),trout(ix,jy,itst)
     dimension  gravf(ix,jy,itst),troutf(ix,jy,itst)
     dimension  sumg(ix,jy),sumt(ix,jy)
     dimension  avgg(ix,jy),avgt(ix,jy)
     dimension xd(ix),yd(jy),zd(lay),iunit(24)
     dimension xint(ix),yint(jy),zint(lay)
     dimension xs(ix),ys(jy),zs(lay)
     dimension xd2(90),yd2(90),xc2(90),yc2(90)

     open(1,file='~/modflow/3d/landdat/griddef.in',
     &    status='old')

c    reading file defining model cells
     read(1,*)
c    reading # of boundaries of constant cell length along
x-axis (ic), along
c    y-axis (ir), , along z-axis (il)
     read(1,*)ici,iri,ili
     read(1,*)
c    (xm,ym) - location of origin
c    in terms of model coordinate system system
     read(1,*)xm,ym
     read(1,*)
c    reading distances of intervals of constant cell width
along x-axis from
c    point defined by xm
     read(1,*)xs(1)
     do i=2,ici
       read(1,*)xs(i),xint(i-1)
     end do
     read(1,*)
c    reading distances of intervals of constant cell width
along y-axis from
c    point defined by ym
     read(1,*)ys(1)
     do j=2,iri
       read(1,*)ys(j),yint(j-1)
     end do
c    reading elevations of intevalss of constant cell width
along z-axis
     read(1,*)zs(1)
     do k=2,ili
       read(1,*)zs(k),zint(k-1)
     end do

c    counting number of cell boundaries
     ic=1
     ir=1
     il=1
     do i=2,ici
       ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
     end do
     do j=2,iri
       ir=ir+nint((ys(j)-ys(j-1))/yint(j-1))
     end do
     do k=2,ili
       il=il+nint((zs(k-1)-zs(k))/zint(k-1))
     end do
c    calculating location of cell boundary along axis
     xd(1)=xs(1)
     yd(1)=ys(1)
```

```
     zd(1)=zs(1)
     icnt=1
     jcnt=1
     kcnt=1
     do i=2,ic
       xd(i)=xd(i-1)+xint(icnt)
       if(nint(xd(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
     end do
     do j=2,ir
       yd(j)=yd(j-1)+yint(jcnt)
       if(nint(yd(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
     end do
     do k=2,il
       zd(k)=zd(k-1)-zint(kcnt)
       if(nint(zd(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
     end do

     open(9,file='check.dat',status='unknown')
     write(9,*)ic,ir,il
     write(9,*)
     do i=1,ic
       write(9,*)xd(i)
     end do
     write(9,*)
     do i=1,ir
       write(9,*)yd(i)
     end do
     write(9,*)
     do i=1,il
       write(9,*)zd(i)
     end do
     close(9)
     ncol=ic-1
     nrow=ir-1
     nlay=il-1

c    computing bc heads - using 2d model output data
c    reading in grid definition file for 2d model

     open(33,file='~/modflow/3d/landdat/griddef.reg',
     &    status='old')
c    reading file defining 2d model cells
     read(33,*)
c    reading # of cell boundaries along x-axis (ic), # cell
bounds along y-axis (ir),
c    and # of cell boundaries along z-axis (il)
     read(33,*)ic2,ir2,il2
     read(33,*)
c    (xm,ym) - location of lower left hand corner grid pt.
c    in terms of model coordinate system system
     read(33,*)xm2,ym2
     read(33,*)
c    reading distances of cell boundaries along x-axis from
c    point defined by xm
     do i=1,ic2
       read(33,*)xd2(i)
     end do
     read(33,*)
c    reading distances of cell boundaries along y-axis from
c    point defined by ym
     do j=1,ir2
       read(33,*)yd2(j)
     end do
     close(33)

c    calculating cell centers of 2d model
     do i=1,ic2-1
```

304

```fortran
        xc2(i)=(xd2(i)+xd2(i+1))/2.0
      end do
      do j=1,ir2-1
        yc2(j)=(yd2(j)+yd2(j+1))/2.0
      end do

      ncol2=ic2-1
      nrow2=ir2-1

      open(8,file='~/modflow/largegrid/landdat/head2.5094',
     &    status='old')
      open(9,file='~/modflow/largegrid/landdat/head3.5094',
     &    status='old')

      do j=1,nrow
        do i=1,ncol
          sumg(i,j)=0.0
          sumt(i,j)=0.0
          avgg(i,j)=0.0
          avgt(i,j)=0.0
        end do
      end do
      icnt=0
      do it=1,itst
        do j=nrow2,3,-1
          read(8,'(30f9.3)')(grav(i,j,it),i=1,ncol2-1)
          read(9,'(30f9.3)')(trout(i,j,it),i=1,ncol2-1)
        end do
      end do

55    continue

      do it=1,itst
        do j=1,nrow
          do i=1,ncol

c    using 2d model data to set up heads for gravel and
trout aquifers
c    determining cell coordinates within 2d model domain
          xcell=(xd(i)+xd(i+1))/2.0 + xm
          ycell= (26730.0-ym) + (yd(j)+yd(j+1))/2.0
c    locating 3d model gridpoint within 2d model domain
          do ii=1,ncol2
            if(xcell.lt.xc2(ii))then
              i2d=ii-1
              go to 32
            end if
          end do
32          continue
          do jj=1,nrow2
            if(ycell.lt.yc2(jj))then
              j2d=jj-1
              go to 33
            end if
          end do
33          continue

c    interpolating between 2d model grid points to
calculate head
c    within 3d model

          h1=grav(i2d,j2d,it)
          h2=grav(i2d+1,j2d,it)
          h3=grav(i2d,j2d+1,it)
          h4=grav(i2d+1,j2d+1,it)

          d11=ycell-yc2(j2d)
```

```fortran
          d12=yc2(j2d+1)-ycell
          dt1=d11+d12
          s1=((dt1-d11)*h1 + (dt1-d12)*h3)/dt1

          d21=xcell-xc2(i2d)
          d22=xc2(i2d+1)-xcell
          dt2=d21+d22
          s2=((dt2-d21)*h1 + (dt2-d22)*h2)/dt2

          d31=ycell-yc2(j2d)
          d32=yc2(j2d+1)-ycell
          dt3=d31+d32
          s3=((dt3-d31)*h2 + (dt3-d32)*h4)/dt3

          d41=xcell-xc2(i2d)
          d42=xc2(i2d+1)-xcell
          dt4=d41+d42
          s4=((dt4-d41)*h3 + (dt4-d42)*h4)/dt4

          w1=xcell-xc2(i2d)
          w3=xc2(i2d+1)-xcell
          w2=ycell-yc2(j2d)
          w4=yc2(j2d+1)-ycell

          wt=w1+w2+w3+w4

          gravf(i,j,it)=(s1*w1+s2*w2+s3*w3+s4*w4)/wt

c    doing the same for the trout
          h1=trout(i2d,j2d,it)
          h2=trout(i2d+1,j2d,it)
          h3=trout(i2d,j2d+1,it)
          h4=trout(i2d+1,j2d+1,it)

          d11=ycell-yc2(j2d)
          d12=yc2(j2d+1)-ycell
          dt1=d11+d12
          s1=((dt1-d11)*h1 + (dt1-d12)*h3)/dt1

          d21=xcell-xc2(i2d)
          d22=xc2(i2d+1)-xcell
          dt2=d21+d22
          s2=((dt2-d21)*h1 + (dt2-d22)*h2)/dt2

          d31=ycell-yc2(j2d)
          d32=yc2(j2d+1)-ycell
          dt3=d31+d32
          s3=((dt3-d31)*h2 + (dt3-d32)*h4)/dt3

          d41=xcell-xc2(i2d)
          d42=xc2(i2d+1)-xcell
          dt4=d41+d42
          s4=((dt4-d41)*h3 + (dt4-d42)*h4)/dt4

          w1=xcell-xc2(i2d)
          w3=xc2(i2d+1)-xcell
          w2=ycell-yc2(j2d)
          w4=yc2(j2d+1)-ycell

          wt=w1+w2+w3+w4

          troutf(i,j,it)=(s1*w1+s2*w2+s3*w3+s4*w4)/wt

          end do
        end do
      end do
```

305

```
c  Producing Output

   open(7,file='bchead.npt',status='unknown')

   do it=1,itst
     do j=1,nrow
       write(7,'(30f9.3)')(gravf(i,j,it),i=1,ncol)
     end do
     do j=1,nrow
       write(7,'(30f9.3)')(troutf(i,j,it),i=1,ncol)
     end do
   end do

   stop
   en
```

## GRIDTYPE.FOR:

```
c   psu-chris berger 1993
c   St. Johns Landfill modeling project
c   This program uses stratigraphy files delineating upper
c   and lower layer boundaries generated by companion
c   program 'transfrm.for'
c   and assigns each model cell
c   a number corresponding to the type of hydrogeological
c    unit within
c   which the computation pt. lies
c   hgu - # of hydrogeological units
    integer hgu
    parameter(ix=200,jy=200,lay=60,hgu=7,dltx=15.0)
c   ix - maximum number of computational grid pts. in x-
direction
c   jy - maximum number of computational grid pts. in y-
direction
c   lay - maximum number of layers

    integer type(ix,jy,lay)

    dimension xd(ix),yd(jy),zd(lay),xs(ix),ys(jy),zs(lay)
    dimension xint(ix),yint(jy),zint(lay)
c   common /info/xd,yd,zd,xm,ym,il,ic,ir,ix,jy,lay,dltx

c   do i=1,8
c   ifile(i)=i
c   end do

c   open(ifile,file='final.txt',status='old')
    open(51,file='~/modflow/3d/landdat/griddef.in',
    &   status='old')

c   initializing 'type' array to zero which will be
c   used as a flag later to help determine if grid pt. has
been missed
    do i=1,ix
      do j=1,jy
        do k=1,lay
          type(i,j,k)=0
        end do
      end do
    end do

c   reading file defining model cells
    read(51,*)
c   reading # of boundaries of constant cell length along
x-axis (ic), along
c   y-axis (ir), , along z-axis (il)
    read(51,*)ici,iri,ili
    read(51,*)
c   (xm,ym) - location of origin
c   in terms of model coordinate system system
    read(51,*)xm,ym
    read(51,*)
c   reading distances of intervals of constant cell width
along x-axis from
c   point defined by xm
    read(51,*)xs(1)
    do i=2,ici
      read(51,*)xs(i),xint(i-1)
    end do
    read(51,*)
c   reading distances of intervals of constant cell width
along y-axis from
c   point defined by ym
```

306

```fortran
      read(51,*)ys(1)
      do j=2,iri
        read(51,*)ys(j),yint(j-1)
      end do
c     reading elevations of intevalss of constant cell width
along z-axis
      read(51,*)zs(1)
      do k=2,ili
        read(51,*)zs(k),zint(k-1)
      end do

c     counting number of cell boundaries
      ic=1
      ir=1
      il=1
      do i=2,ici
        ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
      end do
      do j=2,iri
        ir=ir+nint((ys(j)-ys(j-1))/yint(j-1))
      end do
      do k=2,ili
        il=il+nint((zs(k-1)-zs(k))/zint(k-1))
      end do
c     calculating location of cell boundary along axis
      xd(1)=xs(1)
      yd(1)=ys(1)
      zd(1)=zs(1)
      icnt=1
      jcnt=1
      kcnt=1
      do i=2,ic
        xd(i)=xd(i-1)+xint(icnt)
        if(nint(xd(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
      end do
      do j=2,ir
        yd(j)=yd(j-1)+yint(jcnt)
        if(nint(yd(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
      end do
      do k=2,il
        zd(k)=zd(k-1)-zint(kcnt)
        if(nint(zd(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
      end do

      open(91,file='check.dat',status='unknown')
      write(91,*)ic,ir,il
      write(91,*)
      do i=1,ic
        write(91,*)xd(i)
      end do
      write(91,*)
      do i=1,ir
        write(91,*)yd(i)
      end do
      write(91,*)
      do i=1,il
        write(91,*)zd(i)
      end do

c     reading files containing top and bottom of layer info
and assigning
c     layer type # to computational grid cells

      call assign(type,xd,yd,zd,xm,ym,il,ic,ir)

c     defining extent of water bodies to determine
```

```fortran
c     which surface boundary computational cells are below
surface water-
c     algorithm then calculates parameters used by river
package
c     rewind(ifile)
c     call surface_water(type,ifile,xd,yd,zd,xm,ym,il,ic,ir)

      stop
      end


c*****************************************************
c************************
      subroutine assign(type,xd,yd,zd,xm,ym,il,ic,ir)

      integer hgu
      parameter(ix=200,jy=200,lay=60,hgu=7,dltx=15.0)
      parameter(ncol=1513,nrow=1783)
      integer type(ix,jy,lay)
      dimension xd(ix),yd(jy),zd(lay),ztemp(hgu)
      integer
lfrow(ncol),watrow(ncol),siltrow(ncol),sandrow(ncol),
     -   gravrow(ncol),troutrow(ncol),unconrow(ncol),
     -   subarea(ncol),road(ncol),dike(ncol)
      kk=hgu

      open(1,file='~/modflow/3d/landdat/water',
     &   status='old')
      open(2,file='~/modflow/3d/landdat/landfill',
     &   status='old')
      open(3,file='~/modflow/3d/landdat/silt',
     &   status='old')
      open(4,file='~/modflow/3d/landdat/sand',
     &   status='old')
      open(5,file='~/modflow/3d/landdat/grav',
     &   status='old')
      open(6,file='~/modflow/3d/landdat/trout',
     &   status='old')
      open(7,file='~/modflow/3d/landdat/undiff',
     &   status='old')
c     open(8,file='~/modflow/3d/landdat/crb',
c     &   status='old')

c     skipping first six lines of each file

      do i=1,hgu
        do l=1,6
          read(i,*)
        end do
      end do

c     initializing layer type number to zero
      do i=1,ic
        do j=1,ir
          do l=1,il
            type(i,j,l)=0
          end do
        end do
      end do

      do 300 ii=nrow,1,-1

        read(1,*)(watrow(jj), jj=1,ncol)
        read(2,*)(lfrow(jj), jj=1,ncol)
        read(3,*)(siltrow(jj), jj=1,ncol)
        read(4,*)(sandrow(jj), jj=1,ncol)
        read(5,*)(gravrow(jj), jj=1,ncol)
```

307

```
        read(6,*)(troutrow(jj), jj=1,ncol)
        read(7,*)(unconrow(jj), jj=1,ncol)
c    read(8,*)(crbrow(jj), jj=1,ncol)


        do 305 jj=1,ncol
          ztemp(1)=real(watrow(jj))
          ztemp(2)=real(lfrow(jj))
          ztemp(3)=real(siltrow(jj))
          ztemp(4)=real(sandrow(jj))
          ztemp(5)=real(gravrow(jj))
          ztemp(6)=real(troutrow(jj))
          ztemp(7)=real(unconrow(jj))
c    ztemp(8)=real(crbrow(jj))

          xx=real(jj-1)*dltx
          yy=real(ii-1)*dltx
c    translating from model grid coordinates to origin
(upper-left corner)
c    defined by computational grid
          x=xx-xm
          y=ym-yy
c    determining if data grid point has corresponding cell
center
c    scanning through the coordinates of all the model cell
boundaries
c    and determing if any cells are centered about current
read data coordinates
        do i=1,ic-1
          xtemp=(xd(i)+xd(i+1))/2.0
          ixdist=int(xtemp/dltx)*int(dltx)
          if(ixdist.eq.nint(x))then
            do j=1,ir-1
              ytemp=(yd(j)+yd(j+1))/2.0
              iydist=int(ytemp/dltx)*int(dltx)
              if(iydist.eq.nint(y))then
                do k=1,il-1
                  zelev=(zd(k)+zd(k+1))/2.0
                  do l=2,kk
c    if(zelev.le.ztemp(l-1).and.zelev.gt.ztemp(l))then
                    if(zelev.le.ztemp(l-1))then
                      type(i,j,k)=l-1
c    go to 15
                    end if
                    if(zelev.lt.ztemp(kk))then
                      type(i,j,k)=kk
c    go to 15
                    end if
                  end do
15                continue
                end do
                go to 20
              end if
            end do
          end if
        end do
20      continue
305     continue
300   continue

c    reading file defining subareas

      open(27,file='~/modflow/3d/landdat/subarea',status='old')

        do ik=1,6
          read(27,*)
```

```
        end do

        do 404 ii=nrow,1,-1
          read(27,*)(subarea(jj), jj=1,ncol)
          do 405 jj=1,ncol
            if(subarea(jj).eq.0)go to 405
            xx=real(jj-1)*dltx
            yy=real(ii-1)*dltx
c    translating from model grid coordinates to origin
(upper-left corner)
c    defined by computational grid
            x=xx-xm
            y=ym-yy
c    determining in which model cell subarea gridpoint is
located
          do i=1,ic-1
            if(x.ge.xd(i).and.x.lt.xd(i+1))then
              do j=1,ir-1
                if(y.ge.yd(j).and.y.lt.yd(j+1))then
                  do k=1,il-1
                    if(type(i,j,k).eq.2.or.type(i,j,k).ge.10)then
                      type(i,j,k)=20+subarea(jj)
c    if(subarea(jj).eq.1)type(i,j,k)=21
c    if(subarea(jj).eq.2)type(i,j,k)=26
c    if(subarea(jj).eq.3)type(i,j,k)=22
c    if(subarea(jj).eq.4)type(i,j,k)=24
c    if(subarea(jj).eq.5)type(i,j,k)=25
c    if(subarea(jj).eq.6)type(i,j,k)=23
c    if(subarea(jj).eq.7)type(i,j,k)=27
                    end if
                  end do
                  go to 405
                end if
              end do
            end if
          end do

405     end do
404   end do

      close(27)

c    reading file defining road

      open(28,file='~/modflow/3d/landdat/road',status='old')

        do ik=1,6
          read(28,*)
        end do

        do 504 ii=nrow,1,-1
          read(28,*)(road(jj), jj=1,ncol)
          do 505 jj=1,ncol
            if(road(jj).eq.0)go to 505
            xx=real(jj-1)*dltx
            yy=real(ii-1)*dltx
c    translating from model grid coordinates to origin
(upper-left corner)
c    defined by computational grid
            x=xx-xm
            y=ym-yy
c    determining in which model cell the road gridpoint is
located
          do i=1,ic-1
            if(x.ge.xd(i).and.x.lt.xd(i+1))then
              do j=1,ir-1
                if(y.ge.yd(j).and.y.lt.yd(j+1))then
```

308

```
          do k=1,il-1
            if(type(i,j,k).eq.2.or.
   &              type(i,j,k).ge.10)then
              if(type(i,j,k).eq.24.or.
   &               type(i,j,k).eq.25)then
                type(i,j,k)=31
              else
                type(i,j,k)=30
              end if
            end if
          end do
          go to 505
        end if
      end do
    end if
  end do

505   end do
504 end do

    close(28)

c   reading file defining dike

    open(29,file='~/modflow/3d/landdat/dike',status='old')

    do ik=1,6
      read(29,*)
    end do

    do 604 ii=nrow,1,-1
      read(29,*)(dike(jj), jj=1,ncol)
      do 605 jj=1,ncol
        if(dike(jj).eq.0)go to 605
        xx=real(jj-1)*dltx
        yy=real(ii-1)*dltx
c   translating from model grid coordinates to origin
(upper-left corner)
c   defined by computational grid
        x=xx-xm
        y=ym-yy
c   determining in which model cell the dike gridpoint is
located
        do i=1,ic-1
          if(x.ge.xd(i).and.x.lt.xd(i+1))then
            do j=1,ir-1
              if(y.ge.yd(j).and.y.lt.yd(j+1))then
                do k=1,il-1
                  if(type(i,j,k).eq.2.or.type(i,j,k).ge.10)then
                    if(zd(k+1).ge.15.0)then
                      type(i,j,k)=40
                    end if
                  end if
                end do
                go to 605
              end if
            end do
          end if
        end do

605   end do
604 end do

    close(29)

c   reading file defining landfill
```

```
c     open(39,file='~/modflow/3d/landdat/site',status='old')

c     do ik=1,6
c     read(39,*)
c     end do

c     do 704 ii=nrow,1,-1
c     read(39,*)(site(jj), jj=1,ncol)
c     do 705 jj=1,ncol
c     if(site(jj).ne.0)go to 705
c     xx=real(jj-1)*dltx
c     yy=real(ii-1)*dltx
c   translating from model grid coordinates to origin
(upper-left corner)
c   defined by computational grid
c     x=xx-xm
c     y=ym-yy
c   determining if data grid point has corresponding cell
center
c   scanning through the coordinates of all the model cell
boundaries
c   and determing if any cells are centered about current
read data coordinates
c     do i=1,ic-1
c     xtemp=(xd(i)+xd(i+1))/2.0
c     ixdist=int(xtemp/dltx)*int(dltx)
c     if(ixdist.eq.nint(x))then
c     do j=1,ir-1
c     ytemp=(yd(j)+yd(j+1))/2.0
c     iydist=int(ytemp/dltx)*int(dltx)
c     if(iydist.eq.nint(y))then
c     do k=1,il-1
c     type(i,j,k)=0
c     end do
c     go to 70
c     end if
c     end do
c     end if
c     end do
c     70    continue
c     705   continue
c     704   continue
c     close(39)

    open(89,file='gtype.out',status='unknown')
    do i=1,ic-1
      do j=1,ir-1
        do k=1,il-1
          write(89,122)i,j,k,type(i,j,k)
122       format(4(i4))
        end do
      end do
    end do
    close(89)

    return
    end


c**************************************************
c************************
c   subroutine uses files which define areal extent of water
bodies to determine
c   which surface boundary computational cells are below
surface water-
```

