A One-Dimensional Model of Subsurface Hillslope Flow

Presented to:

US Forest Service Redwood Science Laboratory

December 1997

Submitted by:

Jason C. Fisher

ABSTRACT

A one-dimensional, finite difference model of saturated subsurface flow within a hillslope was developed. The model uses rainfall, elevation data, a hydraulic conductivity, and a storage coefficient to predict the saturated thickness in time and space. The model was tested against piezometric data collected in a swale located in the headwaters of the North Fork Caspar Creek Experimental Watershed near Fort Bragg, California. The model was limited in its ability to reproduce historical piezometric responses.

NOMENCLATURE

ρ	-	fluid density
B	-	dimensionless bedrock elevation
dx	-	spatial step size [m]
h	-	hydraulic head [m]
h_i	-	initial hydraulic head values [m]
H	-	dimensionless hydraulic head
i	-	spatial node subscript
j	-	temporal node subscript
J	-	mass flux of a fluid
K	-	hydraulic conductivity [m d^{-1}]
L	-	horizontal distance from the piping site to the top of the swale [m]
n	-	spatial distance from datum to bedrock [m]
N	-	accretion [in d^{-1}]
P	-	point in space
q	-	darcy velocity [m d^{-1}]
R	-	dimensionless accretion
RMSE	-	root mean squared error
S	-	storage coefficient
$\underline{S}_{0\underline{\varphi}}$	-	specific mass storativity related to potential changes
t	-	time [d]
tn	-	total number of spatial nodes
T	-	dimensionless time
W	-	approximation of H at the i, j node
x	-	spatial distance along the horizontal profile line [m]
X	-	dimensionless spatial distance along the horizontal profile line
y	-	horizontal spatial distance [m]
z	-	vertical spatial distance [m]

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

RATIONALE FOR PROJECT

From hydrologic year 1990 to the present, the U.S. Forest Service has monitored the subsurface hillslope flow of the E-road swale. The swale is located in the Caspar Creek watershed, Fort Bragg, CA. Monitoring has consisted of piezometric data from well sites located throughout the swale with samples being taken on average of every fifteen minutes. In hydrologic year 90 a logging road was built across the middle section of the hillslope followed by a total clear cut of the area during the following year. The development of such a road has resulted in a large build up of subsurface waters behind the road. The road behaves as a dam and road and slope stability are a major concern. Landslides occur during rainstorms when soil saturation reduces soil shear strength. Pore water pressure is the only stability variable that changes over a short time scale. Theory predicts that a slope can move from stability to instability as saturated thickness increases due to rainfall percolation. Previous studies of subsurface hillslope flow indicate that very little is understood about the subject. Further studies in this area will only help to improve existing techniques in the design and maintenance of mountain roads.

OBJECTIVE

The objective of this project is to better understand the hydrologic controls which govern the behavior of subsurface waters. A model will be developed which describes the groundwater system at E-road. An investigation on results of the model will be used to explain the mechanisms behind the historical observed behavior of the subsurface aquifer.

METHODOLOGY

The governing equation which will be used to model the (unconfined) subsurface flow is the Boussinesq equation, a nonlinear parabolic differential equation. This equation oversimplifies the E-road conditions, however, it is the foundation equation which can later be modified to better describe the system. The hydrologic system will be viewed as one dimensional and homogenous throughout the aquifer (hydraulic conductivity = constant). A FORTRAN-77 program will be written which utilizes a finite difference approximation of the governing equation and solves for a numeric solution using a predictor-corrector method. Advantages to the predictor-corrector method are its ability to give solutions of second order correctness while being algebraically explicit.

A completed model will then be tested with a simple hypothetical aquifer. The simulated results are then verified with a one-dimensional analytical solution. Once verification is complete the model will be applied to E-road field conditions. The model will be used to generate, with historical rainfall data, numerical simulations which can then be compared to observed piezometric data.

LITERATURE REVIEW

Field investigations of subsurface hillslope flow has shown that piezometric response is sensitive to rainfall, soil porosity, and topography. Swanston (1967) showed that there is a close relationship between rainfall and pore-water pressure development. As rainfall increases, pore-water pressure increases, rapidly at first, but at a decreasing rate as rainfall continues, reaching an upper limit determined by the thickness of the soil profile. Additional studies of shallow-soiled hillslopes during the wet seasons showed that there was little lag time between rainfall and piezometric response (Swanston, 1967; Haneberg and Gokce, 1992). Furthermore, Hanberg and Gokce (1992) showed that the rate of piezometric rise was dependent on available porosity and rainfall rate.

Based on field evidence, Whipkey (1965), Hewlett and Nutter (1970), and Weyman (1970) suggested that the presence of inhomogeneities in the soil may be a crucial factor in the generation of subsurface stormflow. These inhomogeneities may either be permeability breaks associated with soil horizons that allow shallow saturated conditions to build up. Harr (1977), Mosley (1979), Beven, (1980) suggest inhomogeneities may be the result of structural and biotic macropores in the soil that allow for very fast flow rates. With a significant portion of the total subsurface flow taking place in the macropores, a higher hydraulic conductivity will be perceived for the entire soil profile (Whipkey, 1965; Mosley, 1979). In addition field studies of subsurface stormflow have shown that the direct application of Darcian flow to subsurface flow in forested watersheds may not be realistic (Whipkey, 1965; Mosley, 1979).

The second approximation of Boussinesq (1904), also called the Dupuit-Forchheimer equation, will be used to describe the subsurface system. The equations development, based on Darcy's law and incorporating a non-horizontal bottom, was described by Bear (1972). Both analytical and numerical models have been developed for the nonlinear Boussinesq equation.

Bear (1972), Sloan and Moore (1984), and Buchanan (1990) have all developed analytical solutions to predict piezometric response for subsurface saturated flow on a one-dimensional uniform slope. While these models describe oversimplified groundwater systems quite well, the analytical solutions are unable to describe anything complex in nature (e.g. a system found in the environment). However, attempts have been made to further develop an analytical model which handles complex transient recharge in a sloping aquifer of finite width (Singh, 1991).

A numerical model developed by Hanberg and Gokce (1992), modeled the full, one dimensional Dupuit-Forchheimer equation for a hillside with changing slope angle and transient rainfall. The predicted response rose with the historical observed response but receded more quickly. They hypothesized that seepage out of the bedrock lengthened the observed recession.

Reddi (1990) numerically modeled saturated subsurface flow over two horizontal directions. In the overall down slope direction the Dupuit-Forchheimer approach was used. Flow in the transverse direction was assumed horizontal and was not topographically driven. Their predictions differed significantly from field observations in timing and magnitude.

Jackson and Cudy (1992) additionaly modeled over the two horizontal directions utilizing a finite difference model for hillslope flow. The advancement of this particular model over the two previous numerical models was its ability to deal with complex topography (e.g. topographic holes). They found that peak piezometric response was largely dependent on rainfall rate and storage coefficient while the recession curve was influenced mainly by the hydraulic conductivity. The model was tested against piezometric response measured in a hillslope hollow and showed promising results.

Brown (1995), in a two-dimensional model, incorporated both infiltration partitioning and overland flow into the hillslope flow model. Model simulations using available field data were compared to field observations of rainfall-pore pressure responses and were found to be in reasonable agreement.

Two techniques were looked at for solving the Boussinesq equation. The first technique, utilized by Chu and Willis (1992), reduced the nonlinear equation to a linear partial differential equation and used a forward in time, central in space explicit finite difference method for their simulation model. The second, more robust technique involved a predictor-corrector approximation of the solution (Douglas and Jones, 1963) which Remson (1971) presented for the one dimensional nonlinear Boussinesq equation.

STUDY AREA

The E-Road swale lies in the headwaters of the North Fork Caspar Creek Experimental Watershed, in the Jackson Demonstration State Forest near Fort Bragg, California, USA. The topography is youthful, consisting of uplifted marine terraces that date to the late Tertiary and Quaternary periods (Kilbourne, 1986). The swale occupies an area of 0.9895 acres. Hillslope ranges from 3 to 35% with an average of 19% . Elevations within the study area range from 95 m to 128 m.

The groundwater study first began in hydrologic year 1990 and involved the monitoring of hydraulic heads at specified peziometer sites (R1P2, R2P2, R3P2, R4P2, R5P2, R6P2) over time. During the summer of 1990 the construction of a logging road across the swale took place. In the summer of 1991 the swale was clear-cut utilizing a cable yarding technique. Figure 1 depicts the swale pre road building while figure 2 depicts the swale post road building.

