This is to certify that the

dissertation entitled

Recharge to Discharge Groundwater
Travel Times in the Michigan Basin
and the Effect of Glacial Ice Loading

presented by

John R. Hoaglund III

has been accepted towards fulfillment
of the requirements for
Doctor of Philosophy degree in Geological Sciences

Major professor

Date August 9, 1996

RECHARGE TO DISCHARGE GROUNDWATER TRAVEL TIMES IN THE MICHIGAN BASIN AND THE EFFECT OF GLACIAL ICE LOADING

By

John R. Hoaglund III

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree

DOCTOR OF PHILOSOPHY

Department of Geological Sciences

1996

ABSTRACT

RECHARGE TO DISCHARGE GROUNDWATER TRAVEL TIMES IN THE MICHIGAN BASIN AND THE EFFECT OF GLACIAL ICE LOADING

By

John R. Hoaglund III

Numerical modelling of regional recharge to discharge advective groundwater travel times in the Michigan basin indicates that groundwater residence times are generally much less than 10,000 years for the glacial drift aquifer and greater than 20,000 years for bedrock aquifers. The RAND3D particle tracking solute transport model used the steady state U.S. Geological Survey RASA Modflow flow model. The flow model showed that groundwater heads and flows in the glacial aquifer are controlled by local stream stages and discharges, resulting in localized flow cells accounting for over 90% of the overall model water budget. As a result, the number and extent of flowpaths, from the water table, greater than 10,000 years is insufficient to transport isotopically changing recharge of the Younger Dryas and subsequent warming climatic events to produce isotopically light groundwater anomalies observed in the Saginaw lowlands. Since the Younger Dryas was the last climatic event cold enough to produce the lightest isotopes from recharge, another mechanism is implied. Given the short residence times of shallow groundwater, deep penetration for long duration is required for the emplacement of the light isotopic groundwater mass if the mass is to endure 10,000 years of subsequent discharge. An alternative hypothesis is the reversing of groundwater flow by the loading of glacial meltwater, either directly by the ice or indirectly from high proglacial lake stands.

Groundwater modelling of the Port Huron glacial advance using Modflow and the existing RASA model showed that the effect of ice loading is localized to the region of the ice sheet where groundwater flow is reversed and affects the bedrock aquifers, with a strong downward component of flow beneath the icesheet and an upward component of flow into a large proglacial lake. The modelling shows that given the assumptions in the model, ice loading is a very effective mechanism for introducing light isotopic water from the icesheet deep into the aquifer systems.

To my wife, Rose Bohn,
who believed when no one else did
that we would realize this dream.

"The heart has reasons that reason cannot know."
-Blaise Pascal
In memory of Henner Lerch (May 1, 1963 -- January 19, 1995)
"Shine on you crazy diamond."

ACKNOWLEDGMENTS

I would like to thank Grahame J. Larson for his guidance and friendship, and for introducing me to this complex and interesting problem relevant to my own interests in computer modelling. Research of this magnitude is unlimited and his patience was essential as I learned to focus. I gratefully acknowledge the enriching contributions of my committee members: Grahame J. Larson (advisor and committee chairman), David Long, David Wiggert, and F. William Cambray. I would also like to thank guest committee members Duncan Sibley and Norman Grannemann for invaluable discussions about Michigan geological and geochemical enigmas. I must thank David Westjohn, the master of the Michigan basin, for solving all of the enigmas and relating what little my brain could absorb, structural geologist to structural geologist wanna be.

Financial support for this research was provided by a teaching assistantship, the U.S.G.S., and my wife, Rose Bohn. I would like to thank the students that I was privileged to instruct. Teaching is a two way street and I hope they gained as much from me as I gained from them. I owe a huge thanks to Norm Grannemann for my employment support from the U.S.G.S. and his modelling mentoring. I deeply appreciate the love and support from Rose, who sacrificed many of her own creative pursuits so that we could survive in an era of recession, government shutdowns, and corporate downsizings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1.1 Introduction: The Groundwater Isotopic Anomalies of the Saginaw and Michigan Lowlands

Oxygen-18 and Deuterium depleted groundwater exists in modern groundwater discharge areas (Figure 1) of the Saginaw and Michigan lowlands (Mandle and Westjohn, 1989). This groundwater also plots (Wahrer, 1993; Meissner, 1993; Ging et. al., 1996) on the meteoric water line of Craig (1961). The anomalously light stable isotopic, and isotopically meteoric, signature of the water currently discharging in the Saginaw and Michigan lowlands suggests that the water may represent climatically colder meteoric water that recharged the groundwater system near the end of the last Late Wisconsinan glaciation, approximately 8,000 years ago (Dannemiller and Badalementi, 1988; Long et. al., 1988). Anomalous del-D (δD) and del-18 ($\delta^{18}O$) range from -137.4 to -75.0 ‰ and -18.8 to -11.0 ‰ respectively in the glacial drift in the lowland areas (Wahrer, 1993).

Table 1 shows the observed ranges of isotopes for three hydrogeologic units within the Michigan Basin. The hydrogeologic units of Table 1 show light isotopic anomalies that

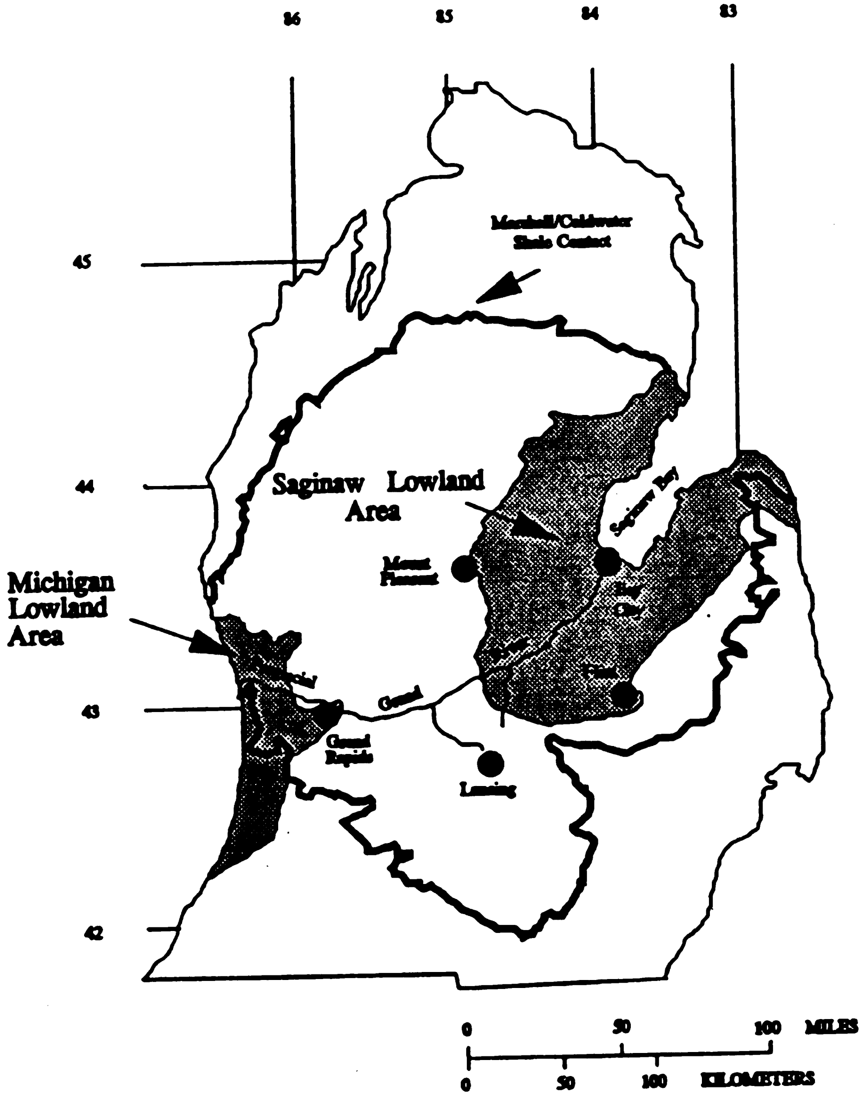

Figure 1. Map showing the location of the Saginaw Lowland Area (modified from Sun, R.J., and Johnston, R.H. in press). From Wahrer, 1993.

Table 1

Hydrogeologic Unit	Range δD ‰	Range $\delta^{18}O$ ‰	Source
Glacial Drift	-137.0 to -56.0	-18.8 to -8.3	Wahrer, 1993
Pennsylvanian	-131.5 to -53.5	-18.19 to -7.8	Meissner, 1993
Marshall	-120 to -50	-17.0 to -8.5	Ging et. al., 1996

extend to the Marshall aquifer (subcrop extent shown on Figure 1) above the Coldwater confining unit (Figure 2). Plots and distribution maps of stable isotopic data of δD and $\delta^{18}O$ for these units are shown for the glacial drift aquifers (Figures 3 and 4 from Wahrer, 1993), the Pennsylvanian aquifers (Figures 5 and 6 from Meissner, 1993), and the Marshall (Mississippian) aquifer (Figures 7 and 8 from Ging et. al., 1996). Fresh and saline groundwater in the Michigan basin plot on the meteoric water line of Craig (1961) to depths as great as the fresh and saline portions of the Marshall sandstone (Figure 2) within the Michigan Basin (Long et. al., 1988; 1991). Below the interface of the saline water and the brine, the isotopic composition deviates from the meteoric water line toward heavier $\delta^{18}O$. Above the interface, the meteoric water is progressively lighter toward the Saginaw lowlands (Figure 4, 6, and 8).

Groundwater in the Saginaw lowlands is commonly saline (Houghton, 1838; Lane, 1899b; and Twenter, 1966) which seemingly contradicts the fact that it is isotopically light and meteoric. A transition zone from fresh water (<1,000 mg/L TDS) through saline water (1,000 mg/L --100,000 mg/L TDS) to brine (>100,000 mg/L TDS) exists within the Pennsylvanian Grand River-Saginaw Aquifer and the Mississipian Marshall Aquifer (Mandle and Westjohn, 1989). Recent geochemical investigations concluded that most water molecules within the transition zone are of meteoric origin while the solute molecules in the zone are derived from dilution of the brine (Long et. al., 1988). Therefore, the transition zone corresponds to a "mixing" zone of meteoric water and brine to produce the saline, meteoric water.

Era	Period	Epoch	Stratigraphic Unit		Hydrogeologic Unit
Cenozoic	Quaternary	Pleistocene Holocene			Glacial-drift aquifers
Cenozoic	Quaternary	Pleistocene Holocene			Glacial till-red beds confining unit
Mesozoic	Jurassic	Late	Unnamed red beds		Glacial till-red beds confining unit
Paleozoic	Pennsylvanian	Middle	Grand River Formation		Grand River-Saginaw aquifer
Paleozoic	Pennsylvanian	Middle / Early	Saginaw Formation		Grand River-Saginaw aquifer
Paleozoic	Pennsylvanian	Early	Saginaw Formation		Saginaw confining unit
Paleozoic	Pennsylvanian	Early	Saginaw Formation	Parma Sandstone	Parma-Bayport aquifer
Paleozoic	Mississippian	Late	Grand Rapids Group: Bayport Limestone		Parma-Bayport aquifer
Paleozoic	Mississippian	Late	Grand Rapids Group: Michigan Formation		Michigan confining unit
Paleozoic	Mississippian	Late	Grand Rapids Group: Michigan Formation	Stray Sandstone	Marshall aquifer
Paleozoic	Mississippian	Early	Marshall Sandstone	Napoleon Sandstone	Marshall aquifer
Paleozoic	Mississippian	Early	Coldwater Shale		Coldwater confining unit

Figure 2. Stratigraphic column and hydrogeologic units in the study area (Modified from Mandle and Westjohn, 1989; stratigraphic column modified from Michigan Geological Survey, 1964, Chart 1). From Meissner, 1993.

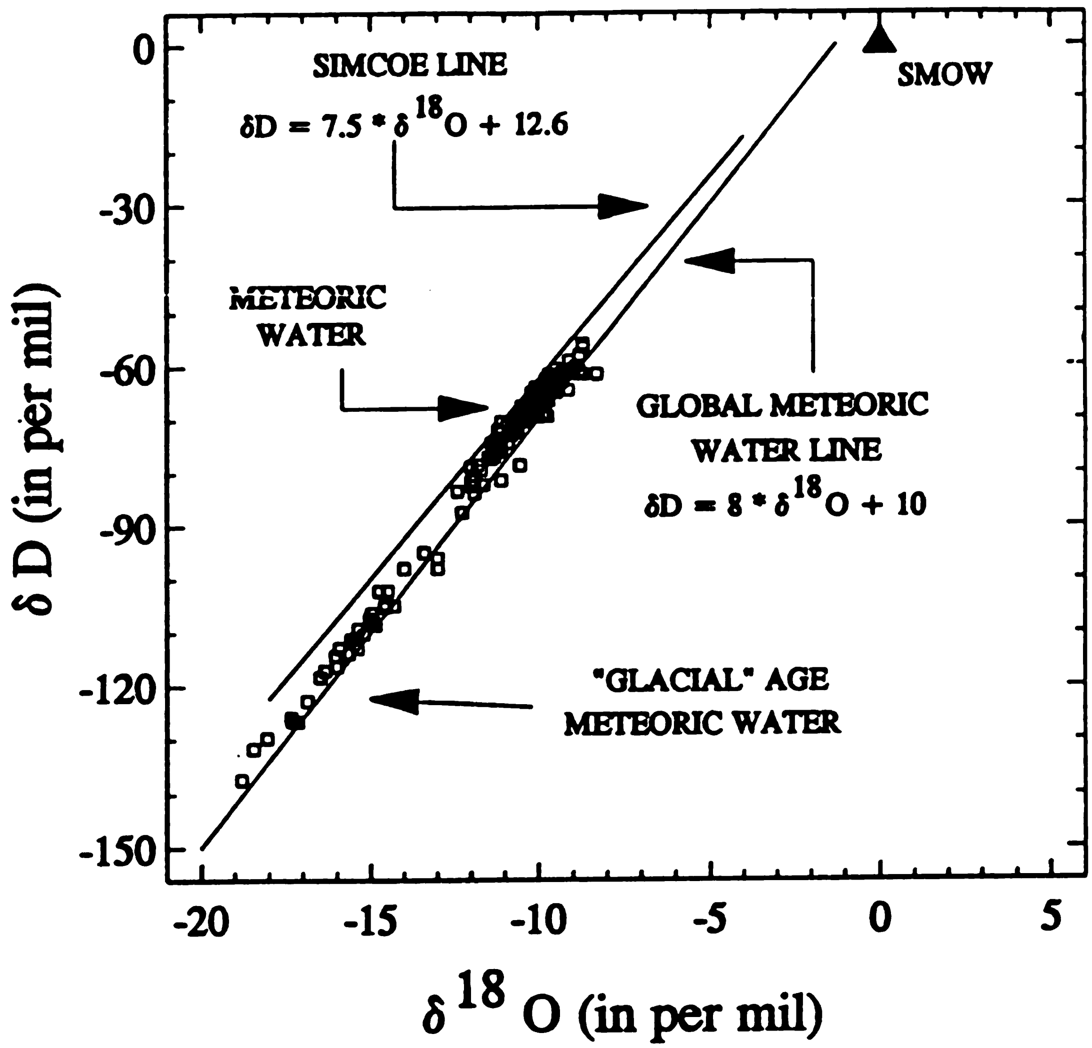

Figure 3. Plot showing the relationship between δD and $\delta^{18}O$ for water from the glacial drift aquifer. The composition of oxygen and hydrogen is reported in terms of the difference of the $^{18}O/^{16}O$ and $D/^{1}H$ ratios relative to Standard Mean Ocean Water (SMOW). The δD and $\delta^{18}O$ values of SMOW are zero. From Wahrer, 1993.

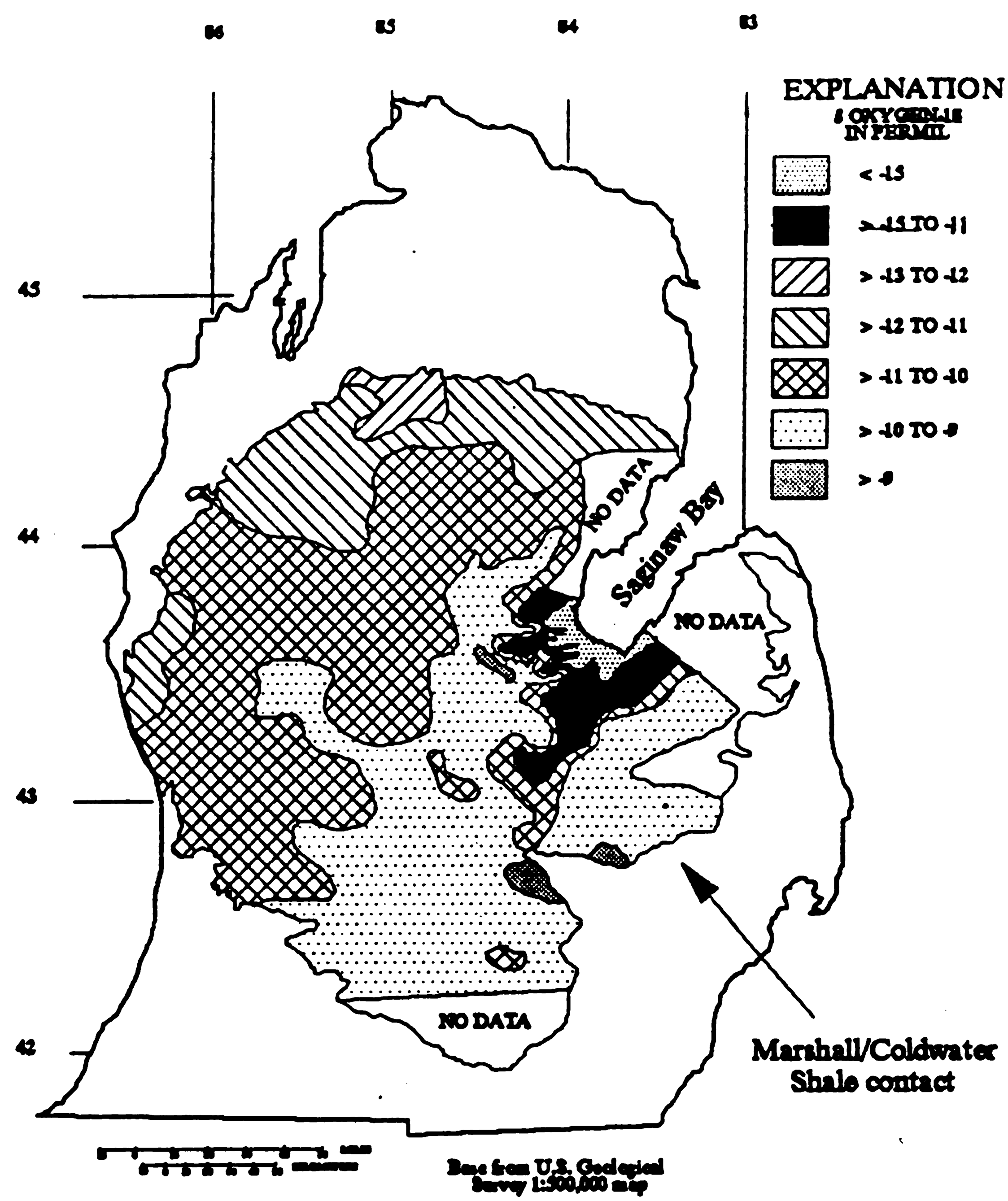

Figure 4. Variation of $\delta^{18}O$ in water from the glacial drift aquifer in the lower Peninsula of Michigan (from Wahrer et. al., 1993). From Wahrer, 1993.

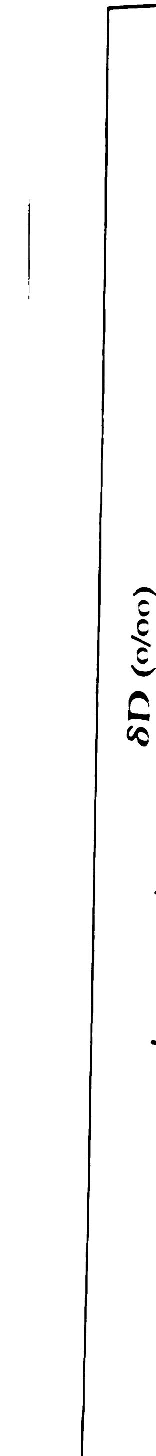

Figure
Saginaw

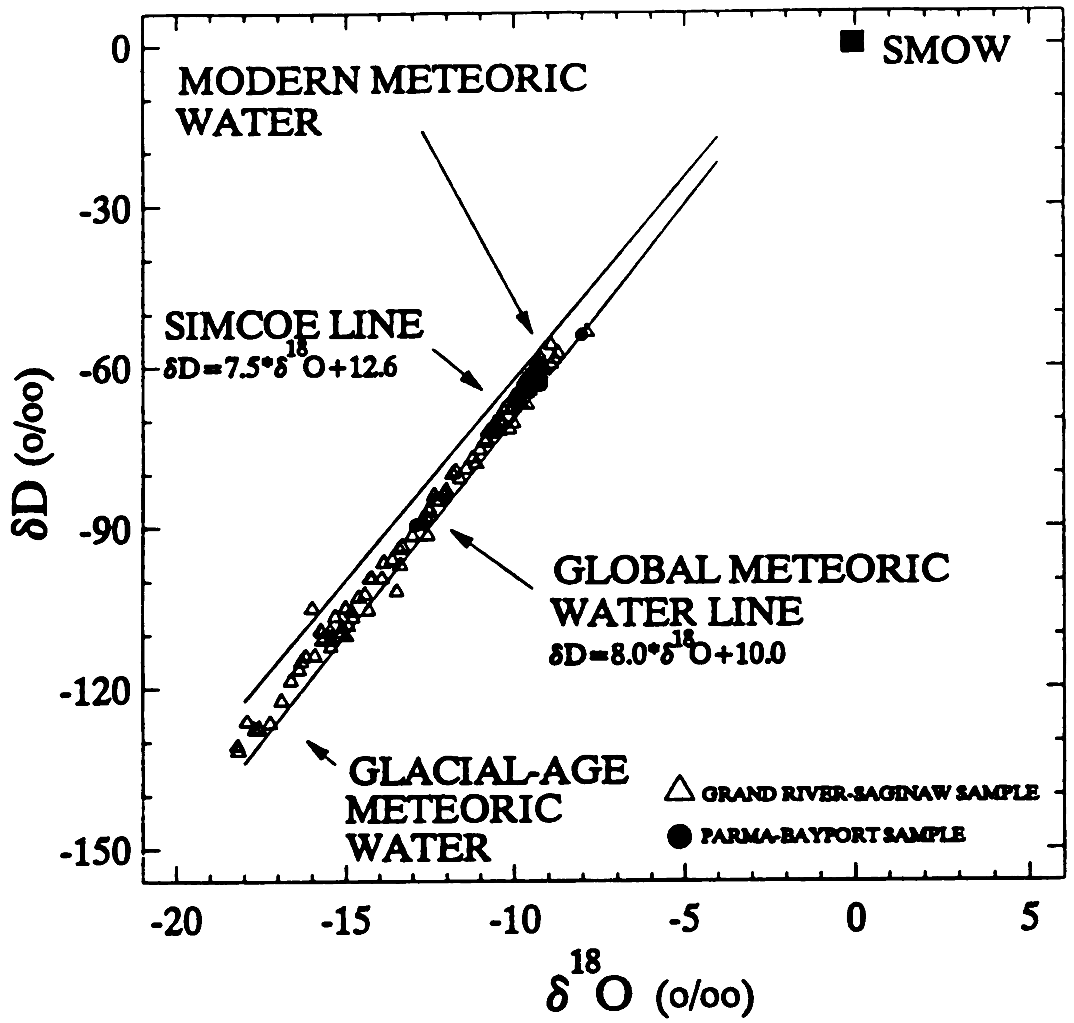

Figure 5. δD versus $\delta^{18}O$ for ground water from Grand River-Saginaw and Parma-Bayport aquifers. (From Meissner, 1993).

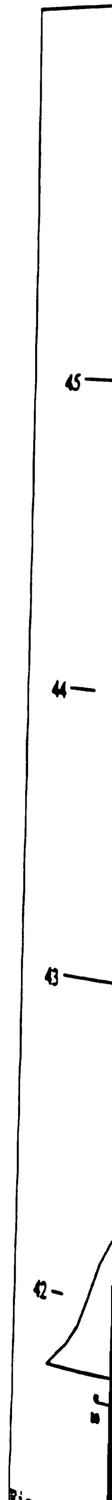

Figure 6.
the Grand

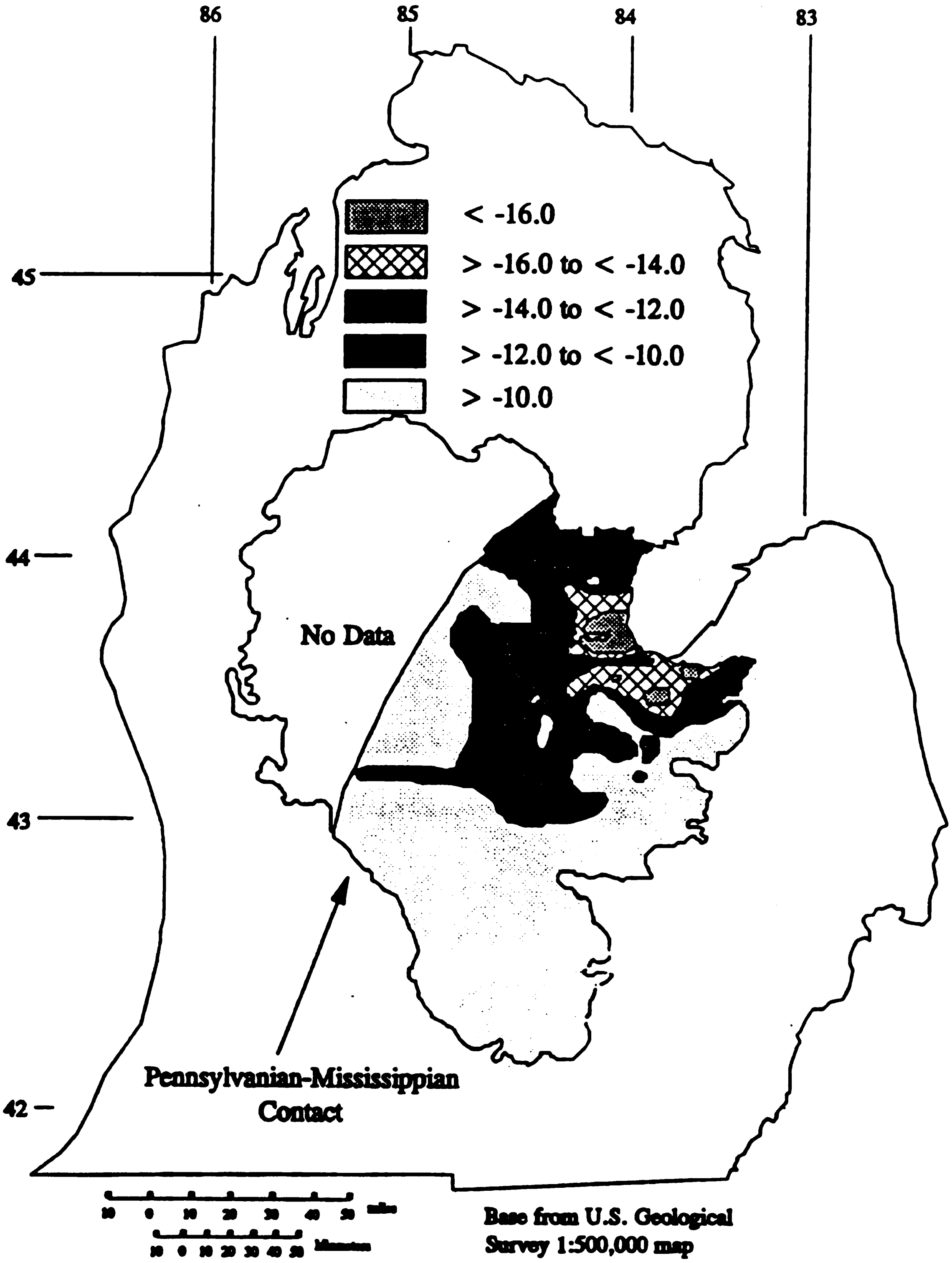

Figure 6. $\delta^{18}O$ distribution for ground water sampled from the Grand River-Saginaw aquifer. (From Meissner, 1993).


DELTA DEUTERIUM, IN PERMIL

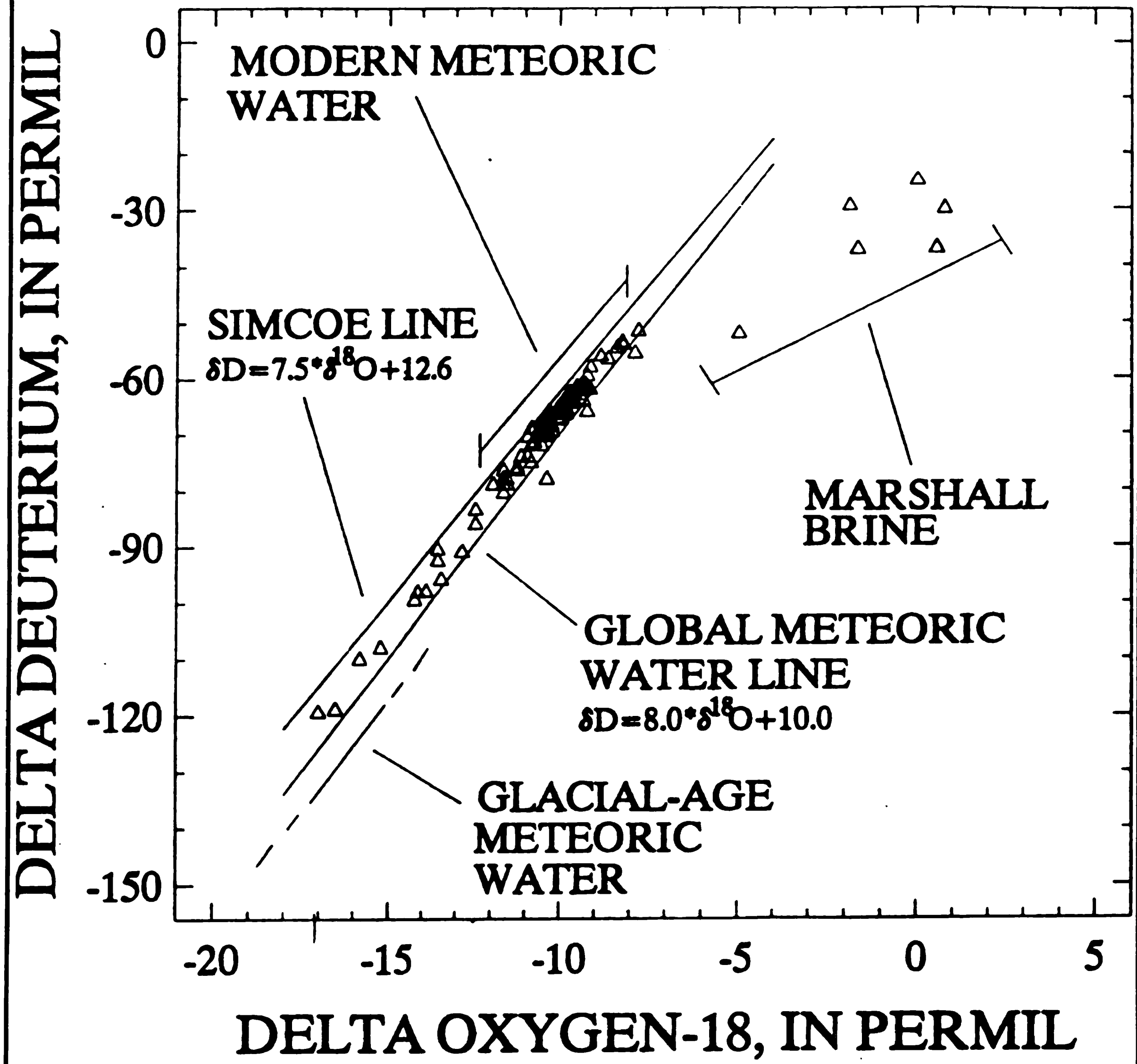

Figure 7. Relation between δD and $\delta^{18}O$ for ground water from the Marshall aquifer. (From Ging et. al., 1996).

Figure 8. δ
Ging et. al, 1

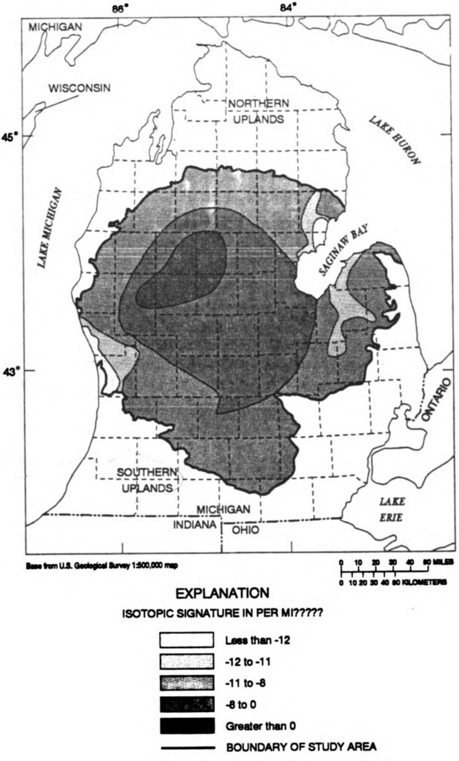

Figure 8. $\delta^{18}O$ distribution for ground water sampled in the Marshall aquifer. (From Ging et. al, 1996).

The m

above

Toth. I

Geolog

2), the

system

the sys

areas w

northern

from th

(Mandie

1.2 Hyp

A genera

isotopes i

1.2.1 The

Long et. al

recharge w

model to ex

The meteoric isotopic composition of the discharging water, together with its salinity (see above), conceptually suggests a hydrodynamic flow system of flow cell(s) (Hubbert, 1940; Toth, 1962, 1963) from surface recharge, to the Marshall sandstone, to surface discharge. Geologically, this groundwater flow system lies above the Coldwater shale aquiclude (Figure 2), the leakance of which is presumed to be small enough to isolate the groundwater flow system from deeper groundwater masses within the Michigan Basin. Groundwater recharges the system (Figure 9) in two areas of relatively high water table generally coincident with areas where the bedrock is above the level of the Great Lakes (Graf et. al. 1966): the northern and southern upland areas (Mandle and Westjohn, 1989). Groundwater discharges from the system (Figure 1) in two lowland areas: the Saginaw and Michigan lowlands (Mandle and Westjohn, 1989; Wahrer, 1993).

1.2 Hypotheses for the Observed Groundwater Isotopic Anomalies

A general advection hypothesis can be proposed to explain the observed pattern of stable isotopes in the discharge zones.

1.2.1 The Advection Hypothesis

Long et. al.(1988) have suggested that climatic changes in the stable isotopic composition of recharge water may be reflected in the groundwater of the Saginaw Lowlands. A conceptual model to explain the saline, isotopically light, and isotopically meteoric water in the Saginaw

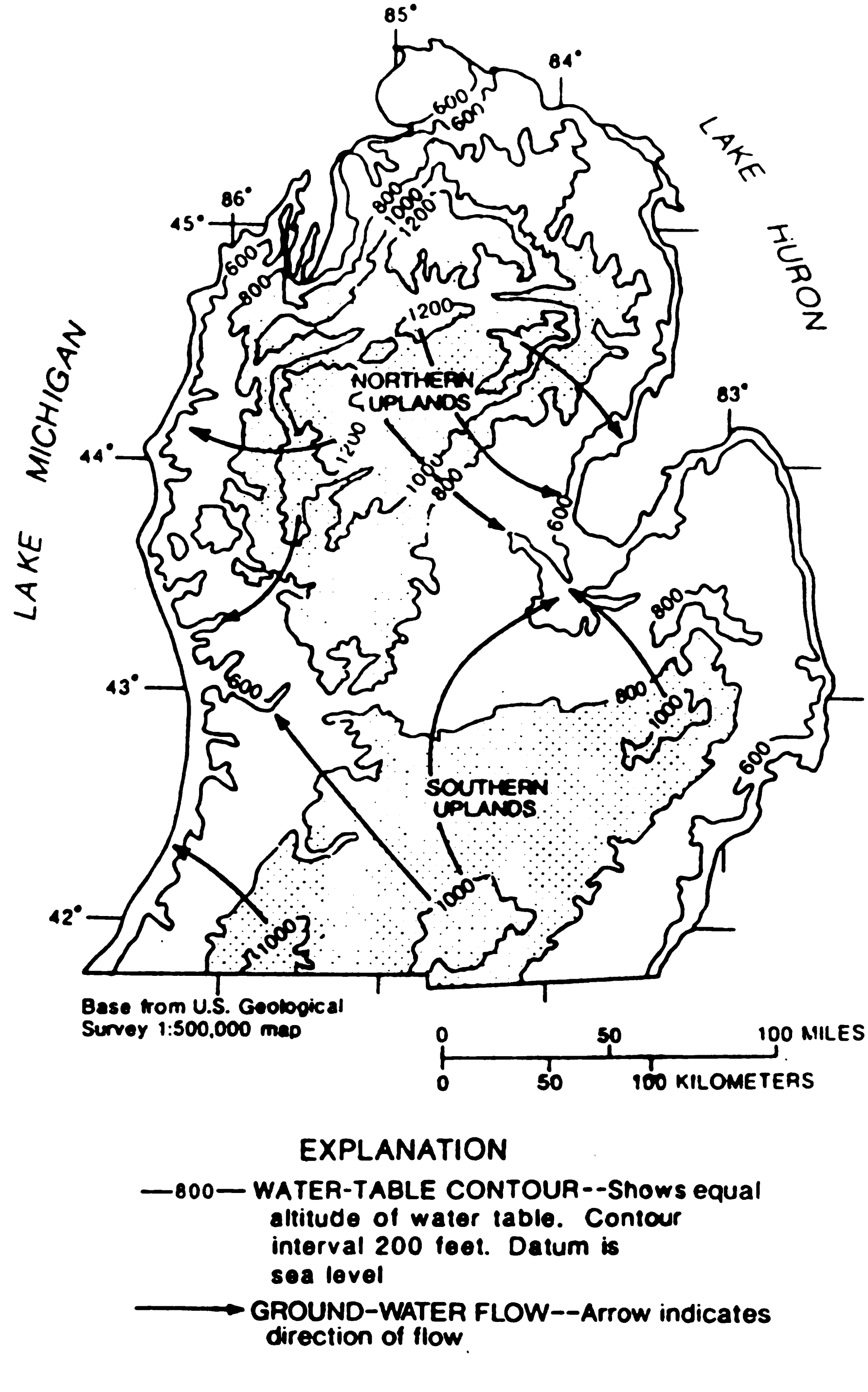

Figure 9. Map showing water table and generalized ground-water-flow directions for the Glacial-drift aquifer in the Lower Peninsula of Michigan (Modified from Mandle and Westjohn, 1989). From Wahrer, 1993.

and M

flow sy

et al., 1

to post

uplands

1993) a

sandston

into mete

1988). 4

(Long et

The hypot

which can

1.) The ligh

this system

of the Sagi

complete r

shallow gro

implies that

southern upl

within 14,00

and Michigan lowlands is as follows: 1.) meteoric water recharges the shallow groundwater flow system in the northern and southern upland areas (Mandle and Westjohn, 1989; Long et. al., 1988). Meteoric water climatically changes isotopic composition with time in response to post-Pleistocene global warming and recharges the system in the northern and southern uplands coincident with modern recharge, 2.) regional groundwater flow paths (Wahrer, 1993) advect meteoric water down to the mixing zone, possibly as deep as the Marshall sandstone, toward upwardly advecting or diffusing brine, 3.) solute from the brine diffuses into meteoric water producing saline water with a meteoric isotopic signature (Long et. al., 1988), 4.) saline water upwardly advects and discharges with its meteoric isotopic signature (Long et. al., 1988).

The hypothesis, stated above as a conceptual model, has several hydrogeologic implications which can be summarized as follows:

1.) The light isotopic signature of the discharge water implies that the age of oldest water in this system is greater than 8,000 years (Long et. al., 1988). Since the modern discharge areas of the Saginaw and Michigan lowlands were covered by glacial ice 14,000 years ago, the complete recharge to discharge cycling (Toth, 1962, 1963) of groundwater through the shallow groundwater flow system must have occurred within the last 14,000 years. This implies that groundwater flows from modern regional recharge areas of the northern and southern uplands to modern regional discharge areas of the Saginaw and Michigan lowlands within 14,000 years.

2.) The meteoric signature implies the mass conservation of the deuterium/protium and oxygen isotopes throughout the recharge to discharge flow cycle. The effects of dispersion, retardation, and mixing are minimal.

3.) The salinity of the discharge water implies that the minimum depth of this shallow groundwater flow system is the mixing zone, locally in the shallow Marshall aquifer or deeper Saginaw aquifer.

1.2.2 Loading Hypotheses

Another hypothesis to explain the isotopically light and meteoric water in the discharge area is the loading of glacial meltwater, either directly by the ice or indirectly from high proglacial lake stands. By either mechanism, the groundwater flow system is at least partially reversed as either an ice/meltwater or proglacial lake hydrostatic load occupies the modern discharge zone. The timing and extent of ice advances and/or proglacial lakes are critical to the loading hypothesis. Though earlier glacial events are known for the area from early Pleistocene 730,000 years ago up to Wisconsinan glaciations 100,000 years ago, Late Wisconsinan drift, from 21,000 years ago to 10,000 years ago, contains the earliest known lakes in the Great Lakes basins (Eschman, 1985).

1.3 St

To test

model wa

and Harb

and Lonn

Michigan

(Hoaglunc

recharge o

isotopic co

1.3 Statement of Procedure

To test the hypothesis of the conceptual advective model, a numerical flow and transport model was constructed. A finite difference, numerical flow model (MODFLOW: McDonald and Harbaugh, 1988) and coupled particle tracking transport model (Random Walk: Prickett and Lonnquist, 1971; Prickett et. al., 1981; Koch and Prickett, 1989) was setup for the Michigan Basin domain from the land surface to the base of the Marshall sandstone (Hoaglund et. al., (in press); Appendix V). The steady state flow simulation, with constant recharge of water (energy), was coupled with a transient particle tracking simulation with the isotopic composition of the recharge water climatically varied through 14,000 years time.

2.1 The
and assu

Porous m

balanced

2.1.1 Da

Darcy's lav

hydraulic p

domain is k

classic pape

is analogou

Darcy's law

CHAPTER 2

2.1 The Flow Model: Development of the governing equation for groundwater flow and assumptions

Porous media groundwater flow is governed by Darcy's law and continuity conditions balanced by the water budget from recharge and storage.

2.1.1 Darcy's law

Darcy's law relates the specific discharge (groundwater flux) linearly to the gradient of the hydraulic potential. The elevation to which the water will rise from any point in the spatial domain is known as hydraulic head. Hydraulic head has been related to fluid potential in a classic paper by Hubbert (1940). As a spatially independent potential variable, hydraulic head is analogous to temperature in thermal diffusion and voltage in electrical conductance. Darcy's law in the x direction is written:

$$q = -K\frac{dh}{dx}$$

The propo

thermal d

electrical c

Since the gro

and since the

units of perm

equipotential

results:

2.1.2 Continui

If a unit cube i

requires that any

consideration is kn

The proportionality constant, K, is known as the conductivity or permeability (analogous to thermal diffusivity in thermal diffusion and conductance as the reciprocal of resistance in electrical conductivity). It is related to the intrinsic permeability, k, by the following:

$$K=\frac{\rho g}{\mu}k$$

Since the groundwater flux can be thought of as a line length of water delivered in a unit time, and since the gradient is dimensionless as a change in head over a change in spatial length, the units of permeability, K, are length per time. Also, if groundwater flux, q, is multiplied by an equipotential surface area (at right angles to the flux), a volume per unit time (discharge Q) results:

$$qA=Q$$

2.1.2 Continuity

If a unit cube is drawn with sides parallel to the x, y, and z axes, conservation of mass requires that any flux into the cube must equal the flux out of the cube. This geometrical consideration is known mathematically as divergence (net outflux) of a vector field and results in the f

Note th

independ

gradient.

homogen

reduces t

2.1.3 Stor

Theis (193

conductance

S_s, is define

hydraulic hea

volume of aq

in the following:

$$\frac{\partial(K_x\frac{\partial h}{\partial x})}{\partial x}+\frac{\partial(K_y\frac{\partial h}{\partial y})}{\partial y}+\frac{\partial(K_z\frac{\partial h}{\partial z})}{\partial z}=0$$

Note that the divergence operator (the second partial derivatives with respects to the independent variables x, y and z) operates on the directional permeability terms as well as the gradient, both of which are functions of the independent variables. If isotropic and homogeneous conditions for the hydraulic conductivity are assumed, the preceding equation reduces to the Laplace equation:

$$\frac{\partial^2 h}{\partial x^2}+\frac{\partial^2 h}{\partial y^2}=0$$

2.1.3 Storage

Theis (1935) defined groundwater storage analogously to capacitance in electrical conductance and heat capacity in thermal diffusion. In three dimensions, the specific storage, S_s, is defined as the volume of water released from a volume of aquifer per unit decline in hydraulic head. The concept of storage enables time dependent changes in head as the unit volume of aquifer obtains, retains, or releases water in storage. Continuity requires that a flux into the

in hydr

equation

2.1.4 A
Dimensio

A source

equation i

A common s

and correspo

dimensional

stacked, con

dimensional f

into the volume be balanced by either a flux out of the volume or a time dependent change in hydraulic head dependent upon the storage coefficient. This results in the diffusion equation:

$$\frac{\partial(K_x\frac{\partial h}{\partial x})}{\partial x^2}+\frac{\partial(K_y\frac{\partial h}{\partial y})}{\partial y^2}+\frac{\partial(K_z\frac{\partial h}{\partial z})}{\partial z^2}=S_s\frac{\partial h}{\partial t}$$

2.1.4 Areal Recharge/Leakance and the Reduction of the Diffusion Equation to Two Dimensions: the governing equations for quasi 3D confined and unconfined groundwater flow

A source term--areal recharge, R, per thickness of aquifer, b--completes the diffusion equation in three dimensions.

$$\frac{\partial(K_x\frac{\partial h}{\partial x})}{\partial x^2}+\frac{\partial(K_y\frac{\partial h}{\partial y})}{\partial y^2}+\frac{\partial(K_z\frac{\partial h}{\partial z})}{\partial z^2}=S_s\frac{\partial h}{\partial t}+\frac{R}{b}$$

A common simplification of hydrogeological problems is to assume that the vertical gradient and corresponding vertical flow within an aquifer layer is zero resulting in horizontal, two-dimensional flow within the layer. In quasi 3D or layer 3D modelling, these layers can be stacked, communicating flow vertically through a leakance term. Horizontal, two-dimensional flow with vertical leakance is usually justified based on the layered nature of aquif

flux

comm

incorp

quasi 3

vertica

horizor

geomet

In the c

above an

S_r are m

The isotropi

aquifers as well as very high horizontal dimension to vertical dimension ratios. The leakance flux is determined as a Darcy function of the difference in head between the two communicating layers controlled by their vertical hydraulic conductivities. The leakance is incorporated into the diffusion equation as a source term similar to the recharge term. In quasi 3D modelling, the medium can be horizontally isotropic or anisotropic independent of vertical isotropy or anisotropy. To simplify the equations the permeabilities are assumed horizontally isotropic and vertically anisotropic implying that $K_x = K_y = K \neq K_z$. Two geometric cases for the layers must be considered, confined and unconfined.

In the confined case, the aquifer with thickness b is confined between impermeable layers above and below with hydraulic head above the top of the aquifer. The coefficients, K and S_s, are multiplied by this thickness resulting in T (Transmissivity) and S (Storativity).

$$T=Kb$$

$$S=S_s b$$

The isotropic heterogeneous form of the diffusion equation in two dimensions is:

In the

flow a

aquifer

by the

an h-de

coefficie

Replacing

an equatio

is known a

equation i

$$\frac{\partial(T\frac{\partial h}{\partial x})}{\partial x}+\frac{\partial(T\frac{\partial h}{\partial y})}{\partial y}=S\frac{\partial h}{\partial t}+R$$

In the unconfined case, the top of the aquifer is bound by the water table. With horizontal flow assumed, this water level corresponds to the hydraulic head at any point in the saturated aquifer. These conditions are called the Dupuit assumptions. The coefficient K is multiplied by the saturated thickness (ie. the water level, h, above the base of the aquifer) resulting in an h-dependent, two-dimensional transmissivity. The two dimensional, unconfined storage coefficient is not distinguished from the three dimensional, unconfined storage coefficient.

$$T=Kh$$

Replacing the transmissivity coefficient in the two-dimensional diffusion equation results in an equation that describes the two-dimensional flow in an unconfined aquifer. This equation is known as the Boussinesq equation (Bear, 1979). The isotropic heterogeneous Boussinesq equation is:

$$\frac{\partial(Kh\frac{\partial h}{\partial x})}{\partial x}+\frac{\partial(Kh\frac{\partial h}{\partial y})}{\partial y}=S\frac{\partial h}{\partial t}+R$$

2.

Th

(ra

in s

det

con

vert

How

hydra

regio

A fur

perme

ellipsoi

strata (

of bedd

directio

anisotro

MODFL

neighbori

2.1.5 Structure and Anisotropy

The Michigan Basin is a structural as well as a stratigraphic basin with both regional dip (inclination of bedding) toward the basin center and dip related to local folding. Analysis of in situ aquifer tests in a horizontal plane and of lab tests in the plane of bedding were used to determine transmissivities and/or hydraulic conductivities corresponding to the principal conductivity directions in the plane of bedding. Analysis of lab tests were used to determined vertical hydraulic conductivities in a direction perpendicular to the plane of bedding. However, at the scale of the Michigan Basin, it was uncertain as to whether or not the hydraulic conductivity tensor needed to be rotated--using the angle of dip--into the plane of regional and/or local-fold-controlled bedding.

A function relating vertical permeability to bedding and bedding-orthogonal principal permeabilities as a function of structural dip, θ, was derived. An hydraulic conductivity ellipsoid is shown in Figure 10 with two equal principal conductivities in the plane of dipping strata (K_h in directions "a") and an unequal principal conductivity perpendicular to the plane of bedding (K_v in direction b). The condition shown is "horizontally" isotropic--one principal direction in the direction of strike and the other in the direction of dip--and vertically anisotropic. The figure also shows stratally-centered nodes (true nodes) of the type used in MODFLOW and the apparent mapped positions as they would appear on a grid. Eight neighboring true nodes, four in each stratum, are connected as they would be related in finite

$$V = \frac{K_h K_v}{\sqrt{K_v^2 \sin^2\theta + K_h^2 \cos^2\theta}}$$

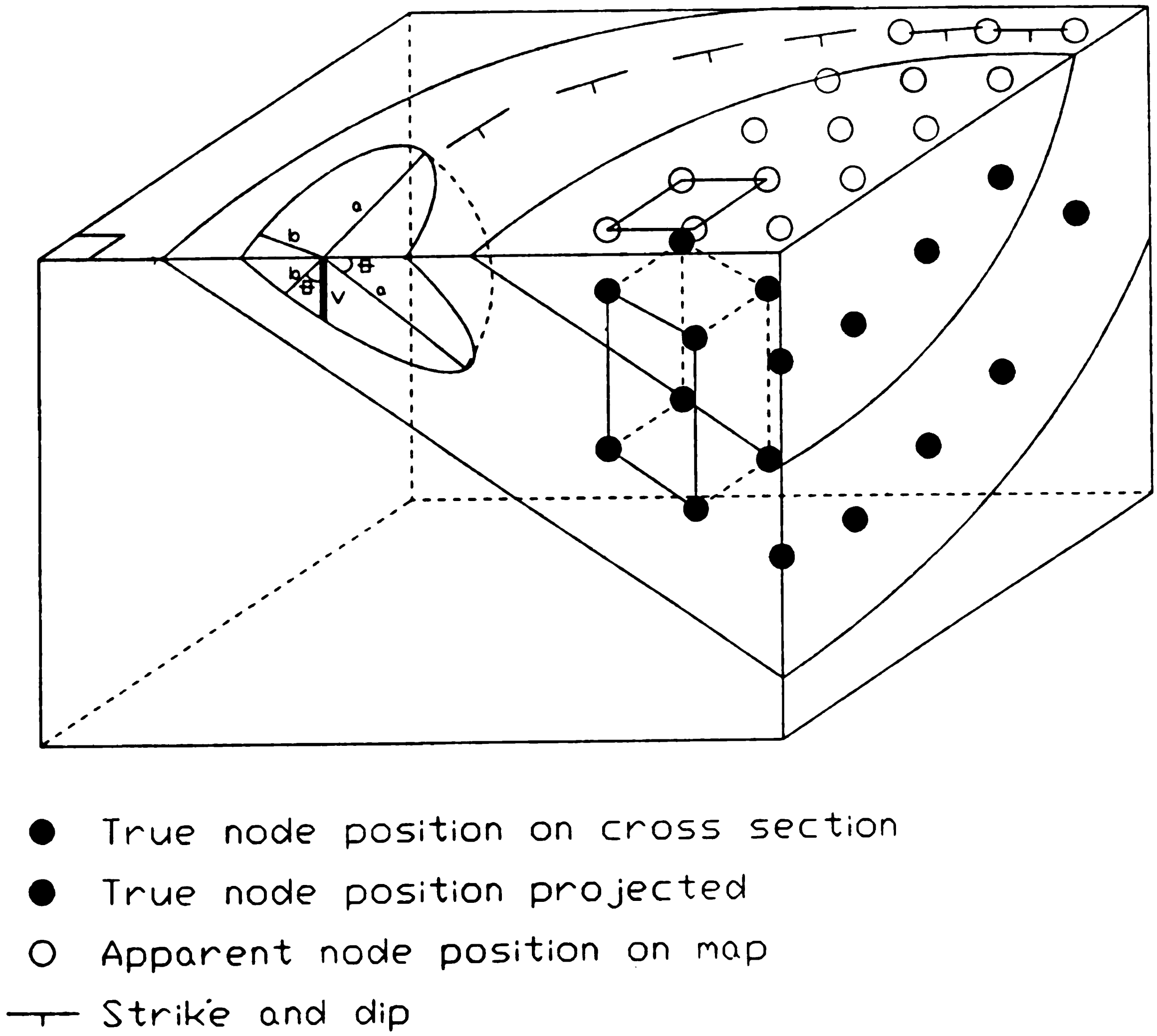

Figure 10. Block diagram showing relationship between
1) true (stratally centered) and apparent node positions,
2) strike and dip of bedding,
3) a-direction of bedding parallel hydraulic conductivity,
4) b-direction of bedding perpendicular hydraulic conductivity, and
5) V-direction of vertical hydraulic conductivity between nodes.
Eight nodes are connected in a prism in the a- and V-directions as they would be related in finite difference approximations. The map or apparent positions of the eight nodes are connected in a square.

differe

connec

connect

the dire

the prin

The dip

stratigrap

conductivi

Figure 11

hydraulic co

one (1). Th

for large dip

difference approximations between them. The four nodes connected within a stratum are connected in the principal directions within the plane of bedding. However, the four nodes connected between strata are connected vertically in the direction "V" of the ellipsoid, not in the direction perpendicular to bedding. The vertical hydraulic conductivity, V, is related to the principal conductivies, K_h and K_v, and the dip, θ, by

$$V=\frac{K_h K_v}{\sqrt{K_v^2 \sin^2\theta + K_h^2 \cos^2\theta}}$$

The dip can be calculated at each node using the aquifer top arrays assuming that this stratigraphic top is structurally controlled. V can then be calculated using the hydraulic conductivity arrays. V can be used in the calculation of VCONT in MODFLOW.

Figure 11 is a graph of the relation for V through 90 degrees of dip using a horizontal hydraulic conductivity, K_h, of 10 and a bedding perpendicular hydraulic conductivity, K_v, of one (1). The graph shows that the correction for 10:1 K_h to K_v ratios is insignificant except for large dips.

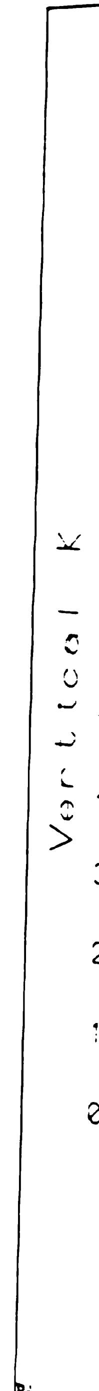

Figure
conducti
paralle
perpendi

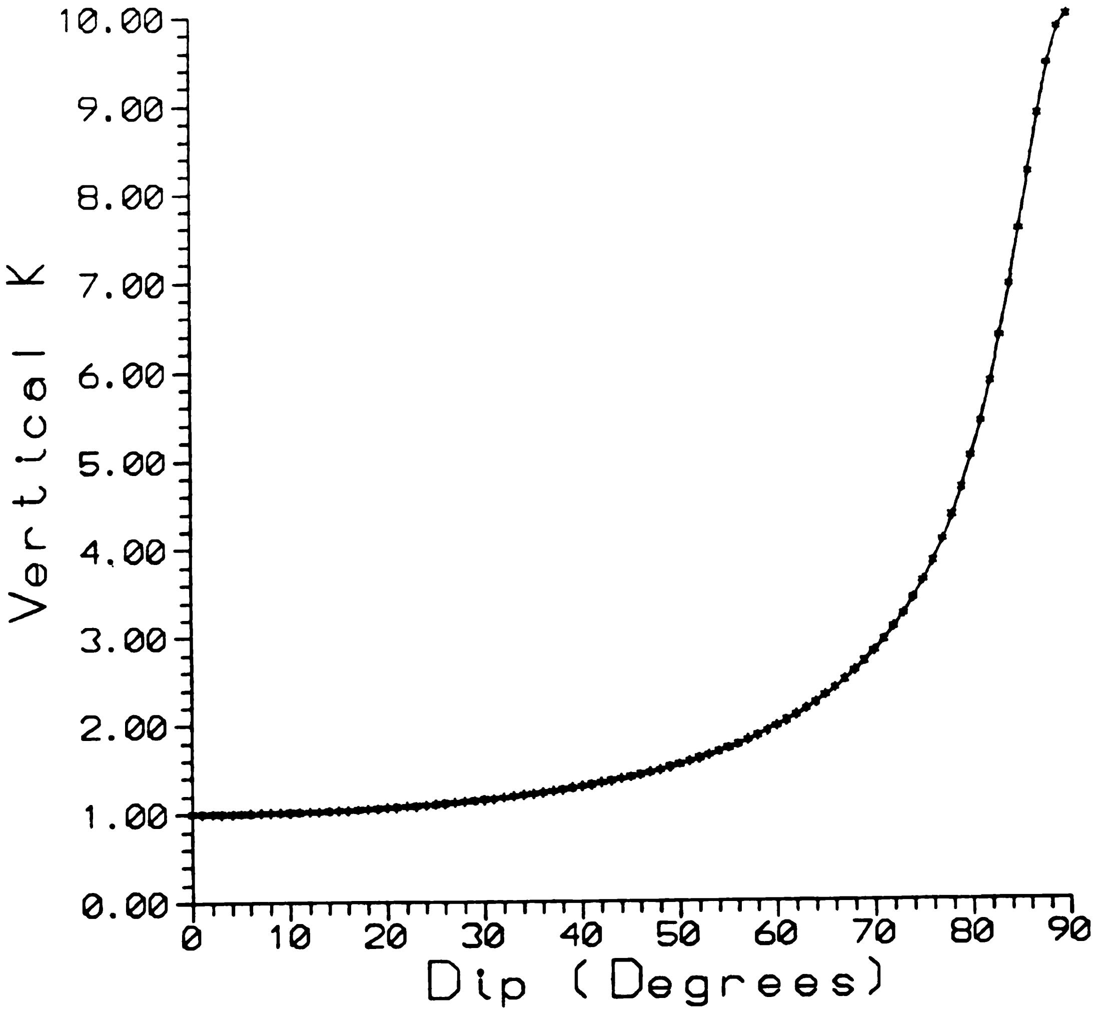

Figure 11. Graph of the relation for vertical hydraulic conductivity through 90 degrees of dip using a bedding parallel hydraulic conductivity of 10 and a bedding perpendicular hydraulic conductivity of 1.

2.2 T

As pa

MOD

Basin

Append

other th

to deter

in sectio

used to

2.2 The Flow Model: Parameters and Domain/Boundary Conditions

As part of their Regional Aquifer Systems Analysis (RASA), the USGS completed a MODFLOW (McDonald and Harbough, 1988) steady state flow model for the Michigan Basin using 4 layers with three interlayered confining units (Hoaglund et. al., (in press); Appendix V). The first layer, the glacial layer, was bound laterally by the Great Lakes. The other three bedrock layers were bound by their respective subcrops. This solution was used to determine the advective groundwater flow field for the solute transport results discussed in section 5.1 of this dissertation. The model domain and modified boundary conditions were used to simulate the hydraulic effect of ice loading in section 5.4.1 of this dissertation.

3.1 Th

Solute t

equation

where "C

hydrodyn

difference

(Bear, 197

dispersion

first partia

solution can

tracking met

with discrete

CHAPTER 3

3.1 The Transport Model: Governing Equation and Solution Technique

Solute transport is governed by the advection-dispersion equation. The advection-dispersion equation in one-dimension is written:

$$D\frac{\partial^2 C}{\partial x^2} - v\frac{\partial C}{\partial x} + \lambda RC = \frac{\partial C}{\partial t}$$

where "C" is the concentration solved in space "x" and time "t", "D" is the coefficient of hydrodynamic dispersion, "λ" is the decay constant, and "R" is retardation. A finite difference solution of the advection-dispersion equation is stable when formulated implicitly (Bear, 1979, p. 259). However, even with advection only (when dispersion is zero) numerical dispersion and overshoot occurs along concentration fronts due to a truncation error in the first partial derivative of concentration (Bear, 1979, p. 261-263). The oscillations in the solution can result in mass balance problems; mass is conserved, not concentration. Particle tracking methods circumvent these problems by solving the concentration solution indirectly with discrete mass particles moved by vector addition of independent advective and dispersive compo

(Prick

in thr

discret

is both

of the

represe

The ad

distribu

random

dispersi

(Bear, 1

of the re

and varia

3.2 The

For the

correspon

Advectio

coupling th

components. The particle tracking was handled by the Random-Walk solute transport model (Prickett and Lonnquist, 1971; Prickett et. al., 1981) modified to interface with MODFLOW in three dimensions (Koch et. al., 1989). In the Random-Walk model, solute mass is discretized into particles free to move independently in space. Vectorial particle movement is both advective and dispersive in a real x-y-z space superimposed on the integer i-j-k grid of the MODFLOW model. The total number and distribution of particles used in a simulation represents the total mass and distribution of solute.

The advection vectors are calculated from the average linear velocity solved from the distribution of groundwater heads from the MODFLOW output. The dispersion vectors are random variables in the Random-Walk model. Analytical solutions of the advection-dispersion equation show that the concentration of solute is normally (guass) distributed (Bear, 1972). Longitudinal and transverse dispersion vectors are calculated from the product of the respective coefficients of hydrodynamic dispersion with random variables of zero means and variances of one.

3.2 The Transport Model: Parameters

For the advection-dispersion equation in three dimensions, the mathematical constants correspond to the coefficient of hydrodynamic dispersion, advection, retardation, and decay. Advection is not an input parameter, but rather is determined from the flow field, thereby coupling the advection-dispersion model to the flow model. Radioactive decay was reduced from t

chemic

isotope

meteori

deuteriu

of the N

(Zachar

transpor

a minim

The onl

Hydroge

atom dis

transient

molecule

line with

to fractio

transport

isotopic

water m

endmem

from the analysis since the model involved transport of stable phases. Retardation and chemical decay was reduced from the analysis assuming deuterium/protium and oxygen isotope interaction with the medium (mineral-water interaction) was not significant. The meteoric isotopic signature of the water from the bedrock units suggests that deuterium/protium and oxygen isotope mass is conserved. In addition, a paragenetic study of the Marshall Sandstone indicated little mineral-water interaction by cement precipitation (Zacharias, 1992). The conservation of isotope mass is critical in justifying a non-reactive transport approach. These studies suggest that retardation and/or decay processes have had a minimal effect on the isotopic composition of the groundwater during transport.

The only remaining transport parameter is the coefficient of hydrodynamic dispersion. Hydrogen bonding of water molecules allows for independent oxygen atom and hydrogen atom dispersion since association of oxygen and hydrogen in individual water molecules is transient. It can therefore theoretically change the isotopic composition of individual water molecules as transport occurs. However, since there is no deviation from the meteoric water line with increasing isotopic mass of either oxygen or hydrogen, dispersion does not appear to fractionate oxygen and hydrogen isotopes. Therefore, on average oxygen and hydrogen transport en masse without "dispersive" mixing of oxygen and hydrogen isotopes of different isotopic mass. Mixing of two endmember waters of different isotopic composition results in water molecules with an isotopic composition that plots on a tie line between the endmembers; if two different meteoric waters mix, the resultant water is meteoric.

Becaus

paramet

Gelhar e

by Krab

Deuteriu

reduced

Because of its scale dependent nature, dispersion is not treated as an invariant physical parameter (Anderson, 1979). Scale dependent values can be assigned using the methods of Gelhar et. al. (1985). For a scale of approximately 1:24,000, a value of 20 meters was used by Krabbenhoft (1991) for a groundwater-lake interaction study involving the transport of Deuterium/Protium and Oxygen isotopes. For the initial investigation, dispersion was reduced from the analysis to test the extent of advective transport.

4.1 Th

4.1.1 S

A recha

transpor

extendin

temperat

the globa

isotopic da

on the ther

models, an

the winter

fractionatio

mentioned.

isotopic com

Isotope mass

CHAPTER 4

4.1 The Loading Function

4.1.1 Section Overview

A recharge isotopic mass input function for the Michigan Basin domain is required for mass transport simulations for the duration of recharge to discharge groundwater flow, a period extending back at least 8,000 years. This chapter will briefly review the thermodynamics of temperature dependent hydrogen and oxygen isotopic fractionation models, and will review the global fractionation models of precipitation empirically derived from modern global isotopic data. Since the fractionation models are a function of temperature, some recent work on the thermal curves derived from stratigraphic isotopic data, semiannual paleoclimatological models, and pollen data will be reviewed as they relate to Michigan. An argument for using the winter semiannual value of these temperature profiles as input into the isotopic fractionation function to derive the isotopic composition of the recharge water will be mentioned. A proceedure for determining the mass of each isotope for a given volume and isotopic composition was derived and is included in Appendix I.

Isotope mass transport simulation requires an isotopic mass input function. The isotopic compos

tempera

curves f

by Webt

by the c

hydroge

the del (

total mas

in a syst

total mas

density c

4.1.2 T

Thermod

of evapo

lighter p

An isoto

defined

reactant

fractiona

composition of the recharge water, expressed as $\delta^{18}O$ and δD, linearly depends on temperature as well as precipitation, latitude, and altitude. Temperature and precipitation curves for the Holocene epoch in the upper midwest have been constrained by pollen curves by Webb and Bryson (1972) as shown in Figure 12. When the temperatures are constrained by the curves, the del (δ) values can be determined and the masses of the stable isotopes of hydrogen and oxygen (not considering ^{17}O) can be calculated from 1.) the ratios derived from the del (δ) values, 2.) stoichiometric consideration of a simple water molecule, and 3.) the total mass of a given volume of water of known isotopic composition. These relations result in a system of four equations and four unknowns. Since the density used to determine the total mass depends on the solution of the masses, the equations are solved iteratively until the density converges (Appendix I).

4.1.2 Thermodynamics of hydrogen and oxygen isotope fractionation

Thermodynamically, the fractionation of stable isotopes results from the physical processes of evaporation and condensation. Evaporation of seawater enriches the vapor phase with the lighter phases, H and ^{16}O, since these phases have higher vapor pressures.

$$H_2O\ (l) \text{ ------> } H_2O\ (g)$$

An isotopic fractionation factor, somewhat analgous to an inverse of a reaction quotient, is defined using the mass ratio of the isotope. The ratio for oxygen, $R_{18}={}^{18}O/{}^{16}O$, in the reactant, H_2O (l), over the ratio for oxygen in the product, H O (g), is the oxygen fractionation factor with a value at 25 °C (Craig and Gordon, 1965) of:

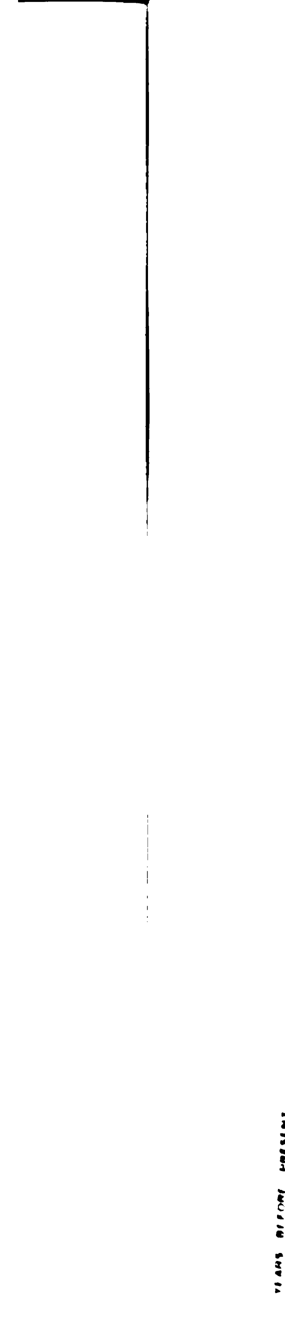

Figure 12
a.) Locatio
Bog (DFB
adapted fro
b.) Plots of
precipitatio
sites of foss

Modified fro

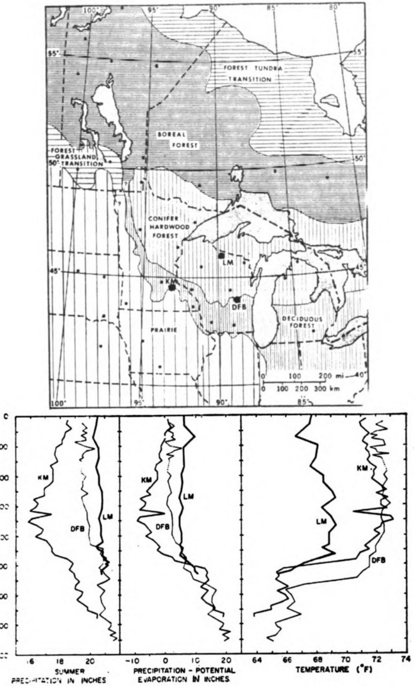

Figure 12

a.) Location of the three fossil pollen sites (large dots)--Lake Mary (LM), Disterhaft Farm Bog (DFB), and Kirchner Marsh (KM)--in their ecologic setting. Vegetation map is adapted from Wright et. al. (1963).

b.) Plots of the reconstructed values of the precipitation during the growing season, the precipitation minus potential evaporation, and the July mean temperature for the three sites of fossil pollen.

Modified from Webb and Bryson, 1972

Similarly

ratio for

(Craig a

The values

enriched

fractionatio

to 1 (Figur

latitude as t

fractionatio

(Faure, 198

fractionatio

Heavy wate

fractionation,

$$\alpha_{18} = \frac{(^{18}O/^{16}O)_l}{(^{18}O/^{16}O)_v} = 1.0092$$

Similarly, the ratio for hydrogen, R_D = D/H where D is Deuterium, in the reactant over the ratio for hydrogen in the product is the hydrogen fractionation factor with a value at 25 °C (Craig and Gordon, 1965) of:

$$\alpha_D = \frac{(D/H)_l}{(D/H)_v} = 1.074$$

The values of the fractionation factors greater than 1.00 indicate that the remaining liquid is enriched in the heavier isotope, ie. the vapor is enriched in the lighter isotope. The fractionation effect decreases with increasing temperature as the fractionation factors decay to 1 (Figure 13; Faure, 1986). This implies that globally, fractionation would increase with latitude as the temperature decreased. Craig and Gordon (1965) showed that the increased fractionation with latitude is even greater than that predicted by evaporative equilibrium (Faure, 1986). The increase in fractionation is due to the additional effect of condensation fractionation.