```
c   algorithm then calculates parameters used by river
package

c   subroutine
surface_water(type,ifile,xd,yd,zd,xm,ym,il,ic,ir)
c   integer hgu
c   parameter(ix=184,jy=146,lay=37,hgu=7,dltx=15.0)
c   integer type(ix,jy,lay)
c   dimension xd(ix),yd(jy),zd(lay),ztemp(hgu),iztemp(hgu)
c   dimension waterarea(ix,jy),rivercell(ix,jy,lay)
c   integer watercnt(ix,jy)

c   kk=hgu
c   initializing array values
c   do i=1,ix
c   do j=1,jy
c   waterarea(i,j)=0.0
c   watercnt(i,j)=0
c   do l=1,lay
c   rivercell(i,j,l)=0.0
c   end do
c   end do
c   end do
c
c   10   read(ifile,*,end=30)xx,yy,(ztemp(l), l=1,kk)
c   10   read(ifile,*,end=30)ixx,iyy,(iztemp(l), l=1,kk)
c   xx=real(ixx)
c   yy=real(iyy)
c   do l=1,kk
c   ztemp(l)=real(iztemp(l))
c   end do
c   c   checking to see if current grid coordinate has a
waterbody layer
c   if(nint(ztemp(1)).eq.nint(ztemp(2)))go to 10
c   checking to see if layer top signifies water body rather
than
c   landfill layer (each have same type#) - assuming water
body have elevation
c   less than 10' msl
c   if(ztemp(1).gt.10.0)go to 10
c
c   translating from model grid coordinates to origin
(lower-left corner)
c   defined by computational grid
c   x=xx-xm
c   y=yy-ym
c   determining if surface water grid point is located
within model - and
c   if it is, finding which model cell above which it lies
c   if(x.gt.(xd(ic)+(xd(ic)-xd(ic-1))))go to 10
c   if(y.gt.(yd(ir)+(yd(ir)-yd(ir-1))))go to 10
c   if(x.lt.(0.0-(xd(2)-xd(1))))go to 10
c   if(y.lt.(0.0-(yd(2)-yd(1))))go to 10

c   do i=1,ic
c   determining x-dimensional range occupied by model
cell
c   if(i.eq.1)then
c   xmin=0.0-(xd(2)-xd(1))
c   xmax=(xd(1)+xd(2))/2.
c   go to 12
c   end if
c   if(i.eq.ic)then
c   xmin=(xd(ic)+xd(ic-1))/2.
c   xmax=xd(ic)+(xd(ic)-xd(ic-1))
c   go to 12
c   end if

c   xmin=(xd(i)+xd(i-1))/2.
c   xmax=(xd(i)+xd(i+1))/2.
c
c   12       if(x.ge.xmin.and.x.lt.xmax)then
c   determining y-dimensional range occupied by model
cell
c   do j=1,ir
c   if(j.eq.1)then
c   ymin=0.0-(yd(2)-yd(1))
c   ymax=(yd(1)+yd(2))/2.
c   go to 13
c   end if
c   if(j.eq.ir)then
c   ymin=(yd(ir)+yd(ir-1))/2.
c   ymax=yd(ir)+(yd(ir)-yd(ir-1))
c   go to 13
c   end if
c   ymin=(yd(j)+yd(j-1))/2.
c   ymax=(yd(j)+yd(j+1))/2.

c   13       if(y.ge.ymin.and.y.lt.ymax)then
c   watercnt(i,j)=watercnt(i,j)+1
c   waterarea(i,j)=waterarea(i,j)+dltx*dltx
c   go to 20
c   end if
c   end do
c   end if
c   end do
c   20   continue
c   go to 10
c   30   continue
c
c
open(25,file='~/modflow/3d/landdat/river.out',status='new'
)

c   do i=1,ic
c   do j=1,ir
c   if(watercnt(i,j).eq.0)go to 41
c   do k=2,il
c   if(type(i,j,k-1).eq.1.and.type(i,j,k).ne.1)then
c   if(k.eq.il)then
c   zmin=(zd(il)+zd(il-1))/2.
c   zmax=zd(il)+(zd(il)-zd(il-1))
c   go to 92
c   end if
c   zmin=(zd(k)+zd(k-1))/2.
c   zmax=(zd(k)+zd(k+1))/2.
c   92       thickness=zmax-zmin
c   rivercell(i,j,k)=waterarea(i,j)/thickness
c   write(25,111)i,j,k,watercnt(i,j),waterarea(i,j),
c   -       rivercell(i,j,k),zd(k)
c111 format(3i4,i5,2f10.0,f8.0)
c   end if
c   end do
c41   end do
c   end do

c   return
c   end
```

310

## R1ADD.FOR:

```
c    Produces river cell output distinguishing between
c    North, Columbia, and Blind sloughs, Smith and Bybee
c    Lakes.
c    Portland State University - 1994, Chris Berger
c    As modified June, 1995 - Thomas Lowry

     parameter
(ncolm=250,nrowm=200,nlaym=60,nperm=20)
     dimension x(ncolm+1),y(nrowm+1),z(nlaym+1),
     *       zz(nrowm,ncolm)
     dimension perlen(nperm)
     dimension itype(ncolm*nrowm)
     real  kcond(10)
     dimension stage(ncolm*nrowm)
     dimension noder(ncolm*nrowm)
     dimension ir(ncolm*nrowm),jr(ncolm*nrowm)
     dimension cond(ncolm*nrowm)
     integer type(ncolm,nrowm,nlaym)
     dimension iunit(24)
     dimension siltz(ncolm,nrowm),dist(ncolm,nrowm)
     dimension
xint(ncolm+1),yint(nrowm+1),zint(nlaym+1)
     dimension xs(ncolm+1),ys(nrowm+1),zs(nlaym+1)
     nper=48
     iunit(13)=14
     open(98,file='~/modflow/3d/landdat/gtype.out',
     &   status='old')

c  reading file with grid type #'s

23   read(98,122,end=33)i,j,k,itemp
     type(i,j,k)=itemp
122     format(4(i4))
     go to 23
33   continue


c    Assinging # of boundaries of constant cell length along
x-axis (ic),
c    along y-axis (ir), , along z-axis (il)

     ici=2
     iri=2
     ili=4

c    (xm,ym) - Assigning location of origin in terms of
model coordinate
c    system

     xm=8100
     ym=15300

c    Assigning distances of intervals of constant cell width
along x-axis from
c    point defined by xm

     xs(1)=0
     xs(2)=7650
     xint(1)=150

c    Assigning distances of intervals of constant cell width
along y-axis from
c    point defined by ym

     ys(1)=0
```

```
     ys(2)=5850
     yint(1)=150

c    Assigning elevations of intevals of constant cell width
along z-axis

     zs(1)=95
     zs(2)=-30
     zs(3)=-50
     zs(4)=-350
     zint(1)=5
     zint(2)=10
     zint(3)=25

c    counting number of cell boundaries

     ic=1
     irr=1
     il=1
     do i=2,ici
       ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
     end do
     do j=2,iri
       irr=irr+nint((ys(j)-ys(j-1))/yint(j-1))
     end do
     do k=2,ili
       il=il+nint((zs(k-1)-zs(k))/zint(k-1))
     end do

c calculating location of cell boundary along axis

     x(1)=xs(1)
     y(1)=ys(1)
     z(1)=zs(1)
     icnt=1
     jcnt=1
     kcnt=1
     do i=2,ic
       x(i)=x(i-1)+xint(icnt)
       if(nint(x(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
     end do
     do j=2,irr
       y(j)=y(j-1)+yint(jcnt)
       if(nint(y(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
     end do
     do k=2,il
       z(k)=z(k-1)-zint(kcnt)
       if(nint(z(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
     end do

     open(91,file='check.dat',status='unknown')
     write(91,*)ic,irr,il
     write(91,*)
     do i=1,ic
       write(91,*)x(i)
     end do
     write(91,*)
     do i=1,irr
       write(91,*)y(i)
     end do
     write(91,*)
     do i=1,il
       write(91,*)z(i)
     end do
     close(91)
```

```fortran
      ncol=ic-1
      nrow=irr-1
      nlay=il-1

c initializing bottom of cell elevations to an unreasonable #

      do i=1,nrowm
       do j=1,ncolm
        zz(i,j)=999.0
       end do
      end do

      open(31,file='~/modflow/3d/landdat/silttop.dat',
     &   status='old')
      do j=1,nrow
       read(31,'(30f7.1)')(siltz(i,j), i=1,ncol)
      end do
      close(31)

      open(32,file='~/modflow/3d/landdat/siltthick.dat',
     &   status='unknown')
      do j=1,nrow
       read(32,'(30f7.1)')(dist(i,j), i=1,ncol)
      end do
      close(32)

      do 30 iper=1,nper
30      perlen(iper)=30.4
      do 40 i=1,10
40      kcond(i)=0.1

      x00=xm
      y00=ym
      istagefile=0
      call  river (ncol,nrow,nlay,ncolm,nrowm,nlaym,nperm,
     *     x,y,z,zz,perlen,kcond,cond,itype,stage,noder,ir,jr,
     *     x00,y00,istagefile,nper,iunit,type,siltz,dist)
      stop
      end
cccccccccccc River package
c
c Documentation
c
c ncol,nrow,nlay = number of computational cells in
column, row, and
c  vertical directions
c ncolm,nrowm,nlaym = max limits of ncol,nrow,nlay
c nperm = max number of stress periods
c (x(i),i=1,ncol+10, computational discretization in x
c (y(i),i=1,nrow+10, computational discretization in y
c (z(i),i=1,nlay+10, computational discretization in z
c zz(i,j), minimum river bottom elevation in cell (i,j)
c     k=1 refuse layer
c     k=2 surface water bodies
c     k=3 silt
c     k=4 sand
c     k=5 grvel
c     k=6 trout
c     k=7 undiff
c     k=8 crb
c     itype=1, Smith/Bybee Lake
c     itype=2, Ramsey Lake
c     itype=3, Columbia River
c     itype=4, Willamette
c     itype=5, Multnomah Channel
c     itype=6, North Portland Harbor
c     itype=7, Columbia Slough
```

```fortran
c     itype=8, north Slough
c     itype=9, Blind Slough
c     itype=10,  small pond near blind slough at SE corner
of landfill
c ir(ico),ico=1,maxnumber river cell, incidence list of river
cell
c     = i index of the (ico)th river cell
c jr(ico),ico=1,maxnumber river cell, incidence list of river
cell
c     = j index of the (ico)th river cell
c itype(ico) =  type of river for the (ico)th river cell
c noder(ico) = number of 15x15 river
c     in the (ico)th computational river cell
c stage(ico) = riverstage at the (ico)th river cell
c cond(ico) = river conductance of the (ico)th river cell
c x00,y00 = coordinates of lower left corner of model area
in state system
c istagefile = 1, river stage input from a data file
c istagefile = 0, river stages generated in this routine
c nper = number of stress periods
c
      subroutine river
(ncol,nrow,nlay,ncolm,nrowm,nlaym,nperm,
     *     x,y,z,zz,perlen,kcond,cond,itype,stage,noder,ir,jr,
     *     x00,y00,istagefile,nper,iunit,type,siltz,dist)
      dimension x(ncolm+1),y(nrowm+1),z(nlaym+1),
     *      zz(nrowm,ncolm),iunit(24)
      dimension perlen(nperm)
      dimension itype(ncolm*nrowm)
      dimension stage(ncolm*nrowm)
      dimension noder(ncolm*nrowm)
      dimension ir(ncolm*nrowm),jr(ncolm*nrowm)
      dimension cond(ncolm*nrowm)
      dimension siltz(ncolm,nrowm),dist(ncolm,nrowm)
      integer type(ncolm,nrowm,nlaym)
      real kcond(10)
      dxs=15.
      dys=15.
      open (11,file="~/modflow/3d/landdat/rivernd2.cjb",
     -     status="old")
      open (12,file="~/modflow/3d/landdat/stage.dat",
     &   status="unknown")
      open (13,file="~/modflow/3d/landdat/riv.npt",
     &   status="unknown")

      mxrivr=ncolm*nrowm
      irivcb=0
      write (13,"(2I10)")mxrivr,irivcb

      do 510 i=1,ncol*nrow
        noder(i)=0
510   continue
        nriv=0
167   read(11,'(2f8.0,i2,f5.0)',end=168) xss,yss,nr,bot
      call xtoi (xss-x00,i,ncol,x,ncolm+1)
      call xtoi (y00-yss,j,nrow,y,nrowm+1)
      if(i.eq.0.or.j.eq.0)go to 167
      zz(i,j)=min(zz(i,j),bot)
      nnn=(i-1)*nrow+j
      noder(nnn) = noder(nnn)+1
      if(noder(nnn).eq.1) goto 169
      do 310 ii=1,nriv
        if(nnn.eq.(ir(ii)-1)*nrow+jr(ii)) goto 167
310     continue
169     nriv=nriv+1
      ir(nriv)=i
      jr(nriv)=j
```

```
          itype(nriv)=nr
          goto 167

c skip the first two record of x and y corrdinates

 168    time=0.0

        do 540 iper=1,nper

c for first 38 time period iper=1-8, getting rid of small
pond (itype=10)

        if(iper.le.38)then
          itmp=0
          do i=1,nriv
           if(itype(i).ne.10)itmp=itmp+1
          end do
        end if
        if(iper.ge.39)itmp=nriv

        write(13,"(I10)")itmp

        time=time+(perlen(iper)+perlen(iper+1))*0.5
        w=2.*3.14159/365.
        if(istagefile.eq.0)then
        do 191 i=1,nriv
c
c Reset river and lake levels
c
        if(itype(i).ge.7.and.itype(i).le.9)then
          if(iper.le.31)rivlev=10.2
          if(iper.ge.32)rivlev=9.74
        endif
        if(itype(i).eq.1)then
          if(iper.le.31)rivlev=10.2
          if(iper.ge.32.and.iper.le.46)rivlev=11.74
          if(iper.ge.47)rivlev=11.74
        endif


        if(itype(i).eq.10)stage(i)=15.0
        if(itype(i).ne.10)stage(i)=rivlev
        if(itype(i).eq.1)stage(i)=rivlev
191     continue
c
c Let Bybee lake fluctuate until 1980, then fix it until 1997
c
               if(iper.ge.30.and.iper.lt.48)then
        do i=1,nriv
          if(itype(i).eq.1)stage(i)=rivlev+3.00
        end do
        endif

        if(iper.eq.1)then
          write (12,*) nriv
          write (12,"(5x,20I7)") (ico,ico=1,nriv)
          write (12,"(5x,20I7)") (ir(ico),ico=1,nriv)
          write (12,"(5x,20I7)") (jr(ico),ico=1,nriv)
          endif
         write(12,"(I5,20(1x,f6.1))") iper,(stage(i),i=1,nriv)
        else
          read(12,*)
          read(12,*)
          read(12,*) (ir(i),i=1,nriv)
          read(12,*) (jr(i),i=1,nriv)
          read(12,*) (stage(i),i=1,nriv)
```

```
          endif

       do 530 ico=1,nriv


c  ii - col #
        ii=ir(ico)

c  jj - row #

        jj=jr(ico)

c    scanning gtype.out file to ensure that rivercell is
c    top most active one

        do l=1,nlay
           if(type(ii,jj,l).ge.3.and.type(ii,jj,l).le.7)then
             layer=l
             go to 137
           end if
        end do
137     continue

c  check to make sure that  river cell is near water

        wmp=stage(ico)
423        if(z(layer+1).gt.wmp)then
           layer=layer+1
           go to 423
          end if

        nnn=(ii-1)*nrow+jj
        thick=1.0

        if(type(ii,jj,layer).ge.3)then
          thick=siltz(ii,jj)-(z(layer)-z(layer+1))/2.0
        end if
        if(thick.lt.1.0)thick=1.0
        cond(ico) =
(kcond(itype(ico))/thick)*noder(nnn)*dxs*dys

c   skipping over landfill pond for 38 stress periods

        if(itype(ico).eq.10.and.iper.le.38)go to 530
        if(itype(ico).eq.10)zz(ii,jj)=10.0
        if(itmp.ge.1) write(13,"(3I10,F10.2,g10.2,f10.0,i10)")
     *   layer,jj,ii,stage(ico),cond(ico),zz(ii,jj),itype(ico)
 530    continue
 540    continue
        return
        end

**************************************************
************************
    subroutine xtoi (x0,i0,nx,x,nm)
    dimension x(nm)
    i0=0
    do 100 i=1,nx+1
      if(x(i).ge.x0) then
        i0=i-1
        return
      endif
100    continue
    return
    en
```

313

## MFPRE.FOR:

```fortran
c    mfpre.for    modflow preprocessor
c    St. Johns Landfill modflow modeling study
c    Portland State University - Chris Berger -1994
c    revised by Tom Lowry - 10/95
c    reads input file "gtype.out" defining type of layer each
c    computational node is associated with and creates
input files
c    for modflow model
***THIS IS THE BASE CASE PREPROCESSOR*****
      integer hgu
      parameter(ix=60,jy=50,lay=40,hgu=8)
      integer type(ix,jy,lay)
      dimension xd(ix),yd(jy),zd(lay),iunit(24)
      dimension xint(ix),yint(jy),zint(lay)
      dimension xs(ix),ys(jy),zs(lay)
      dimension nstp(80)

      common /info/xd,yd,zd,xm,ym,il,ic,ir,type
      common /time/nper,nstp
      common /files/iunit
c     common /river/rhead


      open(98,file='.././././commondata/gtype.out',status='old')

c    Assinging # of boundaries of constant cell length along
x-axis (ic),
c    along y-axis (ir), , along z-axis (il)

      ici=2
      iri=2
      ili=4

c    (xm,ym) - Assigning location of origin in terms of
model coordinate
c    system

      xm=8100
      ym=15300

c    Assigning distances of intervals of constant cell width
along x-axis from
c    point defined by xm

      xs(1)=0
      xs(2)=7650
      xint(1)=150

c    Assigning distances of intervals of constant cell width
along y-axis from
c    point defined by ym

      ys(1)=0
      ys(2)=5850
      yint(1)=150

c    Assigning elevations of intevals of constant cell width
along z-axis

      zs(1)=95
      zs(2)=-30
      zs(3)=-50
      zs(4)=-350
      zint(1)=5
      zint(2)=10
```

```fortran
      zint(3)=25

c    counting number of cell boundaries

      ic=1
      ir=1
      il=1
      do i=2,ici
        ic=ic+nint((xs(i)-xs(i-1))/xint(i-1))
      end do
      do j=2,iri
        ir=ir+nint((ys(j)-ys(j-1))/yint(j-1))
      end do
      do k=2,ili
        il=il+nint((zs(k-1)-zs(k))/zint(k-1))
      end do

c    calculating location of cell boundary along axis

      xd(1)=xs(1)
      yd(1)=ys(1)
      zd(1)=zs(1)
      icnt=1
      jcnt=1
      kcnt=1
      do i=2,ic
        xd(i)=xd(i-1)+xint(icnt)
        if(nint(xd(i)).ge.nint(xs(icnt+1)))icnt=icnt+1
      end do
      do j=2,ir
        yd(j)=yd(j-1)+yint(jcnt)
        if(nint(yd(j)).ge.nint(ys(jcnt+1)))jcnt=jcnt+1
      end do
      do k=2,il
        zd(k)=zd(k-1)-zint(kcnt)
        if(nint(zd(k)).le.nint(zs(kcnt+1)))kcnt=kcnt+1
      end do

      open(91,file='.././data/check.dat',status='unknown')
      write(91,*)ic,ir,il
      write(91,*)
      do i=1,ic
        write(91,*)xd(i)
      end do
      write(91,*)
      do i=1,ir
        write(91,*)yd(i)
      end do
      write(91,*)
      do i=1,il
        write(91,*)zd(i)
      end do
      close(91)


c    reading file with grid type #'s

 23   read(98,*,end=33)i,j,k,itemp
      type(i,j,k)=itemp
      go to 23
 33   continue

c    calling subroutine which types cells for addition of
c    landfill subareas over time

      call addcell
```

314

```
c    calling subroutine which creates basic package input

     call basic

c    calling subroutine which creates body centered flow
     package input

     call bcf2

c    calling subroutine which creates output control file

     call output

c    calling subroutine which creates pcg2 control file

     call pcg2

c    calling subrouting which creates recharge input file
      .
     call recharge

     call drain

     stop
     end

***************************************************
*************************
c    creates basic package input

     subroutine basic
     parameter(ix=60,jy=50,lay=40,hgu=8)
     integer type,ibound(ix,jy,lay),acell,rtype
     dimension xd(ix),yd(jy),zd(lay),shead(ix,jy,lay)
     dimension perlen(80),clfhead(ix,jy)
     dimension xd2(90),yd2(90),xc2(90),yc2(90)
     dimension avgt(90,90),avgg(90,90)
     character*80 headng1,headng2
     character*20 fmtini,fmtinsh

     common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy,lay)
     common /time/nper,nstp(80)
     common /files/iunit(24)
c     common /river/rhead(ix,jy,lay)
     common /landf/ acell(ix,jy,lay)
     common /rivertype/ rtype(ix,jy,lay)

c    defining variables
c    headng1, headng2 - first two lines of basic input file
     (file headings)

     headng1='Modflow Basic Package Input File'
     headng2='St. Johns Landfill Project - PSU 1994'
     nlay=il-1
     nrow=ir-1
     ncol=ic-1

c    NPER - number of stress periods

     nper=48
c    itmuni - time unit code, days=4

     itmuni=4

c    iunit - array which contains unit numbers for package
     inputs
c    iunit(1) - block-centered flow package BCF2
```

```
c    iunit(2) - well package WEL
c    iunit(3) - drain package DRN
c    iunit(4) - river package RIV
c    iunit(5) - evapotranspiration package EVT
c    iunit(6) -reserved for transient leakage package
c    iunit(7) - general head boundary package GHB
c    iunit(8) - recharge package RCH
c    iunit(9) - SIP package
c    iunit(10) - reserved for additional solver
c    iunit(11) - SSOR Package
c    iunit(12) - Output control option  OC
c    iunit(13) - Pre-conditioned conjugate gradient
     package PCG2
c    iunit(14) - Stream Package report STR1
c    iunit(22) - mt3d linking module LKMT18
c    initializing all values in  IUNIT array to zero

     do i=1,24
       iunit(i)=0
     end do

c    assigning packages which are used nonzero unit
     numbers

     iunit(1)=11
     iunit(3)=13
     iunit(4)=14
     iunit(8)=18
     iunit(12)=22
     iunit(13)=23
     iunit(22)=32

c    IAPART - indicates whether array BUFF is separate
     from array RHS
c    IAPART =0 same space,        IAPART=1 separate

     iapart=0

c    istrt -indicates whether starting heads are saved (=0
     no, .ne.0 yes)

     istrt=1

c    locati - location of data to be read into integer arrays
c    (unit number=1) for basic package

     locati=1
     iconsti=1

c    fmtini - format for integer input arrays

     fmtini='(30i3)'
c    iprni - flag determining how input of integer arrays is
     written
c    output files iprni=2 corresponds to 40i2

     iprni=-1

c
c    defining boundary array IBOUND(i,j,k) using type #
c    IBOUND<0  constant head cell
c    IBOUND=0  inactive cell
C     IBOUND>0 variable head cell
c    if type#=0, ibound=0 (undefined)
c    if type#=1, ibound=0 (water)
c    if type#=2, ibound>1 (landfill)
c    if type#=3, ibound>1 (silt layer)
```

315