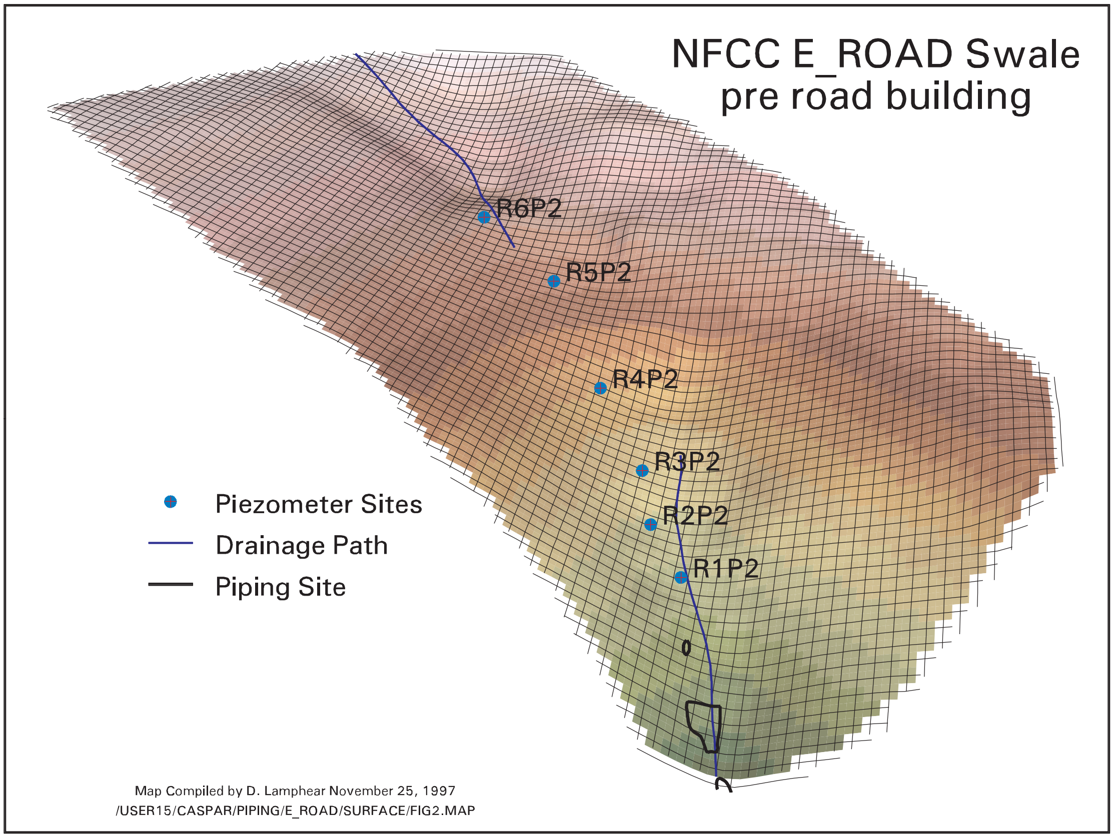

Figure 1: E-Road swale pre road building.

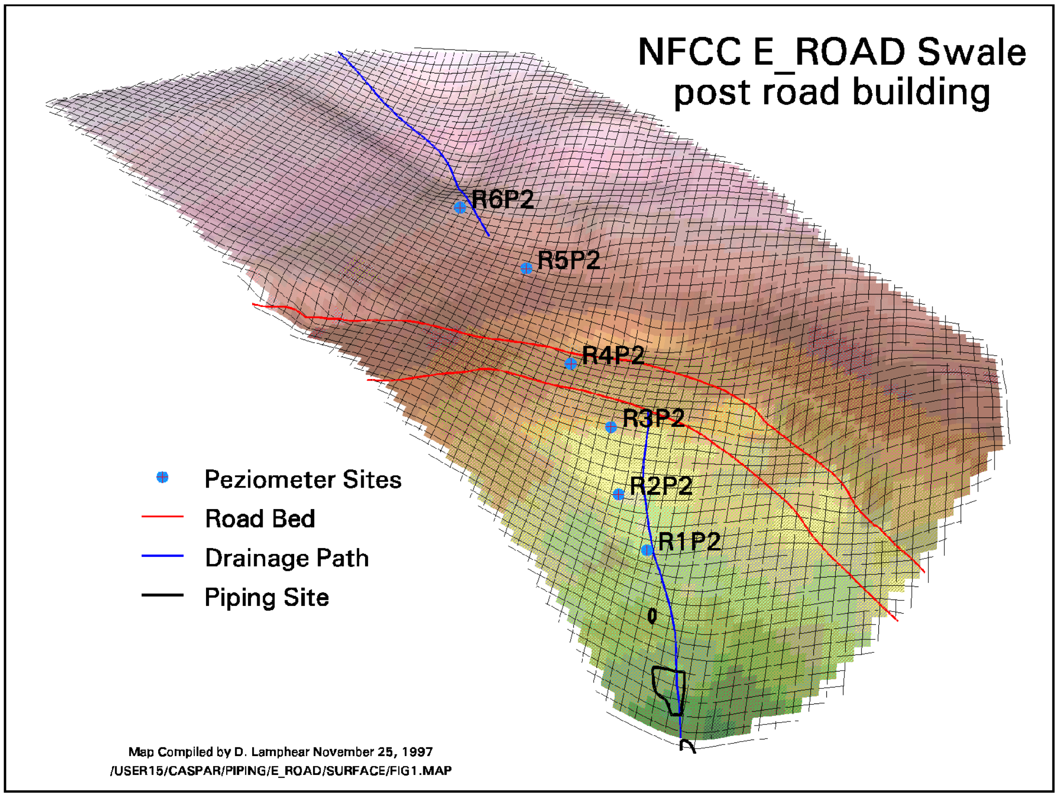

FIGURE 2: E-Road swale post road building

The soil series within the E-Road swale is Vandamme, a clayey, vermiculitic, isomesic typic tropudult, derived from sedimentary rocks, primarily Franciscan greywacke sandstone. Textures of both surface soils and subsoil are loam and clay loam respectively. The permeability within the soil is considered moderately slow (Huff et al., 1985).

Precipitation within the study area is characterized by low-intensity rainfall, prolonged cloudy periods in winter, and relatively dry summers with cool coastal fog. Between October and April 90% of precipitation takes place with a mean annual precipitation of approximately 1190 mm. Temperatures within the swale rage from a maximum of 25 °C in summer to freezing in winter.

MODEL FORMULATION AND DEVELOPMENT

The following steps were used in the development of the ground-water model. These steps follow a commonly used order as outlined in Fetter (1994).

1. Determine purpose of model.
2. Gather data and create a conceptual model of the system.
3. Develop the computer code that will be used.
4. Prepare model design. The model discritization, boundary conditions, and initial conditions are selected.
5. Validate the model by comparing model output to a known head distribution.
6. Predict head distribution with applied stresses.
7. Perform sensitivity analysis on head distribution.
8. Present results.

The first three steps have already been presented. The remaining steps are outlined below.

MODEL DESIGN

The construction of a one-dimensional model first required that the swale system be simplified. In figure 3 a profile line was established through the center of the swale connecting each of the piezometer sites.

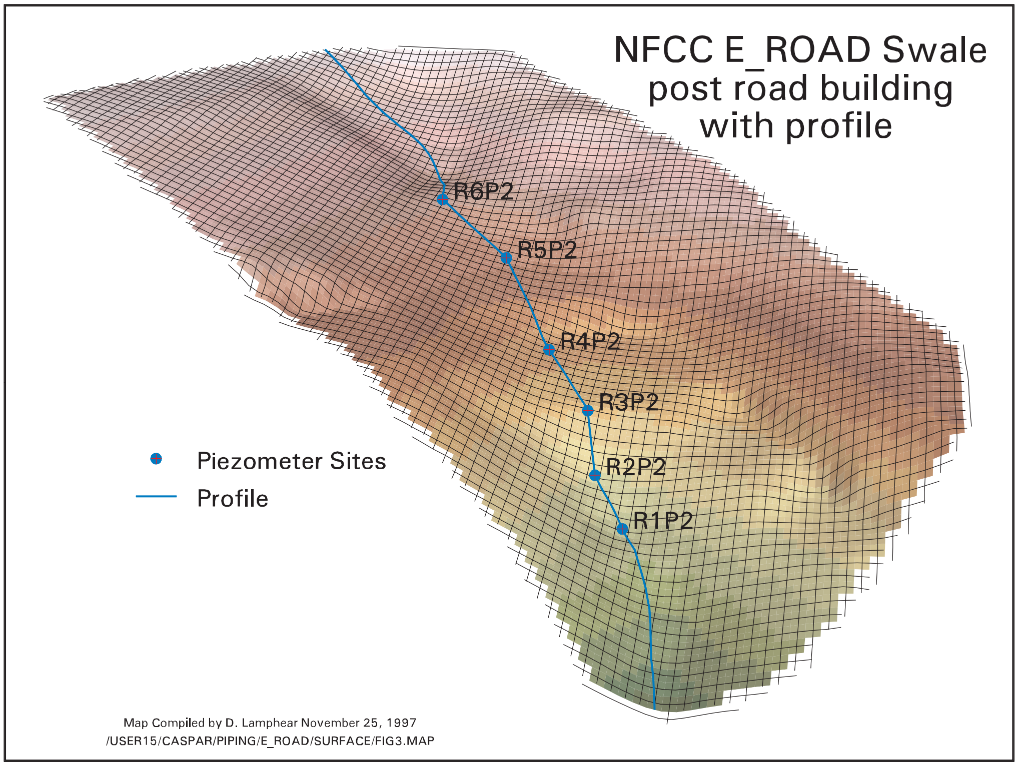

FIGURE 3: E-Road swale post road building

The profile line then allowed for the construction of a side view representation of swale system (Figure 4). With the exception of the bedrock, the spatial locations for each of the side view components (e.g. ground surface, piezometer sites, etc.) were known. With little known about the bedrock it was necessary to construct a bedrock profile which best represented E-Road conditions. A discussion on the establishment of the bedrock profile will be presented later in the report.