Heavy water preferentially condenses from the vapor phase over light water resulting in fractionation, specifically the further enrichment of the vapor phase with the lighter isotopes.

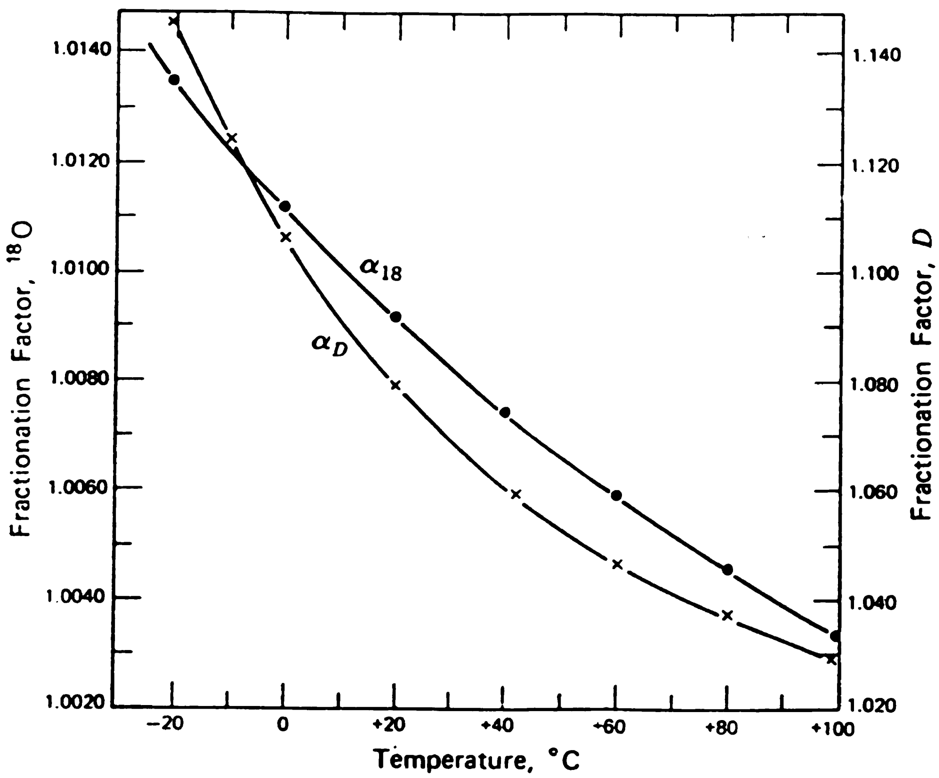

Figure 13. Temperature variation of isotope fractionation factors for evaporation of water. The fractionation factor for ^{18}O (α_{18}) is defined as the ratio of $^{18}O/^{16}O$ in the liquid to $^{18}O/^{16}O$ in water vapor in equilibrium with the liquid. The fractionation factor for D (α_D) is similarly defined as the ratio of D/H in the liquid to D/H in the vapor. The graph illustrates the temperature dependence of these isotopic fractionation factors. Plotted from values listed and discussed by Dansgaard (1964).
From Fauer, 1986

The fr

Oversh

R = rat

R_i = ra

f = the

α = the
vapor,

Since th

"f" is r

other w

isotope

indicate

so the r

H_2O (g) ------> H_2O (l)

The fractionation can be modelled by the Rayleigh distillation equation (Broecker and Oversby, 1971; Faure, 1986):

$$\frac{R}{R_0} = f^{\alpha - 1}$$

R = ratio of the heavy to light isotope in the remaining vapor

R_0 = ratio of the heavy to light isotope in the initial vapor

f = the fraction of vapor remaining

α = the isotope fractionation factor: isotopic ratio of the liquid over the isotopic ratio of the vapor, R_l/R_v

Since the fraction of vapor remaining decreases from 1 (ie. $f < 1$) and alpha is greater than 1, "f" is raised to a positive power so the ratio on the left decreases from 1 (ie. $R/R_0 < 1$). In other words, the model indicates that the remaining vapor is steadily enriched in the lighter isotopes. This effect is also reduced at the higher temperatures of lower latitudes. The model indicates that as alpha decays to 1 with higher temperatures (Figure 13), the power is reduced so the ratio on the left increases.

4.1.3 Global hydrogen and oxygen isotopic fractionation models

The global isotopic composition of meteoric water was first investigated by Craig (1961). Craig (1961) showed a linear relation between $\delta^{18}O$ and δD in meteoric water (Figure 14) according to:

$$\delta D = 8\delta^{18}O + 10$$

Dansgaard (1964) related the $\delta^{18}O$ in precipitation to average annual surface air temperature in degrees centigrade according to:

$$\delta^{18}O = .695T - 13.6$$

However, Dansgaard only looked at $\delta^{18}O$ and δD at coastal sampling stations (Figure 15 and 16). Yurtsever and Gat (1981) presented updated models based upon modern global observations (including inland stations) involving temperature. They refined the linear models by incrementally adding independent variables including precipitation, latitude, and altitude (Tables 2 and 3; Yurtsever and Gat, 1981).

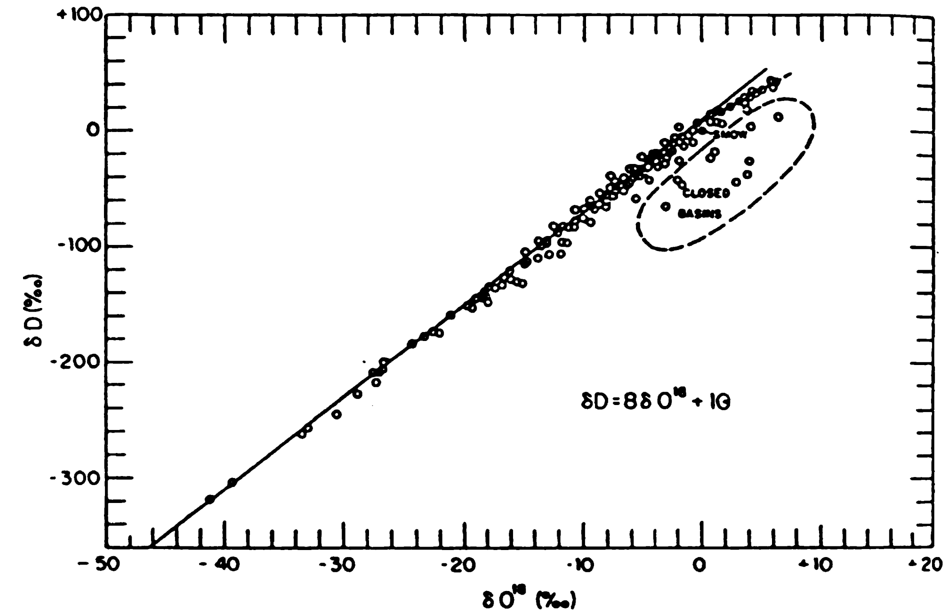

Figure 14. Deuterium and oxygen-18 variations in rivers, lakes, rain, and snow, expressed as per millage enrichments relative to "standard mean ocean water" (SMOW). Points which fit the dashed line at upper end of the curve are rivers and lakes from East Africa.
From Craig, 1961

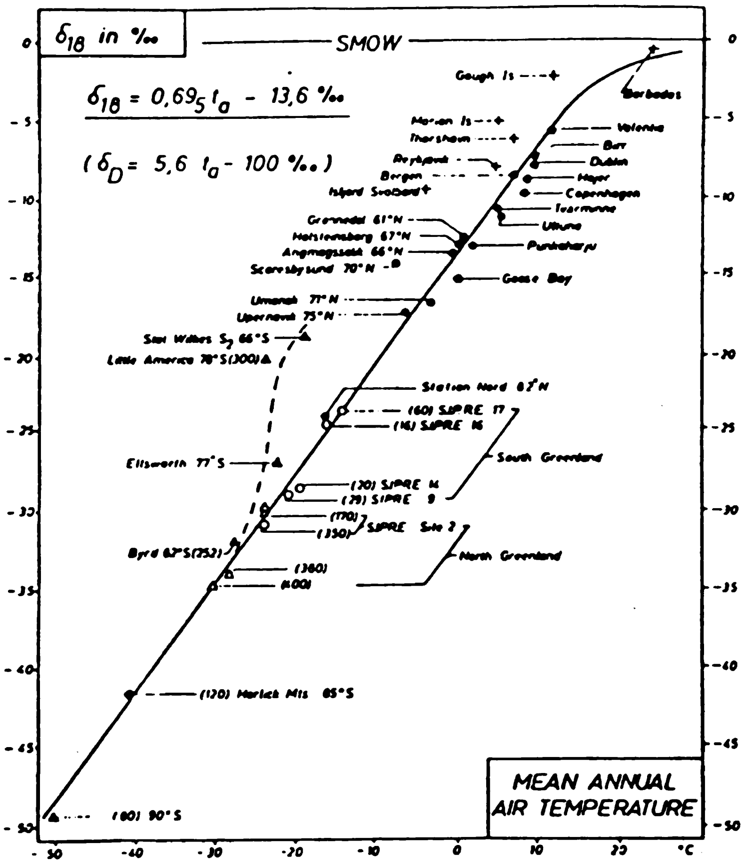

Figure 15. The annual mean $\delta^{18}O$ of precipitation as a function of the annual mean air temperature at surface. The figures in parenthesis indicate the total thickness (in cm) of the investigated snow layers.
From Dansgaard, 1964

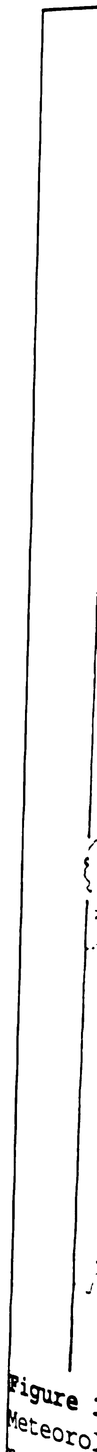

Figure ;
Meteorol
network.
From Dar

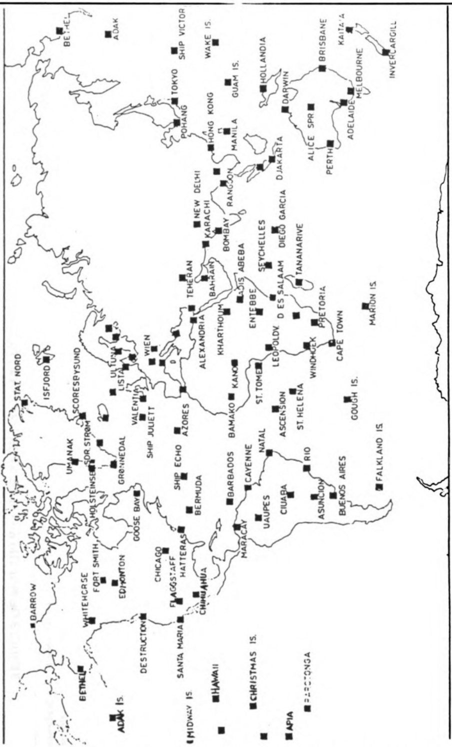

Figure 16. The International Atomic Energy Agency--World Meteorological Organization (I.A.E.A.-W.M.O.) precipitation network.
From Dansgaard, 1964

RESULTS OF MULTIPLE LINEAR REGRESSION ANALYSES

Dependent variable	Independent variables used[a]	n	Regression equation	Multiple correlation coefficient	Standard error of estimate (‰)
Mean $\delta^{18}O$ (‰)	T, P, L, A	91	$\overline{(\delta^{18}O)} = -18.72 + (0.597 \pm 0.088)T - (0.0035 \pm 0.0052)P + (0.106 \pm 0.048)L + (0.0012 \pm 0.0008)A$	0.833	±2.87
	T, P, L	91	$\overline{(\delta^{18}O)} = -16.28 + (0.545 \pm 0.082)T - (0.0067 \pm 0.0048)P + (0.074 \pm 0.044)L$	0.828	±2.90
	T, P	91	$\overline{(\delta^{18}O)} = -11.78 + (0.418 \pm 0.033)T - (0.0084 \pm 0.0048)P$	0.821	±2.92
	T	91	$\overline{(\delta^{18}O)} = -12.18 + (0.390 \pm 0.029)T$	0.815	±2.96
Weighted mean $\delta^{18}O$ (‰)	T, P, L, A	91	$\overline{(\delta^{18}O)}_w = -18.16 + (0.514 \pm 0.086)T + (0.0013 \pm 0.0052)P + (0.098 \pm 0.047)L + (0.0007 \pm 0.0007)A$	0.797	±2.82
	T, P, L	91	$\overline{(\delta^{18}O)}_w = -16.74 + (0.483 \pm 0.080)T - (0.0005 \pm 0.0048)P + (0.080 \pm 0.042)L$	0.795	±2.82
	T, P	91	$\overline{(\delta^{18}O)}_w = -11.88 + (0.345 \pm 0.032)T - (0.0022 \pm 0.0048)P$	0.785	±2.86
	T	91	$\overline{(\delta^{18}O)}_w = -11.99 + (0.338 \pm 0.028)T$	0.785	±2.84

[a] T = average monthly temperature (°C), P = average monthly precipitation (mm), L = latitude (degrees), A = altitude (metres, a.s.l.).

Table 2. From Yurtsever and Gat, 1981

$\delta^{18}O$–δD RELATIONSHIPS FOR ISLAND, COASTAL AND CONTINENTAL STATIONS USING LONG-TERM MEAN VALUES

		r	σ
Island stations	n = 25 $\overline{(\delta D)}_w = (8.47 \pm 0.25)\ \overline{(\delta^{18}O)}_w + (11.11 \pm 1.24)$	0.990	±3.00‰
	n = 25 $\overline{(\delta D)} = (8.51 \pm 0.24)\ \overline{(\delta^{18}O)} + (10.21 \pm 1.04)$	0.991	±2.91‰
Coastal stations	n = 29 $\overline{(\delta D)}_w = (8.07 \pm 0.12)\ \overline{(\delta^{18}O)}_w + (10.44 \pm 1.07)$	0.997	±3.43‰
	n = 29 $\overline{(\delta D)} = (8.03 \pm 0.11)\ \overline{(\delta^{18}O)} + (9.59 \pm 0.95)$	0.997	±3.30‰
Continental stations	n = 15 $\overline{(\delta D)}_w = (8.14 \pm 0.15)\ \overline{(\delta^{18}O)}_w + (9.17 \pm 1.64)$	0.998	±3.08‰
	n = 15 $\overline{(\delta D)} = (8.01 \pm 0.15)\ \overline{(\delta^{18}O)} + (6.49 \pm 1.70)$	0.998	±3.69‰

Table 3. From Yurtsever and Gat, 1981.

4.1.4 I

The m

climato

and sin

the loa

linked

Yurtsev

Given

temper

An 18,0

taken f

fromHo

was tak

that sh

immedia

Paleocli

and Brys

northern

precipita

4.1.4 Paleoclimatology and Michigan

The models of Yurtsever and Gat (1981) can be linked to paleoclimatic thermal and climatological histories. Since paleo-precipitation data is sparse compared to thermal data and since a steady state flow model with constant recharge was used, a simplified model of the loading function involved only temperature and latitude. However, since recharge is linked to precipitation and is fundamental to any flow model, the complete model of Yurtsever and Gat (1981) could be used with assumed or constrained precipitation values. Given the relative magnitudes of the slopes on the Yurtsever and Gat (1981) models, temperature and latitude are the most critical independent variables to constrain.

An 18,000 year to present thermal history for the Greenland icecap is shown in Figure 17 taken from Eddy and Bradley (1991). Eddy and Bradley (1991) modified the curve fromHoughton et. al. (1990) and reference Webb (1991). The principal source of this data was taken from isotope ratios from Greenland ice cores (Eddy and Bradley, 1991), such as that shown in Figure 18 (Faure, 1986). Recent data suggest rapid changes in climate immediately following the Younger Dryas event (Dansgaard et. al., 1989; Alley, et. al., 1993).

Paleoclimate data for Michigan can be related to these thermal histories by pollen data. Webb and Bryson (1972) presented temperature and precipitation curves for three paleo-bogs in the northern midwest (Figure 12). Contoured deviations from annual mean temperatures and precipitation based on pollen are given for the northern hemisphere for mid-Holocene time

Figure 17.
principally i
Temperatur
value at the
From Eddy

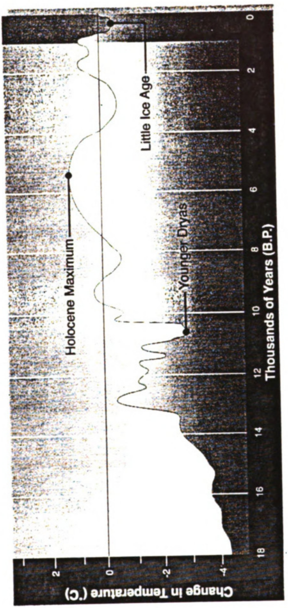

Figure 17. Variations in surface temperature, estimated from a variety of sources, principally isotope ratios from Greenland ice cores for the last 18,000 years. Temperatures shown are in degrees Celsius, in most cases as a departure from the mean value at the turn of the present century of about 15 °C.
From Eddy and Bradley, 1991

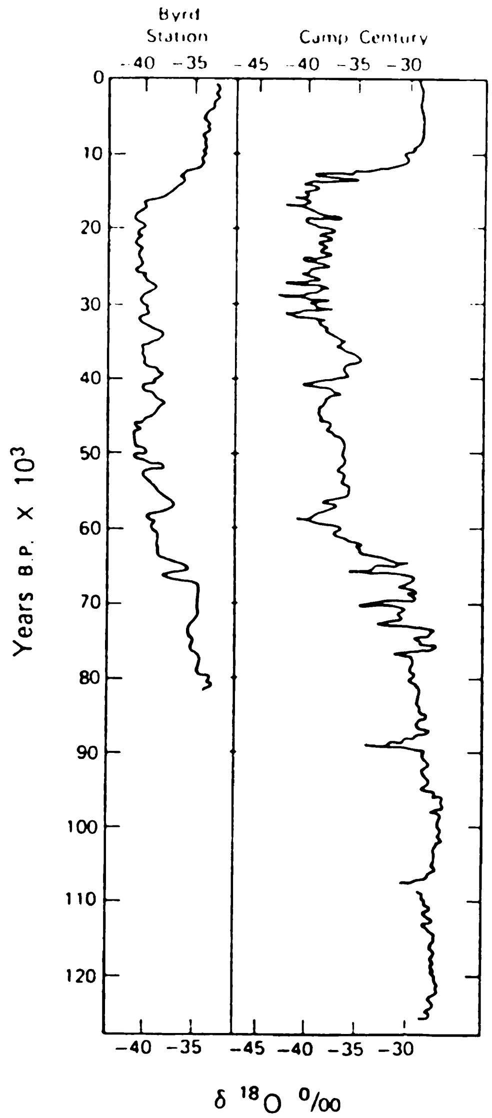

Figure 18. a) Variation in $\delta^{18}O$ in ice cores from Byrd Station, Antarctica, and Camp Century, Greenland. The time scale is based on a theoretical ice-flow model. The more negative $\delta^{18}O$ values from about 70,000 to 12,000 years B.P. in both cores reflect colder climatic conditions during the last ice age.
From Fauer, 1986

in Fr

relatir

COHI

Comn

to sim

18 ka

(1986

simula

compa

semi-a

water

corres

maxim

4.2 Tr

An init

simula

years w

assume

isotopi

ground

in Frenzel et. al., 1992, and Velichko et. al., 1992. Global Circulation Models (GCMs) relating the Greenland ice core data to North America have been constrained by pollen data. COHMAP (1988) used a GCM, the National Center for Atmospheric Research (NCAR) Community Climate Model (CCM), bounded by geologic data and changes in solar radiation to simulate global temperatures, precipitation, and winds, reporting results every 3 ka from 18 ka to present. Kutzbach (1987) summarized the GCM results of Kutzbach and Guetter (1986) for North America for the same time intervals. Webb et. al. (1987) compared model simulations with pollen data. Hyde et. al. (1989) showed that Energy Balance Models (EBM) compared well with the GCM results. Both data and model results report winter and summer semi-annual temperatures. It has been suggested that the isotopic composition of recharge water reflect the winter semi-annual temperature since recharge in temperate climates corresponds to spring melt--after ground thaw and prior to the summer evapotranspiration maximum (Bowser, pers. comm., 1993).

4.2 Transport Initial Condition

An initial isotopic composition (initial condition) must be assumed for the transient transport simulation. If complete flushing of the groundwater flow system occurred within 10,000 years with no mixing, an initial composition equal to the initial recharge composition can be assumed for the whole domain since the final solution would only be affected by subsequent isotopic composition of recharge. A light initial isotopic composition would result if the groundwater flow system were flushed during the last ice advance of the Late Wisconsinan;

the isotop

sheet sourc

the isotopic composition of the groundwater would be isotopically light reflecting the ice sheet source of recharge (melt) water (Craig, 1961).

5.1 Re

To stu

of the

northe

(Koch

of the

PREM

aquife

conduc

4.) aq

bound

or neg

The x

are as

the lay

the pa

CHAPTER 5

5.1 Results of the Transport Simulation

To study isotope transport into the Saginaw lowlands, a 151 row x 161 column sub-domain of the statewide RASA model was extracted, focussing on a region extending from the northern uplands to Saginaw bay (Figure 19), using the PREMOD3D.FOR preprocessor (Koch et. al., 1989). The sub-domain extends from columns 100 to 260 and rows 150 to 300 of the full RASA grid (Figure 19). For each i,j node in each k layer of the sub-domain, PREMOD3D.FOR extracts from the MODFLOW input files and binary head output file 1.) aquifer thickness; 2.) fluxes in the x, y, and z directions which it calculates from model conductances, changes in head, recharge, and grid spacings; 3.) aquifer bottom elevation; and 4.) aquifer top elevation. For each well, river, drain, constant head, and general head boundary, PREMOD3D.FOR computes a sink including x, y and layer k location and positive or negative discharge. Only river and constant head sinks were used in the RASA model. The x and y of any sink corresponds to the center of the node. Constant head and well sinks are assumed to be fully penetrating; the z coordinate of the particle only needs to be within the layer, k, for capture by the sink. River sinks are partially penetrating; the z coordinate of the particle needs to be equal to or greater than the river bottom elevation (stored as the

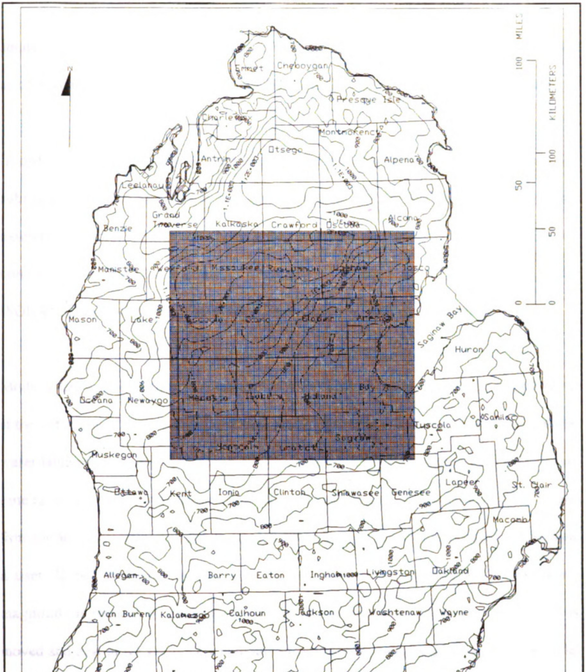

Figure 19. RAND3D 151 row x 161 col particle tracking subdomain of the 470 row x 361 col RASA model. The edges of the full RASA model is shown as a solid border surrounding the lower peninsula. Simulated water table elevations (groundwater potentials for layer 1) are contoured with a 100 foot contour interval.

aquife

all of t

In ord

subrou

extend

compil

ISORA

Model

at the w

water t

time alo

from the

a user

magnitu

moved a

process i

of 100 y

move of

Particles

aquifer top for top-layer sink locations) for capture by the sink. PREMOD3D.FOR writes all of the data to a "filename.RND" input file for RAND3D, the particle transport code.

In order to handle the large sub-domain grid size and to add specific isotope mass input subroutines, the RAND3D code was converted from BASIC to FORTRAN. To utilize extended memory, the FORTRAN code was compiled with an extended memory, 32 bit compiler: Microsoft's Fortran Powerstation. A copy of the FORTRAN code, ISORAN3D.FOR--including the MASSIN subroutine--is included in Appendix I.

Model nodes were chosen in the northern uplands recharge region for isotope mass loading at the water table. The nodes were selected along a groundwater divide in the RASA model water table solution--as far as possible from model river sinks--in order to maximize travel time along a regional groundwater flow path. RAND3D calculates a particle move vector from the advective velocities and a user defined time step. If the vector magnitude exceeds a user defined maximum horizontal move and/or maximum vertical move, the vector magnitude is reduced to the maxima and the time step is fractionalized. The particle is moved and new move vectors are computed from the time remaining in the time step. The process is repeated until the user defined time step is consumed. A particle moving time step of 100 years was used with a maximum horizontal move of 1,000 feet and a maximum vertical move of 10 feet.

Particles were released from the water table at about 10 selected nodes using the continuous loading

various

years, n

model.

case, s

Howev

the ed

To re

single

20,00

comp

of th

a fil

trav

of t

Se

re

S

fl

p

loading function spanning 14,000 years, but all particles discharged to the river sinks at various time intervals. None of the total travel times of any given particle exceeded 2,000 years; most travel times were less than 200 years. To intersect deeper flow paths within the model, the base of the glacial drift (layer 1) was loaded instead of the water table. In this case, some long travel times were encountered when particles entered the bedrock aquifers. However, after the 14,000 year loading was complete, none of the particles had reached even the edges of the Saginaw lowlands.

To record travel times, the ISORAN3D.FOR code was modified to continuously move a single particle until it discharged, moved off of the grid, or until the travel time exceeded 20,000 years. The modifications of the ISORAN3D.FOR code are in Appendix II; the complete code was saved as TRAVELT.FOR. A single particle was released from each node of the 161 x 151 node subdomain. If the particle discharged, the travel time was recorded in a file. If the particle moved off of the grid or if the travel time exceeded 20,000 years, the travel time was recorded in a different file. Of the 161 x 151 = 24,311 nodes, the distribution of the travel times (in 100 year intervals) is shown in Appendix III.

Several particles (693) moved off of the grid as they were loaded on the opposite side of a regional groundwater flow divide delineating flow to the Saginaw lowlands in the model. Several particles (579) had travel times in excess of 20,000 years and were suspended. Their flowpaths are shown on Figure 20 as curves connecting the starting and 20,000-year ending points. Most of these particles (527 or 91%), when suspended, were in bedrock aquifers.

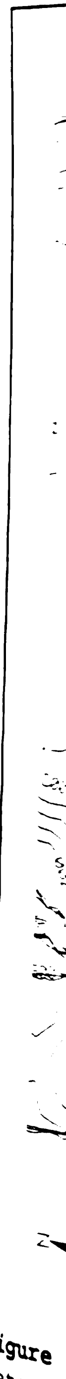

Figure
water ta
years.
depths.
transpor

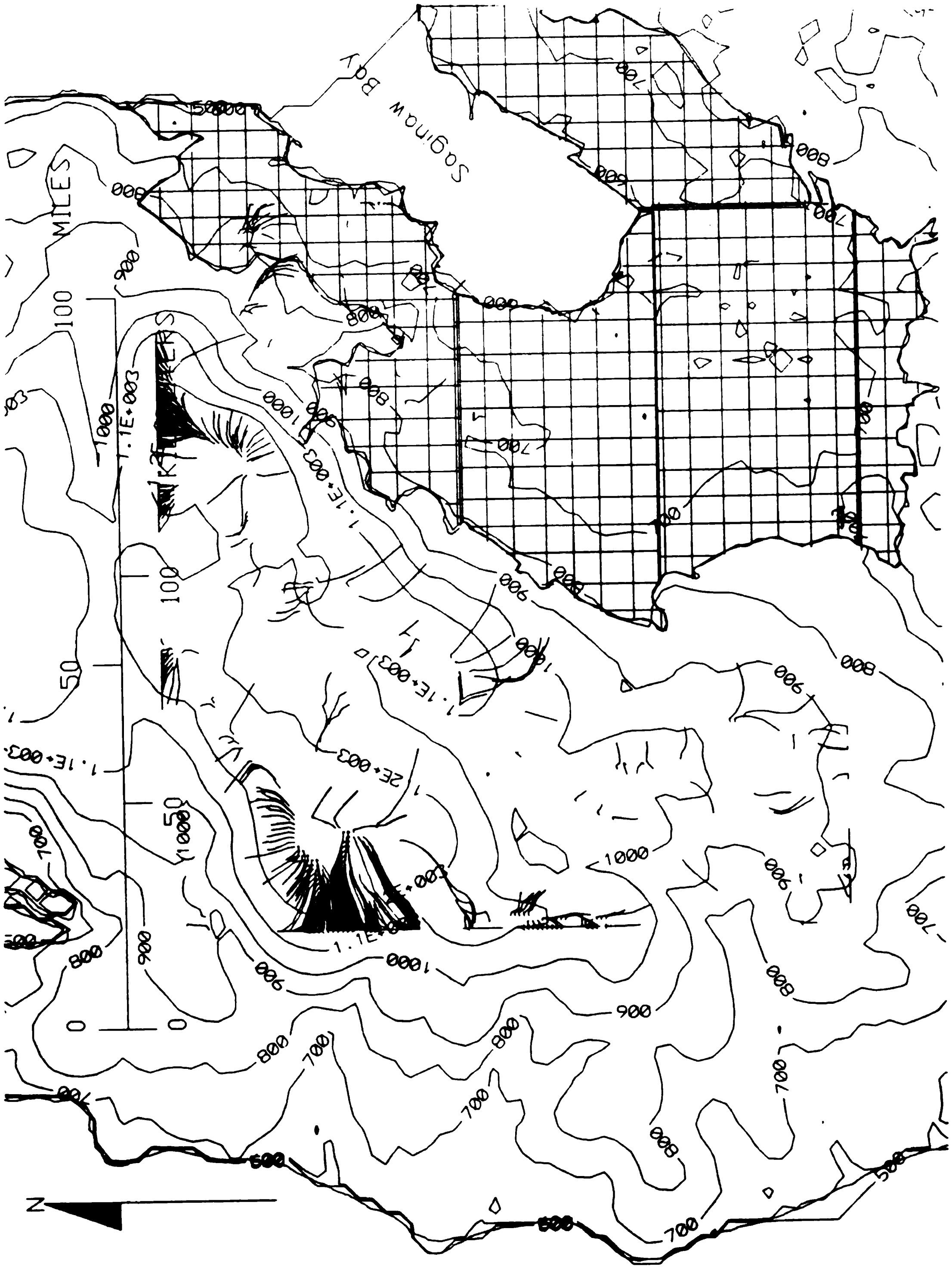

Figure 20. Flowpaths (579) exceeding 20,000 years from the water table to their respective positions at time = 20,100 years. Some paths may cross due to flow at different depths. Most (527 or 91%) were in the bedrock when transport was suspended at time = 20,100 years.

The flo

these f

opposit

within

Most c

shown

had tra

Bay a

Figure

travel

the Y

that c

chang

grou

5.2 C

Con

1.) T
insu
low
wou
the

The flowpaths are relatively short and do not extend into the Saginaw lowlands. Many of these flowpaths are directed away from the Saginaw lowlands, as they were loaded on the opposite side of the regional groundwater flow divide. A few of the flowpaths start and end within the Saginaw lowlands.

Most of the particles (23,039 of 24,311) discharged; a histogram of their travel times is shown in Figure 21. Most of the travel times are less than 500 years. Several particles (712) had travel times greater than 1,000 years, but over half (414) of these were loaded in Saginaw Bay and were stagnated before discharge in the bay. The remaining (298) are shown in Figure 22. Only fourteen (14) of these exceeded 10,000 years (Figure 23), the minimum travel time necessary to record part of the climate warming gradient immediately following the Younger Dryas event. These fourteen flowpaths (Figure 23) together with the flowpaths that exceed 20,000 years (Figure 20) are the only flowpaths that can record the climate change in the model. Their number and extent is insufficient to produce the light isotopic groundwater anomalies observed in the Saginaw lowlands.

5.2 Conclusions and Implications for the Isotopic Anomalies

Conclusions from the groundwater flow and particle transport study are as follows:

1.) The number and extent of flowpaths, from the water table, greater than 10,000 years is insufficient to produce the light isotopic groundwater anomalies observed in the Saginaw lowlands; climatically colder water from the Younger Dryas event and subsequent warming would have advected through the system and discharged. Groundwater residence times in the glacial aquifer from the water table are relatively short.

Figur
parti
discha
(freq

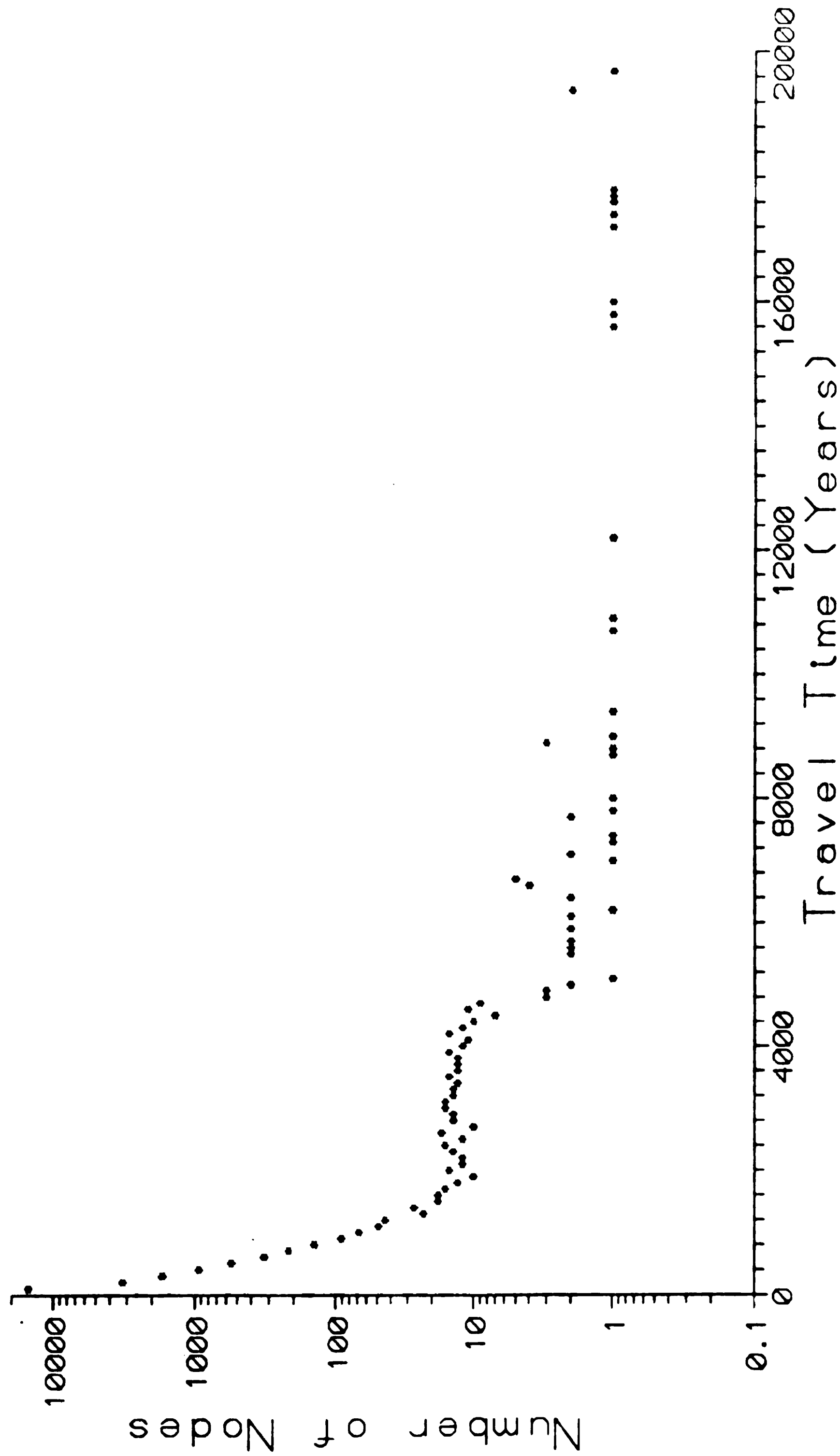

Figure 21. Travel time histogram of 23,039 nodes (1 particle released per node) whose particles ultimately discharged before 20,100 years. Note log scaling on y-axis (frequency).

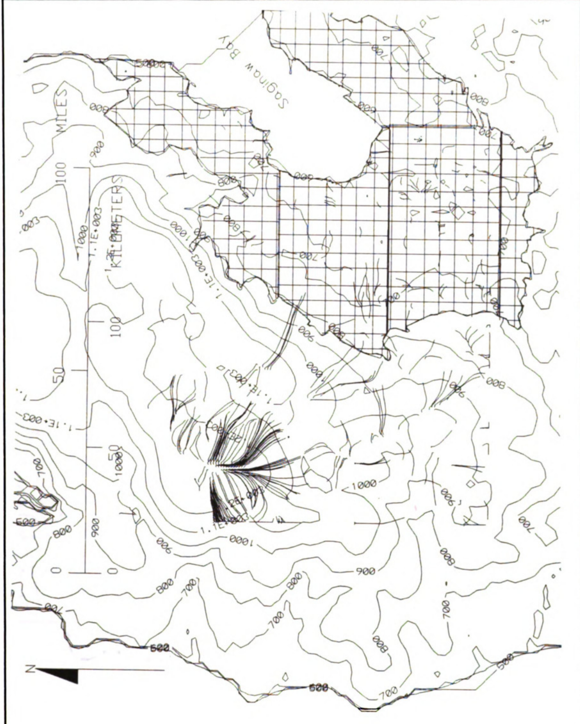

Figure 22. Flowpaths (298) from the water table to point of discharge of particles that discharged before 20,100 years and whose travel times exceeded 1000 years. Some paths may cross due to flow at different depths.

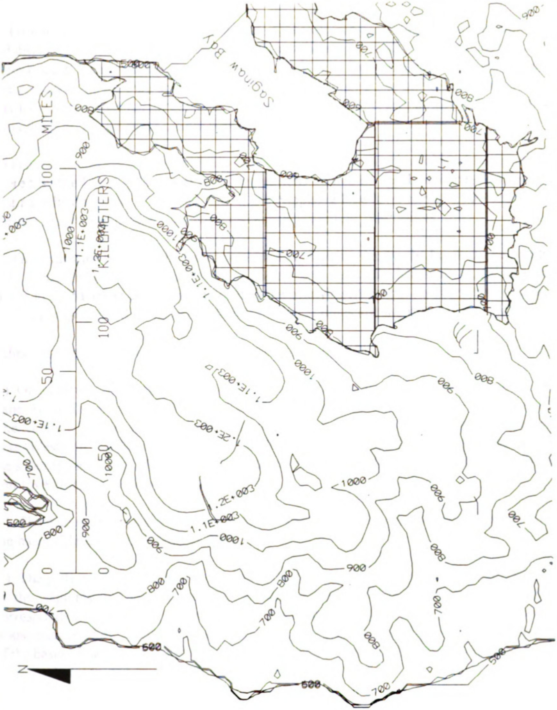

Figure 23. Flowpaths (14) of particles that discharged before 20,100 years whose travel times exceeded 10,000 years.

2.) Gro
and dis
water b
lake an
may ha
(Hoag

3.) Fro
altered
Sagina
present

4.) Gro
drift. H
Heads
flow ca

Other

5.) Gr
Lane.
The Sa
base (V
the sub
unit (V

6.) Th
the bas

7.) Th
in the
corresp
in and
of the
aquife

8.) Th
and 25
the dri
The d
central
tills (W
is gene

2.) Groundwater heads and flows in the glacial aquifer are controlled by local stream stages and discharges, resulting in localized flow cells accounting for about 90% of the overall model water budget. Direct discharge to the Great Lakes is from basins that immediately border the lake and accounts for only about 10% of the overall model water budget. Though recharge may have been somewhat underestimated affecting this result, regional flow is not significant. (Hoaglund et. al., (in press); Appendix V)

3.) From conclusion 2, it follows that groundwater heads and flows would not be significantly altered by the Great Lake water level fluctuations known for the Holocene. However, the Saginaw lowlands were flooded during the Nippising lake level about 4,000 years before present which may have changed the discharge pattern for that region.

4.) Groundwater heads in the bedrock aquifers are controlled by the heads in the overlying drift. Heads in the Saginaw and Parma aquifers closely match the heads in the glacial aquifer. Heads in the Marshall aquifer are affected by the overlying Michigan confining unit. Regional flow can be sustained in the bedrock, but residence times are significantly greater.

Other significant conclusions from previous studies include:

5.) Groundwater discharging in the Saginaw lowlands is commonly saline (Houghton 1838; Lane, 1899; Twenter, 1966; Westjohn and Weaver, 1996a,b), suggesting a bedrock source. The Saginaw aquifer has fresh and saline water near the top of the aquifer and brine at the base (Westjohn and Weaver, 1996a,b). The Marshall aquifer has fresh water in the region of the subcrop, but is typically comprised of saline water or brine beneath the Michigan confining unit (Westjohn and Weaver, 1996b,c).

6.) The map extent of brine increases from the center of the basin with increasing depth into the basin (Mandle and Westjohn, 1989).

7.) The pattern of light isotopic groundwater in the Marshall aquifer differs from the pattern in the Saginaw and glacial aquifers. Modern to light (-11 to -8) isotopic groundwater corresponds with overlying drift along the edges of the subcrop where it is freshwater bearing, in and outside of the lowland areas. The water gets progressively heavier toward the center of the basin. The lightest water is found in the lowland areas as with the glacial and Saginaw aquifers, but it is not as light (Ging et. al., 1996).

8.) The glacial drift thickness in the Saginaw and Michigan lowlands is generally between 50 and 250 feet thick whereas in central and northern Michigan, including the northern uplands, the drift thickness is 400 feet thick and greater (Western Michigan University, 1981, pl. 15). The drift in the lowlands is comprised of clay rich, possibly water-laid tills. The drift in central and northern Michigan has a high percentage of outwash sand and gravel and sandier tills (Westjohn et. al., 1994). The drift in southern Michigan, including the southern uplands, is generally between 100 and 250 feet thick (Western Michigan University, 1981, pl. 15), but has a much higher percentage of sand than the lowland areas (Westjohn et. al., 1994).

9.) The Pennsylvanian, including the Saginaw aquifer sandstone, is thickest along the western edge of the Saginaw lowlands southward to the Howell anticline (Westjohn and Weaver, 1996a).

5.3 Implications of Conclusions on the Isotopic Anomaly

Any explanation for light isotopic water in Michigan must be consistent with 1.) the localization of the anomaly in the Saginaw and Michigan lowlands, 2.) short groundwater residence times in the glacial drift aquifer, and 3.) long groundwater residence times in the bedrock aquifers. With the advective hypothesis eliminated by item 2, the two loading hypotheses remain: ice loading and lake loading. Whether ice loading or lake loading is responsible for the isotopic anomalies in the drift and bedrock, it is the subsequent 10,000 year Holocene time that must be spanned, without complete discharge, to preserve the isotopic anomaly.

5.4 The Alternative Loading Hypotheses:

5.4.1 Ice Loading

Prior to the development of the modern groundwater flow system 10,000 years ago, the groundwater flow system would have been affected by glacial loading. Several researchers have suggested that glaciation may have reversed Pleistocene groundwater flow directions due to a hydraulic connection between the ice sheet, its meltwater, and the groundwater beneath the ice sheet (McGinnis, 1968; Siegel and Mandle, 1984; Siegel, 1991; Filley and Parizek, 1983; Downey, 1986). The area of the icesheet-groundwater hydraulic connection, a discharge area prior to the ice loading, would have corresponded to a recharge area thereby reversing the groundwater flow direction. Light, isotopic groundwater may have been introduced to the aquifer system in the Saginaw lowlands during the groundwater flow reversal in response to the hydraulic load of the icesheet.

In ice loading, glacial meltwater, "including surface meltwater, rain, basal meltwater (geothermal and frictional), and meltwater from heat of deformation" (Carlson, 1994, p. 81; from Paterson, 1981), may collect at the base of the glacier in either a confined or unconfined system in hydraulic connection with the substrate, and with heads a function of the ice thickness. The ice profile is parabolic given by the relation (Mathews, 1974):

$$y=Ax^{1/2}$$

where x is the minimum distance from the ice margin to the profile point (in meters, m), y is the thickness (in meters, m), and A is a coefficient (in root meters, $m^{1/2}$). A value of 4.7 $m^{1/2}$ for the coefficient A fits observed portions of both the Greenland icecap and the Antarctic icesheet, but A does vary from glacier to glacier (Mathews, 1974). In a confined system, the water is pressurized up to 91% of the ice thickness as the loading pressure is a function of the ice density, thickness, and the gravitational acceleration. In an unconfined system, the glacial ice supports a water table that rises from the ice margin as a function of the englacial hydraulic properties of the ice-aquifer system and leakance into the substrate.

Michigan lower peninsula drift stratigraphy records four main Late Wisconsinan ice advances from 21,000 to 10,000 years ago: the Nissouri (maximum 18,000 years ago), the Port Bruce (maximum 14,800 years ago), the Port Huron (maximum 13,000 years ago), and the Greatlakean (maximum 11,800 years ago) (Eschman, 1985). The glacial and lake episodes that affected the lower peninsula from the Late Wisconsinan to recent are summarized in Figure 24.

Three of the four Late Wisconsinan ice advances could have loaded the aquifer systems of the Saginaw Lowlands. The Nissouri advance covered the entire lower peninsula establishing moraines in Indiana and Ohio (Eschman, 1985). The Port Bruce advance established a maximum advance at the Tekonsha moraine in southern Michigan. The Nissouri and Port Bruce advances were the last glacial advances to completely cover the lower peninsula. The Port Bruce advance was therefore the last time that light isotopes could have been loaded statewide, including the Marshall aquifer subcrop outside of the lowland areas (see conclusion 7). The Port Huron advance established the Port Huron moraine which parallels inland the shorelines of modern Lake Huron and the northern half of modern Lake Michigan. Loading of light isotopes from the Port Huron ice would have been much more localized, especially to the Saginaw lowlands. During the Greatlakean, the ice never advanced far enough to load the Saginaw lowlands.

To demonstrate the efficacy of ice loading, the Port Huron advance was modelled hydrologically using MODFLOW (McDonald and Harbaugh, 1988) and the existing RASA

Quaternary Event	Age Before Present	Lake Elevation	Glacial Lake Outlet
NISSOURI STADE	21,000 to 16,100		
ERIE INTERSTADE	16,100 to 15,500		
PORT BRUCE (Cary) STADE	15,500 to 14,500		
MACKINAW (Cary-Port Huron) INTERSTADE	14,500 to 13,000		
Glacial Lake Maumee I	14,500 to 14,000	800'	Fort Wayne outlet
Glacial Lake Maumee II	13,800	760', 780'	Imlay outlet(s) to Glacial Grand
Glacial Lake Arkona	13,700	710', 700', 695' < 580'	Glacial Grand Trent ?
Glacial Lake Chicago	14,500 to 13,700	640'	Chicago (Glenwood)
PORT HURON STADE	13,000 to 12,000		
Glacial Lake Whittlesey	13,000	738'	Ubley
Glacial Lake Saginaw	13,000	695'	Glacial Grand
Glacial Lake Warren	12,000	690'	Glacial Grand
Glacial Lake Chicago	13,000 to 12,000	640'	Chicago (Glenwood)
TWO CREEKAN INTERSTADE	11,850		
GREATLAKEAN STADE	11,800 to 11,000		
Glacial Lake Algonquin	11,500	≤605'	Trent, then Port Huron outlet
Glacial Lake Chicago	11,500	620'	Chicago (Calumet)
NORTH BAY INTERSTADE	10,300		
HOLOCENE/DRIFTWOOD STADE	10,000		
Lake Nippising	5,000 to 4,000	605'	Port Huron outlet

Figure 24. Summary of glacial events and glacial lake events for the lower peninsula of Michigan during Late Wisconsinan and Holocene.

model. The extent of Port Huron ice enabled ice profiling and hydrologic boundary definition within the existing RASA model domain of the lower peninsula, since the moraine (ice edge) is defined entirely within the domain with about 1/3 ice cover and 2/3 ice free. The Port Huron moraine was digitized and the minimum distance from the nodal center to the moraine was calculated for each cell in the RASA model using GLACIER.FOR (Appendix IV). For the ice covered nodes, GLACIER.FOR calculated an ice thickness using the parabolic relation above (Mathews, 1974) and an estimate of the profiling coefficient, A, below. The ice profiling coefficient, A, for the Pleistocene Laurentide icesheet varied regionally from .32 to 4.1 (Mathews, 1974). Clark et. al., 1994 used a thick (Hughes et. al., 1981) and a thin (Boulton et. al, 1985; Fisher et. al., 1985) icesheet thickness reconstruction of the Nissouri advance (maximum 18,000 years BP) to model the glacio-isostatic rebound. They found that the thin reconstruction resulted in the best fit between model results and observed lake outlet and shoreline chronologies, though the thin reconstruction slightly underestimated the amount of tilting suggesting the ice may have been thicker. The thin icesheet thickness reconstruction yields a profiling coefficient, A, of 1.8 which was used to generate the ice thicknesses.

The ice thicknesses were converted to feet and multiplied by 91%, the ice density, to determine the equivalent hydraulic load. The load was added to modern land surface elevations (including 580' for the Great Lakes) to determine the equivalent hydraulic loading head shown in Figure 25. The icesheet loading heads were introduced to the model as a layer 1 source bed; nodes were active in the region of the icesheet, with heads specified to the

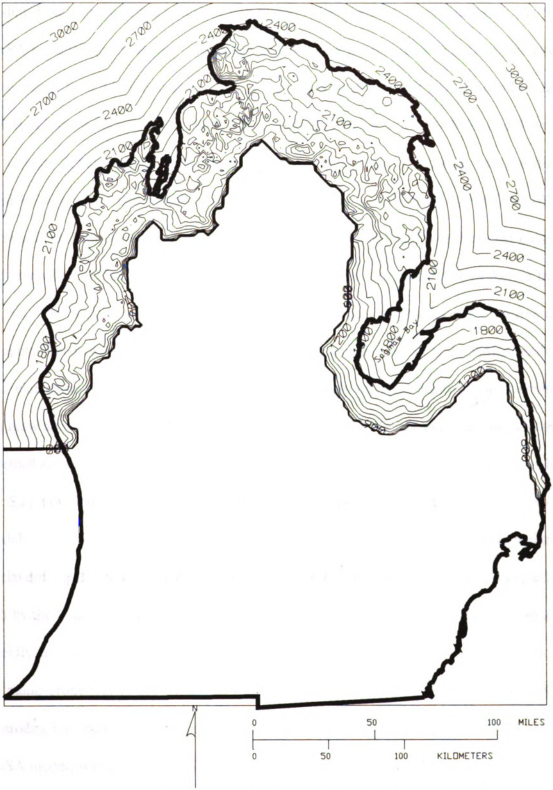

Figure 25. Elevation (heads) of effective hydraulic loading, as 91% of ice thickness added to modern land surface elevations, during the Port Huron advance.

loading heads, and inactive in the ice free region. The glacial drift aquifer (layer 1 in the RASA model) became layer 2. Recharge of the RASA model was specified to the upper active layer: layer 1 in the icesheet region of specified heads where it became uninfluential, and layer 2 in the ice free region. River nodes, assigned to layer 1 in the RASA model, were set to layer 2, but were eliminated in the region overlain by ice. The glacial drift specified head boundary was eliminated in the region overlain by ice (replaced by MODFLOW's automatic no-flow boundary next to inactive glacial drift aquifer nodes in the Great Lakes) and remained unchanged at 580' along the modern Great Lakes shoreline in the ice free region. The Saginaw aquifer (layer 2 in the RASA model) became layer 3. To free up computer Random Access Memory (RAM), the Parma aquifer (layer 3 in the RASA model) was eliminated since heads in the Parma aquifer closely matched heads in the overlying Saginaw aquifer, and since hydraulic communication between the Saginaw aquifer and the Marshall aquifer is more controlled by the Michigan confining unit below the Parma than by the Saginaw confining unit above the Parma. The Marshall aquifer (layer 4 in the RASA model) remained as layer 4. A VCONT controlling leakance between the icesheet and the glacial drift aquifer was calculated by dividing the vertical hydraulic conductivity of the glacial drift by the thickness of the drift. It was thereby assumed that the icesheet load was placed directly on the top of the glacial drift and that leakance was controlled by the vertical hydraulic properties of the subglacial drift evident in modern drift. The same VCONT controlling flow between the glacial drift aquifer and the Saginaw aquifer (layer 1 and 2 in the RASA model) was used between layers 2 and 3 in the glacial simulation. The effect of the Saginaw confining unit and the Parma aquifer on the calculation of leakance between the

Saginaw and Marshall aquifers was considered negligible compared to the effect of the Michigan confining unit since vertical conductivities add in series. Therefore the VCONT between the Parma and Marshall aquifers (layer 3 and 4 in the RASA model), representing the Michigan confining unit, was used between the Saginaw and Marshall aquifers (layers 3 and 4) in the glacial simulation.

The simulated heads in the glacial drift aquifer (Figure 26), the Saginaw aquifer (Figure 27), and the Marshall aquifer (Figure 28) show that the effect of loading is localized to the region of the icesheet where groundwater flow is reversed. Heads in the Saginaw aquifer closely match heads in the overlying glacial drift aquifer, though somewhat dampened and not as affected by the river sinks. On the map of heads in the glacial drift aquifer (Figure 26), a large region of "glacier-ward" drainage southwest of Saginaw Bay is enclosed by the 700' contour. This region was probably occupied by a proglacial lake slightly higher than 700' that was regulated by the glacial grand outlet between the two parallel 700' contours leading downstream to Lake Michigan. The heads in the Saginaw aquifer below this region are considerably higher which indicates that the region would have sustained large upward groundwater flow partially balancing downward flow beneath the icesheet. Heads in the Marshall aquifer below the icesheet are elevated, but are lower than the heads in the Saginaw aquifer particularly along the axis of the Saginaw bay icelobe where they are as much as 250 feet lower. This indicates a strong downward component of flow beneath the icesheet. However, heads in the Marshall aquifer in the ice free region are higher than the heads in the Saginaw aquifer, particularly in the region of the proglacial lake where they are as much as

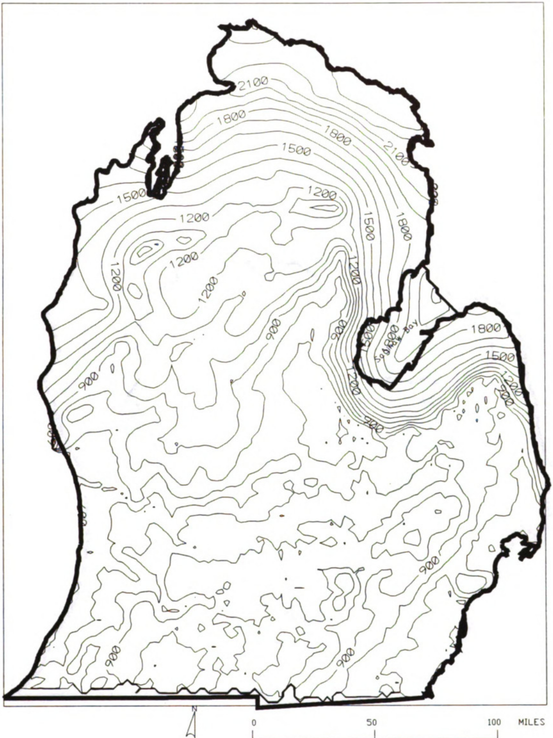

Figure 26. Simulated heads in the glacial drift aquifer in response to ice loading during the Port Huron advance.

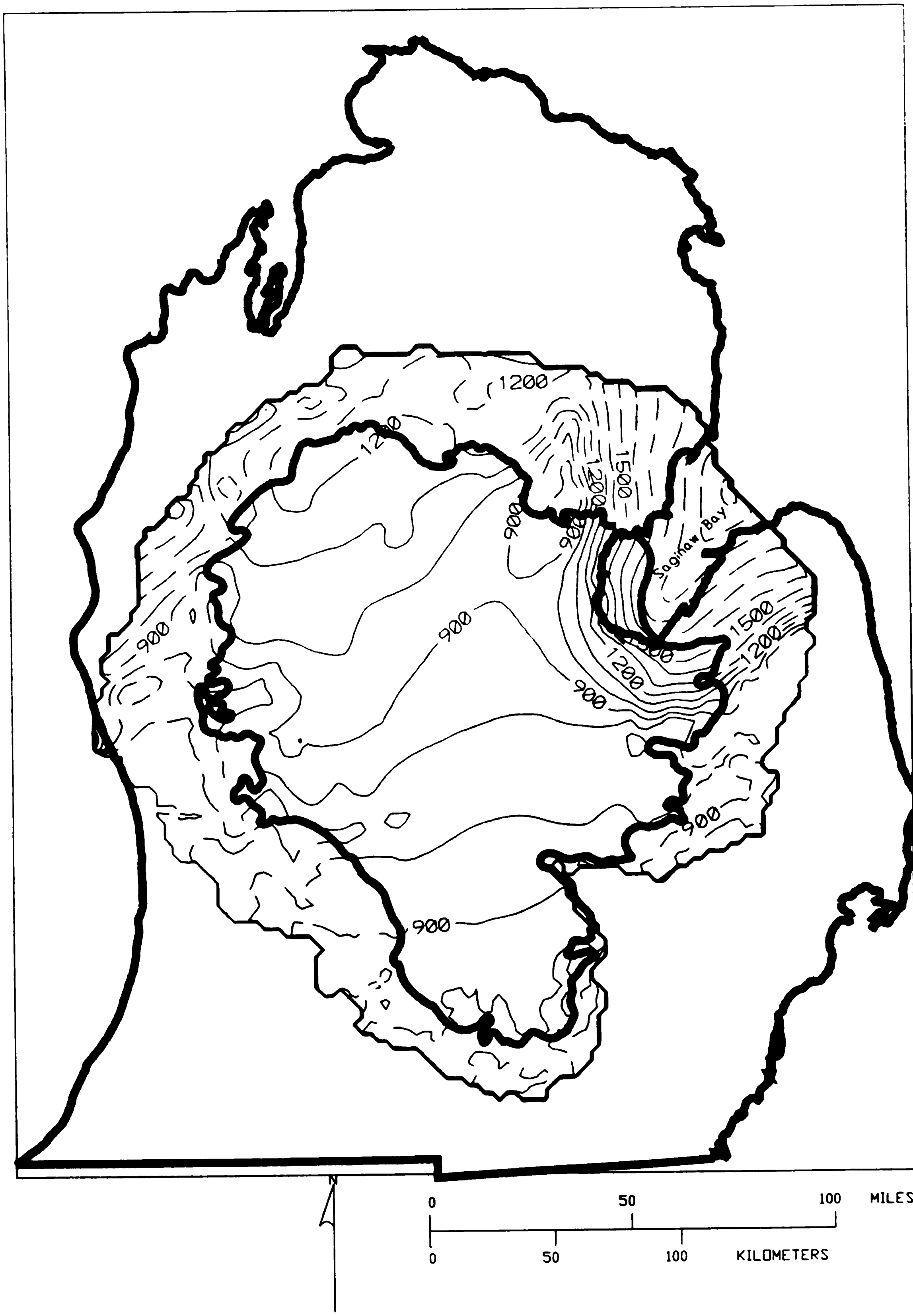

Figure 27. Simulated heads in the Saginaw aquifer in response to ice loading during the Port Huron advance.

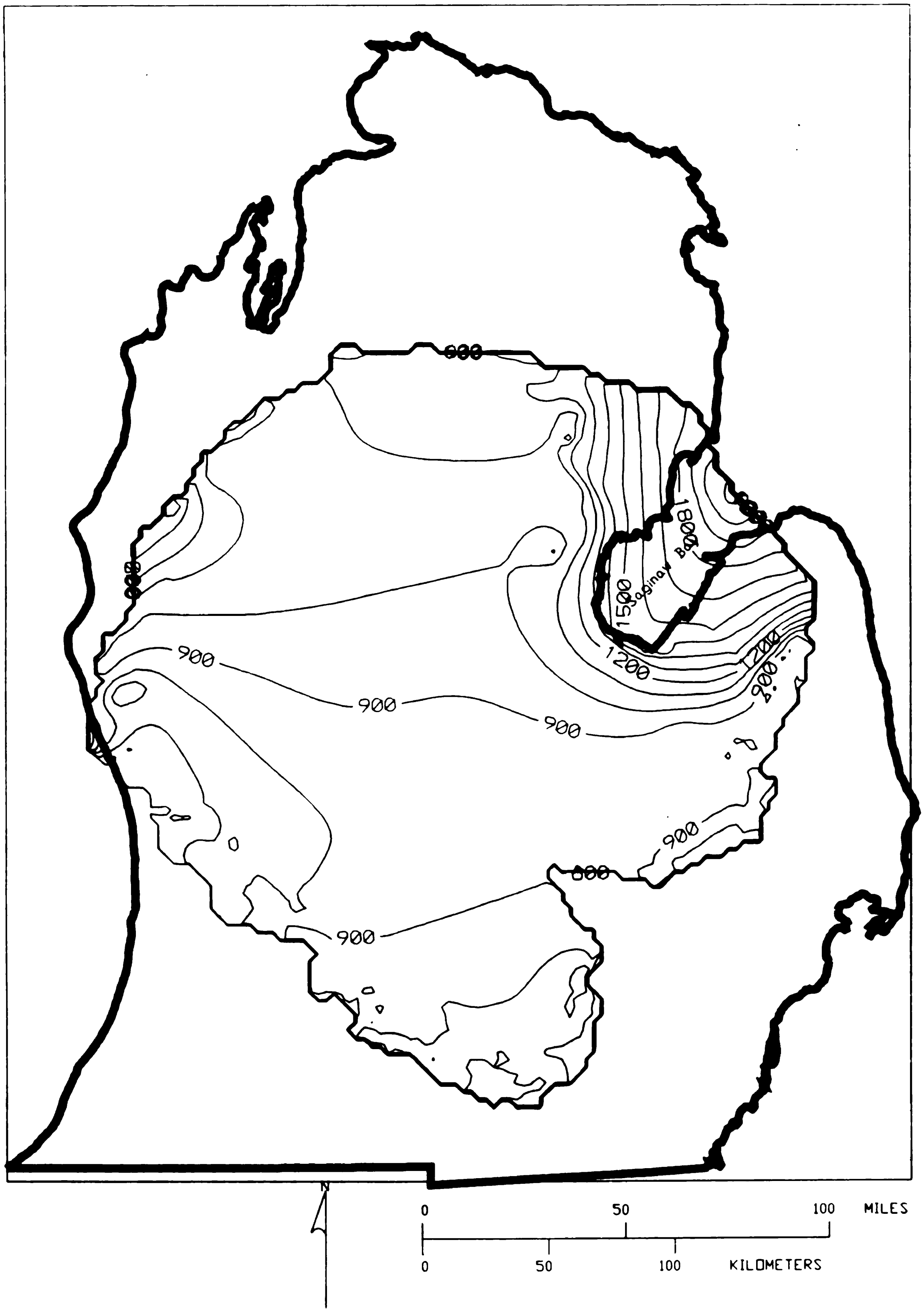

Figure 28. Simulated heads in the Marshall aquifer in response to ice loading during the Port Huron advance.

100 feet higher, indicating an upward component of flow to the Saginaw aquifer. However, the flux across the Michigan confining unit would be considerably less than the upward flux through the glacial drift from the Saginaw aquifer to the proglacial lake.

The general pattern of groundwater flow is downward under the icesheet and upward in the region immediately in front of the icesheet. This shows that given the assumptions in the model, ice loading is a very effective method of introducing light isotopic water from the icesheet to the aquifer system.

5.4.2 Lake Loading

During the Late Wisconsinan glacial and interglacial episodes and the Holocene, water levels in the Great Lakes bounding the basin have drastically changed due to climate, glacial damming, and isostatic rebound (Larsen, 1987; Hansel et. al., 1985; Chrzastowski and Thompson, 1992; Eschman and Karrow, 1985). The glacial lake levels bound the paleohydrologic system laterally and may have reversed groundwater flow during high stages. The glacial and lake episodes that affected the lower peninsula from the Late Wisconsinan to recent are summarized in Figure 24 and are shown in Figure 29. The Nissouri advance may have created the first Great Lake basins (Eschman, 1985). After the Erie Interstade, the Port Bruce advance overrode the Great Lakes basin, including glacial lake Milwaukee in the Lake Michigan basin and glacial lake Leverett in the Lake Erie basin (Hansel et. al., 1985; Figure 29A, Larson and Schaetzel, (in press)). When the ice retreated during the Mackinaw

Figure 29,
during the

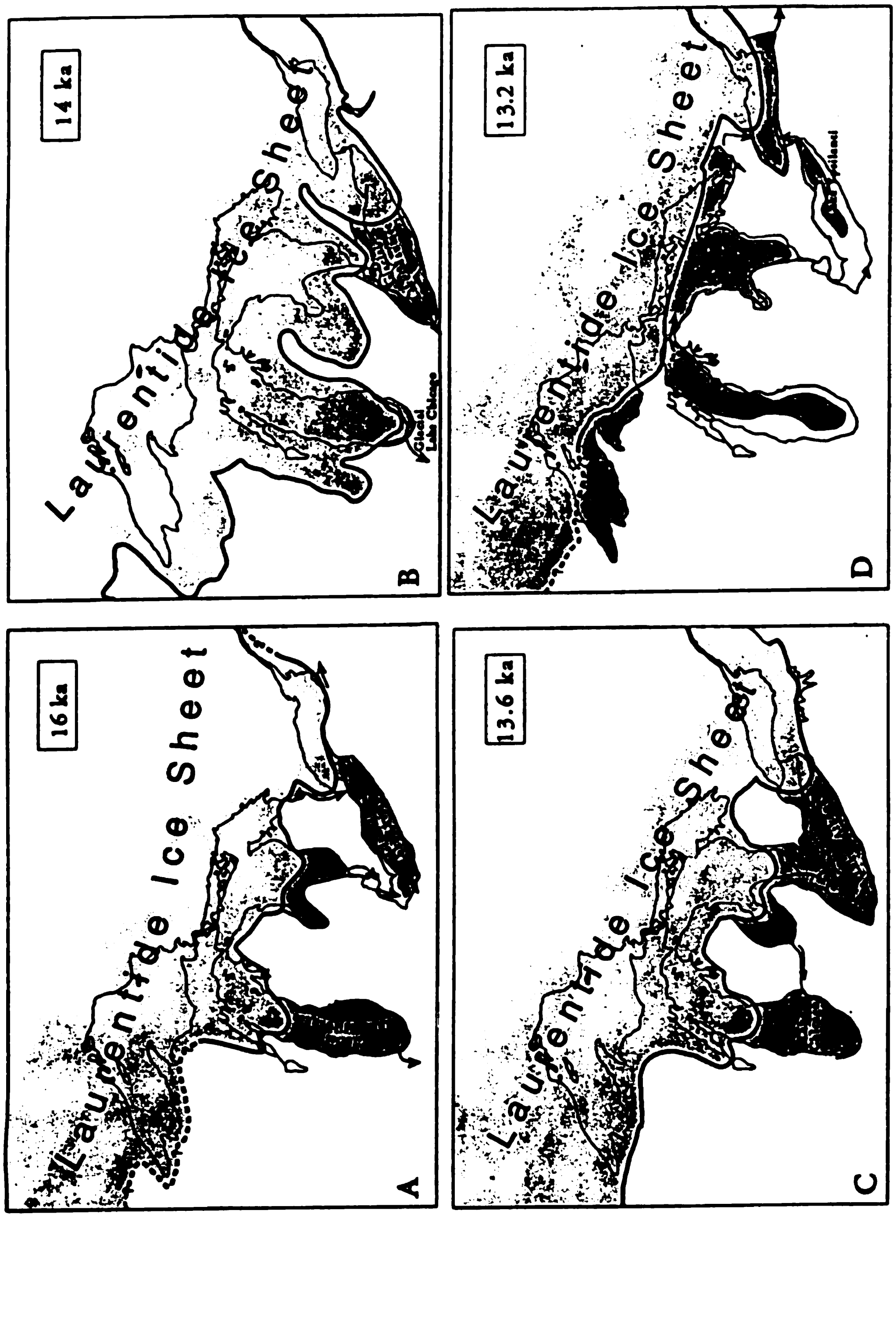

Figure 29, A-D. Glacial lakes and lake levels in response to glacial advances and retreats during the Late Wisconsinan.

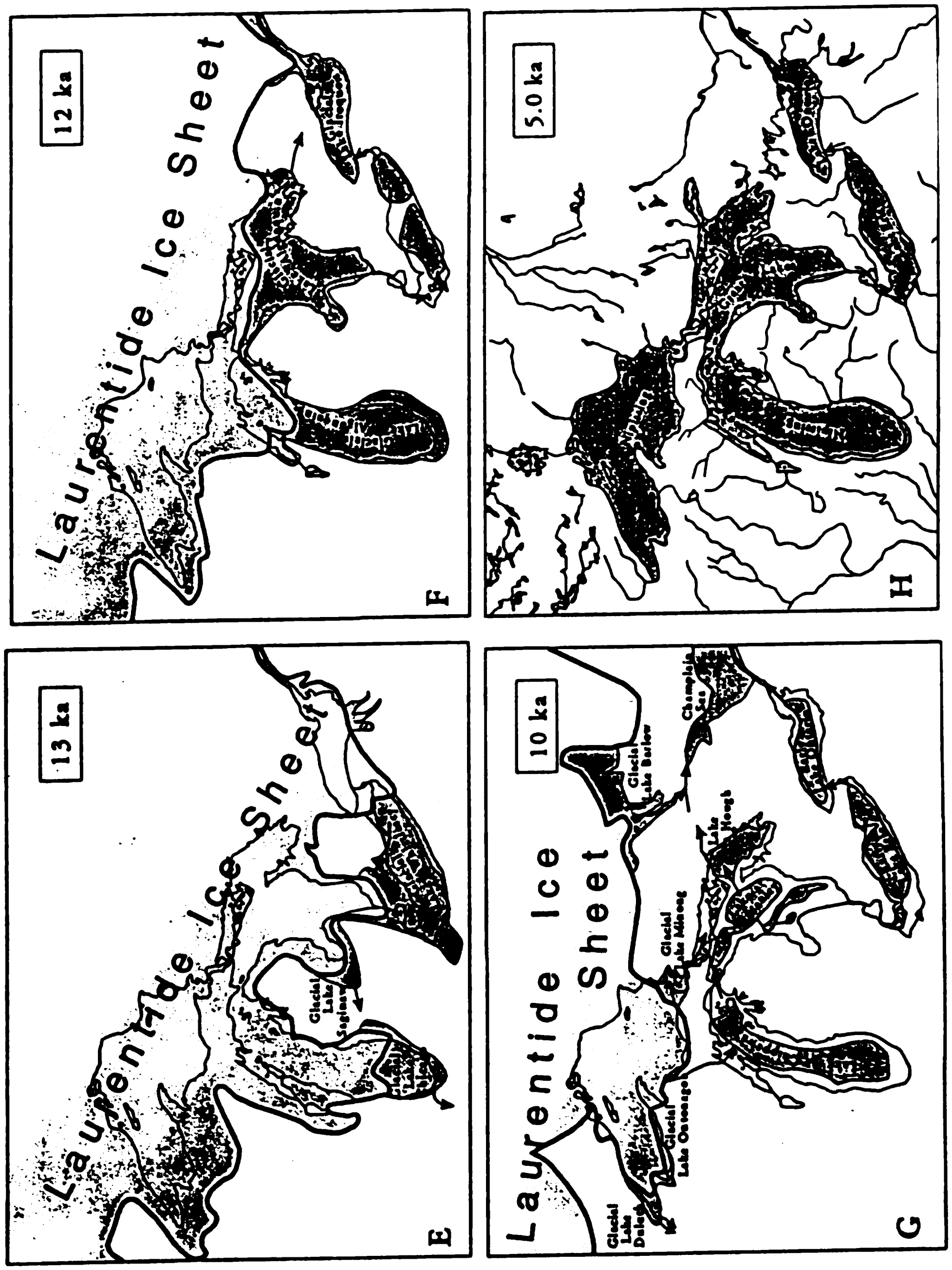

Figure 29, E-H. Glacial lakes and lake levels in response to glacial advances and retreats during the Late Wisconsinan.

intersta

in the I

Larson

approxi

margina

glacial l

lowland

loading

glacial l

and was

Figure

When t

lake Ar

and glac

1985; F

over 12

lowland

last time

glacial la

droppec

interstade, glacial lake Maumee I (800' stage) and glacial lake Chicago (640' stage) formed in the Lake Erie basin and Lake Michigan basin respectively (Larsen, 1987; Figure 29B, Larson and Schaetzel, (in press)). When the ice backed off of the Michigan "thumb" approximately 13,800 years ago, exposing the Imlay outlet(s) at 760' and then 780', an ice-marginal channel formed leading to the glacial Grand River outlet at 710' which drained to glacial lake Chicago (Eschman and Karrow, 1985). This was the first time that the Saginaw lowlands were flooded enough to possibly reverse the groundwater flow direction due to lake loading. When the ice retreated from the Saginaw lowlands approximately 13,700 years ago, glacial lake Arkona occupied the Lake Erie basin, completely flooded the Saginaw lowlands, and was regulated by the glacial grand outlet from 710' to 695' (Eschman and Karrow, 1985; Figure 29C, Larson and Schaetzel, (in press)).