```
c      if type#=4, ibound>1 (sand layer)
c      if type#=5, ibound>1 (gravel layer)
c      if type#=6, ibound=0 (troutdale aquifer)
c      if type#=7, ibound=0 (unconsolidated)
c      if type#=8, ibound=0 (columbia river basalt)
c      if type#>10, ibound>1 (some subdivision of landfill)
       open(77,file='.././2d/output/head.dat',status='old')
       read(77,'(30f6.1)')((clfhead(i,j),i=ncol,1,-1),j=nrow,1,-
1)

     do l=1,nlay
       do j=1,nrow
         do i=1,ncol
           ibound(i,j,l)=0
           if(type(i,j,l).eq.0)ibound(i,j,l)=0
           if(type(i,j,l).eq.1)ibound(i,j,l)=0
           if(type(i,j,l).eq.2)ibound(i,j,l)=-99
           if(type(i,j,l).eq.3)ibound(i,j,l)=l
           if(type(i,j,l).eq.4)ibound(i,j,l)=l
           if(type(i,j,l).eq.5)ibound(i,j,l)=l
           if(type(i,j,l).eq.6)ibound(i,j,l)=l
           if(type(i,j,l).eq.7)ibound(i,j,l)=0

if(type(i,j,l).ge.8.and.type(i,j,l).lt.10)ibound(i,j,l)=0
           if(type(i,j,l).ge.10)then
             if(zd(l).gt.clfhead(i,j))then
               ibound(i,j,l)=0
             else
               ibound(i,j,l)=-99
             endif
           endif
           if(type(i,j,l).eq.40)ibound(i,j,l)=l
           if(zd(l).gt.10.0)ibound(i,j,l)=0

c    Assigning constant head cells to sides of sand, gravel,
c    and troutdale

           if(type(i,j,l).ge.4.and.type(i,j,l).le.6)then
             if(i.eq.ncol.or.j.eq.nrow)ibound(i,j,l)=-l
             if(i.eq.1.or.j.eq.1)ibound(i,j,l)=-l
           endif

c    assigning zero ibound values to areas occupied by old
lake

           if(acell(i,j,l).ne.0)then
             if(acell(i,j,l).gt.0)ibound(i,j,l)=0
           end if
         end do
       end do
     end do

c    hnoflo - head value assigned to inactive cells

     hnoflo=10.0

c    assigning initial heads - using 2d model output data
c    reading in grid definition file for 2d model

c
open(43,file='/u/lowry/modflow/3d/landdat/griddef.2d',
c   &    status='old')
     open(43,file='.././data/griddef.2d',
     &    status='old')

c    reading file defining 2d model cells

     read(43,*)
```

```
c    reading # of cell boundaries along x-axis (ic), # cell
bounds
c    along y-axis (ir),  and # of cell boundaries along z-axis
(il)

     read(43,*)ic2,ir2,il2
     read(43,*)

c    (xm,ym) - location of lower left hand corner grid pt.
c    in terms of model coordinate system system

     read(43,*)xm2,ym2
     read(43,*)

c    reading distances of cell boundaries along x-axis from
c    point defined by xm

     do i=1,ic2
       read(43,*)xd2(i)
     end do
     read(43,*)

c    reading distances of cell boundaries along y-axis from
c    point defined by ym

     do j=1,ir2
       read(43,*)yd2(j)
     end do
     close(43)

c    calculating cell centers of 2d model
     do i=1,ic2-1
       xc2(i)=(xd2(i)+xd2(i+1))/2.0
     end do
     do j=1,ir2-1
       yc2(j)=(yd2(j)+yd2(j+1))/2.0
     end do

c    reading in head data from 2d model
     ncol2=ic2-1
     nrow2=ir2-1
     open(44,file='.././data/bound.dat',status='old')
     do j=nrow2,3,-1
       read(44,'(30f9.3)')(avgg(i,j),i=1,ncol2-1)
     end do

     do j=nrow2,3,-1
       read(44,'(30f9.3)')(avgt(i,j),i=1,ncol2-1)
     end do
     close(44)

     do l=1,nlay
       do j=1,nrow
         do i=1,ncol

c    using 2d model data to set up heads for gravel and
trout aquifers

           if(type(i,j,l).eq.5)then

c    determining cell coordinates within 2d model domain

           xcell=(xd(i)+xd(i+1))/2.0 + xm
           ycell= (26730.0-ym) + (yd(j)+yd(j+1))/2.0

c    locating 3d model gridpoint within 2d model domain
```

```
          do ii=1,ncol2                          36        continue
            if(xcell.lt.xc2(ii))then                      do jj=1,nrow2
              i2d=ii-1                                       if(ycell.lt.yc2(jj))then
              go to 32                                         j2d=jj-1
            end if                                            go to 37
          end do                                           end if
32        continue                                       end do
          do jj=1,nrow2                         37        continue
            if(ycell.lt.yc2(jj))then
              j2d=jj-1                                    h1=avgt(i2d,j2d)
              go to 33                                    h2=avgt(i2d+1,j2d)
            end if                                        h3=avgt(i2d,j2d+1)
          end do                                          h4=avgt(i2d+1,j2d+1)
33        continue
                                                          d11=ycell-yc2(j2d)
c   interpolating between 2d model grid points to         d12=yc2(j2d+1)-ycell
calculate head                                            dt1=d11+d12
c   within 3d model                                        s1=((dt1-d11)*h1 + (dt1-d12)*h3)/dt1

          h1=avgg(i2d,j2d)                                d21=xcell-xc2(i2d)
          h2=avgg(i2d+1,j2d)                              d22=xc2(i2d+1)-xcell
          h3=avgg(i2d,j2d+1)                              dt2=d21+d22
          h4=avgg(i2d+1,j2d+1)                             s2=((dt2-d21)*h1 + (dt2-d22)*h2)/dt2

          d11=ycell-yc2(j2d)                              d31=ycell-yc2(j2d)
          d12=yc2(j2d+1)-ycell                            d32=yc2(j2d+1)-ycell
          dt1=d11+d12                                     dt3=d31+d32
           s1=((dt1-d11)*h1 + (dt1-d12)*h3)/dt1            s3=((dt3-d31)*h2 + (dt3-d32)*h4)/dt3

          d21=xcell-xc2(i2d)                              d41=xcell-xc2(i2d)
          d22=xc2(i2d+1)-xcell                            d42=xc2(i2d+1)-xcell
          dt2=d21+d22                                     dt4=d41+d42
           s2=((dt2-d21)*h1 + (dt2-d22)*h2)/dt2            s4=((dt4-d41)*h3 + (dt4-d42)*h4)/dt4

          d31=ycell-yc2(j2d)                              w1=xcell-xc2(i2d)
          d32=yc2(j2d+1)-ycell                            w3=xc2(i2d+1)-xcell
          dt3=d31+d32                                     w2=ycell-yc2(j2d)
           s3=((dt3-d31)*h2 + (dt3-d32)*h4)/dt3           w4=yc2(j2d+1)-ycell

          d41=xcell-xc2(i2d)                              wt=w1+w2+w3+w4
          d42=xc2(i2d+1)-xcell
          dt4=d41+d42                                      shead(i,j,l)=(s1*w1+s2*w2+s3*w3+s4*w4)/wt
           s4=((dt4-d41)*h3 + (dt4-d42)*h4)/dt4           end if

          w1=xcell-xc2(i2d)                               shead(i,j,l)=10.00
          w3=xc2(i2d+1)-xcell               c     endif
          w2=ycell-yc2(j2d)                       end do
          w4=yc2(j2d+1)-ycell                   end do
                                            end do
          wt=w1+w2+w3+w4
                                         c   locat - location of data to be read into initial head
           shead(i,j,l)=(s1*w1+s2*w2+s3*w3+s4*w4)/wt   arrays.
          end if                          c   (unit number=0) means head is equal to const

          if(type(i,j,l).eq.6)then            locatsh=1
                                              constsh=1.0
c   determining cell coordinates within 2d model domain   fmtinsh='(10g11.4)'
                                              iprnsh=0
          xcell=(xd(i)+xd(i+1))/2.0 + xm
          ycell= (26730.0-ym) + (yd(j)+yd(j+1))/2.0   c   perlen - length of stress period
          do ii=1,ncol2
            if(xcell.lt.xc2(ii))then            perlen(1)=365.0
              i2d=ii-1                          perlen(2)=365.0
              go to 36                          perlen(3)=365.0
            end if                              perlen(4)=365.0
          end do                               perlen(5)=365.0
                                                perlen(6)=365.0
```

```
      perlen(7)=365.0
      perlen(8)=365.0
      perlen(9)=365.0
      perlen(10)=365.0
      perlen(11)=365.0
      perlen(12)=365.0
      perlen(13)=365.0
      perlen(14)=365.0
      perlen(15)=365.0
      perlen(16)=365.0
      perlen(17)=365.0
      perlen(18)=365.0
      perlen(19)=365.0
      perlen(20)=365.0
      perlen(21)=365.0
      perlen(22)=365.0
      perlen(23)=365.0
      perlen(24)=365.0
      perlen(25)=365.0
      perlen(26)=365.0
      perlen(27)=365.0
      perlen(28)=365.0
      perlen(29)=365.0
      perlen(30)=365.0
      perlen(31)=365.0
      perlen(32)=365.0
      perlen(33)=365.0
      perlen(34)=365.0
      perlen(35)=365.0
      perlen(36)=365.0
      perlen(37)=365.0
      perlen(38)=365.0
      perlen(39)=365.0
      perlen(40)=365.0
      perlen(41)=365.0
      perlen(42)=365.0
      perlen(43)=365.0
      perlen(44)=365.0
      perlen(45)=365.0
      perlen(46)=365.0
      perlen(47)=365.0
      perlen(48)=8760.0

c     nstp - number of time steps in stress period

      nstp(1)=1
      nstp(2)=1
      nstp(3)=1
      nstp(4)=1
      nstp(5)=1
      nstp(6)=1
      nstp(7)=1
      nstp(8)=1
      nstp(9)=1
      nstp(10)=1
      nstp(11)=1
      nstp(12)=1
      nstp(13)=1
      nstp(14)=1
      nstp(15)=1
      nstp(16)=1
      nstp(17)=1
      nstp(18)=1
      nstp(19)=1
      nstp(20)=1
      nstp(21)=1
      nstp(22)=1
```

```
      nstp(23)=1
      nstp(24)=1
      nstp(25)=1
      nstp(26)=1
      nstp(27)=1
      nstp(28)=1
      nstp(29)=1
      nstp(30)=1
      nstp(31)=1
      nstp(32)=1
      nstp(33)=1
      nstp(34)=1
      nstp(35)=1
      nstp(36)=1
      nstp(37)=1
      nstp(38)=1
      nstp(39)=1
      nstp(40)=1
      nstp(41)=1
      nstp(42)=1
      nstp(43)=1
      nstp(44)=1
      nstp(45)=1
      nstp(46)=1
      nstp(47)=1
      nstp(48)=24

c     tsmult - multiplier for the length of successive time
steps

      tsmult=1.0

      open(1,file='../../data/bas.npt',status='unknown')

c     writing to modflow input file for basic package

      write(1,'(a80)')headng1
      write(1,'(a80)')headng2
      write(1,'(5i10)')nlay,nrow,ncol,nper,itmuni
      write(1,'(24i3)')iunit
      write(1,'(2i10)')iapart,istrt

      do l=1,nlay
        write(1,'(i10,i10,a20,i10)')locati,iconsti,fmtini,iprni
        do j=1,nrow
          write(1,fmtini)(ibound(i,j,l),i=1,ncol)
        end do
      end do

      write(1,'(f10.0)')hnoflo

      do l=1,nlay

write(1,'(i10,f10.1,a20,i10)')locatsh,constsh,fmtinsh,iprnsh
        do j=1,nrow
          write(1,fmtinsh)(shead(i,j,l),i=1,ncol)
        end do
      end do

      do i=1,nper
        write(1,'(f10.1,i10,f10.1)') perlen(i),nstp(i),tsmult
      end do

      close(1)

      return
      end
```

```
**************************************************
***********************
c    creates bcf2 package input
     subroutine bcf2
c    hconv converts cm/sec to ft/day


parameter(ix=60,jy=50,lay=40,hgu=8,hconv=2834.6278)
     integer type,acell
     dimension xd(ix),yd(jy),zd(lay),laycon(lay),
     .    delr(ix),delc(jy),sf1(ix,jy,lay),bot(lay),
     .    hy(ix,jy,lay),vcont(ix,jy,lay),sf2(ix,jy,lay),
     .    top(lay),thresh(ix,jy,lay),tran(ix,jy,lay),hydc(50)
     dimension siltgrav(ix,jy),vanis(ix,jy,lay)
     character*20 fmtinr,fmtinc,fmtin1,fmtin2,fmtinh,
     .    fmtinv,fmtina,fmtinw

     common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy,lay)
     common /files/iunit(24)
     common /landf/acell(ix,jy,lay)


c    defining variables

     nlay=il-1
     nrow=ir-1
     ncol=ic-1

c    ISS - steady-state flag, if ISS=0 transient

     iss=0

c    ibcfcb - flag and unit number, if =0 cell by cell flow
terms
c    will not be printed or recorded

     ibcfcb=0

c    hdry - head assigned to cells which are converted to
dry

     hdry=888.0

c    iwdflg - flag which determines if wetting capacity
active(=1)

     iwdflg=1

c    wetfct - wetting factor

     wetfct=1.0

c    iwetit - iteration interval for attempting to wet cells

     iwetit=1

c    ihdwet - flag to determine which equation to use to
c    determine if cell is wet

     ihdwet=1

c    laycon - array that contains layer type, if layer is
above minus 5 ft,
c    laycon=3; below minus 5 ft, laycon=0

     do l=1,nlay
       if(zd(l).lt.-5.0)laycon(l)=0
```

```
       if(zd(l).ge.-5.0)laycon(l)=3
     end do

c    trpy - array defining anisotropic factor for each layer,
=1 for isotropic
c    control card for trpy: locats=0,constnts=1.0

     locats=0
     constnts=1.0

c    delr - vector defining cell widths along rows

     do i=1,ncol
       delr(i)=xd(i+1)-xd(i)
     end do
     locatr=iunit(1)
     cnstntr=1.0
     fmtinr='(15f7.1)'
     iprnr=0

c    delc - vector defining cell widths along colummns

     do j=1,nrow
       delc(j)=yd(j+1)-yd(j)
     end do
     locatc=iunit(1)
     cnstntc=1.0
     fmtinc='(15f7.1)'
     iprnc=0

     open(335,file='../../../0d/data/sjlin.npt',status='old')
     read(335,*)hhyd
     read(335,*)aanis
     read(335,*)sy
     close(335)

c    sf1 - primary storage coefficient.  If laycon=1, sf1 -
specfic yield
c    If laycon=3,0; sf1 - confined storage coefficient

     do l=1,nlay
       do j=1,nrow
         do i=1,ncol
           if(laycon(l).eq.1)then
             sf1(i,j,l)=0.1
             if(type(i,j,l).eq.1)sf1(i,j,l)=0.1
             if(type(i,j,l).eq.2)sf1(i,j,l)=0.15
             if(type(i,j,l).ge.10)sf1(i,j,l)=sy
             if(type(i,j,l).eq.3)sf1(i,j,l)=0.1
             if(type(i,j,l).eq.4)sf1(i,j,l)=0.2
             if(type(i,j,l).eq.5)sf1(i,j,l)=0.25
             if(type(i,j,l).eq.6)sf1(i,j,l)=0.15
             if(type(i,j,l).eq.7)sf1(i,j,l)=0.15
             if(type(i,j,l).eq.8)sf1(i,j,l)=0.15
           else if(laycon(l).eq.3.or.laycon(l).eq.0)then
             sf1(i,j,l)=1.0e-6
             if(type(i,j,l).eq.2)sf1(i,j,l)=1.0e-6
             if(type(i,j,l).ge.10)sf1(i,j,l)=1.0e-6
             if(type(i,j,l).eq.3)sf1(i,j,l)=1.0e-6
             if(type(i,j,l).eq.4)sf1(i,j,l)=1.0e-6
             if(type(i,j,l).eq.5)sf1(i,j,l)=1.0e-6
             if(type(i,j,l).eq.6)sf1(i,j,l)=1.0e-6
             if(type(i,j,l).eq.7)sf1(i,j,l)=1.0e-6
             if(type(i,j,l).eq.8)sf1(i,j,l)=1.0e-6
             if(type(i,j,l).eq.1)sf1(i,j,l)=1.0e-6
           end if
         end do
```

319

```
      end do
    end do
    locat1=iunit(1)
    cnstnt1=1.0
    fmtin1='(10g11.4)'
    iprn1=-1

c    tran - transmissivity
c    assigning conductivities to layer types

    hydc(1)=1000
    hydc(2)=1.0e-3*hconv
    hydc(3)=hhyd*hconv
    hydc(5)=6.35e-2*hconv
    hydc(4)=hydc(5)/2
    hydc(6)=1.0e-3*hconv
    hydc(7)=1.0e-6*hconv
    hydc(8)=1.0e-6*hconv
    hydc(21)=1.0e-3*hconv
    hydc(22)=1.0e-3*hconv
    hydc(23)=1.0e-3*hconv
    hydc(24)=1.0e-3*hconv
    hydc(25)=1.0e-3*hconv
    hydc(26)=1.0e-3*hconv
    hydc(27)=1.0e-3*hconv
    hydc(28)=1.0e-3*hconv
    hydc(29)=1.0e-3*hconv
    hydc(30)=1.0e-3*hconv
    hydc(31)=1.0e-3*hconv
    hydc(40)=1.0e-7*hconv
c
c Assign hydraulic conductivity to 3d array
c
    do i=1,ncol
      do j=1,nrow
        do k=1,nlay
          hy(i,j,k)=hydc(type(i,j,k))
        end do
      end do
    end do
c
c Reassign hydraulic conductivity to silt below landfill
c
    do i=1,ncol
      do j=1,nrow
        do k=1,nlay
          if(type(i,j,k+1).ge.3.and.type(i,j,k+1).le.4.and.
    &        type(i,j,k).gt.20)then
            do kk=k+1,nlay
              if(type(i,j,kk).ne.3)goto 335
              hy(i,j,kk)=hydc(type(i,j,kk))/2
335           continue
            end do
          endif
          goto 340
        end do
340     continue
      end do
    end do

    do l=1,nlay
      do j=1,nrow
        do i=1,ncol
          thick=zd(l)-zd(l+1)
          tran(i,j,l)=hy(i,j,l)*thick
        end do
```

```
      end do
    end do

    locata=iunit(1)
    cnstnta=1.0
    fmtina='(10g11.4)'
    iprna=-1

    locath=iunit(1)
    cnstnth=1.0
    fmtinh='(10g11.4)'
    iprnh=-1

c    defining layer bottoms

    do l=1,nlay
      bot(l)=zd(l+1)
    end do
    locatb=0

c    reading in distance from top of silt to top of gravel

    open(77,file='../../data/siltthick.dat',
    &    status='old')
    do j=1,nrow
      read(77,'(30f7.1)')(siltgrav(i,j), i=1,ncol)
    end do
    close(77)

c    vcont - vertical hydraulic conductivity divided by layer
thickness
c    vanis/anis - vertical anistropy factor

    vmin=1e30
    vmax=-1e30
    anis=(1/aanis)
    do l=1,nlay-1
      do j=1,nrow
        do i=1,ncol
          if(type(i,j,l).eq.0)then
            vcont(i,j,l)=0.0
            go to 125
          end if
          vht=hy(i,j,l)
          vhb=hy(i,j,l+1)
          vanis(i,j,l)=anis
          if(type(i,j,l).eq.0.or.type(i,j,l).eq.1)vanis(i,j,l)=
    &        (1/1)
c
c    Change anisotropy under landfill
c
          if(type(i,j,l).ge.20.and.type(i,j,l+1).eq.3)then
            do kk=l+1,nlay-1
              if(type(i,j,kk).ne.3)goto 544
              vanis(i,j,kk)=(anis*2)
            end do
          end if
544       continue
c
c    Calculate vertical conductance (harmonic mean
divided by thickness)
c
          ztop=(zd(l)-zd(l+1))
          zbot=(zd(l+1)-zd(l+2))

          vcont(i,j,l)=vanis(i,j,l)*((2*vht*vhb)
    &        /(ztop*vhb+zbot*vht))
```

```
            if(l.ge.2)then
              if(type(i,j,l).eq.5.and.type(i,j,l-1).eq.1)then
                vht=hydc(3)
                vhb=hy(i,j,l)
                zbot=(zd(l)-zd(l+1))
                ztop=siltgrav(i,j)
                vcont(i,j,l)=vanis(i,j,l)*((2*vht*vhb)
     &                /(ztop*vhb+zbot*vht))
              endif
            end if
125         continue
          end do
        end do
      end do
      locatv=iunit(1)
      cnstntv=1.0
      fmtinv='(10g11.4)'
      iprnv=-1
```

c    sf2 -secondary storage coefficient, write only for
layers where
c    laycon is 2 or 3 - always specific yield

```
      do l=1,nlay
        do j=1,nrow
          do i=1,ncol
            sf2(i,j,l)=0.1
            if(type(i,j,l).eq.1)sf2(i,j,l)=0.1
            if(type(i,j,l).eq.2)sf2(i,j,l)=0.15
            if(type(i,j,l).ge.10)sf2(i,j,l)=sy
            if(type(i,j,l).eq.3)sf2(i,j,l)=0.1
            if(type(i,j,l).eq.4)sf2(i,j,l)=0.2
            if(type(i,j,l).eq.5)sf2(i,j,l)=0.25
            if(type(i,j,l).eq.6)sf2(i,j,l)=0.15
            if(type(i,j,l).eq.7)sf2(i,j,l)=0.15
            if(type(i,j,l).eq.8)sf2(i,j,l)=0.15
          end do
        end do
      end do
      locat2=iunit(1)
      cnstnt2=1.0
      fmtin2='(10g11.4)'
      iprn2=-1
```

c    top - layer tops,  write only when laycon is 2 or 3

```
      do l=1,nlay
        top(l)=zd(l)
      end do
      locatt=0
```

c    wetdry - combination of wetting threshold and a flag to
indicate
c    which neighboring cells can cause a cell to become
wet.
c    write if laycon is 1 or 3 and wetting capacity active
(iwdflg.ne.0)