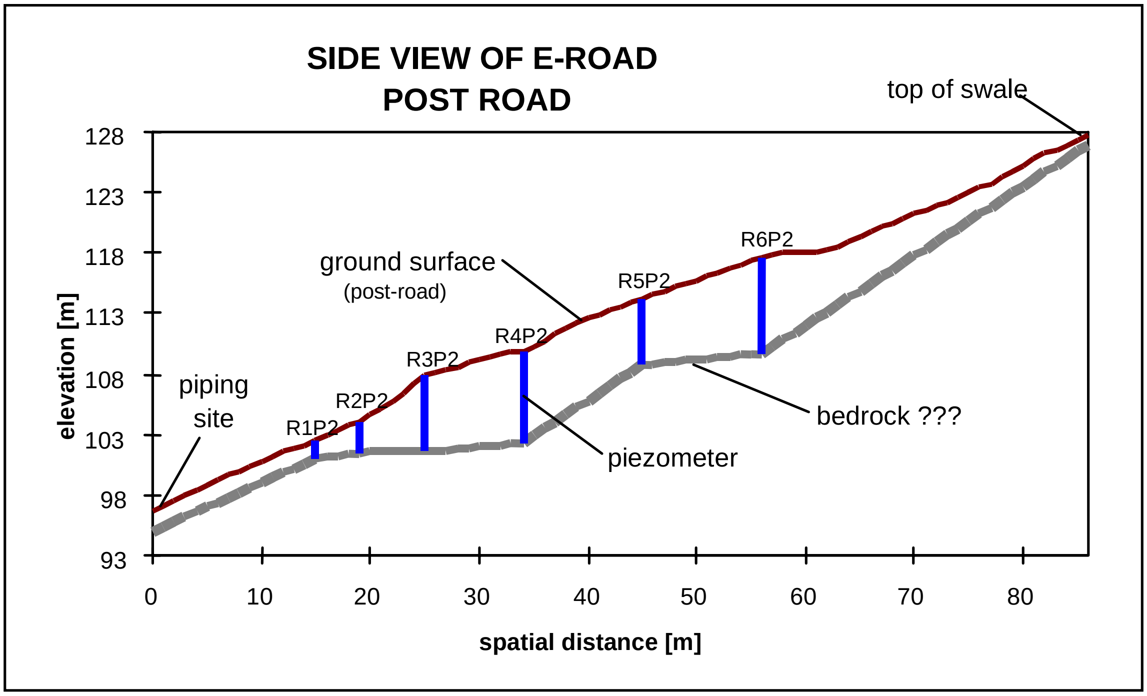

FIGURE 4: Side view of E-Road swale post road building

Graphical information concerning the development of the model can be found in figure 5. The unconfined aquifer system was modeled in one dimension with a lower Dirchlet boundary condition (h_0 = constant) at the piping site and a Neumann boundary condition (h/ x_L = 0) at the top of the swale. An impervious bedrock bottom contains the system in the soil region located between the bedrock and ground surface. Note that an upper limit for the simulated hydraulic heads was placed at the ground surface. That portion of the hydraulic head which exceeds the ground surface will be considered runoff, a hydrologic process which was not accounted for in the model.

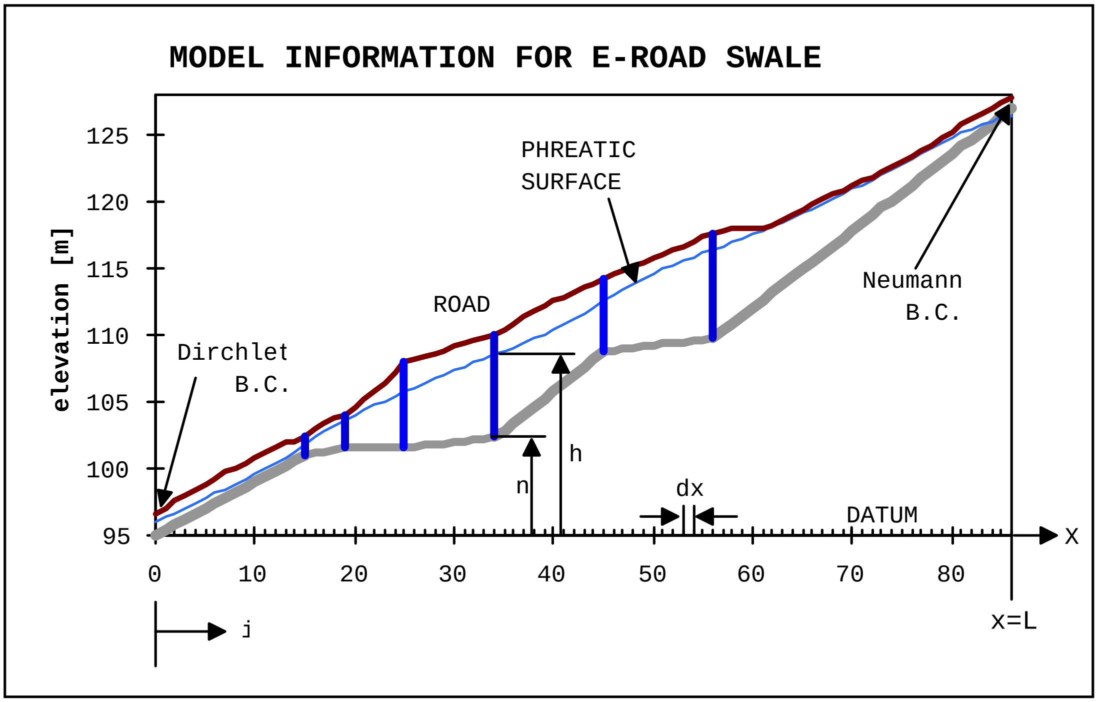

FIGURE 5: Model information for E-Road swale

where:

dx	is a spatial step size (uniform)
h(x,t)	is the spatial distance from datum to phreatic surface
i	is a spatial node subscript
L	is the horizontal distance from the piping site to the top of the swale
n(x)	is the spatial distance from datum to bedrock
x	is the spatial coordinate scheme.

The governing equation for unconfined subsurface flow assuming the aquifer is homogeneous and isotropic is the Boussinesq equation. The development of the Boussinesq equation described by Bear (1972) was achieved by incorporating both the Dupuit approximation and the continuity equation.

DUPUIT APPROXIMATION

Dupuit assumptions:

1. equipotential surfaces are vertical
2. flow essentially horizontal
3. the slope of the water table at some point on it represents the constant hydraulic gradient along a vertical line passing through this point.

The Dupuit assumptions lead to the specific discharge for an isotropic medium as:

$$q_x = -K\left(\frac{\partial h}{\partial x}\right); \quad q_y = -K\left(\frac{\partial h}{\partial y}\right); \quad h = h(x,y,t)$$

where :
q is the specific discharge, darcy velocity
K is the hydraulic conductivity (1)
h is the hydraulic head
x is a spatial location
t is time

CONTINUITY EQUATION

Consider a non-deforming control volume of dimensions δx, δy, δz, parallel, respectively, to the x, y, z coordinates around a point P(x, y, z) in the porous medium domain. The control volume is bounded by a horizontal impervious bottom and the phreatic surface above. Let a vector $\underline{J}$, with components in the x, y, z directions, denote the mass flux of a fluid of density ρ. Over a time interval of δt, the excess of inflow over outflow through the surfaces of the control volume, located at x-δx/2, x+δx/2, y-δy/2, y+δy/2, is expressed as a difference for x and y terms:

$$\left[\underline{J}_x\big|_{x-(\delta x/2),y,z} - \underline{J}_x\big|_{x+(\delta x/2),y,z}\right]\delta x\delta y\delta t + \left[\underline{J}_y\big|_{x,y-(\delta y/2),z} - \underline{J}_x\big|_{x,y+(\delta y/2),z}\right]\delta x\delta y\delta t \qquad (2)$$

By developing in a Taylor series around P and neglecting terms of second order and higher gives:

$$-\left(\frac{\partial \underline{J}_x}{\partial x} + \frac{\partial \underline{J}_y}{\partial y}\right)\delta x\delta y\delta t \qquad (3)$$

In the z direction, only inflow from accretion takes place

$$\rho N(x,y,t)\delta x\,\delta y\delta t \qquad (4)$$

A positive downward N means accretion while a negative upward N means evaporation and/or transpiration . Summing masses in both horizontal (3) and vertical directions (4) gives:

$$-\left(\frac{\partial \underline{J}_x}{\partial x}+\frac{\partial \underline{J}_y}{\partial y}\right)\delta x\delta y\delta t+\rho N(x,y,t)\delta x\delta y\,\delta t \qquad (5)$$

If $\underline{J}=\rho \underline{q}$ then:

$$-\left(\frac{\partial\left(\rho \underline{q}_x\right)}{\partial x}+\frac{\partial\left(\rho \underline{q}_y\right)}{\partial y}\right)\delta x\delta y\delta t+\rho N(x,y,t)\delta x\delta y\delta t \qquad (6)$$

By the principle of mass conservation, this must be equal to the change of mass within the control volume during δt. The change of mass is accounted for in two ways. First, the phreatic surface rises by a vertical distance δh so that a volume of porous medium becomes saturated.

$$\rho S\delta x\delta y\left[h|_{t+\Delta t}-h|_t\right] \qquad (7)$$

where : S is the storage coefficient

Second, the pressure everywhere in the water-saturated control volume hδxδy rises.