When the ice readvanced during the Port Huron stade approximately 13,000 years ago, glacial lake Arkona split into two lakes: glacial lake Whittlesey regulated by the Ubley outlet at 738', and glacial lake Saginaw regulated by the glacial grand outlet at 695' (Eschman and Karrow, 1985; Figure 29E, Larson and Schaetzel, (in press)). When the ice started to retreat a little over 12,000 years ago, a single glacial lake Warren occupied the Lake Erie basin and Saginaw lowlands regulated by the glacial grand outlet at 690' (Eschman and Karrow, 1985). For the last time, these lakes could have reversed groundwater flow across the state of Michigan since glacial lake Chicago had a lower stage of 640' throughout this time. The Great Lake basins dropped below modern levels during the Two Creekan interstade 11,850 years ago.

When

glacial

lower g

Schaet

outlet.

When

29G, L

level m

Schae

since i

than n

5.5 Di

Both

moder

glacia

discha

recha

Sagina

of the

When the ice readvanced during the Greatlakean stade 11,500 years ago, a transgressive glacial lake Algonquin, at its highest stand, partially flooded the Saginaw lowlands while a lower glacial lake formed in the Lake Michigan basin (Larsen, 1987; Figure 29F, Larson and Schaetzel, (in press)). The lake level was most likely not high enough to use the glacial grand outlet, but an east to west hydraulic gradient may have formed.

When the ice retreated, the Great Lake basins dropped below modern levels again (Figure 29G, Larson and Schaetzel, (in press)). Between 4,500 and 4,000 years, the Lake Nippising level may have partially flooded the Saginaw lowlands (Larsen, 1987; Figure 29H, Larson and Schaetzel, (in press)), but this lake could not have been a source of isotopically light water since it was not proglacial and corresponded to the Holocene Maximum, a climate warmer than modern.

5.5 Discussion of Alternative Hypotheses

Both lake and ice loading are plausible mechanisms for reversing groundwater flow in the modern discharge zone. In either case, the light isotopic groundwater mass loaded into the glacial, Saginaw, and Marshall aquifers must have endured 10,000 years of subsequent discharge during the Holocene. Given the short residence time of modern groundwater recharged to the water table in the glacial aquifer--even for the fine grained drift of the Saginaw lowlands-- the penetration from the loading must have been deep, at least to the base of the glacial drift. Clearly, the light isotopic signature of the Saginaw and Marshall aquifer suggests deeper bedrock loading. The bedrock aquifers have relatively long residence times as shown in the particle tracking as 91% of the flowpaths from the water table greater than 20,000 years involved bedrock transport. The long residence time for groundwater in the bedrock aquifers would explain how the isotopic anomaly endured for 10,000 years; the bedrock aquifers acted as a receptacle to store the light groundwater mass and source the overlying drift. However, the long residence time for groundwater in the bedrock aquifers also implies that the loading must have taken a considerable period of time, even though the loading gradients, as indicated in the model, were much larger than modern discharge gradients. Alternatively, the isotopic anomaly may have been capped after loading by either 1) deposition of fine grained tills, water laid tills, and lacustrine deposits, and/or 2) a change in aquifer properties due to loading.

Deep bedrock penetration acting over a long period of time favors the ice loading mechanism. Localization of the anomaly in the Saginaw and Michigan lowlands is more consistent with ice loading specific to the Port Huron advance (since ice coverage during the Nissouri and Port Bruce advances was more extensive and would have affected a larger area), and/or the lake loading mechanism. The meteoric signature of the isotopically light groundwater is most consistent with ice loading since proglacial lakes would have sustained isotopic fractionation; source water from lakes may not be meteoric. Modern to light meteoric water corresponding to the overlying drift is found in the Marshall aquifer subcrop regions (conclusion 7). The localization of the anomaly for all of the bedrock aquifers seems to correspond to thin glacial aquifer (conclusion 8) and thick Saginaw aquifer (conclusion 9). It is possible that a thick, sandy

grou

Mars

mass

sandy glacial aquifer outside of the area of localization (conclusion 8) allowed for complete groundwater transport and discharge of the meltwater, while thick Saginaw aquifer and Marshall aquifer subcrop, under thin drift, acted as a receptacle to store the light groundwater mass and sourced the overlying drift.

APPENDIX I

A. I

Time

mode

tempe

$\delta^{18}O$

recha

Initiall

This ma

APPENDIX I

A. Derivation of the isotopic mass input function

Time intervals of constant $\delta^{18}O$ and δD composition of winter precipitation--calculated by the model of Yurtsever and Gat (1981) corresponding to intervals of constant winter semi-annual temperature and/or precipitation--were used to calculate recharge volumes of constant known $\delta^{18}O$ and δD composition at desired nodal locations of the groundwater flow model. The recharge rate multiplied by the time duration and grid spacing yields a volume of water:

$$V=R\Delta t\Delta x\Delta y$$

Initially assuming a density of 1.00 g/cm^3, the total mass of the water can be calculated:

$$M_T=V\rho$$

This mass, converted to atomic mass units (a.m.u.), must equal the sum of the masses of deuterium, hydrogen, ^{18}O, and ^{16}O--the first of 4 equations and 4 unknowns:

$$M_T = D + H + {}^{18}O + {}^{16}O$$

The ratios of the masses (in a.m.u.) of deuterium to hydrogen and ^{18}O to ^{16}O can be calculated from the definition of del (δ),

$$\delta = \frac{R_{sample} - R_{standard}}{R_{standard}} x 1000$$

resolved for the sample ratio:

$$R_{sample} = R_{standard}\left(\frac{\delta}{1000} + 1\right)$$

The SMOW standard $^{18}O/^{16}O$ ratio is 0.0020052 and the SMOW standard D/H ratio is 0.0001558 (Lohmann, written communication, 1993). Since the $\delta^{18}O$ and δD is known, the equation above provides two more equations in terms of the 4 unknowns:

$$R_{18}=(0.0020052)[\frac{\delta^{18}O}{1000}+1]={}^{18}O/{}^{16}O$$

and

$$R_D=(0.0001558)[\frac{\delta D}{1000}+1]=D/H$$

The final equation to constrain the four separate isotope masses comes from a stoichiometric argument of a simple water molecule (Gram Molecular Weight {GMW} = 18) where 1 molecule of water is comprised of 2 atoms of hydrogen (2 a.m.u.) and 1 atom of ^{16}O (16 a.m.u.). The mass ratio of oxygen to hydrogen is 8, providing the fourth equation:

$$H=(\frac{1}{8})^{16}O$$

The separate equations imply that the water can be expressed molecularly in terms of light water, $H_2{}^{16}O$ (18 a.m.u.), and heavy water, $D_2{}^{18}O$ (22 a.m.u.). This may be somewhat of an oversimplification since nine isotopic configurations of water are possible considering the different combinations of ^{16}O, ^{17}O, ^{18}O, D, and H (Faure, 1986).

Backward substitution of the ratio equations into the equation for total mass yields:

$$M_T=(R_D+1)H+(R_{18}+1)^{16}O$$

Backward substitution of the stoichiometric relation, H = 1/8 O, yields two equations, one in terms of H, the other in terms of O:

$$M_T=(R_D+1)H+(R_{18}+1)8H$$

or

$$M_T=\frac{(R_D+1)^{16}O}{8}+(R_{18}+1)^{16}O$$

Solving the first equation for H gives:

$$H=\frac{M_T}{R_D+8R_{18}+9}$$

and

$$D = R_D H$$

Solving the second equation for ^{16}O gives:

$$(^{16}O) = \frac{M_T}{(\frac{R_D + 1}{8}) + (R_{18} + 1)}$$

and

$$(^{18}O) = R_{18}{}^{16}O$$

The total mass can be re-added, $M_T = H + D + {}^{18}O + {}^{16}O$, and the density can be resolved (Lohmann, pers. comm., 1993):

$$\rho = \frac{M_T}{V}$$

The new density is multiplied by the volume to obtain a new M_T and the process is repeated iteratively until the density converges (Lohmann, pers. comm., 1993).

B. Partitioning the Solvent

Random-Walk is designed to calculate the concentration of single solutes; the model was modified to calculate isotopic masses. In Random-Walk, the total number of particles accumulated at any node represents a mass of the solute for that node. This mass is divided by the volume of the water--the solvent--at the node to obtain a concentration. In general, the more particles used in a simulation, the greater the resolution of the solute concentration. Both the concentration and total number of particles are printed for each node. However, the stable isotopes of hydrogen and oxygen comprise both the solute and the solvent. Table 4 shows the change in each isotopic mass for a 1 $\delta^{18}O$ change in the isotopic composition, along the meteoric water line, of a cubic meter (1,000 kg) of water. The change, in thousandths and 10-thousandths of a kilogram, is extremely small compared to the total mass (1,000 kg). As a result, the number of particles required to represent the total mass and still resolve the change in isotopic mass would be extreme.

To modify the program for isotopes, the solute and solvent were partitioned by establishing a solvent isotopic composition base on the meteoric water line plot. If the lightest water in the groundwater flow system is chosen as the solvent, any heavier water can be represented as the solvent plus the "solute" mass change between the heavy water and the solvent. Since all waters modelled were meteoric, the meteoric water line was used as a tie line between the solvent base and the heavier water. However, any tie line can be used to determine the increase in ^{18}O mass (at the expense of ^{16}O mass) and the increase in ^{2}H mass (at the expense

Table 4: Change in each isotopic mass for a 1 $\delta^{18}O$ change in the isotopic composition, along the meteoric water line, of a cubic meter (1,000 kg) of water.

Isotope	Change in isotopic masses per +1 unit change in $\delta^{18}O$ in 1000 Kg water
^{16}O	$-1.7013105502481734 \times 10^{-3}$ Kg
^{18}O	$1.775769038179963 \times 10^{-3}$ Kg
D	$1.382052769614076 \times 10^{-4}$ Kg
H	$-2.126638128102168 \times 10^{-4}$ Kg

of ^{1}H mass) from the solvent base. As in the case for determining solute concentrations, the more particles used in a simulation, the greater the resolution on the change in isotopic mass. In addition, each isotope mass can be modeled separately in order to maximize the number of particles processed for each isotope.

The isotope loading function is a subroutine, SUBROUTINE MASSIN, to the particle generating subroutine, SUBROUTINE GENP, of the RAND3D code. MASSIN was run as an independent program to determine the temperature range required to produce the range in $\delta^{18}O$ observed in the Saginaw lowlands, from approximately -19 to -9. The formulas of Yurtsever and Gat (1981) were constrained for an "average location" in the northern uplands recharge area, Cadillac, Michigan: Latitude 44.00 degrees; Altitude 1,200 feet = 366 meters; and Precipitation of 20 inches per summer from the pollen studies of Webb and Bryson (1972) converted to 169 millimeters per month (average yearly). The average yearly temperature versus the resulting $\delta^{18}O$ is shown in Table 5. A considerably colder climate with an average yearly temperature of 21°F would be required to produce the light $\delta^{18}O$ of -18 observed in the Saginaw lowlands. Cadillac's modern average yearly temperature is 45.5°F, with a January average temperature of 19.8°F and July average temperature of°68 F (Reference: NOAA Climate Summaries). Even the Younger Dryas event--with a corresponding Michigan July average temperature of 65°F (Webb and Bryson, 1972)--may not have been cold enough. However, the Younger Dryas and other stadial climates were known for extremes between winter and summer temperatures; the average yearly temperature may have been cold enough.

Table 5: Average Yearly Temperature vs. $\delta^{18}O$ using formulas from Yurtsever and Gat, 1981

Cadillac, Michigan
Latitude: 44°
Altitude: 1,200 feet = 365.76 meters
Precipitation: 20 inches/summer = 169.33 mm/month

Temperature °F	$\delta^{18}O$
21	-17.90
24	-16.90
27	-15.89
30	-14.90
33	-13.90
36	-12.91
39	-11.92
42	-10.92
45	-9.92

C. ISORAN3D.FOR Fortran Code

Eliminated lines within border to make TRAVELT.FOR (Subroutine GENP, Appendix II):

```
C NEW RANDOM WALK 3-D MODEL
C   REVISED TO SMALLER GRID ARRAY 1/9/90
C   PLT ROUTINE REVISED 2/15/89
C
C     VARIABLES
C     ---------
C ASPECT - ASPECT RATIO OF SCREEN
C BOT(I,J,K) - ELEVATION OF BOTTOM OF NODE I,J,K (FT)
C BOT(I,J,K) - ELEVATION OF BOTTOM OF NODE I,J,K (FT)
C BOTCAPTURE - BOTTOM ELEVATION AT WHICH PARTICLE MAY BE
C              CAPTURED BY SINK
C CAPTUREMAX - RIVER MAXIUM CAPTURE DISTANCE
C CDX - COLUMN SPACING OF MAP ARRAY
C CDY - ROW SPACING OF MAP ARRAY
C COLUMNS - NUMBER OF COLUMNS IN PLOT ARRAY
C CONC(L) - SINK CONCENTRATION FOR SINK L(MG/L)
C D - TIME STEP FOR MOVE, =DELTA IF SLUG SOURCE PARTICLE
C     = RANDOM FRACTION OF DELTA FOR CONTINUOUS SOURCE PARTICLE
C     (DAYS)
C DD - HORIZONTAL DISTANCE TO MOVE (FT)
C DDX - DISTANCE OF MOVE (FT)
C DELTA - TIME STEP OF MOVE SUBROUTINE (DAYS)
C DELX - COLUMN SPACING (FT)
C DELY - ROW SPACING (FT)
C DL  - LONGITUDINAL DISPERSIVITY (FT)
C DMAX - MAXIMUM HORIZONTAL MOVE DISTANCE BEFORE RECOMPUTING
C        VELOCITY (DAYS)
C DT  - TRANSVERSE DISPERSIVITY (FT)
C DV  - VERTICAL DISPERSIVITY (FT)
C E   - POROSITY
C ENDTIME(M) - ENDTIMES FOR EACH MODEL VELOCITY FILE (DAYS)
C FS - MAP FLAG, IF='A' THEN MAP HAS BEEN RUN
C GS - FLAG INDICATING COEFFICIENTS HAVE BEEN ENTERED IF
C      ='B'
C I1 - X COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW
C       (FT)
C I2 - X COORDINATE OF UPPER LEFT CORNER OF SCREEN WINDOW
C       (FT)
C ICOLXS - COLUMN DISPLAYED IN COLUMN CROSS-SECTION VIEW
C IPEN - COLOR OF SPECIAL FEATURES ON SCRREN
C J1 - Y COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW
C       (FT)
C J2 - Y COORDINATE OF UPPER LEFT CORNER OF SCREEN WINDOW
C       (FT)
C JROWXS - ROW DISPLAYED IN ROW CROSS-SECTION VIEW
C LAMDBA - HALF LIFE FOR FIRST ORDER DECAY (YEARS)
C LL - LAYER OF PARTICLE
C LLX  - X COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
C LLY  - Y COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
```

```
C LLZ  - Z COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
C LN - LINES OF MAP
C LOWERLX - X COORDINATE OF LOWER LEFT CORNER OF PLOT AREA (FT)
C LOWERLY - Y COORDINATE OF LOWER LEFT CORNER OF PLOT AREA (FT)
C MAPTYPES - FLAG INDICATING WHAT VIEW OF THE PROBLEM WAS LAST
C            DISPLAYED ON THE SCREEN , A - TOP VIEW, B - ROW
C            CROSS-SECTION VIEW, C - COLUMN CROSS-SECTION VIEW
C NC  - NUMBER OF COLUMNS
C NR  - NUMBER OF ROWS
C NENDTIME - NUMBER OF TIME PERIODS
C NEWX - X COORDINATE OF PARTICLE AFTER MOVE (FT)
C NEWY - Y COORDINATE OF PARTICLE AFTER MOVE (FT)
C NL  - NUMBER OF LAYERS
C NMAP(I,J) - MAP AND PLOT STORAGE FOR # OF PARTICLES OR
C             CONCENTRATIONS
C NP  - NUMBER OF PARTICLES
C NR  - NUMBER OF ROWS
C NS - NUMBER OF SINKS
C NSPF - NUMBER OF SPECIAL FEATURE FILES (WITH KEYS AND FILE
C       NAMES)
C O   - SLIDE COUNTER
C PM  - PARTICLE MASS (LBS)
C PPP - CONCENTRATION FACTOR, =0.001 FOR PPB, =1 FOR PPM
C       =1000 FOR PPT
C QSUM(L) - DISCHARGE OF SINK L (GPD)
C RS - NAME OF VELOCITY FILE (WITH EXTENSION .RND)
C R3 - DISTANCE FROM SINK TO PARTICLE (FT)
C R4 - VERTICAL DISTANCE FROM PARTICLE TO BOTTOM OF RIVER
C RETARD(K) - RETARDATION FACTORS FOR EACH LAYER AND CONFINING LAYER
C RL - LONGITUDINAL DISPERSION (FT)
C RN - NORMALLY DISTRIBUTED RANDOM NUMBER BETWEEN -6 AND +6
C      PRODUCED IN SUBROUTINE RAND
C RND() - RANDOM NUMBER BETWEEN ZERO AND 1
C       PRODUCED IN FUNCTION RND()
C ROWS - NUMBER OF ROWS IN PLOT ARRAY
C RT - TRANSVERSE DISPERSION (FT)
C RZ - VERTICAL DISPERSION (FT)
C SCALEDEF - DEFAULT X DISTANCE FOR SCREEN WINDOW
C SCALEX - X DIMENSION OF SCREEN WINDOW (I2-I1) (FT)
C SCREENX - X COORDINATE TO PLOT ON SCREEN (FT)
C SCREENY - Y COORDINATE TO PLOT ON SCREEN (FT)
C SIZE - GRID SPACING FOR PLOT AREA (FT)
C SPFFILES(I) - FILE NAMES OF SPECIAL FEATURE FILES
C SPFKEYS(I,J) - KEYS USED TO CALL SPECIAL FEATURE FILE PLOTS, I
C                IS SPECIAL FEATURE COUNTER, J IS 1 OR 2 CONTAINING
C                BOTH UPPER AND LOWER CASE LETTERS
C SPFSAVES - NAME OF FILE WITH SPECIAL FEATURE DATA
C SVIEWX - X DIMENSION OF ZOOM BOX
C SWITCH(I) - ARRAY OF FLAGS INDICATING WHETHER IT IS A
C               SLUG OR CONTINUOUS PARTICLE, 1= CONTINUOUS,
C               0=SLUG
C T2  - TIME (DAYS)
```

```
C TEMPSCR - FLAG, IF=1 IT INDICATES A SCREEN IMAGE HAS BEEN
C           SAVED AS 'TEMP.SCR'
C THICK(I,J,K) - THICKNESS OF AQUIFER OF COLUMN I, ROW J, AND LAYER K
C TOP(I,J,K) - ELEVATION OF TOP OF NODE I,J,K, IF THE TOP WATER TABLE
C                 LAYER THEN =0 UNLESS A RIVER NODE, THEN = BOTTOM OF
C                 RIVER ELEVATION (FT)
C TOPCAPTURE - TOP ELEVATION AT WHICH PARTICLE MAY BE
C              BE CAPTURED BY SINK
C VI(I,J,K) - VELOCITY IN X DIRECTION FROM I,J,K TO I+1,J,K (FT/DAY)
C VJ(I,J,K) - VELCOITY IN Y DIRECTION FROM I,J,K TO I,J+1,K (FT/DAY)
C VK(I,J,K) - VELOCITY IN Z DIRECTION FROM I,J,K TO I,J,K+1 (FT/DAY)
C VX - INTERPOLATED X VELOCITY (FT/DAY)
C VY - INTERPOLATED Y VELOCITY (FT/DAY)
C VZ - INTERPOLATED Z VELCOITY (FT/DAY)
C WD - COLUMNS OF MAP
C X(I) - ARRAY OF X COORDINATES FOR PARTICLES (FT)
C X1(L) - X COORDINATE OF SINK L (FT)
C XP - X COORDINATE OF PARTICLE POSITION IN GRID UNITS
C XXS - FLAG INDICATING A COLOR MONITOR IS PRESENT, ALWAYS='Y'
C       IN THIS VERSION
C Y(I) - ARRAY OF Y COORDINATES FOR PARTICLES (FT)
C Y1(L) - Y COORDINATE OF SINK L (FT)
C YASPECT - ASPECT RATIO OF SCREEN FOR TOP VIEW
C YP - Y COORDINATE OF PARTICLE POSITION IN GRID UNITS
C Z(I) - ARRAY OF Z COORDINATS FOR PARTICLES (FT)
C Z1(L) - LAYER OF SINK L
C Z2S - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON SCREEN
C Z3S - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON SCREEN
C ZASPECT - ASPECT RATIO OF SCREEN FOR CROSS-SECTION VIEW
C ZMAX - MAXIUM VERTICAL MOVE DISTANCE BEFORE RECOMPUTING
C        VELOCITY (DAYS)
C ZZZ - VARIABLE USED TO IMPLEMENT FIRST ORDER DECAY
C
C RECOVER FROM ALL ERRORS WITHOUT BOMBING OUT.
C 40 ON ERROR GOTO 5430
C NAME: THREE DIMENSIONAL RANDOM WALK (MICROCOMPUTER VERSION)
C
C PURPOSE: TO SIMULATE 3-DIMENSIONAL, STEADY OR NONSTEADY
C          MASS TRANSPORT PROBLEMS IN HETEROGENEOUS AQUIFERS
C          WITH THE RANDOM WALK PARTICLE THEORY.
C
C WRITTEN BY:  DONALD KOCH
C              ENGINEERING TECHNOLOGIES ASSOCIATES, INC.
C              3458 ELLICOTT CENTER DRIVE
C              ELLICOTT CITY, MD, 21043
C             (301) 461-9920
C     BASED ON CODE WRITTEN BY
C              THOMAS A. PRICKETT & ASSOCIATES
C             URBANA, ILLINOIS 61801
C              (217)384-0615
C BLOCKS BY JRH
C FLAGS
```

```
      CHARACTER GS,FS,ANSW
      CHARACTER*5 AS
      CHARACTER*40 RS
      REAL LLX,LLY,LLZ,LAMDBA,I1,J1,I2,J2,MAPCONC(161,151),PM
      REAL DELTA,T2,ENDTIME,DELX,DELY
      INTEGER SWITCH(10001),NC,NR,NL,NPER,NENDTIME
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/GEO/THICK,BOT,TOP
      COMMON/FLOW/VI,VJ,VK
      COMMON/SINK/NS,X1,Y1,Z1,QSUM
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/FLAGS/GS,FS,RS
      COMMON/MAPS/NMAP,CONC,MAPCONC,CDX,CDY,I1,J1,I2,J2
      DIMENSION THICK(161,151,4),BOT(161,151,4),TOP(161,151,4)
      DIMENSION VI(161,151,4),VJ(161,151,4),VK(161,151,4)
      DIMENSION X1(5000),Y1(5000),Z1(5000),QSUM(5000)
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION NMAP(161,151),CONC(5000)
      DIMENSION ENDTIME(20)
C END BLOCKS BY JRH
C ALLOW ARRAYS FOR 10000 PARTICLES AND 99 SINKS AND 35 BY 35 VELOCITY ARRAY
C
C INITIALLY SEED RANDOM NUMBER GENERATOR?
      WRITE(*,*)' RANDOM # GENERATION IS SEEDED WITH 1 BY DEFAULT.'
      WRITE(*,*)' CHOOSING A NEW SEED WILL RESULT IN A NEW SEQUENCE'
      WRITE(*,*)' OF RANDOM NUMBERS.  DO YOU'
      WRITE(*,*)' WANT TO RESEED THE RANDOM # GENERATOR (Y OR N)?'
      READ(*,'(A)') ANSW
      IF ((ANSW.EQ.'Y').OR.(ANSW.EQ.'y')) CALL SEED(RND$TIMESEED)
C INITIALIZE VELOCITY FILE (TIME PERIODS) COUNTER
      NPER=0
C INITIALIZE NUMBER OF VELOCITY FILES
      NENDTIME=0
C SET COEFFICIENT FLAG
      GS='F'
C INITIALIZE VARIABLES
      T2=0
      TIME=0
C NUMBER OF SINKS=0
      NS = 0
C NUMBER OF PARTICLES=0
      NP = 0
 760  CONTINUE
C HERE IS THE MAIN MENU.  REFER TO ASCII CODES FOR FRAME CHARACTERS.
C
      WRITE(*,*) ' ╔════════════════════ USER MENU ════════════════════╗'
      WRITE(*,*) ' ║     FORMAT:    COMMAND - DESCRIPTION               ║'
      WRITE(*,*) ' ╠════════════════════════╦══════════════════════════╣'
      WRITE(*,*) ' ║  SET  - MASS TRANSPORT ║   INPUT - READ IN FLOW   ║'
      WRITE(*,*) ' ║         COEFFICIENTS   ║           DATA FILE      ║'
      WRITE(*,*) ' ║                        ║   P - ADD PARTICLES      ║'
```

```
      WRITE(*,*) ' ║ RE -  READ DATA          ║      (POLLUTANTS)        ║'
      WRITE(*,*) ' ║ WR -  WRITE DATA         ║                          ║'
      WRITE(*,*) ' ║                          ║ MOVE - MOVE PARTICLES    ║'
      WRITE(*,*) ' ║                          ║        FOR TIME STEP     ║'
      WRITE(*,*) ' ║ XS - CLEAR SINKS         ║                          ║'
      WRITE(*,*) ' ║ XP - CLEAR PARTICLES     ║ MAP - MAP PARTICLES OR   ║'
      WRITE(*,*) ' ║ XT - SET TIME TO 0       ║       CONCENTRATIONS     ║'
      WRITE(*,*) ' ║                          ║                          ║'
      WRITE(*,*) ' ║ TMNGR - SET VELOCITY     ║                          ║'
      WRITE(*,*) ' ║         FILE ENDTIMES    ║                          ║'
      WRITE(*,*) ' ║         (TRANSIENT)      ║ Q - QUIT                 ║'
      WRITE(*,*) ' ╚══════════════════════════╩══════════════════════════╝'
      WRITE(*,*) ' COMMAND '
      READ(*,'(A)') AS
C JUNCTION POINT FOR MENU COMMAND SELECTION.
      IF ((AS.EQ.'P    ').OR.(AS.EQ.'p    ')) CALL GENP
      IF ((AS.EQ.'XP   ').OR.(AS.EQ.'xp   ')) CALL XP
      IF ((AS.EQ.'RE   ').OR.(AS.EQ.'re   ')) CALL REINCD
      IF ((AS.EQ.'XS   ').OR.(AS.EQ.'xs   ')) CALL XS(NS)
      IF ((AS.EQ.'WR   ').OR.(AS.EQ.'wr   ')) CALL WRFNCD
      IF ((AS.EQ.'XT   ').OR.(AS.EQ.'xt   ')) CALL XT(T2)
      IF ((AS.EQ.'TMNGR').OR.(AS.EQ.'tmngr')) CALL TMNGR
      IF ((AS.EQ.'MOVE ').OR.(AS.EQ.'move ').OR.(AS.EQ.'Move ')) THEN
          CALL MOVE
      ENDIF
      IF ((AS.EQ.'Q    ').OR.(AS.EQ.'q    ')) GOTO 1110
      IF ((AS.EQ.'MAP  ').OR.(AS.EQ.'map  ').OR.(AS.EQ.'Map  ')) THEN
         CALL MAP3D
      ENDIF
      IF ((AS.EQ.'SET  ').OR.(AS.EQ.'set  ')) CALL SET
      IF ((AS.EQ.'EDIT ').OR.(AS.EQ.'edit ')) CALL SET
      IF ((AS.EQ.'INPUT').OR.(AS.EQ.'input')) CALL REFLOW
C BOUNCING GOTO
 1100 GOTO 760
 1110 CONTINUE
      STOP
      END

C XT SUBROUTINE
C INITIATE OR ZERO OUT TIME, SINKS, OR PARTICLE POINTERS.
C  ZERO TIME
      SUBROUTINE XT(T2)
 1120 T2=0
 1130 WRITE (*,*)  '      SIMULATION TIME INITIATED  '
 1140 RETURN
      END

C XS SUBROUTINE
C ZERO SINKS
      SUBROUTINE XS(NS)
 1150 NS=0
 1160 WRITE(*,*)  '      NUMBER OF SINKS INITIATED  '
```

```
 1170 RETURN
      END

C XP SUBROUTINE
C ZERO PARTICLES
      SUBROUTINE XP
      REAL LAMDBA,PM
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001),SWITCH(10001)
 1180 DO 1200 I = 1, NP
 1190   SWITCH(I)=0
 1200 CONTINUE
 1210 WRITE(*,*) '      NUMBER OF PARTICLES INITIATED  '
 1220 NP=0
 1230 RETURN
      END

C    ++++++++++++++++++++++++++++++++++++
C    +  BEGIN PARTICLE INPUT SUBROUTINE  +
C    ++++++++++++++++++++++++++++++++++++
      SUBROUTINE GENP
      INTEGER SWITCH(10001),L,C,S,RC,NPRC
      CHARACTER AS
      REAL X2,Y2,Z2,X3,Y3,Z3
      REAL XPASS,YPASS,ZPASS,LLX,LLY,LLZ,LAMDBA,PM
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/GEO/THICK,BOT,TOP
      COMMON/NPNT/RECH,IBOOL,ALT
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION THICK(161,151,4),BOT(161,151,4),TOP(161,151,4)
 1270 WRITE(*,*) '///PARTICLES\\\'
 1280 WRITE(*,*)
 1290 WRITE(10,*) '///PARTICLES\\\'
 1300 WRITE(10,*)
C  INITIALIZE LINE COUNTER
 1310 L=0
C  INITIALIZE CYLINDER COUNTER
 1320 C=0
C  INITIALIZE SPHERE COUNTER
      S=0
C  INITIALIZE RECTANGLE COUNTER
      RC=0
C  POINT RECTANGLE COUNTER INITIALIZED AT GENP CALL
      NPRC=0
      IF ((DELX.LE.0).OR.(DELY.LE.0)) THEN
         WRITE(*,*) '  YOU MUST ENTER A VELOCITY FILE BEFORE
     $              INITIATING PARTICLES'
         RETURN
      ENDIF
 1330 WRITE(*,*)
 1340 WRITE(*,*) '  ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)'
```

```
      READ(*,'(A)') AS
C QUIT OPTION
 1360 IF ((AS.NE.'Y').AND.(AS.NE.'y')) GOTO 2420
 1370 WRITE(*,*) ' CYLINDER, SPHERE, LINE, ',
     $'RECTANGULAR, OR POINT (C, S, L, R, P) '
      READ(*,'(A)') AS
 1380 IF ((AS.EQ.'C').OR.(AS.EQ.'c')) THEN
         GOTO 2400
 1385 ELSEIF ((AS.EQ.'S').OR.(AS.EQ.'s')) THEN
         GOTO 2150
 1390 ELSEIF ((AS.EQ.'L').OR.(AS.EQ.'l')) THEN
         GOTO 1750
 1400 ELSEIF ((AS.EQ.'R').OR.(AS.EQ.'r')) THEN
         GOTO 1420
      ELSEIF ((AS.EQ.'P').OR.(AS.EQ.'p')) THEN
         GOTO 2416
      ELSE
        GOTO 1370
      ENDIF
C ***********************************************
C INITIATE PARTICLES IN RECTANGULAR PRISM
C ***********************************************
 1420 WRITE(*,*) '    PARTICLES IN A RECTANGULAR PRISM ',RC+1
      WRITE(10,*) '    PARTICLES IN A RECTANGULAR PRISM ',RC+1
 1430 WRITE(*,*) '      ENTER COORDINATES OF LOWER LEFT CORNER (X,Y)'
      READ(*,*) X5,Y5
 1440 WRITE(10,*) ' RECTANGULAR PRISM COORDINATES:'
 1450 WRITE(10,*) ' LOWER LEFT CORNER (X,Y) = ',X5,',',Y5,' FT'
 1460 WRITE(*,*) ' ENTER COORDINATES OF UPPER RIGHT CORNER (X,Y)'
      READ(*,*) X6,Y6
 1470 WRITE(10,*) ' UPPER RIGHT CORNER (X,Y) = ',X6,',',Y6,' FT'
 1472 WRITE(*,*) ' ENTER Z COORDINATES (LOWER,UPPER) '
      READ(*,*) Z5,Z6
 1474 WRITE(10,*) ' Z COORDINATES (LOWER,UPPER) = ',Z5,',',Z6,' FT'
 1480 IF (((X6-X5).LT.0).OR.((Y6-Y5).LT.0)) THEN
         WRITE(*,*) 'CANNOT FORM RECTANGLE--REDO'
         GOTO 1430
      ENDIF
 1490 IF ((X5.LE.(LLX)).OR.(X5.GE.(LLX+DELX*(NC)))) THEN
         WRITE(*,*) '  LOWER LEFT CORNER X COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 1430
      ENDIF
 1500 IF ((Y5.LE.(LLY)).OR.(Y5.GE.(LLY+DELY*(NR)))) THEN
         WRITE(*,*) '  LOWER LEFT CORNER Y COORDINATE IS
```

```
    $   OUTSIDE THE MODEL GRID'
        GOTO 1430
     ENDIF
1510 IF ((Y6.LE.(LLY)).OR.(Y6.GE.(LLY+DELY*(NR)))) THEN
         WRITE(*,*) '  UPPER RIGHT CORNER Y COORDINATE IS
    $    OUTSIDE THE MODEL GRID'
         GOTO 1430
     ENDIF
1520 IF ((X6.LE.(LLX)).OR.(X6.GE.(LLX+DELX*(NC)))) THEN
         WRITE(*,*) '  UPPER RIGHT CORNER X COORDINATE IS
    $    OUTSIDE THE MODEL GRID'
         GOTO 1430
     ENDIF
     IZ=INT((X5-LLX)/DELX+1.0)
     JZ=INT((Y5-LLY)/DELY+1.0)
     IF ((Z5.LT.BOT(IZ,JZ,1)).OR.
    $(Z5.GT.BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL))) THEN
         WRITE(*,*) '  1ST Z COORDINATE OUTSIDE MODEL'
         GOTO 1430
     ENDIF
     IF ((Z6.LT.BOT(IZ,JZ,1)).OR.
    $(Z6.GT.BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL))) THEN
         WRITE(*,*) '  2ND Z COORDINATE OUTSIDE MODEL'
         GOTO 1430
     ENDIF
1530 WRITE(*,*) ' Enter the letter C for CONTINUOUS POLLUTION--
    $otherwise a <RETURN>'
1540 WRITE(*,*) ' any other character will assume a SLUG input'
     READ(*,'(A)')  AS
1550 IF ((AS.EQ.'C').OR.(AS.EQ.'c')) THEN
       WRITE(*,*) ' YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE'
       WRITE(10,*) 'YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE'
     ENDIF
1570 WRITE(*,*) ' ENTER NUMBER OF PARTICLES '
     READ(*,*) M
1580 WRITE(10,*) 'NUMBER OF PARTICLES = ',M
1590 WRITE(*,*)
1600 WRITE(*,*) ' DO YOU WANT TO REDO THIS SCREEN?
    $(Y-YES, <RETURN>-NO)'
     READ(*,'(A)') OKS
1610 IF ((OKS.EQ.'Y').OR.(OKS.EQ.'y')) GOTO 1330
1620 DX = X6  - X5
1630 DY = Y6 - Y5
     DZ = Z6 - Z5
1640 DO 1690 I = 1,M
```

```
 1650   XPASS = X5 + DX*RND()
 1660   YPASS = Y5 + DY*RND()
        ZPASS = Z5 + DZ*RND()
C ADD A PARTICLE
 1670   CALL ADD3D(AS,XPASS,YPASS,ZPASS)
 1690 CONTINUE
 1700 WRITE(*,*)
 1710 WRITE(10,*) '      SYSTEM PARTICLES = ',NP
      WRITE(10,*) 'RECTANGLE DONE'
      RC=RC+1
C RETURN TO BEGINNING OF PARTICLE SUBROUTINE
      GOTO 1330
C ****************************************
C   INITIATE PARTICLES ON 3D LINE
C ****************************************
 1730 WRITE(*,*)
 1740 WRITE(10,*)
 1750 WRITE(*,*) '      LINE NUMBER ',L+1
 1760 WRITE(10,*) '          PARTICLES ON A LINE'
 1770 WRITE(10,*) '             LINE NUMBER',L+1
 1780 WRITE(*,*)
 1790 WRITE(*,*) ' ENTER COORDINATES OF LINE BEGINNING (X,Y,Z) '
      READ(*,*) X2,Y2,Z2
 1800 WRITE(10,*) '             LINE COORDINATES:'
 1810 WRITE(10,*) '             BEGINNING POINT (X,Y,Z) = ',X2,',',Y2,
     $',',Z2,' FT'
 1820 WRITE(*,*) ' ENTER COORDINATES OF END OF LINE (X,Y,Z) '
      READ(*,*) X3,Y3,Z3
 1830 WRITE(10,*) '             END POINT OF LINE (X,Y,Z) = ',X3,',',Y3,
     $',',Z3,' FT'
 1840 IF ((X2.LE.(LLX)).OR.(X2.GE.(LLX+DELX*(NC)))) THEN
         WRITE(*,*) '  BEGINNING POINT X COORDINATE IS OUTSIDE THE MODEL
     $   GRID'
         GOTO 1790
      ENDIF
 1850 IF ((Y2.LE.(LLY)).OR.(Y2.GE.(LLY+DELY*(NR)))) THEN
          WRITE(*,*) '  BEGINNING POINT Y COORDINATE IS OUTSIDE THE
     $    MODEL GRID'
          GOTO 1790
      ENDIF
 1860 IF ((Y3.LE.(LLY)).OR.(Y3.GE.(LLY+DELY*(NR)))) THEN
          WRITE(*,*) '  END POINT Y COORDINATE IS OUTSIDE THE
     $    MODEL GRID'
          GOTO 1790
      ENDIF
```

```
1870 IF ((X3.LE.(LLX)).OR.(X3.GE.(LLX+DELX*(NC)))) THEN
         WRITE(*,*) '  END POINT X COORDINATE IS OUTSIDE THE
    $    MODEL GRID'
         GOTO 1790
     ENDIF
     IZ=INT((X2-LLX)/DELX+1.0)
     JZ=INT((Y2-LLY)/DELY+1.0)
     IF ((Z2.LT.BOT(IZ,JZ,1)).OR.(Z2.GT.(BOT(IZ,JZ,NL)+
    $THICK(IZ,JZ,NL)))) THEN
       WRITE(*,*) '  1ST Z COORDINATE OUTSIDE MODEL'
       GOTO 1790
     ENDIF
     IZ=INT((X3-LLX)/DELX+1.0)
     JZ=INT((Y3-LLY)/DELY+1.0)
     IF ((Z3.LT.BOT(IZ,JZ,1)).OR.(Z3.GT.(BOT(IZ,JZ,NL)+
    $THICK(IZ,JZ,NL)))) THEN
       WRITE(*,*) '  2ND Z COORDINATE OUTSIDE MODEL'
       GOTO 1790
     ENDIF
     WRITE(*,*) ' Enter the letter C for CONTINUOUS POLLUTION-
    $-otherwise a <RETURN>'
     WRITE(*,*) ' or any other character will assume a SLUG input'
     IF ((AS.EQ.'C').OR.(AS.EQ.'c')) THEN
       WRITE(*,*) ' YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE'
       WRITE(*,*)
       WRITE(10,*) 'YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE'
     ENDIF
1880 WRITE(*,*) ' ENTER NUMBER OF PARTICLES '
     READ(*,*) M
1890 WRITE(*,*)
1900 WRITE(10,*) '             NUMBER OF PARTICLES = ',M
1910 WRITE(*,*) 'DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)'
     READ(*,'(A)') OKS
1920 IF ((OKS.EQ.'Y').OR.(OKS.EQ.'y')) GOTO 1330
1930 IF (M.LE.0) GOTO 1330
1940 XPASS = X2
     YPASS = Y2
     ZPASS = Z2
C THE NEXT SUB CALL IS REQUIRED TO INSERT THE FIRST PARTICLE.
1960 CALL ADD3D(AS,XPASS,YPASS,ZPASS)
     IF (M.EQ.1) GOTO 2060
1970 IF (M.GT.1) THEN
       XPASS = X3
       YPASS = Y3
       ZPASS = Z3
```

```
C THE NEXT SUB CALL IS REQUIRED TO INSERT THE SECOND PARTICLE.
        CALL ADD3D(AS,XPASS,YPASS,ZPASS)
      ENDIF
 2000 IF (M.EQ.2) GOTO 2060
 2010 DO 2050 I = 3,M
 2020   XPASS = X2+(I-2)*(X3-X2)/(M-1)
 2030   YPASS = Y2+(I-2)*(Y3-Y2)/(M-1)
        ZPASS = Z2+(I-2)*(Z3-Z2)/(M-1)
C ADD ADDITIONAL PARTICLES
 2040   CALL ADD3D(AS,XPASS,YPASS,ZPASS)
 2050 CONTINUE
 2060 WRITE(10,*) '       SYSTEM PARTICLES = ',NP
      WRITE(10,*) '3D LINE DONE'
 2070 L=L+1
C RETURN TO BEGINNING OF PARTICLE SUBROUTINE
 2080 GOTO 1330
C ****************************************
C   INITIATE PARTICLES IN SPHERE
C ****************************************
 2150 WRITE(*,*)
 2160 WRITE(10,*) '    PARTICLES ON A SPHERE '
 2170 WRITE(*,*)  '      SPHERE NUMBER ',S+1
 2180 WRITE(*,*)
 2190 WRITE(10,*) '           SPHERE NUMBER ',S+1
 2200 WRITE(*,*) '   ENTER SPHERE CENTER COORDINATES (X,Y,Z)'
      READ(*,*) X4,Y4,Z4
 2210 WRITE(10,*) ' SPHERE CENTER COORDINATES (X,Y,Z) = '
     $,X4,',',Y4,',',Z4,' FT'
 2230 WRITE(*,*) '  ENTER SPHERE RADIUS (FT) '
      READ(*,*) R
 2240 WRITE(10,*) ' SPHERE RADIUS = ',R,' FT'
 2242 IF (((X4-R).LT.(LLX)).OR.((X4+R).GT.(LLX+DELX*(NC)))) THEN
          WRITE(*,*) '  DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID
     $     - BAD X'
          GOTO 2200
      ENDIF
 2244 IF (((Y4-R).LT.(LLY)).OR.((Y4+R).GT.(LLY+DELY*(NR)))) THEN
          WRITE(*,*) '  DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID
     $     - BAD Y'
          GOTO 2200
      ENDIF
      IZ=INT((X4-LLX)/DELX+1.0)
      JZ=INT((Y4-LLY)/DELY+1.0)
      IF (((Z4-R).LT.BOT(IZ,JZ,1)).OR.((Z4+R).GT.(BOT(IZ,JZ,NL)+
     $THICK(IZ,JZ,NL)))) THEN
```

```
          WRITE(*,*) '  DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID
     $     - BAD Z '
          GOTO 2200
      ENDIF
      WRITE(*,*) ' Enter the letter C for CONTINUOUS POLLUTION--
     $otherwise a <RETURN>'
      WRITE(*,*) ' or any other character will assume a SLUG input'
      READ(*,'(A)') AS
      IF ((AS.EQ.'C').OR.(AS.EQ.'c')) THEN
        WRITE(*,*) ' YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE'
        WRITE(*,*)
        WRITE(10,*) 'YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE'
      ENDIF
 2250 WRITE(*,*) ' ENTER NUMBER OF PARTICLES '
      READ(*,*) M
 2260 WRITE(10,*) '          NUMBER OF PARTICLES = ',M
 2270 WRITE(*,*)
 2280 WRITE(*,*) 'DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)'
      READ(*,'(A)') OKS
 2290 IF ((OKS.EQ.'Y').OR.(OKS.EQ.'y')) GOTO 1330
 2300 ANG=0
 2310 AG=2*3.14159/SQRT(M)
 2320 DO 2380 I=1,INT(SQRT(M))
 2330    TH=AG*I+ANG
         DO 2370 II=1,(M/INT(SQRT(M)))
           PHI=2*3.14159-TH
           XPASS = X4+R*SIN(PHI)*COS(TH)
           YPASS = Y4+R*SIN(TH)*SIN(PHI)
           ZPASS = Z4+R*COS(PHI)
C CALL TO SUBROUTINE TO INSERT A PARTICLE.
           CALL ADD3D(AS,XPASS,YPASS,ZPASS)
 2370    CONTINUE
 2380 CONTINUE
      WRITE(10,*) '      SYSTEM PARTICLES = ',NP
      WRITE(10,*) '        SPHERE DONE'
 2390 S = S + 1
C RETURN TO BEGINNING OF PARTICLE SUBROUTINE
      GOTO 1330
C ********************************************
C   ROUTINE TO GENERATE PARTICLES FROM A VERTICALLY ORIENTED CYLINDER
C ********************************************
 2400 CONTINUE
      WRITE(10,*) '       PARTICLES ON A CYLINDER'
      WRITE(*,*) '      CYLINDER NUMBER ',C+1
      WRITE(*,*)
```

```
      WRITE(10,*) '          CYLINDER NUMBER ',C+1
2410  WRITE(*,*) ' ENTER CYLINDER CENTER COORDINATES (X,Y)'
      READ(*,*) X4,Y4
      WRITE(10,*) ' CYLINDER CENTER COORDINATES (X,Y) =',X4,',',Y4,' FT'
      WRITE(*,*) '      ENTER CYLINDER RADIUS (FT) '
      READ(*,*) R
      WRITE(10,*) ' CYLINDER RADIUS = ',R,' FT'
      WRITE(*,*)  ' ENTER BOTTOM AND TOP ELEVATION OF CYLINDER (FT) '
      READ(*,*) Z3,Z4
      WRITE(10,*) ' BOTTOM AND TOP ELEVATION OF CYLINDER (FT) '
     $,Z3,',',Z4,' FT'
      IF (((X4-R).LT.(LLX)).OR.((X4+R).GT.(LLX+DELX*(NC)))) THEN
          WRITE(*,*) '  DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID
     $      - BAD X'
          GOTO 2410
      ENDIF
      IF (((Y4-R).LT.(LLY)).OR.((Y4+R).GT.(LLY+DELY*(NR)))) THEN
          WRITE(*,*) '  DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID
     $      - BAD Y'
          GOTO 2410
      ENDIF
      IZ=INT((X4-LLX)/DELX+1.0)
      JZ=INT((Y4-LLY)/DELY+1.0)
      IF ((Z4.LT.BOT(IZ,JZ,1)).OR.(Z4.GT.(BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)
     $ ))) THEN
         WRITE(*,*) ' DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID
     $     - BAD TOP '
         GOTO 2410
      ENDIF
      IF ((Z3.LT. BOT(IZ,JZ,1)).OR.(Z3.GT.(BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)
     $ ))) THEN
         WRITE(*,*) ' DO NOT ENTER PARTICLES OUTSIDE FLOW MODEL GRID
     $     - BAD BOT '
         GOTO 2410
      ENDIF
      IF (Z4.LT.Z3) THEN
        WRITE(*,*) '  TOP OF CYLINDER MUST BE HIGHER THAN BOTTOM'
        GOTO 2410
      ENDIF
      WRITE(*,*) ' Enter the letter C for CONTINUOUS POLLUTION--
     $otherwise a <RETURN>'
      WRITE(*,*) 'or any other character will assume a SLUG input'
      READ(*,'(A)') AS
      IF ((AS.EQ.'C').OR.(AS.EQ.'c')) THEN
        WRITE(*,*) ' YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE'
```

```
        WRITE(*,*)
        WRITE(10,*) 'YOU ARE NOW IN THE CONTINUOUS MODE FOR THIS SOURCE'
      ENDIF
      WRITE(*,*) ' ENTER NUMBER OF PARTICLES '
      READ(*,*) M
      WRITE(10,*) ' NUMBER OF PARTICLES = ',M
      WRITE(*,*)
      WRITE(*,*) 'DO YOU WANT TO REDO THIS SCREEN? (Y-YES, <RETURN>-NO)'
      READ(*,'(A)') OKS
      IF ((OKS.EQ.'Y').OR.(OKS.EQ.'y')) GOTO 2410
      ANG=0
      AG=2*3.14159/M
      DO 2415 I=1,M
         TH=AG*I+ANG
         XPASS = X4+R*SIN(TH)
         YPASS = Y4+R*COS(TH)
         ZPASS = Z3+(Z4-Z3)*RND()
C GO TO SUBROUTINE TO INSERT A PARTICLE.
         CALL ADD3D(AS,XPASS,YPASS,ZPASS)
 2415 CONTINUE
      WRITE(10,*) '       SYSTEM PARTICLES = ',NP
      WRITE(*,*) ' CYLINDER DONE'
      C = C+1
C RETURN TO BEGINNING OF PARTICLE SUBROUTINE
      GOTO 1330
C ***********************************************************************
C INITIATE NON-POINT PARTICLES BY POINT LOADING
C ***********************************************************************
C HARDWIRE THE LOADING POINT
C RAND3D COORDINATE (IR,JR)
C 2416 WRITE(*,*) 'ENTER THE RAND3D COLUMN AND ROW OF THE LOADING POINT:'
C      READ(*,*) IR,JR
 2416 IR = 135
      JR = 70
      WRITE(*,*) LLX,LLY,LLZ
C DETERMINE COORDINATES OF LOWER LEFT CORNER (X,Y) OF CELL
      X5 = (IR-1)*DELX + LLX
      Y5 = (JR-1)*DELY + LLY
C          WRITE(10,*) ' RECTANGULAR PRISM COORDINATES:'
C          WRITE(10,*) ' LOWER LEFT CORNER (X,Y) = ',X5,',',Y5,' FT'
C DETERMINE COORDINATES OF UPPER RIGHT CORNER (X,Y) OF CELL'
      X6=X5 + DELX
      Y6=Y5 + DELY
C          WRITE(10,*) ' UPPER RIGHT CORNER (X,Y) = ',X6,',',Y6,' FT'
C          WRITE(*,*) ' ENTER Z COORDINATES (LOWER,UPPER) '
C DETERMINE COORDINATES OF TOP (Z) AND BOTTOM (Z) OF CELL
```

```
C USING WATER TABLE AND 5 FOOT BELOW WATER TABLE
C           Z5=BOT(IR,JR,NL) + THICK(IR,JR,NL) - 5
C           Z6=BOT(IR,JR,NL) + THICK(IR,JR,NL)
C USING BOTTOM
                                                                    OF
LAYER 1 AND 5 FEET ABOVE
C       Z5=BOT(IR,JR,NL) + THICK(IR,JR,NL)-5
C       Z6=BOT(IR,JR,NL) + THICK(IR,JR,NL)
      Z5=BOT(IR,JR,NL) + 5
      Z6=BOT(IR,JR,NL) + THICK(IR,JR,NL) - 5
C WRITE(10,*) ' Z COORDINATES (LOWER,UPPER) = ',Z5,',',Z6,' FT'
      IF (((X6-X5).LT.0).OR.((Y6-Y5).LT.0)) THEN
         WRITE(*,*) 'CANNOT FORM RECTANGLE--REDO'
         GOTO 2418
      ENDIF
      IF ((X5.LE.(LLX)).OR.(X5.GE.(LLX+DELX*(NC)))) THEN
         WRITE(*,*) '  LOWER LEFT CORNER X COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 2418
      ENDIF
      IF ((Y5.LE.(LLY)).OR.(Y5.GE.(LLY+DELY*(NR)))) THEN
         WRITE(*,*) '  LOWER LEFT CORNER Y COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 2418
      ENDIF
      IF ((Y6.LE.(LLY)).OR.(Y6.GE.(LLY+DELY*(NR)))) THEN
         WRITE(*,*) '  UPPER RIGHT CORNER Y COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 2418
      ENDIF
      IF ((X6.LE.(LLX)).OR.(X6.GE.(LLX+DELX*(NC)))) THEN
         WRITE(*,*) '  UPPER RIGHT CORNER X COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 2418
      ENDIF
      IZ=INT((X5-LLX)/DELX+1.0)
      JZ=INT((Y5-LLY)/DELY+1.0)
      IF ((Z5.LT.BOT(IZ,JZ,1)).OR.
     $(Z5.GT.BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL))) THEN
          WRITE(*,*) '  1ST Z COORDINATE OUTSIDE MODEL'
          GOTO 2418
      ENDIF
      IF ((Z6.LT.BOT(IZ,JZ,1)).OR.
     $(Z6.GT.BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL))) THEN
         WRITE(*,*) '  2ND Z COORDINATE OUTSIDE MODEL'
         GOTO 2418
      ENDIF
C CONTINUOUS SOURCING ASSUMED FOR NON-POINT PROBLEM
      AS=C
C DETERMINE NUMBER OF PARTICLES = M
      WRITE(*,*) 'GOING TO MASS'
```

```
      CALL MASSIN(M)
      WRITE(*,*) 'BACK FROM MASS'
C DETERMINE ADDING COORDINATES
      DX = X6  - X5
      DY = Y6 - Y5
      DZ = Z6 - Z5
      DO 2417 I = 1,M
         XPASS = X5 + DX*RND()
         YPASS = Y5 + DY*RND()
         ZPASS = Z5 + DZ*RND()
C ADD A PARTICLE
         CALL ADD3D(AS,XPASS,YPASS,ZPASS)
 2417 CONTINUE
 2418 CONTINUE
      WRITE(10,*) '      SYSTEM PARTICLES = ',NP
      WRITE(10,*) 'POINT LOAD RECTANGLE DONE'
      NPRC=NPRC+1
C RETURN TO BEGINNING OF PARTICLE SUBROUTINE
      GOTO 1330
C SUBROUTINE GENP CLOSING LINES
 2420 WRITE(*,*)
 2430 WRITE(10,*)
 2440 WRITE(10,*) '         TOTAL SYSTEM PARTICLES = ',NP
 2450 WRITE(10,*) '\\\\\///////'
 2460 WRITE(*,*) '         TOTAL SYSTEM PARTICLES = ',NP
 2470 WRITE(*,*) '\\\\\//////'
 2480 RETURN
      END
C *************************************
C  SUBROUTINE TO ADD A PARTICLES
C *************************************
      SUBROUTINE ADD3D(AS,XPASS,YPASS,ZPASS)
      INTEGER SWITCH(10001)
      CHARACTER AS
      REAL XPASS,YPASS,ZPASS,LAMDBA,PM
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
 2100 NP=NP+1
 2110 X(NP)=XPASS
 2120 Y(NP)=YPASS
      Z(NP)=ZPASS
      IF ((AS.EQ.'C').OR.(AS.EQ.'c')) THEN
         SWITCH(NP)=1
      ELSE
         SWITCH(NP)=0
      ENDIF
 2130 RETURN
      END
C  END OF ADD A PARTICLE SECTION OF PARTICLE GENERATOR

C***************************************************************************
C  MOVE ROUTINE !!!!!!!!!!!!!!!!!
```

```
C  MAKE SURE ALL DATA NECESSARY TO MAKE A MOVE HAS BEEN SPECIFIED.
C*****************************************************************************
      SUBROUTINE MOVE
      CHARACTER GS,FS
      CHARACTER*40 RS
      REAL LLX,LLY,LLZ,LAMDBA,I1,J1,I2,J2,MAPCONC(161,151),PM
      INTEGER SWITCH(10001)
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/GEO/THICK,BOT,TOP
      COMMON/FLOW/VI,VJ,VK
      COMMON/SINK/NS,X1,Y1,Z1,QSUM
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/FLAGS/GS,FS,RS
      COMMON/MAPS/NMAP,CONC,MAPCONC,CDX,CDY,I1,J1,I2,J2
      DIMENSION THICK(161,151,4),BOT(161,151,4),TOP(161,151,4)
      DIMENSION VI(161,151,4),VJ(161,151,4),VK(161,151,4)
      DIMENSION X1(5000),Y1(5000),Z1(5000),QSUM(5000)
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION NMAP(161,151),CONC(5000)
      DIMENSION ENDTIME(20)
 2510 IF (GS.NE.'B') THEN
          WRITE(*,*) 'PLEASE CHOOSE COEFFICIENTS MENU ITEM NOW!!!'

          RETURN
      ENDIF
 2515 IF ((NENDTIME.GT.0).AND.(NPER.EQ.NENDTIME)
     $.AND.(T2.EQ.ENDTIME(NPER))) THEN
         WRITE(*,*) '  ALL VELOCITY FILES USED, SIMULATION COMPLETE'

         RETURN
      ENDIF
 2520 IF((NENDTIME.GT.0).AND.(T2.EQ.ENDTIME(NPER))) THEN
          WRITE(*,*) '  READ IN THE NEXT VELOCITY VECTOR FILE NOW!'

          RETURN
      ENDIF
 2530 IF (NP.EQ.0) THEN
         WRITE(*,*) 'DO SOMETHING ELSE-THERE ARE NO PARTICLES TO MOVE!'

         RETURN
      ENDIF
C 2542 IF (ASPECT.LE.0) THEN
C         WRITE(*,*) ' YOU DID NOT ENTER AN ASPECT RATIO FOR THE SCREEN '
C
C         RETURN
C      ENDIF
C 2544 IF (((SCALEDEF.LE.0).AND.(TEMPSCR.EQ.0)).OR.((SCALEX.LE.0).AND.
C     $(TEMPSCR.EQ.1))) THEN
C         WRITE(*,*) '  YOU DID NOT ENTER A DEFAULT SCREEN WIDTH '
C
C         RETURN
```

```
C       ENDIF
 2550 WRITE(*,*) '        PRESENT SIMULATION TIME (DAYS) = ',T2
 2560 WRITE(10,*) '      PRESENT SIMULATION TIME = ',T2,'  DAYS'
 2570 IF (NENDTIME.LT.0) THEN
         WRITE(*,*) '  TIME REMAINING FOR THIS VELOCITY FILE IS '
     $   ,(ENDTIME(NPER)-T2),' DAYS'
      ENDIF
 2580 WRITE(*,*) '  ENTER INCREMENTAL SIMULATION TIME (DAYS) [DELTA]'
      READ(*,*) DELTA
 2590 IF (DELTA.LE.0) THEN
         WRITE(*,*) '   DELTA MUST BE GREATER THAN ZERO!!!',
     $   '---REDO THIS INPUT NOW PLEASE.'
 2600    GOTO 2580
      ENDIF
C ASSUME STEADY STATE IF NUMBER OF PERIODS HAVE NOT BEEN ENTERED
 2602 IF (NENDTIME.LE.0) GOTO 2630
 2604 IF ((DELTA+T2).GT.ENDTIME(NPER)) THEN
         WRITE(*,*) '  TIME STEP TOO LARGE, ENTER A SMALLER TIME STEP'
         DELTA=ENDTIME(NPER)-T2
         GOTO 2580
      ENDIF
 2630 WRITE(10,*) '        INCREMENTAL SIMULATION TIME = ',DELTA,' DAYS'
      WRITE(*,*)  '        INCREMENTAL SIMULATION TIME = ',DELTA,'DAYS'
 2640 WRITE(*,*) ' HOW OFTEN DO YOU WANT TO COMPUTE VELOCITY VECTORS'
      WRITE(*,*) ' ENTER MAXIMUM HORIZONTAL MOVE (FT) [DMAX] '
      READ(*,*) DMAX
      WRITE(*,*) '  ENTER MAXIMUM VERTICAL MOVE (FT) [ZMAX] '
      READ(*,*) ZMAX
 2650 WRITE(*,*) ' DMAX = ',DMAX,' FT   ZMAX = ',ZMAX,' FT'
 2660 WRITE(10,*) 'DMAX = ',DMAX,' FT   ZMAX = ',ZMAX,' FT'
 2670 WRITE(*,*)
 2680 WRITE(10,*)
C  LOOP TO ZERO OUT PURGE WELL CONCENTRATION ARRAY.
 2730 DO 2750 K = 1, NS
 2740   CONC(K)=0
 2750 CONTINUE
C  INITIALIZE PARTICLE COUNTER
 2770 K=0
C  INITIALIZE 1ST ORDER DECAY CHECK
      ZZZ=0
C  START LOOP ON PARTICLE MOVE HERE. (SEE LOOP TERMINUS AT 3800 OR 3860)
 2790 K=K+1
C  EXAMINE SWITCH TO DETECT IF CONTINUOUS POLLUTION OR NOT.
 2830 IF (SWITCH(K).EQ.1) THEN
         D=DELTA*RND()
      ELSE
         D=DELTA
      ENDIF
C  SKIP FIRST ORDER DECAY TO SAVE TIME IF CONSERVATIVE POLLUTION
      IF (LAMDBA.GE.1E10) GOTO 2880
C  FIRST ORDER DECAY IMPLEMENTED IN THESE STATEMENTS
      ZZZ=ZZZ+(1-.5**(D/LAMDBA/365))
```

```
      IF (ZZZ.GE.1) THEN
         GOTO 2832
      ELSE
         GOTO 2880
      ENDIF
 2832 X(K)=X(NP)
      Y(K)=Y(NP)
      Z(K)=Z(NP)
      SWITCH(K)=SWITCH(NP)
      NP=NP-1
      ZZZ=ZZZ-1.0
C IF LAST PARTICLE DECAYS JUMP TO END OF ROUTINE
      IF (K.GT.NP) GOTO 3660
C  LOOP BACK FOR DECAY ON NEXT PARTICLE
      GOTO 2830
C  +++++++++++++++++++++++++++++++++++++++++++
C  +  WHEN ALL PARTICLES ARE PROCESSED, THIS  +
C  +  NEXT STATEMENT RETURNS YOU TO THE MENU  +
C  +++++++++++++++++++++++++++++++++++++++++++
 2880 IF (K.GT.NP) THEN
         T2=T2+DELTA
         GOTO 3660
      ENDIF
 2890 XP=(X(K)-LLX)/DELX+.5
 2900 YP=(Y(K)-LLY)/DELY+.5
 2910 VX=.00001
 2920 VY=.00001
      VZ=1E-10
 2930 I=INT(XP)
 2940 J=INT(YP)
      IZ=INT(XP+.5)
      JZ=INT(YP+.5)
C  FIND LAYER OF PARTICLE AND INTERPOLATE THE VERTICAL VELOCITY
      DO 2941 KLK=1,(NL-1)
      IF ((Z(K).GE.BOT(IZ,JZ,KLK)).AND.(Z(K).LT.TOP(IZ,JZ,KLK))) THEN
        LL=KLK
        AZ=(Z(K)-BOT(IZ,JZ,KLK))/(TOP(IZ,JZ,KLK)-BOT(IZ,JZ,KLK))
        IF (KLK.EQ.1) THEN
           VKBOT=0.0
        ELSE
           VKBOT=VK(IZ,JZ,KLK-1)
        ENDIF
        V3=(AZ*VK(IZ,JZ,KLK)+(1-AZ)*VKBOT)/E
        GOTO 2948
      ENDIF
      IF ((Z(K).GE.TOP(IZ,JZ,KLK)).AND.(Z(K).LT.BOT(IZ,JZ,KLK+1))) THEN
        LL=KLK+NL
        V3=VK(IZ,JZ,KLK)/E
        GOTO 2948
      END IF
 2941 CONTINUE
      IF (Z(K).GE.BOT(IZ,JZ,NL)) THEN
```

```
      LL=NL
      AZ=(Z(K)-BOT(IZ,JZ,NL))/THICK(IZ,JZ,NL)
      V3=(AZ*VK(IZ,JZ,NL)+(1-AZ)*VK(IZ,JZ,NL-1))/E
    ENDIF
C  IF IN CONFINING LAYER, THEN SKIP HORIZONTAL INTERPOLATION
 2948 IF (LL.GT.NL) THEN
       V1=0
       V2=0
       GOTO 2990
    ENDIF
 2950 AX=XP-INT(XP)
 2960 AY=YP-INT(YP)
C HORIZONTAL VELOCITY INTERPOLATION
 2970 V1=((1-AY)*VI(I,J,LL)+AY*VI(I,J+1,LL))/E
 2980 V2=((1-AX)*VJ(I,J,LL)+AX*VJ(I+1,J,LL))/E
 2990 VX=VX+V1
 3000 VY=VY+V2
C APPLY RETARDATION COEFFICIENTS TO VELOCITY
    VX=VX/RETARD(LL)
    VY=VY/RETARD(LL)
    VZ=(VZ+V3)/RETARD(LL)
 3040 IF ((D-ABS(DMAX/VX)).LT.0) GOTO 3070
 3050 F=ABS(DMAX/VX)
 3060 GOTO 3080
 3070 F=D
 3080 IF ((F-ABS(DMAX/VY)).LT.0) GOTO 3092
 3090 F=ABS(DMAX/VY)
 3092 IF ((F-ABS(ZMAX/VZ)).LT.0) GOTO 3100
    F=ABS(ZMAX/VZ)
C  CALCULATE REMAINING TIME IN MOVE
 3100 D=D-F
 3110 DX=VX*F
 3120 DY=VY*F
    DZ=VZ*F
 3130 PHI=ATAN(DY/DX)
 3140 DD=SQRT(DX*DX+DY*DY)
    DDX=SQRT(DX*DX+DY*DY+DZ*DZ)
 3210 RN=0
 3220 GOTO 3280
C RANDOM NUMBER SUBROUTINE REMOVED HERE
 3280 IF (DL.GT.0) CALL RAND(RN)
 3290 RL=(SQRT(2*DL*DD)/DD)*RN
 3300 IF (DT.GT.0) CALL RAND(RN)
 3310 RT=(SQRT(2*DT*DD)/DD)*RN
 3312 IF (DV.GT.0) CALL RAND(RN)
 3314 RV=(SQRT(2*DV*DDX))*RN
 3340 OLDX = X(K)
 3350 OLDY = Y(K)
    OLDZ = Z(K)
 3360 IF (I.GE.NC) THEN
       XP=NC+.5
       GOTO 3380
```

```
      ENDIF
 3370 XP=XP+(DX+RL*DX+RT*DY)/DELX
 3380 IF (J.GE.NR) THEN
        YP=NR+.5
        GOTO 3400
      ENDIF
 3390 YP=YP+(DY+RL*DY-RT*DX)/DELY
 3400 IF (XP.LE.1) XP=1.01
 3410 IF (XP.GE.NC-.01) XP=NC-.01
 3420 IF (YP.GE.NR-.01) YP=NR-.01
 3430 IF (YP.LT.1) YP=1.01
C  MOVE PARTICLE IN Z DIMENSION
      Z(K)=Z(K)+(DZ+RV)
C  CHECK TO SEE IF PARTICLE HAS BEEN MOVED ABOVE WATER TABLE
      IF (Z(K).GT.(BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL))) THEN
        Z(K)=BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)-.001
      ENDIF
C  CHECK TO SEE IF PARTICLE HAS BEEN MOVED BELOW BOTTOM
      IF (Z(K).LT.BOT(IZ,JZ,1)) Z(K)=BOT(IZ,JZ,1)+.001
C  THESE 4 STATEMENTS MOVE THE PARTICLE OUT OF A ZERO VELOCITY FIELD
C  IF THEY HAPPEN TO GET THERE
 3432 III=INT(XP)
 3434 JJJ=INT(YP)
C  SKIP THIS CHECK IF PARTICLE IN CONFINING LAYER
      IF (LL.GT.NL) GOTO 3440
C MOVE PARTICLE OUT OF ZERO VELOCITY GRID
 3436 IF ((VI(III,JJJ,LL).EQ.0).AND.(VJ(III,JJJ,LL).EQ.0).AND.
     $(VI(III,JJJ+1,LL).EQ.0).AND.(VJ(III+1,JJJ,LL).EQ.0)) THEN
          CALL ZEROVG(XP,YP,III,JJJ,LL,NC,NR)
      ENDIF
C  TRANSLATE PARTICLE POSITION BACK TO REAL COORDINATES
 3440 NEWX = (XP-.5)*DELX+LLX
 3450 NEWY = (YP-.5)*DELY+LLY
 3460 X(K)=NEWX
 3470 Y(K)=NEWY
C THIS CODE REMOVES THE PARTICLE IF NEAR A SINK
C
C  IF PARTICLE IS IN CONFINING LAYER SKIP THE SINK CAPTURE
      IF (LL.GT.NL) THEN
         GOTO 3580
      ENDIF
 3510 DO 3570 LP=1,NS
C  FIND ROW AND COLUMN OF SINK
 3520      II=INT((X1(LP)-LLX)/DELX+1.0)
 3525      JJ=INT((Y1(LP)-LLY)/DELY+1.0)
C  FIND LAYER OF SINK
           KK=Z1(LP)
 3540     IF ((X(K).EQ.X1(LP)).AND.(Y(K).EQ.Y1(LP)).AND.(Z(K).LT.
     $    TOP(II,JJ,KK)).AND.(Z(K).GE.BOT(II,JJ,KK))) THEN
            GOTO 3600
C  REMOVE PARTICLE
         ENDIF
```

```
 3530     R3=SQRT((X(K)-X1(LP))*(X(K)-X1(LP))+(Y(K)-Y1(LP))
     $    *(Y(K)-Y1(LP)))
C  CHECK IF PARTICLE REACHES SINK DURING TIME STEP
C  PARTICLE MUST BE WITHIN MAX MOVE OF WELL TO BE CAPTURED
          CAPTUREMAX=DMAX
C  SET BOTTOM OF LAYER AS CAPTURE ZONE - ASSUMES FULL PENETRATION
          BOTCAPTURE=BOT(II,JJ,KK)
C   SET TOP OF LAYER AS CAPTURE ZONE
 3560     IF (KK.LT.NL) THEN
 3561        TOPCAPTURE=TOP(II,JJ,KK)
          ELSE
C  IF RIVER IN TOP LAYER THEN MODIFY CAPTURE ZONE
 3562        TOPCAPTURE=BOT(II,JJ,KK)+THICK(II,JJ,KK)
 3563        IF (TOP(II,JJ,NL).NE.0) THEN
 3564            R4=QSUM(LP)*F/DELX/DELY/7.48/E
                 BOTCAPTURE=TOP(II,JJ,NL)-R4
 3565            CAPTUREMAX=SQRT(DELX*DELY)
                 IF (BOTCAPTURE.LT.BOT(II,JJ,NL)) THEN
                      BOTCAPTURE=BOT(II,JJ,NL)
                 ENDIF
             ENDIF
          ENDIF
C  SKIP CAPTURE IF INJECTION WELL
 3566 IF (QSUM(LP).LE.0) THEN
          GOTO 3570
      ENDIF
C  REMOVE PARTICLE IN SINK
 3567 IF ((R3.LT.SQRT(QSUM(LP)*DELTA/(3.141592*THICK(II,JJ,KK)*E*7.48)))
     $.AND.(R3.LT.CAPTUREMAX).AND.(Z(K).GE.BOTCAPTURE).AND.
     $(Z(K).LE.TOPCAPTURE)) GOTO 3600
 3570 CONTINUE
C  LOOP BACK TO FINISH MOVE FOR REMAINING TIME INCREMENT
 3580 IF (D.GT.0) GOTO 2890
C LOOP BACK FOR NEXT PARTICLE
 3590 GOTO 2790
C!!!
 3600 WRITE(10,*) 'PARTICLE EXITED AT SINK NUMBER ',LP
C CALC CONCENTRATION ENTERING SINK
 3610 CONC(LP)=CONC(LP)+(PM*119872)/(QSUM(LP)*DELTA)
C  REMOVE PARTICLE BY
 3620 X(K)=X(NP)
C  PUTTING LAST PARTICLE IN POSITION
 3630 Y(K)=Y(NP)
C  OF REMOVED PARTICLE
      Z(K)=Z(NP)
 3640 NP=NP-1
C  LOOP BACK TO NEXT PARTICLE
 3650 GOTO 2830
C  END OF PARTICLE MOVE AND BEGIN PRINTOUT OF SINK CONCENTRATIONS.
 3660 CONTINUE
 3670 WRITE(10,*) 'NP=',NP
 3690 DO 3715 I= 1,NS
```

```
 3700   IF ((QSUM(I).GT.0).AND.(CONC(I).GT.0)) THEN
          WRITE(10,*) 'CONCENTRATION IN PUMPED WELL NUMBER ',I,
     $    ', IN PPM, IS= ',CONC(I)
        ENDIF
 3710   IF ((QSUM(I).GT.0).AND.(CONC(I).GT.0)) THEN
          WRITE(10,*) 'MASS EXITING IN PUMPED WELL NUMBER ',I,
     $    ', IN LBS, IS= ',CONC(I)*QSUM(I)*DELTA/119872
        ENDIF
 3715 CONTINUE
 3720 WRITE(10,*)
      WRITE(10,*)
      WRITE(10,*)
      WRITE(10,*)
C END OF MOVE SUBROUTINE
C SET ALL PARTICLES TO SLUG TYPE AT END OF MOVE
 3830 DO 3850 I=1,NP
 3840  SWITCH(I)=0
 3850 CONTINUE
C   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C   +  BUG OUT TO MAIN MENU AS THIS IS THE END OF SUBROUTINE MOVE.  +
C   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 3910 RETURN
      END
C ********************************
C  NORMALLY DISTRIBUTED RANDOM NUMBER GENERATOR
C ********************************
      SUBROUTINE RAND(RN)
 3230 RN=-6
 3240 DO 3260 I=1,12
 3250     RN=RN+RND()
 3260 CONTINUE
 3270 RETURN
      END
C ********************************
C  UNIFORMLY DISTRIBUTED RANDOM NUMBER GENERATOR
C ********************************
      FUNCTION RND()
      REAL RNDY
      CALL RANDOM(RNDY)
      RND=RNDY
      RETURN
      END
C **********************************
C  ++++++++++++++++++++++++++++++
C  +       MAP PARTICLES OR       +
C  +       CONCENTRATION          +
C  ++++++++++++++++++++++++++++++
      SUBROUTINE MAP3D
      CHARACTER CS,CCS,GS,FS,MAPTYPES
      CHARACTER*40 RS
      INTEGER WD,LN,III,JJJ,LAYER,LL
      REAL LLX,LLY,LLZ,LAMDBA,I1,J1,I2,J2
```

```
      REAL PPP,MAXY,MINY,MAPCONC(161,151),PM
      INTEGER SWITCH(10001)
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/GEO/THICK,BOT,TOP
      COMMON/FLOW/VI,VJ,VK
      COMMON/SINK/NS,X1,Y1,Z1,QSUM
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/FLAGS/GS,FS,RS
      COMMON/MAPS/NMAP,CONC,MAPCONC,CDX,CDY,I1,J1,I2,J2
      DIMENSION THICK(161,151,4),BOT(161,151,4),TOP(161,151,4)
      DIMENSION VI(161,151,4),VJ(161,151,4),VK(161,151,4)
      DIMENSION X1(5000),Y1(5000),Z1(5000),QSUM(5000)
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION NMAP(161,151),CONC(5000)
      DIMENSION ENDTIME(20)
C DEFINE GRID SPACING AS FUNCTION OF LENGTH, WIDTH, AND COORDINATES
C OF MAP AREA.  DEFAULTS SET MAP AREA TO FULL GRID
C
C WD--COLUMNS OF MAP, LN--LINES OF MAP
C CDX--X GRID SPACING, CDY--Y GRID SPACING
C (I1,J1)--(X,Y) COORDINATE OF LL CORNER OF MAP
C (I2,J2)--(X,Y) COORDINATE OF UR CORNER OF MAP
C 4020 J2 = J1 + ((WD - 1) * CDY)
C 4030 I2 = I1 + ((LN - 1) * CDX)
C OR
C 4020 CDX=(I2-I1)/WD
C 4030 CDY=(J2-J1)/LN
C
C SET DEFAULTS TO SUB-GRID  (HARDWIRED)
C WD = NC, NUMBER OF COLUMNS OF SUB-GRID OUT OF PREMOD3D
C LN = NR, NUMBER OF ROWS OF SUB-GRID OUT OF PREMOD3D
      WD=NC
      LN=NR
      CDX=DELX
      CDY=DELY
C REAL I1 AND J1 REAL COORDINATE OF LOWER LEFT CORNER OF SUB-GRID
      J1=LLY
      I1=LLX
C DON/TOM/PETER INVERTED ROW NUMBERING
C RAND3D (1,1) AT MODFLOW (1,NROW); ARRAYS INDEXED WITH (II,JJ)
C THEREFORE (I1,J1) AND (I2,J2) FORM WINDOW IN
C (X,Y) CARTESIAN FRAME OF RAND3D
 4020 J2 = J1 + ((LN - 1) * CDY)
 4030 I2 = I1 + ((WD - 1) * CDX)
C DEBUG
      WRITE(*,*) WD,LN,I1,J1,I2,J2

 4170   WRITE(*,*) ' ENTER LAYER OF MODEL TO MAP.  FOR ALL PARTICLES IN
     $  ALL LAYERS VISIBLE, ENTER ZERO: '
        READ(*,*) LAYER
C  CHECK SELECTION
```

```
        IF (LAYER.GT.(NL+(NL-1))) GOTO 4170

C   CHOICES SELECTED, NOW PRINT THE MAP
C  +++++++++++++++++
C  + WRITE(*,*) THE MAP +
C  +++++++++++++++++
C  INITIALIZE THE ARRAY
 4250 DO 4290 J = 1 , LN
 4260   DO 4280 I = 1 , WD
 4270     NMAP(I,J)=0
          MAPCONC(I,J)=0.0
 4280   CONTINUE
 4290 CONTINUE
C TEST EACH PARTICLE TO SEE WHERE IT FALLS IN THE MAPPED AREA
 4300 DO 4370 MMP = 1,NP
C SKIP PARTICLE IF NOT CONTINUOUS
          IF (SWITCH(MMP).EQ.1) GOTO 4370

C PLAN VIEW MAPS BLOCK
C  FIND COLUMN OF PARTICLE
          IZ=INT((X(MMP)-LLX)/DELX+1.0)
C  FIND ROW OF PARTICLE
          JZ=INT((Y(MMP)-LLY)/DELY+1.0)
C  BEGIN LAYER TESTING FOR MAPS
C  SKIP LAYER TESTING IF ALL PARTICLES ARE TO BE INCLUDED
          IF (LAYER.EQ.0) GOTO 4310
C  FIND TOP LAYER OF PARTICLE
C  IF Z >= TO BOTTOM OF TOP LAYER (NL), THEN LL=NL
          IF (Z(MMP).GE.BOT(IZ,JZ,NL)) THEN
            LL=NL
            GOTO 4305
          ENDIF
C  FIND LAYER OF PARTICLE
C  IF Z BETWEEN TOP AND BOT OF LAYER KLK, LL=KLK
          DO 4301 KLK=1,(NL-1)
            IF((Z(MMP).GE.BOT(IZ,JZ,KLK)).AND.(Z(MMP).LT.
     $       TOP(IZ,JZ,KLK))) THEN
              LL=KLK
              GOTO 4305
            ENDIF
C  FIND CONFINING LAYER OF PARTICLE
C  IF Z BETWEEN TOP OF LAYER KLK AND BOT OF KLK+1, LL=KLK+NL
            IF ((Z(MMP).GE.TOP(IZ,JZ,KLK)).AND.(Z(MMP).LT.
     $       BOT(IZ,JZ,KLK+1))) THEN
                LL=KLK+NL
                GOTO 4305
            ENDIF
 4301     CONTINUE
C SKIP IF IN WRONG LAYER
 4305     IF (LL.NE.LAYER) GOTO 4370
C INCREMENT PARTICLE IF IN CORRECT LAYER
 4310     CONTINUE
```

```
          NMAP(IZ,JZ)=NMAP(IZ,JZ) + 1
 4370 CONTINUE
C  COMPUTE THICKNESS (TEMPTHCK) AND CALCULATE MASS IN CELL
 4480 DO 4590 J = 1,LN
 4490   DO 4580 I = 1,WD
C  FIND INDICES OF THICKNESS GRID FOR MIDDLE OF THIS MAP GRID
C  NOTE I,J INDICES ARE FROM RAND3D (1,1)
C  II,JJ ARE FROM RAND3D (1,1)
C  THEREFORE II=I AND JJ=J
 4500     XX=(I-1)*CDX+I1
 4510     YY=(J-1)*CDY+J1
 4540     II=INT((XX-LLX)/DELX+1.0)
 4550     JJ=INT((YY-LLY)/DELY+1.0)
C PLAN VIEW MAP SUBBLOCK FOR CALCULATING TEMPTHCK FOR
C CONCENTRATION CALCULATION
C IF OUTSIDE GRID GO TO END OF LOOP
            IF ((II.LT.1).OR.(II.GT.WD)) GOTO 4580
            IF ((JJ.LT.1).OR.(JJ.GT.LN)) GOTO 4580
            TEMPTHCK=0
            IF (LAYER.EQ.0) THEN
              DO 4555 K=1,NL-1
                TEMPTHCK=TEMPTHCK+THICK(II,JJ,K)*RETARD(K)
                TEMPTHCK=TEMPTHCK+(BOT(II,JJ,K+1)-TOP(II,JJ,K))
     $          *RETARD(K+NL)
 4555         CONTINUE
              TEMPTHCK=TEMPTHCK+THICK(II,JJ,NL)*RETARD(NL)
            ELSEIF (LAYER.LE.NL) THEN
              TEMPTHCK=THICK(II,JJ,LAYER)
            ELSE
              TEMPTHCK=BOT(II,JJ,LAYER-NL+1)-TOP(II,JJ,LAYER-NL)
            END IF
C THE FOLLOWING HAS BEEN MOVED TO PROGRAM MASSOUT
C MASSOUT CALCULATES 018 MASS ABOVE BASE FROM DEL18
C FOR AQUIFER OF INTEREST
C
C CALCULATE CONCENTRATION
C 4560     IF (NMAP(I,J).LT.0) GOTO 4580
C          V=E*CDX*CDY*TEMPTHCK
C SET ISOTOPIC COMPOSITION OF BASE WATER
C LIGHTEST IN SYSTEM ON METEORIC WATER LINE
C          AMUKG=
C          DEL18BS=-16.00
C          DELDBS=8*DEL18BS + 10
C CALCULATE BASE MASSES OF O18,O16,D,H IN KG
C          CALL AMUMASS(V,DEL18BS,DELDBS,O18,O16,D,H)
C          BS18=O18*AMUKG
C          BS16=O16*AMUKG
C          BSD=D*AMUKG
C          BSH=H*AMUKG
C ADD PARTICLE MASS TO BASE MASS FOR DEL18 MASS??????
C          DEL18=BS18 + NMAP(I,J)*PM
C  CONCENTRATION CALCULATION
```

```
C 4570      MAPCONC(I,J)=(16030*NMAP(I,J)*PM)/(E*CDX*CDY*TEMPTHCK)*PPP
C CALCULATE CELL MASS
 4560   MAPCONC(I,J)=NMAP(I,J)*PM
 4580   CONTINUE
 4590 CONTINUE

C WRITE ALL MAPS
 4600 WRITE(*,*) 'ACCUMULATED TIME = ',T2,' DAYS',' PARTICLES= ',NP
      WRITE(10,*) '  MAP AT TIME =',T2,' DAYS OF '
      WRITE(*,*) '   TOP VIEW   LAYER=',LAYER
      WRITE(10,*) '   TOP VIEW   LAYER=',LAYER
      IF (LAYER.EQ.0) THEN
        WRITE(*,*) 'ALL'
        WRITE(10,*) 'ALL'
      ELSE
        WRITE(*,*) LAYER
        WRITE(10,*) LAYER
      ENDIF
C WRITE NMAP, INVERT SO OUTPUT IS IN MAP ORIENTATION
 4615    DO 4617 J = LN,1,-1
           WRITE(10,'(15I5)') (NMAP(I,J),I=1,WD)
 4617    CONTINUE
C WRITE MAPCONC,INVERT SO OUTPUT IS IN MAP ORIENTATION
      WRITE(10,*) ' CELL MASS ABOVE BASE MASS (IN KG) '
      DO 4640 J = LN,1,-1
         WRITE(10,'(8E10.5)') (MAPCONC(I,J),I=1,WD)
 4640 CONTINUE
 4670 CONTINUE
 5050 WRITE(*,*)
 5060 WRITE(*,*) 'X AND Y COORDINATES'
 5070 WRITE(*,*) ' SHOWN ARE IN FEET'
 5080 WRITE(*,*) '    FROM ORIGIN'
 5100 RETURN
      END


C ************************************************
C  Subroutine to read plume data from external file
C ************************************************
      SUBROUTINE REINCD
      CHARACTER GS,FS
      CHARACTER*40 FILES,RS
      REAL LLX,LLY,LLZ,LAMDBA,I1,J1,I2,J2,MAPCONC(161,151),PM
      INTEGER SWITCH(10001)
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/FLAGS/GS,FS,RS
      COMMON/MAPS/NMAP,CONC,MAPCONC,CDX,CDY,I1,J1,I2,J2
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION ENDTIME(20)
      DIMENSION NMAP(161,151),CONC(5000)
```

```
 5120 WRITE(*,*) ' ENTER FILENAME CONTAINING PLUME DATA '
 5130 WRITE(*,*) ' Enter filename, including extension: '
      READ(*,'(A)') FILES
 5140 OPEN (11,FILE=FILES,STATUS='UNKNOWN')
      WRITE(10,*)
      WRITE(10,*) '  READING IN PLUME DATA FROM EXTERNAL FILE ',FILES
      WRITE(10,*)
 5150 READ(11,'(A)') AS
C  INPUT NAME OF FLOW DATA FILE
 5160 READ(11,'(A)') RS
C  INPUT NAME OF SPF FILE
 5165 READ(11,'(A)') SPFSAVES
 5170 READ(11,*) T2
      READ(11,*) E
      READ(11,*) DL
      READ(11,*) DT
      READ(11,*) DV
      READ(11,*) PM
      READ(11,*) LAMDBA
      READ(11,*) I1
      READ(11,*) J1
      READ(11,*) CDX
      READ(11,*) CDY
      READ(11,*) NP
      READ(11,*) I2
      READ(11,*) J2
C      READ(11,*) O
      READ(11,*)
C      READ(11,*) SCALEX
      READ(11,*)
      READ(11,*) NL
C      READ(11,*) TEMPSCR
      READ(11,*)
C      READ(11,*) MAPTYPES
      READ(11,*)
C      READ(11,*) JROWXS
      READ(11,*)
C      READ(11,*) ICOLXS
      READ(11,*)
C  READ IN RETARDATION COEFFICIENTS
C  FIX:  !!!!!!!!!!!!!!! IMPLIED READ ON ONE LINE !!!!!!!!!!!!!!!!
      DO 5175 I=1,NL+(NL-1)
        READ(11,*) RETARD(I)
 5175 CONTINUE
 5180 IF (NP.EQ.0) GOTO 5230
 5190 DO 5220 I = 1, NP
 5200   READ(11,*) X(I),Y(I),Z(I),SWITCH(I)
 5220 CONTINUE
 5230 CLOSE (11)
C  OPEN AND READ IN DATA FROM FLOW DATA FILE
 5240 CALL REFLOW
C  OPEN AND READ SPECIAL FEATURE FILE
```

```
C 5245 IF (LEN(SPFSAVES).GT.0) GOSUB 12290
  5250 GS='B'
  5260 FS='A'
 5270 RETURN
      END
C ******************************************************
C  Subroutine to write data to external file
C ******************************************************
      SUBROUTINE WRFNCD
      CHARACTER GS,FS
      CHARACTER*40 FILES,RS
      REAL LLX,LLY,LLZ,LAMDBA,I1,J1,I2,J2,MAPCONC(161,151),PM
      INTEGER SWITCH(10001)
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/FLAGS/GS,FS,RS
      COMMON/MAPS/NMAP,CONC,MAPCONC,CDX,CDY,I1,J1,I2,J2
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION NMAP(161,151),CONC(5000)
      DIMENSION ENDTIME(20)
      WRITE(*,*) 'ENTER THE FILENAME TO STORE PLUME DATA'
 5290 WRITE(*,*) ' Enter the name of the external file to be opened,
     $ including ext. '
      READ(*,'(A)') FILES
 5320 OPEN (12,FILE=FILES,STATUS='UNKNOWN')
      WRITE(10,*)
      WRITE(10,*) '  WRITING PLUME DATA TO EXTERNAL FILE ',FILES
      WRITE(10,*)
 5330 WRITE(12,'(A)') 'name: ',FILES
C WRITE NAME OF FLOW DATA FILE
  5340 WRITE(12,*) RS
C WRITE NAME OF SPECIAL FEATURES FILE
C 5345 WRITE(12,*) SPFSAVES
      WRITE(12,*)
 5350 WRITE(12,*) T2
      WRITE(12,*) E
      WRITE(12,*) DL
      WRITE(12,*) DT
      WRITE(12,*) DV
      WRITE(12,*) PM
      WRITE(12,*) LAMDBA
      WRITE(12,*) I1
      WRITE(12,*) J1
      WRITE(12,*) CDX
      WRITE(12,*) CDY
      WRITE(12,*) NP
      WRITE(12,*) I2
      WRITE(12,*) J2
C      WRITE(12,*) O
      WRITE(12,*)
C      WRITE(12,*) SCALEX
```

```
      WRITE(12,*)
      WRITE(12,*) NL
C      WRITE(12,*) TEMPSCR
      WRITE(12,*)
C?       WRITE(12,*) CHRS(34)
C      WRITE(12,*) MAPTYPES
      WRITE(12,*)
C?       WRITE(12,*) CHRS(34)
C      WRITE(12,*) JROWXS
      WRITE(12,*)
C      WRITE(12,*) ICOLXS
      WRITE(12,*)
C  WRITE RETARDATION COEFFICIENTS
      DO 5355 I=1, NL+(NL-1)
        WRITE(12,*) RETARD(I)
 5355 CONTINUE
      WRITE(12,*)
 5360 IF (NP.EQ.0) GOTO 5410
C WRITE PARTICLE LOCATIONS
 5370 DO 5400 I=1,NP
 5380   WRITE(12,*) X(I),Y(I),Z(I),SWITCH(I)
 5400 CONTINUE
 5410 CLOSE (12)
 5420 RETURN
      END

C ********************************************************
C  SUBROUTINE FOR SETTING COEFFICIENTS
C ********************************************************
C
C  SET COEFFICIENT FLAG
      SUBROUTINE SET
      CHARACTER GS,FS
      CHARACTER*40 RS
      INTEGER SWITCH(10001)
      REAL LLX,LLY,LLZ,LAMDBA,PM
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/FLAGS/GS,FS,RS
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
 5490 WRITE(*,*)
      GS='B'
 5500 WRITE(*,*) '//////BASIC TRANSPORT COEFFICIENTS\\\\\'
 5510 WRITE(*,*) '   ENTER POROSITY [E] '
      READ(*,*)  E
 5520 WRITE(10,*) ' POROSITY = ',E
      IF ((E.LE.0).OR.(E.GT.1)) THEN
         WRITE(*,*) '  BAD POROSITY, PLEASE REENTER '
         GOTO 5510
      ENDIF
 5560 WRITE(*,*) ' ENTER PARTICLE MASS (LBS/PARTICLE) [PM] '
      READ(*,*) PM
```

```
 5570 WRITE(10,*) '  PARTICLE MASS (LBS/PARTICLE) = ',PM
      IF (PM.LE.0) THEN
         WRITE(*,*) '  BAD PARTICLE MASS, PLEASE REENTER '
         GOTO 5560
      ENDIF
 5580 WRITE(*,*) ' ENTER LONGITUDINAL DISPERSIVITY (FT) [DL] '
      READ(*,*) DL
 5590 WRITE(10,*) ' LONGITUDINAL DISPERSIVITY (FT)=', DL
 5592 WRITE(*,*) '     ENTER TRANSVERSE DISPERSIVITY (FT) [DT] '
      READ(*,*) DT
 5594 WRITE(10,*) ' TRANSVERSE DISPERSIVITY (FT)=',DT
      WRITE(*,*) '  ENTER VERTICAL DISPERSIVITY (FT) [DV] '
      READ(*,*)  DV
      WRITE(10,*) ' VERTICAL DISPERSIVITY (FT)=',DV
      WRITE(*,*) ' ENTER HALF-LIFE FOR FIRST ORDER DECAY (YEARS)'
      WRITE(*,*) '(ZERO FOR NONE) [LAMDBA] '
      READ(*,*) LAMDBA
      IF (LAMDBA.LE.0) LAMDBA=1E10
      WRITE(10,*) ' HALF-LIFE FOR FIRST ORDER DECAY (YEARS)=',LAMDBA
      DO 5635 I=1,NL+(NL-1)
C  SET DEFAULT VALUES
        IF (RETARD(I).LE.0) RETARD(I)=1
 5630   WRITE(*,*) ' ENTER RETARDATION COEFFICIENT [RETARD] FOR
     $   LAYER ',I
        READ(*,*) RETARD(I)
        IF (RETARD(I).LT.1) THEN
           WRITE(*,*) '  BAD RETARDATION COEFFICIENT, PLEASE REENTER '
           GOTO 5630
        ENDIF
        IF (I.LE.NL) THEN
           WRITE(10,*) 'RETARDATION COEFFICIENT FOR LAYER ',I
           WRITE(10,*) ' IS ',RETARD(I)
        ENDIF
        IF (I.GT.NL) THEN
           WRITE(10,*) 'RETARDATION COEFFICIENT FOR
     $       CONFINING LAYER ',I-NL
           WRITE(10,*) ' IS ',RETARD(I)
        ENDIF
 5635 CONTINUE
 5640 WRITE(*,*)
 5660 RETURN
      END
C  ****************************************************
C   READ VELOCITY AND THICKNESS ARRAY DATA
C  ****************************************************
      SUBROUTINE REFLOW
      CHARACTER*40 FILES
      REAL LLX,LLY,LLZ
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/GEO/THICK,BOT,TOP
      COMMON/FLOW/VI,VJ,VK
      COMMON/SINK/NS,X1,Y1,Z1,QSUM
```

```
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      DIMENSION THICK(161,151,4),BOT(161,151,4),TOP(161,151,4)
      DIMENSION VI(161,151,4),VJ(161,151,4),VK(161,151,4)
      DIMENSION X1(5000),Y1(5000),Z1(5000),QSUM(5000)
      DIMENSION ENDTIME(20)
 9210 WRITE(*,*)
 9220 WRITE(*,*) 'Enter the name of the external .RND file for input:'
      READ(*,'(A)') FILES
      OPEN(11,FILE=FILES,STATUS='UNKNOWN')
      WRITE(*,*)  'NOW PROCESSING FILE ',FILES
      WRITE(10,*) 'INPUT VELOCITY FILE ',FILES
C 9280 READ(11,*) CTIME,TIMESTEP
C 9290 READ(11,*) NSTEPS,DE,PLASMER,KS
C INCREMENT PERIOD COUNTER
 9292 NPER=NPER+1
C INPUT GRID CONSTANTS FOR FLOW MODEL
 9300 READ(11,'(3I3,5F10.1)') NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
C INPUT THICKNESSES (FT) AND VELOCITIES (FT/DAY)
 9310 DO 9360 I=1,NC
 9320   DO 9350 J=1,NR
 9322       DO 9342 K=1,NL
 9330           READ(11,*)II,JJ,KK,VTHICK,VVI,VVJ,VVK,TBOT,TTOP
 9340           THICK(II,JJ,KK)=VTHICK
                VI(II,JJ,KK)=VVI
                VJ(II,JJ,KK)=VVJ
                VK(II,JJ,KK)=VVK
 9341           BOT(II,JJ,KK)=TBOT
                TOP(II,JJ,KK)=TTOP
 9342       CONTINUE
 9350   CONTINUE
 9360 CONTINUE
C 9370 DO 9390 I=1,NC+1
C 9380     READ(11,*) DELX(I)
C 9390 CONTINUE
C 9400 DO 9420 J=1,NR+1
C 9410    READ(11,*) DELY(J)
C 9420 CONTINUE
C INPUT NUMBER OF SINKS
C 9430 READ(11,*) NS
C INITIALIZE SINK COUNTER AND LABEL SINK LIST
 9450 WRITE(10,*) 'Sink # ','I coordinate','J coordinate',
     $'K coordinate',' Withdrawal rate'
 9458 I=0
C INPUT SINKS AS COORDINATES, AND DISCHARGE (GPD)
 9460 I=I+1
 9470 READ(11,*,END=9500) X1(I),Y1(I),Z1(I),QSUM(I)
      Z1(I)=ABS(Z1(I))
 9480 WRITE(10,*) I,X1(I),Y1(I),Z1(I),QSUM(I)
      GOTO 9460
 9500 NS=I
 9504 WRITE(10,*) 'There are ',NS,' sinks'
      WRITE(10,*)
```

```
      CLOSE (11)
 9510 WRITE(10,*)
      WRITE(10,*)
      WRITE(10,*)
C 9520 WRITE(*,*) 'STARTING TIME MUST BE LESS THAN ',CTIME,
C     $' AND GREATER THAN OR EQUAL TO ',CTIME-TIMESTEP
 9530 WRITE(*,*) '   ENTER STARTING TIME OF SIMULATION '
      READ(*,*) T2
C 9540 IF ((T2.LE.CTIME).OR.(T2.LT.CTIME-TIMESTEP)) THEN
C         WRITE(*,*) 'STARTING TIME MUST BE LESS THAN ',CTIME,
C     $   ' AND GREATER THAN ',CTIME-TIMESTEP
C         GOTO 21030
C      ENDIF
 9550 WRITE(*,*)  '         STARTING TIME OF SIMULATION (DAYS) = ',T2
 9560 WRITE(10,*) '         STARTING TIME OF SIMULATIN (DAYS) =  ',T2
 9570 WRITE(10,*)
      WRITE(10,*)
C 9580 WRITE(*,*) 'DO YOU WANT TO REDO THIS SCREEN DATA ? (Y-Yes, N-No) '
C      READ(*,'(A)') ANSS
C 9590 IF ((ANSS.NE.'N').AND.(ANSS.NE.'n').AND.(ANSS.NE.'Y').AND.
C     $(ANSS.NE.'y')) THEN
C        WRITE(*,*) '(Y-Yes, N-No)'
C        GOTO 21080
C      ENDIF
C 9600 IF ((ANSS.EQ.'Y').OR.(ANSS.EQ.'y')) THEN GOTO 21030
      CLOSE (11)
C ENTER ADDITIONAL NON-POINT LOADING PROBLEM ARRAYS
C      WRITE(*,*) 'ENTER FILENAME CONTAINING RECHARGE ARRAY: '
C      READ(*,*) RECHIN
C      OPEN (55,FILE=RECHIN,STATUS='UNKNOWN')
C      WRITE(*,*) 'ENTER FILENAME CONTAINING LOADING BOOLEAN ARRAY: '
C      READ(*,*) BOOLIN
C      OPEN (56,FILE=BOOLIN,STATUS='UNKNOWN')
C      WRITE(*,*) 'ENTER FILENAME CONTAINING SURFACE ELEVATIONS: '
C      READ(*,*) ALTIN
C      OPEN(57,FILE=ALTIN,STATUS='UNKNOWN')
C      DO 9605 J=1,NR
C            READ(55,*) (RECH(I,J), I = 1,NC)
C            READ(56,*) (IBOOL(I,J), I = 1,NC)
C            READ(57,*) (ALT(I,J), I = 1,NC)
C 9605 CONTINUE
C      CLOSE(55)
C      CLOSE(56)
C      CLOSE(57)
 9610 RETURN
      END

C CALLED FROM SUBROUTINE MOVE 10400
C  ****************************************************************
C    SUBROUTINE TO MOVE PARTICLE OUT OF ZERO VELOCITY GRID
C  ****************************************************************
      SUBROUTINE ZEROVG(XP,YP,III,JJJ,K,NC,NR)
```

```
      COMMON/FLOW/VI,VJ,VK
      DIMENSION VI(161,151,4),VJ(161,151,4),VK(161,151,4)
10440 XP=III+1.01
10470 YP=JJJ+1.01
10480 IF (III+2.GE.NC) THEN
         XP=III-.01
         GOTO 10500
      ENDIF
10490 IF (VJ(III+2,JJJ,K).EQ.0) XP=III-.01
10500 IF (JJJ+2.GE.NR) THEN
         YP=JJJ-.01
         GOTO 10520
      ENDIF
10510 IF (VI(III,JJJ+2,K).EQ.0) YP=JJJ-.01
10520 RETURN
      END

C  ************************************************************
C   ROUTINE TO INPUT VELOCITY FILE END TIMES
C
C    NENDTIME - NUMBER OF VELOCITY FILES TO BE USED DURING SIMULATION
C    ENDTIME() - ARRAY OF ENDING TIMES FOR EACH VELOCITY FILE
C  ************************************************************
      SUBROUTINE TMNGR
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      DIMENSION ENDTIME(20)
12870 WRITE(*,*)
12880 WRITE(*,*) '  IF MULTIPLE VELOCITY FILES ARE TO BE USED
     $(TRANSIENT SIMULATION)'
12890 WRITE(*,*) '  THE USER MUST ENTER THE NUMBER OF VELOCITY FILES
     $THAT WILL BE USED'
12900 WRITE(*,*) '  AND THE ENDING TIME FOR EACH VELOCITY FILE'
12910 WRITE(*,*)
12920 WRITE(*,*) '  ENTER THE NUMBER OF VELOCITY FILES (TIME PERIODS) '
      READ(*,*)     NENDTIME
12950 IF (NENDTIME.LE.0) THEN
           WRITE(*,*) '  STEADY STATE ANALYSIS'
           RETURN
      ENDIF
12960 IF (NENDTIME.GT.20) THEN
          WRITE(*,*) '  MAXIMUM NUMBER OF TIME PERIODS IS 20 '
          GOTO 12920
      ENDIF
12970 DO 13010 I=1,NENDTIME
12980   WRITE(*,*) ' ENTER THE ENDING TIME (DAYS) FOR VELOCITY FILE #',I
        READ(*,*)   ENDTIME(I)
        IF (I.EQ.1) GOTO 13010
13002   IF (ENDTIME(I).LT.ENDTIME(I-1)) THEN
            WRITE(*,*) ' ENDING TIME MUST BE GREATER THAN PREVIOUS TIME'
            GOTO 12980
        ENDIF
13010 CONTINUE
```

```
13020 RETURN
      END
      SUBROUTINE MASSIN(M)
C  METEORIC H AND O ISOTOPE MASS INPUT SUBROUTINE
C  DEL18 BASED ON THE EMPIRICAL FORMULAS OF MODERN METEORIC
C  WATER BY YURTSEVER AND GAT, 1981 (AN UPDATE OF DANSGAARD, 1964)
C  WITH TEMPERATURE AND PRECIPITATION CONSTRAINED BY
C  POLLEN CURVES OF WEBB AND BRYSON, 1972.
C  DELD IS BASED ON THE CRAIG, 1961 METEORIC WATER LINE.
C  PROGRAM BY JOHN HOAGLUND
C  WITH THANKS TO K.C. LOHMANN, UNIV. OF MICHIGAN
      REAL LLX,LLY,LLZ,LAMDBA,PM
      DOUBLE PRECISION LAT,ALT
      REAL DELTA,T2,ENDTIME,DELX,DELY
      INTEGER SWITCH(10001),NC,NR,NL,NPER,NENDTIME,M
      DOUBLE PRECISION R,V,D,H,O18,O16
      DOUBLE PRECISION DEL18BS,DELDBS,DEL18,DELD,AMUKG
      DOUBLE PRECISION BS18,BS16,BSD,BSH,DFF18,DFF16,DFFD,DFFH
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      DIMENSION RETARD(7),ENDTIME(20),X(10001),Y(10001),Z(10001)
C SET CONVERSION FACTOR
C 1 AMU =  1.6605E-24 g = 1.6605E-27 KG
      AMUKG=1.6605E-27
C MODEL DOMAIN (INDEPENDENT) VARIABLES HARDWIRED AS
C POINT SPECIFIC
C RECHARGE OF 8 INCHES/YEAR IN FT/DAY
      R=.001826484
C LATITUDE OF 44 DEGREES
      LAT=44.0
C ALTITUDE OF 1200 FEET IN METERS
      ALT=1200*(0.3048)
C DELTA CANNOT BE ZERO, SET TO 100 YEARS
      IF (DELTA.EQ.0.) DELTA = 36500
C BEGIN
C CALCULATE VOLUME AS V = R DT DX DY IN METERS CUBED
C CONVERT TO SI
C R*DELX*DELY IN FT CUBED / DAY
C CONVERT TO METERS CUBED / DAY
      V = R*DELTA*DELX*DELY*(0.3048)*(0.3048)*(0.3048)
C SET ISOTOPIC COMPOSITION OF BASE WATER
C LIGHTEST IN SYSTEM ON METEORIC WATER LINE
      DEL18BS=-19.00
      DELDBS=8*DEL18BS + 10
C CALCULATE BASE MASSES OF O18,O16,D,H IN KG
      CALL AMUMASS(V,DEL18BS,DELDBS,O18,O16,D,H)
      BS18=O18*AMUKG
      BS16=O16*AMUKG
      BSD=D*AMUKG
      BSH=H*AMUKG
C GET DEL18 FROM YURTSEVER AND GAT THERMAL, PRECIPITATION, LATITUDE,
```

```
C AND ALTITUDE RELATION

C FIRST GET TEMPERATURE AND PRECIPITATION FROM WEBB AND BRYSON, 1972
C T2 IS ELAPSED TIME IN DAYS
      WRITE(*,*) 'CALLING POLLEN'
      CALL POLLEN(T2,TEMP,PRECIP)
      WRITE(*,*) 'BACK FROM POLLEN'
C THEN APPLY FORMULA FROM YURTSEVER AND GAT FOR DEL18
      A=0.597
      B=-0.0035
      C=0.106
      D=0.0012
      DEL18= -18.72 + A*TEMP + B*PRECIP + C*LAT + D*ALT
      WRITE(*,*) 'DEL 18 = ',DEL18
C DETERMINE DELD FROM DEL18 USING
C THE CRAIG (1961) METEORIC WATER LINE
      DELD=8*DEL18 + 10
C CALCULATE RECHARGE MASSES OF O18,O16,D,H
      CALL AMUMASS(V,DEL18,DELD,O18,O16,D,H)
C CALCULATE DIFFERENCE BETWEEN RECHARGE MASSES AND
C BASE MASSES OF O18,O16,D,H.  MODEL THIS MASS
C DIFFERENCE FOR MAXIMUM RESOLUTION ON DEL
      DFF18=O18*AMUKG-BS18
      DFF16=O16*AMUKG-BS16
      DFFD=D*AMUKG-BSD
      DFFH=H*AMUKG-BSH
C      WRITE(*,*) 'USING A DEL18 BASE OF ',DEL18BS,' THE ADDITIONAL '
C      WRITE(*,*) 'MASSES OF ISOTOPES (IN KG PER 1000 KG) ARE: '
C      WRITE(*,*) 'O18: ',DFF18
C      WRITE(*,*) 'O16: ',DFF16
C      WRITE(*,*) 'D: ',DFFD
C      WRITE(*,*) 'H: ',DFFH
C PM IS PARTICLE MASS IN KG TO
C CALCULATE M, THE NUMBER OF PARTICLES, AS M=DFF18/PM
      M=DFF18/PM
      WRITE(*,*) M
      RETURN
      END

C SUBROUTINE TO CALCULATE AMU MASSES OF GIVEN ISOTOPIC COMPOSITION

      SUBROUTINE AMUMASS(V,DEL18,DELD,O18,O16,D,H)
      INTEGER COUNT
      DOUBLE PRECISION V,DEL18,DELD,O18,O16,D,H
      DOUBLE PRECISION RSTD18,RSTDD,R18,RD,MT
      DOUBLE PRECISION RHO,RHOLD,CONVRG,AMUKG
C SET CONVERGENCE CRITERIA
      CONVRG = 1.00
C SET CONSTANTS
      RSTD18 = 0.0020052
      RSTDD = 0.0001558
C SET CONVERSION FACTOR
```

```
C 1 AMU =  1.6605E-24 g = 1.6605E-27 KG
      AMUKG = 1.6605E-27
C CALCULATE R18 AND RD
      R18 = RSTD18*(DEL18/1000 + 1)
      RD = RSTDD*(DELD/1000 + 1)
C INITIALIZE DENSITY OF WATER TO 1000 KG/M
      RHO = 1000
C INITIALIZE COUNTER
      COUNT = 0
C BEGIN MAIN LOOP
 1999 COUNT = COUNT + 1
C CALCULATE TOTAL MASS
      MT = V*RHO
C CONVERT TOTAL MASS TO AMU (1 AMU =  1.6605E-24 g = 1.6605E-27 KG)
      MT = MT/AMUKG
C CALCULATE MASS OF H
      H = MT/(RD + 8*R18 + 9)
C CALCULATE MASS OF D
      D = RD*H
C CALCULATE MASS OF O16
      O16 = MT/( (RD+1)/8  +  (R18+1) )
C CALCULATE MASS OF O18
      O18 = R18*O16
C RE-ADD MASS
      MT = H + D + O18 +  O16
C CONVERT MASS TO KILOGRAMS (1 AMU =  1.6605E-24 g = 1.6605E-27 KG)
      MT = MT*AMUKG
      RHOLD = RHO
C RECALCULATE DENSITY
      RHO = MT/V
C TEST CLOSURE, IF NOT GO TO LOOP START
      IF (ABS(RHO-RHOLD).LT.CONVRG) GOTO 2001
      GOTO 1999
 2001 CONTINUE
C      WRITE(*,*) 'AFTER ',COUNT,' ITERATIONS: '
C      WRITE(*,*) DEL18
C      WRITE(*,*) ' O16  (KG) ',O16
C      WRITE(*,*) ' O18  (KG) ',O18
C      WRITE(*,*) ' D    (KG) ',D
C      WRITE(*,*) ' H    (KG) ',H
      RETURN
      END
      SUBROUTINE POLLEN(T2,TEMP,PRECIP)
C VALUES REFLECT MEAN ANNUAL TEMPERATURES
C FOLLOWING MEAN JULY TEMPERATURES FROM WEBB AND BRYSON
C ASSUME STARTING TIME OF 15000 YEARS BP
      IF ((T2.GE.0).AND.(T2.LT.(7000*365))) THEN
         TEMP=((45-18)/(7000*365))*T2 + 18
         PRECIP=20
      ENDIF
      IF (T2.GE.(7000*365)) THEN
C         TEMP=45
```

```
          TEMP=20
         PRECIP=20
      ENDIF
C CONVERT TEMPERATURE TO DEGREES CENTIGRADE
      TEMP=(TEMP-32)*5/9
C CONVERT SUMMER PRECIP IN INCHES TO MONTHLY PRECIP IN MILLIMETERS
      PRECIP=(PRECIP/36)*(0.3048)*1000
      RETURN
      END
```