```
      do l=1,nlay
        do j=1,nrow
          do i=1,ncol
            thresh(i,j,l)=-(zd(l)-zd(l+1))/1.0
            if(type(i,j,l).eq.0)thresh(i,j,l)=0.0
            if(type(i,j,l).eq.1)thresh(i,j,l)=0.0
```

c    assigning thresholds to area of landfill which was

c    occupied by old lake

```
            if(acell(i,j,l).gt.0)then
              thresh(i,j,l)=0.0
            end if
          end do
        end do
      end do
      locatw=iunit(1)
      cnstntw=1.0
      fmtinw='(10g11.4)'
      iprnw=-1

      open(1,file='.././data/bcf2.npt',status='unknown')

      write(1,104)iss,ibcfcb,hdry,iwdflg,wetfct,iwetit,ihdwet
104   format(2i10,f10.0,i10,f10.2,2i10)

      write(1,'(40i2)')(laycon(i), i=1,nlay)
      write(1,'(i10,f10.1)')locats,constnts
      write(1,'(i10,f10.1,a20,i10)')locatr,cnstntr,fmtinr,iprnr
      write(1,fmtinr)(delr(i),i=1,ncol)
      write(1,'(i10,f10.1,a20,i10)')locatc,cnstntc,fmtinc,iprnc
      write(1,fmtinc)(delc(j),j=1,nrow)

      do l=1,nlay

      write(1,'(i10,f10.1,a20,i10)')locat1,cnstnt1,fmtin1,iprn1
        do j=1,nrow
          write(1,fmtin1)(sf1(i,j,l), i=1,ncol)
        end do

        if(laycon(l).eq.0.or.laycon(l).eq.2)then

      write(1,'(i10,f10.1,a20,i10)')locata,cnstnta,fmtina,iprna
        do j=1,nrow
          write(1,fmtina)(tran(i,j,l), i=1,ncol)
        end do
      end if

        if(laycon(l).eq.1.or.laycon(l).eq.3)then

      write(1,'(i10,f10.1,a20,i10)')locath,cnstnth,fmtinh,iprnh
        do j=1,nrow
          write(1,fmtinh)(hy(i,j,l), i=1,ncol)
        end do
      end if

        if(laycon(l).eq.1.or.laycon(l).eq.3)then
          cnstntb=bot(l)
          write(1,'(i10,f10.1)')locatb,cnstntb
      end if

        if(l.ne.nlay)then

      write(1,'(i10,f10.1,a20,i10)')locatv,cnstntv,fmtinv,iprnv
        do j=1,nrow
          write(1,fmtinv)(vcont(i,j,l), i=1,ncol)
        end do
      end if

        if(laycon(l).eq.3.or.laycon(l).eq.2)then

      write(1,'(i10,f10.1,a20,i10)')locat2,cnstnt2,fmtin2,iprn2
        do j=1,nrow
          write(1,fmtin2)(sf2(i,j,l), i=1,ncol)
```

```fortran
          end do
        end if

        if(laycon(l).eq.3.or.laycon(l).eq.2)then
           cnstntt=top(l)
           write(1,'(i10,f10.1)')locatt,cnstntt
        end if

      if(laycon(l).eq.3.or.laycon(l).eq.1.and.iwdflg.ne.0)then

      write(1,'(i10,f10.1,a20,i10)')locatw,cnstntw,fmtinw,iprnw
          do j=1,nrow
             write(1,fmtinw)(thresh(i,j,l),i=1,ncol)
           end do
         end if
       end do
       close(1)
      -
       open(33,file='../../data/balance.npt',status='unknown')
       do l=1,nlay
         if(l.ne.nlay)then

      write(33,'(i10,f10.1,a20,i10)')locatv,cnstntv,fmtinv,iprnv
          do j=1,nrow
             write(33,fmtinv)(vcont(i,j,l), i=1,ncol)
           end do
         end if
       end do
       close(33)

       return
       end

c **************************************************
c ************************
c    creates output control input
       subroutine output
       parameter(ix=60,jy=50,lay=40,hgu=8)
       integer hdpr,ddpr,hdsv,ddsv
       common /time/nper,nstp(80)
       common /files/iunit(24)

c    ihedfm - code for the format which heads will be
c written

       ihedfm=0

c    iddnfm - code for the format which drawdowns will be
c printed

       iddnfm=0

c    ihedun - unit # to which heads will be written if
c    they are saved on disk

       ihedun=0

c    iddnun - unit # to which drawdowns will be written if
c    they are saved on disk

       iddnun=0

c    incode - is the head/drawdown output code (=0, all
c layers
c    treated the same)
```

```fortran
       incode=0
c    ihddfl- is a head and drawdown output flag

       ihddfl=1

c    ibudfl - budget print flag (if.ne.0 will be printed)

       ibudfl=1

c    icbcfl - cell-by-cell flow term flag

       icbcfl=1

c    hdpr - output flag for head printout

       hdpr=0

c    ddpr - output flag for drawdown printout

       ddpr=0

c    hdsv  - output flag for head save

       hdsv=0

c    ddsv - output flag for drawdown save

       ddsv=0

       open(51,file='../../data/oc.npt',status='unknown')

c    writing to output control input file

       write(51,'(4i10)')ihedfm,iddnfm,ihedun,iddnun
       do ip=1,nper
         do i=1,nstp(ip)
           write(51,'(4i10)')incode,ihddfl,ibudfl,icbcfl
           write(51,'(4i10)')hdpr,ddpr,hdsv,ddsv
         end do
       end do

       close(51)

       return
       end

c **************************************************
c ************************
c    creates pcg2 input
       subroutine pcg2

c    mxiter- maximum  outer iterations  (=1 for linear
c problems,
c    >1 for nonlinear problems)

       mxiter=90

c    iter1 - is the maximum number of inner iterations

       iter1=15

c    npcond - flag used to select the matrix preconditioning
c method

       npcond=1
```

322

```
c   hclose - is the head change criterion for convergence,
in units of length
    hclose=0.05

c   rclose residual criterion for convergence, in unist of
cubic
c   length per time

    rclose=0.05

c   relaxation parameter used with npcond=1

    relax=1.0

c   used when npcond=2

    nbpol=2

c   iprpcg- printout interval for pcb

    iprpcg=0

c   multpcg - is a flag which controls printing from solver
c   (=0, everthing is printed)

    multpcg=0

c   ipcgcd - flag which is used when npcond=1

    ipcgcd=0

    open(51,file='../../data/pcg2.npt',status='unknown')

c   writing to pcg2 input file

    write(51,'(3i10)')mxiter,iter1,npcond

write(51,231)hclose,rclose,relax,nbpol,iprpcg,multpcg,ipc
gcd
 231  format(3f10.3,4i10)

    close(51)
    end

c **************************************************
c **************************
c   creates recharge input file
    subroutine recharge
c   conv is to convert in/yr to ft/day.

parameter(ix=60,jy=50,lay=40,conv1=0.00022831,conv=
0.0)
    integer type
    dimension xd(ix),yd(jy),zd(lay)
    dimension locat(80),cnstnt(80),recha(ix,jy,50)
    real
icol(5000,50),jrow(5000,50),klay(5000,50),itmp(50),
    &   itype(5000,50),head,cond,elev
    character*20 fmtin

    common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy,lay)
    common /time/nper,nstp(80)
    common /files/iunit(24)
    common /rivertype/ rtype(ix,jy,lay)

c   defining variables

    nlay=il-1
    nrow=ir-1
    ncol=ic-1

c   nrchop - recharge option code (=3), recharge is
applied to highest
c   active cell in each vertical column

    nrchop=3

c   irchcb - is a flag and unit number

c   irchcb=50

    irchcb=0

c   inrech - recharge read flag

    inrech=0

c   inirch - only need if nrchop=2

    inirch=0

c   read in recharge values

    open(34,file='../../../0d/data/sjlin.npt',status='old')
    do i=1,4
      read(34,*)
    end do
    read(34,*)rech1
    read(34,*)rech2
    read(34,*)rech3
    read(34,*)rech4
    read(34,*)rech5
    read(34,*)rech6
    close(34)

c   initializing recharge array

    do it=1,nper
      do i=1,ncol
        do j=1,nrow
          recha(i,j,it)=0.0
        end do
      end do
    end do

c   assigning appropriate recharge to each subarea

    do it=1,nper
      if(it.le.32)fact=1.0
      if(it.eq.33.or.it.eq.41)fact=rech2/rech1
      if(it.eq.42.or.it.eq.43)fact=rech3/rech1
      if(it.eq.44.or.it.eq.45)fact=rech4/rech1
      if(it.eq.46.or.it.eq.47)fact=rech5/rech1
      if(it.ge.48)fact=rech6/rech1

      do i=1,ncol
        do j=1,nrow
          do k=1,nlay
            if(type(i,j,k).eq.3)then
c Assign 8 inches recharge to silt
              recha(i,j,it)=rech1*conv1*0.8
              goto 45
            endif
```

```fortran
        if(type(i,j,k).eq.21)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.43)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.22)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.44)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.23)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.45)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.24)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.47)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.25)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.46)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.26)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.46)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.27)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.46)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.28)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.47)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.29)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.45)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.30)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.45)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.31)then
          recha(i,j,it)=rech1*conv*fact

    if(it.gt.45)recha(i,j,it)=rech1*conv*(rech6/rech1)
        goto 45
        endif
        if(type(i,j,k).eq.40)then

          recha(i,j,it)=rech1*conv*0.6
        goto 45
        endif
        end do
 45     continue
        end do
      end do
    end do
c    goto 111
c    read in river cells to make zero recharge in them

    open(9,file='../../data/riv3d.npt',status='old')

    read(9,'(2i10)')mxrivr,irivcb
    do it=1,nper
      read(9,'(i10)')itmp(it)
      if(itmp(it).lt.0)go to 26
      do im=1,itmp(it)
        read(9,105)klay(im,it),jrow(im,it),
    .        icol(im,it),head,cond,elev,
    .        itype(im,it)
      end do
 26   end do
    close(9)
 105  format(3i10,f10.2,g10.2,f10.0,i10)

c    make recharge zero in river cells

    do it=1,nper
      do im=1,itmp(it)
        if(itype(im,it).ge.1.and.itype(im,it).le.10)then
          recha(icol(im,it),jrow(im,it),klay(im,it))=0.0
        endif
      end do
    end do


c111  continue
    do i=1,nper
      cnstnt(i)=1.0
    end do
    fmtin='(10g11.4)'
    iprn=0
    do i=1,nper
      locat(i)=iunit(8)
    end do

    open(1,file='../../data/rch.npt',status='unknown')

    write(1,'(2i10)')nrchop,irchcb

    do it=1,nper
      write(1,'(2i10)')inrech,inirch

write(1,'(i10,f10.1,a20,i10)')locat(it),cnstnt(it),fmtin,iprn
      do j=1,nrow
        write(1,fmtin)(recha(i,j,it), i=1,ncol)
      end do
    end do
    close(1)


    return
    end
```

324

```
*************************************************
***********************

c    creates drain input file
     subroutine drain

     parameter(ix=60,jy=50,lay=40)
     integer type
     dimension xd(ix),yd(jy),zd(lay),ncnt(50)
     dimension
nldrn(ix*jy,50),nrdrn(ix*jy,50),ncdrn(ix*jy,50)
     dimension elev(ix*jy,50),cond(ix*jy,50)

     common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy,lay)
     common /time/nper,nstp(80)
     common /files/iunit(24)

c    defining variables

     nlay=il-1
     nrow=ir-1
     ncol=ic-1

c    idrncb - is a flag and unit number on which cell-by-
cell flow terms will
c    be recorded

     idrncb=0

     do it=1,nper
       ncnt(it)=0
     end do

     do it=1,nper
       do j=1,nrow
         do i=1,ncol
           do l=1,nlay

c    adding drains at top of landfill

                if(abs(type(i,j,l)).eq.2.or.type(i,j,l).ge.10)then
                  ncnt(it)=ncnt(it)+1
                  nrdrn(ncnt(it),it)=j
                  ncdrn(ncnt(it),it)=i
                  nldrn(ncnt(it),it)=l
                  elev(ncnt(it),it)=zd(l)
                  cond(ncnt(it),it)=1000.0
                  go to 123
                end if

c    adding drains where top of silt layer is at ground
surface

                if(abs(type(i,j,l)).eq.3.and.zd(l).ge.10.0)then
                  ncnt(it)=ncnt(it)+1
                  nrdrn(ncnt(it),it)=j
                  ncdrn(ncnt(it),it)=i
                  nldrn(ncnt(it),it)=l
                   elev(ncnt(it),it)=zd(l)
                    cond(ncnt(it),it)=1000.0
                    go to 123
                  end if
                end do
123         end do
          end do
        end do

c    mxdrn - max number of drain cells active at one time
```

```
     mxdrn=ncol*nrow
     do itt=1,nper
       mxdrn=max(ncnt(itt),mxdrn)
     end do

     open(1,file='../../data/drn.npt',status='unknown')

     write(1,'(2i10)')mxdrn,idrncb

     do itt=1,nper
       write(1,'(i10)')ncnt(itt)
       do i=1,ncnt(itt)
         write(1,'(3i10,f10.1,f10.3)')nldrn(i,itt),nrdrn(i,itt),
.            ncdrn(i,itt),elev(i,itt),cond(i,itt)
       end do
     end do

     close(1)

     return
     end
*************************************************
***********************
c    creates input file delineating areas of landfill added
     subroutine addcell
     parameter(ix=60,jy=50,lay=40)
     integer type
     dimension
xd(ix),yd(jy),zd(lay),time(50),itype(10000,50)
     dimension
icol(10000,50),jrow(10000,50),klay(10000,50),
.       head(10000,50),cond(10000,50),elev(10000,50)
     integer rtype(ix,jy,lay)
     integer acell(ix,jy,lay),izcnt(50),itmp(50)

     common /info/xd,yd,zd,xm,ym,il,ic,ir,type(ix,jy,lay)
     common /files/iunit(24)
     common /landf/ acell
     common /time/nper,nstp(80)
     common /rivertype/ rtype
     nper=48

c    defining variables

     nlay=il-1
     nrow=ir-1
     ncol=ic-1

c    nzone- # of subarea groups added

     nzone=5

c    time - time when subarea group added

c    subarea 2 added

     time(1)=2.0*365.0

c    subarea 1 added

     time(2)=3.0*365.0

c    subarea 3&PLC added

     time(3)=5.0*365.0
```

```
c   subarea 4 added and dike added

    time(4)=38.0*365.0

c   subarea 5 added

    time(5)=40.0*365.0

c   intializing variable 'izcnt' which counts the number of
c   cells added in each subarea

    do ii=1,6
      izcnt(ii)=0
    end do

    do l=1,nlay
      do i=1,ncol
        do j=1,nrow
          acell(i,j,l)=0.
        end do
      end do
    end do

    do l=1,nlay
      do i=1,ncol
        do j=1,nrow

          if(type(i,j,l).eq.22.or.type(i,j,l).eq.27.
     .       or.type(i,j,l).eq.28)then
            izone=1
            acell(i,j,l)=izone
            izcnt(izone)=izcnt(izone)+1
          end if
          if(type(i,j,l).eq.21)then
            izone=2
            acell(i,j,l)=izone
            izcnt(izone)=izcnt(izone)+1
          end if
          if(type(i,j,l).eq.23.or.type(i,j,l).eq.29.or.
     .       type(i,j,l).eq.30)then
            izone=3
            acell(i,j,l)=izone
            izcnt(izone)=izcnt(izone)+1
          end if
          if(type(i,j,l).eq.24.or.type(i,j,l).eq.40.or.
     .       type(i,j,l).eq.31)then
            izone=4
            acell(i,j,l)=izone
            izcnt(izone)=izcnt(izone)+1
          end if
          if(type(i,j,l).eq.25)then
            izone=5
            acell(i,j,l)=izone
            izcnt(izone)=izcnt(izone)+1
          end if
        end do
      end do
    end do


    open(1,file='../../data/addcell.npt',status='unknown')

    write(1,'(i10)')nzone

    do iz=1,nzone
      write(1,'(i10,f10.1)')izcnt(iz),time(iz)
      do l=1,nlay
```

```
        do i=1,ncol
          do j=1,nrow
            if(abs(acell(i,j,l)).eq.iz)then
              thr=-(zd(l)-zd(l+1))/1.0
              write(1,'(4i4,f10.1,i5)')i,j,l,acell(i,j,l),thr,
     .             type(i,j,l)
            end if
          end do
        end do
      end do
    end do

    close(1)

c   read river input file to determine river cells which are
not
c   related to old landfill lake

    open(9,file='../../data/riv3d.npt',status='old')

    read(9,'(2i10)')mxrivr,irivcb
    do it=1,nper
      read(9,'(i10)')itmp(it)
      if(itmp(it).lt.0)go to 26
      do im=1,itmp(it)
        read(9,105)klay(im,it),jrow(im,it),
     .       icol(im,it),head(im,it),cond(im,it),elev(im,it),
     .       itype(im,it)
        if(it.eq.1)head(im,it)=10.00
        rtype(icol(im,it),jrow(im,it),klay(im,it))=itype(im,it)
      end do
26  end do
    close(9)

c   adding river cells corresponding to old lake

    xleak=0.1
    do j=1,nrow
      do i=1,ncol

c   for stress periods 1-2, subareas 1,2,3,4,5 and dike are
lake

        do it=1,2
          do l=2,nlay
            if(acell(i,j,l-1).ne.0)then
              if(type(i,j,l).eq.3.and.zd(l).le.10.0)then
                itmp(it)=itmp(it)+1
                icnt=itmp(it)
                icol(icnt,it)=i
                jrow(icnt,it)=j
                klay(icnt,it)=l
                head(icnt,it)=9.99
                area=(yd(j+1)-yd(j))*(xd(i+1)-xd(i))
                cond(icnt,it)=xleak*area
                elev(icnt,it)=0.0
                go to 32
              end if
            end if
          end do
        end do
32      continue
      end do
    end do

c   for third stress period, subareas 1,3,4,5 and dike are
lake

      it=3
```

326

```
          do l=2,nlay                                        if(acell(i,j,l-1).ge.4)then
            if(acell(i,j,l-1).ge.2)then                        if(type(i,j,l).eq.3.and.zd(l).le.10.0)then
              if(type(i,j,l).eq.3.and.zd(l).le.10.0)then         itmp(it)=itmp(it)+1
                itmp(it)=itmp(it)+1                              icnt=itmp(it)
                icnt=itmp(it)                                    icol(icnt,it)=i
                icol(icnt,it)=i                                  jrow(icnt,it)=j
                jrow(icnt,it)=j                                  klay(icnt,it)=l
                klay(icnt,it)=l                                  head(icnt,it)=9.99
                head(icnt,it)=9.99                               area=(yd(j+1)-yd(j))*(xd(i+1)-xd(i))
                area=(yd(j+1)-yd(j))*(xd(i+1)-xd(i))              cond(icnt,it)=xleak*area
                 cond(icnt,it)=xleak*area                         elev(icnt,it)=0.0
                 elev(icnt,it)=0.0                                go to 36
                 go to 33                                       end if
              end if                                           end if
            end if                                           end do
          end do                                   36      continue
  33      continue                                         end do

c    for  stress periods 4-5, subareas 3,4,5 and dike are    c    for stress periods 33-34, subareas 4,5 and dike are
lake                                                     lake

          do it=4,5                                          do it=33,34
            do l=2,nlay                                        do l=2,nlay
              if(acell(i,j,l-1).ge.3)then                        if(acell(i,j,l-1).ge.4)then
                if(type(i,j,l).eq.3.and.zd(l).le.10.0)then         if(type(i,j,l).eq.3.and.zd(l).le.10.0)then
                  itmp(it)=itmp(it)+1                              itmp(it)=itmp(it)+1
                  icnt=itmp(it)                                    icnt=itmp(it)
                  icol(icnt,it)=i                                  icol(icnt,it)=i
                  jrow(icnt,it)=j                                  jrow(icnt,it)=j
                  klay(icnt,it)=l                                  klay(icnt,it)=l
                  head(icnt,it)=9.99                               head(icnt,it)=9.99
                  area=(yd(j+1)-yd(j))*(xd(i+1)-xd(i))             area=(yd(j+1)-yd(j))*(xd(i+1)-xd(i))
                   cond(icnt,it)=xleak*area                         cond(icnt,it)=xleak*area
                   elev(icnt,it)=0.0                                elev(icnt,it)=0.0
                   go to 34                                         go to 37
                end if                                           end if
              end if                                           end if
            end do                                           end do
  34      continue                                   37      continue
          end do                                           end do

c    for  stress periods 6-20, subareas 4,5 and dike are lake   c    for stress periods 35-36, subareas 4,5 and dike are
                                                        lake

          do it=6,20                                         do it=35,36
            do l=2,nlay                                        do l=2,nlay
              if(acell(i,j,l-1).ge.4)then                        if(acell(i,j,l-1).ge.4)then
                if(type(i,j,l).eq.3.and.zd(l).le.10.0)then         if(type(i,j,l).eq.3.and.zd(l).le.10.0)then
                  itmp(it)=itmp(it)+1                              itmp(it)=itmp(it)+1
                  icnt=itmp(it)                                    icnt=itmp(it)
                  icol(icnt,it)=i                                  icol(icnt,it)=i
                  jrow(icnt,it)=j                                  jrow(icnt,it)=j
                  klay(icnt,it)=l                                  klay(icnt,it)=l
                  head(icnt,it)=9.99                               head(icnt,it)=9.99
                  area=(yd(j+1)-yd(j))*(xd(i+1)-xd(i))             area=(yd(j+1)-yd(j))*(xd(i+1)-xd(i))
                   cond(icnt,it)=xleak*area                         cond(icnt,it)=xleak*area
                   elev(icnt,it)=0.0                                elev(icnt,it)=0.0
                   go to 35                                         go to 38
                end if                                           end if
              end if                                           end if
            end do                                           end do
  35      continue                                   38      continue
          end do                                           end do
c    for stress periods 21-32, subareas 4,5 and dike are
lake                                                     c    for stress periods 37-38, subareas 4,5 and dike are
                                                        lake
          do it=21,32
            do l=2,nlay
```

327

```
        do it=37,38
          do l=2,nlay
            if(acell(i,j,l-1).ge.4)then
              if(type(i,j,l).eq.3.and.zd(l).le.10.0)then
                itmp(it)=itmp(it)+1
                icnt=itmp(it)
                icol(icnt,it)=i
                jrow(icnt,it)=j
                klay(icnt,it)=l
                head(icnt,it)=9.99
                area=(yd(j+1)-yd(j))*(xd(i+1)-xd(i))
                 cond(icnt,it)=xleak*area
                 elev(icnt,it)=0.0
                go to 78
              end if
            end if
          end do
 78       continue
        end do
      end do
    end do

c    writing river input file which includes river cells
defined by
c    old landfill lake

    open(19,file='././/data/riv2.npt',status='unknown')

    write(19,'(2i10)')mxrivr,irivcb
    do it=1,nper
      write(19,'(i10)')itmp(it)
      if(itmp(it).lt.0)go to 46
      do im=1,itmp(it)
        write(19,100)klay(im,it),jrow(im,it),
    .       icol(im,it),head(im,it),cond(im,it),elev(im,it),
    .       itype(im,it),
    .       acell(icol(im,it),jrow(im,it),klay(im,it)-1)
      end do
 46   end do
    close(19)

100  format(3i10,f10.2,g10.2,f10.0,2i5)
105  format(3i10,f10.2,g10.2,f10.0,i10)
    return
    end
```

## Modifications to MODFLOW code:

In addition to the modifications added in Appendix B, the following was added to the BCF2.FOR subprogram, inserted at line 272. Subroutine BCF2TOM reads results from the two-dimensional mounding model and updates the constant head boundary at the landfill for every time step to reflect the rising mound in the landfill and the growth of the landfill area over time. Subroutines BCF2GRAV, BCF2BYB, and BCF2POST resest the constant head cells along the sand, gravel, and Troutdale units for the time prior to 1982 (BCF2GRAV), from 1982-1997 (BCF2BYB), and after 1997 (BCF2POST). These subroutines were written and added to the source code by Thomas Lowry.