$$h\underline{S}_{0\varphi}\left(\frac{\partial h}{\partial t}\right) \qquad (8)$$

where : $\underline{S}_{0\varphi}$ is the specific mass storativity related to potential changes

Therefor, total change in mass is the addition of equations (7) and (8):

$$\rho S\delta x\delta y\left(\frac{\partial h}{\partial t}\right)+h\underline{S}_{0\varphi}\left(\frac{\partial h}{\partial t}\right) \qquad (10)$$

However, since S >> $\underline{S}_{0\varphi}$, total change in mass during a δt time interval is :

$$\rho S\delta x\delta y\left(\frac{\partial h}{\partial t}\right)\delta t \qquad (11)$$

Setting equations (6) and (11) equal to each other, the entire continuity equation can be written as:

$$-\left(\frac{\partial(\rho \underline{q}_x)}{\partial x}+\frac{\partial(\rho \underline{q}_y)}{\partial y}\right)\delta x\delta y\delta t+\rho N(x,y,t)\delta x\delta y\delta t=\rho S\delta x\delta y\left(\frac{\partial h}{\partial t}\right)\delta t \qquad (12)$$

Dividing through by δx, δy, δt, and ρ gives:

$$-\frac{\partial(\underline{q}_x)}{\partial x}-\frac{\partial(\underline{q}_y)}{\partial y}+N=S\left(\frac{\partial h}{\partial t}\right) \qquad (13)$$

Introducing Dupuit's assumptions (1) for q_x and q_y gives:

$$-\frac{\partial}{\partial x}\left[-Kh\frac{\partial h}{\partial x}\right]-\frac{\partial}{\partial y}\left[-Kh\frac{\partial h}{\partial y}\right]+N=S\left(\frac{\partial h}{\partial t}\right)$$

or (14)

$$\frac{\partial}{\partial x}\left[Kh\frac{\partial h}{\partial x}\right]+\frac{\partial}{\partial y}\left[Kh\frac{\partial h}{\partial y}\right]+N=S\left(\frac{\partial h}{\partial t}\right)$$

which is known as Boussinesq's equation for unsteady flow in a phreatic aquifer with accretion. For a homogeneous aquifer, K = constant:

$$K\left[\frac{\partial}{\partial x}\left(h\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(h\frac{\partial h}{\partial y}\right)\right]+N=S\frac{\partial h}{\partial t} \qquad (15)$$

For a non-horizontal bottom, see figure 2, let h(x, y, t) represent the elevation of the phreatic surface and n(x, y) denote the elevation of the aquifer's bottom, both with respect to a datum. The Boussinesq equation now becomes:

$$K\left[\frac{\partial}{\partial x}\left((h-n)\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left((h-n)\frac{\partial h}{\partial y}\right)\right]+N=S\frac{\partial h}{\partial t}$$

or (16)

$$K\left[\frac{\partial}{\partial x}\left(h\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(h\frac{\partial h}{\partial y}\right)-\frac{\partial}{\partial x}\left(n\frac{\partial h}{\partial x}\right)-\frac{\partial}{\partial y}\left(n\frac{\partial h}{\partial y}\right)\right]+N=S\frac{\partial h}{\partial t}$$

Averaging the system over the y-direction with a unit depth allows for the one dimensional form of the governing equation:

$$K\left[\frac{\partial}{\partial x}\left(h\frac{\partial h}{\partial x}\right)-\frac{\partial}{\partial x}\left(n\frac{\partial h}{\partial x}\right)\right]+N=S\frac{\partial h}{\partial t} \qquad (17)$$

Dividing through by the hydraulic conductivity, K, gives:

$$\frac{S}{K}\frac{\partial h}{\partial t} = \frac{\partial}{\partial x}\left(h\frac{\partial h}{\partial x}\right) - \frac{\partial}{\partial x}\left(n\frac{\partial h}{\partial x}\right) + \frac{N}{K} \tag{18}$$

Equation (18) is the final form of the governing equation which was used to develop the mathematical model used to describe unconfined subsurface hillslope flow. The initial and boundary conditions are described as follows:

initial conditions :

$$h(x,0) = f(x) = h_0(x)$$

boundary conditions :

$$h(0,t) = h$$

$$\left.\frac{\partial h}{\partial x}\right|_{x=\mathrm{L}} = 0$$

In order to solve equation (18), a non-linear partial differential equation, a finite difference approximation was made and solved utilizing a predictor-corrector method.

PREDICTOR-CORRECTOR

The predictor-corrector method, as outlined by Remson (1971), first requires that the governing equation (18) be transformed into its non-dimensional form:

Transformations :

$$H = \frac{h}{\mathrm{L}};\quad B = \frac{n}{\mathrm{L}};\quad X = \frac{x}{\mathrm{L}};\quad T = \frac{Kt}{S\mathrm{L}};\quad R = \frac{N}{K}$$

Governing Equation :

$$\frac{\partial H}{\partial T} = \frac{\partial}{\partial X}\left[H\frac{\partial H}{\partial X}\right] - \frac{\partial}{\partial X}\left[B\frac{\partial H}{\partial X}\right] + R$$

initial conditions : (19)

$$H(X,0) = H_0(X)$$

boundary conditions :

$$H(0,T) = H$$

$$\left.\frac{\partial H}{\partial X}\right|_{X=1} = 0$$

Expanding using the chain rule and solving for $\left(\partial^2 H / \partial X^2\right)$ gives:

$$\frac{\partial^2 H}{\partial X^2} = \frac{1}{(H-B)}\left[\frac{\partial H}{\partial T} - \left(\frac{\partial H}{\partial X}\right)^2 + \left(\frac{\partial B}{\partial X}\right)\left(\frac{\partial H}{\partial X}\right) - R\right] \tag{20}$$

With the governing equation now in its non-dimensional form the predictor-corrector scheme may be implemented. The predictor solves for a half time step ahead while the corrector, utilizing the predictor solution, solves for a full time step ahead.

PREDICTOR: Advanced the solution to the end of the time step.

$$\frac{d^2\left(H_{i,j+0.5}\right)}{dX^2} = \frac{1}{H_{i,j} - B_i}\left[\left(\frac{dH_{i,j}}{dT}\right) - \left(\frac{dH_{i,j}}{dX}\right)^2 + \left(\frac{dB_i}{dX}\right)\left(\frac{dH_{i,j}}{dX}\right) - R\right] \tag{21}$$

Note that i and j represent spatial and temporal subscripts, respectivily, and that the approximation of H at the i, j node is represented as $W_{i,j}$. Substituting the following finite difference approximations into equation (21),

$$\frac{d^2\left(H_{i,j+0.5}\right)}{dX^2} \cong \frac{W_{i+1,j+0.5} - 2W_{i,j+0.5} + W_{i-1,j+0.5}}{(\Delta X)^2} \quad \Rightarrow \text{ central difference approx.}$$

$$\frac{dH_{i,j}}{dT} \cong \frac{W_{i+1,j+0.5} - W_{i,j}}{\Delta T/2} \quad \Rightarrow \text{ forward difference in time approx.}$$

$$\frac{dH_{i,j}}{dX} \cong \frac{W_{i,j} - W_{i-1,j}}{\Delta X} \quad \Rightarrow \text{ backward difference approx.}$$

$$\frac{dB_{i,j}}{dX} \cong \frac{B_i - B_{i-1}}{\Delta X} \quad \Rightarrow \text{ backward difference approx.}$$

gives the predictor as:

$$W_{i+1,j+0.5} - \left[2 + \frac{2(\Delta X)^2}{\Delta T\left(W_{i,j} - B_i\right)}\right]W_{i,j+0.5} + W_{i-1,j+0.5} = \frac{-2(\Delta X)^2 W_{i,j}}{\Delta T\left(W_{i,j} - B_i\right)} \quad \bullet\bullet\bullet$$

$$\bullet\bullet\bullet \quad + \frac{\left(W_{i,j} - W_{i-1,j}\right)}{\left(W_{i,j} - B_i\right)}\left[W_{i-1,j} - W_{i,j} + B_i - B_{i-1}\right] - \frac{(\Delta X)^2 R}{\left(W_{i,j} - B_i\right)} \tag{22}$$

CORRECTOR

$$\frac{d^2\left(H_{i,j+1}-H_{i,j}\right)}{2dX^2}=\frac{1}{H_{i,j+0.5}-B_i}\left[\left(\frac{dH_{i,j+0.5}}{dT}\right)-\left(\frac{dH_{i,j+0.5}}{dX}\right)^2+\left(\frac{dB_i}{dX}\right)\left(\frac{dH_{i,j+0.5}}{dX}\right)-R\right] \quad (23)$$