APPENDIX II

APPENDIX II: CHANGES TO ISORAN3D.FOR TO MAKE TRAVELT.FOR

A. Changes to Program MAIN (changed and/or new lines within border):

```
C NEW RANDOM WALK 3-D MODEL
C   REVISED TO SMALLER GRID ARRAY 1/9/90
C   PLT ROUTINE REVISED 2/15/89
C
C     VARIABLES
C     ---------
C ASPECT - ASPECT RATIO OF SCREEN
C BOT(I,J,K) - ELEVATION OF BOTTOM OF NODE I,J,K (FT)
C BOT(I,J,K) - ELEVATION OF BOTTOM OF NODE I,J,K (FT)
C BOTCAPTURE - BOTTOM ELEVATION AT WHICH PARTICLE MAY BE
C              CAPTURED BY SINK
C CAPTUREMAX - RIVER MAXIUM CAPTURE DISTANCE
C CDX - COLUMN SPACING OF MAP ARRAY
C CDY - ROW SPACING OF MAP ARRAY
C COLUMNS - NUMBER OF COLUMNS IN PLOT ARRAY
C CONC(L) - SINK CONCENTRATION FOR SINK L(MG/L)
C D - TIME STEP FOR MOVE, =DELTA IF SLUG SOURCE PARTICLE
C     = RANDOM FRACTION OF DELTA FOR CONTINUOUS SOURCE PARTICLE
C     (DAYS)
C DD - HORIZONTAL DISTANCE TO MOVE (FT)
C DDX - DISTANCE OF MOVE (FT)
C DELTA - TIME STEP OF MOVE SUBROUTINE (DAYS)
C DELX - COLUMN SPACING (FT)
C DELY - ROW SPACING (FT)
C DL  - LONGITUDINAL DISPERSIVITY (FT)
C DMAX - MAXIMUM HORIZONTAL MOVE DISTANCE BEFORE RECOMPUTING
C        VELOCITY (DAYS)
C DT  - TRANSVERSE DISPERSIVITY (FT)
C DV  - VERTICAL DISPERSIVITY (FT)
C E   - POROSITY
C ENDTIME(M) - ENDTIMES FOR EACH MODEL VELOCITY FILE (DAYS)
C FS - MAP FLAG, IF='A' THEN MAP HAS BEEN RUN
C GS - FLAG INDICATING COEFFICIENTS HAVE BEEN ENTERED IF
C     ='B'
C I1 - X COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW
C       (FT)
C I2 - X COORDINATE OF UPPER LEFT CORNER OF SCREEN WINDOW
```

```
C       (FT)
C ICOLXS - COLUMN DISPLAYED IN COLUMN CROSS-SECTION VIEW
C IPEN - COLOR OF SPECIAL FEATURES ON SCRREN
C J1 - Y COORDINATE OF LOWER LEFT CORNER OF SCREEN WINDOW
C       (FT)
C J2 - Y COORDINATE OF UPPER LEFT CORNER OF SCREEN WINDOW
C       (FT)
C JROWXS - ROW DISPLAYED IN ROW CROSS-SECTION VIEW
C LAMDBA - HALF LIFE FOR FIRST ORDER DECAY (YEARS)
C LL - LAYER OF PARTICLE
C LLX  - X COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
C LLY  - Y COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
C LLZ  - Z COORDINATE OF LOWER LEFT CORNER OF MODEL GRID (FT)
C LN - LINES OF MAP
C LOWERLX - X COORDINATE OF LOWER LEFT CORNER OF PLOT AREA (FT)
C LOWERLY - Y COORDINATE OF LOWER LEFT CORNER OF PLOT AREA (FT)
C MAPTYPES - FLAG INDICATING WHAT VIEW OF THE PROBLEM WAS LAST
C            DISPLAYED ON THE SCREEN , A - TOP VIEW, B - ROW
C            CROSS-SECTION VIEW, C - COLUMN CROSS-SECTION VIEW
C NC  - NUMBER OF COLUMNS
C NR  - NUMBER OF ROWS
C NENDTIME - NUMBER OF TIME PERIODS
C NEWX - X COORDINATE OF PARTICLE AFTER MOVE (FT)
C NEWY - Y COORDINATE OF PARTICLE AFTER MOVE (FT)
C NL  - NUMBER OF LAYERS
C NMAP(I,J) - MAP AND PLOT STORAGE FOR # OF PARTICLES OR
C             CONCENTRATIONS
C NP  - NUMBER OF PARTICLES
C NR  - NUMBER OF ROWS
C NS - NUMBER OF SINKS
C NSPF - NUMBER OF SPECIAL FEATURE FILES (WITH KEYS AND FILE
C        NAMES)
C O   - SLIDE COUNTER
C PM  - PARTICLE MASS (LBS)
C PPP - CONCENTRATION FACTOR, =0.001 FOR PPB, =1 FOR PPM
C       =1000 FOR PPT
C QSUM(L) - DISCHARGE OF SINK L (GPD)
C RS - NAME OF VELOCITY FILE (WITH EXTENSION .RND)
C R3 - DISTANCE FROM SINK TO PARTICLE (FT)
C R4 - VERTICAL DISTANCE FROM PARTICLE TO BOTTOM OF RIVER
C RETARD(K) - RETARDATION FACTORS FOR EACH LAYER AND CONFINING LAYER
C RL - LONGITUDINAL DISPERSION (FT)
C RN - NORMALLY DISTRIBUTED RANDOM NUMBER BETWEEN -6 AND +6
C      PRODUCED IN SUBROUTINE RAND
C RND() - RANDOM NUMBER BETWEEN ZERO AND 1
C       PRODUCED IN FUNCTION RND()
C ROWS - NUMBER OF ROWS IN PLOT ARRAY
C RT - TRANSVERSE DISPERSION (FT)
C RZ - VERTICAL DISPERSION (FT)
C SCALEDEF - DEFAULT X DISTANCE FOR SCREEN WINDOW
C SCALEX - X DIMENSION OF SCREEN WINDOW (I2-I1) (FT)
C SCREENX - X COORDINATE TO PLOT ON SCREEN (FT)
```

```
C SCREENY - Y COORDINATE TO PLOT ON SCREEN (FT)
C SIZE - GRID SPACING FOR PLOT AREA (FT)
C SPFFILES(I) - FILE NAMES OF SPECIAL FEATURE FILES
C SPFKEYS(I,J) - KEYS USED TO CALL SPECIAL FEATURE FILE PLOTS, I
C              IS SPECIAL FEATURE COUNTER, J IS 1 OR 2 CONTAINING
C              BOTH UPPER AND LOWER CASE LETTERS
C SPFSAVES - NAME OF FILE WITH SPECIAL FEATURE DATA
C SVIEWX - X DIMENSION OF ZOOM BOX
C SWITCH(I) - ARRAY OF FLAGS INDICATING WHETHER IT IS A
C             SLUG OR CONTINUOUS PARTICLE, 1= CONTINUOUS,
C             0=SLUG
C T2  - TIME (DAYS)
C TEMPSCR - FLAG, IF=1 IT INDICATES A SCREEN IMAGE HAS BEEN
C           SAVED AS 'TEMP.SCR'
C THICK(I,J,K) - THICKNESS OF AQUIFER OF COLUMN I, ROW J, AND LAYER K
C TOP(I,J,K) - ELEVATION OF TOP OF NODE I,J,K, IF THE TOP WATER TABLE
C                LAYER THEN =0 UNLESS A RIVER NODE, THEN = BOTTOM OF
C                RIVER ELEVATION (FT)
C TOPCAPTURE - TOP ELEVATION AT WHICH PARTICLE MAY BE
C             BE CAPTURED BY SINK
C VI(I,J,K) - VELOCITY IN X DIRECTION FROM I,J,K TO I+1,J,K (FT/DAY)
C VJ(I,J,K) - VELCOITY IN Y DIRECTION FROM I,J,K TO I,J+1,K (FT/DAY)
C VK(I,J,K) - VELOCITY IN Z DIRECTION FROM I,J,K TO I,J,K+1 (FT/DAY)
C VX - INTERPOLATED X VELOCITY (FT/DAY)
C VY - INTERPOLATED Y VELOCITY (FT/DAY)
C VZ - INTERPOLATED Z VELCOITY (FT/DAY)
C WD - COLUMNS OF MAP
C X(I) - ARRAY OF X COORDINATES FOR PARTICLES (FT)
C X1(L) - X COORDINATE OF SINK L (FT)
C XP - X COORDINATE OF PARTICLE POSITION IN GRID UNITS
C XXS - FLAG INDICATING A COLOR MONITOR IS PRESENT, ALWAYS='Y'
C       IN THIS VERSION
C Y(I) - ARRAY OF Y COORDINATES FOR PARTICLES (FT)
C Y1(L) - Y COORDINATE OF SINK L (FT)
C YASPECT - ASPECT RATIO OF SCREEN FOR TOP VIEW
C YP - Y COORDINATE OF PARTICLE POSITION IN GRID UNITS
C Z(I) - ARRAY OF Z COORDINATS FOR PARTICLES (FT)
C Z1(L) - LAYER OF SINK L
C Z2S - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON SCREEN
C Z3S - STRING OF KEYS PRESSED TO PUT SPECIAL FEATURES ON SCREEN
C ZASPECT - ASPECT RATIO OF SCREEN FOR CROSS-SECTION VIEW
C ZMAX - MAXIUM VERTICAL MOVE DISTANCE BEFORE RECOMPUTING
C        VELOCITY (DAYS)
C ZZZ - VARIABLE USED TO IMPLEMENT FIRST ORDER DECAY
C
C RECOVER FROM ALL ERRORS WITHOUT BOMBING OUT.
C 40 ON ERROR GOTO 5430
C NAME: THREE DIMENSIONAL RANDOM WALK (MICROCOMPUTER VERSION)
C
C PURPOSE: TO SIMULATE 3-DIMENSIONAL, STEADY OR NONSTEADY
C          MASS TRANSPORT PROBLEMS IN HETEROGENEOUS AQUIFERS
C          WITH THE RANDOM WALK PARTICLE THEORY.
```

```
C
C WRITTEN BY:  DONALD KOCH
C              ENGINEERING TECHNOLOGIES ASSOCIATES, INC.
C              3458 ELLICOTT CENTER DRIVE
C              ELLICOTT CITY, MD, 21043
C             (301) 461-9920
C     BASED ON CODE WRITTEN BY
C              THOMAS A. PRICKETT & ASSOCIATES
C             URBANA, ILLINOIS 61801
C              (217)384-0615
C BLOCKS BY JRH
C FLAGS
      CHARACTER GS,FS,ANSW
      CHARACTER*5 AS
      CHARACTER*40 RS
      REAL LLX,LLY,LLZ,LAMDBA,I1,J1,I2,J2,MAPCONC(161,151),PM
      REAL DELTA,T2,ENDTIME,DELX,DELY,DMAX,ZMAX
      INTEGER SWITCH(10001),NC,NR,NL,NPER,NENDTIME,ISUNK
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/GEO/THICK,BOT,TOP
      COMMON/FLOW/VI,VJ,VK
      COMMON/SINK/NS,X1,Y1,Z1,QSUM
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/FLAGS/GS,FS,RS
      COMMON/MAPS/NMAP,CONC,MAPCONC,CDX,CDY,I1,J1,I2,J2
      COMMON/DIG/DIGX,DIGY,DIGZ
      DIMENSION DIGX(202),DIGY(202),DIGZ(202)
      DIMENSION THICK(161,151,4),BOT(161,151,4),TOP(161,151,4)
      DIMENSION VI(161,151,4),VJ(161,151,4),VK(161,151,4)
      DIMENSION X1(5000),Y1(5000),Z1(5000),QSUM(5000)
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION NMAP(161,151),CONC(5000)
      DIMENSION ENDTIME(20)
C END BLOCKS BY JRH
C ALLOW ARRAYS FOR 10000 PARTICLES AND 99 SINKS AND 35 BY 35 VELOCITY ARRAY
C
C INITIALLY SEED RANDOM NUMBER GENERATOR?
      WRITE(*,*)' RANDOM # GENERATION IS SEEDED WITH 1 BY DEFAULT.'
      WRITE(*,*)' CHOOSING A NEW SEED WILL RESULT IN A NEW SEQUENCE'
      WRITE(*,*)' OF RANDOM NUMBERS.  DO YOU'
      WRITE(*,*)' WANT TO RESEED THE RANDOM # GENERATOR (Y OR N)?'
      READ(*,'(A)') ANSW
      IF ((ANSW.EQ.'Y').OR.(ANSW.EQ.'y')) CALL SEED(RND$TIMESEED)
C INITIALIZE VELOCITY FILE (TIME PERIODS) COUNTER
      NPER=0
C INITIALIZE NUMBER OF VELOCITY FILES
      NENDTIME=0
C SET COEFFICIENT FLAG
      GS='F'
C INITIALIZE VARIABLES
```

```
      T2=0
      TIME=0
C NUMBER OF SINKS=0
      NS = 0
C NUMBER OF PARTICLES=0
      NP = 0
 760  CONTINUE
C HERE IS THE MAIN MENU.  REFER TO ASCII CODES FOR FRAME CHARACTERS.
C
      WRITE(*,*) ' ╔══════════════════ USER MENU ═══════════════════╗'
      WRITE(*,*) ' ║    FORMAT:   COMMAND - DESCRIPTION              ║'
      WRITE(*,*) ' ╠════════════════════════╦════════════════════════╣'
      WRITE(*,*) ' ║  SET   - MASS TRANSPORT ║   INPUT - READ IN FLOW ║'
      WRITE(*,*) ' ║          COEFFICIENTS   ║           DATA FILE    ║'
      WRITE(*,*) ' ║                         ║   P - ADD PARTICLES    ║'
      WRITE(*,*) ' ║  RE -   READ DATA       ║       (POLLUTANTS)     ║'
      WRITE(*,*) ' ║  WR -   WRITE DATA      ║                        ║'
      WRITE(*,*) ' ║                         ║ MOVE - MOVE PARTICLES  ║'
      WRITE(*,*) ' ║                         ║        FOR TIME STEP   ║'
      WRITE(*,*) ' ║  XS - CLEAR SINKS       ║                        ║'
      WRITE(*,*) ' ║  XP - CLEAR PARTICLES   ║ MAP - MAP PARTICLES OR ║'
      WRITE(*,*) ' ║  XT - SET TIME TO 0     ║       CONCENTRATIONS   ║'
      WRITE(*,*) ' ║                         ║                        ║'
      WRITE(*,*) ' ║  TMNGR - SET VELOCITY   ║                        ║'
      WRITE(*,*) ' ║          FILE ENDTIMES  ║                        ║'
      WRITE(*,*) ' ║          (TRANSIENT)    ║ Q - QUIT               ║'
      WRITE(*,*) ' ╚═════════════════════════╩════════════════════════╝'
      WRITE(*,*) ' COMMAND '
      READ(*,'(A)') AS
C JUNCTION POINT FOR MENU COMMAND SELECTION.
      IF ((AS.EQ.'P    ').OR.(AS.EQ.'p    ')) CALL GENP
      IF ((AS.EQ.'XP   ').OR.(AS.EQ.'xp   ')) CALL XP
      IF ((AS.EQ.'RE   ').OR.(AS.EQ.'re   ')) CALL REINCD
      IF ((AS.EQ.'XS   ').OR.(AS.EQ.'xs   ')) CALL XS(NS)
      IF ((AS.EQ.'WR   ').OR.(AS.EQ.'wr   ')) CALL WRFNCD
      IF ((AS.EQ.'XT   ').OR.(AS.EQ.'xt   ')) CALL XT(T2)
      IF ((AS.EQ.'TMNGR').OR.(AS.EQ.'tmngr')) CALL TMNGR
      IF ((AS.EQ.'MOVE ').OR.(AS.EQ.'move ').OR.(AS.EQ.'Move ')) THEN
          CALL MOVE (DMAX,ZMAX,ISUNK)
      ENDIF
      IF ((AS.EQ.'Q    ').OR.(AS.EQ.'q    ')) GOTO 1110
      IF ((AS.EQ.'MAP  ').OR.(AS.EQ.'map  ').OR.(AS.EQ.'Map  ')) THEN
         CALL MAP3D
      ENDIF
      IF ((AS.EQ.'SET  ').OR.(AS.EQ.'set  ')) CALL SET
      IF ((AS.EQ.'EDIT ').OR.(AS.EQ.'edit ')) CALL SET
      IF ((AS.EQ.'INPUT').OR.(AS.EQ.'input')) CALL REFLOW
C BOUNCING GOTO
 1100 GOTO 760
 1110 CONTINUE
      STOP
      END
```

B. Changes to subroutine GENP (changed and/or new lines within border):

```
C    ++++++++++++++++++++++++++++++++++++
C    +  BEGIN PARTICLE INPUT SUBROUTINE  +
C    ++++++++++++++++++++++++++++++++++++
      SUBROUTINE GENP
      INTEGER SWITCH(10001),L,C,S,RC,NPRC
      CHARACTER AS
      REAL DMAX,ZMAX
      INTEGER ISUNK
      INTEGER IZ,JZ,IR,JR
      DOUBLE PRECISION X5,Y5,Z5,X6,Y6,Z6
      REAL XPASS,YPASS,ZPASS,LLX,LLY,LLZ,LAMDBA,PM,DELDIG
      COMMON/DIG/DIGX,DIGY,DIGZ
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/GEO/THICK,BOT,TOP
      COMMON/NPNT/RECH,IBOOL,ALT
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION DIGX(202),DIGY(202),DIGZ(202)
      DIMENSION THICK(161,151,4),BOT(161,151,4),TOP(161,151,4)
      DIMENSION TRVL(161,151)
      DIMENSION ENDTIME(20)
 1270 WRITE(*,*) '///PARTICLES\\\'
 1280 WRITE(*,*)
 1290 WRITE(10,*) '///PARTICLES\\\'
 1300 WRITE(10,*)
C  INITIALIZE LINE COUNTER
 1310 L=0
C  INITIALIZE CYLINDER COUNTER
 1320 C=0
C  INITIALIZE SPHERE COUNTER
      S=0
C  INITIALIZE RECTANGLE COUNTER
      RC=0
C  POINT RECTANGLE COUNTER INITIALIZED AT GENP CALL
      NPRC=0
      IF ((DELX.LE.0).OR.(DELY.LE.0)) THEN
         WRITE(*,*) '  YOU MUST ENTER A VELOCITY FILE BEFORE
     $                INITIATING PARTICLES'
         RETURN
      ENDIF
 1330 WRITE(*,*)
 1340 WRITE(*,*) '  ADD PARTICLES TO SYSTEM? (Y-YES, <RETURN>-NO)'
      READ(*,'(A)') AS
C QUIT OPTION
 1360 IF ((AS.NE.'Y').AND.(AS.NE.'y')) GOTO 2480
```

```
C GET MOVE PARAMETERS
      OPEN (7, FILE='TRAVEL.MTX',STATUS='UNKNOWN')
      OPEN (8, FILE='TRAVEL.1',STATUS='UNKNOWN')
      OPEN (9, FILE='TRAVEL.2',STATUS='UNKNOWN')
      OPEN (13,FILE='PSIGT1K.XYZ',STATUS='UNKNOWN')
      OPEN (14,FILE='PSIGT10K.XYZ',STATUS='UNKNOWN')
      OPEN (15,FILE='PSIGT20K.XYZ',STATUS='UNKNOWN')
      WRITE(*,*) ' ENTER THE TIME STEP TO BE USED FOR ALL MOVES: '
      READ(*,*) DELTA
C WRITE DIGITIZER COORDINATES EVERY 1 MULTIPLES OF DELTA
      DELDIG=1*DELTA
      WRITE(*,*) ' ENTER THE MAXIMUM HORIZONTAL MOVE (DMAX): '
      READ(*,*) DMAX
      WRITE(*,*) ' ENTER THE MAXIMUM VERTICAL MOVE (ZMAX): '
      READ(*,*) ZMAX
C ADD PARTICLES AT NODAL POINTS
      DO 2500 IR = 1,NC
        DO 2495 JR = 1,NR
C ********************************************************************
C INITIATE A POINT PARTICLE BY POINT LOADING
C ********************************************************************
C DETERMINE COORDINATES OF LOWER LEFT CORNER (X,Y) OF CELL
      X5 = (IR-1)*DELX + LLX + .001
      Y5 = (JR-1)*DELY + LLY + .001
C          WRITE(10,*) ' RECTANGULAR PRISM COORDINATES:'
C          WRITE(10,*) ' LOWER LEFT CORNER (X,Y) = ',X5,',',Y5,' FT'
C DETERMINE COORDINATES OF UPPER RIGHT CORNER (X,Y) OF CELL'
      X6=X5 + DELX - .002
      Y6=Y5 + DELY - .002
C          WRITE(10,*) ' UPPER RIGHT CORNER (X,Y) = ',X6,',',Y6,' FT'
C          WRITE(*,*) ' ENTER Z COORDINATES (LOWER,UPPER) '
C DETERMINE COORDINATES OF TOP (Z) AND BOTTOM (Z) OF CELL
C USING WATER TABLE AND 5 FOOT BELOW WATER TABLE
C          Z5=BOT(IR,JR,NL) + THICK(IR,JR,NL) - 5
C          Z6=BOT(IR,JR,NL) + THICK(IR,JR,NL)
C USING BOTTOM
                                                                   OF
LAYER 1 AND 5 FEET ABOVE
C       Z5=BOT(IR,JR,NL)
C AND THE WATER TABLE
C       Z6=BOT(IR,JR,NL) + THICK(IR,JR,NL)
      Z5=BOT(IR,JR,NL) + THICK(IR,JR,NL) - .001
      Z6=BOT(IR,JR,NL) + .001
C WRITE(10,*) ' Z COORDINATES (LOWER,UPPER) = ',Z5,',',Z6,' FT'
      IF (((X6-X5).LT.0).OR.((Y6-Y5).LT.0)) THEN
         WRITE(*,*) 'CANNOT FORM RECTANGLE--REDO'
```

```
         GOTO 2418
      ENDIF
      IF ((X5.LT.(LLX)).OR.(X5.GT.(LLX+DELX*(NC)))) THEN
         WRITE(*,*) '  LOWER LEFT CORNER X COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 2418
      ENDIF
      IF ((Y5.LT.(LLY)).OR.(Y5.GT.(LLY+DELY*(NR)))) THEN
         WRITE(*,*) '  LOWER LEFT CORNER Y COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 2418
      ENDIF
      IF ((Y6.LT.(LLY)).OR.(Y6.GT.(LLY+DELY*(NR)))) THEN
         WRITE(*,*) '  UPPER RIGHT CORNER Y COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 2418
      ENDIF
      IF ((X6.LT.(LLX)).OR.(X6.GT.(LLX+DELX*(NC)))) THEN
         WRITE(*,*) '  UPPER RIGHT CORNER X COORDINATE IS
     $   OUTSIDE THE MODEL GRID'
         GOTO 2418
      ENDIF
      IZ=INT((X5-LLX)/DELX+1.0)
      JZ=INT((Y5-LLY)/DELY+1.0)
      IF ((Z5.LT.BOT(IZ,JZ,1)).OR.
     $(Z5.GT.BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL))) THEN
          WRITE(*,*) '  1ST Z COORDINATE OUTSIDE MODEL'
          GOTO 2418
      ENDIF
      IF ((Z6.LT.BOT(IZ,JZ,1)).OR.
     $(Z6.GT.BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL))) THEN
         WRITE(*,*) '  2ND Z COORDINATE OUTSIDE MODEL'
         GOTO 2418
      ENDIF

C DETERMINE ADDING COORDINATES
      M=1
      DX = X6  - X5
      DY = Y6 - Y5
      DZ = Z6 - Z5
      DO 2416 I = 1,M
         XPASS = X5 + DX/2
         YPASS = Y5 + DY/2
         ZPASS = Z5
C ADD A PARTICLE
         CALL ADD3D(A3,XPASS,YPASS,ZPASS)
 2416 CONTINUE
C CALL MOVE UNTIL PARTICLE IS OUT OF SYSTEM, RECORD TRAVEL TIME
```

```
      T2=0.0
      ID = 1
      ISUNK=0
      WRITE(10,*) 'STARTING 1 PARTICLE AT IR= ',IR,' AND JR= ',JR
      WRITE (*,*) 'STARTING 1 PARTICLE AT IR= ',IR,' AND JR= ',JR
C 2417 write(*,*) 'calling move for ',t2
C STORE DIGITIZER COORDINATE FOR INITIAL LOADING POINT
      CALL DIGIT(ID)
 2417 CALL MOVE (DMAX,ZMAX,ISUNK)
C STORE DIGITIZER COORDINATES FOR SELECTED TIME STEP
      IF (((T2/DELDIG)-INT(T2/DELDIG)).EQ.0) THEN
        ID=ID+1
        CALL DIGIT(ID)
      ENDIF
      IF (ISUNK.GT.0) THEN
C IF REMOVED BY SINK
         IF (ISUNK.EQ.1) THEN
C   WRITE TO TRAVEL.1
           WRITE(8,2419) T2/365,IR,JR,X(1),Y(1),Z(1)
           IF (((T2/365).GT.1000).AND.((T2/365).LE.10000)) THEN
C   WRITE TO PSIGT1K.XYZ
              DO 1500 I = 1,ID
                WRITE(13,*) DIGX(I),DIGY(I),DIGZ(I)
 1500         CONTINUE
              WRITE(13,*)
           ENDIF
           IF ((T2/365).GT.10000) THEN
C   WRITE TO PSIGT10K.XYZ
              DO 1600 I = 1,ID
                WRITE(14,*) DIGX(I),DIGY(I),DIGZ(I)
 1600         CONTINUE
              WRITE(14,*)
           ENDIF
           TRVL(IR,JR)=T2/365
         ENDIF
C IF MOVED OFF GRID
         IF (ISUNK.EQ.2) THEN
C   WRITE TO TRAVEL.2
           WRITE(9,2419) T2/365,IR,JR,X(1),Y(1),Z(1)
           TRVL(IR,JR)=T2/365
           NP=NP-1
         ENDIF
         GOTO 2418
      ENDIF
C IF T2 GREATER THAN 20,000 YEARS
```

```
      IF (T2.GT.7300000) THEN
C   WRITE TO TRAVEL.2
         WRITE(9,2419) T2/365,IR,JR,X(1),Y(1),Z(1)
C   WRITE TO PSIGT20K.XYZ
         DO 1700 I = 1,ID
           WRITE(15,*) DIGX(I),DIGY(I),DIGZ(I)
 1700    CONTINUE
         WRITE(15,*)
         TRVL(IR,JR)=T2/365
         NP=NP-1
         GOTO 2418
      ENDIF
 2419 FORMAT(F10.0,2I5,3F11.2)
      GOTO 2417
 2418 CONTINUE
      WRITE(*,*) ' TOTAL TRAVEL TIME (DAYS) = ',T2
      WRITE(10,*) ' TOTAL TRAVEL TIME = ',T2,'  DAYS'
C      WRITE(10,*) 'POINT LOAD RECTANGLE DONE'
      NPRC=NPRC+1
C LOAD NEXT PARTICLE


 2495   CONTINUE
 2500 CONTINUE
C WRITE OUT TRAVEL ARRAY
      DO 2497 JR = 1,NR
         WRITE(7,'(8E10.5)') (TRVL(IR,JR), IR=1,NC)
 2497 CONTINUE
C END OF ROUTINE
 2480 RETURN
      END
```

C. Added subroutine DIGIT to store digitizer coordinates:

```
C ************************************
C  SUBROUTINE TO STORE DIGITIZER COORDINATES
C ************************************
      SUBROUTINE DIGIT(ID)
      REAL LLX,LLY,LLZ,LAMDBA,PM,SCALE
      REAL DELTA,T2,ENDTIME,DELX,DELY
      INTEGER SWITCH(10001),NC,NR,NL,NPER,NENDTIME
      COMMON/DIG/DIGX,DIGY,DIGZ
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
```

```
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION DIGX(202),DIGY(202),DIGZ(202)
      DIMENSION ENDTIME(20)
C ON 1:500,000 BASEMAP, 41,666.66 FEET PER MAP INCH
      SCALE=41666.666
      DIGX(ID)=X(1)/SCALE
      DIGY(ID)=Y(1)/SCALE
      DIGZ(ID)=Z(1)
      RETURN
      END
```