```
      SUBROUTINE bcf2tom(bot,wetdry,ncol,nrow,nlay,ibound,hnew,clfhead,
     &      hy,cv,top,ibuff,delc,delr)
c-----Version 1 29Jun1995 - Tom Lowry
c     ***********************************************************************
c     Change landfill ibound values and head values to reflect rising
c     head in the landfill
c     ***********************************************************************
c     Specifications:

      double precision hnew
      parameter(thresh=-2.5)

      dimension bot(ncol,nrow,nlay),wetdry(ncol,nrow,nlay),
     &      ibound(ncol,nrow,nlay),hnew(ncol,nrow,nlay),
     &      type(60,40,40),delc(nrow),clfhead(51,39),
     &      hy(ncol,nrow,nlay),cv(ncol,nrow,nlay),
     &      top(ncol,nrow,nlay),delr(ncol),ibuff(51,39,39)
      real vht,vhb
c
c     Read anisotropy from input file
c
      open(11,file='./.././0d/data/sjlin.npt',status='old')
      read(11,*)
      read(11,*)vvanis
      close(11)
      vanis=1/vvanis
c
c     Read gtype file and head level in landfill
c
      open(199,file='././.././commondata/gtype.out',status='old')

 233  read(199,*,end=333)ic,jr,kl,itemp
      type(ic,jr,kl)=itemp
      go to 233
 333  continue
      close(199)
c
c     Scan cells in landfill and reset head value for any landfill
c     cells that have become wet
c     Change hnew in landfill cell to reflect 'wetting' of constant head cells
c
      do id=1,ncol
        do jd=1,nrow
          do kd=1,nlay
            if(ibuff(id,jd,kd).eq.1)then
              if(type(id,jd,kd).gt.20.and.
     &            type(id,jd,kd).le.30.and.
     &            type(id,jd,kd+1).eq.3)then
c
c     Check cells above current wet cells to see if they're wet
c
                do kkd=kd,2,-1
                  if(type(id,jd,kkd).gt.20.and.
     &                type(id,jd,kkd).le.30)then
                    if(bot(id,jd,kkd-1).le.clfhead(id,jd))then
                      ibound(id,jd,kkd)=-93
                      hnew(id,jd,kkd)=clfhead(id,jd)
                      wetdry(id,jd,kkd)=thresh
                      vht=abs(top(id,jd,kkd)-
     &                    top(id,jd,kkd+1))
                      vhb=abs(bot(id,jd,kkd)-
     &                    bot(id,jd,kkd+1))
                      cv(ii,jj,ll)=vanis*delc(jj)*delr(ii)*
     &                    ((2*hy(ii,jj,ll)*hy(ii,jj,ll+1))/
     &                    (vht*hy(ii,jj,ll+1)+vhb*
     &                    hy(ii,jj,ll)))
                    else
```

330

```
                cv(id,jd,kkd)=0.0
                ibound(id,jd,kkd)=0
                wetdry(id,jd,kkd)=thresh
              endif
            endif
          end do
        goto 887
      endif
    endif
   end do
887     continue
  end do
  end do
c
c   Make sure all cells below water table to bottom of troutdale are active
c

   do idd=2,ncol-1
     do jdd=2,nrow-1
       do kdd=1,nlay
         if(ibound(idd,jdd,kdd).ne.0)then
           do kkd=kdd,nlay

c   Skip constant head cells

             if(ibound(idd,jdd,kkd).lt.0.and.type(idd,jdd,kkd)
   &           .ge.20.and.type(idd,jkk,kkd).le.30)goto 130
             if(type(idd,jdd,kkd).gt.6)goto 130

c   Assign active ibound value

             ibound(idd,jdd,kkd)=97
130            if(kkd.eq.nlay)goto 140
           end do
         endif
       end do
140     continue
     end do
   end do
   return
   end


   subroutine bcf2grav(ibound,hnew,hold,ncol,nrow,nlay)
c-----Version 2 21May,1997 - Tom Lowry
c   ****************************************************************
c   Subroutine to reset constant values for sides of sand, gravel, and
c   troutdale for times before 1982
c   ****************************************************************
c
c   Specifications:
c   ------------------------------------------------------------------
   double precision hnew
c
   dimension ibound(ncol,nrow,nlay),hnew(ncol,nrow,nlay),
   &    hold(ncol,nrow,nlay)
   dimension grav(200,150),trout(200,150),sand(200,150)
   dimension silt(200,150)
   integer type(200,150,60)


c
c   Read gtype file and head level in landfill
c
   open(199,file='../../../commondata/gtype.out',status='old')

233 read(199,*,end=333)ic,jr,kl,itemp
   type(ic,jr,kl)=itemp
   go to 233
```

331

```
  333  continue
      close(199)
      open(76,file='././data/ssbpre82.npt',status='unknown')
c
c     Read heads in silt, sand, gravel, and troutdale to reinitialize
c     constant head values
c
      do j=1,nrow
        read(76,'(51f9.3)')(silt(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(sand(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(grav(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(trout(i,j),i=1,ncol)
      end do
c
c     Reset ibound values in silt so constant head cells don't go dry
c
      do i=1,51,50
        do j=1,39,38
          elev=47.5
          do k=10,20
            if(silt(i,j).lt.elev)then
              elev=elev-5.0
            else
              do kk=k+1,nlay
                if(type(i,j,kk).eq.3)ibound(i,j,kk)=-kk
              end do
              goto 1001
            endif
          end do
 1001     continue
        end do
      end do


c
c     Reinitialize constant head values
c
      do i=1,ncol
        do j=1,nrow
          do l=1,nlay
            if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.3)then
              hnew(i,j,l)=silt(i,j)
            end if
            if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.4)then
              hnew(i,j,l)=sand(i,j)
            end if
            if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.5)then
              hnew(i,j,l)=grav(i,j)
            end if
            if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.6)then
              hnew(i,j,l)=trout(i,j)
            end if
            if(ibound(i,j,l).ge.0.and.type(i,j,l).eq.7)then
              ibound(i,j,l)=0
            end if
          end do
        end do
      end do
      return
      end

      subroutine bcf2byb(ibound,hnew,hold,ncol,nrow,nlay)
```

332

```
c-----Version 2 21May,1997 - Tom Lowry
c      *********************************************************************
c     Subroutine to reset constant values for sides of sand, gravel, and
c     troutdale gravel for time between 1982 and 1997
c      *********************************************************************
c
c     Specifications:
c     ----------------------------------------------------------------
      double precision hnew
c
      dimension ibound(ncol,nrow,nlay),hnew(ncol,nrow,nlay),
     &    hold(ncol,nrow,nlay)
      dimension grav(200,150),trout(200,150),sand(200,150)
      dimension silt(200,150)
      integer type(200,150,60)


c
c     Read gtype file and head level in landfill
c
      open(199,file='./././../commondata/gtype.out',status='old')

 233  read(199,*,end=333)ic,jr,kl,itemp
      type(ic,jr,kl)=itemp
      go to 233
 333  continue
      close(199)
      open(76,file='././../data/ssb82_95.npt',status='unknown')
c
c     Read heads in silt, sand, gravel, and troutdale to reinitialize
c     constant head values
c
      do j=1,nrow
        read(76,'(51f9.3)')(silt(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(sand(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(grav(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(trout(i,j),i=1,ncol)
      end do
c
c     Reset ibound values in silt so constant head cells don't go dry
c
      do i=1,51,50
        do j=1,39,38
          elev=47.5
          do k=10,20
            if(silt(i,j).lt.elev)then
              elev=elev-5.0
            else
              do kk=k+1,nlay
               if(type(i,j,kk).eq.3)ibound(i,j,kk)=-kk
              end do
              goto 1001
            endif
          end do
 1001     continue
        end do
      end do
c
c     Reinitialize constant head values
c
      do i=1,ncol
        do j=1,nrow
```

333

```fortran
          do l=1,nlay
             if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.3)then
               hnew(i,j,l)=silt(i,j)
             end if
             if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.4)then
               hnew(i,j,l)=sand(i,j)
             end if
             if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.5)then
               hnew(i,j,l)=grav(i,j)
             end if
             if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.6)then
               hnew(i,j,l)=trout(i,j)
             end if
             if(ibound(i,j,l).ge.0.and.type(i,j,l).eq.7)then
                ibound(i,j,l)=0
             end if
          end do
        end do
      end do
      return
      end

      subroutine bcf2post(ibound,hnew,hold,ncol,nrow,nlay)
c-----Version 2 21May,1997 - Tom Lowry
c     ********************************************************************
c     Subroutine to reset constant values for sides of sand, gravel and
c     troutdale after 1997
c     ********************************************************************
c
c     Specifications:
c     -----------------------------------------------------------------
      double precision hnew
c
      dimension ibound(ncol,nrow,nlay),hnew(ncol,nrow,nlay),
     &     hold(ncol,nrow,nlay)
      dimension grav(200,150),trout(200,150),sand(200,150)
      dimension silt(200,150)
      integer type(200,150,60)


c
c     Read gtype file and head level in landfill
c
      open(199,file='./././commondata/gtype.out',status='old')

 233  read(199,*,end=333)ic,jr,kl,itemp
      type(ic,jr,kl)=itemp
      go to 233
 333  continue
      close(199)
      open(76,file='././data/ssbpost95.npt',status='unknown')
c
c     Read heads in silt, sand, gravel, and troutdale to reinitialize
c     constant head values
c
      do j=1,nrow
        read(76,'(51f9.3)')(silt(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(sand(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(grav(i,j),i=1,ncol)
      end do
      do j=1,nrow
        read(76,'(51f9.3)')(trout(i,j),i=1,ncol)
      end do
c
```

334

```
c      Reset ibound values in silt so constant head cells don't go dry
c
       do i=1,51,50
         do j=1,39,38
          elev=47.5
          do k=10,20
            if(silt(i,j).lt.elev)then
              elev=elev-5.0
            else
              do kk=k+1,nlay
               if(type(i,j,kk).eq.3)ibound(i,j,kk)=-kk
              end do
               goto 1001
             endif
           end do
 1001      continue
         end do
       end do
c
c      Reinitialize constant head values
c
       do i=1,ncol
         do j=1,nrow
           do l=1,nlay
             if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.3)then
               hnew(i,j,l)=silt(i,j)
             end if
             if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.4)then
               hnew(i,j,l)=sand(i,j)
             end if
             if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.5)then
               hnew(i,j,l)=grav(i,j)
             end if
             if(ibound(i,j,l).lt.0.and.type(i,j,l).eq.6)then
               hnew(i,j,l)=trout(i,j)
             end if
             if(ibound(i,j,l).ge.0.and.type(i,j,l).eq.7)then
               ibound(i,j,l)=0
             end if
           end do
         end do
       end do
       return
       end
```

 

The following has been added to the main MODFLOW program,

MFLOWEM.FOR to call the subroutines in BCF2.FOR as outlined above. The first

part reads the geology and the head file from the two-dimensional mounding model

and sends that information to the BCF2TOM subroutine in the BCF2.FOR

subprogram. This was written and added to the source code at line 107 by Thomas

Lowry.

```
c
```

```fortran
c Read in grid types and set ibuff=1 where landfill is active
c
      open(199,file='..\.\.\.\.\commond\gtype.out',status='old')

233   read(199,*,end=333)ic,jr,kl,itemp
      type(ic,jr,kl)=itemp
      go to 233
333   continue
      close(199)
      do i=1,51
        do j=1,39
          do k=1,39
            if(type(i,j,k).eq.26)ibuff(i,j,k)=1
          end do
        end do
      end do
c
c     Read landfill constant head
c     .
      if(kper.gt.69)goto 557
      if(kper.eq.1)then
        do i=1,51
          do j=1,39
            clfhead(i,j)=9.995
          end do
        end do
        goto 557
      endif
      open(92,file='..\.\.\.\2d\output\head.dat',status='old')
      read(92,'(30f6.1)')((clftemp(i,j),i=153,1,-1),j=117,1,-1)
      ii=0
      jj=0
c
c     Match small 2d grid (153 x 117) to larger 3d grid (51 x 39)
c
      do i=153,1,-3
        do j=117,1,-3
          ii=i/3
          jj=j/3
          clfhead(ii,jj)=clftemp(i,j)
        end do
      end do
557   if(kper.ge.70)then
        do i=1,51
          do j=1,39
            clfhead(i,j)=9.995
          end do
        end do
        goto 557
      endif
```

The following is added to the MFLOWEM.FOR program to call the added

subroutines in the BCF2.FOR subprogram which update the constant head boundaries

in the sand, gravel, and Troutdale units.

```fortran
c Calling subroutine to change ibound and head values in landfill as
c needed to compensate for varying head - Tom
c
```

```
      IF(IUNIT(1).GT.0) CALL bcf2tom(X(LCBOT),X(LCWETD),NCOL,NROW,NLAY,
     &    X(LCIBOU),X(LCHNEW),clfhead,x(lchy),x(lccv),x(lctop),ibuff,
     &    x(lcdelc),x(lcdelr))
c
c Calling subroutines to reset gravel values for constant head - Tom
c
     if(kper.eq.2.or.kper.eq.47)then
       if(iunit(1).gt.0) call bcf2grav(X(LCIBOU),X(LCHNEW),X(LCHOLD),
     &      ncol,nrow,nlay)
     endif
     if(kper.eq.32)then
       if(iunit(1).gt.0) call bcf2byb(X(LCIBOU),X(LCHNEW),X(LCHOLD),
     &      ncol,nrow,nlay)
     endif
     if(kper.eq.47)then
       if(iunit(1).gt.0) call bcf2post(X(LCIBOU),X(LCHNEW),X(LCHOLD),
     &      ncol,nrow,nlay)
     endif
```

337

**Appendix D  - Transport Model Source Code and Documentation**

The transport model is designed to apply particle tracking, dispersion, and source/sink mixing to the three-dimensional velocity field developed in the three-dimensional flow model.  Input into the model is via a binary file produced by an addition to the MODFLOW code of the three-dimensional flow model.  The addition comes as part of the MT3D package and works by inserting it into the MODFLOW source code at the appropriate point.  Since MT3D and the MODFLOW addition are part of a commercial package, they will not be reviewed here.

In addition to the binary output file from MODFLOW, MT3D requires input parameters which are unique to the transport model.  To accomplish this, a pre-processor (MTINPUT.FOR) was written using FORTRAN that reads 5 user defined files and combines the information into a form which can then be used by MT3D.  The five user defined files set parameters for the basic model layout (BTN.OUT), the advective transport (ADV.OUT), the dispersive transport (DSP.OUT), the reaction conditions (RCT.OUT), and the source/sink mixing conditions (SSM.OUT).

Output (MT3D.OUT) from the transport model is a formatted grid file that designates a concentration at each nodal point at each output time (output is once for each year of the simulation).  This file is very large (~100 Mb) and because of this it is recommended that it be converted to binary form for use in any 'real-time' visualization programs (such as AVS - Advanced Visualization Systems).  Several post-processing programs were also used to 'split' the output file into various forms

for use in producing contour plots, line plots, etc. Since the post-processing programs

are dependent upon the type of software the user is using to visualize the results and

are simply data manipulation programs, the post-processing programs will not be

reviewed here.
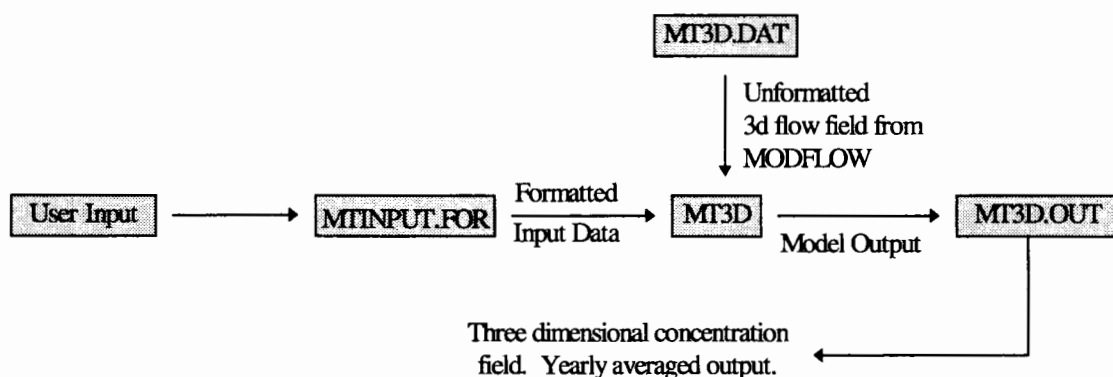


**Figure 93 - Transport model program flow chart. User input includes porosity, dispersivity, and source concentration. MT3D.DAT is an unformatted file produced by the 3d flow model that contains the spatially and temporally varying head distribution which is then converted internally to a three-dimensional velocity distribution for use in the transport calculations.**

As compared to the other upstream models, the program layout for the

transport model is fairly simple. The input files for the base case condition and the

source code for the pre-processor are shown below.

| Program Name | Input Files Needed | Output Files Produced | Description of Output |
|---|---|---|---|
| BTN.OUT<br>ADV.OUT<br>DSP.OUT<br>RCT.OUT<br>SSM.OUT | None | NA | USER DEFINED:<br>Grid layout, timing information, etc.<br>Advection package parameters<br>Dispersion package parameters<br>Reaction package parameters<br>Source/Sink package parameters |
| MTINPUT.FOR | BTN.OUT<br>ADV.OUT<br>DSP.OUT<br>RCT.OUT<br>SSM.OUT | BTN.INP<br>ADV.INP<br>DSP.INP<br>RCT.INP<br>SSM.INP | Grid layout, timing information, etc.<br>Advection package parameters<br>Dispersion package parameters<br>Reaction package parameters<br>Source/Sink package parameters |
| MT3D | BTN.INP<br>ADV.INP<br>DSP.INP<br>RCT.INP<br>SSM.INP<br>MT3D.DAT | MT3D.OUT | Yearly averaged, 3d concentration field. |

**Table 8 - Input files, and output files for the contaminant transport model. The *.OUT files are user defined data files. MTINPUT.FOR is the preprocessor that sorts the data files in such a form than can be used by the commercial package, MT3D.**

## BTN.OUT

```
176                 NOBS
48                  NPER
0                   CINACT
0                   IFMTCN
0                   IFMTNP
0                   IFMTRF
0                   IFMTDP
F                   SAVUCN
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 24 NPRS(Should equal NSTP)
F                   CHKMAS
365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365
365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 365 8760 PERLEN
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 24 NSTP
1                   TSMULT
0                   DT0
9500                MXSTRN
DAY FT LB           TUNIT LUNIT MUNIT
Y                   ADVECTION
Y                   DISPERSION
Y                   SOURCE/SINK
F                   REACTION
100 0 (10F8.2) (20I2) 4      IREAD CNSTNT FMTIN IFMTIN IPRN
0 0 .3 .40 .25 .25 .25 .4 .25 .25 .25 .25 .25 .25 .25 .25 .4 .4 POROSITY
0 0 -1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 ICBUND
0 0  1 0 0 0 0 0 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 0 SCONC
```

BTN.OUT is a user defined input file which sets up the global model parameters such
as timing, porosity, initial concentration, and which packages are active. Definitions
of each variable can be found in the source code for MTINPUT.FOR shown below.

## ADV.OUT

```
2     MIXELM
.9    PERCEL
20000 MXPART
1     ITRACK
.9    WD
.00001 DCEPS
0     NPLANE
0     NPL
10    NPH
2     NPMIN
30    NPMAX
1     SRMULT
1     INTERP
0     NLSINK
8     NPSINK
0.01  DCHMOC
```

ADV.OUT sets up the parameters for the advection package. It includes the setting of
the maximum courant number, interpolation methods, and weighting factors.

Definitions of each variable can be found in the source code for MTINPUT.FOR

shown below.

### DSP.OUT

*0 0 4 1 1 5 5 1 4 4 4 4 4 4 4 4 4 4 4*
*0.1 .01 0*

*Lines above are organized as follows:*
*From left to right on first line:*

*Dispersivity - undefined*
*Dispersivity - water*
*Dispersivity - landfill*
*Dispersivity - silt*
*Dispersivity - sand*
*Dispersivity - gravel*
*Dispersivity - Troutdale Aquifer*
*Dispersivity - unconsolidated*
*Dispersivity - Columbia River Basalt*

*From left to right on second line:*

*Ratio of horizontal transverse dispersivity to the longitudinal dispersivity.*
*Ratio of vertical transverse dispersivity to the longitudinal dispersivity.*
*Diffusion Coefficient*

DSP.OUT sets the parameters for the dispersion package. It sets the dispersivity for

each defined geologic unit, the ratio of the horizontal transverse dispersivity to the

longitudinal dispersivity, the ratio of vertical transverse dispersivity to the longitudinal

dispersivity, and the diffusion coefficient. Definitions of each variable can be found in

the source code for MTINPUT.FOR shown below.

### RCT.OUT

*2 0*
*.3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 .3*
*50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50*
*2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2*
*0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0*
*0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0*

*ISOTHM,IREACT*
*UNDENS,WADENS,LADENS,SIDENS,SADENS,GRDENS,TRDENS,UNDENS,CRDENS - Bulk density for each geologic layer.*
*UNSP1,WASP1,LASP1,SISP1,SASP1,GRSP1,TRSP1,UNSP1,CRSP1 - First sorption constant for each geologic layer.*
*UNSP2,WASP2,LASP2,SISP2,SASP2,GRSP2,TRSP2,UNSP2,CRSP2 - Second sorption constant for each geologic layer.*
*UNRC1,WARC1,LARC1,SIRC1,SARC1,GRRC1,TRRC1,UNRC1,CRRC1 - First rate constant for first order decay.*
*UNRC2,WARC2,LARC2,SIRC2,SARC2,GRRC2,TRRC2,UNRC2,CRRC2 - Second rate constant for first order decay.*

RCT.OUT is used to designate reaction rate parameters for the reaction package. The reaction package is deactivated in the St. Johns Landfill model so this input file is only used as 'dummy' input. Definitions of each variable can be found in the source code for MTINPUT.FOR shown below.

## SSM.OUT

```
F T T F T F
15000
-1      .
0
-1
0
0
0 0 0 0 0


FWEL,FDRN,FRCH,FEVT,FRIV,FGHB
MXSS
INCRCH
CONC1 - CRCH
INCEVT
CONC2 - CEVT
NSS
KSS,ISS,JSS,CSS,ITYPE
```

SSM.OUT sets the parameters for the source/sink package. This includes setting of well conditions, general head boundaries, drains, evapotranspiration (deactivated in this case), and recharge. In addition, it allows for setting the contaminant concentration of any inflow into the system. Definitions of each variable can be found in the source code for MTINPUT.FOR shown below.

## MTINPUT.FOR

```
c    MTINPUT.FOR MT3D Input File Preprocessor
c    St. Johns Landfill Contaminant Transport Model
c    Portland State University - Thomas Lowry

c    Parameter statement to set number of layers (LAY),
c    number of columns
c    (COL), number of rows (ROWS), and number of
c    observation points
c    within the grid (OBS).