Introducing the following finite difference approximations into equation (23),

$$\frac{d^2\left(H_{i,j}\right)}{dX^2}\cong\frac{W_{i+1,j}-2W_{i,j}+W_{i-1,j}}{(\Delta X)^2} \quad \Rightarrow \text{ central difference approx.}$$

$$\frac{d^2\left(H_{i,j+1}\right)}{dX^2}\cong\frac{W_{i+1,j+1}-2W_{i,j+1}+W_{i-1,j+1}}{(\Delta X)^2} \quad \Rightarrow \text{ central difference approx.}$$

$$\frac{dH_{i,j}}{dT}\cong\frac{W_{i+1,j+1}-W_{i,j}}{\Delta T} \quad \Rightarrow \text{ forward difference in time approx.}$$

$$\frac{dH_{i,j+0.5}}{dX}\cong\frac{W_{i,j+0.5}-W_{i-1,j+0.5}}{\Delta X} \quad \Rightarrow \text{ backward difference approx.}$$

$$\frac{dB_i}{dX}\cong\frac{B_i-B_{i-1}}{\Delta X} \quad \Rightarrow \text{ backward difference approx.}$$

yields the corrector equation as:

$$W_{i+1,j+1}-\left[2+\frac{2(\Delta X)^2}{\left(W_{i,j+0.5}-B_i\right)\Delta T}\right]W_{i,j+1}+W_{i-1,j+1}=\frac{-2(\Delta X)^2W_{i,j}}{\left(W_{i,j+0.5}-B_i\right)\Delta T} \quad \bullet\bullet\bullet$$

$$\bullet\bullet\bullet \quad +\frac{2\left(W_{i,j+0.5}-W_{i-1,j+0.5}\right)}{\left(W_{i,j+0.5}-B_i\right)}\left[-W_{i,j+0.5}+W_{i-1,j+0.5}+B_i-B_{i-1}\right] \quad \bullet\bullet\bullet \qquad (24)$$

$$\bullet\bullet\bullet \quad -\frac{2(\Delta X)^2R}{\left(W_{i,j+0.5}-B_i\right)}-\left(W_{i+1,j}-2W_{i,j}+W_{i-1,j}\right)$$

The governing equation (18) is now reduced to two systems of algebraic equations. Computer code which simultaneously solves each of the systems algebraic equations has been developed and may be found in appendix A.

PHASES OF MODEL DEVELOPMENT

A three phase process was incorporated in order to develop a working model which describes the E-road system. Associated with each phase is a more complex numerical model. The three phases are described below.

PHASE I

The first phase, shown in figure 6, consists of a homogenous isotropic groundwater system. In the vertical direction the system is contained below by an impervious horizontal bedrock layer. The ground surface may be viewed as an infinite distance from the bedrock allowing water to propagate upward through the soil with no boundary limitations. In the horizontal direction, Dirchlet boundary conditions (h_0 = constant and h_L = constant) contain the system.

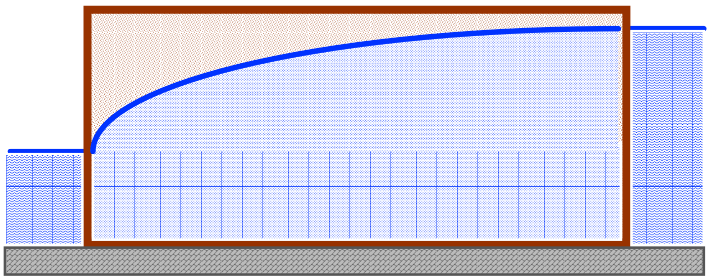

FIGURE 6: Side view of Phase I

PHASE II

The second phase, shown in figure 7, consists of a system identical to Phase I with the following exceptions. Where Phase I viewed the right hand horizontal boundary condition as a Dirchlet B.C. (h_L = constant), Phase II views the boundary as a Neumann B.C. ($\Delta h_L/\Delta t = 0$). The hydrologic process of accretion was accounted for in the Phase II model.

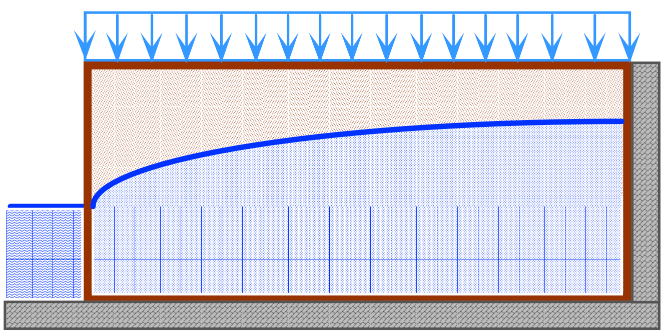

FIGURE 7: Side view of Phase II

PHASE III

In the third phase, shown in figure 8, the topography of the E-Road ground surface was identified by the model and simulated hydraulic heads were not allowed to exceed ground surface elevations. Those hydraulic heads which would theoretically breach the surface were considered surface runoff. A limitation for the model was its inability to then allow the established runoff to infiltrate back into the groundwater. The model was additionally modified to incorporate a non-horizontal bedrock bottom. Specifically the bedrock configuration for the E-Road swale. Due to the difficult nature of defining bedrock elevations, a method of interpolation was used. With known bedrock elevations identified during the boring of the piezometric holes, a linear interpolation was used to fabricate bedrock elevations between these known elevations. To complete the interpolations at the boundaries of the system guesses were made as to where the bedrock elevations were located.

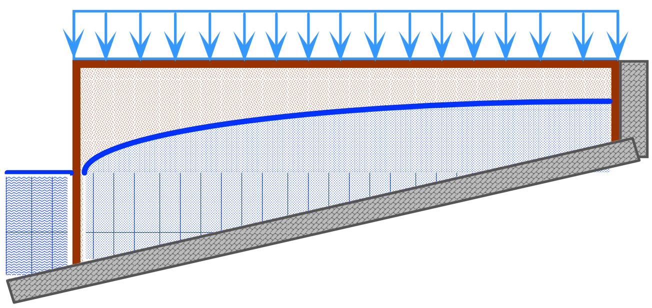

FIGURE 8: Side view of Phase III

VALIDATION

MODEL COMPARISON

As a check of the finite difference approximation for Phase II, the problem was solved and compared with numerical results found by Yeh and Tauxe (1971). Yeh and Tauxe had previously validated their results with published experimental data. Initial and boundary conditions are as follows:

$$h(x, 0) = 1.0$$

$$h(0, T) = \frac{h_{1,0}}{2} = 0.5 \tag{25}$$

$$\left.\frac{\partial h(x, T)}{\partial x}\right|_{x=L=1.0} = 0$$

Figure 9 shows the comparison of water table profiles for different dimensionless times, T. The solid lines represent results found with the Phase II model while the circles represent solutions found by Yeh and Tauxe (1971). Note that the comparisons are excellent.

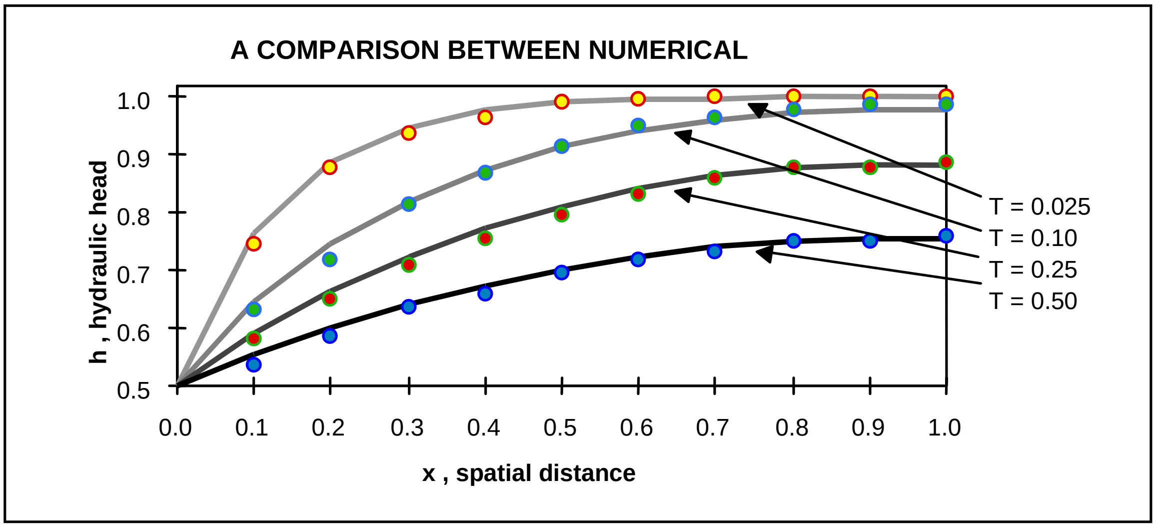

Figure 9: Comparisons between phase II solutions, shown above as solid lines, and those numerical solutions found by Yeh and Tauxe [1971], shown above as circles.

STEADY STATE

To further validate the model an experiment was performed for Phase II conditions to reproduce a known hydraulic phenomenon. Water will always flow from a higher potential to a lower potential. Setting up the model with boundary conditions at $h_{0,t}$ = 10 m and $h_{L,0}$ = 15 m we would expect to see the higher potential hydraulic head at x = L, $h_{L,0}$, to drop over time until it eventually reaches the lower potential hydraulic head at x = 0, $h_{0,t}$. At that time, when the two hydraulic heads are equal ($h_{0,t} = h_{L,0}$), the system will have achieved stability and remain constant throughout time. Hydraulic heads of equal potential imply a stagnate system. The models response, shown in figure 10, behaves as expected reaching steady state after approximately 50 days.