D. Changes to subroutine MOVE (changed and/or new lines within border):

```
C***************************************************************************
C  MOVE ROUTINE !!!!!!!!!!!!!!!!!
C  MAKE SURE ALL DATA NECESSARY TO MAKE A MOVE HAS BEEN SPECIFIED.
C***************************************************************************
      SUBROUTINE MOVE (DMAX,ZMAX,ISUNK)
      CHARACTER GS,FS
      CHARACTER*40 RS
      REAL LLX,LLY,LLZ,LAMDBA,I1,J1,I2,J2,MAPCONC(161,151),PM
      REAL DMAX,ZMAX,VKBOT
      INTEGER SWITCH(10001),ISUNK
      COMMON/DOMAIN/NC,NR,NL,DELX,DELY,LLX,LLY,LLZ
      COMMON/GEO/THICK,BOT,TOP
      COMMON/FLOW/VI,VJ,VK
      COMMON/SINK/NS,X1,Y1,Z1,QSUM
      COMMON/TIME/NPER,T2,NENDTIME,ENDTIME,DELTA
      COMMON/TRANS/NP,PM,E,DL,DT,DV,LAMDBA,RETARD,X,Y,Z,SWITCH
      COMMON/FLAGS/GS,FS,RS
      COMMON/MAPS/NMAP,CONC,MAPCONC,CDX,CDY,I1,J1,I2,J2
      DIMENSION THICK(161,151,4),BOT(161,151,4),TOP(161,151,4)
      DIMENSION VI(161,151,4),VJ(161,151,4),VK(161,151,4)
      DIMENSION X1(5000),Y1(5000),Z1(5000),QSUM(5000)
      DIMENSION RETARD(7),X(10001),Y(10001),Z(10001)
      DIMENSION NMAP(161,151),CONC(5000)
      DIMENSION ENDTIME(20)
 2510 IF (GS.NE.'B') THEN
          WRITE(*,*) 'PLEASE CHOOSE COEFFICIENTS MENU ITEM NOW!!!'

          RETURN
      ENDIF
 2515 IF ((NENDTIME.GT.0).AND.(NPER.EQ.NENDTIME)
     $.AND.(T2.EQ.ENDTIME(NPER))) THEN
         WRITE(*,*) '  ALL VELOCITY FILES USED, SIMULATION COMPLETE'

         RETURN
```

```
      ENDIF
 2520 IF((NENDTIME.GT.0).AND.(T2.EQ.ENDTIME(NPER))) THEN
          WRITE(*,*) '  READ IN THE NEXT VELOCITY VECTOR FILE NOW!'

          RETURN
      ENDIF
 2530 IF (NP.EQ.0) THEN
         WRITE(*,*) 'DO SOMETHING ELSE-THERE ARE NO PARTICLES TO MOVE!'

         RETURN
      ENDIF
C 2542 IF (ASPECT.LE.0) THEN
C         WRITE(*,*) ' YOU DID NOT ENTER AN ASPECT RATIO FOR THE SCREEN '
C
C         RETURN
C      ENDIF
C 2544 IF (((SCALEDEF.LE.0).AND.(TEMPSCR.EQ.0)).OR.((SCALEX.LE.0).AND.
C     $(TEMPSCR.EQ.1))) THEN
C         WRITE(*,*) '  YOU DID NOT ENTER A DEFAULT SCREEN WIDTH '
C
C         RETURN
C       ENDIF
C 2550 WRITE(*,*) '        PRESENT SIMULATION TIME (DAYS) = ',T2
C 2560 WRITE(10,*) '      PRESENT SIMULATION TIME = ',T2,'  DAYS'
 2570 IF (NENDTIME.LT.0) THEN
         WRITE(*,*) '  TIME REMAINING FOR THIS VELOCITY FILE IS '
     $    ,(ENDTIME(NPER)-T2),' DAYS'
      ENDIF
C 2580 WRITE(*,*) '  ENTER INCREMENTAL SIMULATION TIME (DAYS) [DELTA]'
C      READ(*,*) DELTA
C 2590 IF (DELTA.LE.0) THEN
C         WRITE(*,*) '   DELTA MUST BE GREATER THAN ZERO!!!',
C     $   '---REDO THIS INPUT NOW PLEASE.'
C 2600    GOTO 2580
C      ENDIF
C ASSUME STEADY STATE IF NUMBER OF PERIODS HAVE NOT BEEN ENTERED
C 2602 IF (NENDTIME.LE.0) GOTO 2630
C 2604 IF ((DELTA+T2).GT.ENDTIME(NPER)) THEN
C         WRITE(*,*) '  TIME STEP TOO LARGE, ENTER A SMALLER TIME STEP'
C         DELTA=ENDTIME(NPER)-T2
C         GOTO 2580
C      ENDIF
C 2630 CONTINUE
C 2630 WRITE(10,*) '        INCREMENTAL SIMULATION TIME = ',DELTA,' DAYS'
C      WRITE(*,*)  '        INCREMENTAL SIMULATION TIME = ',DELTA,'DAYS'
C 2640 WRITE(*,*) ' HOW OFTEN DO YOU WANT TO COMPUTE VELOCITY VECTORS'
C      WRITE(*,*) ' ENTER MAXIMUM HORIZONTAL MOVE (FT) [DMAX] '
C      READ(*,*) DMAX
```

```
C       WRITE(*,*) '  ENTER MAXIMUM VERTICAL MOVE (FT) [ZMAX] '
C       READ(*,*) ZMAX
C 2650 WRITE(*,*) ' DMAX = ',DMAX,' FT   ZMAX = ',ZMAX,' FT'
C 2660 WRITE(10,*) 'DMAX = ',DMAX,' FT   ZMAX = ',ZMAX,' FT'
C 2670 WRITE(*,*)
C 2680 WRITE(10,*)
C  LOOP TO ZERO OUT PURGE WELL CONCENTRATION ARRAY.
 2730 DO 2750 K = 1, NS
 2740   CONC(K)=0
 2750 CONTINUE
C  INITIALIZE PARTICLE COUNTER
 2770 K=0
C  INITIALIZE 1ST ORDER DECAY CHECK
      ZZZ=0
C  START LOOP ON PARTICLE MOVE HERE. (SEE LOOP TERMINUS AT 3800 OR 3860)
 2790 K=K+1
C  EXAMINE SWITCH TO DETECT IF CONTINUOUS POLLUTION OR NOT.
 2830 IF (SWITCH(K).EQ.1) THEN
         D=DELTA*RND()
      ELSE
         D=DELTA
      ENDIF
C  SKIP FIRST ORDER DECAY TO SAVE TIME IF CONSERVATIVE POLLUTION
      IF (LAMDBA.GE.1E10) GOTO 2880
C  FIRST ORDER DECAY IMPLEMENTED IN THESE STATEMENTS
      ZZZ=ZZZ+(1-.5**(D/LAMDBA/365))
      IF (ZZZ.GE.1) THEN
         GOTO 2832
      ELSE
         GOTO 2880
      ENDIF
 2832 X(K)=X(NP)
      Y(K)=Y(NP)
      Z(K)=Z(NP)
      SWITCH(K)=SWITCH(NP)
      NP=NP-1
      ZZZ=ZZZ-1.0
C IF LAST PARTICLE DECAYS JUMP TO END OF ROUTINE
      IF (K.GT.NP) GOTO 3660
C  LOOP BACK FOR DECAY ON NEXT PARTICLE
      GOTO 2830
C  ++++++++++++++++++++++++++++++++++++++++++++
C  +  WHEN ALL PARTICLES ARE PROCESSED, THIS  +
C  +  NEXT STATEMENT RETURNS YOU TO THE MENU  +
C  ++++++++++++++++++++++++++++++++++++++++++++
 2880 IF (K.GT.NP) THEN
         T2=T2+DELTA
         GOTO 3660
      ENDIF
 2890 XP=(X(K)-LLX)/DELX+.5
 2900 YP=(Y(K)-LLY)/DELY+.5
```

```
 2910 VX=.00001
 2920 VY=.00001
      VZ=1E-10
 2930 I=INT(XP)
 2940 J=INT(YP)
      IZ=INT(XP+.5)
      JZ=INT(YP+.5)
C  FIND LAYER OF PARTICLE AND INTERPOLATE THE VERTICAL VELOCITY
      DO 2941 KLK=1,(NL-1)
      IF ((Z(K).GE.BOT(IZ,JZ,KLK)).AND.(Z(K).LT.TOP(IZ,JZ,KLK))) THEN
        LL=KLK
        AZ=(Z(K)-BOT(IZ,JZ,KLK))/(TOP(IZ,JZ,KLK)-BOT(IZ,JZ,KLK))
        IF (KLK.EQ.1) THEN
           VKBOT=0.0
        ELSE
           VKBOT=VK(IZ,JZ,KLK-1)
        ENDIF
        V3=(AZ*VK(IZ,JZ,KLK)+(1-AZ)*VKBOT)/E
        GOTO 2948
      ENDIF
      IF ((Z(K).GE.TOP(IZ,JZ,KLK)).AND.(Z(K).LT.BOT(IZ,JZ,KLK+1))) THEN
        LL=KLK+NL
        V3=VK(IZ,JZ,KLK)/E
        GOTO 2948
      END IF
 2941 CONTINUE
      IF (Z(K).GE.BOT(IZ,JZ,NL)) THEN
        LL=NL
        AZ=(Z(K)-BOT(IZ,JZ,NL))/THICK(IZ,JZ,NL)
        V3=(AZ*VK(IZ,JZ,NL)+(1-AZ)*VK(IZ,JZ,NL-1))/E
      ENDIF
C  IF IN CONFINING LAYER, THEN SKIP HORIZONTAL INTERPOLATION
 2948 IF (LL.GT.NL) THEN
         V1=0
         V2=0
         GOTO 2990
      ENDIF
 2950 AX=XP-INT(XP)
 2960 AY=YP-INT(YP)
C HORIZONTAL VELOCITY INTERPOLATION
 2970 V1=((1-AY)*VI(I,J,LL)+AY*VI(I,J+1,LL))/E
 2980 V2=((1-AX)*VJ(I,J,LL)+AX*VJ(I+1,J,LL))/E
 2990 VX=VX+V1
 3000 VY=VY+V2
C APPLY RETARDATION COEFFICIENTS TO VELOCITY
      VX=VX/RETARD(LL)
      VY=VY/RETARD(LL)
      VZ=(VZ+V3)/RETARD(LL)
C CALCULATE F TIME USAGE (TAKE SMALLEST OF D, DMAX/VX, DMAX/VY, ZMAX/VZ)
 3040 IF ((D-ABS(DMAX/VX)).LT.0) GOTO 3070
 3050 F=ABS(DMAX/VX)
 3060 GOTO 3080
```

```
 3070 F=D
 3080 IF ((F-ABS(DMAX/VY)).LT.0) GOTO 3092
 3090 F=ABS(DMAX/VY)
 3092 IF ((F-ABS(ZMAX/VZ)).LT.0) GOTO 3100
      F=ABS(ZMAX/VZ)
C  CALCULATE REMAINING TIME IN MOVE
C      write(*,*) ' d ',d, ' minus f ',f,' equals ',d-f
 3100 D=D-F
 3110 DX=VX*F
 3120 DY=VY*F
      DZ=VZ*F
 3130 PHI=ATAN(DY/DX)
 3140 DD=SQRT(DX*DX+DY*DY)
      DDX=SQRT(DX*DX+DY*DY+DZ*DZ)
 3210 RN=0
 3220 GOTO 3280
C RANDOM NUMBER SUBROUTINE REMOVED HERE
 3280 IF (DL.GT.0) CALL RAND(RN)
 3290 RL=(SQRT(2*DL*DD)/DD)*RN
 3300 IF (DT.GT.0) CALL RAND(RN)
 3310 RT=(SQRT(2*DT*DD)/DD)*RN
 3312 IF (DV.GT.0) CALL RAND(RN)
 3314 RV=(SQRT(2*DV*DDX))*RN
 3340 OLDX = X(K)
 3350 OLDY = Y(K)
      OLDZ = Z(K)
 3360 IF (I.GE.NC) THEN
         XP=NC+.5
         GOTO 3380
      ENDIF
 3370 XP=XP+(DX+RL*DX+RT*DY)/DELX
 3380 IF (J.GE.NR) THEN
        YP=NR+.5
        GOTO 3400
      ENDIF
 3390 YP=YP+(DY+RL*DY-RT*DX)/DELY
C EDGE BOUNCE
 3400 IF (XP.LE.1) XP=1.01
 3410 IF (XP.GE.NC-.01) XP=NC-.01
 3420 IF (YP.GE.NR-.01) YP=NR-.01
 3430 IF (YP.LE.1) YP=1.01
C  MOVE PARTICLE IN Z DIMENSION
      Z(K)=Z(K)+(DZ+RV)
C  CHECK TO SEE IF PARTICLE HAS BEEN MOVED ABOVE WATER TABLE
      IF (Z(K).GT.(BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL))) THEN
        Z(K)=BOT(IZ,JZ,NL)+THICK(IZ,JZ,NL)-.001
        WRITE(*,*) 'POP DOWN'
      ENDIF
C  CHECK TO SEE IF PARTICLE HAS BEEN MOVED BELOW BOTTOM
      IF (Z(K).LT.BOT(IZ,JZ,1)) THEN
        Z(K)=BOT(IZ,JZ,1)+.001
        WRITE(*,*) 'POP UP'
```

```
      ENDIF
C  CHECK TO SEE IF PARTICLE HAS MOVED OFF OF PLAYING FIELD
      IF (  (X(K).LT.(LLX + DELX/2))
     $.OR.  (X(K).GT.(LLX + (NC-1)*DELX + DELX/2))
     $.OR.  (Y(K).LT.(LLY + DELY/2))
     $.OR.  (Y(K).GT.(LLY + (NR-1)*DELY + DELY/2))   ) THEN
         WRITE(*,*) 'PARTICLE MOVED OFF GRID'
         ISUNK=2
         GOTO 2790
      ENDIF
C NEWZ
      NEWZ=Z(K)
C  THESE 4 STATEMENTS MOVE THE PARTICLE OUT OF A ZERO VELOCITY FIELD
C  IF THEY HAPPEN TO GET THERE
 3432 III=INT(XP)
 3434 JJJ=INT(YP)
C  SKIP THIS CHECK IF PARTICLE IN CONFINING LAYER
      IF (LL.GT.NL) GOTO 3440
C MOVE PARTICLE OUT OF ZERO VELOCITY GRID
 3436 IF ((VI(III,JJJ,LL).EQ.0).AND.(VJ(III,JJJ,LL).EQ.0).AND.
     $(VI(III,JJJ+1,LL).EQ.0).AND.(VJ(III+1,JJJ,LL).EQ.0)) THEN
C          ISUNK=2
          WRITE(*,*) 'CALLING ZEROVG'
          WRITE(10,*) 'CALLING ZEROVG'
          CALL ZEROVG(XP,YP,III,JJJ,LL,NC,NR)
      ENDIF
C  TRANSLATE PARTICLE POSITION BACK TO REAL COORDINATES
 3440 NEWX = (XP-.5)*DELX+LLX
 3450 NEWY = (YP-.5)*DELY+LLY
 3460 X(K)=NEWX
 3470 Y(K)=NEWY
C      write(*,*) ' x, y, z ',x(k),y(k),z(k)
C      write(*,*) ' i, j, ll ', i , j, ll
C  CHECK TO SEE IF PARTICLE HAS STALLED
      IF (  (X(K).EQ.OLDX).AND.(Y(K).EQ.OLDY).AND.(Z(K).EQ.OLDZ)  ) THEN
         WRITE(*,*) 'PARTICLE STALLED'
         WRITE(10,*) 'PARTICLE STALLED'
         ISUNK=2
         GOTO 2790
      ENDIF
C THIS CODE REMOVES THE PARTICLE IF NEAR A SINK
C
C  IF PARTICLE IS IN CONFINING LAYER SKIP THE SINK CAPTURE
      IF (LL.GT.NL) THEN
         GOTO 3580
      ENDIF
 3510 DO 3570 LP=1,NS
C  FIND ROW AND COLUMN OF SINK
```

```
 3520      II=INT((X1(LP)-LLX)/DELX+1.0)
 3525      JJ=INT((Y1(LP)-LLY)/DELY+1.0)
C  FIND LAYER OF SINK
           KK=Z1(LP)
 3540     IF ((X(K).EQ.X1(LP)).AND.(Y(K).EQ.Y1(LP)).AND.(Z(K).LT.
     $    TOP(II,JJ,KK)).AND.(Z(K).GE.BOT(II,JJ,KK))) THEN
             GOTO 3600
C  REMOVE PARTICLE
          ENDIF
 3530     R3=SQRT((X(K)-X1(LP))*(X(K)-X1(LP))+(Y(K)-Y1(LP))
     $    *(Y(K)-Y1(LP)))
C  CHECK IF PARTICLE REACHES SINK DURING TIME STEP
C  PARTICLE MUST BE WITHIN MAX MOVE OF WELL TO BE CAPTURED
          CAPTUREMAX=DMAX
C  SET BOTTOM OF LAYER AS CAPTURE ZONE - ASSUMES FULL PENETRATION
          BOTCAPTURE=BOT(II,JJ,KK)
C   SET TOP OF LAYER AS CAPTURE ZONE
 3560     IF (KK.LT.NL) THEN
 3561        TOPCAPTURE=TOP(II,JJ,KK)
          ELSE
C  IF RIVER IN TOP LAYER THEN MODIFY CAPTURE ZONE
 3562        TOPCAPTURE=BOT(II,JJ,KK)+THICK(II,JJ,KK)
 3563        IF (TOP(II,JJ,NL).NE.0) THEN
 3564            R4=QSUM(LP)*F/DELX/DELY/7.48/E
                 BOTCAPTURE=TOP(II,JJ,NL)-R4
 3565            CAPTUREMAX=SQRT(DELX*DELY)
                 IF (BOTCAPTURE.LT.BOT(II,JJ,NL)) THEN
                      BOTCAPTURE=BOT(II,JJ,NL)
                 ENDIF
             ENDIF
          ENDIF
C  SKIP CAPTURE IF INJECTION WELL
 3566 IF (QSUM(LP).LE.0) THEN
          GOTO 3570
      ENDIF
C  REMOVE PARTICLE IN SINK
 3567 IF ((R3.LT.SQRT(QSUM(LP)*DELTA/(3.141592*THICK(II,JJ,KK)*E*7.48)))
     $.AND.(R3.LT.CAPTUREMAX).AND.(Z(K).GE.BOTCAPTURE).AND.
     $(Z(K).LE.TOPCAPTURE)) GOTO 3600
 3570 CONTINUE
C  LOOP BACK TO FINISH MOVE FOR REMAINING TIME INCREMENT
 3580 IF (D.GT.0) GOTO 2890
C       pause
C       write(*,*) 'going for next particle'
C LOOP BACK FOR NEXT PARTICLE
 3590 GOTO 2790
C!!!
 3600 WRITE(10,*) 'PARTICLE EXITED AT SINK NUMBER ',LP
      ISUNK=1
C CALC CONCENTRATION ENTERING SINK
 3610 CONC(LP)=CONC(LP)+(PM*119872)/(QSUM(LP)*DELTA)
C  REMOVE PARTICLE BY
```

```
 3620 X(K)=X(NP)
C  PUTTING LAST PARTICLE IN POSITION
 3630 Y(K)=Y(NP)
C  OF REMOVED PARTICLE
      Z(K)=Z(NP)
 3640 NP=NP-1
C  LOOP BACK TO NEXT PARTICLE
 3650 GOTO 2830
C  END OF PARTICLE MOVE AND BEGIN PRINTOUT OF SINK CONCENTRATIONS.
 3660 CONTINUE
C       write (*,*) ' end of time ',t2
C 3670 WRITE(10,*) 'NP=',NP
 3690 DO 3715 I= 1,NS
 3700   IF ((QSUM(I).GT.0).AND.(CONC(I).GT.0)) THEN
          WRITE(10,*) 'CONCENTRATION IN PUMPED WELL NUMBER ',I,
     $    ', IN PPM, IS= ',CONC(I)
        ENDIF
 3710   IF ((QSUM(I).GT.0).AND.(CONC(I).GT.0)) THEN
          WRITE(10,*) 'MASS EXITING IN PUMPED WELL NUMBER ',I,
     $    ', IN LBS, IS= ',CONC(I)*QSUM(I)*DELTA/119872
        ENDIF
 3715 CONTINUE
 3720 CONTINUE
C 3720 WRITE(10,*)
C       WRITE(10,*)
C       WRITE(10,*)
C       WRITE(10,*)
C  END OF MOVE SUBROUTINE
C SET ALL PARTICLES TO SLUG TYPE AT END OF MOVE
 3830 DO 3850 I=1,NP
 3840  SWITCH(I)=0
 3850 CONTINUE
C  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  +  BUG OUT TO MAIN MENU AS THIS IS THE END OF SUBROUTINE MOVE.  +
C  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 3910 RETURN
      END
```

APPENDIX III

APPENDIX III: SUMMARY OF TRAVEL TIMES

A. TRAVEL TIMES FOR PARTICLES DISCHARGED

Travel Time	Number of Nodes
100	15141
200	3224
300	1695
400	939
500	554
600	324
700	217
800	142
900	91
1000	68
1100	49
1200	44
1300	23
1400	27
1500	18
1600	18
1700	16
1800	13
1900	10
2000	15
2100	12
2200	12
2300	14
2400	16
2500	12
2600	17
2700	10
2800	14
2900	14
3000	16

3100	16
3200	14
3300	14
3400	13
3500	15
3600	13
3700	13
3800	13
3900	15
4000	12
4100	11
4200	15
4300	12
4400	10
4500	7
4600	11
4700	9
4800	3
4900	3
5000	2
5100	1
5500	2
5600	2
5700	2
5900	2
6100	2
6200	1
6400	2
6600	4
6700	5
7000	1
7100	2
7300	1
7400	1
7700	2
7800	1
8000	1
8700	1
8800	1
8900	3
9000	1
9400	1
10700	1

10900	1
12200	1
15600	1
15800	1
16000	1
17200	1
17400	1
17600	1
17700	1
17800	1
19400	2
19700	1
	Sub-total Nodes: 23039

B: TRAVEL TIMES FOR PARTICLES THAT MOVED OFF OF THE GRID

Travel Time	Number of Nodes
100	163
200	5
300	16
400	10
500	13
600	15
700	11
800	9
900	10
1000	11
1100	16
1200	12
1300	13
1400	13
1500	14
1600	8
1700	9
1800	7
1900	10
2000	9
2100	11
2200	9
2300	11
2400	10
2500	13
2600	10
2700	13

2800	10
2900	13
3000	10
3100	13
3200	13
3300	13
3400	12
3500	13
3600	12
3700	13
3800	12
3900	14
4000	10
4100	14
4200	11
4300	10
4400	8
4500	11
4600	6
4700	6
4800	5
4900	4
5000	1
5100	1
5400	1
5500	1
6100	1
7500	1
10500	1
15500	1
16000	1

C. TRAVEL TIMES EXCEEDING 20000 YEARS:

Travel Time	Number of Nodes
>20000	579

TOTAL NODES: 24311

APPENDIX IV

APPENDIX IV: GLACIER.FOR Fortran Code

```
      REAL X(2),Y(2),Z,THICK(361,470),MINDIS(361,470)
      REAL MINX,MAXX,MINY,MAXY,MINZ,MAXZ,DELX,DELY
      REAL PYTHAG,XSTOR,YSTOR
      INTEGER FLAG,GRDX,GRDY,COUNT,INOUT
      INTEGER IBOOL(361,470)
      CHARACTER*40 FILIN, BNDRY
      CHARACTER*12 OUTFIL,     GRDFIL
      CHARACTER    OUTNAM(12), GRDNAM(12)
      CHARACTER*8  FILOUT, SHITOUT
      CHARACTER*4  OUT, GRD, BINAS
      COMMON MINX,MAXX,MINY,MAXY,DELX,DELY,GRDX,GRDY,IBOOL,MINDIS
      EQUIVALENCE (OUTNAM(1),FILOUT),(OUTNAM(9),OUT)
      EQUIVALENCE (OUTFIL,OUTNAM(1))
      EQUIVALENCE (GRDNAM(1),SHITOUT),(GRDNAM(9),GRD)
      EQUIVALENCE (GRDFIL,GRDNAM(1))
C INITIALIZE COUNTER AND DOMAIN AND FILE VARIABLES
      COUNT = 0
      OUT = '.DIS'
      GRD = '.GRD'
      DELX=1000
      DELY=1000
      GRDX=361
      GRDY=470
      MINX=290568
      MAXY=557128
      MAXX = MINX + (GRDX-1)*DELX
      MINY = MAXY - (GRDY-1)*DELY
C INITIALIZE MINDIS ARRAY FOR RASTER GRID TO MAX GRID-ENCLOSED LENGTH
      A=MAXX-MINX
      B=MAXY-MINY
      DIST=PYTHAG(A,B)
      DO 10 J = 1,GRDY
        DO 20 I = 1,GRDX
          MINDIS(I,J)=DIST
 20     CONTINUE
 10   CONTINUE
C BINARY OR ASCII VARIABLE IS DSAA FOR ASCII OR DSBB FOR BINARY
      BINAS='DSAA'
      WRITE (*,*) ' *************************** *'
      WRITE (*,*) ' *         GLACIER                              *'

      WRITE (*,*) ' *           by                                    *'
```

```
      WRITE (*,*) ' *       John Hoaglund                         *'
      WRITE (*,*) ' *****************************'
      WRITE (*,*) 'ENTER THE MORAINE CRSTM-XYZ INPUT FILENAME: '
      READ(*,'(A)') FILIN
      WRITE(*,*) 'ENTER THE FIRST 8 CHARACTERS OF THE OUTPUT FILENAME: '
      READ(*,'(A8)') FILOUT
      SHITOUT = FILOUT
      WRITE(*,'(A)') OUTFIL
      WRITE(*,'(A)') GRDFIL
C OPEN FILES
C OPEN INPUT FILE OF CRSTM XYZ COORDINATES
      OPEN (11,FILE=FILIN,STATUS='OLD')
C OPEN MINDIS VALUE MATRIX FILE
      OPEN (12,FILE=OUTFIL,STATUS='UNKNOWN')
C OPEN SURFER GRID FILE
      OPEN (17,FILE=GRDFIL,STATUS='UNKNOWN')

C READ ACTIVE PIXELS FROM A BOOLEAN MATRIX FILE
      WRITE(*,*)'ENTER THE NAME OF THE FLOOD FILLED BOOLEAN ARRAY FILE'
      READ(*,'(A)') BNDRY
      OPEN (13,FILE=BNDRY,STATUS='UNKNOWN')
      DO 7700 J=1,GRDY
         READ (13,2700) (IBOOL(I,J),I=1,GRDX)
 7700 CONTINUE
C MAIN LOOP
C READ A CRSTM XYZ FILE, CHECK DISTANCES
      K=1
 1000 READ (11,*,END=1010) XSTOR,YSTOR
      COUNT = COUNT + 1
      WRITE (*,*) 'ENDPOINT # ',COUNT
      X(K)=XSTOR
      Y(K)=YSTOR
C CALCULATE NEAREST I J ADDRESS OF LINE SEGMENT ENDPOINT
      I = NINT((X(K)-MINX)/DELX) + 1
      J = NINT((MAXY-Y(K))/DELY) + 1
C ENTER 1 IN BOOLEAN ARRAY
C      IBOOL(I,J) = 1
       IF (COUNT.EQ.1) THEN
C             X0=X(K)
C             Y0=Y(K)
              K=K+1
              GOTO 1000
       ENDIF
C CALCULATE DISTANCE TO LINE SEGMENT
C SWEEP RASTER GRID TO FIND ALL BOUNDARY
C PIXELS (PIXELS WITHIN DELX OF THE LINE SEGMENT)
C AND UPDATE MINDIS FOR ENTIRE CURVE FOR EACH RASTER GRID NODE
      CALL RSTRDS(X(1),Y(1),X(2),Y(2))
C IF K=1, UPDATE TO 2; IF K = 2 RESET TO 1
      IF (K.EQ.1) THEN
          K=2
      ELSE
```

```
          K=1
      ENDIF
      GOTO 1000
 1010 CONTINUE

C CALCULATE ICE THICKNESSES FROM IBOOL AND MINDIS
      WRITE (*,*) 'FOR DISTANCES FROM ICE MARGIN IN METERS,'
      WRITE (*,*) 'ENTER THE ICE PROFILE CONSTANT, A:'
      READ (*,*) A
      MINZ=0
      MAXZ=MINZ
      DO 5555 J=1,GRDY
        DO 6666 I = 1, GRDX
            IF (IBOOL(I,J).NE.0) THEN
                THICK(I,J)= A*SQRT(MINDIS(I,J))
            ELSE
                THICK(I,J)=0.
            ENDIF
            IF (THICK(I,J).LT.MINZ) MINZ=THICK(I,J)
            IF (THICK(I,J).GT.MAXZ) MAXZ=THICK(I,J)
 6666   CONTINUE
 5555 CONTINUE
C WRITE THE MINDIS MATRIX AND THICKNESS MATRIX
      Z=0
      OPEN (14,FILE='GLACIER.THK',STATUS='UNKNOWN')

 1020 DO 2000 J = 1,GRDY
        WRITE(12,2500) (MINDIS(I,J), I = 1,GRDX)
        WRITE(14,2500) (THICK(I,J), I = 1,GRDX)
 2000 CONTINUE
C ??????FORMAT WIDTH TOO WIDE TO DISPLAY WHOLE MATRIX??????
 2500 FORMAT(8E10.5)

C CONVERT A MATRIX TO A SURFER GRID
 2001 WRITE(17,'(A)') BINAS
      WRITE(17,*) GRDX,GRDY
      WRITE(17,*) MINX,MAXX
      WRITE(17,*) MINY,MAXY
      WRITE(17,*) MINZ,MAXZ
C NOTE SURFER GRIDS ARE STORED WITH INCREASING Y FROM TOP DOWN
      DO 2010 J = GRDY,1,-1
         WRITE(17,2500) (THICK(I,J), I = 1,GRDX)
 2010 CONTINUE

C ??????FORMAT WIDTH TOO WIDE TO DISPLAY WHOLE MATRIX??????
 2700 FORMAT(25I3)
C CLOSE FILES AND QUIT
      CLOSE(11)
      CLOSE(12)
      CLOSE(17)
      CLOSE(13)
      CLOSE(14)
```

```
      STOP
      END
      FUNCTION PYTHAG(A,B)
      REAL A,B,PYTHAG
C THIS FUNCTION FROM PRESS ET. AL. NUMERICAL RECIPES IN FORTRAN
C COMPUTES PYTHAGOREAN ROOT WITHOUT DESTRUCTIVE UNDERFLOW OR OVERFLOW
      REAL ABSA,ABSB
      ABSA=ABS(A)
      ABSB=ABS(B)
      IF (ABSA.GT.ABSB) THEN
         PYTHAG=ABSA*SQRT(1.+(ABSB/ABSA)**2)
      ELSE
        IF (ABSB.EQ.0.) THEN
           PYTHAG=0.
        ELSE
           PYTHAG=ABSB*SQRT(1.+(ABSA/ABSB)**2)
        ENDIF
      ENDIF
      RETURN
      END

      SUBROUTINE RSTRDS(X1,Y1,X2,Y2)
      REAL MINX,MAXX,MINY,MAXY,DELX,DELY,MINDIS(361,470)
      REAL MDIST,DIST,X1,Y1,X2,Y2
      REAL X, Y
      REAL XI,YI,T,BPLC,BPLI,BPLJ,BPDI,BPDJ
      REAL PYTHAG
      INTEGER GRDX,GRDY,IBOOL(361,470)
      COMMON MINX,MAXX,MINY,MAXY,DELX,DELY,GRDX,GRDY,IBOOL,MINDIS
C  INTEGER ADDRESS FROM REAL COORDINATE
C      I = NINT((X(K)-MINX)/DELX) + 1
C      J = NINT((MAXY-Y(K))/DELY) + 1
C  REAL COORDINATE FROM INTEGER ADDRESS
C       X(I,J) = (I-1)*DELX + MINX
C       Y(I,J) = MAXY-(J-1)*DELY
      DO 100 J = 1 , GRDY
        DO 200 I = 1,GRDX
C  CALCULATE X AND Y FOR THE NODE
          X = (I-1)*DELX + MINX
          Y = MAXY-(J-1)*DELY
C  CALCULATE DISTANCE TO EACH ENDPOINT OF THE (SINGULAR) LINE SEGMENT
C  STORE MINIMUM
          A=X-X1
          B=Y-Y1
          MDIST=PYTHAG(A,B)
          A=X-X2
          B=Y-Y2
          DIST=PYTHAG(A,B)
          IF (DIST.LT.MDIST) MDIST=DIST
C  CALCULATE MINIMUM DISTANCE TO LINE AT R0.
C  USING PARALLEL-PERPENDICULAR DECOMPOSITION
C  B IS  (X-X1)i + (Y-Y1)j
```

```
C  V IS (X2-X1)i + (Y2-Y1)j
C CALCULATE B PARALLEL COEFFICIENT AND COMPONENTS
          A=(X2-X1)
          B=(Y2-Y1)
          BPLC=((X-X1)*A+(Y-Y1)*B)/(PYTHAG(A,B)**2)
          BPLI=BPLC*A
          BPLJ=BPLC*B
C CALCULATE B PERPENDICULAR COEFFICIENTS
          BPDI=(X-X1)-BPLI
          BPDJ=(Y-Y1)-BPLJ
C CALCULATE INTERSECTION POINT, R0
          XI=X1+BPLI
          YI=Y1+BPLJ
C CALCULATE T PARAMETER OF INTERSECTION POINT ON LINE OF LINE SEGMENT
          IF (A.EQ.0) THEN
             T=(YI-Y1)/B
          ELSE
             T=(XI-X1)/A
          ENDIF
C  IF R0 BETWEEN ENDPOINTS, RESET MDIST
          IF ((T.GT.0.).AND.(T.LT.1.)) THEN
             A=BPDI
             B=BPDJ
             MDIST=PYTHAG(A,B)
          ENDIF
C  FOR ANY POINT WITHIN HALF-DELX UPDATE BOOLEAN ARRAY
          IF (MDIST.LT.(DELX/2)) IBOOL(I,J) = 1
C  KEEP TRACK OF MINDIS IN ARRAY FOR COMPARISON WITH OTHER
C  LINE SEGMENTS (IE OTHER RSTRDS SUBROUTINE CALLS)
          IF (MDIST.LT.MINDIS(I,J)) MINDIS(I,J) = MDIST
 200    CONTINUE
 100  CONTINUE
      RETURN
      END
```

APPENDIX V

APPENDIX V:
Table of Contents

List of Figures

List of Tables

Digital Simulation of Ground-Water Flow in the Glaciofluvial, Saginaw, and Marshall Aquifers, Central Lower Peninsula of Michigan

By J.R. Hoaglund, G.C. Huffman, and N.G. Grannemann

ABSTRACT

A steady-state, numerical model was developed to simulate groundwater flow in four regional aquifers in Michigan's Lower Peninsula as part of the U.S. Geological Survey's Regional Aquifer-Systems Analysis program. The model treats Lakes Michigan, Huron, St. Clair, and Erie as well as the St. Clair and Detroit River connecting channels as constant-head boundaries. These boundaries represent three sides of the modeled area. Basin divides near the southern border of the Lower Peninsula were simulated as no-flow boundaries. Although the model was developed to quantify regional groundwater flow in the aquifer system, it also can be used to estimate groundwater discharge to the lakes and connecting channels. Recharge rates were developed by analyzing streamflow, precipitation, and basin-characteristics data for 114 basins where streamflow is unregulated. Groundwater discharge to approximately 2,000 stream segments also can be estimated in 44 major basins.

The MODFLOW computer code was used to calculate groundwater flow in the modeled area. Four layers were delineated so that each layer represents an aquifer. The model has about 106,000 active, equally spaced, 1-kilometer, finite-difference cells in the uppermost layer. About 1,500 cells are needed to represent the Great Lakes' shoreline and connecting channels.

INTRODUCTION

Pleistocene glacial deposits and Jurassic through Mississippian bedrock units form a regional system of aquifers and confining units in the central Lower Peninsula of Michigan that was studied as part of the Regional Aquifer-System Analysis (RASA) program of the U.S. Geological Survey (USGS). The Michigan Basin RASA project is one of 28 USGS hydrogeologic investigations of regional aquifer systems of the United States (Sun and Weeks, 1991). The purposes of the RASA program are to define the regional geohydrology and geochemistry of the major regional ground-water systems of the United States and to establish a framework of background information that can be used for regional assessment of ground-water resources. As part of the RASA program, the Michigan Basin RASA project defined the geohydrology and geochemistry of aquifers in the central Lower Peninsula of Michigan that are primary sources of ground-water supply for human needs.

The Michigan Basin aquifer system is a major source of ground water in the central Lower Peninsula of Michigan. The study area includes 22,000 mi^2 of Michigan's Lower Peninsula (fig. 1). Three of the Great Lakes bound the Lower Peninsula. Lake Michigan borders the Lower Peninsula on the west side, Lake Huron borders the northeast side, and Lake Erie borders the southeast side. The St. Clair River, Lake St. Clair, and the Detroit River connect Lake Huron with Lake Erie on the east. A surface-water drainage

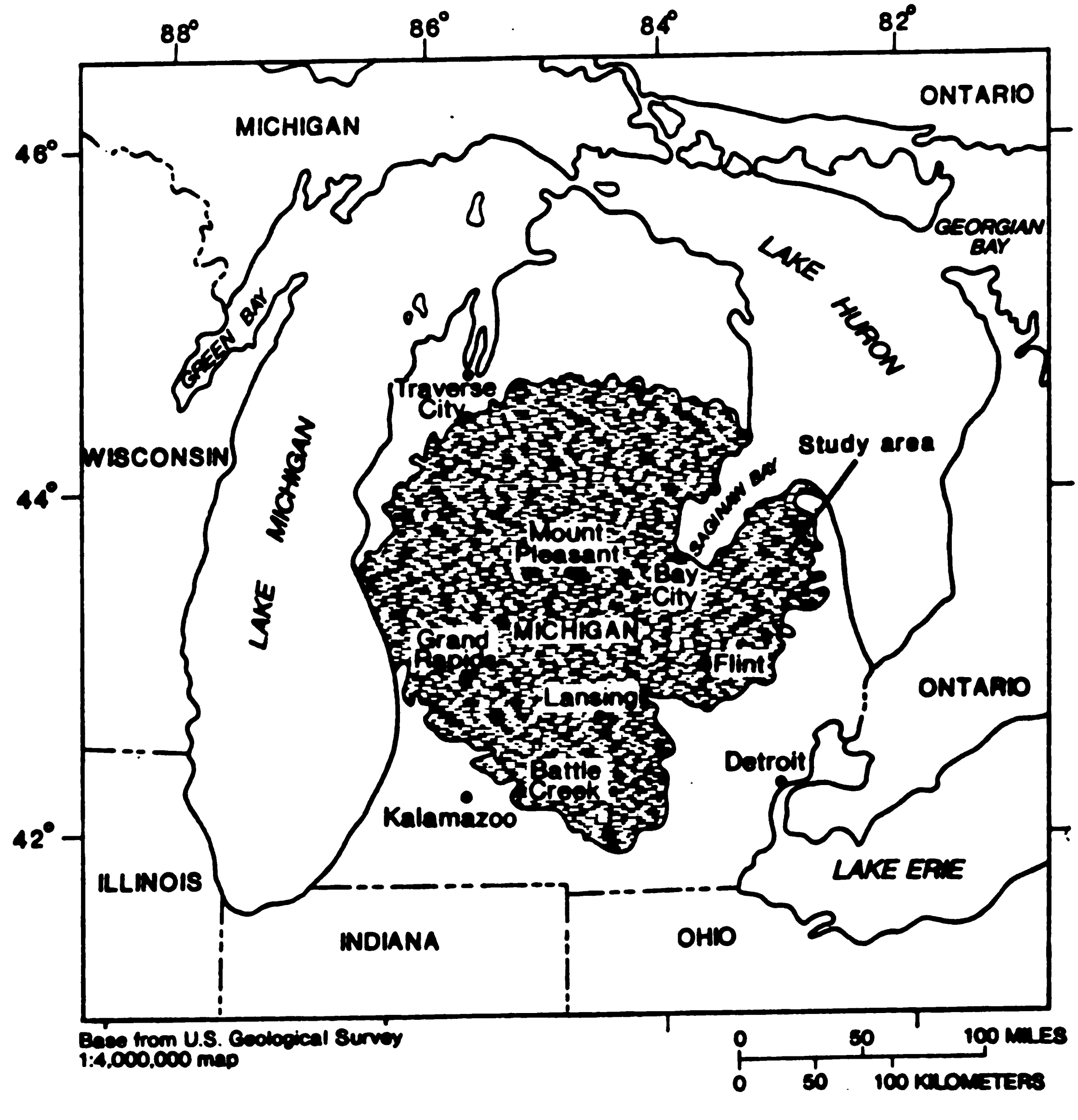

Figure 1. Location of the RASA Study Area in the Lower Peninsula of Michigan.

divide extends in a north--south direction, dividing the peninsula approximately in half. Rivers on the western half of the peninsula drain into Lake Michigan; those on the eastern half drain into Lakes Huron and Erie. An east-west trending surface-water divide that is near the southern border of Michigan separates water flowing northward in Michigan from that flowing southward into Indiana and Ohio.

Ground water has become an increasingly important source of water in Michigan. In 1987, more than 600,000 people in the study area used ground water for municipal supplies in 182 communities (Baltusis and others, 1992). In addition, rural water users are nearly entirely dependent on ground water for supply.

Purpose and Scope

A purpose of the Michigan Basin RASA study, as defined by Mandle (1986), is to "develop a regional ground-water-flow model to simulate present, and possibly, paleo-ground-water-flow directions" and to "evaluate hydrologic data through model simulation to suggest a feasible, effective network for monitoring future water-use, water levels, and changes in water quality." A better understanding of the natural flow system that existed prior to large-scale ground-water pumpage, as well as withdrawal of brine, gas, and oil, is the main result of this effort. Another major objective is to simulate flow conditions based on information obtained by analyzing the hydrogeologic framework (Westjohn and Weaver, 1996) and the geochemistry of ground water (Long and others, 1996) as well as from basic hydrologic principles.

This report summarizes the basic concepts of the predevelopment ground-water flow system, design and calibration of the digital model, and sensitivity of the model to selected hydrologic parameters and boundary conditions. It also describes the regional flow system as defined by the results of the calibrated model. Limitations for using the model are also reported.

Although the study area is 22,000 mi^2, the modeled area incorporates nearly the entire Lower Peninsula of Michigan so the Great Lakes could be used as boundaries on three sides of the model.

Previous Simulations of Ground-water Flow in the Michigan Basin

Simulation has been used to analyze regional flow of ground water at a few locations in Michigan, however, most simulations have been done to better understand ground-water conditions at sites of ground-water contamination. Vanlier and Wheeler (1968) developed an electric analog model for the Lansing metropolitan area as part of a study of Clinton, Eaton, and Ingham Counties. Holtschlag and Luukkonen (1996) are updating this model using digital modeling techniques. McDonald and Fleck (1978) developed a regional model that included the Marshall aquifer and overlying glaciofluvial deposits near Muskegon, Michigan. Stark and others (1983) developed a regional ground-water flow modei of the Wurtsmith Air Force Base area in the northeastern part of the study area. Grannemann and Twenter (1985) used a regional flow model of the Battle Creek area to analyze ground-water flow at Verona well field in southwestern Michigan. Lynch and Grannemann (1996) are updating this model. Mandle and Westjohn (1989) developed a model to simulate ground-water flow for the entire study area as part of the RASA project. The simulations documented in this report are continuation of the work started by Mandle and Westjohn for the RASA project.

Acknowledgments

HYDROLOGY

Physiography

The entire Lower Peninsula of Michigan, which is in the Central Lowland physiographic province (Fenneman, 1938, p. 559), is underlain by layered sedimentary rock and almost entirely overlain by glacial deposits. Continental glaciers scoured the bedrock surface and greatly altered the physiography of the study area. This scouring ultimately lead to the formation of the Great Lakes in preglacial valleys which were pathways for glacial ice advances. Ice advancing along these scoured valleys is the probable reason for the highly lobate character of late Wisconsinan ice advances which created the physiography of the area (Eschman, 1985, p. 162). Topographic features of the Lower Peninsula are related to deposition of glacial sediment during the last glaciation. Upland and lowland physiographic provinces were described by Leverett and Taylor (1915). Three lowland and two upland areas are shown on Figure 2. Land surface altitudes range from about 850 to 1,725 ft in the Northern uplands and from about 825 to 1,200 ft in the Southern uplands. Land surface altitudes in the Saginaw and Michigan lowlands range from 580 to about 700 ft; in the Erie lowlands they range from 572 to about 750 ft.

Lowland areas are generally flat to gently sloping toward Saginaw Bay, Lake Michigan, Lake Huron, Lake Erie or the connecting channels. Much of the lowland area is currently used for agriculture but urban land use is also important. The Southern uplands also primarily have agricultural and urban land uses but about one fourth of this area is forested. The Northern uplands are primarily forested.

Hydrogeologic Framework

The Michigan Basin is an ovate shaped accumulation of sedimentary rocks in the Lower Peninsula of Michigan and parts of Michigan's Upper Peninsula, Wisconsin, Illinois, Indiana, Ohio, and Ontario, Canada. The maximum thickness of Precambrian through Jurassic rocks is about 17,500 ft (Lillienthal, 1978). Paleozoic through Jurassic rocks are mantled by glacial deposits that are the result of the Wisconsinan and, possibly, earlier glaciations. Ice from the last glaciation receded from Michigan about 10,000 years ago

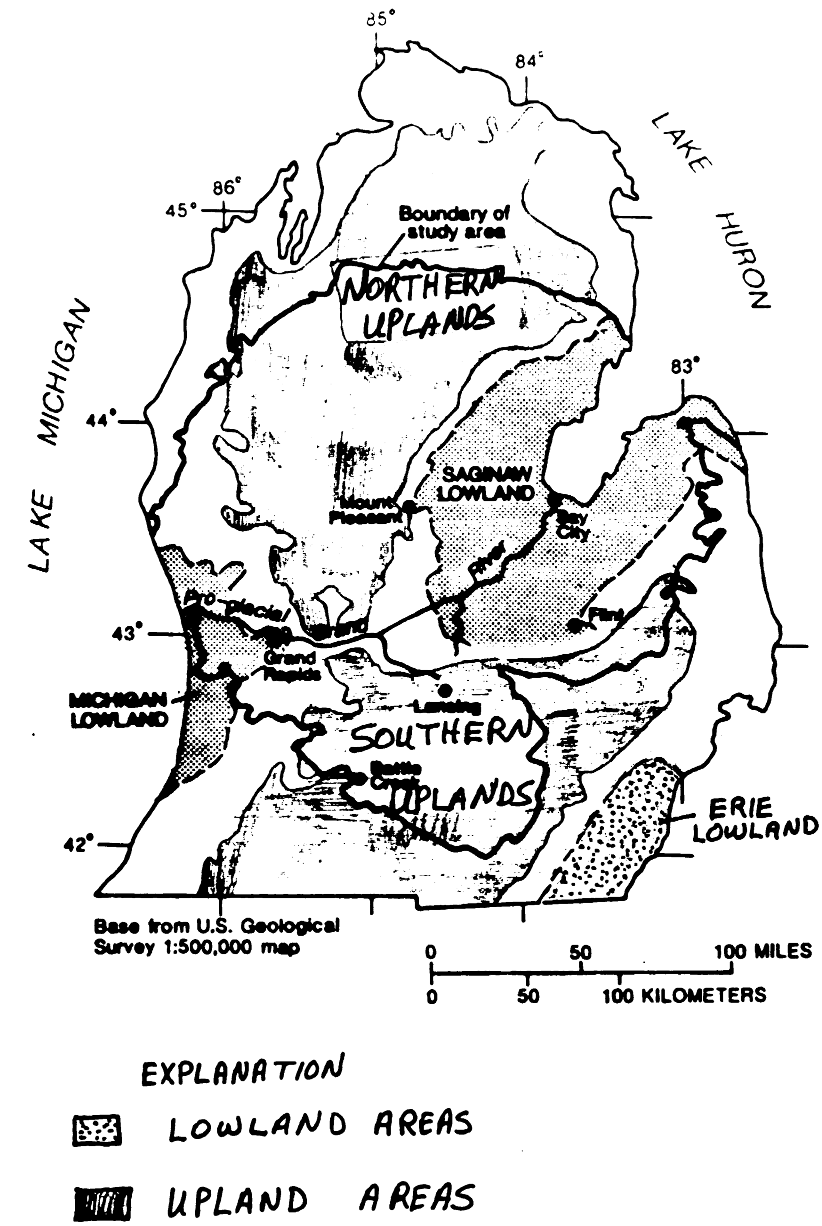

Figure 2. Physiographic features, Lower Peninsula of Michigan.

(Eschman, 1985, p. 164).

Four major aquifers are identified in the Michigan Basin RASA study area: the glaciofluvial, Saginaw, Parma-Bayport, and Marshall aquifers (fig. 3). Glaciofluvial aquifers dominantly consist of thick sequences of sand and gravel; however, in parts of the study area, they consist of sand and gravel beds within till or other fine-grained glacial deposits. For this study, the uppermost aquifer is referred to as a single unit even though it is composed of multiple sand and gravel layers. "Red beds" of Jurassic age overlie the Saginaw aquifer in the west-central part of the study area. These "red beds" are dominantly composed of red mud, poorly consolidated red shale, gypsum, and minor amounts of sandstone. More than 400 ft of freshwater-bearing glacial deposits overlie the Jurassic "red beds." Together with the fine-grained glacial deposits, "red beds" form subregional confining units (Westjohn and others, 1994).

Sandstones and shales of the Saginaw and Grand River Formations are intercalated and constitute the dominant lithology in these rock units. For characterization of the hydrogeological framework of the Michigan Basin aquifer system, the composite thickness of sandstone in these units was grouped to form the Saginaw aquifer (Westjohn and Weaver, 1996a). Sandstones, which are the most productive aquifer material in the unit, generally are less than 100 ft thick except in the east-central part of the basin, where the composite thickness of sandstone ranges from 200 to 370 ft (Westjohn and Weaver, 1996a).

In most areas of the basin, shale underlies the Saginaw aquifer. This shale, which

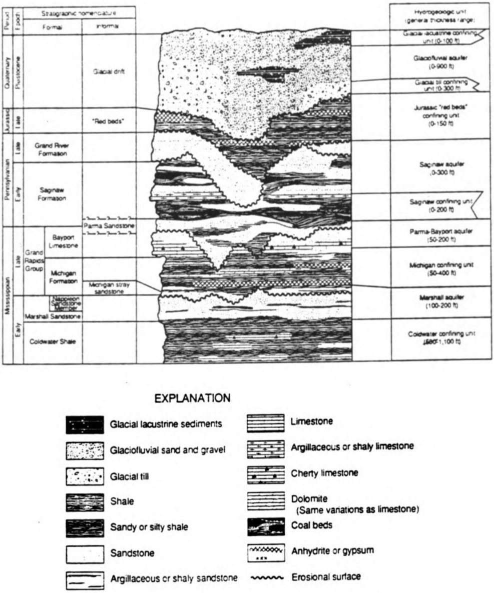

Figure 3.--Stratigraphic nomenclature, hydrologeologic units, and rock units for the RASA study area (from Westjohn and others, 1994).

constitutes the Saginaw confining unit and separates the Saginaw aquifer from the Parma-Bayport aquifer, ranges in thickness form 0 to 300 ft (Westjohn and Weaver, 1996a). The Parma-Bayport aquifer ranges from 100 to 150 ft in thickness and consists of the Parma Sandstone and the Bayport Limestone (Westjohn and Weaver, 1996a). The Parma Sandstone contains sandstone, shale, siltstone, and thin lenses of limestone. The Bayport Limestone is predominantly limestone, sandstone, and sandy limestone.

Underlying the Parma-Bayport aquifer is the Michigan confining unit, which is an intercalated sequence of thin bedded limestone, dolomite, shale, gypsum, anhydrite, and lenses of sandstone. The unit ranges in thickness from 50 ft near the fringes of the subcrop area to about 400 ft over the central part of the study area (Westjohn and Weaver, 1996b).

The Marshall aquifer is the lowermost aquifer in the RASA study area. It includes the Marshall Sandstone and sandstones that form the lower part of the Michigan Formation (fig. 3). The basal unit of the Marshall Sandstone consists of 50 to 100 ft of poorly permeable micaceous sandstone or micaceous siltstone that overlies the Coldwater Shale (Westjohn and Weaver, 1996b). Above this unit is a permeable, fine- to medium-grained sandstone which is generally 50 to 100 ft in thickness. This unit is commonly referred to as the lower Marshall Sandstone. The lithologic relations of strata that overlie this unit are complex, however, in most of the study area, a sandstone unit known as the upper Marshall or Napoleon Sandstone Member is present. Its thickness ranges from 50 to 125 ft and it is hydraulically similar to the lower Marshall Sandstone.

The Coldwater confining unit is primarily a shale with local occurrences of limestone, dolomite, and sandstone. Thickness of the Coldwater ranges from 500 ft in the east to 1,100 ft in the west. This unit forms the lower boundary of the aquifer system.

Because the modeled area was extended to the Great Lakes shoreline and south of the study area to near the border between Michigan and Indiana or Ohio, geologic data were obtained from existing sources for most of the modeled area that is not included in the study area.

Regional Water Budget

To help define the ground-water flow system, it is useful to estimate an average regional water budget. This hydrologic budget balances precipitation with the sum of evapotranspiration, overland flow to streams, and base flow of streams. Pumpage, which is estimated to be less than 0.1 in./yr (Baltusis and others, 1992), is a small fraction of the base flow and, therefore, is not included in the regional water budget. For this report, base flow is assumed to equal ground-water recharge under steady-state conditions.

Precipitation averages about 32 in./yr in the Lower Peninsula of Michigan and ranges from 28 to 40 in./yr (fig. 4). Total stream runoff averages about 12 in./yr and ranges from 8 to 16 in./yr. Base flow averages 8 in./yr and ranges from about 2 to 15 in./yr (Holtschlag, in press). An expression of the average hydrologic budget is illustrated in the following equation:

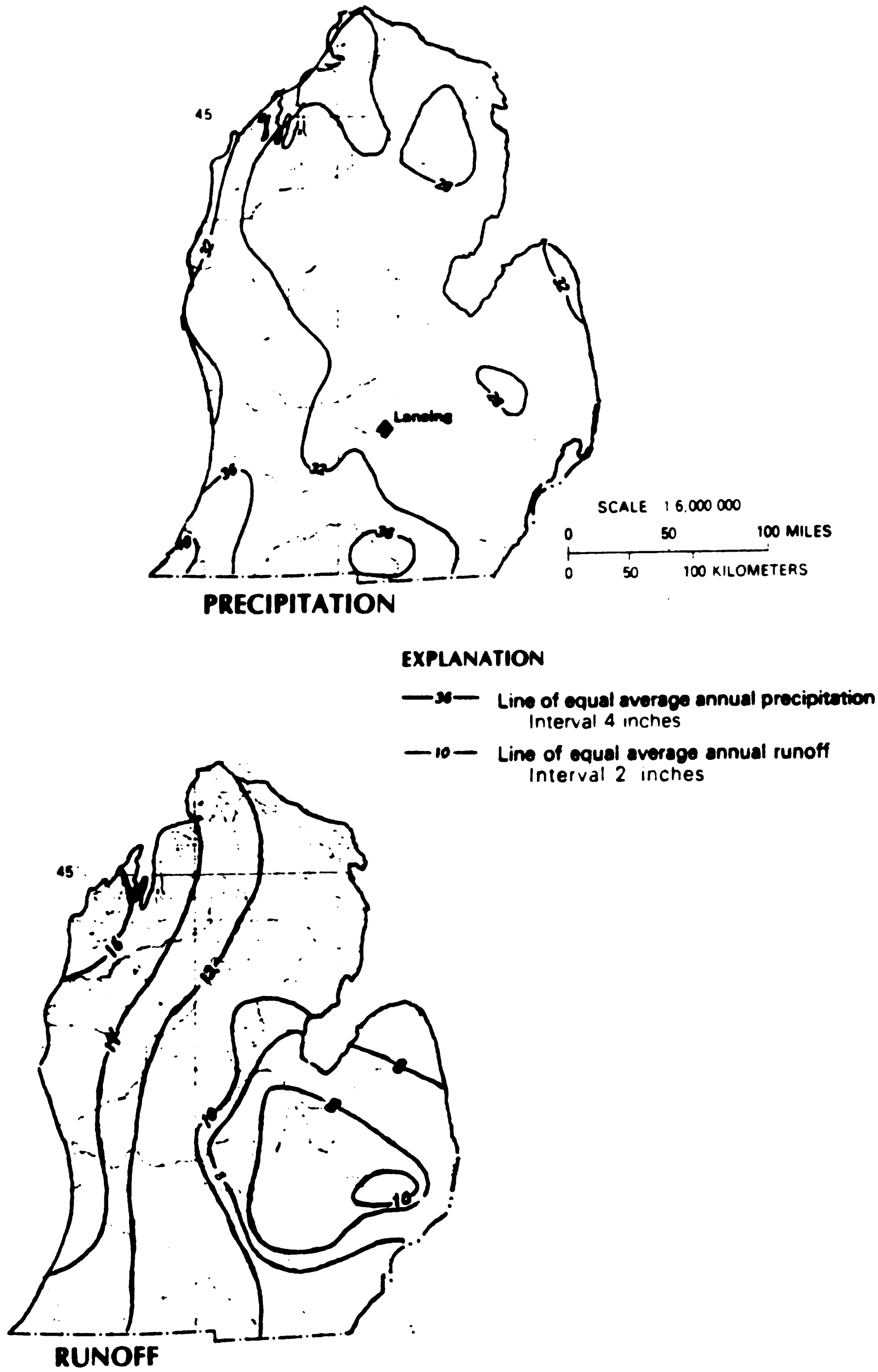

Figure 4.--Average annual precipitation and runoff in Michigan's Lower Peninsula, 1951-80. (Sources of data: Precipitation data from unpublished map by D.A. Olson, National Oceanographic and Atmospheric Administration. Runoff data from Gebert, Graczyk, and Krug (1985)).

Precipitation (32 in./yr) = Evapotranspiration (20 in./yr) + Overland runoff (4 in./yr) + Base flow (8 in./yr).

This estimate of the hydrologic budget provides a basis for developing a conceptual model of the hydrologic cycle for the Michigan Basin RASA study area. Local estimates for these rates may vary considerably from the averages.

Generalized Ground-Water Flow System

Difference in hydraulic head caused by topographic relief is the most significant driving force for ground-water flow in the aquifer system. Altitude of the land surface ranges from 1,725 ft in west-central part to 572 ft at Lake Erie in the southeastern part of the Lower Peninsula. Except for a few areas where bedrock is near land surface, the water table is in the glaciofluvial aquifer. The water table closely follows trends in the land surface. Figure 5 illustrates the relation between land surface elevation and elevation of the water table. The two main areas where the water table is highest coincide with the Northern and Southern uplands. Low water-table altitudes coincide with the Saginaw, Michigan, and Erie lowlands as well as with an elongated northeast-southwest depression that trends from Saginaw Bay to Lake Michigan. The depression is located in the proglacial Grand River valley, which is the site of the present day Grand, Maple, Bad, and Saginaw Rivers (fig. 2). Generalized ground-water flow directions in the glaciofluvial aquifer are toward the lowland areas, the Great Lakes, and the proglacial Grand River.

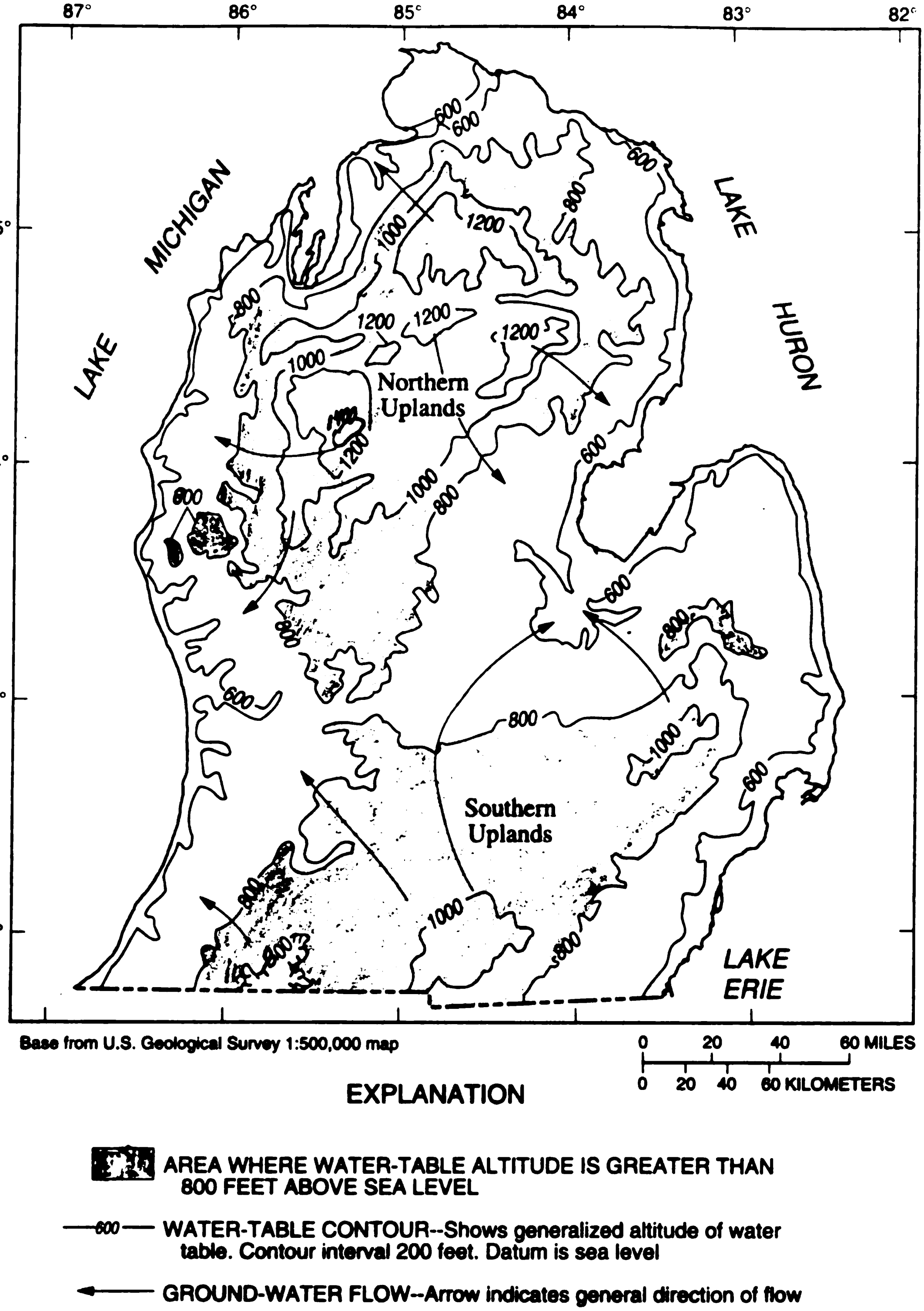

Figure 5.--Relation of water table to upland areas of the Lower Peninsula of Michigan (modified from Mandle and Westjohn, 1989).

Predevelopment freshwater head maps for the Saginaw and Marshall aquifers were prepared using data from reported and measured water-level records collected in the late 1800's and early 1900's and some records from early oil and gas exploration in the 1930's (Barton and others, 1996). The distribution of freshwater head in these two bedrock aquifers is similar to that of the water table although the magnitude of the heads are less (figs. 6 and 7). Vugrinovich (1986) noted a similar distribution of hydraulic heads. He concluded that predevelopment hydraulic head in both aquifers are generally in equilibrium with the present day land-surface elevations.

Within an aquifer system in which ground-water density is variable, the directions of ground-water movement may not be readily discernable by using a map of hydraulic head. Where ground-water density is relatively uniform, the ground-water movement is orthogonal to the hydraulic head contours. However, where ground-water density varies, the flow directions are not necessarily perpendicular to head contours because a driving force caused by density and the geometry of the aquifer may alter ground-water flow directions.

The Great Lakes form the lateral boundaries for ground-water flow on three sides of the Lower Peninsula. A no-flow boundary near the southern border of Michigan delineates flow in Indiana and extreme southern Michigan from flow in the study area. Vertically, the water table is the upper flow boundary and the Coldwater confining unit is the lower boundary. It is assumed that the thick shale sequence of the Coldwater Shale

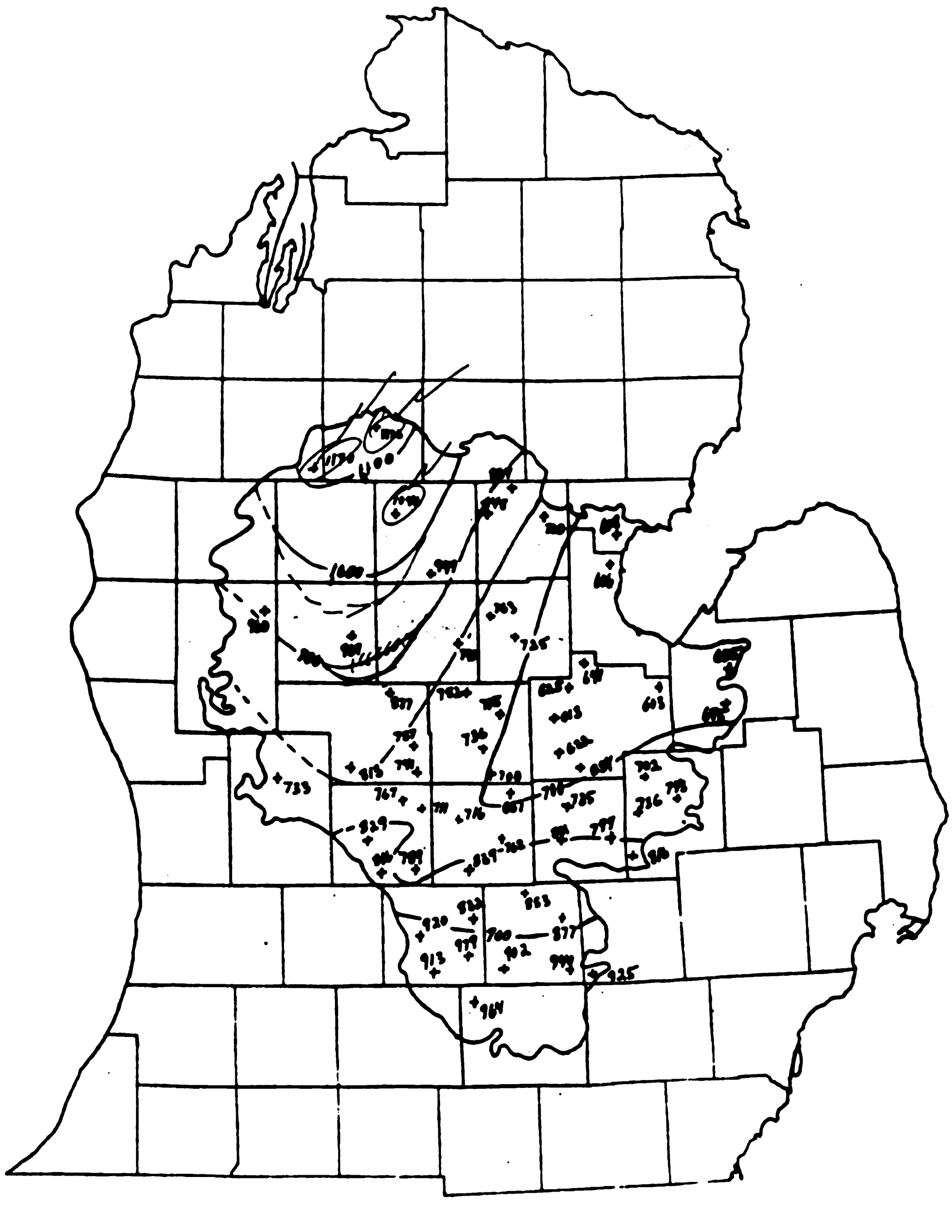

Figure 6.--Potentiometric surface for the Saginaw aquifer (from Barton and others, 1996).

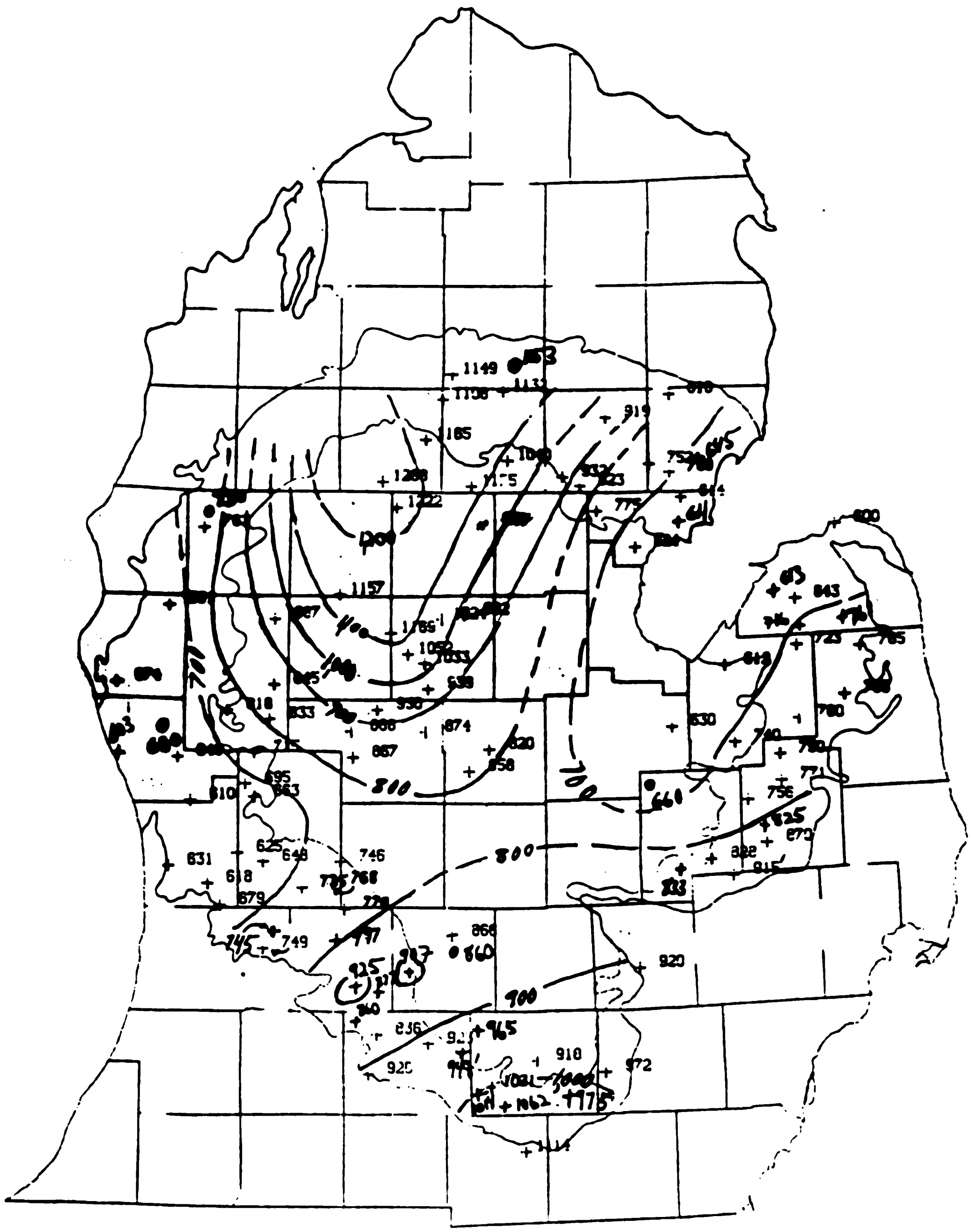

Figure 7.--Potentiometric surface for the Marshall aquifer (from Barton and others, 1996).

completely restricts water flowing vertically. Most ground-water flow in the glaciofluvial, Saginaw, Parma-Bayport, and Marshall aquifers is horizontal. Flow in the confining units that separate the aquifers is mostly vertical.

Saline Ground Water and Brine

A major hydrologic feature of the Michigan Basin aquifer system is the presence of saline water near surface in the lowland areas (Wahrer and others, 1996) and saline water and brine down dip in the Parma-Bayport and Marshall aquifers (Westjohn and Weaver, 1996c; Ging and others, 1996). Saline water also occurs in the Saginaw aquifer in the west-central part of the study area and in the Saginaw Lowlands (Westjohn and Weaver, 1996c; Meissner and others, 1996). Because of density differences related to higher dissolved-solids concentrations of these fluids, brine tends to accumulate down dip. It is unknown whether the brine had reached hydraulic equilibrium prior to large-scale withdrawals of brine for human uses, however, for the purpose of this study, densities of the fluids in the aquifer system are assumed to be constant in time so that the effects of solute transport on ground-water flow can be ignored.

SIMULATION OF GROUND-WATER FLOW

Ground-water flow in the aquifer system was simulated using a multilayer finite-difference model. A computer program, most commonly known as MODFLOW (McDonald and Harbaugh, 1988), was used. The original program was designed to simulate flow of water with constant density. Input to MODFLOW was modified to allow simulation of ground water whose density is variable from place to place but does not change with time. The procedure is similar to one used by Kontis and Mandle (1988) for the Trescott-Larson finite-difference program to simulate ground-water flow. For steady-state, predevelopment ground-water flow conditions, these assumptions are reasonable.

Prior to stresses caused by human activities--such as ground-water pumping, natural gas removal, brine withdrawal, and other alterations such as coal mining activity and agricultural drainage--the aquifer system was nearly in a state of equilibrium. At present, pumping and other alterations have changed the aquifer system so that the steady-state assumption may not apply. Therefore, the model was calibrated using data that duplicates predevelopment measured heads and flow conditions as closely as possible. Sensitivity and parameter analysis was conducted using MODFLOWP (Hill, 1992) to assess the effects of ranges of hydraulic characteristics on model behavior.

Model Description

The model was designed by defining boundary conditions, and spatial discretization for the finite-difference calculations, by assigning values for the hydraulic characteristics of the aquifers and confining units, and by estimating recharge rates for the ground-water system. The sources of data were evaluated during model design and analysis.