     INTEGER
IC,IR,IL,NOBS,NROW,NCOL,NLAY,COL,ROW,LAY,OBS
     INTEGER
NPRS,A,B,F,G,H,TTYPE,UNTYPE,WATYPE,LATYPE,SIT
YPE
     INTEGER
SATYPE,GRTYPE,TRTYPE,UCTYPE,SA1TYPE,SA2TYPE
     INTEGER
SA3TYPE,SA4NTYPE,SA5ETYPE,PLCTYPE,SA5ATYPE
     INTEGER RDTYPE,EDTYPE,SA5WTYPE,SA4STYPE
     REAL LAPOR
     PARAMETER (LAY=39, COL=51, ROW=39,
OBS=176, NPR=50)
     CHARACTER HEADING(2)*80
     CHARACTER TRNOP(4)*2
     CHARACTER FMTIN*20,IFMTIN*20
     CHARACTER*4 CHKMAS, SAVUCN, TUNIT, LUNIT,
MUNIT, ADVECTION
     CHARACTER*4 DISPERSION, SOURCESINK,
REACTION
     CHARACTER*6 WELLNM(OBS),WWELLNM(OBS)
     DIMENSION LAYCON(LAY), DELR(COL),
DELC(ROW), HTOP(COL, ROW)
     DIMENSION DZ(COL, ROW, LAY), PRSITY(COL,
ROW, LAY)
     DIMENSION ICBUND(COL, ROW,
LAY),SCONC(COL, ROW, LAY)
     DIMENSION
IOBS(OBS),JOBS(OBS),KOBS(OBS),ELEV(LAY)
     DIMENSION IIOBS(OBS),JJOBS(OBS),KKOBS(OBS)
     DIMENSION XINT(COL), YINT(ROW), ZINT(LAY),
TIMPRS(40)
     DIMENSION XS(COL), YS(ROW), ZS(LAY),
TYPE(COL, ROW, LAY)
     DIMENSION
XADD(COL),YADD(ROW),WXADD(COL),WYADD(ROW
)
     DIMENSION
WELLX(OBS),WELLY(OBS),WELLZ(OBS),WELLNO(OBS
)
     DIMENSION
DIST(COL),PERLEN(NPR),NSTP(NPR),NPRS(NPR)
     DIMENSION WWELLNO(OBS)
     COMMON
/INFO/TYPE,NLAY,NROW,NCOL,IREAD,CNSTNT
     COMMON /INFO/FMTIN,IPRN,NPER
     NROW=ROW
     NCOL=COL
     NLAY=LAY
     OPEN(10,FILE='BTN.INP',STATUS='UNKNOWN')
     OPEN(20,FILE='well.ind',STATUS='UNKNOWN')
     OPEN(50,FILE='btn.out',STATUS='OLD')
     OPEN(25,FILE='calwell.npt',STATUS='OLD')

OPEN(51,FILE='~/modflow/3d/landdat/griddef.in',STATU
S='OLD')
```

```
OPEN(100,FILE='~/metro/commondata/gtype.out',STATU
S='OLD')

OPEN(101,FILE='PLUME.XCOL',STATUS='UNKNOWN'
)

OPEN(102,FILE='PLUME.YROW',STATUS='UNKNOWN
')

OPEN(103,FILE='PLUME.ZLAY',STATUS='UNKNOWN')

c    Enter headings for output

     HEADING(1)='3-D Contaminant Transport Model'
     HEADING(2)='Test Run '

c    Read in parameters from BTN.OUT.  This is the basic
c    transport information.
c    NPER = number of stress periods.
c    CINACT = value for indicating an inactive cell.
c    IFMTCN = flag whether to print the calculated
c    concentration.
c    IFMTNP = flag whether to print the number of
particles
c    in each cell.
c    IFMTRF = flag whether to print the retardation factor.
c    IFMTDP = flag whether to print the dispersion
c    coefficient.
c    SAVUCN = flag whether to save concentration in
c    unformatted file.
c    NPRS = flag indicating frequency of output.
c    NNPRS = total number of time steps.
c    TIMPRS = array of time steps at which simulation
results
c    are printed.
c    CHKMAS = flag whether to include mass balance
c    summary in output.
c    PERLEN = length of stress period.
c    NSTP = number or time steps in stress period.
c    TSMULT = multiplier for the length of successive time
c    steps.
c    DT0 = user specified transport size.
c    MXSTRN = maximum number of transport steps
allowed
c    for one time step.
c    TUNIT = time units.
c    LUNIT = length units.
c    MUNIT = mass units.

     READ(50,*)NOBS
     READ(50,*)NPER
     READ(50,*)CINACT
     READ(50,*)IFMTCN
     READ(50,*)IFMTNP
     READ(50,*)IFMTRF
     READ(50,*)IFMTDP
     READ(50,*)SAVUCN
     READ(50,*)(NPRS(I),I=1,NPER)
     READ(50,*)CHKMAS
     READ(50,*)(PERLEN(I),I=1,NPER)
     READ(50,*)(NSTP(I),I=1,NPER)
     READ(50,*)TSMULT
     READ(50,*)DT0
     READ(50,*)MXSTRN
     READ(50,*)TUNIT,LUNIT,MUNIT
```

```
c    This loop will produce TIMPRS based on the NPRS          c    Always use 0.
c    value.  TIMPRS is the                                    c    FMTIN = format specifier for reading real arrays
c    frequency of output for concentration results.           c    IFMTIN = format specifier for reading integer arrays.
                                                              c    IPRN = format specifier for printing.  See manual for
     ICOUNT=0                                                 c    codes.
     PRLN=0
     NPRSS=0                                                      READ(50,*)IREAD,CNSTNT,FMTIN,IFMTIN,IPRN
     NSTPP=0
     DO I=1,NPER                                              c    Initialize entire matrix to insure no missed cells.  Care
       PRLN=PRLN+PERLEN(I)                                    must be taken
       NPRSS=71                                               c    to insure actual parameters are set in all cells.  This
       NSTPP=71                                               c    procedure will
     END DO                                                   c    prevent any errors due to no data found.
     DO I=1,NSTPP
       TIMPRS(I)=INT(PRLN/NPRSS)+ICOUNT                           DO K=1,NLAY
       ICOUNT=TIMPRS(I)                                           DO J=1,NROW
                                                                    DO I=1,NCOL
IF(TIMPRS(NSTPP).GT.PRLN)TIMPRS(NSTPP)=PRLN                            DELR(J)=1
     END DO                                                           DELC(I)=1
                                                                      HTOP(I,J)=1
c    Read which packages to use.                                      DZ(I,J,K)=1
                                                                      PRSITY(I,J,K)=1
     READ(50,*)ADVECTION                                               ICBUND(I,J,K)=1
     READ(50,*)DISPERSION                                              SCONC(I,J,K)=1
     READ(50,*)SOURCESINK                                           END DO
     READ(50,*)REACTION                                            END DO
     DO I=1,4                                                    END DO
       TRNOP(I)='F'
     END DO                                                   c    Read cell widths in the x direction (width of rows).
     IF (ADVECTION .EQ. 'Y') TRNOP(1)='T'                     c    Read dummy variables, D*, to extract correct
     IF (DISPERSION .EQ. 'Y') TRNOP(2)='T'                    information
     IF (SOURCESINK .EQ. 'Y') TRNOP(3)='T'                    c    from file.
     IF (REACTION .EQ. 'Y') TRNOP(4)='T'                      c    ICI,IRI,ILI = number of cell boundaries in the x, y,
                                                              and z
c    Type each cell to corresponding geologic unit.           c    directions.

     NCELLS=NCOL*NROW*NLAY                                        READ(51,*)
     DO L=1, NCELLS                                               READ(51,*) ICI,IRI,ILI
       READ (100,*) F,G,H,TTYPE                                   READ(51,*)
       TYPE(F,G,H)=TTYPE                                          READ(51,*) DXM,DYM
     END DO                                                       READ(51,*)
550  format (20i3)
                                                              c    Read distances of x,y, and z cell boundaries.
c    Type=0, undefined
c    Type=1, water                                                READ(51,*) XS(1)
c    Type=2, landfill                                             DO I=2,ICI
c    Type=3, silt                                                   READ(51,*)XS(I),XINT(I-1)
c    Type=4, sand                                                 END DO
c    Type=5, gravel                                               READ(51,*)
c    Type=6, Troutdale aquifer                                    READ(51,*)YS(1)
c    Type=7, unconsolidated                                       DO J=2,IRI
c    Type=21, subarea 1                                             READ(51,*)YS(J),YINT(J-1)
c    Type=22, subarea 2                                           END DO
c    Type=23, subarea 3                                           READ(51,*)
c    Type=24, subarea 4 north                                     READ(51,*) ZS(1)
c    Type=25, subarea 5 east                                      DO K=2,ILI
c    Type=26, subarea 5a                                            READ(51,*)ZS(K),ZINT(K-1)
c    Type=27, subarea 5 west                                     END DO
c    Type=28, subarea 4 south
c    Type=29, power line corridor                             c    Calculate number of columns, rows, and layers.
c    Type=30, road
c    Type=40, engineered dike                                     IC=0
                                                                 IR=0
c    Read data to set array readers.                             IL=0
c    IREAD = read code for array readers.  Always use            DO I=2,ICI
100.                                                               IC=IC+NINT((XS(I)-XS(I-1))/XINT(I-1))
c    CNSTNT = value if parameter is constant in all cells.       END DO
```

```
    DO J=2,IRI
     IR=IR+NINT((YS(J)-YS(J-1))/YINT(J-1))
    END DO
    DO K=2,ILI
     IL=IL+NINT(ABS(ZS(K)-ZS(K-1))/ZINT(K-1))
    END DO
    NCOL=IC
    NROW=IR
    NLAY=IL
```

c    Place cell row widths into input array.

```
    B=2
    DO L=2,ICI
     A=NINT(XS(L)-XS(L-1))/XINT(L-1)
     DO K=B,B+A-1
      DO I=1,NCOL
       DELR(I)=XINT(L-1)
      END DO
     END DO
     B=B+A
    END DO
```

c    Put column addresses into data array for output to
c    PLUME.XCOL.
c    This file is used for post processing visualization
c    programs.

```
    XADD(1)=DELR(1)/2
    WXADD(1)=0
    DO I=1,NCOL-1
     XADD(I+1)=XADD(I)+((DELR(I)+DELR(I+1))/2)

WXADD(I+1)=WXADD(I)+((DELR(I)+DELR(I+1))/2)
    END DO
```

c    Place cell column widths into input array.

```
    B=2
    DO L=2,IRI
     A=NINT(YS(L)-YS(L-1))/YINT(L-1)
     DO K=B,B+A-1
      DO J=1,NROW
       DELC(J)=YINT(L-1)
      END DO
     END DO
     B=B+A
    END DO
```

c    Put row addresses into data array for output to
c    PLUME.YROW.
c    This file is used for post processing visualization
c    programs.

```
    YADD(1)=DELC(1)/2
    WYADD(1)=0
    DO J=1,NROW-1
     YADD(J+1)=YADD(J)+((DELC(J)+DELC(J+1))/2)

WYADD(J+1)=WYADD(J)+((DELC(J)+DELC(J+1))/2)
    END DO
```

c    Place top elevation of cells in first layer relative to
same
c    datum as heads.

```
    DO J=1,NROW
```

```
     DO I=1,NCOL
      HTOP(I,J)=ZS(1)
     END DO
    END DO
```

c    Place layer thickness for each cell into input array.

```
    B=2
    DO L=2,ILI
     A=NINT(ZS(L-1)-ZS(L))/ZINT(L-1)
     DO K=B,B+A-1
      DO J=1,NROW
       DO I=1,NCOL
        DZ(I,J,K-1)=ZINT(L-1)
       END DO
      END DO
     END DO
     B=B+A
    END DO
```

c    Based on layer elevation determine whether layer is
c    considered confined (0) or unconfined/variable (1).

```
    RLAY=0
    DO K=1, NLAY-1
     RLAY=DZ(1,1,K)+RLAY
     ELEV(1)=HTOP(1,1)
     ELEV(K+1)=HTOP(1,1)-RLAY
    END DO

    DO K=1,NLAY
     LAYCON(K)=0
     IF(ELEV(K).GE.0)LAYCON(K)=1
    END DO
```

c    Read porosity for each geologic layer type.
c    UNPOR = undefined layer porosity
c    WAPOR = water layer porosity
c    LAPOR = landfill layer porosity
c    SIPOR = silt layer porosity
c    SAPOR = sand layer porosity
c    GRPOR = gravel layer porosity
c    TRPOR = troutdale aquifer porosity
c    UCPOR = unconsolidated layer porosity
c    SA1POR = subarea 1 porosity
c    SA2POR = subarea 2 porosity
c    SA3POR = subarea 3 porosity
c    SA4NPOR = subarea 4 north porosity
c    SA5EPOR = subarea 5 east porosity
c    SA5APOR = subarea 5a porosity
c    SA5WPOR = subarea 5 west porosity
c    SA4SPOR = subarea 4 south porosity
c    PLCPOR = power line corridor porosity
c    RDPOR = road porosity
c    EDPOR = engineered porosity

```
READ(50,*)UNPOR,WAPOR,LAPOR,SIPOR,SAPOR,GRP
OR,TRPOR,UCPOR,
    $
SA1POR,SA2POR,SA3POR,SA4NPOR,SA5EPOR,SA5AP
OR,SA5WPOR,SA4SPOR,
    $  PLCPOR,RDPOR,EDPOR
```

c    Based on geologic type, place porosity for each cell.

346

```fortran
      DO K=1, NLAY
        DO J=1, NROW
          DO I=1, NCOL
            IF (TYPE(I,J,K).EQ.0)PRSITY(I,J,K)=UNPOR
            IF (TYPE(I,J,K).EQ.1)PRSITY(I,J,K)=WAPOR
            IF (TYPE(I,J,K).EQ.2)PRSITY(I,J,K)=LAPOR
            IF (TYPE(I,J,K).EQ.3)PRSITY(I,J,K)=SIPOR
            IF (TYPE(I,J,K).EQ.4)PRSITY(I,J,K)=SAPOR
            IF (TYPE(I,J,K).EQ.5)PRSITY(I,J,K)=GRPOR
            IF (TYPE(I,J,K).EQ.6)PRSITY(I,J,K)=TRPOR
            IF (TYPE(I,J,K).EQ.7)PRSITY(I,J,K)=UCPOR
            IF (TYPE(I,J,K).EQ.21)PRSITY(I,J,K)=SA1POR
            IF (TYPE(I,J,K).EQ.22)PRSITY(I,J,K)=SA2POR
            IF (TYPE(I,J,K).EQ.23)PRSITY(I,J,K)=SA3POR
            IF
(TYPE(I,J,K).EQ.24)PRSITY(I,J,K)=SA4NPOR
            IF
(TYPE(I,J,K).EQ.25)PRSITY(I,J,K)=SA5EPOR
            IF
(TYPE(I,J,K).EQ.26)PRSITY(I,J,K)=SA5APOR
            IF
(TYPE(I,J,K).EQ.27)PRSITY(I,J,K)=SA5WPOR
            IF
(TYPE(I,J,K).EQ.28)PRSITY(I,J,K)=SA4SPOR
            IF (TYPE(I,J,K).EQ.29)PRSITY(I,J,K)=PLCPOR
            IF (TYPE(I,J,K).EQ.30)PRSITY(I,J,K)=RDPOR
            IF (TYPE(I,J,K).EQ.40)PRSITY(I,J,K)=EDPOR
          END DO
        END DO
      END DO

c   Read type of cell for each geologic layer type.
c   UNTYPE = undefined layer type
c   WATYPE = water layer type
c   LATYPE = landfill layer type
c   SITYPE = silt layer type
c   SATYPE = sand layer type
c   GRTYPE = gravel layer type
c   TRTYPE = troutdale aquifer type
c   UCTYPE = unconsolidated layer type
c   SA1TYPE = subarea 1 type
c   SA2TYPE = subarea 2 type
c   SA3TYPE = subarea 3 type
c   SA4NTYPE = subarea 4 north type
c   SA5ETYPE = subarea 5 east type
c   SA5ATYPE = subarea 5a type
c   SA5WTYPE = subarea 5 west type
c   SA4STYPE = subarea 4 south type
c   PLCTYPE = power line corridor type
c   RDTYPE = road type
c   EDTYPE = engineered type


READ(50,*)UNTYPE,WATYPE,LATYPE,SITYPE,SATYPE,
GRTYPE,TRTYPE,
    $
UCTYPE,SA1TYPE,SA2TYPE,SA3TYPE,SA4NTYPE,SA5E
TYPE,
    $
SA5ATYPE,SA5WTYPE,SA4STYPE,PLCTYPE,RDTYPE,E
DTYPE

c   Based on geologic layer type, determine whether the
c   boundary cells
c   are inactive (0), a constant concentration cell (-1),
c   or an active variable cell (1).
c   This is also known as the IBOUND array.
```

```fortran
      DO K=1, NLAY
        DO J=1, NROW
          DO I=1, NCOL
            IF (TYPE(I,J,K).EQ.0)ICBUND(I,J,K)=UNTYPE
            IF (TYPE(I,J,K).EQ.1)ICBUND(I,J,K)=WATYPE
            IF (TYPE(I,J,K).EQ.2)ICBUND(I,J,K)=LATYPE
            IF (TYPE(I,J,K).EQ.3)ICBUND(I,J,K)=SITYPE
            IF (TYPE(I,J,K).EQ.4)ICBUND(I,J,K)=SATYPE
            IF (TYPE(I,J,K).EQ.5)ICBUND(I,J,K)=GRTYPE
            IF (TYPE(I,J,K).EQ.6)ICBUND(I,J,K)=TRTYPE
            IF (TYPE(I,J,K).EQ.7)ICBUND(I,J,K)=UCTYPE
            IF
(TYPE(I,J,K).EQ.21)ICBUND(I,J,K)=SA1TYPE
            IF
(TYPE(I,J,K).EQ.22)ICBUND(I,J,K)=SA2TYPE
            IF
(TYPE(I,J,K).EQ.23)ICBUND(I,J,K)=SA3TYPE
            IF
(TYPE(I,J,K).EQ.24)ICBUND(I,J,K)=SA4NTYPE
            IF
(TYPE(I,J,K).EQ.25)ICBUND(I,J,K)=SA5ETYPE
            IF
(TYPE(I,J,K).EQ.26)ICBUND(I,J,K)=SA5ATYPE
            IF
(TYPE(I,J,K).EQ.27)ICBUND(I,J,K)=SA5WTYPE
            IF
(TYPE(I,J,K).EQ.28)ICBUND(I,J,K)=SA4STYPE
            IF
(TYPE(I,J,K).EQ.29)ICBUND(I,J,K)=PLCTYPE
            IF
(TYPE(I,J,K).EQ.30)ICBUND(I,J,K)=RDTYPE
            IF
(TYPE(I,J,K).EQ.40)ICBUND(I,J,K)=EDTYPE
c
c   Making part of grid inactive that will never see
c   contaminant
c
            if(i.ge.1.and.i.le.10)then
              if(j.ge.6.and.j.le.39)then
                icbund(i,j,k)=0
              endif
            endif
            if(i.ge.41.and.i.le.51)then
              if(j.ge.32.and.j.le.39)then
                icbund(i,j,k)=0
              endif
            endif
            if(i.ge.35.and.i.le.51)then
              if(j.ge.1.and.j.le.7)then
                icbund(i,j,k)=0
              endif
            endif

          END DO
        END DO
      END DO


c   Read starting concentrations for each geologic layer
type.
c   UNCONC = undefined layer initial concentration
c   WACONC = water layer initial concentration
c   LACONC = landfill layer initial concentration
c   SICONC = silt layer initial concentration
c   SACONC = sand layer initial concentration
c   GRCONC = gravel layer initial concentration
```

347

```
c   TRCONC = troutdale aquifer initial concentration
c   UCCONC = unconsolidated layer initial concentration
c   SA1CONC = subarea 1 initial concentration
c   SA2CONC = subarea 2 initial concentration
c   SA3CONC = subarea 3 initial concentration
c   SA4NCONC = subarea 4 north initial concentration
c   SA5CONC = subarea 5 east initial concentration
c   SA5ACONC = subarea 5a initial concentration
c   SA5WCONC = subarea 5 west initial concentration
c   SA4SCONC = subarea 4 south initial concentration
c   PLCCONC = powerline corridor initial concentration
c   RDCONC = road initial concentration
c   EDCONC = engineered initial concentration


READ(50,*)UNCONC,WACONC,LACONC,SICONC,SAC
ONC,GRCONC,TRCONC,
    $
UCCONC,SA1CONC,SA2CONC,SA3CONC,SA4NCONC,
SA5ECONC,
    $
SA5ACONC,SA5WCONC,SA4SCONC,PLCCONC,RDCO
NC,EDCONC

c   Place starting concentrations into each cell.