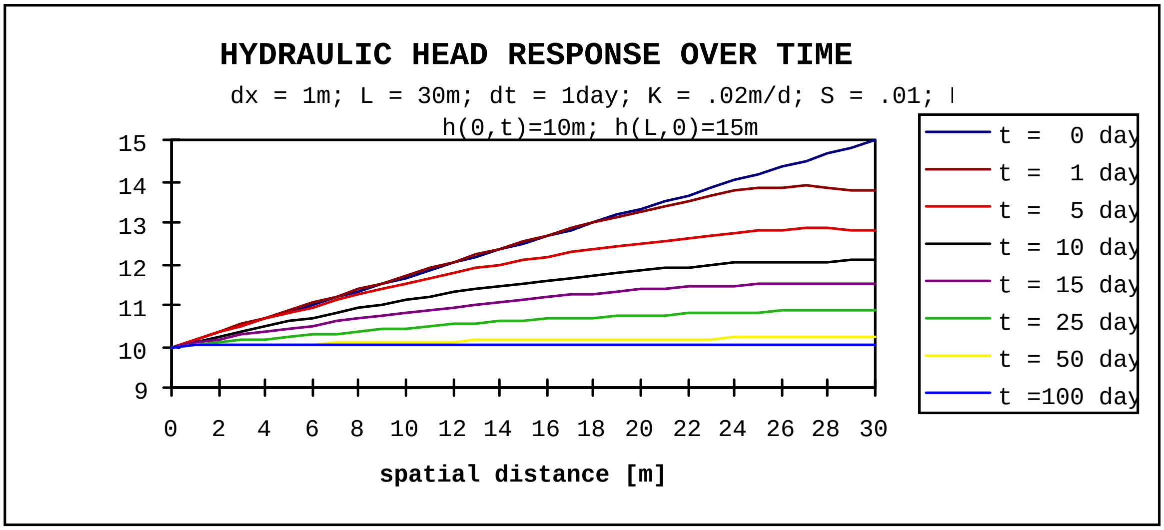

FIGURE 10: Hydraulic head response over time with Phase II conditions.

MODEL RESULTS

HISTORICAL AND SIMULATED

A comparison between historical and simulated results can be found in figure 11. The comparison was made by observing over time both the historical and predicted hydraulic head response at the R4P2 piezometric hole. The time period chosen for observation was between 1/1/95 and 2/15/95. Table 1 gives the model variables used in the simulation.

VARIABLE	VALUE
dx	1 m
dt	1 day
K	0.025 m/d
S	0.01
n(0)	94 m
n(L)	118 m
h(0,t)	96 m
h(L,0)	127 m

TABLE 1: Model variables

A piezometer response was simulated by inputting into the model historical rainfall rates associated with the time period of the simulation. These rainfall rates were not the actual recorded rainfall rates for the area; the precipitation was scaled to reflect water loss due to piping, evapotranspiration, and soil moisture storage. The initial groundwater conditions represent the water levels of the 100 day simulation period.

A large difference was found between historical and simulated results. While their general response to rainfall was similar, the model consistently over predicted hydraulic heads by approximately 5.5 m. This over prediction of hydraulic head was believed to have resulted from an inaccurate representation of the bedrock within the swale. The 1-d model represents the bedrock as a distinct boundary to the system when in reality the bedrock can not so easily be defined. The actual bedrock layer consists of shale pieces which become increasingly more dense as depth increases. Those bedrock elevations defined through the boring of the piezometric holes are believed to have been within the upper boundary of the actual bedrock layer. A large portion of the permeable media would then be unaccounted for by the model resulting in the over prediction of the hydraulic head.

An additional difference between historical data and model predictions was found for the drainage of the swale. During times of little or no rainfall those hydraulic heads simulated by the model would subside more quickly when compared to historical subsidence levels. The increase in subsidence rate was believed to have resulted from the assumption of homogeneity within the aquifer system.

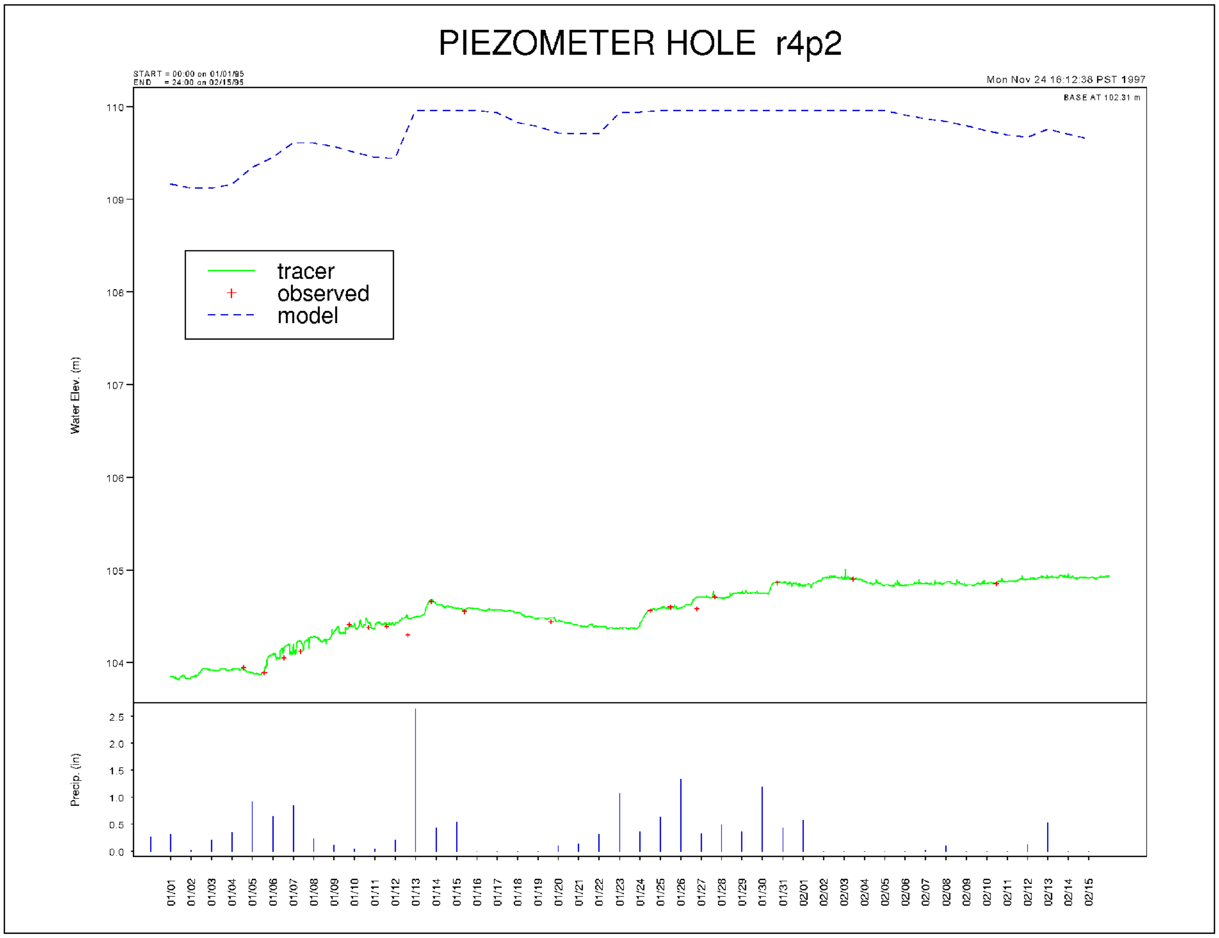

FIGURE 11: Historical data vs. model simulation for the R4P2 site.

SENSITIVITY

Both the simulation approach and an evaluation of the root mean squared error (RMSE) were used to describe the sensitivity of the models variables. The following equation describes the RMSE.

$$RMSE = \sqrt{\frac{\sum_{j=1}^{tn} \left\{ (h - h_i)^2 \right\}_j}{tn}} \tag{26}$$

where:

h is the head at a node after some parameter variation (m)

h_i is the initial head values (m)

tn is the total number of nodes

RMSE is the root mean squared error value

PHASE I SENSITIVITY

A sensitivity analysis for Phase I was performed on dx, dt, K, and S utilizing the RMSE approach. The base case simulation which all other simulations were compared can be found in table 2.

VARIABLE	VALUE
dx	0.1 m
L	10 m
dt	1 day
t	20 days
K	0.02 m/day
S	0.01
h(0, t)	2 m
h(L, t)	5 m

TABLE 2: Phase I model variables

Figure 12 shows the RMSE versus the spatial step size, dx. It was inherent in the numerical approximation of hydraulic heads that the smaller the spatial step size the greater the accuracy within the solution. A linear relationship was observed between spatial step size and RMSE.