Finite-Difference Grid

The Michigan RASA model grid is comprised of square finite-difference cells that are 3,281 ft (1 km) on a side. There are 361 columns and 470 rows that correspond to a subset of the 633 column by 733 row Earth Resources Data Analysis System (ERDAS) data set that was developed by Michigan State University's Center for Remote Sensing (CRS) to cover the State of Michigan (Lusch and Enslin, 1984). The RASA subset covers the Lower Peninsula of Michigan and extends into the Great Lakes (fig. 8). RASA array index (1, 1) is the northwesternmost model cell and corresponds to CRS-ERDAS array index (273, 264). The 361 column by 470 row grid is flush with the ERDAS lower right corner. The Center for Remote Sensing has adopted a transverse mercator (TM) metric coordinate system with a basis at 86.0000 degrees west longitude and 44.0000 degrees north latitude with an easting of 359,987 m and a northing of 344,917 m (David Lusch, personal communication). The CRSTM coordinates of the ERDAS array are reported on the upper left corners of the cells. The basis for the CRSTM geodetic and CRSTM coordinates, and the upper left corner CRSTM coordinate of the model cell in row 1 and column 1, were originally reported by the Center for Remote Sensing. The RASA model

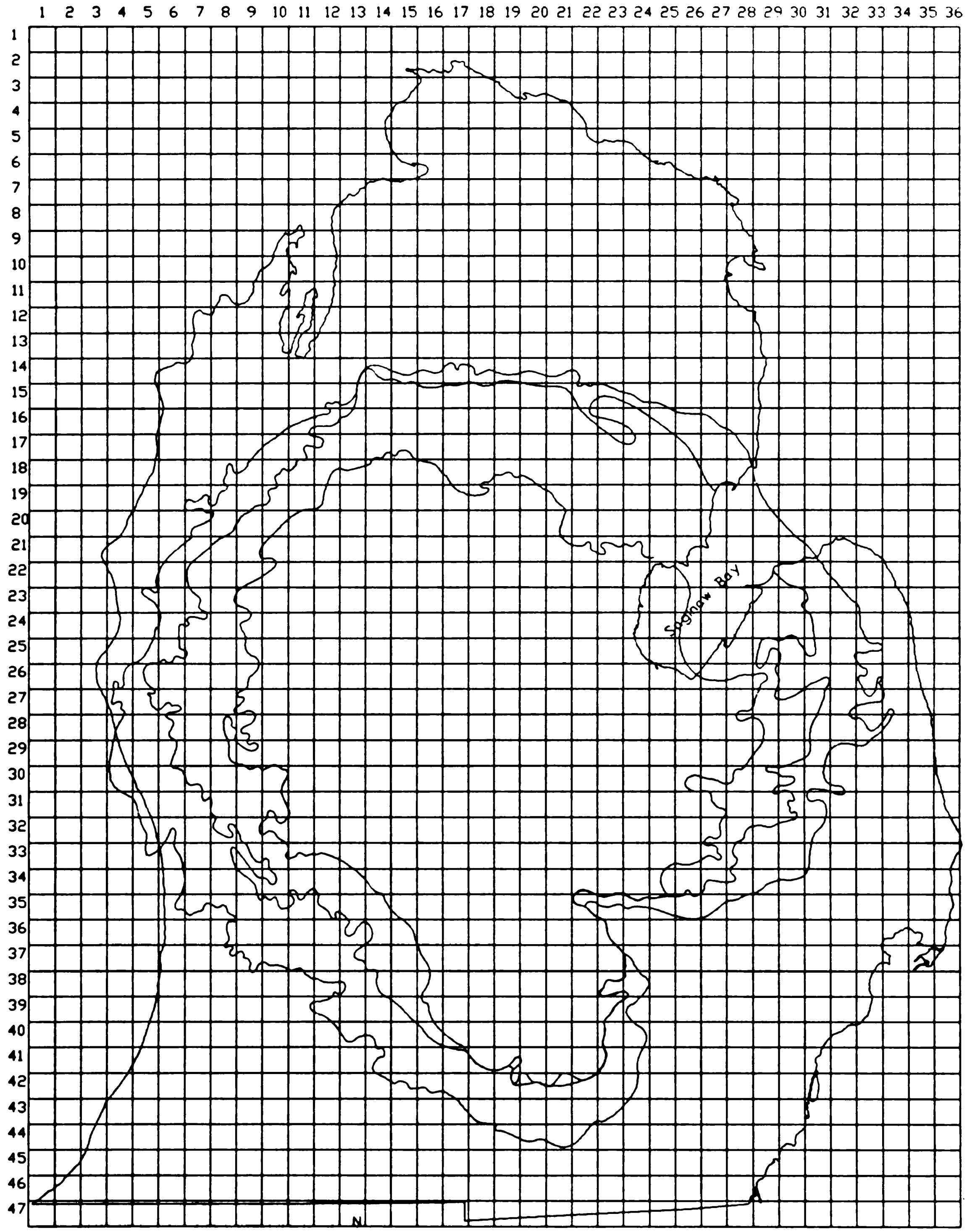

Figure 8.--Grid for RASA ground-water flow model. Each square represents 100 finite difference cells.

grid, four-corner CRSTM coordinates were calculated using 1,000 m increments from the reported CRSTM coordinate of the model cell in row 1, column 1. Geodetic coordinates were calculated from CRSTM coordinates using a TM-to-Geodetic algorithm as listed in computer program TM2GEO.FOR (Appendix A). RASA model coordinates are summarized in the following table for the four corners of the model grid as well as the digitizer origin used for maps digitized for the RASA study (table 1).

RASA geologic and potentiometric head data, including lines and points, were digitized from a 1:500,000 scale, Lambert conformal conic projection basemap provided by the Midcontinent Mapping Center of the Earth Science Information Center, Rolla, Missouri. The digitizing origin was placed at the base of the map along 87 degrees west longitude and the latitude was interpolated at 41.6094 degrees. The latitude and longitude were used in GEO2TM.FOR (Appendix A) to calculate the CRSTM coordinate of the digitizer origin. The x- and y-axis offsets from the digitizer origin to the lower left corner of the RASA model grid were calculated as 13,406 and 7,689 meters, respectively. The offsets were converted to 1.056 and 0.605 inches for the x- and y-axis, respectively. An angle of 0.83 degrees between the CRSTM x-axis and the digitizer x-axis was calculated by vector methods, plotting the upper left corners of the RASA model cell at row 1, column 470 and row 361, column 470 in digitizer coordinates and dotting the vector formed between these coordinates and x-axis vector. Digitizer coordinates were transformed to CRSTM coordinates by rotation transformation, translation transformation, and scale transformation. The best visual fit of the digitized state border

Table 1. Model grid coordinates, basis for coordinates, and digitizer origin relative to the Michigan State University Center for Remote Sensing Transverse Mercator (CRSTM).

RASA Model Grid Coordinates, Basis, and Digitizer Origin	CRSTM[a] Easting (meters)	CRSTM[a] Nothing (meters)	Latitude (degrees)	Longitude (degrees)
CRSTM[a] Basis	359987	344917	44.0000	86.0000
Centroid Row 1, Column 1	290568	557128	45.9068	86.8950
Upper Left Corner, Row 1, Column 1	290068	557628	45.9113	86.9015
Upper Right Corner, Row 1 Column 361	651068	557628	45.8533	82.2508
Lower Left Corner, Row 1, Column 470	290068	087628	41.6799	86.8400
Lower Right Corner, Row 470, Column 361	651068	087628	41.6298	82.5058
Digitizer Origin	276662	079939	41.6094	87.0000

[a] Center for Remote Sensing, Michigan State University, transverse mercator

line with the pixelated state outline supplied by the Center for Remote Sensing was acheived with offsets of 1.047 and 0.70 inches in the x and y coordinates, respectively. A rotation angle of 0.70 degrees was used.

Digital Representation of Model Layers

Geological surface configuration maps and isopach maps were digitally reproduced from the original hydrogeological reports for the glaciofluvial, Saginaw, Parma-Bayport, and Marshall aquifers and the confining units that separate them (Westjohn and others, 1994; Westjohn and Weaver, 1996a; and Westjohn and Weaver, 1996b). The surfaces were reproduced using computer program WANGRID.FOR (Appendix A), which uses the model grid and Gauss Seidel iteration to solve a finite-difference approximation of the LaPlace equation between specified-value (Dirichlet) and specified-rate (Neuman) boundaries. Surface configuration maps can be reproduced when the edges (boundary) of the map are treated as zero-specified-rate (no flow) boundaries and the contour lines as specified-value boundaries meeting the "no flow" boundaries at right angles. Isopach maps can be digitally reproduced when both the edges (specified as zero thickness at the boundary) of the map and contour lines are treated as specified-value boundaries. Since the finite-difference approximation to the LaPlace equation reduces to the average of the four surrounding nodes of a five star operator (Wang and Anderson, 1982), the solution becomes the linear interpolation between the contour lines. The linear interpolation is greatly improved if the original data is included in the input along with the digitized contour lines. Specified values of hand-drawn contour lines and data points were assigned to nearest model grid nodes. Unassigned nodes were thus linearly interpolated between assigned nodes.

The altitude of bedrock top for each model cell within the RASA study area was linearly interpolated using WANGRID.FOR and data from Westjohn and others (1994). The top of bedrock corresponds locally to the top of the Saginaw aquifer or the top of Jurassic "red beds." Between the outer boundaries of study area and the modeled area, land surface elevation contours from the Center for Remote Sensing's ERDAS database were linearly interpolated using computer program WANGRID.FOR to smooth the data. Glacial thickness contours from the Western Michigan University Hydrogeologic Atlas's Glacial Drift Thickness map (Western Michigan University, 1981) were downloaded from the ERDAS database and linearly interpolated using WANGRID.FOR. The drift thickness grid was subtracted from the land elevation grid to produce bedrock top elevations for the RASA model grid. Because of this technique, outside of the RASA study area, the resulting bedrock top map is highly correlated to the land elevation map. No alternative bedrock top map was available at the time of model construction for the region outside of the RASA study area. The resultant bedrock top map for the RASA study area was merged with the bedrock top map produced by subtracting drift thickness from land elevation in the ERDAS database to produce a combined bedrock top map for the Lower Peninsula (fig. 9). The combined bedrock top map represents a major angular

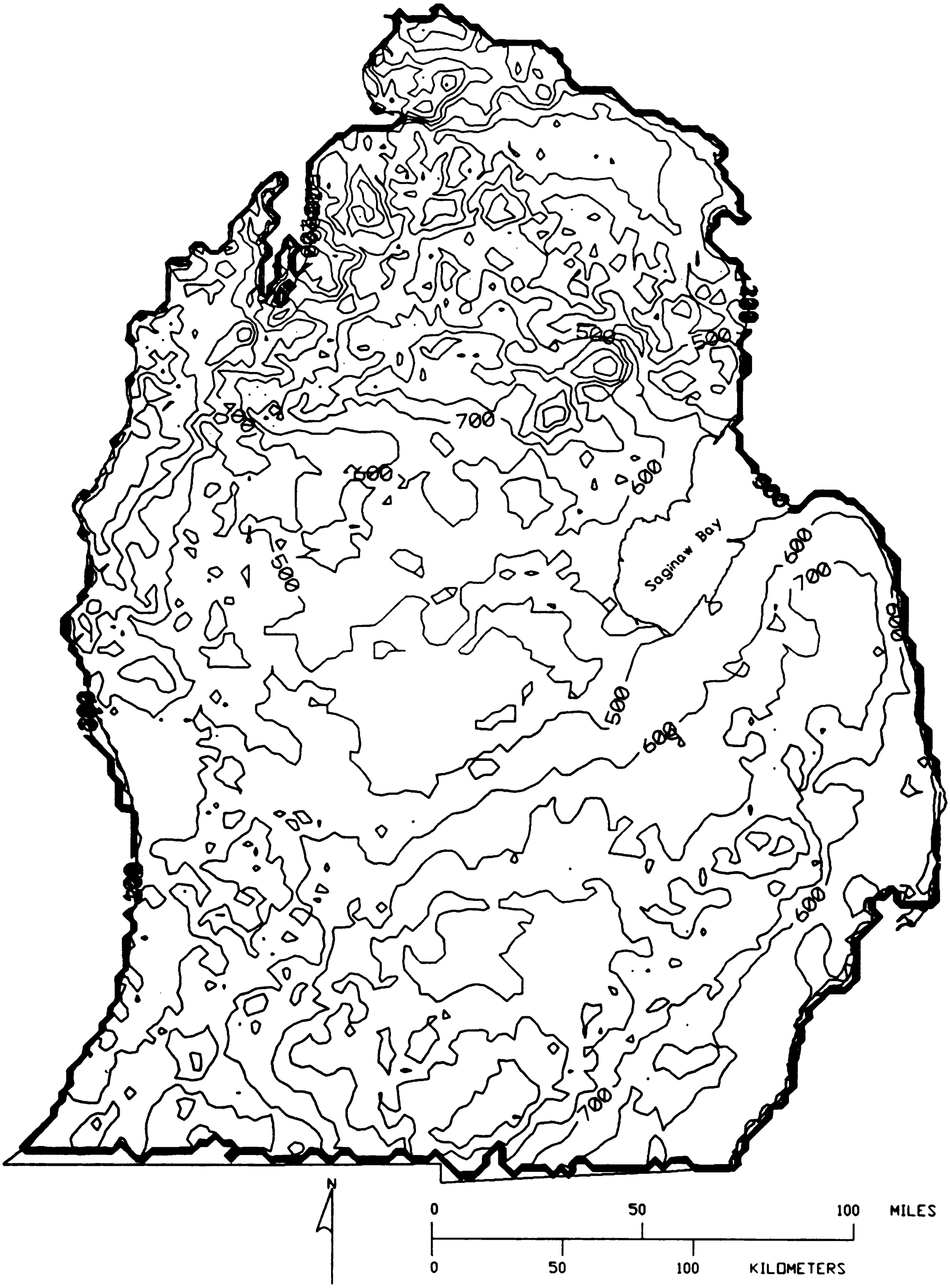

Figure 9.--Altitude of bedrock top interpolated from data provided by Westjohn and others (1994) and Western Michigan University (1981).

unconformity to the hydrogeologic flow system, the base of the glaciofluvial aquifer which is the base of model layer 1.

The top of the Saginaw aquifer forms the top of model layer 2 (fig 10, Appendix B). The base of model layer 2 was constructed by adding the Saginaw confining unit thickness to the top of the Parma-Bayport aquifer in the subregion of the Saginaw aquifer (fig. 11, Appendix B). Interpolations for these surfaces were done with WANGRID.FOR using data and contours from Westjohn and Weaver (1996a).

The top of the Parma-Bayport aquifer forms the top of layer 3 (fig 12, Appendix B). The top of the Michigan confining unit forms the base of the Parma-Bayport aquifer, which is the base of model layer 3. Interpolations for these surfaces were done with WANGRID.FOR (fig. 13, Appendix B).

The top of the Marshall aquifer was linearly interpolated in the subregion of the Michigan confining unit using WANGRID.FOR to construct the top of model layer 4 (fig. 14, Appendix B). The top of the Coldwater confining unit was linearly interpolated in the subregion of the Marshall aquifer (RASA study area) using WANGRID.FOR to construct the base of the Marshall aquifer, which is the base of model layer 4 (fig. 15, Appendix B). In the Marshall subcrop area, the bedrock surface was used as the top of the Marshall aquifer.

Digital Representation of model layers under Saginaw Bay

To complete the aquifer system for model implementation, the layers needed to be completed under Saginaw Bay where no data existed to construct the layers. This construction consisted of three steps. First, contacts for each aquifer and confining unit were drawn across the Bay. Then isopachs for each aquifer and confining unit were interpolated under the Bay using data for that part of the unit that does not underlie Saginaw Bay. Finally, the surface of each layer was constructed by subtracting the isopach from the bottom of the overlying unit starting with 530 ft above sea level, which was the assumed base of glacial deposits beneath the Bay. The reconstruction was therefore accomplished by linear interpolation and linear projection following a conservation of mass argument and an assumed base of drift angular unconformity at 530 ft in Saginaw Bay. An anticlinal structure on the Marshall aquifer resulted from the reconstruction. Though arguably a by-product of the method chosen, anticlinal structures under Saginaw Bay are also interpreted on the oil and gas chart of Cohee and others (1951), and Lane and Hubbard (1895, pl. 68 nos. 1 and 2).

Adjustments to interpolated layers

Since each top and bottom surface was contoured and interpolated independently, the hydrogeologic units were ill-defined in some regions where the bottom of a layer was defined higher than its top. In these regions, the aquifer was assumed to pinch out. A program was developed, TOPDOWN.FOR (Appendix A), to read in all of the model surfaces and redefine any bottom surface higher than a top to be 1 ft lower than the top. The program started from the topmost layer and worked through the model downward, thus propogating any consecutive errors downward. An example of the relationship among the surfaces of each model layer is illustrated in figure 16 which shows a cross section of the layer surfaces along a west to east transect of model row 240.

The Saginaw and Parma-Bayport aquifers, corresponding to model layers 2 and 3, pinch out interior to the Marshall aquifer subcrop. Therefore, model layer 1 must communicate directly with model layer 4 in the region between the Saginaw and Marshall subcrops. A program was developed, ANGUNCON.FOR (Appendix A), to define layers 2 and 3 to be 1-foot thick between layers 1 and 4 in this region. The 1-foot layers in the subcrop region were given the same hydraulic properties, both vertically and horizontally, as the overlying glacial deposits to simulate the direct hydraulic connection between model layers 1 and 4.

Boundary Conditions

Lateral boundaries for the model include specified heads and no-flow conditions for layer 1 and no-flow conditions for layers 2, 3, and 4 that coincide with the Marshall Sandstone/Coldwater Shale contact. MODFLOW boundary arrays were generated using the program FLOODFIL.FOR (Appendix A) which fills integers interior to closed polygons corresponding to the digitized geologic contacts.

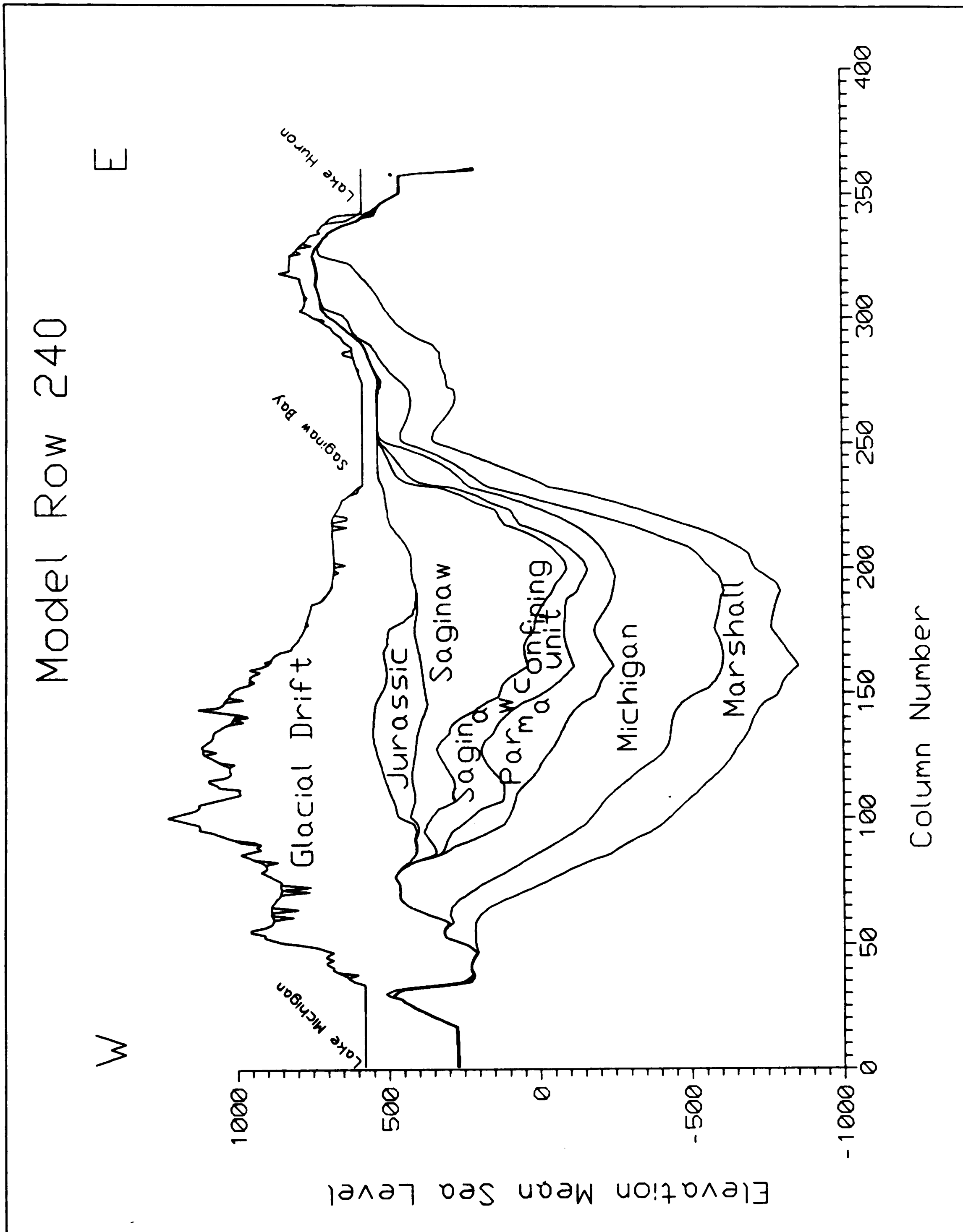

Figure 16.-- Cross section along model row 240 showing structural relations among aquifers and confining units.

Layer 1 is bounded by Lake Michigan to the west and northwest, Lake Huron to the Northeast and east, and Lake Erie to the southeast. The connecting channels of the St. Clair and Detroit Rivers and Lake St. Clair bound the eastern part of the area (fig. 17). The boundary was assigned constant heads of 580 ft for lakes Michigan and Huron, linearly interpolated heads between 580 and 575 ft for the St. Clair River, constant heads of 575 ft for Lake St. Clair, linearly interpolated heads between 575 and 572 ft for the Detroit River, and constant heads of 572 ft for Lake Erie. The southern boundary consists of drainage divides forming a continuous no-flow boundary. It is assumed that the surface drainage divides coincide with the ground-water divide in this area. Layers 2 and 3 are active interior to the Coldwater confining unit/Marshall aquifer contact.

In the Saginaw Bay area, the glaciofluvial aquifer is simulated as part of the specified-head boundary. A small portion of Lake Michigan that is underlain by the Marshall aquifer has the same boundary conditions as Saginaw Bay.

Stream and Lake Elevations

Internal boundaries that represent the major streams and natural lakes in the modeled area are simulated using the RIVER module of MODFLOW. River reaches in the model were identified from the Center for Remote Sensing's ERDAS data set (fig. 18). The data set contains pixelated drainages corresponding to the Michigan Hydrologic Unit

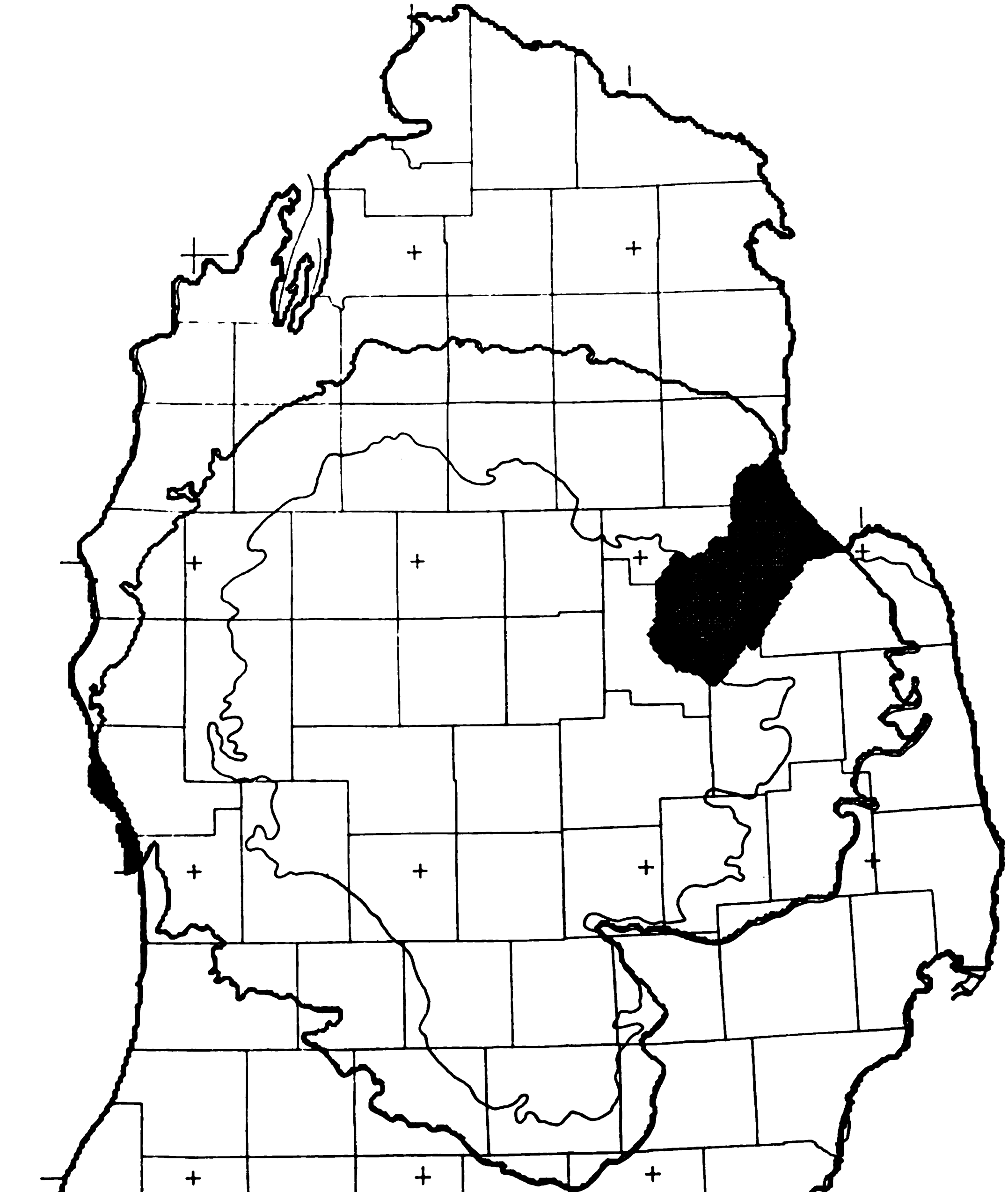

Figure 17.--Boundaries for layer 1 of the RASA ground-water flow model.

Figure 18.-- Locations of nodes used to simulate streams and lakes in the ground-water flow model.

map (U.S. Geological Survey, 1974). The river data set was modified by eliminating pixels at stream confluences to minimize the number of pixels used to define confluences. Some pixels were eliminated at stream headlands to separate stream reaches.

River stages were set by gridding the water table surface and taking the values of the grid at river reach locations. The water table was gridded by linearly interpolating the hand drawn water table contours (Mandle and Westjohn, 1989), specified heads at boundaries, and river and lake elevations using WANGRID.FOR (Appendix A). Data for the water table interpolation included 1,220 river crossings and 609 lake-level observations.

River widths were set using program CANOE.FOR (Appendix A) which orders the stream segments using Horton stream ordering (Horton, 1932) and puts the segments and reaches in downstream order. River stages were reset by linearly interpolating the stages from headland to confluence, confluence to confluence, and confluence to mouth. Uphill or flat segments were corrected to within 5 ft by reading the correct stages from 7.5 and 15 minute topographic maps and reprocessing the river file with CANOE.FOR. Up to five stream orders resulted from the processing. The Grand River, a fifth order stream, was followed along its length to determine widths corresponding to the different stream orders. The widths, obtained from discharge records at gaging stations at the starting confluence of each order along the Grand River, were used to set widths for each order of stream for the Lower Peninsula. The widths are 3.28, 6.56, 65.6, 197, and 262 ft for first through fifth order streams, respectively. It was assumed that all streambeds had a thickness of 1 ft and a vertical hydraulic conductivity of 0.28 ft/d.

Model Input Data

Hydraulic Characteristics

Hydraulic conductivities were summarized into model grids for the four-layer model. Each simulated layer used maps of percent coarse- and fine-grained material with a corresponding representative hydraulic conductivity estimate for each texture (table 2). Model vertical and horizontal hydraulic conductivities for the layers were calculated in series and in parallel using equations 1 and 2, respectively:

$$Kz = 100/((dc/Kvc) + (df/Kvf)) \qquad (1)$$

where

Kz = equivalent vertical hydraulic conductivity for the model layer (L/T),

dc = percent coarse-grained material in the layer (dim),

df = percent fine-grained material in the layer (dim),

Kvc = estimated vertical hydraulic conductivity for coarse-grained material (L/T)

Kvf= estimated vertical hydraulic conductivity for fine-grained material (L/T).

$$Kh = ((Khc * dc) + (Khf * df))/ 100 \qquad (2)$$

where

Kh = equivalent horizontal hydraulic conductivity for the model layer (L/T),

dc = percent sand or sandstone in the layer (dim),

df =percent fine-grained particles or rocks in layer (dim),

Khc=estimated horizontal hydraulic conductivity of coarse-grained material (L/T),

Khf=estimated horizontal hydraulic conductivity of fine-grained material(L/T).

The equations were modified from Freeze and Cherry (1979, p. 34) and were programmed into PRMPRCNT.FOR (Appendix A). The program thus calculates a different vertical and horizontal hydraulic conductivity at each node using the respective percent coarse- and fine-grained material as well as total thickness of the unit at each node. It is important to remember that the hydraulic characteristics used in the model for each layer represent values for the entire thickness of the layer and will generally be smaller than values determined for individual aquifers using other methods such as aquifer tests.

Horizontal hydraulic conductivities

Model layer 1.-- A map illustrating the percent coarse-grained material in the glaciofluvial aquifer was reproduced for model layer one using WANGRID.FOR. The percentages needed to be extended beyond the RASA study area boundary to cover the entire modeled area. Inside the study area, data from Westjohn and others (1994) was used. Geologic interpretation, based on major glacial ice-marginal positions, was used to estimate the percentage of coarse-grained material outside of the study area. The resulting map is shown in figure 19.

Once the percent of each material type was digitally completed for the modeled area, the coarse-grained portion was assigned a horizontal hydraulic conductivity of 50 ft/d and a vertical hydraulic conductivity of 5 ft/d. The fine-grained portion was assigned a hydraulic conductivity of 2.83 x 10-4 ft/d both horizontally and vertically. Model vertical and horizontal hydraulic conductivities were calculated in series and in parallel using equations 1 and 2 for vertical and horizontal hydraulic conductivities, respectively, as programmed in PRMPRCNT.FOR. The distribution of horizontal hydraulic conductivity for layer 1 is shown in figure 20.

<u>Model layer 2</u> .-- Vertical and horizontal hydraulic conductivity values for the Saginaw aquifer were calculated for each cell of model layer 2 using equations 1 and 2. The coarse-grained rock was assigned a hydraulic conductivity of 2.83 ft/d both horizontally and vertically. The fine-grained rock was assigned a horizontal hydraulic conductivity of 2.83×10^{-4} ft/d and a vertical hydraulic conductivity of 2.83×10^{-6} ft/d. Percent coarse-grained material for the aquifer was determined using maps that contour aquifer and Saginaw confining unit tops (Westjohn and Weaver, 1996a) to calculate the thickness of the aquifer at each cell. Thickness of the sandstone in the aquifer (Westjohn and Weaver, 1996a) was divided by total thickness and multiplied by 100 to obtain the percent sandstone for the aquifer at each cell. The distribution of horizontal hydraulic

Figure 19.-- Percent sand and gravel for glaciofluvial aquifer used to calculate hydraulic conductivity estimates for model layer 1.

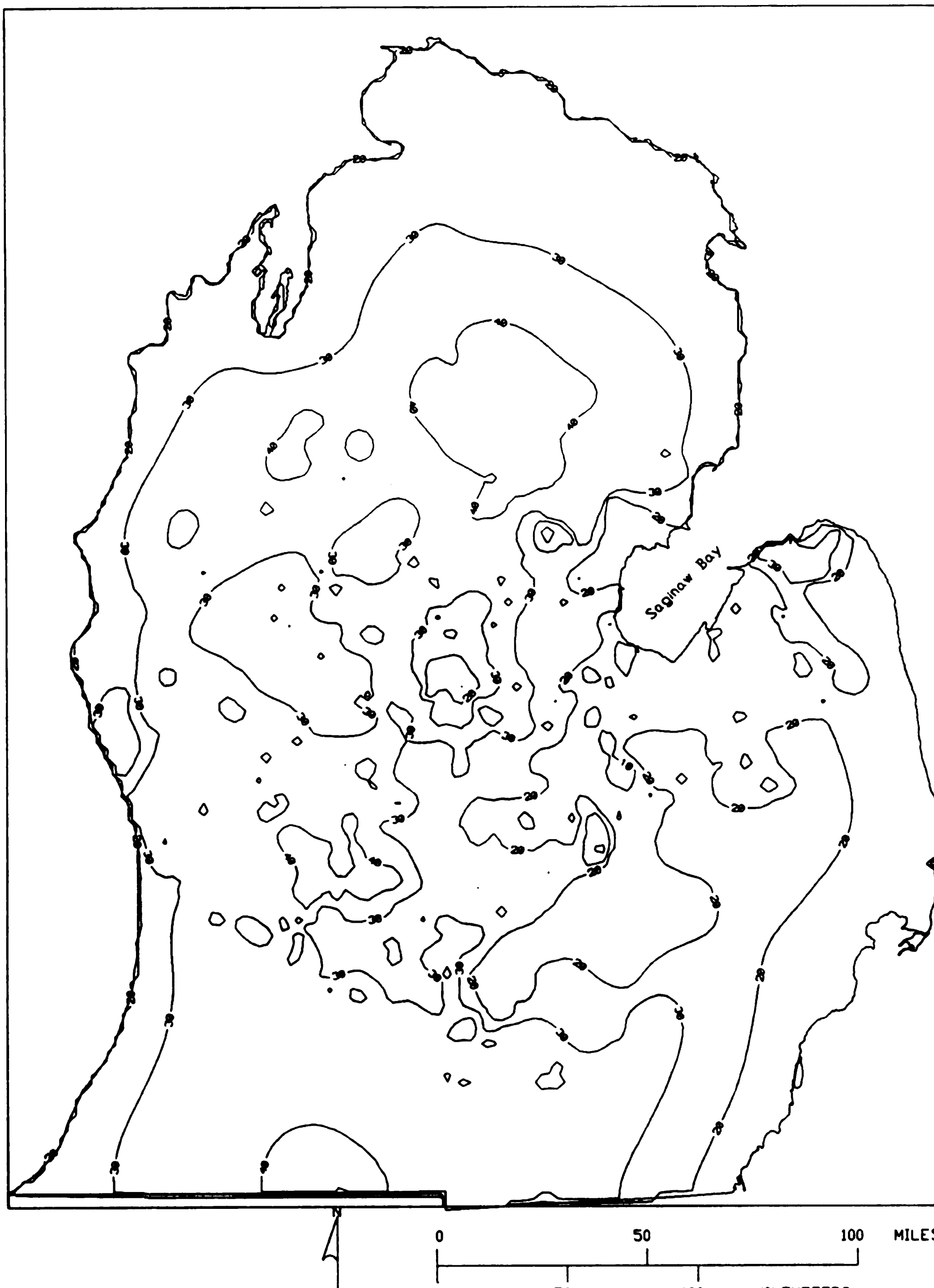

Figure 20.--Horizontal hydraulic conductivity (ft/day) estimates for the glaciofluvial aquifer, model layer 1.

conductivities is shown on figure 21.

Model Layer 3 .-- Percent sandstone within the Parma aquifer was not discerned as the aquifer is believed to be relatively homogeneous (D. Westjohn, personal comm., 1995). The horizontal hydraulic conductivity was assigned the value 7.09 ft/d. The vertical hydraulic conductivity for the whole aquifer was assigned the value 1.13 x 10-3 ft/d.

Model Layer 4.-- Vertical and horizontal hydraulic conductivity values for the Marshall aquifer were calculated for each cell of model layer 4 using equations 1 and 2. The coarse-grained rock was assigned a hydraulic conductivity of 1.42 x 10-1 ft/d both horizontally and vertically; fine-grained rock was assigned a horizontal hydraulic conductivity of 1.42 x 10-3 ft/d both horizontally and vertically. Percent coarse-grained rock for the aquifer was determined using maps that contour aquifer and Coldwater confining unit tops (Westjohn and Weaver, 1996b) to calculate the thickness of the aquifer at each cell. Thickness of the sandstone in the aquifer (Westjohn and Weaver, 1996b) was divided by total thickness and multiplied by 100 to obtain the percent coarse-grained rock for the aquifer at each cell. The distribution of horizontal hydraulic conductivities is shown on figure 22.

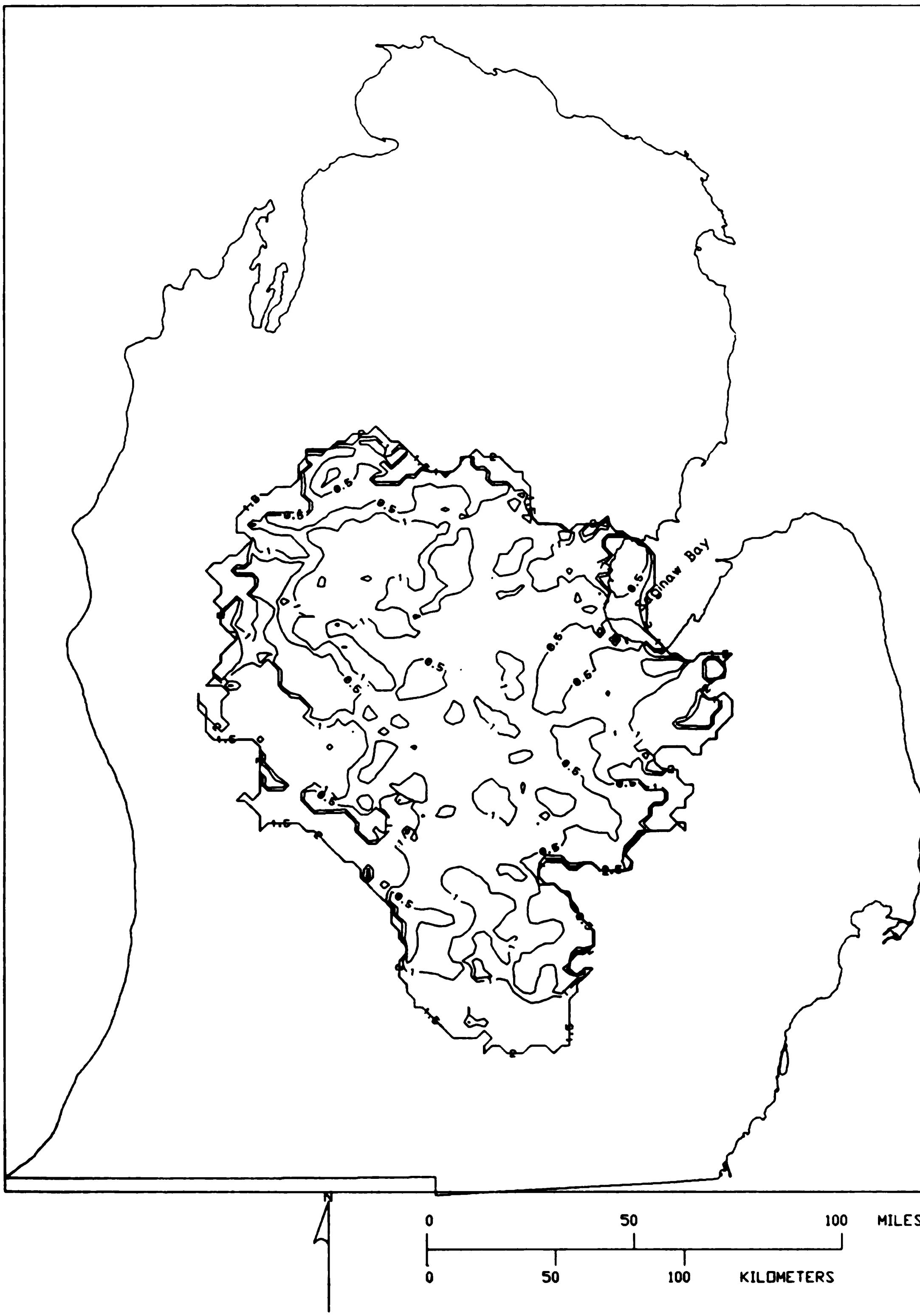

Figure 21.--Horizontal hydraulic conductivity (ft/day) estimates for the Saginaw aquifer, model layer 2.

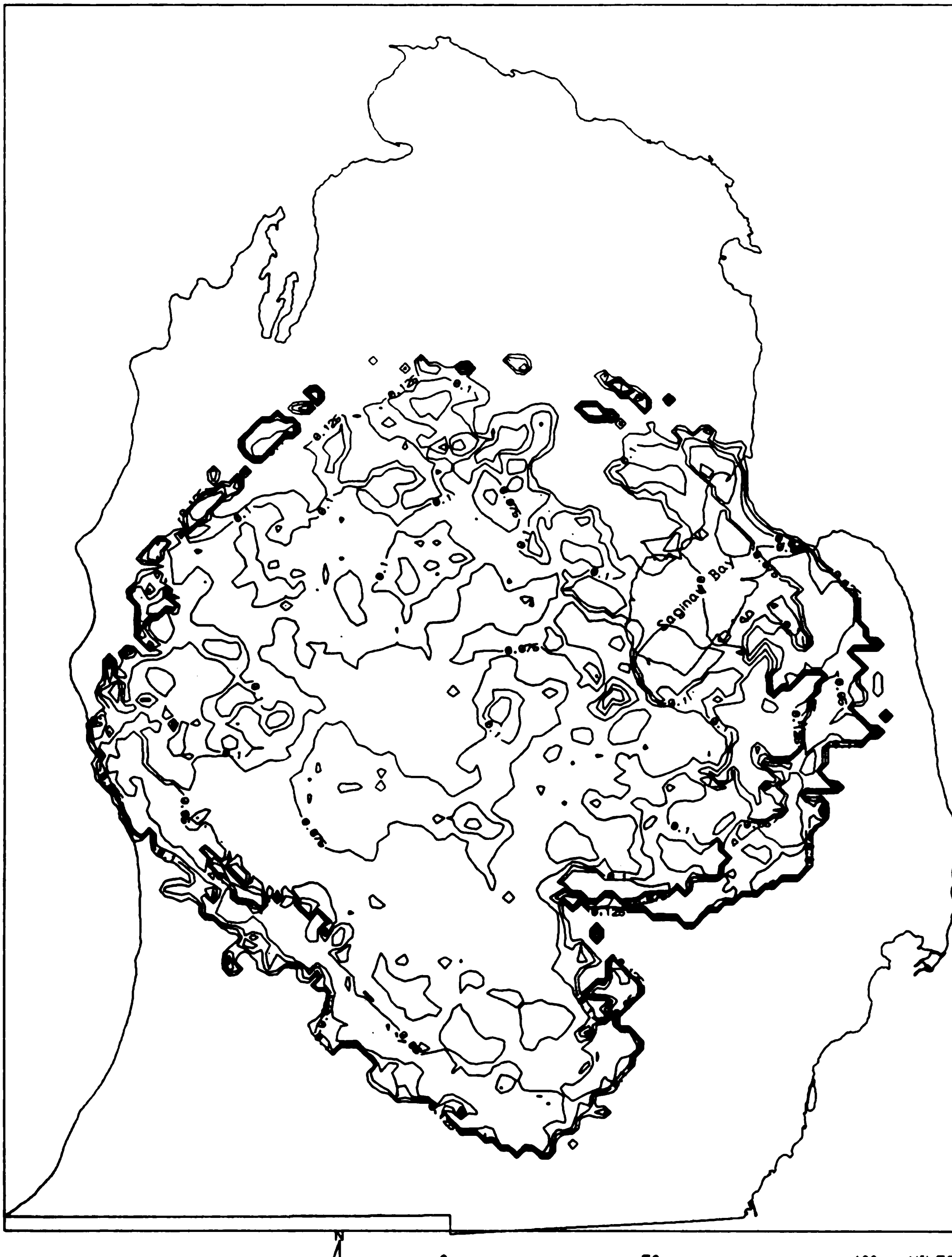

Figure 22.--Horizontal hydraulic conductivity (ft/day) estimates for the Marshall aquifer, model layer 4.

Vertical hydraulic properties

The vertical hydraulic conductivities for the aquifers and confining units were calculated using PRMPRCNT.FOR and used in program VCONT.FOR (Appendix A) to calculate vertical leakance between layers. This calculation requires surface configuration grids of the top and bottom of adjacent layers as well as vertical hydraulic conductivities of intervening confining units. The program VCONT.FOR calculates a leakance value for each active node by implementing equation 52 of the MODFLOW documentation (McDonald and Harbaugh, 1988, p. 5-16).

Leakance values for the confining unit that separates the glaciofluvial and Saginaw aquifers as calculated by VCONT.FOR used the top and bottom surface configurations of the glaciofluvial and Saginaw aquifers and their respective model vertical hydraulic conductivities, as well as the vertical hydraulic conductivity of the intervening Jurassic confining unit. The Jurassic confining unit was assigned a vertical hydraulic conductivity of 2.83×10^{-4} ft/d.

Leakance values for the confining unit that separates the Saginaw and Parma aquifers was calculated using the surface configurations of the Saginaw and the Parma aquifers and their respective model vertical hydraulic conductivities, as well as the vertical hydraulic conductivity of the intervening Saginaw confining unit. The Saginaw confining unit was assigned a vertical hydraulic conductivity of 2.83×10^{-4} ft/d.

Leakance values for the confining unit that separates the Parma and the Marshall aquifers were calculated using the surface configurations of the Parma and Marshall aquifers and their respective model vertical hydraulic conductivities, as well as the vertical hydraulic conductivity of the intervening Michigan confining unit. The Michigan confining unit is believed to comparmentalize the Michigan basin (D. Westjohn, personal communication, 1995) and was assigned a vertical hydraulic conductivity of 2.83×10^{-7} ft/d.

The horizontal and vertical hydraulic conductivities for the coarse and fine portions of each aquifer used in conjunction with the coarse and fine percentages to construct model conductivities and VCONTs, and the vertical hydraulic conductivities of the confining units used to construct VCONTS are summarized in table 2. The bedrock aquifer estimates were based on a study by Westjohn and others (1990) for the RASA project.

Ground-water Recharge

Ground-water recharge rates were estimated by an independent study for the RASA modeled area (Holtschlag, in press). The estimates were made by analysis of streamflow data from 114 basins throughout the study area. Ground-water runoff (base flow) was identified for each basin. This information was then statistically related to land characteristics in each basin and to prescipitation. An average basin recharge was determined for each basin. The local variation of recharge rates within gaged basins and

Table 2. Summary of horizontal and vertical hydraulic conductivities for the aquifers and confining units used to construct model conductivities and VCONTs.

	Texture	Horizontal Hydraulic Conductivity in feet per day	Vertical Hydraulic Conductivity in feet per day	Horizontal Hydraulic Conductivity in centimeters per second	Vertical Hydraulic Conductivity in centimeters per second
Glaciofluvial aquifer	Coarse grained	50	5	1.75×10^{-2}	1.75×10^{-3}
Fine-grained till or lacustrine deposits in Glaciofluvial aquifer	Fine grained	2.83×10^{-4}	2.83×10^{-4}	1×10^{-7}	1×10^{-7}
Glacial till-- J_R "red beds" confining unit	--	--	2.83×10^{-4}	--	1×10^{-7}
Saginaw aquifer	Coarse grained	2.83	2.83	1×10^{-3}	1×10^{-3}
Intercalated fine-grained rock in Saginaw aquifer	Fine grained	2.83×10^{-4}	2.83×10^{-6}	1×10^{-7}	1×10^{-9}
Saginaw confining unit	--	--	2.83×10^{-4}	--	1×10^{-7}
Parma-Bayport aquifer	--	7.09	1.13×10^{-3}	2.5×10^{-3}	4×10^{-7}
Michigan confining unit	--	--	2.83×10^{-7}	--	1×10^{-10}
Marshall aquifer	Coarse grained	1.42×10^{-1}	1.42×10^{-1}	5×10^{-5}	5×10^{-5}
Intercalated fine-grained rock in Marshall aquifer	Fine grained	1.42×10^{-3}	1.42×10^{-3}	5×10^{-7}	5×10^{-7}

ungaged areas was determined by identifying the statistical relation between average basin recharge rates and selected land characteristics. The relation was used to determine arecharge rate for each active node of model layer 1. The recharge rates vary from 0.19 to 22.3 in./yr and average 8.41 in./yr.

Model Calibration and Sensitivity Analysis

The model was calibrated by estimating parameters using ModflowP. The model was parameterized by identifying model inputs that were sensitive to trial and error calibration and least well defined. The conductivity of the streambed was parameterized by

$$COND = \frac{KLW}{M} = \beta_1 W$$

where β_1 = KL/M is the parameter initially set to 0.010763911 ft/s which corresponds to L = 1000 meters, M = 1 foot, and K = 10^{-4} cm/s. Two model inputs, the VCONT arrays between layers 1 and 2 and layers 3 and 4, were multiplied by scaling parameters, β_2 and β_3, initially set to 1.0.

ModflowP (Hill, 1992) uses either a sensitivity equation method or an adjoint state method to minimize a sum of squares objective function. The sensitivity equation method was chosen since the number of data points far exceeded the number of parameters estimated.

Theory of Sensitivity Equation methods

The actual value of a phenomena is related to the actual parameters by

$$\eta = X\beta$$

where $\boldsymbol{\eta}$ is the ND x 1 matrix of actual values, $\boldsymbol{\beta}$ is the NP x 1 matrix of actual parameters, and X is the ND x NP sensitivity matrix given by

$$X = \frac{\partial \eta}{\partial \beta}$$

The modelled value of the phenomena is related to the estimated parameters by

$$\hat{Y} = Xb$$

where $\hat{\mathbf{Y}}$ is the ND x 1 solution matrix, b is the NP x 1 matrix of estimated parameters, and X is the ND x NP sensitivity matrix approximated by

$$X = \frac{\partial \hat{Y}}{\partial b}$$

in the calibrated model.

Measured data, Y, is related to actual values through an error term and is related to modelled values through a residual term

$$Y=\eta+\epsilon=\hat{Y}+e$$

where ϵ is the ND x 1 matrix of errors and e is the ND x 1 matrix of residuals. In a calibrated model, the residual is in the same range as the error. The 1 x ND transpose matrix of the errors multiplied by the ND x 1 matrix of errors results in the scalar sum of squares function. Substituting the relation between actual values and measured data for the error yields the following sum of squares function

$$S=\epsilon^t\epsilon=(Y-X\beta)^t(Y-X\beta)$$

Minimization of the sum of squares function results in an ordinary least squares estimator. The sum of squares function is minimized by setting the matrix derivatives to zero at η = b. Solving that equation for b results in the ordinary least squares estimator (Beck and Arnold, 1977, pg. 235).

$$b_{OLS}=(X^tX)^{-1}X^tY$$

If weights are included in the estimation, the maximum likelihood estimator is of the form

$$b_{ML} = (X'WX)^{-1} X'WY$$

where W is the weight matrix (Beck and Arnold, 1977, pg. 259). The weight matrix is the inverse of the covariance matrix; weights are the reciprocal of the data variance. If prior information is included, the biased MAP estimator is of the form (Beck and Arnold, 1977, pg. 271)

$$b_{MAP} = \mu_\beta + P_{MAP} X'W(Y - X\mu_\beta)$$

$$P_{MAP}^{-1} = X'WX + V_\beta^{-1}$$

where μ_β and V_β are the expected value and variance of the prior information respectively.

If the model is non-linear in the parameters, iteration is required and multiple minima may be present. The Gauss-Newton method with prior information is (Beck and Arnold, 1977, pg. 341).

$$b^{(k+1)} = b^{(k)} + P^{(k)}[X^{t(k)} W(Y - \eta^{(k)}) + V_\beta^{-1}(\mu - b^{(k)})]$$

$$P^{-1(k)} = X^{t(k)} W X^{(k)} + V_\beta^{-1}$$

where k is the iteration level. ModflowP uses a modified form of the Gauss Newton method (Hill, 1992 pg. 76)

$$(C'X'WXC+I\mu)C^{-1}d_{(k)}=C'X'W(Y-\hat{Y})$$

with damping

$$b_{(k+1)}=\rho d_{(k)}+b_{(k)}$$

where C is a diagonal scaling matrix, I is the identity matrix, μ is the Marquardt parameter (Marquardt, 1963), and ρ is the damping parameter.

Weights

The data for heads and flow are weighted for the regression. In the sensitivity equation, the weight matrix is the inverse of the covariance matrix of the observation errors. For uncorrelated errors, the weights are just the inverses of the variances along the main diagonal. MODFLOWP allows weights to be entered as standard deviations, variances, or confidence percentages. For head measurements, confidence intervals were assumed to equal 4 standard deviations.

Heads in the glacial layer were assumed known to within 5 ft which resulted in a standard deviation of 2.5 and a weight of 0.16. Heads in the Saginaw and Marshall aquifers are assumed known to within 10 ft which resulted in a standard deviation of 5 and a weight percentage of 0.04.

For flow measurements, a confidence percentage of + or - 5 percent (10 percent) of the total estimated flow, was entered. MODFLOWP took the inverse of the 10 percent of estimated flow as the weight for each of the 44 flow measurements.

Model Output

Potentiometric Surfaces

Simulated regional flow is most affected by, and therefore most readily summarized by, the solution of hydraulic head for the glacial aquifer (layer 1) shown in figure 23. The output compares well with the regional distribution of head shown in figure 5 (Mandle and Westjohn, 1989) based on stream crossings and lake elevation data. Figure 24 shows a comparison of simulated head versus measured head for layer 1. The simulated head generally falls along the line of agreement from low to high head, but a slight trend exists where low heads are too high and high heads are too low. This may be related to the distribution of groundwater recharge (see Limitations of the Model below). The water table (hydraulic head by the Dupuit assumptions of the model) solution is highest in the northern and southern upland areas defined by the water table map of Mandle and Westjohn (1989).

The solution of hydraulic head for the glacial aquifer (layer 1; figure 23) is strongly imprinted on the solution of hydraulic head for the Saginaw (layer 2) aquifer shown in figure 25. The regions of high and low head are coincident with the glacial aquifer and head is only

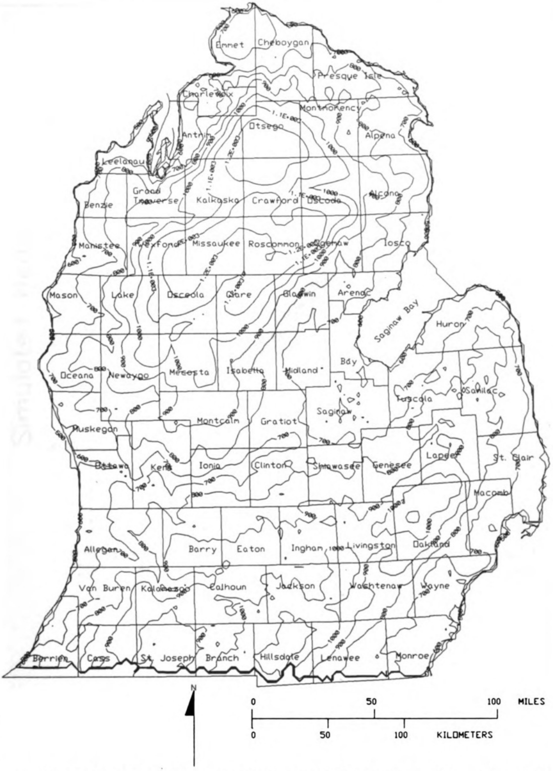

Figure 23. Simulated hydraulic head (Dupuit water table) for the glacial aquifer, model layer 1.

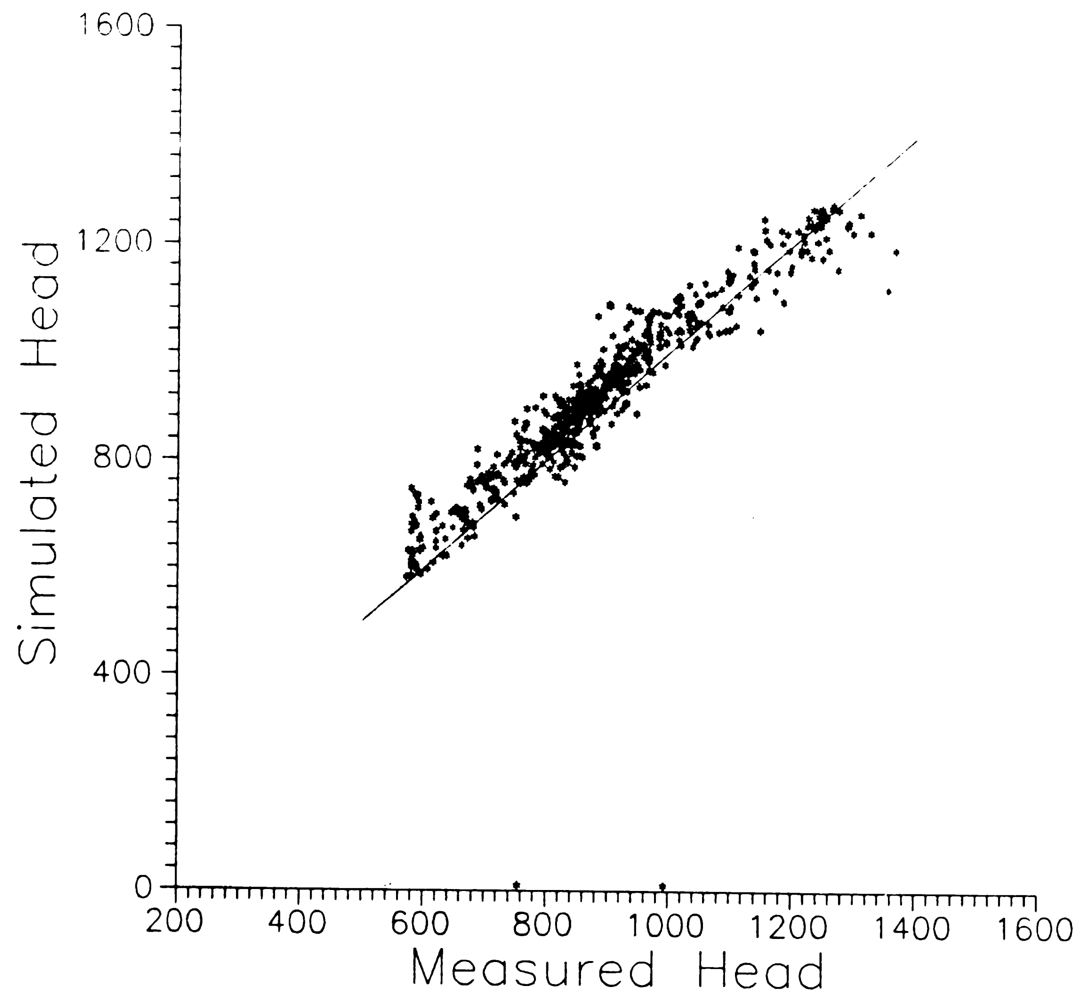

Figure 24. Simulated versus measured head for the glacial aquifer, model layer 1.

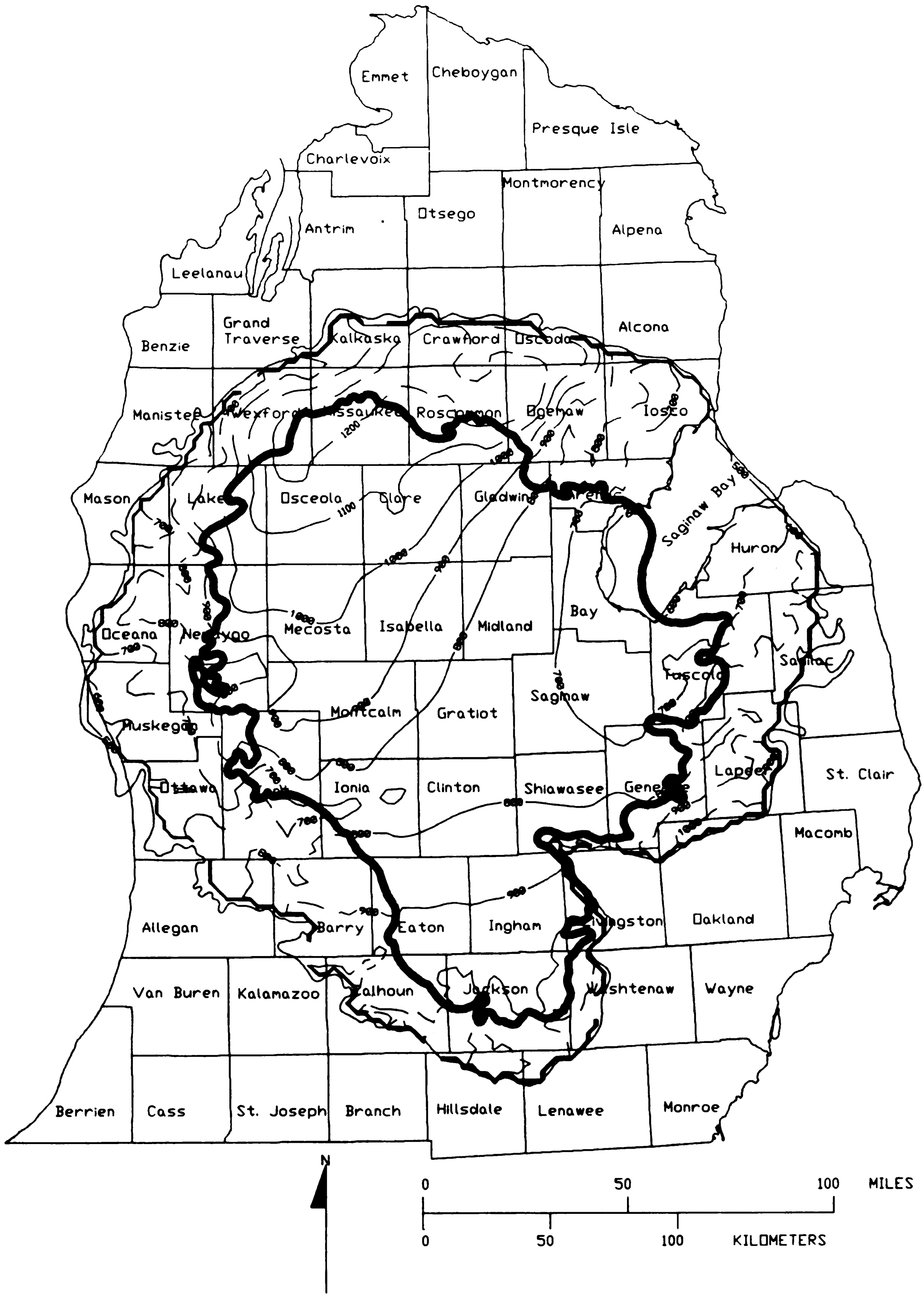

Figure 25. Simulated hydraulic head for the Saginaw aquifer, model layer 2.

moderately dampened (lowered) by the overlying glacial till--Jurassic "red beds" confining unit. The output compares well with the regional distribution of head shown in figure 6 (Barton and others, 1996) based on well data. Figure 26 shows a comparison of simulated head versus measured head for layer 2, the Saginaw aquifer. The simulated head generally falls along the line of agreement from low to high head.

The solution of hydraulic head for the Saginaw aquifer (layer 2; figure 25) is strongly imprinted on the solution of hydraulic head for the Parma (layer 3) aquifer shown in figure 27. The regions of high and low head are coincident with the Saginaw aquifer and head is only moderately dampened (lowered) by the overlying Saginaw confining unit. Data for calibration was not available for the Parma aquifer.

The solution of hydraulic head for the Marshall aquifer (layer 4) shown in figure 28 is considerably dampened, though the high and low areas are still controlled by the solution in the glacial aquifer (layer 1). The output compares well with the regional distribution of head shown in figure 7 (Barton and others, 1996) based on well data. Figure 29 shows a comparison of simulated versus measured head for layer 4, the Marshall aquifer. The simulated head generally falls along the line of agreement from low to high head. The dampening of the solution of hydraulic head in the Marshall aquifer (layer 4) is due to the low vertical hydraulic conductivity of the overlying Michigan confining unit (low VCONT between layers 3 and 4). The Michigan confining unit compartmentalizes the Michigan basin in the study area (Westjohn and Weaver, 1996b).

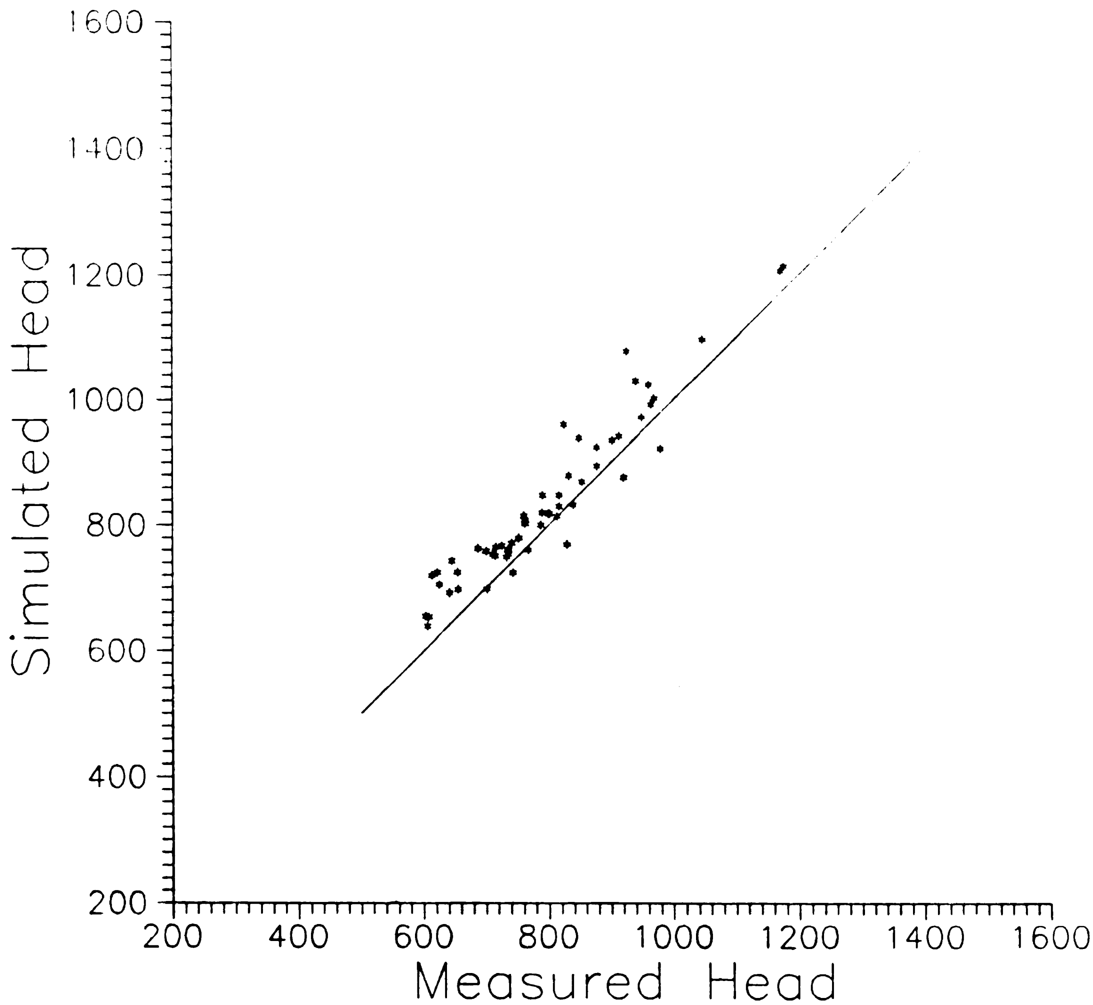

Figure 26. Simulated versus measured head for the Saginaw aquifer, model layer 2.

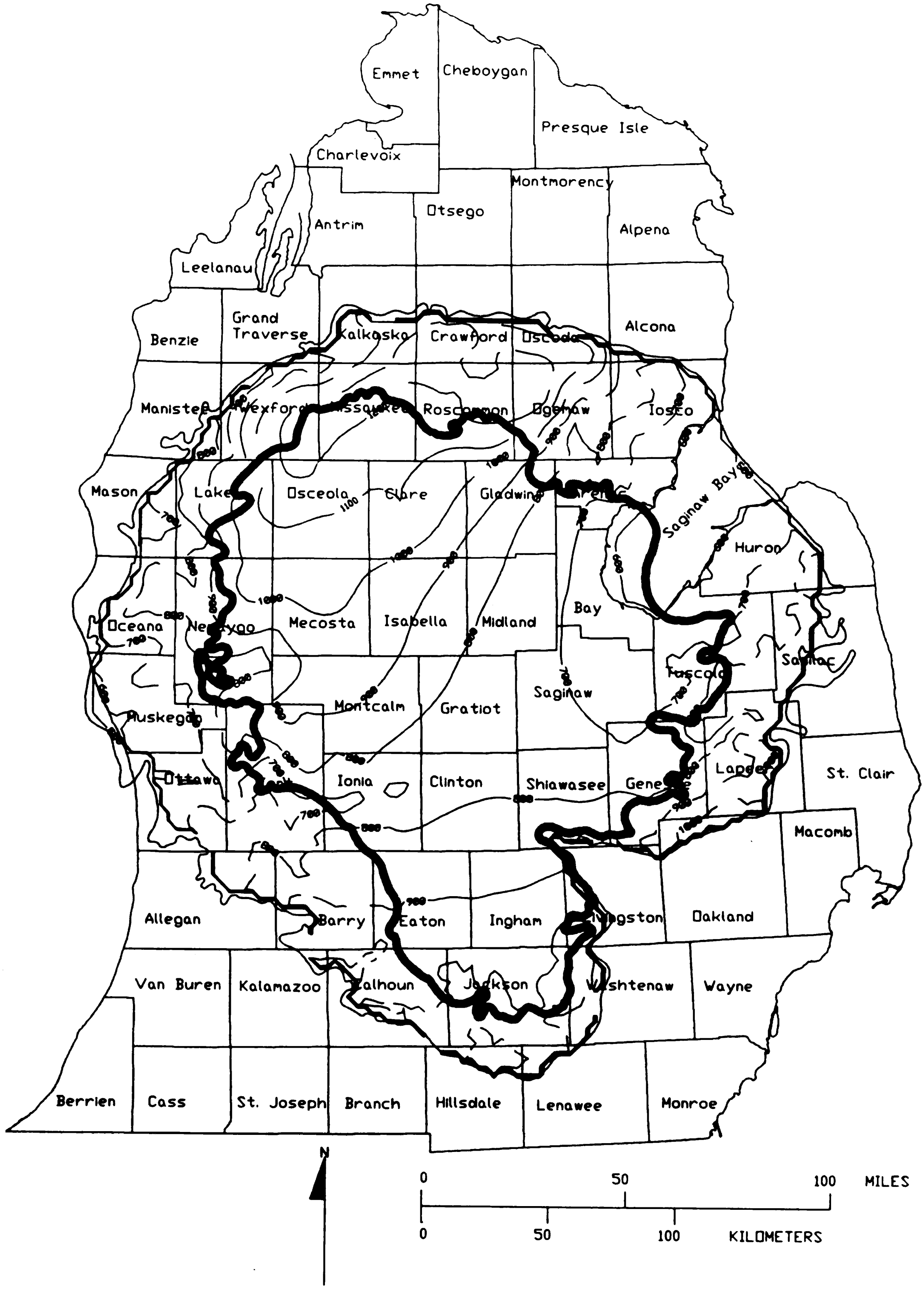

Figure 27. Simulated hydraulic head for the Parma-Bayport aquifer, model layer 3.