    DO K=1, NLAY
      DO J=1, NROW
        DO I=1, NCOL
          IF (TYPE(I,J,K).EQ.0)SCONC(I,J,K)=UNCONC
          IF (TYPE(I,J,K).EQ.1)SCONC(I,J,K)=WACONC
          IF (TYPE(I,J,K).EQ.2)SCONC(I,J,K)=LACONC
          IF (TYPE(I,J,K).EQ.3)SCONC(I,J,K)=SICONC
          IF (TYPE(I,J,K).EQ.4)SCONC(I,J,K)=SACONC
          IF (TYPE(I,J,K).EQ.5)SCONC(I,J,K)=GRCONC
          IF (TYPE(I,J,K).EQ.6)SCONC(I,J,K)=TRCONC
          IF (TYPE(I,J,K).EQ.7)SCONC(I,J,K)=UCCONC
          IF
(TYPE(I,J,K).EQ.21)SCONC(I,J,K)=SA1CONC
          IF
(TYPE(I,J,K).EQ.22)SCONC(I,J,K)=SA2CONC
          IF
(TYPE(I,J,K).EQ.23)SCONC(I,J,K)=SA3CONC
          IF
(TYPE(I,J,K).EQ.24)SCONC(I,J,K)=SA4NCONC
          IF
(TYPE(I,J,K).EQ.25)SCONC(I,J,K)=SA5ECONC
          IF
(TYPE(I,J,K).EQ.26)SCONC(I,J,K)=SA5ACONC
          IF
(TYPE(I,J,K).EQ.27)SCONC(I,J,K)=SA5WCONC
          IF
(TYPE(I,J,K).EQ.28)SCONC(I,J,K)=SA4SCONC
          IF
(TYPE(I,J,K).EQ.29)SCONC(I,J,K)=PLCCONC
          IF
(TYPE(I,J,K).EQ.30)SCONC(I,J,K)=RDCONC
          IF
(TYPE(I,J,K).EQ.40)SCONC(I,J,K)=EDCONC
          END DO
        END DO
      END DO


c   Read well positions in which an observation point
c   is located.
```

```
    DO I=1,NOBS
      READ
(25,*)WELLNM(I),WELLX(I),WELLY(I),WELLZ(I)
      WELLNO(I)=1
    END DO

c   Convert well positions to cell indices - Z direction.

    DO KK=1,NOBS
      DO K=1,NLAY-1
        IF(WELLZ(KK).EQ.0)THEN
          KOBS(KK)=1
        ELSE

IF(ELEV(K).GT.WELLZ(KK).AND.ELEV(K+1).LE.WELL
Z(KK))THEN
          AVE=(ELEV(K)+ELEV(K+1))/2
          IF(WELLZ(KK).GE.AVE)THEN
            KOBS(KK)=K
          ELSE
            KOBS(KK)=K+1
          ENDIF
        ELSE
          GOTO 1000
        ENDIF
      ENDIF
 1000    CONTINUE
    END DO
  END DO


c   Convert well positions to cell indices - X direction.

    DO II=1,NOBS
      IF(WELLZ(II).EQ.0)THEN
        IOBS(II)=1
      ELSE

IF(WELLX(II).GE.8100.AND.WELLX(II).LE.15750)THEN
        TEMP=WELLX(II)-8100
        value=temp/150.0
        ivalue=int(value)
        if((value-ivalue).ge..5)then
          iobs(ii)=ivalue+1
        else
          iobs(ii)=ivalue
        endif
      ENDIF
    ENDIF
    END DO

c   Convert well positions to cell indices - Y direction.

    DO JJ=1,NOBS
      IF(WELLZ(JJ).EQ.0)THEN
        JOBS(JJ)=1
      ELSE

IF(WELLY(JJ).LE.15300.AND.WELLY(JJ).GE.9450)THE
N
        TEMP=15300-WELLY(JJ)
        value=temp/150.0
        jvalue=int(value)
        if((value-jvalue).ge..5)then
          jobs(jj)=jvalue+1
        else
          jobs(jj)=jvalue
```

```fortran
            endif
          ENDIF
        ENDIF
      END DO

c   Get rid of wells without cell indices and adjust NOBS.

      NNOBS=0
      DO I=1,NOBS

IF(IOBS(I).NE.1.AND.JOBS(I).NE.1.AND.KOBS(I).NE.1)
THEN
          NNOBS=NNOBS+1
          IIOBS(NNOBS)=IOBS(I)
          JJOBS(NNOBS)=JOBS(I)
          KKOBS(NNOBS)=KOBS(I)
          WWELLNM(NNOBS)=WELLNM(I)
          WWELLNO(NNOBS)=WELLNO(I)
        ELSE
          CONTINUE
        ENDIF
      END DO

c   Add observation point below Bybee Lake

      NNOBS=NNOBS+1
      IIOBS(NNOBS)=12
      JJOBS(NNOBS)=24
      KKOBS(NNOBS)=19
      WWELLNM(NNOBS)='BYBEE'
      WWELLNO(NNOBS)=NNOBS

c   Produce output to Basic Transport File BTN.INP

      PRINT *,'Writing to basic transport input file
BTN.INP.'

      WRITE (10,100) HEADING(1)
      WRITE (10,100) HEADING(2)
 100  FORMAT (A80)
      WRITE (10,110) NLAY,NROW,NCOL,NPER
 110  FORMAT (4I10)
      WRITE (10,120) TUNIT,LUNIT,MUNIT
 120  FORMAT (3A4)
      WRITE (10,130)
TRNOP(1),TRNOP(2),TRNOP(3),TRNOP(4)
 130  FORMAT (10A2)
      WRITE (10,140) (LAYCON(K),K=1,NLAY)
 140  FORMAT (40I2)
      WRITE (10,150) IREAD,CNSTNT,FMTIN,IPRN
 150  FORMAT (I10,F10.0,10X,A20,I10)
      WRITE (10,160) (DELR(I),I=1,NCOL)
 160  FORMAT (10F8.2)
      WRITE (10,150) IREAD,CNSTNT,FMTIN,IPRN
      WRITE (10,160) (DELC(J),J=1,NROW)
      WRITE (10,150) IREAD,CNSTNT,FMTIN,IPRN
      DO J=1,NROW
        WRITE (10,160) (HTOP(I,J),I=1,NCOL)
      END DO
      DO K=1, NLAY
        WRITE (10,150) IREAD,CNSTNT,FMTIN,IPRN
        DO J=1,NROW
          WRITE (10,160) (DZ(I,J,K),I=1,NCOL)
        END DO
      END DO
      DO K=1, NLAY
        WRITE (10,150) IREAD,CNSTNT,FMTIN,IPRN
        DO J=1,NROW
          WRITE (10,160) (PRSITY(I,J,K),I=1,NCOL)
        END DO
      END DO
      DO K=1, NLAY
        WRITE (10,155) IREAD,CNSTNT,IFMTIN,IPRN
        DO J=1,NROW
          WRITE (10,165) (ICBUND(I,J,K),I=1,NCOL)
        END DO
      END DO
 155  FORMAT (I10,I10,10X,A20,I10)
 165  FORMAT (20I2)
      DO K=1, NLAY
        WRITE (10,150) IREAD,CNSTNT,FMTIN,IPRN
        DO J=1,NROW
          WRITE (10,160) (SCONC(I,J,K),I=1,NCOL)
        END DO
      END DO
      WRITE (10,170) CINACT
 170  FORMAT (F10.0)
      WRITE (10,180)
IFMTCN,IFMTNP,IFMTRF,IFMTDP,SAVUCN
 180  FORMAT (4I10,A10)
      WRITE (10, 190) NPRSS
 190  FORMAT (I10)
      IF(NPRSS.GT.0) WRITE(10,
200)(TIMPRS(I),I=1,NPRSS)
 200  FORMAT (8F10.0)
      WRITE (10, 210) NNOBS
 210  FORMAT (I10)
      WRITE (20,*) ' WELL NAME','    LAY ','  COL ',
     & ' ROW '
      WRITE (20,226)
 226  FORMAT (44('_'))
      DO I=1,NNOBS
        WRITE (10, 220) KKOBS(I),IIOBS(I),JJOBS(I)
        WRITE (20, 225)
WWELLNM(I),KKOBS(I),IIOBS(I),JJOBS(I)
      END DO
 220  FORMAT (3I10)
 225  FORMAT (A10,3I10)
      WRITE (10, 230) CHKMAS
 230  FORMAT (A10)
      DO I=1,NPER
        WRITE (10, 240) PERLEN(I), NSTP(I), TSMULT
        WRITE (10, 250) DT0, MXSTRN
      END DO
 240  FORMAT (F10.0,I10,F10.0)
 250  FORMAT (F10.0,I10)

c   Produce output for AVS and post processing usage.

      PRINT *,'Writing grid dimensions to PLUME.*.'
      DO I=1,NCOL
        WRITE(101,260)XADD(I)
      END DO
 260  FORMAT(F8.1)
      DO J=1,NROW
        WRITE(102,260)YADD(J)
      END DO
      DO K=1,NLAY-1
        WRITE(103,260)((ELEV(K)+ELEV(K+1))/2)
      END DO
      WRITE(103,260)(((ELEV(NLAY-1)+ELEV(NLAY))/2)-
DZ(1,1,NLAY))
```

349

```
c    Call subroutine to produce advection package input
file.

     IF (ADVECTION .EQ. 'Y') CALL ADVEC

c    Call subroutine to produce dispersion package input
file.

     IF (DISPERSION .EQ. 'Y') CALL DISPER

c    Call subroutine to produce source/sink mixing input
file.

     IF (SOURCESINK .EQ. 'Y') CALL SSM

c    Call subroutine to produce reaction input file.

     IF (REACTION .EQ. 'Y') CALL REACT

     STOP
     END

***************************************************
*****************************

     SUBROUTINE ADVEC

c  Subroutine to produce advection input file.
c  Data for ADVEC is read from ADV.OUT.

     OPEN(60,FILE='adv.out',STATUS='OLD')
     OPEN(11,FILE='ADV.INP',STATUS='UNKNOWN')

c  Read type of solving technique.
c  1=method of characteristics, 2= modified method of
c  characteristics
c  3=hybrid method of characteristics, 4=upstream finite
c  difference.

     READ(60,*) MIXELM

c  Read Courant number and maximum number or
particles
c  per cell.
c  Courant number (PERCEL) determines the maximum
c  number of cells one
c  particle can move in one given time step.

     READ(60,*) PERCEL
     READ(60,*) MXPART

c  Read which particle tracking algorithm to use.  1=Euler,
c  2=fourth order Runge-Kutta, 3=hybrid of 1 and 2.
c  WD is concentration weighting factor.

     READ(60,*) ITRACK
     READ(60,*) WD

c  DCEPS=relative concentration gradient to be
considered
c  negligible.
c  NPLANE=determines random or fixed pattern for initial
c  placement.
c  NPL=number of initial particles per cell where cell
c  conc.<=DCEPS.
c  NPH=number of initial particles per cell where cell
c  conc.>DCEPS.

c  NPMIN=minimum number of moving particles per cell.
c  NPMAX=maximum number of moving particles per cell.
c  SRMULT=multiplier for particle number at source cells.

     READ(60,*) DCEPS
     READ(60,*) NPLANE
     READ(60,*) NPL
     READ(60,*) NPH
     READ(60,*) NPMIN
     READ(60,*) NPMAX
     READ(60,*) SRMULT

c  INTERP=concentration interpolation method to use.
c  NLSINK=determines NPLANE in MMOC for sink cells.
c  NPSINK=determines NPH in MMOC for sink cells.

     READ(60,*) INTERP
     READ(60,*) NLSINK
     READ(60,*) NPSINK

c  DCHMOC=critical relative concentration in the HMOC
c  scheme.

     READ(60,*) DCHMOC

c  Produce output file.

     PRINT *,'Writing to advection input file ADV.INP.'

     WRITE(11,100)MIXELM,PERCEL,MXPART
 100 FORMAT(I10,F10.2,I10)
     IF(MIXELM.LE.3.AND.MIXELM.GE.1)
WRITE(11,110)ITRACK,WD
 110 FORMAT(I10,F10.2)
     IF(MIXELM.EQ.1.OR.MIXELM.EQ.3)
WRITE(11,120)DCEPS,NPLANE,
     $NPL,NPH,NPMIN,NPMAX,SRMULT
 120 FORMAT(F10.7,5I10,F10.7)
     IF(MIXELM.EQ.2.OR.MIXELM.EQ.3)
WRITE(11,130)INTERP,NLSINK,NPSINK
 130 FORMAT(3I10)
     IF(MIXELM.EQ.3) WRITE(11,140)DCHMOC
 140 FORMAT(F10.3)
     RETURN
     END
***************************************************
*****************************

     SUBROUTINE DISPER

c    Subroutine to produce dispersivity output package.
c    Data for DISPER is read from DSP.OUT.

     INTEGER LAY,COL,ROW,IREAD,IPRN
     PARAMETER (LAY=39,COL=51,ROW=39)
     DIMENSION
TYPE(COL,ROW,LAY),AL(COL,ROW,LAY),TRPT(LAY)
     DIMENSION TRPV(LAY),DMCOEF(LAY)
     CHARACTER FMTIN*20
     COMMON
/INFO/TYPE,NLAY,NROW,NCOL,IREAD,CNSTNT
     COMMON /INFO/FMTIN,IPRN,NPER
     OPEN(70,FILE='dsp.out',STATUS='OLD')
     OPEN(12,FILE='DSP.INP',STATUS='UNKNOWN')

c    Read longitudinal dispersivity for each geologic layer.
```

350

```
READ(70,*)UDDISP,WADISP,LADISP,SIDISP,SADISP,G
RDISP,TRDISP,
     $
UNDISP,SA1DISP,SA2DISP,SA3DISP,SA4NDISP,SA5ED
ISP,
     $
SA5ADISP,SA5WDISP,SA4SDISP,PLCDISP,RDDISP,ED
DISP

     DO K=1, NLAY
       DO J=1, NROW
         DO I=1, NCOL
           IF (TYPE(I,J,K).EQ.0)AL(I,J,K)=UDDISP
           IF (TYPE(I,J,K).EQ.1)AL(I,J,K)=WADISP
           IF (TYPE(I,J,K).EQ.2)AL(I,J,K)=LADISP
           IF (TYPE(I,J,K).EQ.3)AL(I,J,K)=SIDISP
           IF (TYPE(I,J,K).EQ.4)AL(I,J,K)=SADISP
           IF (TYPE(I,J,K).EQ.5)AL(I,J,K)=GRDISP
           IF (TYPE(I,J,K).EQ.6)AL(I,J,K)=TRDISP
           IF (TYPE(I,J,K).EQ.7)AL(I,J,K)=UNDISP
           IF (TYPE(I,J,K).EQ.21)AL(I,J,K)=SA1DISP
           IF (TYPE(I,J,K).EQ.22)AL(I,J,K)=SA21DISP
           IF (TYPE(I,J,K).EQ.23)AL(I,J,K)=SA3DISP
           IF (TYPE(I,J,K).EQ.24)AL(I,J,K)=SA4NDISP
           IF (TYPE(I,J,K).EQ.25)AL(I,J,K)=SA5EDISP
           IF (TYPE(I,J,K).EQ.26)AL(I,J,K)=SA5ADISP
           IF (TYPE(I,J,K).EQ.27)AL(I,J,K)=SA5WDISP
           IF (TYPE(I,J,K).EQ.28)AL(I,J,K)=SA4SDISP
           IF (TYPE(I,J,K).EQ.29)AL(I,J,K)=PLCDISP
           IF (TYPE(I,J,K).EQ.30)AL(I,J,K)=RDDISP
           IF (TYPE(I,J,K).EQ.40)AL(I,J,K)=EDDISP
         END DO
       END DO
     END DO

c    Read ratio of transverse and vertical disp. to
longitudinal.


     READ(70,*)TRATIO,VRATIO,DIFF
     DO K=1, NLAY
       TRPT(K)=TRATIO
     END DO

     DO K=1, NLAY
       TRPV(K)=VRATIO
     END DO

     DO K=1, NLAY
       DMCOEF(K)=DIFF
     END DO

c    Produce output to dispersion package input file.

     PRINT *,'Writing to dispersion input file DSP.INP'

150  FORMAT (I10,F10.0,10X,A20,I10)
     DO K=1,NLAY
       WRITE (12,150) IREAD,CNSTNT,FMTIN,IPRN
       DO J=1,NROW
         WRITE (12,160) (AL(I,J,K),I=1,NCOL)
       END DO
     END DO
160  FORMAT (10F8.2)
     WRITE (12,150) IREAD,CNSTNT,FMTIN,IPRN
     WRITE(12,170)(TRPT(K),K=1,NLAY)
```

```
     WRITE (12,150) IREAD,CNSTNT,FMTIN,IPRN
170  FORMAT(10F8.2)
     WRITE(12,170)(TRPV(K),K=1,NLAY)
     WRITE (12,150) IREAD,CNSTNT,FMTIN,IPRN
     WRITE(12,170)(DMCOEF(K),K=1,NLAY)
     RETURN
     END

**************************************************
*******************************

     SUBROUTINE SSM

c  Subroutine to produce source/sink mixing package input.
c  Data for SSM is read from file SSM.OUT.

     INTEGER LAY,COL,ROW,IREAD,IPRN
     PARAMETER (LAY=39,COL=51,ROW=39)
     DIMENSION
TYPE(COL,ROW,LAY),CRCH(COL,ROW),CEVT(COL,RO
W)
     DIMENSION
KSS(100),ISS(100),JSS(100),CSS(100),ITYPE(100)
     CHARACTER
FMTIN*20,FWEL,FDRN,FRCH,FEVT,FRIV,FGHB
     COMMON
/INFO/TYPE,NLAY,NROW,NCOL,IREAD,CNSTNT
     COMMON /INFO/FMTIN,IPRN,NPER
     OPEN(80,FILE='ssm.out',STATUS='OLD')
     OPEN(13,FILE='SSM.INP',STATUS='UNKNOWN')

c  Read which options to use for SSM package.
c  FWEL = well option
c  FDRN = drain option
c  FRCH = recharge option
c  FEVT = evapotranspiration option
c  FRIV = river option
c  FGHB = general-head-dependent boundary option

     READ(80,*)FWEL,FDRN,FRCH,FEVT,FRIV,FGHB

c  Read maximum number of all point sinks and sources
c  simulated.

     READ(80,*)MXSS

c  Read flag to specify whether concentration of recharge
flux
c  will be read.

     READ(80,*)INCRCH

c  Read concentration of recharge flux.
c  If concentration is negative, recharge acts as discharge
with
c  concentration always set equal to the concentration of
the
c  aquifer
c  at the cell where discharge occurs.
c  Positive concentration will be used as user specified
c  concentration for
c  recharge cells.

     READ(80,*)CONC1

     DO J=1, NROW
       DO I=1, NCOL
```

```fortran
      CRCH(NCOL,NROW)=CONC1
      END DO
      END DO

c Read flag to specify whether concentration of
c evapotranspiration flux
c will be read.

      READ(80,*)INCEVT

c Read concentration of evapotranspiration flux.  List up
to 4
c concentrations.
c If specified concentration is higher than that of aquifer,
c aquifer concentration
c will be used.

      READ(80,*)CONC2

      DO J=1, NROW
       DO I=1, NCOL
         CEVT(NCOL,NROW)=CONC2
      END DO
      END DO

c Read number of point sources whose concentrations
need to
c be specified.
c Unspecified point sources are assumed to have a zero
c concentration.

      READ(80,*)NSS

c Read cell indices (layer, row, column) of the point
source
c for which a
c concentration is to be specified.  Also read specified
c concentration (CSS)
c as well as type of point source as indicated below:
c 1 = constant head cell
c 2 = well
c 3 = drain
c 4 = river
c 5 = general-head-dependent boundary cell

      DO L=1,NSS
        READ(80,*)KSS(L),ISS(L),JSS(L),CSS(L),ITYPE(L)
      END DO

c Produce output to source/sink mixing package input file.

      PRINT *,'Writing to source/sink input file SSM.INP.'

 150 FORMAT (I10,F10.0,10X,A20,I10)

      WRITE(13,100)FWEL,FDRN,FRCH,FEVT,FRIV,FGHB
 100 FORMAT(6A2)
      WRITE(13,110)MXSS
 110 FORMAT(I10)
      DO M=1,NPER
        IF(FRCH.EQ.'T')WRITE(13,120)INCRCH
        IF(FRCH.EQ.'T'.AND.INCRCH.GE.0)THEN
         WRITE (13,150) IREAD,CNSTNT,FMTIN,IPRN

      WRITE(13,130)((CRCH(I,J),I=1,NCOL),J=1,NROW)
        ENDIF
        IF(FEVT.EQ.'T')WRITE(13,120)INCEVT
        IF(FEVT.EQ.'T'.AND.INCEVT.GE.0)THEN
         WRITE (13,150) IREAD,CNSTNT,FMTIN,IPRN

      WRITE(13,130)((CEVT(I,J),I=1,NCOL),J=1,NROW)
        ENDIF
        WRITE(13,120)NSS
        IF(NSS.GT.0)THEN
         DO L=1,NSS

      WRITE(13,140)KSS(L),ISS(L),JSS(L),CSS(L),ITYPE(L)
         END DO
        ENDIF
      END DO
 120 FORMAT(I10)
 130 FORMAT(10F8.2)
 140 FORMAT(3I10,F10.8,I10)
      RETURN
      END

c*************************************************
c*****************************

      SUBROUTINE REACT

c Subroutine to produce chemical reaction package input.
c Data for REACT is read from file RCT.OUT.
c NTYPE is the number of geologic layers in the
c computational area.