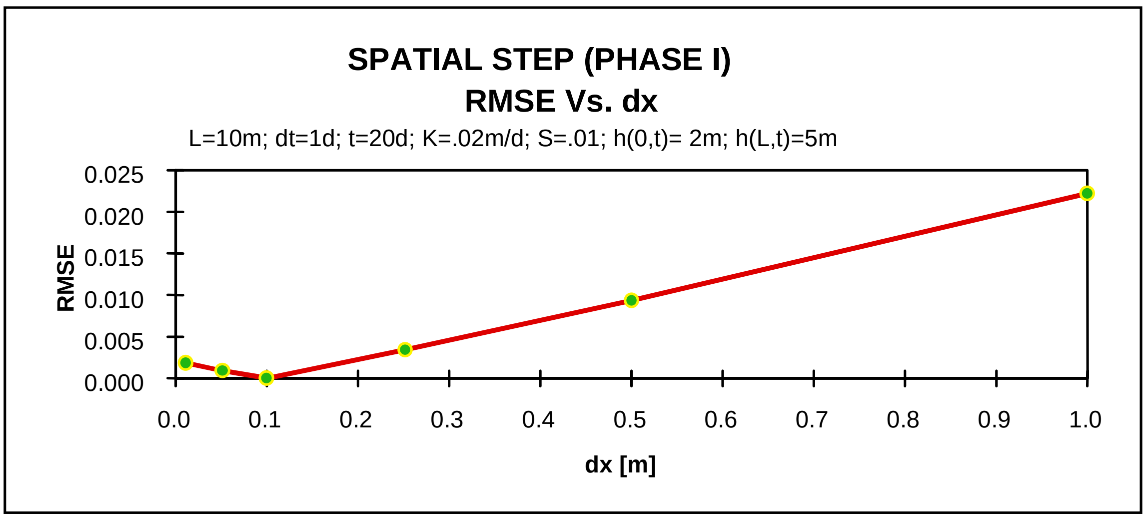

FIGURE 12: Sensitivity analysis for dx under Phase I conditions.

Figure 13 shows the RMSE versus the temporal step size, dt. Smaller temporal step sizes resulted in greater accuracy within the solution. As Δt increased past 1.5 m the RMSE values exponentially grew in size indicating a loss of accuracy within the solution.

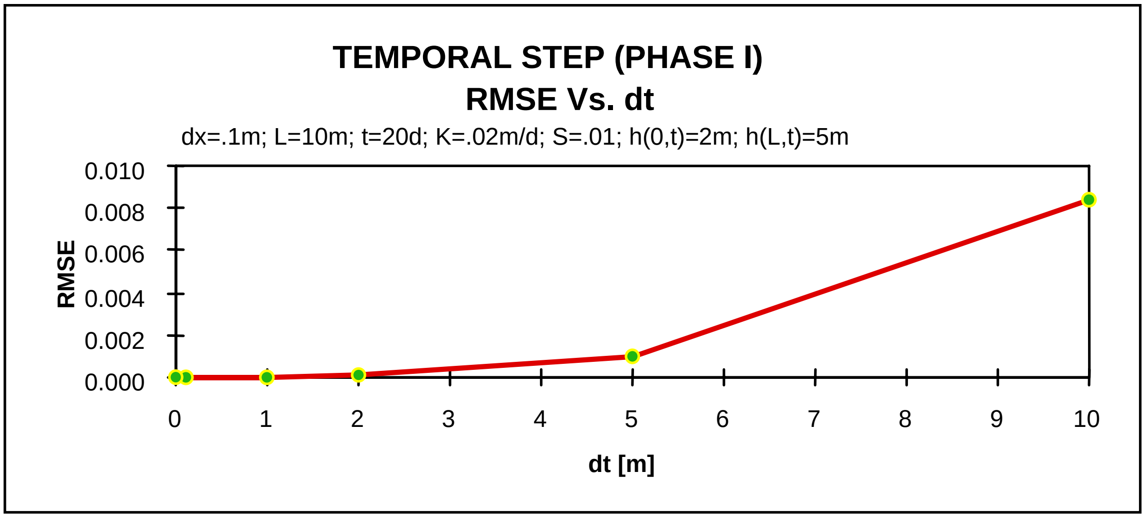

FIGURE 13: Sensitivity analysis for dt under Phase I conditions.

The hydraulic conductivity, K, is a function of both fluid and medium properties. Figure 14, demonstrates how changes in K effect the solution of the Phase I model. The sensitivity analysis shows that magnitudes of K between 0.002 and 0.08 m/day resulted in little to no change within the solution. However, K values less than 0.002 m/day resulted in extremely rapid changes in RMSE while K values greater than 0.08 m/day showed a gradual increase in RMSE.

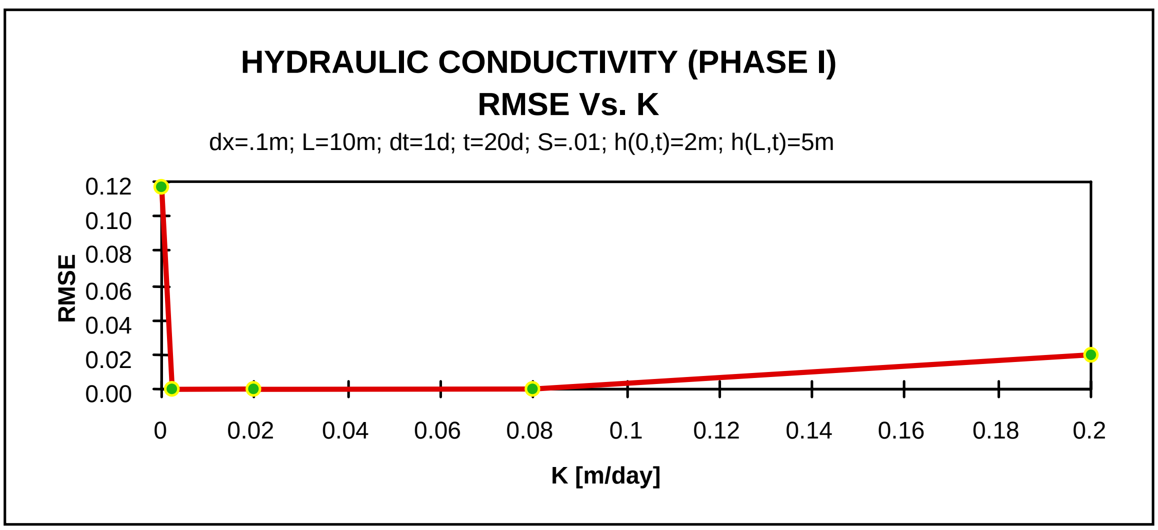

FIGURE 14: Sensitivity analysis for hydraulic conductivity, K, under Phase I conditions.

The storage coefficient, S, is defined as the volume of water released or taken into storage per unit cross-sectional area per unit change in the hydraulic head. Results from the sensitivity analysis, shown in figure 15, indicate that there will be no change within the solution for S values between 0.005 and 0.05.

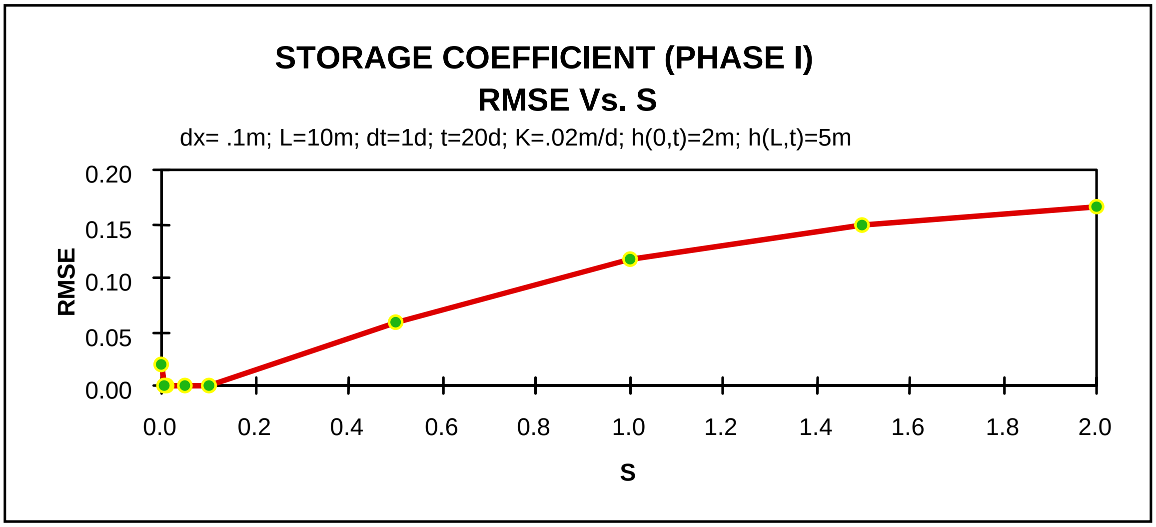

FIGURE 15: Sensitivity analysis for the storage coefficient, S, under Phase I conditions.

PHASE II SENSITIVITY

A sensitivity analysis for Phase II was performed on dx, dt, K, S, and N. Both dx and dt utilized the RMSE while K, S, and N relied upon a simulation approach for the sensitivity analysis. For the RMSE analysis the base case simulation can be found in table 3.

VARIABLE	VALUE
dx	0.1 m
L	10 m
dt	1 day
t	20 days
K	0.02 m/day
S	0.01
N	0 in/day
h(0, t)	2 m
h(L, 0)	5 m

TABLE 3: Phase II model variables

Manipulations of Δx within the Phase II model are shown in figure 16. It was inherit in the model that the smaller the spatial step size the greater the accuracy within the solution. A comparison between Phase I (Figure 12) and Phase II sensitivity results for Δx shows that RMSE values are slightly greater for Phase II results. Additionally, the linearity between Δx and RMSE found in the Phase I model was not seen in the Phase II model. The absence of linearity was the result of the Neumann boundary condition found in the Phase II model. In order to deal with the no-flux boundary condition the model views h_L equal to h_{L-dx} for all time steps. Setting the two hydraulic heads equal results in an increase in the deviation between solutions of differing spatial steps as well as a limiting effect on the RMSE as Δx increases in size.