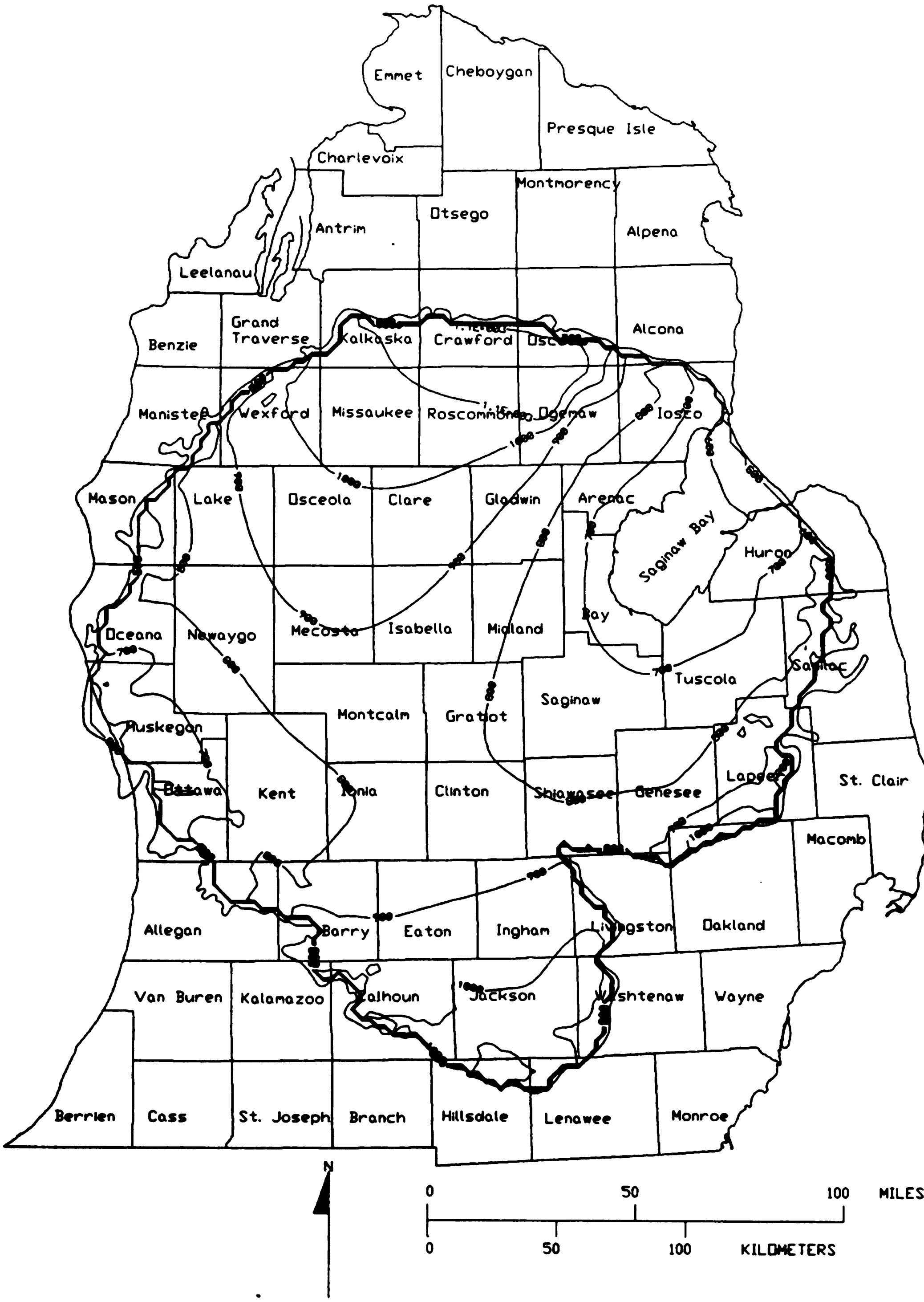

Figure 28. Simulated hydraulic head for the Marshall aquifer, model layer 4.

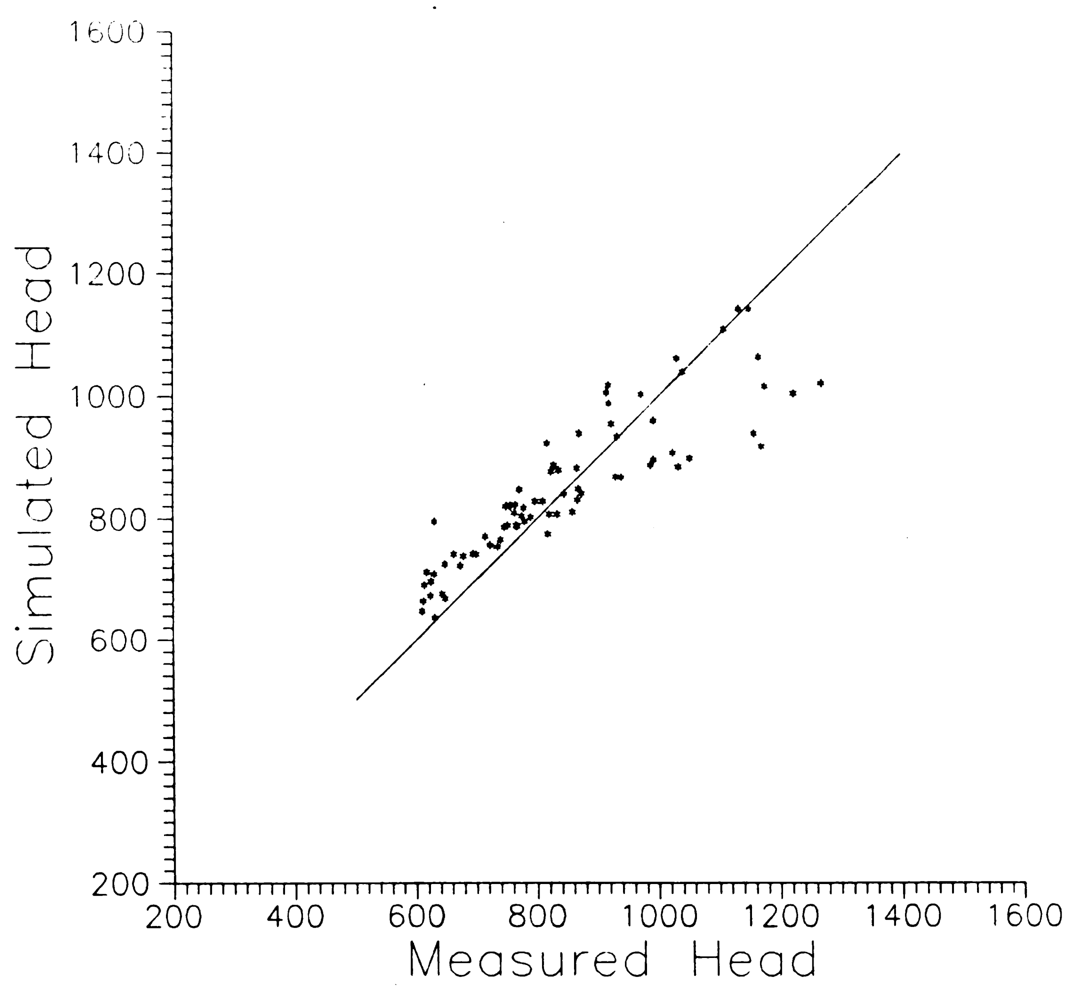

Figure 29. Simulated versus measured head for the Marshall aquifer, model layer 4.

Characteristics of Regional and Base Flow Based on Simulations

Regional groundwater flow is from the upland areas to the Great Lakes perpendicular to the contour lines. The overall pattern of the water table solution is affected by the internal river boundaries that display a dendritic pattern shown in figure 18. The rivers divide the flow system into localized flow cells. As a result, most groundwater flow discharges to the rivers as opposed to the Great Lakes (see Regional Groundwater Budget). Preliminary regional particle tracking confirms that localized flow cells exist, drastically reducing the groundwater residence time within layer 1 (Hoaglund, 1996). Groundwater flow is regionally downward into the bedrock aquifers except in the Saginaw Lowlands where flow is upward. Bedrock conductivities are considerably lower resulting in regional groundwater flow paths of long residence time. However, the volume of water incorporated into these flow paths is considerably lower than in the glacial aquifer. The glacial aquifer transmits the most water, though with the considerably shorter residence times. The model was also calibrated against stream discharge measurements. The data plots along a line of agreement on a graph (figure 30) of simulated versus measured stream discharge for river basins ranging from less than 10 cubic feet per second (cfs) up to 1000 cfs.

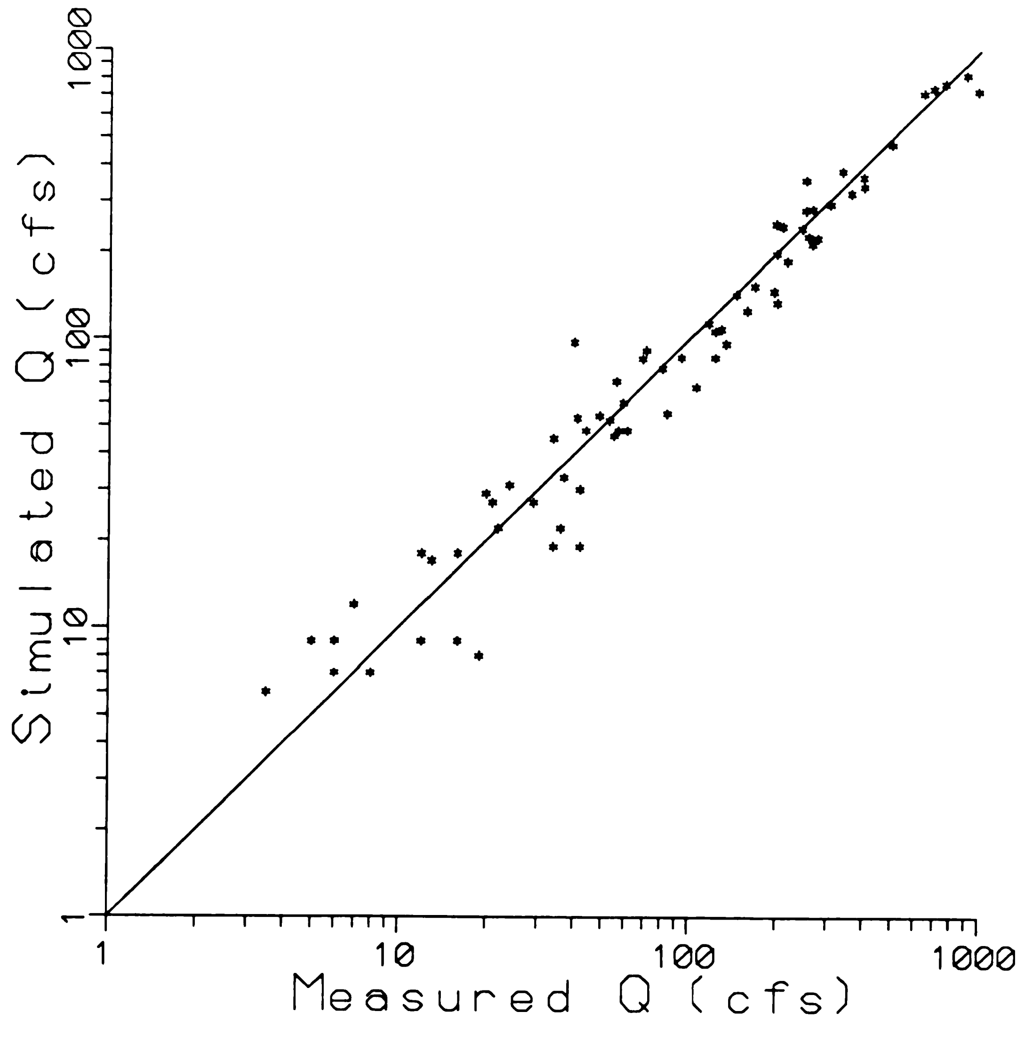

Figure 30. Simulated versus measured stream discharge (cfs).

Regional Groundwater Budget

The volumetric budget for the entire model is shown in Table 3. Groundwater inflow is almost entirely derived from the groundwater recharge at a rate of 25,124 cfs. A relatively small rate of 49 cfs is derived from flow from rivers. Groundwater outflow is divided between flow to rivers (23,902 cfs or approximately 95%) and flow to the Great Lakes constant heads (1272 cfs or approximately 5%). Groundwater inflow and outflow balances within -0.01 percent. The low groundwater outflow to constant heads relative of flow to rivers confirms that localized flow cells exist within the glacial aquifer (layer 1) which direct flow to rivers.

Ground-Water Discharge to the Great Lakes

Though only 5% of the model volumetric outflow budget, direct groundwater flow to the Great Lakes is of great concern to Great Lakes management and study. Figure 31 shows direct groundwater discharge (cfs) from the glacial aquifer (layer 1) calculated for each constant head cell, plotted against the shoreline length measured from southwest Michigan clockwise to southeast Michigan. Twenty-eight (28) coastal cities are tagged to the plot from a map of Michigan. The calculated discharge is variable cell to cell, lower in bays and near rivers where flow either parallels the coastline or is diverted to rivers. The calculated discharge is lowest in the Saginaw lowlands due to low model conductivity and thickness (low

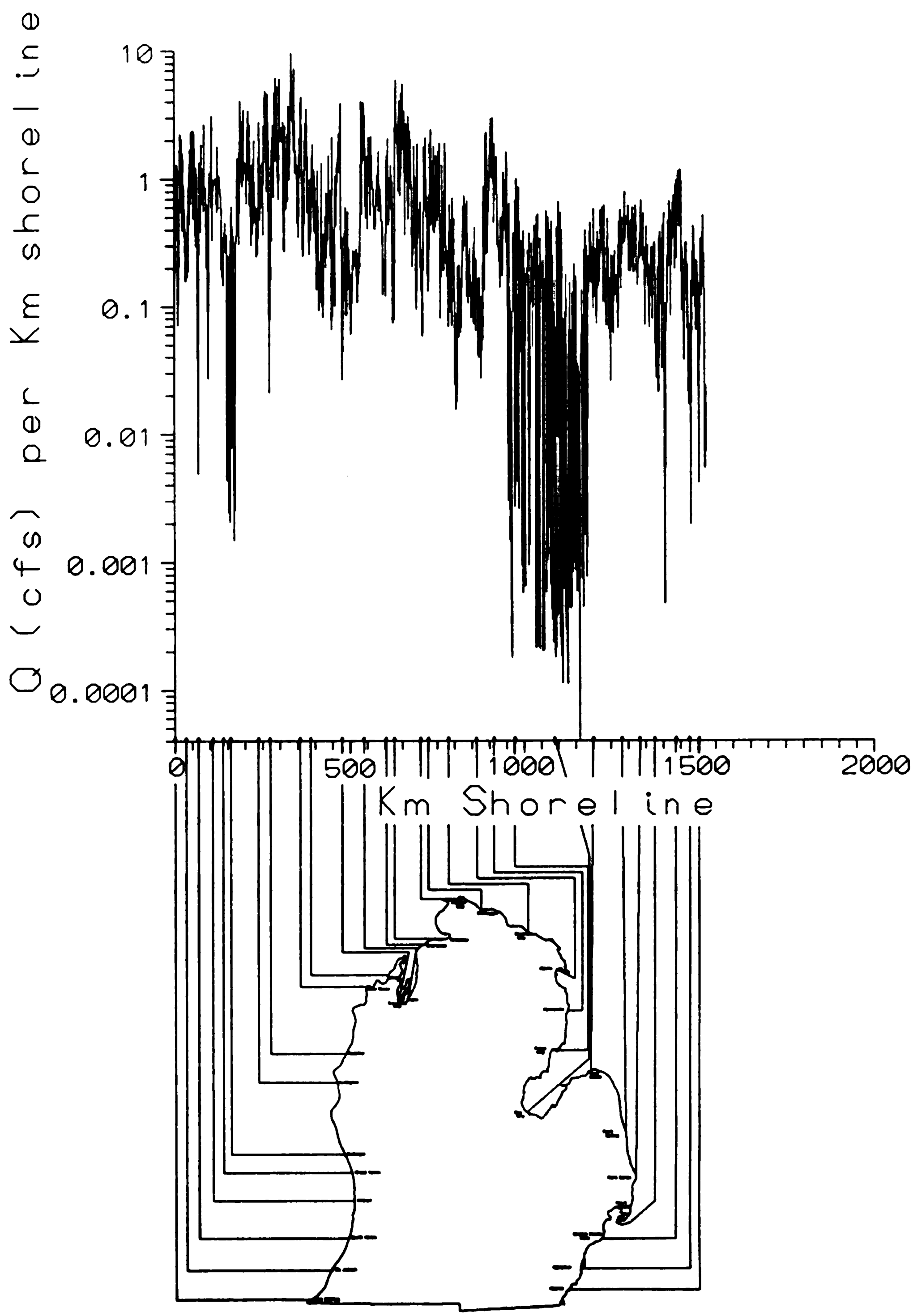

Figure 31. Direct groundwater discharge (cfs) to the Great Lakes from the glacial aquifer per kilometer of shoreline versus kilometer of shoreline.

transmissivity) and lower gradients for the region. The highest discharge is calculated for the shoreline between Manistee and Glen Haven due to high model conductivity and thickness (high transmissivity) and steep gradients from the northern uplands.

Effects of Variable-Density Fluids on Ground-Water Flow

Limitations of the Model

Darcy Flow Conditions

The MODFLOW model solves a governing equation derived from Darcy's law and continuity. Furthermore, the Michigan RASA model was constructed assuming heterogeneous and horizontally isotropic flow conditions. The model would therefore not be valid in karst regions or other non-darcy flow conditions, nor in anisotropic flow conditions such as a fracture dominated flow condition.

Recharge

The recharge rates used in the model were extrapolated from selected stream base flow measurements related to basin characteristics and precipitation. This method assumes that all recharge water of the selected basins discharges at the stream as baseflow without losses to, or gains from, deep seepage. The extrapolated recharge rates of the model may be similarly underestimated or overestimated. Therefore the deep seepage component of flow in the model may not be properly accounted for in the model budget. Generally, model recharge would be underestimated in the regional recharge areas due to unaccounted for deep seepage losses, and overestimated in regional discharge areas due to unaccounted for deep seepage gains.

Steady State Assumption

The Michigan RASA model assumes steady state which implies that the Michigan Basin has reached equilibrium. Head and flow measurements used to calibrate the model were assumed to be time invariant. The flow budget for the model, including recharge, was also assumed to be time invariant. Since the model is a pre-development simulation, the effects of municipal withdrawal were not modeled.

The variable density solution technique of Kontis and Mandle (1988) requires a steady state simulation. Though this is consistent with the Michigan RASA model, any transient simulation using the model will require a transient variable density solution technique.

Scale

All model parameters and boundary conditions are scale dependent. As a result, interpretations from the model should be limited to the scale of the investigation.

Stream nodes

Approximately 10% of layer 1 nodes are stream boundary nodes forming a dendritic pattern over the entire solution area. As a result, boundary influences must be considered when evaluating the heads or the effect of stresses on heads in layer 1. For example, data on the elevations of stream crossings were eliminated from the head calibration data set since this same data was used to set the stage boundaries in the river package.

Variable density

SUMMARY

REFERENCES

Baltusis, M.A., Quigley, M.S., and Mandle, R.J., 1992, Municipal ground-water development and withdrawals in the central Lower Peninsula of Michigan, 1870-1987: U.S. Geological Survey Open-File Report 91-215, 89 p.

Barton, G.C., Mandle, R.J., and Baltusis, M.A., 1996, Predevelopment freshwater heads in the glaciofluvial, Saginaw, and Marshall aquifers in the Michigan Basin: U.S. Geological Survey Open-File Report 95-319, 24 p.

Beck, J.V. and K.J. Arnold, 1977, Parameter Estimation in Science and Engineering: Wiley and Sons, New York, 501 p.

Cohee, G.V., Macha, C., and Holk, M., 1951, Thickness and lithology of Upper Devonian and Carboniferous rocks in Michigan: U.S. Geological Survey, Oil and Gas Investigations, Preliminary Chart OC-41

Eschman, D.J., 1985, Summary of the Quaternary history of Michigan, Ohio, and Indiana: Journal of Geological Education, 1985, v. 33, pp. 161-167

Fenneman, N.M., 1938, Physiography of Eastern United States, McGraw-Hill, 692 p.

Freeze, R.A., and Cherry, J.A., 1979, Groundwater: Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 604 p.

Gebert, W.A., Graczyk, D.I., and Krug, W.R., 1985, Average annual runoff in the United States, 1951-80: U.S. Geological Survey Open-File Report 85-627, scale 1:2,000,000.

Ging, P.B., Long, D.T., and Lee, R.W., 1996, Selected geochemical characteristics of ground water from the Marshall aquifer, central Lower Peninsula of Michigan: U.S. Geological Survey, Water-Resources Investigations Report 94-4220, 21 p.

Grannemann, N.G., and Twenter, F.R., 1985, Geohydrology and ground-water flow at Verona well field, Battle Creek, Michigan: U.S. Geological Survey Water-Resources Investigations Report 85-4056, 54 p.

Hill, M.C., 1992, A computer program (MODFLOWP) for estimating parameters of a transient, three-dimensional, ground-water flow model using non-linear regression: U.S. Geological Survey, Open-File Report 91-489, 358 p.

Hoaglund, J.R., 1996, Recharge to Discharge Groundwater Travel times in the Michigan Basin and the Effect of Glacial Ice Loading: Unpublished Doctoral Thesis, Michigan State University, 274 p.

Holtschlag, D.J., (in press), A generalized estimate of ground-water-recharge rates in the Lower Peninsula of Michigan: U.S. Geological Survey Water-Supply Paper 2437, 81 p.

Holtschlag, D.J., Luukkonen, C.L., 1996, Simulation of ground-water flow in the Saginaw aquifer, Clinton, Eaton, and Ingham Counties, Michigan: U.S. Geological Survey, Water-Resources Investigations Report 96-174, 49 p.

Horton, R.E., 1945, Erosional development of streams and their drainage basins, hydrophysical approach to quantitative morphology: Geological Society of America Bulletin, v. 56, no. 3, p. 275-370.

Kontis, A., and Mandle, R.J., 1988, Modifications to the three-dimensional ground-water-flow model to account for variable-density and multiaquifer effects: U.S. Geological SurveyWater-Resources Investigations Report 87-4265, 78 p.

Lane, A.C. and Hubbard, L.L., 1895, The geology of Lower Michigan 1885-1893 with reference to deep borings: Geological Survey of Michigan, Volume 5, Part II, ____ p.

Leverett, F., and Taylor, F.B., 1915, The Pleistocene of Indiana and Michigan and history of the Great Lakes: U.S. Geological Survey Monograph 53, 529 p.

Lillienthal, R.T., 1978, Stratigraphic cross-sections of the Michigan Basin: Michigan Department of Natural Resources, Geological Survey Division, Report of Investigation 19, 35 p.

Long, D.T., Meissner, B.D., and Wahrer, M.A., in press, Geochemical evolution of groundwater in the glaciofluvial, Saginaw, and Marshall regional aquifers, Central Lower Peninsula of Michigan: Solute-solute diagrams: Water Resources Investigations Report.

Lusch, D.P. and W. Enslin, 1984. "Microcomputer based statewide digital land surface information," in Proceedings of the Pecora IX Symposium on Spatial Information Technology for Remote Sensing, Today and Tomorrow, Sioux Falls, S. Dakota, p. 40-43.

Lynch, E.A., and Grannemann, N.G., 1996, Assessment of geohydrology and ground-water flow at Verona well field, Battle Creek, Michigan: U.S. Geological Survey Water-Resources Investigations Report ______, __ p.

Mandle, R.J., 1986, Plan of study for the regional aquifer system analysis of the Michgan Basin: U.S. Geological Survey Open-File Report 86-494, 23 p.

Mandle, R.J., and Westjohn, D.B., 1989, Geohydrologic framework and ground-water flow in the Michigan basin: American Water Resources Association Monograph Series 13, p. 83-110.

Marquardt, D.W. , 1963, An algorithm for least squares estimation of nonlinear parameters: Journal of the Society of Industrial and Applied Mathematics, v. 11, no. 2, p. 431-441.

McDonald, M.G., and Fleck, W.B., 1978, Model analysis of the impact on ground-water conditions of the Muskegon County wastewater disposal system, Michigan: U.S. Geological Survey Open-File Report 78-99, 63 p.

McDonald, M.G., and Harbaugh, A.W., 1988, A modular three-dimensional finite-difference ground-water-flow model: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 6, Chapt. A1.

Meissner, B.D., Long, D.T., and Lee, R.W., 1996, Selected geochemical characteristics of water from the Saginaw aquifer in the Michigan Basin: U.S. Geological Survey Water-Resources Investigations Report 93-4220, 29 p.

Press, W.H., S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, 1992, Numerical Recipes in Fortran. The art of scientific computing, 2nd Ed.: Cambridge University Press, New York, 963 p.

Stark, J.R., Cummings, T.R., and Twenter, F.R., 1983, Ground-water contamination at Wurtsmith Air Force Base, Michigan: U.S. Geological Survey Water-Resources Investigations Report 83-4002, 93 p.

Sun, R.J., and Weeks, J.B., 1991, Bibliography of Regional Aquifer-System Analysis program of the U.S. Geological Survey, 1978-91: U.S. Geological Survey Water-Resources Investigations Report 91-4122, 92 p.

U.S. Geological Survey, 1974, Hydrogeologic Unit Map--1974, State of Michigan, 2 pl., scale 1:500,000.

Vanlier, K.E., and Wheeler, M.L., 1968, Analog simulation of ground-water development of the Saginaw Formation, Lansing Metropolitan area, Michigan: Tri-County Regional Planning Commission, 40 p.

Vugrinovich, R.G., 1986, Patterns of regional subsurface fluid movement in the Michigan Basin: Michigan Geological Survey, Open File Report 86-6, 27 p.

Wahrer, M.A., Long, D.T., and Lee, R.W., 1996, Selected geochemical characteristics of ground water from the glaciofluvial aquifer in the central Lower Peninsula of Michigan: U.S. Geological Survey Water-Resources Investigations Report 94-4017, 18 p.

Wang, H.F., and Anderson, M.P., 1982, Introduction to groundwater modeling: W.H. Freeman and Company, San Francisco, 237 p.

Western Michigan University, Department of Geology, College of Arts and Sciences, 1981, Hydrogeologic Atlas of Michigan: U.S. Environmental Protection Agency Underground Injection Control Program Report, 35 pl., scale 1:500,000.

Westjohn, D.B., Olsen, H.W., and Willden, A.T., 1990, Matrix-controlled hydraulic properties of Mississippian and Pennsylvanian Sandstones from the Michigan Basin: U.S. Geological Survey Water-Resources Investigations Report 93-4152, 14 p.

Westjohn, D.B., and Weaver, T.L., 1996a, Hydrogeology of Pennsylvanian and Late Mississippian rocks in the central Lower Peninsula of Michigan: U.S. Geological Survey, Water-Resources Investigations Report 94-4107, 44 p.

Westjohn, D.B., and Weaver, T.L., 1996b, Hydrogeologic framework of Mississippian rocks in the central Lower Peninsula of Michigan: U.S. Geological Survey, Water-Resources Investigations Report 94-4246, 46 p.

Westjohn, D.B., and Weaver, T.L., 1996c, Configuration of freshwater/saline-water interface and geologic controls on distribution of freshwater in a regional aquifer system, central Lower Peninsula of Michigan: U.S. Geological Survey Water-Resources Investigations Report 94-4242, 44 p.

Westjohn, D.B., and Weaver, T.L., 1996, Hydrogeologic framework of the Michigan Basin Regional Aquifer System: Central Lower Peninsula of Michigan: U.S. Geological Survey Professional Paper 1418-A, 44 p.

Westjohn, D.B., Weaver, T.L., and Zacharias, K.F., 1994, Hydrogeology of Pleistocene glacial deposits and Jurassic "red beds" in the central Lower Peninsula of Michgan: U.S. Geological Survey Water Resources Investigation Report 93-4152, 14 p.

APPENDIX A

APPENDIX A--1. Program TM2GEO.FOR

```
C REPRODUCTION OF THE INVERSE PROCESSING (TM TO GEODETIC) PART
C OF THE CRSTM CODE OF BRIAN BUCKLEY (CRS)
C FROM U.S.G.S. BULL. 1532 BY SNYDER AS
C DELIVERED BY DAVE LUSCH (CRS)
C EARTH STUFF VARIABLE DECLARATIONS
      DOUBLE PRECISION PI,X0,Y0,LONG0,LAT0,K0,M0,AR,ESQ,EPSQ
C VARIABLES OF SOLUTION
      DOUBLE PRECISION LAT,LONG,X,Y
C RADIAN EQUIVALENTS
      DOUBLE PRECISION PHI0,PHI1
C INVERSE COEFFICIENTS
      DOUBLE PRECISION E1,N1,T1,C1,R1,D,M,MU
      CHARACTER*40 FILIN
C OPEN INPUT FILES
      WRITE (*,*) '***********************************'
      WRITE (*,*) '*                                 *'
      WRITE (*,*) '*           TM2GEO                *'
      WRITE (*,*) '*             by                  *'
      WRITE (*,*) '*        JOHN HOAGLUND            *'
      WRITE (*,*) '*                                 *'
      WRITE (*,*) '***********************************'
      WRITE (*,*) 'ENTER THE CRSTM X,Y,Z FILENAME: '
      READ (*,'(A)') FILIN
      OPEN (11,FILE=FILIN,STATUS='UNKNOWN')
      OPEN (12,FILE='GEO.OUT',STATUS='UNKNOWN')
C BEGIN PROCESSING
      PI=3.1415926535897932384626433832795028841971693993751l
C SCALE ON CENTRAL MERIDIAN (0.9996 FOR UTM)
      K0=0.9996
C EQUATORIAL RADIUS OR SEMIMAJOR AXIS OF THE CLARK (1866) ELLIPSOID
      AR=6378206.4
C POLAR RADIUS OR SEMIMINOR AXIS OF THE CLARK (1866) ELLIPSOID
C       BR=   UNKNOWN, CAN BE CALCULATED FROM AR AND ESQ
C ECCENTRICITY OF THE CLARK (1866) ELLIPSOID SQUARED CALCULATED
C       ESQ=((1-(BR*BR)/(AR*AR)))
C ECCENTRICITY OF THE CLARK (1866) ELLIPSOID GIVEN PAGE 232
      ESQ=.00676866
      E=SQRT(ESQ)
C ORIGIN OF CRSTM AS GIVEN BY DAVE LUSCH
      X0=359987
      Y0=344917
      LONG0=-86.000
      LAT0=44.000
C FOR A GIVEN X,Y CALCULATE LATITUDE AND LONGITUDE
C LONGITUDE IS ASSUMED TO BE GIVEN WEST OF GREENWICH
```

```
C THEREFORE USE NEGATIVE AS PER PG XII
 2000 READ(11,*,END=2001) X,Y,Z
      X=X-X0
      Y=Y-Y0
C FORWARD PROCESSING COEFFICIENTS
      PHI0=LAT0*PI/180
C EQUATION 3-21 FOR M0
C       M0=AR*((1-ESQ/4-3*(ESQ**2)/64-5*(ESQ**3)/256)*LAT0-
C      $(3*ESQ/8+3*(ESQ**2)/32+45*(ESQ**3)/1024)*SIN(2*PHI0)+
C      $(15*(ESQ**2)/256+45*(ESQ**3)/1024)*SIN(4*PHI0)-
C      $(35*(ESQ**3)/3072)*SIN(6*PHI0))
C USE SIMPLIFIED FORMULA 3-22 FOR M0 OF CLARK ELLIPSOID
      M0=111132.0894*LAT0-16216.94*SIN(2*PHI0)+17.21*SIN(4*PHI0)-
     $0.02*SIN(6*PHI0)
C EQNS 8-12 AND 8-15
      EPSQ=ESQ/(1-ESQ)
C  USE 8-20 FOR M
      M = M0 + Y/K0
C EQN 3-24 SOLVED FOR E1
      E1=(1-SQRT(1-ESQ))/(1+SQRT(1-ESQ))
C EQN 8-19
      MU=M/(AR*(1-ESQ/4-3*ESQ*ESQ/64-5*ESQ**3/256))
C EQN 3-26
      PHI1=MU + (3*E1/2-27*(E1**3)/32)*SIN(2*MU) + (21*(E1**2)/16-
     $55*(E1**4)/32)*SIN(4*MU) + (151*(E1**3)/96)*SIN(6*MU)
C NOW EQNS 8-21 TO 8-25
      N1=AR/SQRT(1-ESQ*(SIN(PHI1)*SIN(PHI1)))
      T1=TAN(PHI1)*TAN(PHI1)
      C1=EPSQ*COS(PHI1)*COS(PHI1)
      R1=AR*(1-ESQ)/SQRT((1-ESQ*SIN(PHI1)*SIN(PHI1))**3)
      D=X/(N1*K0)
C CALCULATE LAT AND LONG USING EQNS 8-17 AND 8-18
      LAT=( PHI1-(N1*TAN(PHI1)/R1)*(D*D/2-(5+3*T1+10*C1-4*C1*C1
     $-9*EPSQ)*(D**4)/24+(61+90*T1+298*C1+45*T1*T1-252*EPSQ-3*C1*C1)
     $*(D**6)/720) )*(180/PI)
      LONG=LONG0+( (D-(1+2*T1+C1)*(D**3)/6+(5-2*C1+28*T1-3*C1*C1+
     $8*EPSQ+24*T1*T1)*D**5/120)/COS(PHI1) )*(180/PI)
      LONG=-LONG
      WRITE (12,*) LAT,LONG,Z
      GOTO 2000
 2001 CONTINUE
      STOP
      END
```

APPENDIX A--2. Program GEO2TM.FOR

```
C AN ATTEMPT TO REPRODUCE THE CRSTM CODE OF BRIAN BUCKLEY (CRS)
C FROM U.S.G.S. BULL. 1532 BY SNYDER AS
C DELIVERED BY DAVE LUSCH (CRS)
C EARTH STUFF VARIABLE DECLARATIONS
```

```
      DOUBLE PRECISION PI,X0,Y0,LONG0,LAT0,K0,M0,AR,BR,ESQ,EPSQ
C VARIABLES OF SOLUTION
      DOUBLE PRECISION LAT,LONG,X,Y
C RADIAN EQUIVALENTS
      DOUBLE PRECISION PHI0,LAMBDA0,PHI,LAMBDA
C FORWARD COEFFICIENTS
      DOUBLE PRECISION N,T,C,A,M
      PI=3.14159265358979323846264338327950288419716939937511
C SCALE ON CENTRAL MERIDIAN (0.9996 FOR UTM)
      K0=0.9996
C EQUATORIAL RADIUS OR SEMIMAJOR AXIS OF THE CLARK (1866) ELLIPSOID
      AR=6378206.4
C POLAR RADIUS OR SEMIMINOR AXIS OF THE CLARK (1866) ELLIPSOID
C      BR=   UNKNOWN, CAN BE CALCULATED FROM AR AND ESQ
C ECCENTRICITY OF THE CLARK (1866) ELLIPSOID SQUARED CALCULATED
C      ESQ=((1-(BR*BR)/(AR*AR)))
C ECCENTRICITY OF THE CLARK (1866) ELLIPSOID GIVEN PAGE 232
      ESQ=.00676866
C ORIGIN OF CRSTM AS GIVEN BY DAVE LUSCH
      X0=359987
      Y0=344917
      LONG0=-86.000
      LAT0=44.000
C FOR A GIVEN LATITUDE AND LONGITUDE CALCULATE X,Y
C LONGITUDE IS ASSUMED TO BE GIVEN WEST OF GREENWICH
C THEREFORE USE NEGATIVE AS PER PG XII
      READ(*,*) LAT,LONG
      LONG = - LONG
      PHI=LAT*PI/180
      LAMBDA=LONG*PI/180
C FORWARD PROCESSING COEFFICIENTS
      PHI0=LAT0*PI/180
      LAMBDA0=LONG0*PI/180
C EQUATION 3-21 FOR M0
C      M0=AR*((1-ESQ/4-3*(ESQ**2)/64-5*(ESQ**3)/256)*LAT0-
C     $(3*ESQ/8+3*(ESQ**2)/32+45*(ESQ**3)/1024)*SIN(2*PHI0)+
C     $(15*(ESQ**2)/256+45*(ESQ**3)/1024)*SIN(4*PHI0)-
C     $(35*(ESQ**3)/3072)*SIN(6*PHI0))
C USE SIMPLIFIED FORMULA 3-22 FOR M0 OF CLARK ELLIPSOID
      M0=111132.0894*LAT0-16216.94*SIN(2*PHI0)+17.21*SIN(4*PHI0)-
     $0.02*SIN(6*PHI0)
C EQNS 8-12 AND 8-15
      EPSQ=ESQ/(1-ESQ)
      write (*,*) epsq
      N=AR/SQRT(1-ESQ*(SIN(PHI)*SIN(PHI)))
      write (*,*) n
      T=TAN(PHI)*TAN(PHI)
      write (*,*) t
      C=EPSQ*COS(PHI)*COS(PHI)
      write (*,*) c
```

```
      A=COS(PHI)*(LONG-LONG0)*(PI/180)
      write (*,*) a
C EQUATION 3-21 FOR M
C      M=AR*((1-ESQ/4-3*(ESQ**2)/64-5*(ESQ**3)/256)*LAT-
C     $(3*ESQ/8+3*(ESQ**2)/32+45*(ESQ**3)/1024)*SIN(2*PHI)+
C     $(15*(ESQ**2)/256+45*(ESQ**3)/1024)*SIN(4*PHI)-
C     $(35*(ESQ**3)/3072)*SIN(6*PHI))
C USE SIMPLIFIED FORMULA 3-22 FOR M OF CLARK ELLIPSOID
      M=111132.0894*LAT-16216.94*SIN(2*PHI)+17.21*SIN(4*PHI)-
     $0.02*SIN(6*PHI)
C CALCULATE X AND Y WITH EQNS 8-9 AND 8-10
      X=K0*N*(A+(1-T+C)*(A**3)/6+(5-18*T+T**2+72*C-58*EPSQ)*(A**5)/120)
      Y=K0*(M-M0+N*TAN(PHI)*((A**2)/2+(5-T+9*C+4*C**2)*(A**4)/24+
     $(61-58*T+T**2+600*C-330*EPSQ)*(A**6)/720))
      X=X+X0
      Y=Y+Y0
      WRITE(*,*) X,Y
      STOP
      END
```

APPENDIX A–3. Program WANGRID.FOR

```
      REAL X,Y,Z,ZM(361,470),CONVRG
      REAL MINX,MAXX,MINY,MAXY,MINZ,MAXZ,DELX,DELY
      INTEGER GRDX,GRDY,COUNT,ITMAX
      INTEGER IBOUND(361,470)
      CHARACTER*40 FILIN
      CHARACTER*12 OUTFIL,     GRDFIL
      CHARACTER    OUTNAM(12), GRDNAM(12)
      CHARACTER*8  FILOUT, SHITOUT
      CHARACTER*4  OUT, GRD, BINAS
      EQUIVALENCE (OUTNAM(1),FILOUT),(OUTNAM(9),OUT)
      EQUIVALENCE (OUTFIL,OUTNAM(1))
      EQUIVALENCE (GRDNAM(1),SHITOUT),(GRDNAM(9),GRD)
      EQUIVALENCE (GRDFIL,GRDNAM(1))
C INITIALIZE COUNTER AND DOMAIN AND FILE VARIABLES
      COUNT = 0
      OUT = '.WNG'
      GRD = '.GRD'
      DELX=1000
      DELY=1000
      GRDX=361
      GRDY=470
C CENTROID MINIMUM X IS 290568 AT 1,1 AND 1,470
C CENTROID MAXIMUM Y IS 557128 AT 1,1
      MINX=290568
      MAXY=557128
      MAXX = MINX + (GRDX-1)*DELX
      MINY = MAXY - (GRDY-1)*DELY
C BINARY OR ASCII VARIABLE IS DSAA FOR ASCII OR DSBB FOR BINARY
```

```
      BINAS='DSAA'
      WRITE (*,*) ' *****************************'
      WRITE (*,*) ' *          WANGRID          *'
      WRITE (*,*) ' *             by            *'
      WRITE (*,*) ' *       John Hoaglund       *'
      WRITE (*,*) ' *****************************'
      WRITE (*,*) 'ENTER THE NAME OF THE IBOUND INPUT FILE: '
      READ(*,'(A)') FILIN
      OPEN (11,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) 'ENTER THE NAME OF THE CRSTM X,Y,Z INPUT DATA FILE'
      READ(*,'(A)') FILIN
      OPEN (10,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) 'ENTER THE FIRST 8 CHARACTERS OF THE OUTPUT FILENAME: '
      READ(*,'(A8)') FILOUT
      SHITOUT = FILOUT
      WRITE(*,'(A)') OUTFIL
      WRITE(*,'(A)') GRDFIL
C OPEN FILES
      OPEN (12,FILE=OUTFIL,STATUS='UNKNOWN')
      OPEN (17,FILE=GRDFIL,STATUS='UNKNOWN')
C PROMPT FOR ITERATIONS MAXIMUM AND CONVERGENCE
      WRITE(*,*) 'ENTER MAXIMUM ITERATIONS:   '
      READ(*,*) ITMAX
      WRITE(*,*) 'ENTER CONVERGENCE:  '
      READ(*,*) CONVRG
C  INITIALIZE ALL ZM(I,J) VALUES TO BE 1000.
       DO 5 J=1,GRDY
       DO 5 I=1,GRDX
         ZM(I,J) = 1000.
5      CONTINUE
C READ IBOUND FILE FOR IBOUND ARRAY VALUES
      DO 110 J = 1,GRDY
        READ(11,'(25I3)') (IBOUND(I,J),I=1,GRDX)
 110  CONTINUE
C READ A CRSTM XYZ DATA FILE, STORE ZM(I,J)=Z AND IBOUND(I,J)=-1
 1000 READ (10,*,END=34) X,Y,Z
      COUNT = COUNT + 1
      IF ((COUNT.EQ.1).OR.(Z.LT.MINZ)) MINZ = Z
      IF ((COUNT.EQ.1).OR.(Z.GT.MAXZ)) MAXZ = Z
C CALCULATE NEAREST I J ADDRESS
      I = NINT((X-MINX)/DELX) + 1
      J = NINT((MAXY-Y)/DELY) + 1
C SKIP IF OUT OF RANGE
      IF ((I.GT.GRDX).OR.(I.LT.1)) GOTO 1000
      IF ((J.GT.GRDY).OR.(J.LT.1)) GOTO 1000
C ASSIGN Z TO GRID
      ZM(I,J) = Z
C ENTER -1 IN IBOUND BOOLEAN ARRAY
      IBOUND(I,J) = -1
      GOTO 1000
```

```
C SOLVE LAPLACE EQUATION WITH GAUSS-SEIDEL ITERATION
C IN ACTIVE (IBOUND=1), UNFILLED
C GRID NODES USING MODIFIED WANG AND ANDERSON, 1982
C REGIONAL FLOW SYSTEM EXAMPLE, FIGURE 2-10
 34    CONTINUE
C  KEEP TRACK OF NUMBER OF ITERATIONS AND OF LARGEST ERROR
       NUMIT = 0
C MAIN ITERATION LOOP
 35    AMAX = 0.
       NUMIT = NUMIT + 1
C  SWEEP INTERIOR POINTS WITH 5-POINT OPERATOR
       DO 40 J=1,GRDY
       DO 40 I=1,GRDX
C SKIP ANY BOUNDARY CONDITION POINTS
         IF (IBOUND(I,J).LT.1) GOTO 40
         OLDVAL = ZM(I,J)
C IF BLOCK TO ASSIGN FICTITIOUS NODES TO ANY 0 (INACTIVE) NEXT TO 1 (ACTIVE)
C THIS IS THE SAME AS SPECIFYING GRADIENT OF LAPLACE TO ZERO
         IF (IBOUND(I-1,J).EQ.0) ZM(I-1,J)=ZM(I,J)
         IF (IBOUND(I+1,J).EQ.0) ZM(I+1,J)=ZM(I,J)
         IF (IBOUND(I,J-1).EQ.0) ZM(I,J-1)=ZM(I,J)
         IF (IBOUND(I,J+1).EQ.0) ZM(I,J+1)=ZM(I,J)
         ZM(I,J) = (ZM(I-1,J) + ZM(I+1,J) + ZM(I,J-1) + ZM(I,J+1))/4.
         ERR = ABS(ZM(I,J) - OLDVAL)
         IF(ERR.GT.AMAX) AMAX=ERR
 40    CONTINUE
C DO ANOTHER ITERATION IF LARGEST ERROR GREATER THAN CONVRG
C AND MAXIMUM ITERATIONS ARE NOT EXCEEDED
       IF(NUMIT.GT.ITMAX) GOTO 1010
       IF(AMAX.GT.CONVRG) GO TO 35
C WANG AND ANDERSON SOLUTION METHOD IS DONE.
C WRITE THE MATRIX
 1010 DO 2000 J = 1,GRDY
       DO 1500 K = 1,GRDX
C FILTER OUT UNUSED GRID VALUES, ESTABLISH MAX AND MIN
        IF (IBOUND(K,J).EQ.0) ZM(K,J) = 0.0
        IF (ZM(K,J).GT.MAXZ) MAXZ=ZM(K,J)
        IF (ZM(K,J).LT.MINZ) MINZ=ZM(K,J)
 1500  CONTINUE
       WRITE(12,2500) (ZM(I,J), I = 1,GRDX)
 2000 CONTINUE
 2500 FORMAT(10F8.1)
C CONVERT THE MATRIX TO A SURFER GRID
 2001 WRITE(17,'(A)') BINAS
      WRITE(17,*) GRDX,GRDY
      WRITE(17,*) MINX,MAXX
      WRITE(17,*) MINY,MAXY
      WRITE(17,*) MINZ,MAXZ
C NOTE SURFER GRIDS ARE STORED WITH INCREASING Y FROM TOP DOWN
      DO 2010 J = GRDY,1,-1
```

```
           WRITE(17,2500) (ZM(I,J), I = 1,GRDX)
 2010 CONTINUE
      WRITE(*,*) 'ITERATIONS COMPLETED ',NUMIT
C CLOSE FILES AND QUIT
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(17)
      STOP
      END
```

APPENDIX A–4. Program TOPDOWN.FOR

```
C     PROGRAM TOPDOWN.FOR BY JOHN HOAGLUND
      REAL TOP(361,470,4), BOT(361,470,4)
      INTEGER IBOUND(361,470)
      CHARACTER*40 FILIN
      NROW=470
      NCOL=361
      NLAY=4
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE TOP OF LAYER 1:'
      READ(*,'(A)') FILIN
      OPEN(11,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(21,FILE=FILIN//'.MOD',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE BOT OF LAYER 1: '
      READ(*,'(A)') FILIN
      OPEN(12,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(22,FILE=FILIN//'.MOD',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE TOP OF LAYER 2:'
      READ(*,'(A)') FILIN
      OPEN(13,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(23,FILE=FILIN//'.MOD',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE BOT OF LAYER 2: '
      READ(*,'(A)') FILIN
      OPEN(14,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(24,FILE=FILIN//'.MOD',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE TOP OF LAYER 3:'
      READ(*,'(A)') FILIN
      OPEN(15,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(25,FILE=FILIN//'.MOD',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE BOT OF LAYER 3: '
      READ(*,'(A)') FILIN
      OPEN(16,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(26,FILE=FILIN//'.MOD',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE TOP OF LAYER 4:'
      READ(*,'(A)') FILIN
      OPEN(17,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(27,FILE=FILIN//'.MOD',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE BOT OF LAYER 4: '
```

```
      READ(*,'(A)') FILIN
      OPEN(18,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(28,FILE=FILIN//'.MOD',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE OPERATIONAL BOUNDARY IBOUND ARRAY: '
      READ(*,'(A)') FILIN
      OPEN(20,FILE=FILIN,STATUS='UNKNOWN')
C BEGIN READING
      DO 100 J = 1,NROW
        READ(11,'(10F8.1)') (TOP(I,J,1), I = 1,NCOL)
        READ(12,'(10F8.1)') (BOT(I,J,1), I = 1,NCOL)
        READ(13,'(10F8.1)') (TOP(I,J,2), I = 1,NCOL)
        READ(14,'(10F8.1)') (BOT(I,J,2), I = 1,NCOL)
        READ(15,'(10F8.1)') (TOP(I,J,3), I = 1,NCOL)
        READ(16,'(10F8.1)') (BOT(I,J,3), I = 1,NCOL)
        READ(17,'(10F8.1)') (TOP(I,J,4), I = 1,NCOL)
        READ(18,'(10F8.1)') (BOT(I,J,4), I = 1,NCOL)
        READ(20,'(25I3)') (IBOUND(I,J), I = 1,NCOL)
 100  CONTINUE
      DO 300 J = 1,NROW
         DO 200 I = 1, NCOL
           IF (IBOUND(I,J).EQ.0) GOTO 200
           DO 150 K = 1,4
             DIFF=TOP(I,J,K)-BOT(I,J,K)
C CORRECT DIFFS LESS THAN OR EQUAL TO ZERO
             IF (DIFF.LE.0) THEN
               COREC=ABS(DIFF)
               BOT(I,J,K)=BOT(I,J,K) - COREC-1.00
             ENDIF
             IF (K.EQ.4) GOTO 150
C CORRECT DIFFDNS LESS THAN ZERO ONLY
             DIFFDN=BOT(I,J,K)-TOP(I,J,K+1)
             IF(DIFFDN.LT.0) THEN
               COREC=ABS(DIFFDN)
               TOP(I,J,K+1)=TOP(I,J,K+1)-COREC-1.00
             ENDIF
 150       CONTINUE
 200     CONTINUE
C BEGIN WRITING
         WRITE(21,'(10F8.1)') (TOP(I,J,1),I = 1,NCOL)
         WRITE(22,'(10F8.1)') (BOT(I,J,1),I = 1,NCOL)
         WRITE(23,'(10F8.1)') (TOP(I,J,2),I = 1,NCOL)
         WRITE(24,'(10F8.1)') (BOT(I,J,2),I = 1,NCOL)
         WRITE(25,'(10F8.1)') (TOP(I,J,3),I = 1,NCOL)
         WRITE(26,'(10F8.1)') (BOT(I,J,3),I = 1,NCOL)
         WRITE(27,'(10F8.1)') (TOP(I,J,4),I = 1,NCOL)
         WRITE(28,'(10F8.1)') (BOT(I,J,4),I = 1,NCOL)
 300  CONTINUE
      STOP
      END
```

APPENDIX A-5. Program ANGUNCON.FOR

```
C     PROGRAM ANGUNCON BY JOHN HOAGLUND
      REAL GRID(361,470)
      REAL TOP(361,470,4), BOT(361,470,4)
      INTEGER IBOUND(361,470),LAYNUM,COUNT
      CHARACTER*40 FILIN
      NROW=470
      NCOL=361
      NLAY=4
      WRITE(*,*) 'ENTER THE UNCONFORMITY STRUCTURAL TOP ARRAY ',
     $'FILENAME :'
      READ(*,'(A)') FILIN
      OPEN(10,FILE=FILIN,STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE IBOUND ARRAY OF THE FIRST LAYER ',
     $'PINCHED OUT BY THE UNCONFORMITY: '
      READ(*,'(A)') FILIN
      OPEN(9,FILE=FILIN,STATUS='UNKNOWN')
      WRITE(*,*) 'THE TOPMOST LAYER IN THE MODEL IS LAYER NUMBER 1 '
      WRITE(*,*) 'ENTER THE LAYER NUMBER OF THE FIRST LAYER ',
     $'PINCHED OUT BY THE UNCONFORMITY: '
      READ(*,*) LAYNUM
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE TOP OF LAYER 1:'
      READ(*,'(A)') FILIN
      OPEN(11,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(21,FILE=FILIN//'.UNC',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE BOT OF LAYER 1: '
      READ(*,'(A)') FILIN
      OPEN(12,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(22,FILE=FILIN//'.UNC',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE TOP OF LAYER 2:'
      READ(*,'(A)') FILIN
      OPEN(13,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(23,FILE=FILIN//'.UNC',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE BOT OF LAYER 2: '
      READ(*,'(A)') FILIN
      OPEN(14,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(24,FILE=FILIN//'.UNC',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE TOP OF LAYER 3:'
      READ(*,'(A)') FILIN
      OPEN(15,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(25,FILE=FILIN//'.UNC',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE BOT OF LAYER 3: '
      READ(*,'(A)') FILIN
      OPEN(16,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(26,FILE=FILIN//'.UNC',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE TOP OF LAYER 4:'
      READ(*,'(A)') FILIN
      OPEN(17,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
```

```
      OPEN(27,FILE=FILIN//'.UNC',STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE 8 CHARACTER NAME OF THE BOT OF LAYER 4: '
      READ(*,'(A)') FILIN
      OPEN(18,FILE=FILIN//'.DAT',STATUS='UNKNOWN')
      OPEN(28,FILE=FILIN//'.UNC',STATUS='UNKNOWN')
C BEGIN READING
      DO 100 J = 1,NROW
        READ(9,'(25I3)') (IBOUND(I,J), I = 1,NCOL)
        READ(10,'(10F8.1)') (GRID(I,J), I = 1,NCOL)
        READ(11,'(10F8.1)') (TOP(I,J,1), I = 1,NCOL)
        READ(12,'(10F8.1)') (BOT(I,J,1), I = 1,NCOL)
        READ(13,'(10F8.1)') (TOP(I,J,2), I = 1,NCOL)
        READ(14,'(10F8.1)') (BOT(I,J,2), I = 1,NCOL)
        READ(15,'(10F8.1)') (TOP(I,J,3), I = 1,NCOL)
        READ(16,'(10F8.1)') (BOT(I,J,3), I = 1,NCOL)
        READ(17,'(10F8.1)') (TOP(I,J,4), I = 1,NCOL)
        READ(18,'(10F8.1)') (BOT(I,J,4), I = 1,NCOL)
 100  CONTINUE
      DO 300 J = 1,NROW
         DO 200 I = 1, NCOL
           COUNT=0
           DO 150 K = LAYNUM,(NLAY-1)
             IF (IBOUND(I,J).EQ.0) THEN
                TOP(I,J,K)=GRID(I,J) + (NLAY-LAYNUM-COUNT)
                IF (K.EQ.LAYNUM) BOT(I,J,K-1)=GRID(I,J) +
     $                                 (NLAY-LAYNUM-COUNT)
                BOT(I,J,K)=TOP(I,J,K)-1.0
                IF (K.EQ.(NLAY-1)) BOT(I,J,K)=GRID(I,J)
             ENDIF
             COUNT=COUNT+1
 150       CONTINUE
 200     CONTINUE
C BEGIN WRITING
         WRITE(21,'(10F8.1)') (TOP(I,J,1),I = 1,NCOL)
         WRITE(22,'(10F8.1)') (BOT(I,J,1),I = 1,NCOL)
         WRITE(23,'(10F8.1)') (TOP(I,J,2),I = 1,NCOL)
         WRITE(24,'(10F8.1)') (BOT(I,J,2),I = 1,NCOL)
         WRITE(25,'(10F8.1)') (TOP(I,J,3),I = 1,NCOL)
         WRITE(26,'(10F8.1)') (BOT(I,J,3),I = 1,NCOL)
         WRITE(27,'(10F8.1)') (TOP(I,J,4),I = 1,NCOL)
         WRITE(28,'(10F8.1)') (BOT(I,J,4),I = 1,NCOL)
 300  CONTINUE
      STOP
      END
```

APPENDIX A–6. Program FLOODFIL.FOR

```
      REAL X,Y,XI,YI,MINDIS(361,470)
      REAL MINX,MAXX,MINY,MAXY,DELX,DELY
      REAL PYTHAG
```

```
      REAL SCAN,YBAK,YFOR,X0,Y0,YFOR0
      REAL PASS(100)
      INTEGER GRDX,GRDY,COUNT,INOUT,NSCAN
      INTEGER IBOOL(361,470),INBOOL(361,470),IBBOOL(361,470)
      CHARACTER*1 ANSWR
      CHARACTER*40 FILIN
      CHARACTER*12 GRDFIL
      CHARACTER    GRDNAM(12)
      CHARACTER*8  SHITOUT,FILOUT
      CHARACTER*4  GRD, BINAS
      COMMON/GRID/MINX,MAXX,MINY,MAXY,DELX,DELY,GRDX,GRDY,IBBOOL,MINDIS
      COMMON/SORT/SCAN(100,470),X(2),Y(2),YBAK,YFOR,NSCAN
      EQUIVALENCE (GRDNAM(1),SHITOUT),(GRDNAM(9),GRD)
      EQUIVALENCE (GRDFIL,GRDNAM(1))
C INITIALIZE FILE VARIABLES
      GRD = '.BOL'
C NSCAN IS THE MAXIMUM NUMBER OF LINE SEGMENTS THAT WILL INTERSECT
C A SINGLE HORIZONTAL SCANLINE.  IT DIMENSIONS THE SCAN(NSCAN,GRDY)
C ARRAY. NSCAN SHOULD BE INCREASED FOR EXTREMELY IRREGULAR POLYGONS
      NSCAN=100
      DELX=1000
      DELY=1000
C UL CORNER COORDINATES GIVEN FOR THE FOUR CORNER PIXELS
C IMPLY 361,470 GRID, BUT (Y-1000) LL AND LR CORNERS
      GRDX=361
      GRDY=470
C CENTROID MINIMUM X COORDINATE AT RASA 1,1
      MINX=290568
C CENTROID MAXIMUM Y COORDINATE AT RASA 1,1
      MAXY=557128
      MAXX = MINX + (GRDX-1)*DELX
      MINY = MAXY - (GRDY-1)*DELY
C BINARY OR ASCII VARIABLE IS DSAA FOR ASCII OR DSBB FOR BINARY
      BINAS='DSAA'
      WRITE (*,*) ' *****************************'
      WRITE (*,*) ' *         FLOODFILL         *'
      WRITE (*,*) ' *            by             *'
      WRITE (*,*) ' *      John Hoaglund        *'
      WRITE (*,*) ' *****************************'
      WRITE (*,*)
      WRITE (*,*) 'VECTOR IS CORRECTOR, BUT RASTER IS FASTER!!'
      WRITE (*,*)
      WRITE (*,*) 'ENTER THE CLOSED-CURVE CRSTM-XYZ INPUT FILENAME: '
      READ(*,'(A)') FILIN
      WRITE(*,*) 'ENTER THE FIRST 8 CHARACTERS OF THE OUTPUT FILENAME: '
      READ(*,'(A8)') FILOUT
      SHITOUT = FILOUT
      WRITE(*,'(A)') GRDFIL
C OPEN FILES
C OPEN INPUT FILE OF CRSTM XYZ COORDINATES
```

```
      OPEN (11,FILE=FILIN,STATUS='OLD')
C OPEN BOOLEAN MATRIX FILE
      OPEN (17,FILE=GRDFIL,STATUS='UNKNOWN')
C INITIALIZE COUNTER
      COUNT=0
 8000 IF (COUNT.GT.0) THEN
        WRITE(*,*) 'DO YOU WANT TO OPEN ANOTHER CLOSED-CURVE CRSTM-XYZ',
     $   ' INPUT FILENAME (Y,y or N,n) : '
        READ(*,'(A)') ANSWR
        IF ((ANSWR.EQ.'n').OR.(ANSWR.EQ.'N')) GOTO 9000
        CLOSE(11)
        WRITE (*,*) 'ENTER THE CLOSED-CURVE CRSTM-XYZ INPUT FILENAME: '
        READ(*,'(A)') FILIN
C OPEN INPUT FILE OF CRSTM XYZ COORDINATES
        OPEN (11,FILE=FILIN,STATUS='OLD')
      ENDIF
C INITIALIZE COUNTER AGAIN
      COUNT = 0
C INITIALIZE MINDIS ARRAY FOR RASTER GRID TO MAX GRID-ENCLOSED LENGTH
      A=MAXX-MINX
      B=MAXY-MINY
      DIST=PYTHAG(A,B)
      DO 10 J = 1,GRDY
        DO 20 I = 1,GRDX
          MINDIS(I,J)=DIST
 20     CONTINUE
 10   CONTINUE
C INITIALIZE SCAN ARRAY TO NEGATIVE 1
      DO 5 J = 1, GRDY
        DO 6 I = 1,NSCAN
           SCAN(I,J) = -1
 6      CONTINUE
 5    CONTINUE

C READ A CRSTM XYZ FILE, DEFINE BOUNDARY PIXELS IN BOOLEAN MATRIX

      K=1
 1000 READ (11,*,END=1010) X(K),Y(K)
      COUNT = COUNT + 1
      WRITE(*,*) 'VERTEX #',COUNT
C CALCULATE NEAREST I J ADDRESS OF LINE SEGMENT ENDPOINT
      I = NINT((X(K)-MINX)/DELX) + 1
      J = NINT((MAXY-Y(K))/DELY) + 1
C ENTER 1 IN BOOLEAN ARRAY
      IBBOOL(I,J) = 1
      IF (COUNT.EQ.1) THEN
             X0=X(K)
             Y0=Y(K)
             K=K+1
             GOTO 1000
```

```
      ENDIF
      IF (COUNT.EQ.2) YFOR0=Y(2)
C DEFINE LINE SEGMENT FROM TWO ENDPOINTS, CHECK LINE SEGMENT LENGTH
C IF GREATER THAN DELX, THEN SWEEP RASTER GRID TO FIND ALL BOUNDARY
C PIXELS (PIXELS WITHIN DELX OF THE LINE SEGMENT)
C DEFINE X,Y INTERSECTIONS FOR EACH J-ROW Y
      A=X(2)-X(1)
      B=Y(2)-Y(1)
      DIST=PYTHAG(A,B)
      IF (DIST.GT.DELX) CALL RSTRDS(X(1),Y(1),X(2),Y(2))
      CALL NTRSXN(A,B,COUNT)
C PREPARE TO READ NEXT ENDPOINT
C IF K=1, UPDATE TO 2; IF K = 2 RESET TO 1
      IF (K.EQ.1) THEN
          K=2
      ELSE
          K=1
      ENDIF
      GOTO 1000
 1010 CONTINUE
C CONNECT LAST POINT WITH FIRST POINT, LAST POINT STORAGE
C DETERMINED BY SEGMENT COUNT ODD OR EVEN:
C LAST POINT IS (X(1),Y(1)) IF ODD, (X(2),Y(2)) IF EVEN
      DIVITWO=COUNT/2.0
      REINT=NINT(COUNT/2.0)
C IF EVEN, MOVE LAST POINT STORAGE TO (X(1),Y(1)) POSITION
      IF ((DIVITWO).EQ.(REINT)) THEN
         X(1)=X(2)
         Y(1)=Y(2)
      ENDIF
C NOW INITIAL POINT IS (X2,Y2) AND LAST POINT IS (X1,Y1)
      A=X0-X(1)
      B=Y0-Y(1)
      DIST=PYTHAG(A,B)
      IF (DIST.GT.DELX) CALL RSTRDS(X0,Y0,X(1),Y(1))
      CALL NTRSXN(A,B,COUNT)
C TEST IF FIRST POINT IS LOCAL MINIMUM VERTEX INTERSECTION
      YBAK=Y(1)
      YFOR=YFOR0
      DO 1300 J=1,GRDY
        YI= MAXY-(J-1)*DELY
        IF (YI.EQ.Y0) THEN
           IF (((YFOR.LT.YI).AND.(YBAK.LT.YI))
     $      .OR.((YFOR.GT.YI).AND.(YBAK.GT.YI))) THEN
             XI=X0
C STORE XI ONCE MORE FOR INTERSECTION
             DO 1302 I = 1,NSCAN
                IF (SCAN(I,J).LT.0) THEN
                    SCAN(I,J) = XI
                    GOTO 1303
```

```
              ENDIF
 1302        CONTINUE
 1303        CONTINUE
            ENDIF
         ENDIF
 1300 CONTINUE
C SORT THE SCAN ARRAY
      DO 1400 J = 1,GRDY
         DO 1450 I = 1,NSCAN
            PASS(I)=SCAN(I,J)
 1450    CONTINUE
         CALL HPSORT(NSCAN,PASS)
         DO 1475 I = 1,NSCAN
            SCAN(I,J) = PASS(I)
 1475    CONTINUE
 1400 CONTINUE
C BOUNDARY COMPLETE

C FLOOD FILL INTERIOR REGIONS OF RASTER GRID
C FOR EACH ROW
C RE-INITIALIZE INBOOL ARRAY
      DO 6000 J = 1,GRDY
        DO 6500 I = 1,GRDX
           INBOOL(I,J) = 0
 6500   CONTINUE
 6000 CONTINUE
C PROMPT FOR INTERIOR REGION IDENTIFIER BOOLEAN INTEGER
      WRITE(*,*) 'ENTER THE BOOLEAN INTEGER FOR THE INTERIOR REGION:'
      READ (*,*) INOUT
      DO 3000 J = 1, GRDY
        DO 3100 K = 1,NSCAN,2
          IF (SCAN(K,J).LT.0) GOTO 3100
          DO 3500 I = 1,GRDX
           XI = MINX + (I-1)*DELX
           IF((XI.GE.SCAN(K,J)).AND.(XI.LE.SCAN(K+1,J)))
     $               INBOOL(I,J)=INOUT
 3500     CONTINUE
 3100   CONTINUE
 3000 CONTINUE
      DO 9100 J = 1,GRDY
         DO 9200 I = 1,GRDX
            IF (INBOOL(I,J).NE.0) IBOOL(I,J)=INBOOL(I,J)
 9200    CONTINUE
 9100 CONTINUE
      GOTO 8000
 9000 CONTINUE
C WRITE OUTPUT FILES
      OPEN (13,FILE='BOUNDARY.FIL',STATUS='UNKNOWN')
      DO 7700 J = 1,GRDY
C WRITE BOUNDARY.FIL BOOLEAN ARRAY
```

```
        WRITE(13,2700) (IBBOOL(I,J),I=1,GRDX)
C WRITE THE INTERIOR BOOLEAN ARRAY
        WRITE(17,2750) (IBOOL(I,J),I=1,GRDX)
 7700 CONTINUE
 2700 FORMAT(75I1)
 2750 FORMAT(25I3)
C CLOSE FILES AND QUIT
      CLOSE(11)
      CLOSE(17)
      CLOSE(13)
      STOP
      END

C FUNCTION PYTHAG FROM NUMERICAL RECIPES
      FUNCTION PYTHAG(A,B)
      REAL A,B,PYTHAG
C THIS FUNCTION FROM PRESS ET. AL., 1992, NUMERICAL RECIPES IN FORTRAN
C COMPUTES PYTHAGOREAN ROOT WITHOUT DESTRUCTIVE UNDERFLOW OR OVERFLOW
C FUNCTION NOT REPRODUCED HERE, SEE PRESS ET. AL., 1992, P. 62-63.

C SUBROUTINE HPSORT FROM NUMERICAL RECIPES
      SUBROUTINE hpsort(n,ra)
C THIS SUBROUTINE FROM PRESS ET. AL., 1992, NUMERICAL RECIPES IN FORTRAN
C SORTS AN ARRAY OF REAL NUMBERS { RA(N) }FROM LOWEST TO HIGHEST
C SUBROUTINE NOT REPRODUCED HERE, SEE PRESS ET. AL., 1992, P. 329

      SUBROUTINE RSTRDS(X1,Y1,X2,Y2)
C DECLARE GLOBAL VARIABLES
      REAL MINX,MAXX,MINY,MAXY,DELX,DELY,MINDIS(361,470)
      REAL X1,Y1,X2,Y2
      INTEGER GRDX,GRDY,IBBOOL(361,470)
C DECLARE LOCAL VARIABLES
      REAL X, Y, A, B, MDIST, DIST
      REAL XI,YI,T,BPLC,BPLI,BPLJ,BPDI,BPDJ
      REAL PYTHAG
      COMMON/GRID/MINX,MAXX,MINY,MAXY,DELX,DELY,GRDX,GRDY,IBBOOL,MINDIS
C  INTEGER ADDRESS FROM REAL COORDINATE
C      I = INT((X(K)-MINX)/DELX) + 1
C      J = INT((MAXY-Y(K))/DELY) + 1
C  REAL COORDINATE FROM INTEGER ADDRESS
C      X(I,J) = (I-1)*DELX + MINX
C      Y(I,J) = MAXY-(J-1)*DELY
      DO 100 J = 1 , GRDY
        DO 200 I = 1,GRDX
C  CALCULATE X AND Y FOR THE NODE
          X = (I-1)*DELX + MINX
          Y = MAXY-(J-1)*DELY
C  CALCULATE DISTANCE TO EACH ENDPOINT OF THE (SINGULAR) LINE SEGMENT
C  STORE MINIMUM
          A=X-X1
```

```
          B=Y-Y1
          MDIST=PYTHAG(A,B)
          A=X-X2
          B=Y-Y2
          DIST=PYTHAG(A,B)
          IF (DIST.LT.MDIST) MDIST=DIST
C  CALCULATE MINIMUM DISTANCE TO LINE AT R0.
C  USING PARALLEL-PERPENDICULAR DECOMPOSITION
C  B IS  (X-X1)i + (Y-Y1)j
C  V IS (X2-X1)i + (Y2-Y1)j
C CALCULATE B PARALLEL COEFFICIENT AND COMPONENTS
          A=(X2-X1)
          B=(Y2-Y1)
          BPLC=((X-X1)*A+(Y-Y1)*B)/(PYTHAG(A,B)**2)
          BPLI=BPLC*A
          BPLJ=BPLC*B
C CALCULATE B PERPENDICULAR COEFFICIENTS
          BPDI=(X-X1)-BPLI
          BPDJ=(Y-Y1)-BPLJ
C CALCULATE INTERSECTION POINT, R0
          XI=X1+BPLI
          YI=Y1+BPLJ
C CALCULATE T PARAMETER OF INTERSECTION POINT ON LINE OF LINE SEGMENT
          IF (A.EQ.0) THEN
             T=(YI-Y1)/B
          ELSE
             T=(XI-X1)/A
          ENDIF
C  IF R0 BETWEEN ENDPOINTS, RESET MDIST
          IF ((T.GT.0.).AND.(T.LT.1.)) THEN
             A=BPDI
             B=BPDJ
             MDIST=PYTHAG(A,B)
          ENDIF
C  FOR ANY POINT WITHIN HALF-DELX UPDATE BOOLEAN ARRAY
          IF (MDIST.LT.(DELX/2)) IBBOOL(I,J) = 1
C  KEEP TRACK OF MINDIS IN ARRAY FOR COMPARISON WITH OTHER
C  LINE SEGMENTS (IE OTHER RSTRDS SUBROUTINE CALLS)
          IF (MDIST.LT.MINDIS(I,J)) MINDIS(I,J) = MDIST
 200    CONTINUE
 100  CONTINUE
      RETURN
      END

      SUBROUTINE NTRSXN(A,B,COUNT)
C DECLARE GLOBAL VARIABLES
      REAL MINX,MAXX,MINY,MAXY,DELX,DELY,MINDIS(361,470)
      INTEGER GRDX,GRDY,IBBOOL(361,470),COUNT,NSCAN
      REAL A,B,X,Y
      REAL SCAN,YBAK,YFOR
```

```
C DECLARE LOCAL VARIABLES
      REAL T,YI,XI
      COMMON/GRID/MINX,MAXX,MINY,MAXY,DELX,DELY,GRDX,GRDY,IBBOOL,MINDIS
      COMMON/SORT/SCAN(100,470),X(2),Y(2),YBAK,YFOR,NSCAN
C FIND INTERSECTIONS FOR EACH J-ROW Y VALUE
C FIRST TEST FOR DUPLICATE POINTS
      IF ((A.EQ.0).AND.(B.EQ.0)) GOTO 2001
       DO 1300 J = 1, GRDY
         YI=MAXY-(J-1)*DELY
C CALCULATE T PARAMETER OF INTERSECTION POINT ON LINE OF LINE SEGMENT
C DEAL WITH TANGENTS AND NON-INTERSECTIONS FIRST, B=0 IMPLIES Y(1)=Y(2)
         IF (B.EQ.0) THEN
           IF (YI.EQ.Y(1)) THEN
              T=0
           ELSE
              T=2
           ENDIF
           GOTO 1301
         ENDIF
         T=(YI-Y(1))/B
C IF T BETWEEN 0 AND 1 INCLUSIVE, INTERSECTION EXISTS
 1301    IF ((T.LE.1).AND.(T.GE.0)) THEN
           XI=X(1) + A*T
C STORE XI FOR (J,NSCAN)
           DO 1302 I = 1,NSCAN
              IF (SCAN(I,J).LT.0) THEN
                  SCAN(I,J) = XI
                  GOTO 1303
              ENDIF
 1302      CONTINUE
 1303      CONTINUE
C IF VERTEX INTERSECTION, CHECK IF VERTEX LOCAL MAX OR MIN
C LOGIC REQUIRES LOCAL MAX OR MIN VERTEX TO CORRESPOND TO T=0
C ON NEXT SUBROUTINE CALL, THEREFORE STORE Y(1) AS YBAK
           IF ((T.EQ.0).OR.(T.EQ.1)) THEN
               IF (T.EQ.1) YBAK=Y(1)
               IF ((T.EQ.0).AND.(COUNT.NE.1)) THEN
                   YFOR=Y(2)
                   IF (((YFOR.LT.Y(1)).AND.(YBAK.LT.Y(1)))
     $              .OR.((YFOR.GT.Y(1)).AND.(YBAK.GT.Y(1)))) THEN
C STORE XI ONCE MORE (TWICE TOTAL) FOR LOCAL MAX OR MIN
                      DO 1304 I = 1,NSCAN
                        IF (SCAN(I,J).LT.0) THEN
                            SCAN(I,J) = XI
                            GOTO 1305
                        ENDIF
 1304                 CONTINUE
 1305                 CONTINUE
                   ENDIF
               ENDIF
```

```
          ENDIF
        ENDIF
 1300   CONTINUE
 2001 CONTINUE
      RETURN
      END
```