      INTEGER LAY,COL,ROW,IREAD,IPRN
      PARAMETER
(LAY=39,COL=51,ROW=39,NTYPE=19)
      DIMENSION TYPE(COL,ROW,LAY)
      DIMENSION
TYPEG(NTYPE),RHOB(LAY),SP1(LAY),SP2(LAY)
      DIMENSION RC1(LAY),RC2(LAY)
      CHARACTER FMTIN*20
      COMMON
/INFO/TYPE,NLAY,NROW,NCOL,IREAD,CNSTNT
      COMMON /INFO/FMTIN,IPRN,NPER
      OPEN(90,FILE='rct.out',STATUS='OLD')
      OPEN(14,FILE='RCT.INP',STATUS='UNKNOWN')

c Read which, if any, sorption type is simulated
(ISOTHM).
c 1 = linear isotherm
c 2 = Freundlich isotherm
c 3 = Langmuir isotherm
c 0 = no sorption isotherm is simulated.
c IREACT = flag indicating whether first-order rate
reaction
c is simulated.
c 1 = first order reactions are simulated.
c 0 = no first order reactions are simulated.

      READ(90,*)ISOTHM,IREACT

c Read bulk density of the porous medium in the aquifer.
c RHOB is a 1 dimensional array.  One value for each
layer.
c Layer values are read from RCT.OUT.  Value given to
each
c computational
c layer will be determined from the number of cells in
each
c layer which fit
```

```
c  a certain type. The type with the greatest number of
cells in
c  each
c  computational layer will be the type for the entire layer.

      READ(90,*)
UDDENS,WADENS,LADENS,SIDENS,SADENS,GRDENS,
TRDENS,
      $
UNDENS,SA1DENS,SA2DENS,SA3DENS,SA4NDENS,SA
5EDENS,SA5ADENS,
      $
SA5WDENS,SA4SDENS,PLCDENS,RDDENS,EDDENS

      READ(90,*)
UDSP1,WASP1,LASP1,SISP1,SASP1,GRSP1,TRSP1,
      $
UNSP1,SA1SP1,SA2SP1,SA3SP1,SA4NSP1,SA5ESP1,SA5
ASP1,
      $  SA5WSP1,SA4SSP1,PLCSP1,RDSP1,EDSP1

      READ(90,*)
UDSP2,WASP2,LASP2,SISP2,SASP2,GRSP2,TRSP2,
      $
UNSP2,SA1SP2,SA2SP2,SA3SP2,SA4NSP2,SA5ESP2,SA5
ASP2,
      $  SA5WSP2,SA4SSP2,PLCSP2,RDSP2,EDSP2

      READ(90,*)
UDRC1,WARC1,LARC1,SIRC1,SARC1,GRRC1,TRRC1,
      $
UNRC1,SA1RC1,SA2RC1,SA3RC1,SA4NRC1,SA5ERC1,S
A5ARC1,
      $  SA5WRC1,SA4SRC1,PLCRC1,RDRC1,EDRC1

      READ(90,*)
UDRC2,WARC2,LARC2,SIRC2,SARC2,GRRC2,TRRC2,
      $
UNRC2,SA1RC2,SA2RC2,SA3RC2,SA4NRC2,SA5ERC2,S
A5ARC2,
      $  SA5WRC2,SA4SRC2,PLCRC2,RDRC2,EDRC2

      TYPE0=0
      TYPE1=0
      TYPE2=0
      TYPE3=0
      TYPE4=0
      TYPE5=0
      TYPE6=0
      TYPE7=0
      TYPE21=0
      TYPE22=0
      TYPE23=0
      TYPE24=0
      TYPE25=0
      TYPE26=0
      TYPE27=0
      TYPE28=0
      TYPE29=0
      TYPE30=0
      TYPE40=0
      DO K=1,NLAY

c  This loop will determine the most common type in each
computational layer.

      DO I=1,NCOL

      DO J=1,NROW
        IF(TYPE(I,J,K).EQ.0)TYPE0=TYPE0+1
        IF(TYPE(I,J,K).EQ.1)TYPE1=TYPE1+1
        IF(TYPE(I,J,K).EQ.2)TYPE2=TYPE2+1
        IF(TYPE(I,J,K).EQ.3)TYPE3=TYPE3+1
        IF(TYPE(I,J,K).EQ.4)TYPE4=TYPE4+1
        IF(TYPE(I,J,K).EQ.5)TYPE5=TYPE5+1
        IF(TYPE(I,J,K).EQ.6)TYPE6=TYPE6+1
        IF(TYPE(I,J,K).EQ.7)TYPE7=TYPE7+1
        IF (TYPE(I,J,K).EQ.21)TYPE21=TYPE21+1
        IF (TYPE(I,J,K).EQ.22)TYPE22=TYPE22+1
        IF (TYPE(I,J,K).EQ.23)TYPE23=TYPE23+1
        IF (TYPE(I,J,K).EQ.24)TYPE24=TYPE24+1
        IF (TYPE(I,J,K).EQ.25)TYPE25=TYPE25+1
        IF (TYPE(I,J,K).EQ.26)TYPE26=TYPE26+1
        IF (TYPE(I,J,K).EQ.27)TYPE27=TYPE27+1
        IF (TYPE(I,J,K).EQ.28)TYPE28=TYPE28+1
        IF (TYPE(I,J,K).EQ.29)TYPE29=TYPE29+1
        IF (TYPE(I,J,K).EQ.30)TYPE30=TYPE30+1
        IF (TYPE(I,J,K).EQ.40)TYPE40=TYPE40+1
      END DO
      END DO
      TYPEG(1)=TYPE0
      TYPEG(2)=TYPE1
      TYPEG(3)=TYPE2
      TYPEG(4)=TYPE3
      TYPEG(5)=TYPE4
      TYPEG(6)=TYPE5
      TYPEG(7)=TYPE6
      TYPEG(8)=TYPE7
      TYPEG(9)=TYPE21
      TYPEG(10)=TYPE22
      TYPEG(11)=TYPE23
      TYPEG(12)=TYPE24
      TYPEG(13)=TYPE25
      TYPEG(14)=TYPE26
      TYPEG(15)=TYPE27
      TYPEG(16)=TYPE28
      TYPEG(17)=TYPE29
      TYPEG(18)=TYPE30
      TYPEG(19)=TYPE40

      MAX=TYPEG(1)
      TYPE(1,1,K)=0
      DO L=1,NTYPE
        IF (TYPEG(L).GT.MAX) THEN
          MAX=TYPEG(L)
          TYPE(1,1,K)=L-1
        ENDIF
      END DO

c  Determine and type the bulk density.

      IF (TYPE(1,1,K).EQ.0)RHOB(K)=UDDENS
      IF (TYPE(1,1,K).EQ.1)RHOB(K)=WADENS
      IF (TYPE(1,1,K).EQ.2)RHOB(K)=LADENS
      IF (TYPE(1,1,K).EQ.3)RHOB(K)=SIDENS
      IF (TYPE(1,1,K).EQ.4)RHOB(K)=SADENS
      IF (TYPE(1,1,K).EQ.5)RHOB(K)=GRDENS
      IF (TYPE(1,1,K).EQ.6)RHOB(K)=TRDENS
      IF (TYPE(1,1,K).EQ.7)RHOB(K)=UNDENS
      IF (TYPE(1,1,K).EQ.21)RHOB(K)=SA1DENS
      IF (TYPE(1,1,K).EQ.22)RHOB(K)=SA2DENS
      IF (TYPE(1,1,K).EQ.23)RHOB(K)=SA3DENS
      IF (TYPE(1,1,K).EQ.24)RHOB(K)=SA4NDENS
      IF (TYPE(1,1,K).EQ.25)RHOB(K)=SA5EDENS
      IF (TYPE(1,1,K).EQ.26)RHOB(K)=SA5ADENS
```

353

```
        IF (TYPE(1,1,K).EQ.27)RHOB(K)=SA5WDENS
        IF (TYPE(1,1,K).EQ.28)RHOB(K)=SA4SDENS
        IF (TYPE(1,1,K).EQ.29)RHOB(K)=PLCDENS
        IF (TYPE(1,1,K).EQ.30)RHOB(K)=RDDENS
        IF (TYPE(1,1,K).EQ.40)RHOB(K)=EDDENS


c  Using same criteria as above, determine first sorption
coefficient.

        IF (TYPE(1,1,K).EQ.0)SP1(K)=UDSP1
        IF (TYPE(1,1,K).EQ.1)SP1(K)=WASP1
        IF (TYPE(1,1,K).EQ.2)SP1(K)=LASP1
        IF (TYPE(1,1,K).EQ.3)SP1(K)=SISP1
        IF (TYPE(1,1,K).EQ.4)SP1(K)=SASP1
        IF (TYPE(1,1,K).EQ.5)SP1(K)=GRSP1
        IF (TYPE(1,1,K).EQ.6)SP1(K)=TRSP1
        IF (TYPE(1,1,K).EQ.7)SP1(K)=UNSP1
        IF (TYPE(1,1,K).EQ.21)SP1(K)=SA1SP1
        IF (TYPE(1,1,K).EQ.22)SP1(K)=SA2SP1
        IF (TYPE(1,1,K).EQ.23)SP1(K)=SA3SP1
        IF (TYPE(1,1,K).EQ.24)SP1(K)=SA4NSP1
        IF (TYPE(1,1,K).EQ.25)SP1(K)=SA5ESP1
        IF (TYPE(1,1,K).EQ.26)SP1(K)=SA5ASP1
        IF (TYPE(1,1,K).EQ.27)SP1(K)=SA5WSP1
        IF (TYPE(1,1,K).EQ.28)SP1(K)=SA4SSP1
        IF (TYPE(1,1,K).EQ.29)SP1(K)=PLCSP1
        IF (TYPE(1,1,K).EQ.30)SP1(K)=RDSP1
        IF (TYPE(1,1,K).EQ.40)SP1(K)=EDSP1

c  Determine second sorption coefficient.

        IF (TYPE(1,1,K).EQ.0)SP2(K)=UDSP2
        IF (TYPE(1,1,K).EQ.1)SP2(K)=WASP2
        IF (TYPE(1,1,K).EQ.2)SP2(K)=LASP2
        IF (TYPE(1,1,K).EQ.3)SP2(K)=SISP2
        IF (TYPE(1,1,K).EQ.4)SP2(K)=SASP2
        IF (TYPE(1,1,K).EQ.5)SP2(K)=GRSP2
        IF (TYPE(1,1,K).EQ.6)SP2(K)=TRSP2
        IF (TYPE(1,1,K).EQ.7)SP2(K)=UNSP2
        IF (TYPE(1,1,K).EQ.21)SP2(K)=SA1SP2
        IF (TYPE(1,1,K).EQ.22)SP2(K)=SA2SP2
        IF (TYPE(1,1,K).EQ.23)SP2(K)=SA3SP2
        IF (TYPE(1,1,K).EQ.24)SP2(K)=SA4NSP2
        IF (TYPE(1,1,K).EQ.25)SP2(K)=SA5ESP2
        IF (TYPE(1,1,K).EQ.26)SP2(K)=SA5ASP2
        IF (TYPE(1,1,K).EQ.27)SP2(K)=SA5WSP2
        IF (TYPE(1,1,K).EQ.28)SP2(K)=SA4SSP2
        IF (TYPE(1,1,K).EQ.29)SP2(K)=PLCSP2
        IF (TYPE(1,1,K).EQ.30)SP2(K)=RDSP2
        IF (TYPE(1,1,K).EQ.40)SP2(K)=EDSP2

c  Determine the first order rate constant for the dissolved
c  phase.

        IF (TYPE(1,1,K).EQ.0)RC1(K)=UDRC1
        IF (TYPE(1,1,K).EQ.1)RC1(K)=WARC1
        IF (TYPE(1,1,K).EQ.2)RC1(K)=LARC1
        IF (TYPE(1,1,K).EQ.3)RC1(K)=SIRC1
        IF (TYPE(1,1,K).EQ.4)RC1(K)=SARC1
        IF (TYPE(1,1,K).EQ.5)RC1(K)=GRRC1
        IF (TYPE(1,1,K).EQ.6)RC1(K)=TRRC1
        IF (TYPE(1,1,K).EQ.7)RC1(K)=UNRC1
        IF (TYPE(1,1,K).EQ.21)RC1(K)=SA1RC1
        IF (TYPE(1,1,K).EQ.22)RC1(K)=SA2RC1
        IF (TYPE(1,1,K).EQ.23)RC1(K)=SA3RC1
        IF (TYPE(1,1,K).EQ.24)RC1(K)=SA4NRC1
```

```
        IF (TYPE(1,1,K).EQ.25)RC1(K)=SA5ERC1
        IF (TYPE(1,1,K).EQ.26)RC1(K)=SA5ARC1
        IF (TYPE(1,1,K).EQ.27)RC1(K)=SA5WRC1
        IF (TYPE(1,1,K).EQ.28)RC1(K)=SA4SRC1
        IF (TYPE(1,1,K).EQ.29)RC1(K)=PLCRC1
        IF (TYPE(1,1,K).EQ.30)RC1(K)=RDRC1
        IF (TYPE(1,1,K).EQ.40)RC1(K)=EDRC1

c  Determine the first order rate constant for the sorbed
phase.


        IF (TYPE(1,1,K).EQ.0)RC2(K)=UDRC2
        IF (TYPE(1,1,K).EQ.1)RC2(K)=WARC2
        IF (TYPE(1,1,K).EQ.2)RC2(K)=LARC2
        IF (TYPE(1,1,K).EQ.3)RC2(K)=SIRC2
        IF (TYPE(1,1,K).EQ.4)RC2(K)=SARC2
        IF (TYPE(1,1,K).EQ.5)RC2(K)=GRRC2
        IF (TYPE(1,1,K).EQ.6)RC2(K)=TRRC2
        IF (TYPE(1,1,K).EQ.7)RC2(K)=UNRC2
        IF (TYPE(1,1,K).EQ.21)RC2(K)=SA1RC2
        IF (TYPE(1,1,K).EQ.22)RC2(K)=SA2RC2
        IF (TYPE(1,1,K).EQ.23)RC2(K)=SA3RC2
        IF (TYPE(1,1,K).EQ.24)RC2(K)=SA4NRC2
        IF (TYPE(1,1,K).EQ.25)RC2(K)=SA5ERC2
        IF (TYPE(1,1,K).EQ.26)RC2(K)=SA5ARC2
        IF (TYPE(1,1,K).EQ.27)RC2(K)=SA5WRC2
        IF (TYPE(1,1,K).EQ.28)RC2(K)=SA4SRC2
        IF (TYPE(1,1,K).EQ.29)RC2(K)=PLCRC2
        IF (TYPE(1,1,K).EQ.30)RC2(K)=RDRC2
        IF (TYPE(1,1,K).EQ.40)RC2(K)=EDRC2
      END DO

c  Produce output to reaction package input file, RCT.INP.

      PRINT *,'Writing to reaction input file RCT.INP.'

150 FORMAT (I10,F10.0,10X,A20,I10)
      WRITE (14,100)ISOTHM,IREACT
100 FORMAT(2I10)
      IF(ISOTHM.GT.0) THEN
        WRITE (14,150) IREAD,CNSTNT,FMTIN,IPRN
        WRITE(14,110)(RHOB(K),K=1,NLAY)
        WRITE (14,150) IREAD,CNSTNT,FMTIN,IPRN
        WRITE(14,110)(SP1(K),K=1,NLAY)
        WRITE (14,150) IREAD,CNSTNT,FMTIN,IPRN
        WRITE(14,110)(SP2(K),K=1,NLAY)
      ENDIF
      IF(IREACT.GT.0) THEN
        WRITE (14,150) IREAD,CNSTNT,FMTIN,IPRN
        WRITE(14,110)(RC1(K),K=1,NLAY)
        WRITE (14,150) IREAD,CNSTNT,FMTIN,IPRN
        WRITE(14,110)(RC2(K),K=1,NLAY)
      ENDIF
110 FORMAT(10F8.2)
      RETURN
      END
```

# Appendix E  - Base Case Parameter Values

| Parameter | Area Applied | Water Balance Model | Particle Tracking Model | Regional Flow Model | 2-D Leachate Mounding Model | 3-D Flow Model | Contaminant Transport Model |
|---|---|---|---|---|---|---|---|
| Mound height (ft, COP) | Applied to silt | NA | NA | 25 | NA | Results from 2DLMM* | NA |
| Chloride Source Conc. (mg/l) | Landfill | NA | NA | NA | NA | NA | 1000 |
| Regional recharge (in./yr) | Non-water area outside landfill | NA | NA | Variable, based on USGS. Ave =8.3 | NA | 8.0 | NA |
| Local Recharge (in/yr) | Landfill | Variable, 10 to 1 | NA | NA | Variable, 10 to 1 | NA | NA |
| Vertical Conductivity $K_z$, (cm/sec) | Silt outside landfill area | NA | NA | $9\times10^{-6}$ | NA | $2\times10^{-6}$ | NA |
| | Silt in landfill area | $1\times10^{-6}$ | $1\times10^{-6}$ | $1\times10^{-6}$ | $1\times10^{-6}$ | $1\times10^{-6}$ | NA |
| | Sand | NA | NA | $3.175\times10^{-3}$ | NA | $3.175\times10^{-3}$ | NA |
| | Gravel | NA | NA | $6.35\times10^{-3}$ | NA | $6.35\times10^{-3}$ | NA |
| | Troutdale | NA | NA | $1.27\times10^{-3}$ | NA | $1.27\times10^{-3}$ | NA |
| Horizontal Conductivity $K_x$, $K_y$ (cm/sec) | Silt outside landfill area | NA | NA | $9\times10^{-5}$ | NA | $9\times10^{-5}$ | NA |
| | Silt in landfill area | $9\times10^{-5}$ | $9\times10^{-5}$ | $9\times10^{-5}$ | $9\times10^{-5}$ | $9\times10^{-5}$ | NA |
| | Sand | NA | NA | $3.175\times10^{-2}$ | NA | $3.175\times10^{-2}$ | NA |
| | Gravel | NA | NA | $6.35\times10^{-2}$ | NA | $6.35\times10^{-2}$ | NA |
| | Troutdale | NA | NA | $1.27\times10^{-2}$ | NA | $1.27\times10^{-2}$ | NA |
| Storage Coefficient | Silt | NA | NA | $1\times10^{-4}$ × unit thickness | NA | $1.0\times10^{-6}$ | NA |
| | Sand | NA | NA | $1\times10^{-4}$ × unit thickness | NA | $1.0\times10^{-6}$ | NA |
| | Gravel | NA | NA | $1\times10^{-4}$ × unit thickness | NA | $1.0\times10^{-6}$ | NA |
| | Troutdale | .NA | NA | $1\times10^{-4}$ × unit thickness | NA | $1.0\times10^{-6}$ | NA |
| Specific Yield | Silt | NA | NA | 0.08 | NA | 0.08 | NA |
| | Landfill | 0.20 | 0.20 | NA | 0.20 | NA | NA |
| | Sand | NA | NA | 0.20 | NA | 0.20 | NA |
| | Gravel | NA | NA | 0.15 | NA | 0.15 | NA |
| | Troutdale | NA | NA | 0.15 | NA | 0.15 | NA |
| Dispersivity (ft) | Silt | NA | NA | NA | NA | NA | 1 |
| | Sand | NA | NA | NA | NA | NA | 1 |
| | Gravel | NA | NA | NA | NA | NA | 5 |
| | Troutdale | NA | NA | NA | NA | NA | 5 |
| Effective Porosity | Silt | NA | 0.40 | NA | NA | NA | 0.40 |
| | Sand | NA | 0.25 | NA | NA | NA | 0.25 |
| | Gravel | NA | 0.25 | NA | NA | NA | 0.25 |
| | Troutdale | NA | 0.25 | NA | NA | NA | 0.25 |

| Parameter | Area Applied | Water Balance Model | Particle Tracking Model | Regional Flow Model | 2-D Leachate Mounding Model | 3-D Flow Model | Contaminant Transport Model |
|---|---|---|---|---|---|---|---|
| CR/WR** Conductance (ft²/day) | Silt | NA | NA | 50,000 | NA | NA | NA |
| | Sand | NA | NA | 50,000 | NA | NA | NA |
| | Gravel below WR** | NA | NA | 50,000 | NA | NA | NA |
| CR/WR Water Levels | | NA | NA | From Army Corp Data | NA | NA | NA |
| Slough Levels (ft, COP) | Pre 1982 | 10.2 | 10.2 | From Army Corp Data | 10.2 | From Army Corp Data | From Army Corp Data |
| | 1983-1986 | 9.74 | 9.74 | From Army Corp Data | 9.74 | From Army Corp Data | From Army Corp Data |
| | Post 1986 | 9.74 | 9.74 | From Army Corp Data | 9.74 | From Army Corp Data | From Army Corp Data |
| Lake Levels (ft, COP) | Pre 1982 | 10.2 | 10.2 | From Army Corp Data | 10.2 | From Army Corp Data | From Army Corp Data |
| | 1983-1986 | 11.74 | 11.74 | From Army Corp Data | 11.74 | From Army Corp Data | From Army Corp Data |
| | Post 1986 | 11.74 | 11.74 | From Army Corp Data | 11.74 | From Army Corp Data | From Army Corp Data |

*2DLMM = Two-Dimensional Leachate Mounding Model

**CR/WR = Columbia and Willamette Rivers

| Parameter | Area Applied | Water Balance Model | Particle Tracking Model | Regional Flow Model | 2-D Leachate Mounding Model | 3-D Flow Model | Contaminant Transport Model |
|---|---|---|---|---|---|---|---|
| Boundary Conditions | Top of silt | NA | NA | Water table | NA | Water table | NA |
| | Top of landfill | Leachate mound | Leachate mound | Constant head | Leachate mound | Constant head | Fixed concentration |
| | Lateral silt away from NPR*** | NA | NA | Head dependent flux | NA | No flow | No concentration |
| | Lateral silt at NPR*** | NA | NA | No flow | NA | NA | NA |
| | Top of Gravel | Fixed head (11.64 to 12.1 ft) | Fixed head (11.64 to 12.1 ft) | NA | Fixed head (11.64 to 12.1 ft) | NA | NA |
| | Lateral sand away from NPR*** | NA | NA | Head dependent flux | NA | Given head (predicted from regional flow model) | No concentration |
| | Lateral sand at NPR*** | NA | NA | No flow | NA | NA | NA |
| | Lateral gravel away from NPR*** | NA | NA | Head dependent flux | NA | Fixed head (predicted from regional flow model) | No concentration |
| | Lateral gravel at NPR*** | NA | NA | No flow | NA | NA | NA |
| | Lateral Troutdale away from NPR*** | NA | NA | Head dependent flux | NA | Fixed head (predicted from regional flow model) | No concentration |
| | Lateral Troutdale at NPR*** | NA | NA | No flow | NA | NA | NA |
| | Bottom of Troutdale | NA | NA | Prescribed flow as predicted by USGS Portland basin model | NA | Fixed head (predicted from regional flow model) | No concentration |

***NPR = North Portland Road

357