APPENDIX A--7. Program CANOE.FOR

```
      INTEGER IRVBND(361,470),ISTORE(361,470,2)
      INTEGER IBOUND(361,470)
      INTEGER IBOLCON(100,3000),JBOLCON(100,3000)
      INTEGER KOUNT,NSEG
      INTEGER TOT(3000)
      CHARACTER*1 FLAG(361,470)
      INTEGER ORDER(361,470),CONF(6)
      INTEGER A,B,C,D
      REAL STORE(361,470,3)
      CHARACTER*40 FILIN
      NROW=470
      NCOL=361
      NLAY=4
      WRITE(*,*) 'ENTER THE NAME OF THE INPUT RIVER PACKAGE FILE: '
      READ(*,'(A)') FILIN
      OPEN(10,FILE=FILIN,STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE FILENAME OF THE IBOUND ARRAY: '
      READ(*,'(A)') FILIN
      OPEN(11,FILE=FILIN,STATUS='UNKNOWN')
      WRITE(*,*) 'ENTER THE NAME OF THE OUTPUT FILE: '
      READ(*,'(A)') FILIN
      OPEN(21,FILE=FILIN//'.FL1',STATUS='UNKNOWN')
      OPEN(22,FILE=FILIN//'.FL2',STATUS='UNKNOWN')
      OPEN(23,FILE=FILIN//'.FL3',STATUS='UNKNOWN')
      OPEN(24,FILE=FILIN//'.OR1',STATUS='UNKNOWN')
      OPEN(25,FILE=FILIN//'.OR2',STATUS='UNKNOWN')
      OPEN(26,FILE=FILIN//'.OR3',STATUS='UNKNOWN')
      OPEN(27,FILE=FILIN//'.STG',STATUS='UNKNOWN')
      OPEN(28,FILE='PROBLEM.SEG',STATUS='UNKNOWN')
      OPEN(29,FILE='NEWRIVER.RIV',STATUS='UNKNOWN')
C INITIALIZE IRVBND AND ORDER TO ZERO AND READ MODEL IBOUND ARRAY
      DO 10 J=1,NROW
        DO 15 I=1,NCOL
           IRVBND(I,J)=0
           ORDER(I,J)=0
           DO 13 K = 1,3
             STORE(I,J,K)=0.0
 13        CONTINUE
 15     CONTINUE
        READ(11,'(25I3)') (IBOUND(I,J),I=1,NCOL)
 10   CONTINUE
```

```
C BEGIN READING
      READ(10,50) MXRIVR,IRIVCB
      READ(10,75) ITMP
      WRITE(29,50) MXRIVR,IRIVCB
      WRITE(29,75) ITMP
 25   READ(10,100,END=125) ILAY,J,I,STAGE,COND,RBOT,INTB,IBASIN
      STORE(I,J,1)=STAGE
      STORE(I,J,2)=COND
      STORE(I,J,3)=RBOT
      ISTORE(I,J,1)=INTB
      ISTORE(I,J,2)=IBASIN
      IRVBND(I,J)=1
      GOTO 25
C RIVER PACKAGE FORMATS
 50   FORMAT (2I10)
 75   FORMAT (I10)
 100  FORMAT (3I10,3F10.0,I10,I7)
 101  FORMAT (3I10,F10.1,F10.5,F10.1,I10,I7)
 125  CONTINUE
C SET FLAG TO IRVBND
      DO 130 J = 1,NROW
        DO 140 I = 1,NCOL
          IF (IRVBND(I,J).EQ.0) FLAG(I,J)='0'
          IF (IRVBND(I,J).EQ.1) FLAG(I,J)='1'
 140     CONTINUE
 130  CONTINUE
C      WRITE(21,50) MXRIVR,IRIVCB
C      WRITE(21,75) ITMP
C  ****
C DETERMINE HEADWATERS, CONFLUENCES, AND MOUTHS
C
      INTB=0
      IBASIN=0
        DO 1776 L = 2,469
         DO 1215 K = 2,360
           IF (IRVBND(K,L).EQ.1) THEN
            NRIV=0
            I=K
            J=L
            A=I-1
            B=I+1
            C=J-1
            D=J+1
C CALCULATE TOTAL NEIGHBOR RIVERS
C THREE TOP
            NRIV = IRVBND(A,C) + IRVBND(I,C) + IRVBND(B,C)
C TWO MIDDLE
            NRIV=NRIV + IRVBND(A,J) + IRVBND(B,J)
C THREE BOTTOM
            NRIV = NRIV + IRVBND(A,D) + IRVBND(I,D) + IRVBND(B,D)
```

```
C CALCULATE STAR ON-NEIGHBORS AND DIAG ON-NEIGHBORS
            ISTAR=IRVBND(I,C) + IRVBND(I,D) + IRVBND(A,J) + IRVBND(B,J)
            IDIAG=IRVBND(A,C) + IRVBND(A,D) + IRVBND(B,C) + IRVBND(B,D)
C CONVERT ISTAR AND IDIAG TO OFF-NEIGHBORS
            ISTAR=4-ISTAR
            IDIAG=4-IDIAG
C DETERMINE ENDPONTS
            IF (NRIV.EQ.1) FLAG(I,J)='S'
C DETERMINE CONFLUENCE NODES FROM ISTAR AND IDIAG
C FIVE NODE
            IF (NRIV.EQ.5) THEN
              IF (ISTAR.EQ.3) FLAG(I,J)='*'
            ENDIF
C FOUR NODE
            IF (NRIV.EQ.4) THEN
              IF ((ISTAR.EQ.1).OR.(ISTAR.EQ.4)) FLAG(I,J)='*'
            ENDIF
C THREE NODE
            IF (NRIV.EQ.3) THEN
              IF ((ISTAR.EQ.1).OR.(ISTAR.EQ.4)) FLAG(I,J)='*'
              IF (ISTAR.EQ.3) THEN
C     TEST: IF DIAGONAL ONS ARE NOT DIAGONALLY OPPOSED
C           THEN DIAGONAL OFFS ARE NOT DIAGONALLY OPPOSED
C           THEREFORE A CONFLUENCE
                 IF (IRVBND(A,C)+IRVBND(A,D).EQ.0) FLAG(I,J)='*'
                 IF (IRVBND(B,C)+IRVBND(B,D).EQ.0) FLAG(I,J)='*'
                 IF (IRVBND(A,C)+IRVBND(B,C).EQ.0) FLAG(I,J)='*'
                 IF (IRVBND(A,D)+IRVBND(B,D).EQ.0) FLAG(I,J)='*'
              ENDIF
              IF (ISTAR.EQ.2) THEN
C     TEST: IF STAR ONS ARE IN THE SAME ROW OR COLUMN
C           THEN STAR OFFS ARE IN THE SAME COL OR ROW
C           THEREFORE NOT A CONFLUENCE
C           OTHERWISE A CONFLUENCE
                 IF (IRVBND(A,L)+IRVBND(B,L).EQ.0) GOTO 1812
                 IF (IRVBND(K,C)+IRVBND(K,D).EQ.0) GOTO 1812
                 FLAG(I,J)='*'
1812             CONTINUE
              ENDIF
            ENDIF
C DETERMINE MOUTHS
C IF ENDPOINT IS NEAR - IBOUND, RECLASSIFY AS A MOUTH
            IF (FLAG(I,J).EQ.'S') THEN
                IF (IBOUND(A,C).LT.0) FLAG(I,J)='M'
                IF (IBOUND(K,C).LT.0) FLAG(I,J)='M'
                IF (IBOUND(B,C).LT.0) FLAG(I,J)='M'
                IF (IBOUND(A,L).LT.0) FLAG(I,J)='M'
                IF (IBOUND(B,L).LT.0) FLAG(I,J)='M'
                IF (IBOUND(A,D).LT.0) FLAG(I,J)='M'
                IF (IBOUND(K,D).LT.0) FLAG(I,J)='M'
```

```
                IF (IBOUND(B,D).LT.0) FLAG(I,J)='M'
            ENDIF
           ENDIF
 1215    CONTINUE
 1776   CONTINUE
C *****
C BEGIN CANOEING
C
C RESET FLAGS FOR ST. JOE RIVER IN SOUTH
C OUT
      FLAG(90,458)='1'
C IN
      FLAG(38,456)='1'
      NSEG=0
C SWEEP AND SYSTEMATICALLY FIND ALL HEADWATERS
      DO 1861 J = 1,NROW
        DO 1865 I = 1,NCOL
          IF (FLAG(I,J).EQ.'S') THEN
            NSEG=NSEG+1
C START NUMBERING REACHES FOR SEGMENT
            KOUNT=0
            ORDER(I,J)=1
            K=I
            L=J
C INCREMENT LOOP FOR REACHES PER SEGMENT
 1872       A=K-1
            B=K+1
            C=L-1
            D=L+1
            KOUNT=KOUNT+1
C IF THE NODE IS A CONFLUENCE, STORE IT
C CHECK NEIGHBORS TO CONDITIONALLY SET
C OR NOT SET (NS) ORDER:  NSEG=NSEG+1,KOUNT=1
            IF (FLAG(K,L).EQ.'*') THEN
               TOT(NSEG)=KOUNT
               IBOLCON(KOUNT,NSEG)=K
               JBOLCON(KOUNT,NSEG)=L
               NS=0
C THREE TOP
               IF ((IRVBND(A,C).EQ.1).AND.(ORDER(A,C).EQ.0)) NS=NS+1
               IF ((IRVBND(K,C).EQ.1).AND.(ORDER(K,C).EQ.0)) NS=NS+1
               IF ((IRVBND(B,C).EQ.1).AND.(ORDER(B,C).EQ.0)) NS=NS+1
C TWO MIDDLE
               IF ((IRVBND(A,L).EQ.1).AND.(ORDER(A,L).EQ.0)) NS=NS+1
               IF ((IRVBND(B,L).EQ.1).AND.(ORDER(B,L).EQ.0)) NS=NS+1
C THREE BOTTOM
               IF ((IRVBND(A,D).EQ.1).AND.(ORDER(A,D).EQ.0)) NS=NS+1
               IF ((IRVBND(K,D).EQ.1).AND.(ORDER(K,D).EQ.0)) NS=NS+1
               IF ((IRVBND(B,D).EQ.1).AND.(ORDER(B,D).EQ.0)) NS=NS+1
C IF NUMBER OF NEIGHBOR RIVERS NOT SET (NS) IS GREATER THAN 1, OR 0, FIND A
```

```
NEW HEADWATER
               IF (NS.NE.1) GOTO 1865
C ELSE INCREMENT NUMBER OF SEGMENTS, SET REACH KOUNT TO 1,
C      STORE CONFLUENCE NODE REACH IN SEG-REACH CONNECTIVITY BOOLEAN, AGAIN
               NSEG=NSEG+1
               KOUNT=1
               IBOLCON(KOUNT,NSEG)=K
               JBOLCON(KOUNT,NSEG)=L
C      SET CONFLUENCE ORDER BY COUNTING ORDERED NEIGHBORS
C       AND KEEPING TRACK THE MAX ORDERS
C          RE-INITIALIZE
               M=1
               DO 1890 N = 1,6
                  CONF(N)=0
 1890          CONTINUE
C          THREE TOP
               IF (ORDER(A,C).GT.0) THEN
                  CONF(M)=ORDER(A,C)
                  M=M+1
               ENDIF
               IF (ORDER(K,C).GT.0) THEN
                  CONF(M)=ORDER(K,C)
                  M=M+1
               ENDIF
               IF (ORDER(B,C).GT.0) THEN
                  CONF(M)=ORDER(B,C)
                  M=M+1
               ENDIF
C          TWO MIDDLE
               IF (ORDER(A,L).GT.0) THEN
                  CONF(M)=ORDER(A,L)
                  M=M+1
               ENDIF
               IF (ORDER(B,L).GT.0) THEN
                  CONF(M)=ORDER(B,L)
                  M=M+1
               ENDIF
C          THREE BOTTOM
               IF (ORDER(A,D).GT.0) THEN
                  CONF(M)=ORDER(A,D)
                  M=M+1
               ENDIF
               IF (ORDER(K,D).GT.0) THEN
                  CONF(M)=ORDER(K,D)
                  M=M+1
               ENDIF
               IF (ORDER(B,D).GT.0) THEN
                  CONF(M)=ORDER(B,D)
                  M=M+1
               ENDIF
```

```
C            TAKE ORDERS THAT ARE DIFFERENT, ELSE
C            ORDERS ARE THE SAME
               MXCNT=0
               MAX=CONF(1)
               MIN=MAX
               DO 1903 N = 1,M
                  IF (CONF(N).GT.MAX) THEN
                      MAX = CONF(N)
                      MXCNT=0
                  ENDIF
                  IF ((CONF(N).NE.0).AND.(CONF(N).LT.MIN)) MIN = CONF(N)
                  IF (CONF(N).EQ.MAX) MXCNT=MXCNT+1
 1903          CONTINUE
C          NOW TAKE MAX OR MAX+1 OF CONF(M) ARRAY ENTRIES
               IF ((MAX.EQ.MIN).OR.(MXCNT.GT.1)) ORDER(K,L) = MAX + 1
               IF ((MAX.NE.MIN).AND.(MXCNT.EQ.1)) ORDER(K,L) = MAX
C      WRITE CONFLUENCE TO NEW RIVER PACKAGE FILE USING STREAM ORDER FOR
CONDUCTANCE
C      AND RBOT (DEPTH FROM STAGE)
C               IF (ORDER(K,L).EQ.1) COND=(0.010763911)*1*3.28084
C               IF (ORDER(K,L).EQ.2) COND=(0.010763911)*2*3.28084
C               IF (ORDER(K,L).EQ.3) COND=(0.010763911)*20*3.28084
C               IF (ORDER(K,L).EQ.4) COND=(0.010763911)*60*3.28084
C               IF (ORDER(K,L).EQ.5) COND=(0.010763911)*80*3.28084
C               WRITE(29,101) ILAY,L,K,STORE(K,L,1),COND,
C     $          STORE(K,L,3),ISTORE(K,L,1),ISTORE(K,L,2)
C      AND CONTINUE DOWNSTREAM
               GOTO 1914
            ENDIF
C STORE NON-CONFLUENCE NODE REACH IN SEG-REACH CONNECTIVITY BOOLEAN
            IBOLCON(KOUNT,NSEG)=K
            JBOLCON(KOUNT,NSEG)=L
C WRITE NON-CONFLUENCE TO NEW RIVER PACKAGE FILE USING STREAM ORDER FOR
CONDUCTANCE
C AND RBOT (DEPTH FROM STAGE)
C            IF (ORDER(K,L).EQ.1) COND=(0.010763911)*1*3.28084
C            IF (ORDER(K,L).EQ.2) COND=(0.010763911)*2*3.28084
C            IF (ORDER(K,L).EQ.3) COND=(0.010763911)*20*3.28084
C            IF (ORDER(K,L).EQ.4) COND=(0.010763911)*60*3.28084
C            IF (ORDER(K,L).EQ.5) COND=(0.010763911)*80*3.28084
C            WRITE(29,101) ILAY,L,K,STORE(K,L,1),COND,
C     $       STORE(K,L,3),ISTORE(K,L,1),ISTORE(K,L,2)

C IF END OF THE LINE (IE. FLAG = M OR S AGAIN), STORE TOT(L) FOR SEGMENT
C IF SEGMENT ENDS ON A HEADLAND (FLAG=S) OR MOUTH (FLAG=M), COUNT IT! NEXT
C SEGMENT WILL BE INCREMENTED AT TOP OF LOOP (1865 CONTINUE STATEMENT)

            IF ((FLAG(K,L).EQ.'M').OR.
     $       ((FLAG(K,L).EQ.'S').AND.(KOUNT.GT.1))) THEN
                TOT(NSEG)=KOUNT
```

```
               GOTO 1865
           ENDIF
 1914      CONTINUE
C ELSE, FIND NEIGHBORING UNORDERED RIVER NODE
C SET ITS ORDER
C AND MOVE THERE
C THREE TOP
C CONTINUE ST. JOE IN SOUTH FROM OUT TO IN
           IF ((K.EQ.90).AND.(L.EQ.458)) THEN
              ORDER(38,456)=ORDER(K,L)
              K=38
              L=456
              GOTO 1872
           ENDIF
C END ST JOE CONTINUITY
           IF ((IRVBND(A,C).EQ.1).AND.(ORDER(A,C).EQ.0)) THEN
              IF (FLAG(A,C).NE.'*') ORDER(A,C)=ORDER(K,L)
              L = C
              K = A
              GOTO 1872
           ENDIF
           IF ((IRVBND(K,C).EQ.1).AND.(ORDER(K,C).EQ.0)) THEN
              IF (FLAG(K,C).NE.'*') ORDER(K,C)=ORDER(K,L)
              L = C
              K = K
              GOTO 1872
           ENDIF
           IF ((IRVBND(B,C).EQ.1).AND.(ORDER(B,C).EQ.0)) THEN
              IF (FLAG(B,C).NE.'*') ORDER(B,C)=ORDER(K,L)
              L = C
              K = B
              GOTO 1872
           ENDIF
C TWO MIDDLE
           IF ((IRVBND(A,L).EQ.1).AND.(ORDER(A,L).EQ.0)) THEN
              IF (FLAG(A,L).NE.'*') ORDER(A,L)=ORDER(K,L)
              L = L
              K = A
              GOTO 1872
           ENDIF
           IF ((IRVBND(B,L).EQ.1).AND.(ORDER(B,L).EQ.0)) THEN
              IF (FLAG(B,L).NE.'*') ORDER(B,L)=ORDER(K,L)
              L = L
              K = B
              GOTO 1872
           ENDIF
C THREE BOTTOM
           IF ((IRVBND(A,D).EQ.1).AND.(ORDER(A,D).EQ.0)) THEN
              IF (FLAG(A,D).NE.'*') ORDER(A,D)=ORDER(K,L)
              L = D
```

```
               K = A
               GOTO 1872
            ENDIF
            IF ((IRVBND(K,D).EQ.1).AND.(ORDER(K,D).EQ.0)) THEN
               IF (FLAG(K,D).NE.'*') ORDER(K,D)=ORDER(K,L)
               L = D
               K = K
               GOTO 1872
            ENDIF
            IF ((IRVBND(B,D).EQ.1).AND.(ORDER(B,D).EQ.0)) THEN
               IF (FLAG(B,D).NE.'*') ORDER(B,D)=ORDER(K,L)
               L = D
               K = B
               GOTO 1872
            ENDIF
C LAST TOT(NSEG)=KOUNT FOR SINGLE STRAND STREAMS WITH NO MOUTH
            TOT(NSEG)=KOUNT
C CLOSING ENDIF FOR SEARCH FOR S HEADLAND
          ENDIF
 1865   CONTINUE
 1861 CONTINUE
C OUTPUTS
C WRITE FLAG AND ORDER IN TIERS FOR VIEWING
      DO 2000 J = 1,NROW
        WRITE(21,2001) (FLAG(I,J), I = 1,123)
        WRITE(22,2001) (FLAG(I,J), I = 124,246)
        WRITE(23,2002) (FLAG(I,J), I = 247,361)
        WRITE(24,2004) (ORDER(I,J), I = 1,123)
        WRITE(25,2004) (ORDER(I,J), I = 124,246)
        WRITE(26,2005) (ORDER(I,J), I = 247,361)
 2000 CONTINUE
C WRITE IBOLCON ARRAY TO SCREEN FOR EDITING
C      DO 3000 J = 1,NSEG
C           WRITE(*,*) (IBOLCON(I,J),I=1,50)
C 3000 CONTINUE
C 3500 FORMAT (25I3)
C USE IBOLCON AND JBOLCON ARRAYS TO INTERPOLATE STAGES BETWEEN
C HEADLANDS AND CONFLUENCES
      DO 4000 L = 1,NSEG
C DEBUG WRITE
        WRITE (27,*) L,TOT(L)
C ENDEBUG
        DIFF=STORE(IBOLCON(1,L),JBOLCON(1,L),1)-
     $  STORE(IBOLCON(TOT(L),L),JBOLCON(TOT(L),L),1)
C DEBUG BLOCKS
        IF (TOT(L).EQ.0) THEN
           WRITE (27,*) '0 REACHES FOR SEGMENT #'
           WRITE (27,*) L
           GOTO 4000
        ENDIF
```

```
C DIFF CRITERION FOR PROBLEM SEGS.  IF < 0, SEGMENT IS UPHILL
      IF (DIFF.LE.-30) THEN
C WRITE TO .STG FILE
         WRITE(27,5000) IBOLCON(1,L),JBOLCON(1,L),
     $   IBOLCON(TOT(L),L),JBOLCON(TOT(L),L)
         WRITE(27,*) STORE(IBOLCON(1,L),JBOLCON(1,L),1),
     $   STORE(IBOLCON(TOT(L),L),JBOLCON(TOT(L),L),1),DIFF
C WRITE TO PROBLEM SEGS FILE
         WRITE(28,5000) IBOLCON(1,L),JBOLCON(1,L),
     $   IBOLCON(TOT(L),L),JBOLCON(TOT(L),L)
         WRITE(28,*) STORE(IBOLCON(1,L),JBOLCON(1,L),1),
     $   STORE(IBOLCON(TOT(L),L),JBOLCON(TOT(L),L),1),DIFF
C DETERMINE LAT LONG AND QUADRANGLE OF PROBLEM SEG STARTING REACH
         I=IBOLCON(1,L)
         J=JBOLCON(1,L)
         CALL QUAD(I,J)
C SKIP WRITING PROBLEM SEGS IN NEW RIVER FILE
C         GOTO 4000
      ENDIF
C END DEBUG BLOCKS
C DETERMINE DECREMENT FOR INTERPOLATION OF STAGE
      IF (TOT(L).GT.1) THEN
         DECREM=DIFF/(TOT(L)-1)
      ENDIF
C WRITE OUT I,J,STORE(I,J,1) TO .STG FILE (UNIT 27) AND NEW RIVER FILE
      I=IBOLCON(1,L)
      J=JBOLCON(1,L)
C WRITE STARTING NODE FOR SEGMENT TO NEW RIVER PACKAGE FILE USING
C STREAM ORDER FOR CONDUCTANCE
C AND RBOT (DEPTH FROM STAGE)
C COND = FACTORS FOR MODFLOWP
      IF (ORDER(I,J).EQ.1) COND=1*3.28084
      IF (ORDER(I,J).EQ.2) COND=2*3.28084
      IF (ORDER(I,J).EQ.3) COND=20*3.28084
      IF (ORDER(I,J).EQ.4) COND=60*3.28084
      IF (ORDER(I,J).EQ.5) COND=80*3.28084
      WRITE(29,101) ILAY,J,I,STORE(I,J,1),COND,
     $ STORE(I,J,3),ISTORE(I,J,1),ISTORE(I,J,2)
      WRITE(27,*) I,J,STORE(I,J,1)
      IF (TOT(L).LT.2) GOTO 4000
      DO 4500 K = 2,TOT(L)
        I=IBOLCON(K,L)
        J=JBOLCON(K,L)
C INTERPOLATE THE STAGE AND SUBTRACT 2.3 FOOT DEPTH FOR RBOT
        STORE(I,J,1)=
     $  STORE(IBOLCON(1,L),JBOLCON(1,L),1)-(K-1)*DECREM
        STORE(I,J,3)=STORE(I,J,1) - 2.3
C WRITE REMAINING INTERPOLATED NODES FOR SEGMENT TO NEW RIVER PACKAGE
C FILE USING STREAM ORDER FOR CONDUCTANCE
C AND RBOT (DEPTH FROM STAGE)
```

```
C COND = FACTORS FOR MODFLOWP
          IF (ORDER(I,J).EQ.1) COND=1*3.28084
          IF (ORDER(I,J).EQ.2) COND=2*3.28084
          IF (ORDER(I,J).EQ.3) COND=20*3.28084
          IF (ORDER(I,J).EQ.4) COND=60*3.28084
          IF (ORDER(I,J).EQ.5) COND=80*3.28084
          WRITE(29,101) ILAY,J,I,STORE(I,J,1),COND,
     $    STORE(I,J,3),ISTORE(I,J,1),ISTORE(I,J,2)
          WRITE(27,*) I,J,STORE(I,J,1)
 4500   CONTINUE
 4000 CONTINUE
 5000 FORMAT('UPHILL OR FLAT SEGMENT FROM COL ',I3,' ROW ',I3,
     $' TO COL ',I3,' ROW ',I3)
 2001 FORMAT(123A1)
 2002 FORMAT(115A1)
 2004 FORMAT(123I1)
 2005 FORMAT(115I1)
C WRITE NUMBER OF SEGS FOUND
      WRITE(*,*) NSEG
      STOP
      END
      SUBROUTINE QUAD(I,J)
      DOUBLE PRECISION X,Y,LAT,LONG
      REAL RMN, RMW
      INTEGER DN,MN,SN,DW,MW,SW
      CALL NAVI (I,J,X,Y)
      CALL TM2GEO (X,Y,LAT,LONG)
C CLASSIFY LAT LONG INTO 7.5 MINUTE QUADS, 7.5 MIN = .125 DEGREE
      K=INT((86.875-LONG)/.125) + 1
      L=INT((45.875-LAT)/.125) + 1
C TRANSLATE K AND L TO LETTER NUMBER DESIGNATION?
      LL=CHAR((INT(L/2) + 6) + 65)
      KK=INT(K/2) + 15
C CONVERT LAT LONG TO DEGREE MINUTE SECOND
      DN=INT(LAT)
      RMN=(LAT-DN)*60
      MN=INT(RMN)
      SN=INT((RMN-MN)*60)
      DW=INT(LONG)
      RMW=(LONG-DW)*60
      MW=INT(RMW)
      SW=INT((RMW-MW)*60)
      WRITE(28,*)
      WRITE(28,2003) DN,MN,SN,DW,MW,SW,LL,KK
 2003 FORMAT ('STARTING REACH AT: ',3I3,5X,3I3,'     QUAD: ',A1,I2)
      RETURN
      END
C SUBROUTINE NAVI
      SUBROUTINE NAVI(I,J,X,Y)
      DOUBLE PRECISION X,Y,MINX,MAXY,DELX,DELY
```

```
      INTEGER GRDX,GRDY
      DELX=1000
      DELY=1000
      GRDX=361
      GRDY=470
C CENTROID MINIMUM X IS 290568 AT 1,1 AND 1,470
C CENTROID MAXIMUM Y IS 557128 AT 1,1
      MINX=290568
      MAXY=557128
      MAXX = MINX + (GRDX-1)*DELX
      MINY = MAXY - (GRDY-1)*DELY
      X = MINX + (I-1)*DELX
      Y = MAXY - (J-1)*DELY
      RETURN
      END
C REPRODUCTION OF THE INVERSE PROCESSING (TM TO GEODETIC) PART
C OF THE CRSTM CODE OF BRIAN BUCKLEY (CRS)
C FROM U.S.G.S. BULL. 1532 BY SNYDER AS
C DELIVERED BY DAVE LUSCH (CRS)
C EARTH STUFF VARIABLE DECLARATIONS
      SUBROUTINE TM2GEO(X,Y,LAT,LONG)
      DOUBLE PRECISION PI,X0,Y0,LONG0,LAT0,K0,M0,AR,ESQ,EPSQ
C VARIABLES OF SOLUTION
      DOUBLE PRECISION LAT,LONG,X,Y
C RADIAN EQUIVALENTS
      DOUBLE PRECISION PHI0,PHI1
C INVERSE COEFFICIENTS
      DOUBLE PRECISION E1,N1,T1,C1,R1,D,M,MU
C      CHARACTER*40 FILIN
C OPEN INPUT FILES
C      WRITE (*,*) '***********************************'
C      WRITE (*,*) '*                                 *'
C      WRITE (*,*) '*             TM2GEO              *'
C      WRITE (*,*) '*               by                *'
C      WRITE (*,*) '*          JOHN HOAGLUND          *'
C      WRITE (*,*) '*                                 *'
C      WRITE (*,*) '***********************************'
C      WRITE (*,*) 'ENTER THE CRSTM X,Y,Z FILENAME: '
C      READ (*,'(A)') FILIN
C      OPEN (11,FILE=FILIN,STATUS='UNKNOWN')
C      OPEN (12,FILE='GEO.OUT',STATUS='UNKNOWN')
C BEGIN PROCESSING
      PI=3.14159265358979323846264338327950288419716939937511
C SCALE ON CENTRAL MERIDIAN (0.9996 FOR UTM)
      K0=0.9996
C EQUATORIAL RADIUS OR SEMIMAJOR AXIS OF THE CLARK (1866) ELLIPSOID
      AR=6378206.4
C POLAR RADIUS OR SEMIMINOR AXIS OF THE CLARK (1866) ELLIPSOID
C      BR=  UNKNOWN, CAN BE CALCULATED FROM AR AND ESQ
C ECCENTRICITY OF THE CLARK (1866) ELLIPSOID SQUARED CALCULATED
```

```
C      ESQ=((1-(BR*BR)/(AR*AR)))
C ECCENTRICITY OF THE CLARK (1866) ELLIPSOID GIVEN PAGE 232
      ESQ=.00676866
      E=SQRT(ESQ)
C ORIGIN OF CRSTM AS GIVEN BY DAVE LUSCH
      X0=359987
      Y0=344917
      LONG0=-86.000
      LAT0=44.000
C FOR A GIVEN X,Y CALCULATE LATITUDE AND LONGITUDE
C LONGITUDE IS ASSUMED TO BE GIVEN WEST OF GREENWICH
C THEREFORE USE NEGATIVE AS PER PG XII
C 2000 READ(11,*,END=2001) X,Y,Z
C 2000 CONTINUE
      X=X-X0
      Y=Y-Y0
C FORWARD PROCESSING COEFFICIENTS
      PHI0=LAT0*PI/180
C EQUATION 3-21 FOR M0
C      M0=AR*((1-ESQ/4-3*(ESQ**2)/64-5*(ESQ**3)/256)*LAT0-
C     $(3*ESQ/8+3*(ESQ**2)/32+45*(ESQ**3)/1024)*SIN(2*PHI0)+
C     $(15*(ESQ**2)/256+45*(ESQ**3)/1024)*SIN(4*PHI0)-
C     $(35*(ESQ**3)/3072)*SIN(6*PHI0))
C USE SIMPLIFIED FORMULA 3-22 FOR M0 OF CLARK ELLIPSOID
      M0=111132.0894*LAT0-16216.94*SIN(2*PHI0)+17.21*SIN(4*PHI0)-
     $0.02*SIN(6*PHI0)
C EQNS 8-12 AND 8-15
      EPSQ=ESQ/(1-ESQ)
C  USE 8-20 FOR M
      M = M0 + Y/K0
C EQN 3-24 SOLVED FOR E1
      E1=(1-SQRT(1-ESQ))/(1+SQRT(1-ESQ))
C EQN 8-19
      MU=M/(AR*(1-ESQ/4-3*ESQ*ESQ/64-5*ESQ**3/256))
C EQN 3-26
      PHI1=MU + (3*E1/2-27*(E1**3)/32)*SIN(2*MU) + (21*(E1**2)/16-
     $55*(E1**4)/32)*SIN(4*MU) + (151*(E1**3)/96)*SIN(6*MU)
C NOW EQNS 8-21 TO 8-25
      N1=AR/SQRT(1-ESQ*(SIN(PHI1)*SIN(PHI1)))
      T1=TAN(PHI1)*TAN(PHI1)
      C1=EPSQ*COS(PHI1)*COS(PHI1)
      R1=AR*(1-ESQ)/SQRT((1-ESQ*SIN(PHI1)*SIN(PHI1))**3)
      D=X/(N1*K0)
C CALCULATE LAT AND LONG USING EQNS 8-17 AND 8-18
      LAT=( PHI1-(N1*TAN(PHI1)/R1)*(D*D/2-(5+3*T1+10*C1-4*C1*C1
     $-9*EPSQ)*(D**4)/24+(61+90*T1+298*C1+45*T1*T1-252*EPSQ-3*C1*C1)
     $*(D**6)/720) )*(180/PI)
      LONG=LONG0+( (D-(1+2*T1+C1)*(D**3)/6+(5-2*C1+28*T1-3*C1*C1+
     $8*EPSQ+24*T1*T1)*D**5/120)/COS(PHI1) )*(180/PI)
      LONG=-LONG
```

```
C      WRITE (12,*) LAT,LONG,Z
C      GOTO 2000
C 2001 CONTINUE
      RETURN
      END
```

APPENDIX A--8. Program PRMPRCNT.FOR

```
C  PROGRAM TO GENERATE HORIZONTAL AND VERTICAL PERMEABILITY ARRAYS
C  FOR MODFLOW (AND VCONT) FROM HORIZONTAL AND VERTICAL
C  PERMEABILITY OF "SAND" AND "CLAY" AND PERCENT SAND
C  DECLARATIONS
      REAL KV1,KH1,KV2,KH2
      REAL VPERM,HPERM
      REAL PRCS,KVINV
      INTEGER NCOL,NROW
      CHARACTER*40 FILIN
      DIMENSION VPERM(361,470),HPERM(361,470)
      DIMENSION KV1(361,470), KH1(361,470)
      DIMENSION KV2(361,470), KH2(361,470)
      DIMENSION PRCS(361,470)
C  PROMPT FOR FILENAMES
      WRITE(*,*) ' ENTER THE NUMBER OF COLUMNS IN THE MODEL: '
      READ(*,*) NCOL
      WRITE(*,*) ' ENTER THE NUMBER OF ROWS IN THE MODEL: '
      READ(*,*) NROW
C PROMPT BLOCK FOR OUTPUT FILES
      WRITE(*,*) ' ENTER THE FIRST EIGHT CHARACTERS OF THE ',
     $' OUTPUT FILENAME:'
      READ(*,'(A)') FILIN
      OPEN ( 9,FILE=FILIN//'.KH ',STATUS='UNKNOWN')
      OPEN (10,FILE=FILIN//'.KV ',STATUS='UNKNOWN')
C PROMPT BLOCK FOR HORIZONTAL HYDRAULIC CONDUCTIVITY
      WRITE(*,*) ' ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) ' HORIZONTAL HYDRAULIC CONDUCTIVITY OF THE "SAND":'
      READ(*,'(A)') FILIN
      OPEN (11,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) ' ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) ' HORIZONTAL HYDRAULIC CONDUCTIVITY OF THE "CLAY":'
      READ(*,'(A)') FILIN
      OPEN (12,FILE=FILIN,STATUS='OLD')
C PROMPT BLOCK FOR VERTICAL HYDRAULIC CONDUCTIVITY AND PERCENT "SAND"

      WRITE(*,*) ' ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) ' VERTICAL HYDRAULIC CONDUCTIVITY OF THE "SAND":'
      READ(*,'(A)') FILIN
      OPEN (13,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) ' ENTER THE FILENAME OF THE ARRAY CONTAINING '
```

```
      WRITE(*,*) ' VERTICAL HYDRAULIC CONDUCTIVITY OF THE "CLAY":'
      READ(*,'(A)') FILIN
      OPEN (14,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) 'ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) 'PERCENT "SAND":'
      READ(*,'(A)') FILIN
      OPEN (15,FILE=FILIN, STATUS='OLD')
C  READ IN DATA
      DO 200 J = 1,NROW
         READ(11,100) (KH1(I,J),I = 1,NCOL)
         READ(12,100) (KH2(I,J),I = 1,NCOL)
         READ(13,100) (KV1(I,J),I = 1,NCOL)
         READ(14,100) (KV2(I,J),I = 1,NCOL)
         READ(15,101) (PRCS(I,J),I = 1,NCOL)
C IN AND OUT FORMATS
 100     FORMAT(8E10.5)
 101     FORMAT(10F8.1)
 200  CONTINUE
C CALCULATE VCONT ARRAY
      DO 300 J=1,NROW
        DO 400 I = 1 ,NCOL
C REDEFINE PERCENT AS FRACTION
          PRCS(I,J)=PRCS(I,J)/100.0
C CALCULATE HPERM USING FREEZE AND CHERRY EQN 2.32 MODIFIED FOR PERCENT
          HPERM(I,J)=PRCS(I,J)*KH1(I,J) + (1.0 - PRCS(I,J))*KH2(I,J)
C CALCULATE VPERM USING FREEZE AND CHERRY EQN 2.31 MODIFIED FOR PERCENT
          KVINV=PRCS(I,J)/KV1(I,J) + (1.0 - PRCS(I,J))/KV2(I,J)
          VPERM(I,J)=1/KVINV
 400    CONTINUE
 300  CONTINUE
C WRITE OUT THE VCONT ARRAY
      DO 500 J = 1,NROW
        WRITE(9,100) (HPERM(I,J), I = 1,NCOL)
        WRITE(10,100) (VPERM(I,J), I = 1,NCOL)
 500  CONTINUE
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(14)
      CLOSE(15)
      STOP
      END
```

APPENDIX A–9. VCONT.FOR

```
C  PROGRAM TO GENERATE VCONT ARRAYS FOR MODFLOW FROM VERTICAL
C  PERMEABILITY AND STRUCTURAL TOP AND BOTTOM DATA
C  DECLARATIONS
      REAL DV1,DV2,DZ1,DZ2,DZC,KV
```

```
      INTEGER NCOL,NROW
      CHARACTER ANSW
      CHARACTER*40 FILIN
      DIMENSION VCONT(361,470),TOP(361,470,2),BOT(361,470,2)
      DIMENSION KV(361,470,3)
C  PROMPT FOR FILENAMES
      WRITE(*,*) ' ENTER THE NUMBER OF COLUMNS IN THE MODEL: '
      READ(*,*) NCOL
      WRITE(*,*) ' ENTER THE NUMBER OF ROWS IN THE MODEL: '
      READ(*,*) NROW
C PROMPT BLOCK FOR UPPER ACTIVE LAYER
      WRITE(*,*) ' ENTER THE FIRST EIGHT CHARACTERS OF THE ',
     $' OUTPUT FILENAME:'
      READ(*,'(A)') FILIN
      OPEN (10,FILE=FILIN//'.VCT',STATUS='UNKNOWN')
      WRITE(*,*) ' ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) ' STRUCTURAL TOPS OF THE UPPER ACTIVE LAYER: '
      READ(*,'(A)') FILIN
      OPEN (11,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) ' ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) ' STRUCTURAL BOTTOMS OF THE UPPER ACTIVE LAYER: '
      READ(*,'(A)') FILIN
      OPEN (12,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) 'ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) 'VERTICAL PERMEABILITY OF THE UPPER ACTIVE LAYER: '
      READ(*,'(A)') FILIN
      OPEN (13,FILE=FILIN, STATUS='OLD')
C PROMPT BLOCK FOR INACTIVE INTERMEDIATE LAYER
      WRITE(*,*) 'DOES AN INACTIVE INTERMEDIATE LAYER EXIST ANYWHERE '
      WRITE(*,*) 'BETWEEN THE TWO ACTIVE MODFLOW LAYERS (Y y or N n)? '
      READ(*,'(A)') ANSW
      IF ((ANSW.EQ.'N').OR.(ANSW.EQ.'n')) GOTO 100
      WRITE(*,*) 'ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) 'VERTICAL PERMEABILITY OF THE INACTIVE  '
      WRITE(*,*) 'INTERMEDIATE LAYER: '
      READ(*,'(A)') FILIN
      OPEN (14,FILE=FILIN, STATUS='OLD')
C EXTENT BOOLEAN OF INACTIVE LAYER??
C PROMPT BLOCK FOR LOWER ACTIVE LAYER
 100  WRITE(*,*) ' ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) ' STRUCTURAL TOPS OF THE LOWER ACTIVE LAYER: '
      READ(*,'(A)') FILIN
      OPEN(15,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) ' ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) ' STRUCTURAL BOTTOMS OF THE LOWER ACTIVE LAYER: '
      READ(*,'(A)') FILIN
      OPEN (16,FILE=FILIN,STATUS='OLD')
      WRITE(*,*) 'ENTER THE FILENAME OF THE ARRAY CONTAINING '
      WRITE(*,*) 'VERTICAL PERMEABILITY OF THE LOWER ACTIVE LAYER: '
      READ(*,'(A)') FILIN
```

```
      OPEN (17,FILE=FILIN, STATUS='OLD')
C  READ IN DATA
      DO 200 J = 1,NROW
         READ(11,*) (TOP(I,J,1),I = 1,NCOL)
         READ(12,*) (BOT(I,J,1),I = 1,NCOL)
         READ(13,'(8E10.5)') (KV(I,J,1),I = 1,NCOL)
         READ(15,*) (TOP(I,J,2),I = 1,NCOL)
         READ(16,*) (BOT(I,J,2),I = 1,NCOL)
         READ(17,'(8E10.5)') (KV(I,J,3),I = 1,NCOL)
         IF ((ANSW.EQ.'N').OR.(ANSW.EQ.'n')) GOTO 200
         READ(14,'(8E10.5)') (KV(I,J,2),I = 1,NCOL)
 200  CONTINUE
C CALCULATE VCONT ARRAY
      DO 300 J=1,NROW
        DO 400 I = 1 ,NCOL
          DV1=TOP(I,J,1)-BOT(I,J,1)
          DV2=TOP(I,J,2)-BOT(I,J,2)
          DZ1=DV1/2
          DZ2=DV2/2
C CALCULATION FOR EXISTING INACTIVE INTERMEDIATE LAYER IF DZC POSITIVE

          DZC=BOT(I,J,1)-TOP(I,J,2)
C CALCULATION FOR NO INACTIVE INTERMEDIATE LAYER IF DZC=0

          IF (DZC.EQ.0) KV(I,J,2)=1
          IF ((DZ1.EQ.0).AND.(DZ2.EQ.0)) THEN
             VCONT(I,J) = 1.0
             GOTO 400
          ENDIF
C CALCULATE VCONT ARRAY USING EQUATION 52 PG. 5-16 OF MODFLOW
          VCONT(I,J)= 1/(DZ1/KV(I,J,1) + DZC/KV(I,J,2) + DZ2/KV(I,J,3))

 400    CONTINUE
 300  CONTINUE
C WRITE OUT THE VCONT ARRAY
      DO 500 J = 1,NROW
        WRITE(10,'(8E10.5)') (VCONT(I,J), I = 1,NCOL)
 500  CONTINUE
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      IF ((ANSW.NE.'N').OR.(ANSW.NE.'n')) CLOSE(14)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      STOP
      END
```

APPENDIX B

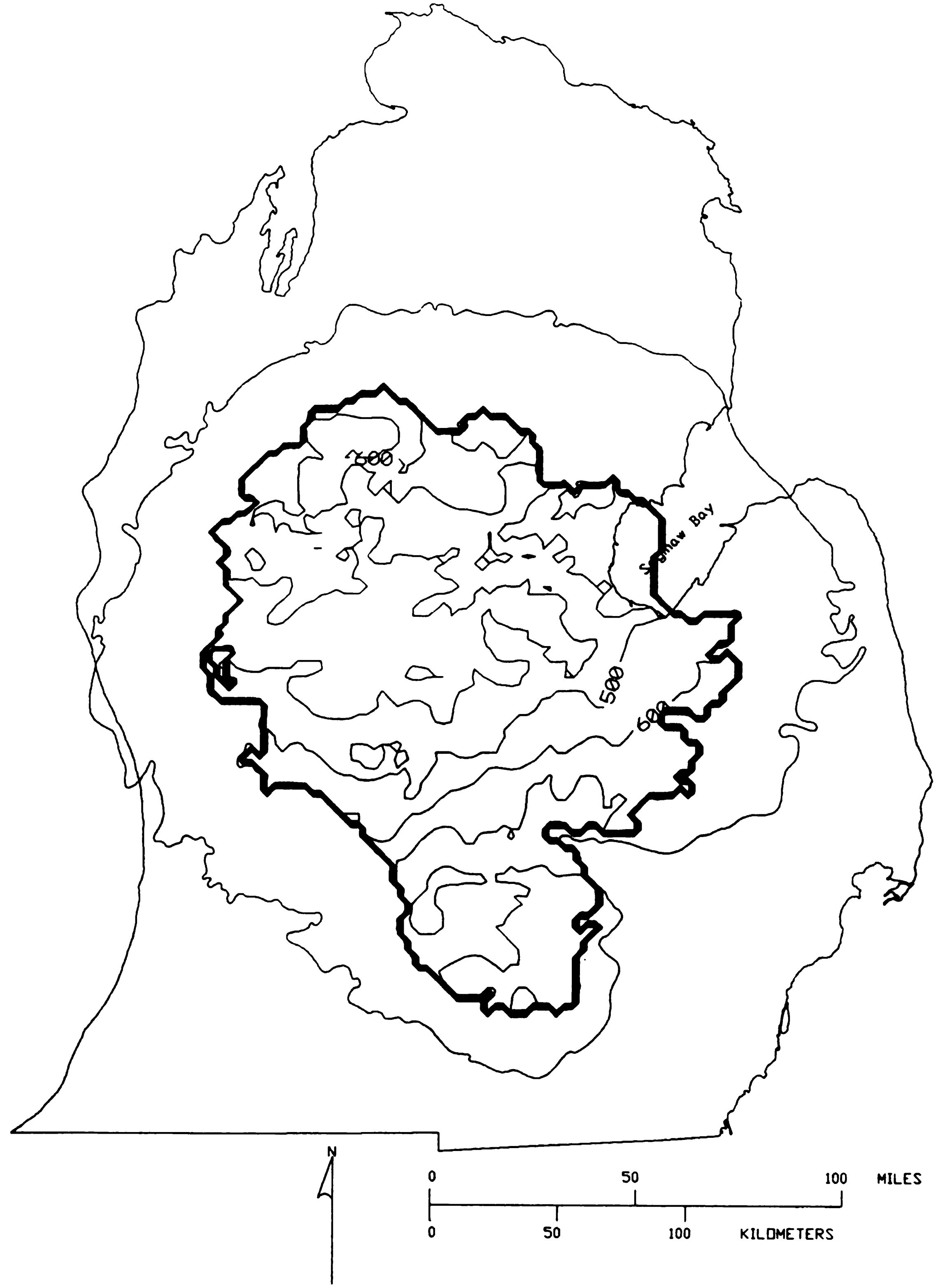

Figure 10. Altitude of Saginaw aquifer top, top of model layer 2, interpolated from data provided by Westjohn and Weaver (1996a).

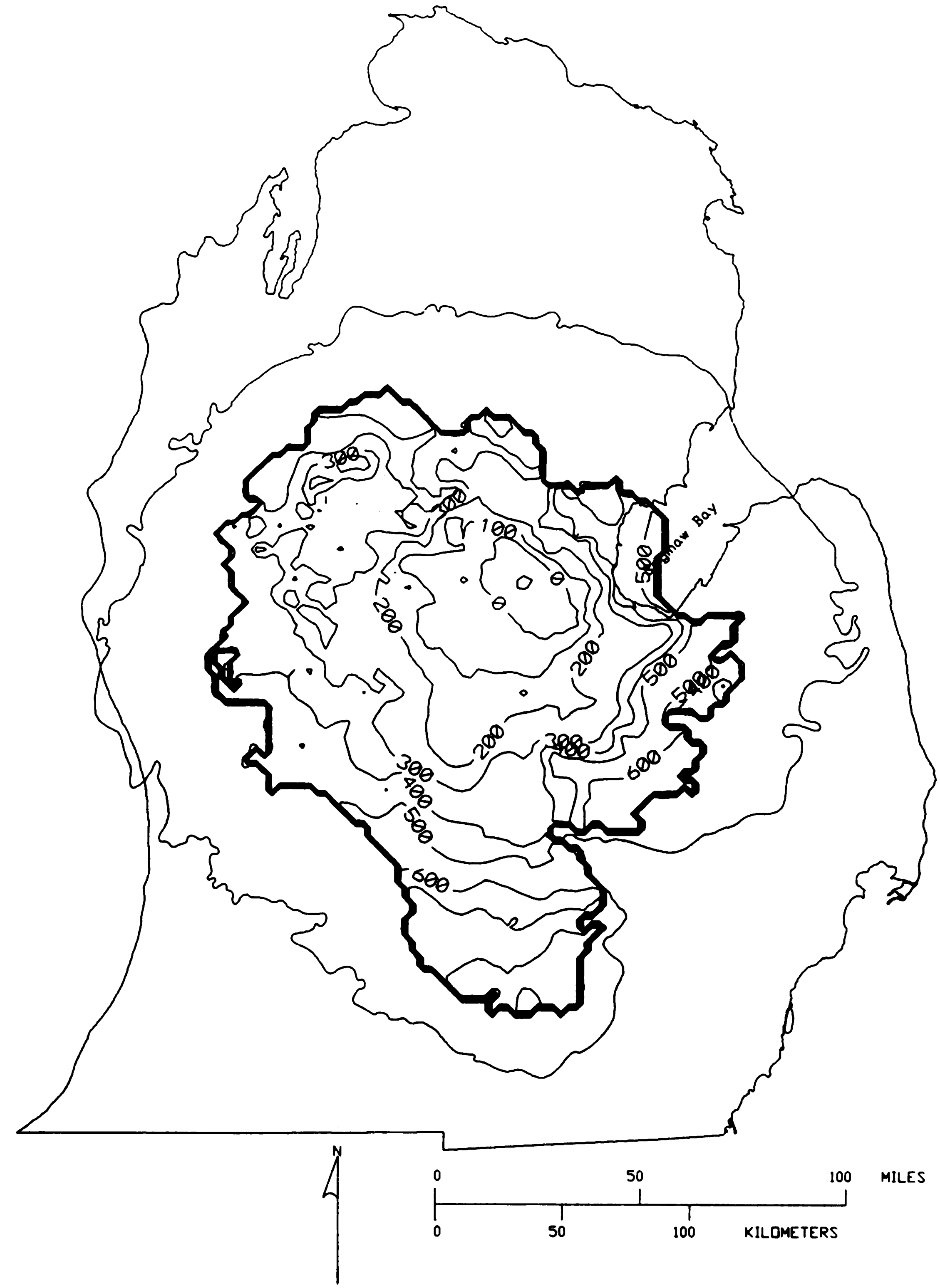

Figure 11. Altitude of Saginaw confining unit top, bottom of model layer 2, digitally constructed from interpolated data provided by Westjohn and Weaver (1996a)

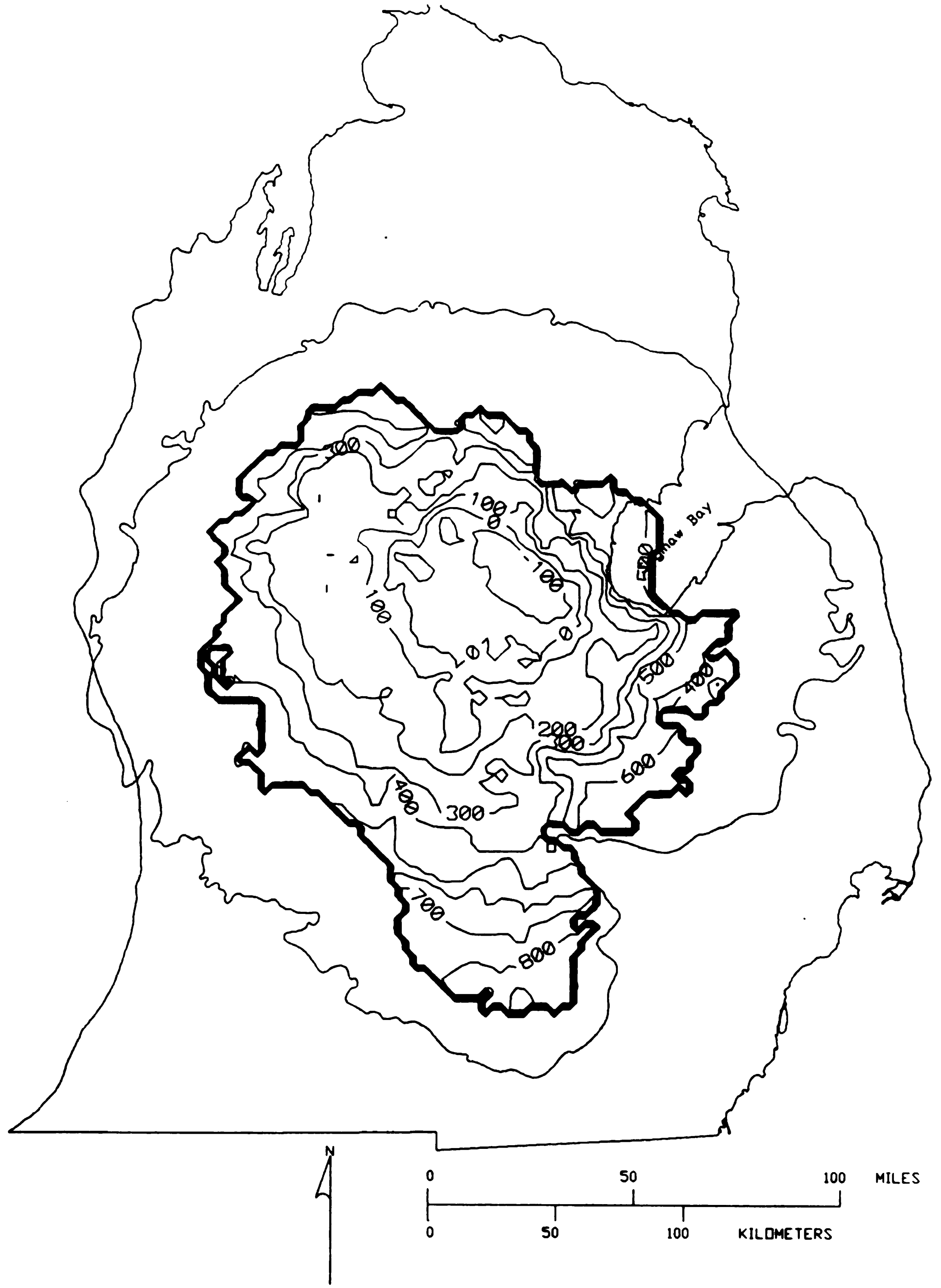

Figure 12. Altitude of the Parma-Bayport aquifer top, top of model layer 3, interpolated from data provided by Westjohn and Weaver (1996a).

Figure 13. Altitude of the Michigan confining unit top, bottom of model layer 3, interpolated from data provided by Westjohn and Weaver (1996b).

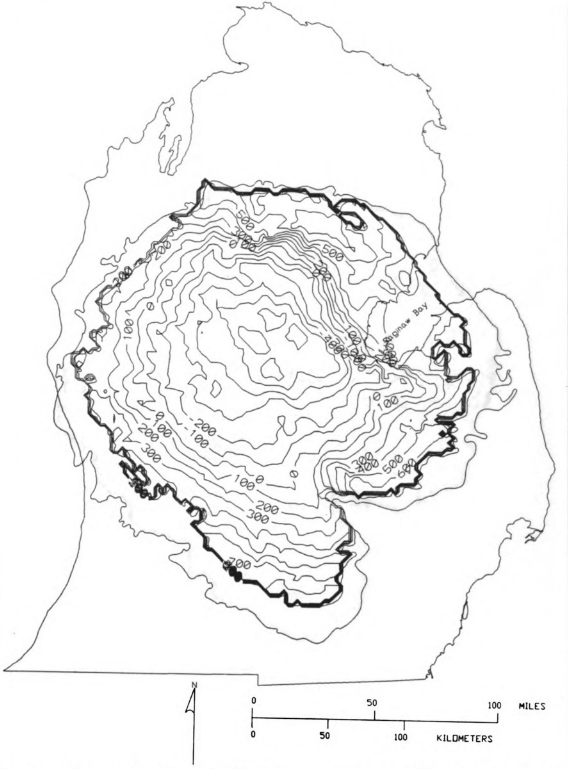

Figure 14. Altitude of the top of the Marshall aquifer, top of model layer 4, interpolated from data provided by Westjohn and Weaver (1996b).

Figure 15. Altitude of the Coldwater confining unit top, bottom of model layer 4, interpolated from data provided by Westjohn and Weaver (1996b).

LIST OF REFERENCES

LIST OF REFERENCES

Alley, R.B., D.A. Meese, C.A. Shuman, A.J. Gow, K.C. Taylor, P.M. Grootes, J.W.C. White, M. Ram, E.D. Waddington, P.A. Mayewski, and G.A. Zielinski, 1993. Abrupt increase in Greenland snow accumulation at the end of the Younger Dryas event. Nature Vol. 362, p. 527-529.

Anderson, M.P., 1979. Using models to simulate the movement of contaminants through groundwater flow systems. Chemical Rubber Company (CRC) Critical Reviews of Environmental Control, Vol. 9, No. 2, p. 97-156.

Bear, J., 1972. Dynamics of Fluids in Porous Media. American Elsevier, New York, 764 p.

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

Boulton, G.S., G.D. Smith, A.S. Jones, and J. Newsome, 1985. Glacial geology and glaciology of the last mid-latitude ice sheets. Geological Society of London Journal, Vol. 142, p. 447-474.

Broecker, W.S. and V.M. Oversby, 1971. Chemical Equilibria in the Earth. McGraw-Hill, New York. 318 p.

Carlson, C.A. 1994. Identification of an isotopic and hydrochemical anomaly in the discharge area of the Fox Hills Aquifer, South Central North Dakota: Evidence for Pleistocene subglacial recharge? Unpublished doctoral thesis, Department of Geological Sciences, Michigan State University, East Lansing, Michigan. 153 p.

Chrzastowski, M., and T. Thompson, 1992. "Late Wisconsinan and Holocene coastal evolution of the southern shore of Lake Michigan," p. 397-413 in Quaternary Coasts of the United States: Marine and Lacustrine Systems, Fletcher, C.H. and J.F. Wehmiller eds. SEPM Special Publication No. 48, 450 p.

Clark, J.A., M. Hendriks, T.J. Timmermans, C. Struck, and K.J. Hilverda, 1994. Glacial isostatic deformation of the Great Lakes region. Geological Society of America Bulletin, Vol. 106, p. 19-31.

COHMAP Project Members, 1988. Climatic changes of the last 18,000 years: Observations and model simulations. Science. Vol. 241, p. 1043-1052.

Craig, H., 1961. Isotopic variations in meteoric waters. Science, 133, p. 1702-1703.

Craig, H. and L.I. Gordon, 1965. "Deuterium and oxygen-18 variations in the ocean and the marine atmosphere," in Stable Isotopes in Oceanographic Studies and Paleotemperatures. Spoleto, July 26-27, 1965. Consiglio Nazionale della Ricerche, Laboratorio di Geologia Nucleare, Pisa, p. 1-22.

Dannemiller, G.T., and L.S. Badalementi, 1988. Oxygen isotope analysis of ground water in the Michigan basin. Geological Society of America, North Central Section, 22nd Annual Meeting, Akron, Ohio. Abstracts with programs, Vol. 20, Issue No. 5, p. 341-2.

Dansgaard, W., 1964. Stable isotopes in precipitation. Tellus, Vol. 16, No. 4, p 436-468.

Dansgaard, W., J.W.C. White, and S.J. Johnson, 1989. The abrupt termination of the Younger Dryas climate event. Nature, Vol. 339. p 532-534.

Downey, J.S., 1986. Geohydrology of bedrock aquifers in the Northern Great Plains in parts of Montana, North Dakota, South Dakota, and Wyoming. U.S. Geological Survery Professional Paper 1402-E, 87 p.

Eddy, J.A. and R.S. Bradley, 1991. Changes in time in the temperature of the earth. Earthquest, Vol. 5 No. 1.

Eschman, D.F. 1985. Summary of the Quaternary history of Michigan, Ohio, and Indiana. Journal of Geological Education, Vol. 33, p. 161-167.

Eschman, D.F. and P.F. Karrow, 1985. "Huron basin glacial lakes: a review," in Quaternary Evolution of the Great Lakes, P.F. Karrow and P.E. Calkin editors. Geological Association of Canada Special Paper 30, 258 p.

Faure, G. 1986. Principles of Isotope Geology. John Wiley and Sons, New York. 589 p.

Filley, T.H., and Paarizek, R.R., 1983, Dynamics of groundwater flow associated with a continental glacier. Geological Society of America, Abstracts with Programs, Vol. 15, p. 572.

Fisher, D.A., N. Reeh, and K. Langley, 1985. Objective reconstructions of the late Wisconsinan Laurentide ice sheet and the significance of deformable beds. Geographie physique et Quaternaire, Vol. 39, p. 229-238.

Frenzel, B., B. Becker, H-J. Beug, K. Brunnacker, D. Busche, P. Frankenberg, P. Fritz, M.A. Geyh, H. Grotzfeld, H. Hagedorn, K. Hecker, J. Hovermann, A. Kessler, W. v. Konigswald, K. Krumsiek, W. Lauer, H. Mensching, H. Moser, K.O. Munnich, Chr. Sonntag, R. Vinken, W. Westendorf, 1992. "Climates at about 7,000 to 6,500 years before present," in Atlas of Paleoclimates and Paleoenvironments of the Northern Hemisphere: Late Pleistocene-Holocene, B. Frenzel, M. P-ecsi, and A. A. Velichko editors. Budapest: Geographical Research Institute, Hungarian Academy of Sciences.

Gelhar, L.W., A. Mantoglu, C. Welty, and K.R. Rehfeldt, 1985. A review of field-scale physical solute transport processes in saturated and unsaturated porous media. Electric Power Resource Institute, Palo Alto, California. Report EA-4190, 116 p.

Ging, P.B., D.T. Long, and R.W. Lee, 1996. Selected geochemical characteristics of groundwater from the Marshall Aquifer in the Central lowqer peninsula of Michigan. U.S. Geological Survey, Water Resources Investigation Report 94-4220, 19 p.

Graf, D.L., W.F. Meents, I. Friedman, and N.F. Shrimp, 1966. The origin of saline formation waters, III: Calcium Chloride Waters. Illinois Geological Survey Circular 397, Tables 1 and 2.

Hansel, A.K., D.M. Mickelson, A. F. Schneider, and C.E. Larsen, 1985. "Late Wisconsinan and Holocene history of the Lake Michigan basin," in Quaternary Evolution of the Great Lakes, P.F. Karrow and P.E. Calkin editors. Geological Association of Canada Special Paper 30, 258 p.

Hoaglund, J.R., G.C. Huffman, and N.J. Grannemann, (in press). Digital simulation of groundwater flow in the Glaciofluvial, Saginaw, and Marshall Aquifers, Central Lower Peninsula, Michigan. Water Supply Paper XXXX.

Houghton D., 1838. "First annual report of the State Geologist," in Geologic Reports of Douglass Houghton, First State Geologist of Michigan 1837-1845, G. N. Fuller, ed. The Michigan Historical Commission, 1928, p. 90 - 153.

Houghton, J.T., G.J. Jenkins, and J.J. Ephraums (editors), 1990. Climate Change: The IPCC Scientific Assessment. Cambridge University Press, Cambridge, U.K.

Hubbert, M.K., 1940. The theory of groundwater motion. Journal of Geology, Vol. 48, p. 785-944.

Hughes, T., G.H. Denton, B.G. Anderson, D.H. Schilling, J.L. Fastook, and C.S. Lingle, 1981. "The last great ice sheet: A global view," in The last great ice sheets, G.H. Denton and T. Hughes, editors. John Wiley and Sons, New York, p. 263-317.

Hyde, W.T., T.J. Crowley, K. Kim, and G.R. North, 1989. Comparison of GCM and Energy Balance Model simulations of seasonal temperature changes over the past 18,000 years. Journal of Climate, Vol. 2. p. 864-887.

Koch, D., P. Mattejat and T. Prickett, 1989. Three dimensional solute transport using the random-walk algorithm and Modflow: User's manual for Premod3d preprocessor and user's manual for Ran3D three dimensional random walk. Engineering Technologies Associates, Inc., Ellicott City, Maryland and T. Prickett and Associates, Urbana, Illinois. U.S. Army Toxic and Hazardous Materials Agency contract #DAAA15-87-D-0220, task 5.

Krabbenhoft, D. P., 1990. Estimating groundwater exchange with lakes. 1. The stable isotope mass balance method. 2. Calibration of a three-dimensional, solute transport model to a stable isotope plume. Water Resources Research, Vol. 26, No. 10, p. 2445-2462.

Kutzbach, J.E. and P.J. Guetter, 1986. The influence of changing orbital parameters and surface boundary conditions on climate simulations for the past 18,000 years. Journal of the Atmospheric Sciences, Vol. 43. p. 1726-1759.

Kutzbach, J.E., 1987. "Model simulations of the climatic patterns during the deglaciation of North America," in The Geology of North America, Volume K-3: North America and Adjacent Oceans During the Last Deglaciation, W.F. Ruddimann and H.E. Wright, Jr. editors. 501 p.

Lane, A.C., 1899a. Water resources of the Lower Peninsula of Michigan. U.S. Geological Survey Water Supply and Irrigation Paper 30, 97 p.

----- 1899b. Lower Michigan Mineral Waters, A study into the connection between their chemical composition and mode of occurrence. U.S. Geological Survey Water Supply and Irrigation Paper 31, 97 p.

Larsen, C.E., 1987. Geological history of glacial lake Algonquin and the upper Great Lakes, U.S. Geological Survey Bulletin 1801. 36 p.

Larson, G.J. and Schaetzel, R., (in press). "Glacial history and physical geography of the Great Lakes basin," in Great Lakes Policy and Management: A Binational Perspective, Taylor, W. H. ed., Michigan State University Press.

Long, D.T., T.P. Wilson, M.J. Takacs, and D.H. Rezabek, 1988. Stable isotope geochemistry of near-surface groundwater: East-central Michigan Basin. Geological Society of America Bulletin, Vol. 100, p. 25-39.

Long, D.T., D.B. Westjohn, and N.G. Grannemann, 1991. Cl and O-18 distributions in Michigan basin groundwater. Geological Society of America, Abstracts with Programs, Vol.

23, No. 5.

Mandle, R. J. and D.B. Westjohn, 1989. Geohydrologic framework and ground-water flow in the Michigan Basin. American Water Resources Association Monograph Series 13. p. 83-110.

Mathews, W.H., 1974. Surface profiles of the Laurentide ice sheet in its marginal areas. Journal of Glaciology, Vol. 13, p. 37-43.

McDonald, M.G. and A.W. Harbaugh, 1988. A modular three-dimensional finite difference groundwater flow model. Techniques of Water Resources Investigations of the U.S. Geological Survey, Book 6, Chapter A1, Modeling Techniques.

McGinnis, L.D., 1968. Glaciation as a possible cause of mineral deposition. Economic Geology, Vol. 63, p. 390-400.

Meissner, B.D. 1993. The geochemistry and source for solutes in groundwater from the Pennsylvanian bedrock sequence in the Michigan basin. Unpublished masters thesis, Department of Geological Sciences, Michigan State University, East Lansing, Michigan. 115 p.

Paterson, W.S.B. 1981. The physics of glaciers [2nd ed.]. Pergamon Press Inc., Elmsford, NY. 380 p.

Prickett, T.A., and C.G. Lonnquist, 1971. Selected digital computer techniques for groundwater resource evaluation. Illinois State Water Survey, Bulletin 55. 62 p.

Prickett, T.A., T.G. Naymik, and C.G. Lonnquist, 1981. A 'random-walk' solute transport model for selected groundwater quality evaluations. Illinois State Water Survey, Bulletin 65, 103 p.

Siegel, D.I., 1991. Evidence for dilution of deep, confined groundwater by vertical recharge of isotopically heavy Pleistocene water. Geology, Vol. 19, p. 433-436.

Siegel, D.I., and R.J. Mandle, 1984. Isotopic evidence for glacial meltwater recharge to the Cambrian-Ordovician aquifer, North-Central United States. Quaternary Research, Vol. 22. p. 328-335

Sun, R.J. and R.H. Johnston, 1994. Regional Aquifer-System Analysis program of the U.S. Geological Survey, 1978-1992. U.S. Geological Survey, Circular 1099, 126 p.

Theis, C.V., 1935. The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using groundwater storage. Transactions of the American

Geophysical Union, Vol. 2, p. 519-524.

Toth, J., 1962. A theory of groundwater motion in small drainage basins in central Alberta. Journal of Geophysical Research, Vol. 67, p. 4375-4387.

Toth, J., 1963. A theoretical analysis of groundwater flow in small drainage basins. Journal of Geophysical Research, Vol. 68, p. 4795-4812.

Twenter, F.R., 1966. [Map (color) of] General availability and quality of groundwater in the bedrock deposits of Michigan. State Resources Planning Division, Michigan Dept. of Commerce, and Water Resources Commission, Michigan Dept. of Conservation, 1 sheet, scale 1:1,000,000.

Velichko, A. A., V.A. Klimanov, and I.I. Borzenkova, 1992. "Climates between 6,000 and 5,500 years before present," in Atlas of Paleoclimates and Paleoenvironments of the Northern Hemisphere: Late Pleistocene-Holocene, B. Frenzel, M. P-ecsi, and A. A. Velichko editors. Budapest: Geographical Research Institute, Hungarian Academy of Sciences.

Wahrer, M. , 1993. The geochemistry and source of solutes in groundwater from the glacial drift regional aquifer, Michigan Basin. Unpublished masters thesis, Dept. of Geolgical Sciences, Michigan State University, 111 p.

Webb, T. , 1991. "The spectrum of temporal climatic variability," in Global Changes of the Past, R.S. Bradley editor. OIES, Boulder, Colorado.

Webb, T., P.J. Bartlein, J.E. Kutzbach, 1987. "Climate change in eastern North America during the past 18,000 years; Comparisons of pollen data with model results," in The Geology of North America, Volume K-3: North America and Adjacent Oceans During the Last Deglaciation, W.F. Ruddimann and H.E. Wright, Jr. editors. 501 p.

Webb, T. and R.A. Bryson, 1972. Late- and postglacial climatic change in the Northern Midwest, USA: Quantitative estimates derived from fossil pollen spectra by multivariate statistical analysis. Quaternary Research, Vol. 2, p. 70-115.

Western Michigan University, Department of Geology, College of Arts and Sciences, 1981. Hydrogeologic Atlas of Michigan. U.S. Environmental Protection Agency Underground Injection Control Program Report, 35 pl., scale 1:500,000.

Westjohn, D.B. and T.L. Weaver, 1996a. Hydrogeologic framework of Pennsylvanian and late Mississippian rocks in the central lower peninsula of Michigan. U.S. Geological Survey, Water Resources Investigation Report 94-4107, 44 p.

Westjohn, D.B. and T.L. Weaver, 1996b. Configuration of freshwater/saline-water interface and geological controls on distribution of freshwater in a regional aquifer system, central lower peninsula of Michigan. U.S. Geological Survey, Water Resources Investigation Report 94-4242, 44 p.

Westjohn, D.B. and T.L. Weaver, 1996c. Hydrogeologic framework of Mississippian rocks in the central lower peninsula of Michigan. U.S. Geological Survey, Water Resources Investigation Report 94-4246, 46 p.

Westjohn, D.B., T.L. Weaver, and K.F. Zacharias, 1994. Hydrogeology of Pleistocene glacial deposits and Jurassic "red beds" in the central lower peninsula of Michigan. U.S. Geological Survey, Water Resources Investigation Report 93-4152, 14 p.

Yurtsever, Y. and J.R. Gat, 1981. "Atmospheric waters," in Stable Isotope hydrology, deuterium and oxygen-18 in the water cycle. International Atomic Energy Agency Technical Report 210, p. 103-142.

Zacharias, K, 1992. Clay Mineral Diagenesis in the Marshall Sandstone, Michigan Basin. Unpublished masters thesis, Department of Geological Sciences, Michigan State University, East Lansing, Michigan. 95 p.