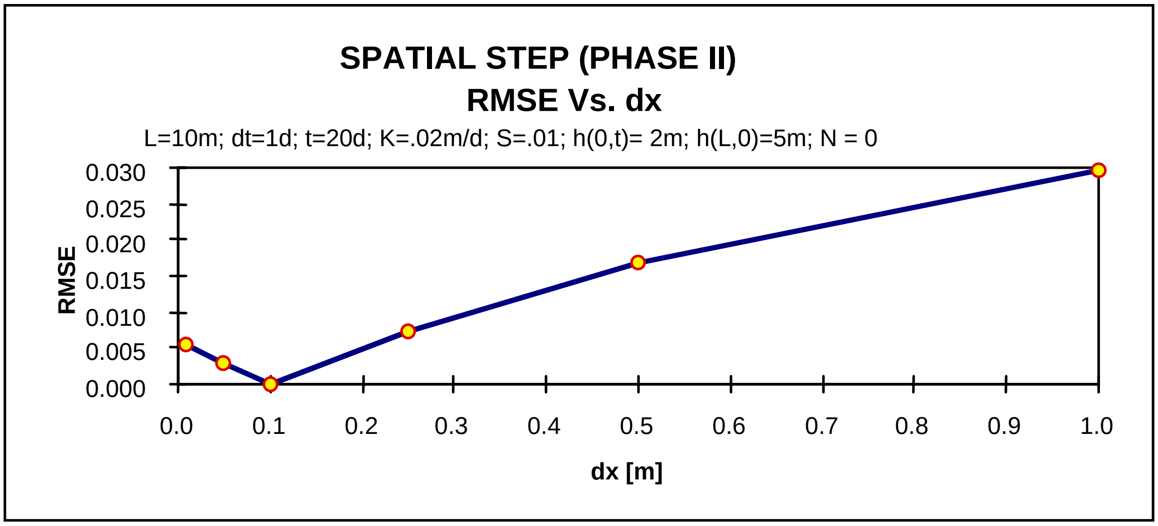

FIGURE 16: Sensitivity analysis for dx under Phase II conditions.

Figure 17 shows RMSE versus the temporal step size. As the temporal step size increased the RMSE values exponentially grew in size indicating a loss of accuracy within the solution. A comparison between Phase I (Figure 13) and Phase II sensitivity results for Δt indicate that the Phase II model was slightly more sensitive to changes in dt.

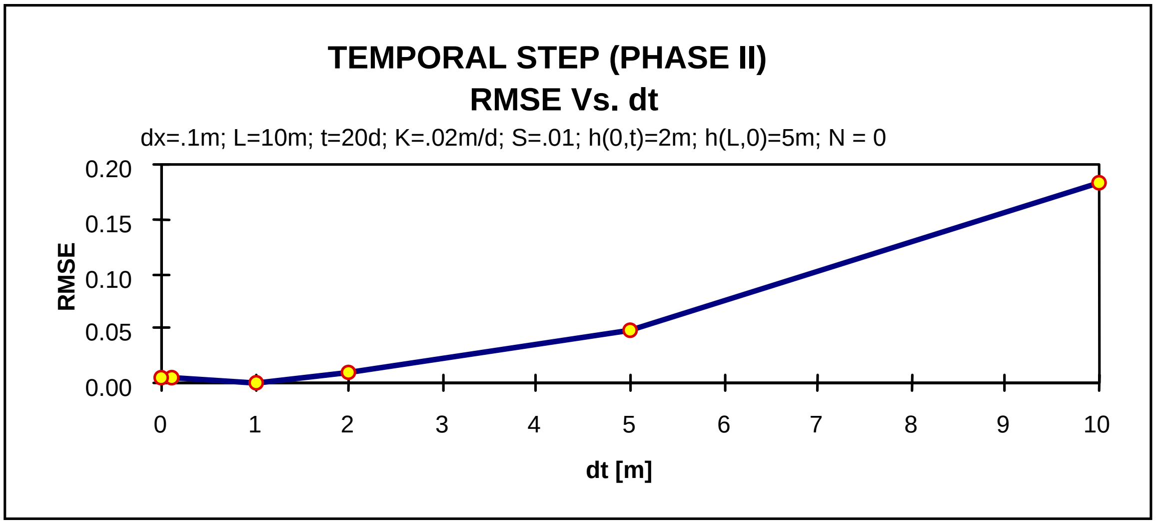

FIGURE 17: Sensitivity analysis for dt under Phase II conditions.

A sensitivity analysis on the hydraulic conductivity was performed utilizing a simulation approach. Figure 18, shows a number of phreatic surfaces generated with differing magnitudes of K. After 20 days a hydraulic conductivity of .0002 m/d resulted in a phreatic surface far from steady state conditions where a K of 0.2 m/d easily achieved steady state.

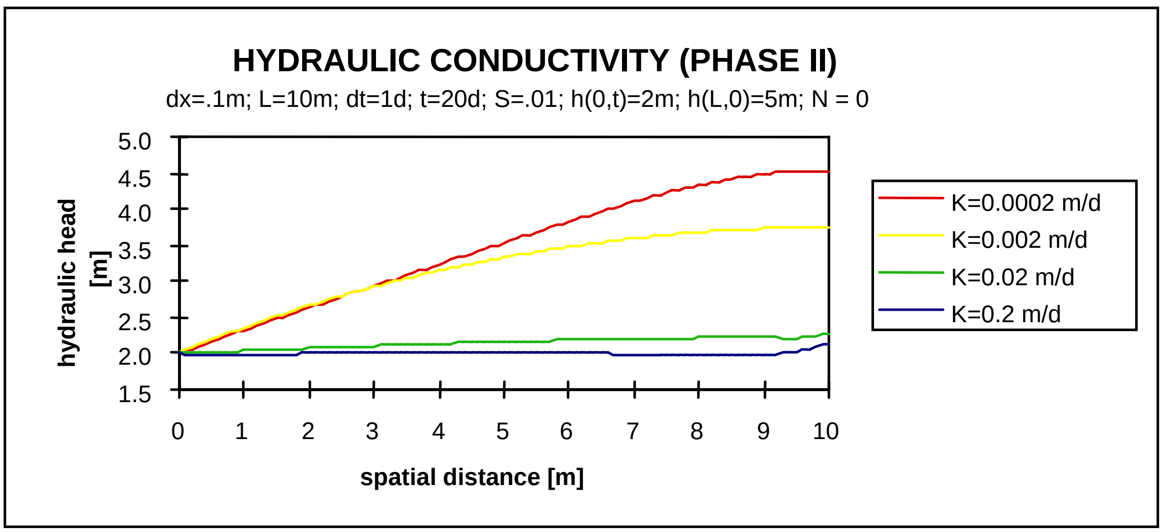

FIGURE 18: Sensitivity analysis for the hydraulic conductivity, K, under Phase II conditions.

Figure 19, shows a number of phreatic surfaces generated utilizing different magnitudes of the storage coefficient. Results from the sensitivity analysis indicate that smaller S values allowed for more rapid movement of water where larger S values hindered the movement of water within the aquifer.

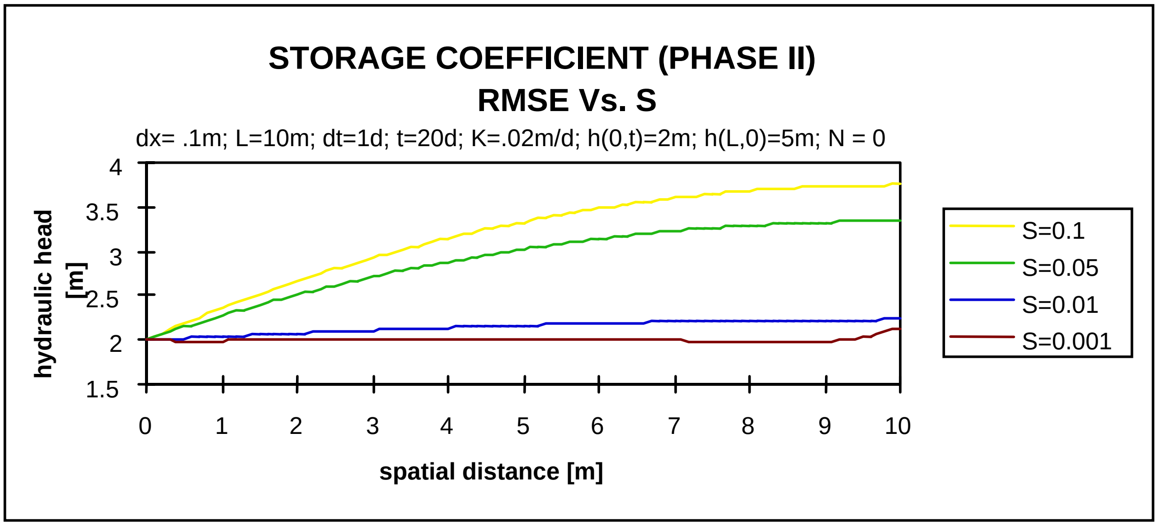

FIGURE 19: Sensitivity analysis for the storage coefficient, S, under Phase II conditions.

Figure 20, shows the hydraulic head response resulting from different magnitudes of accretion held constant over time, t. Results indicate a mounding affect where the larger the N value the larger the mound size.

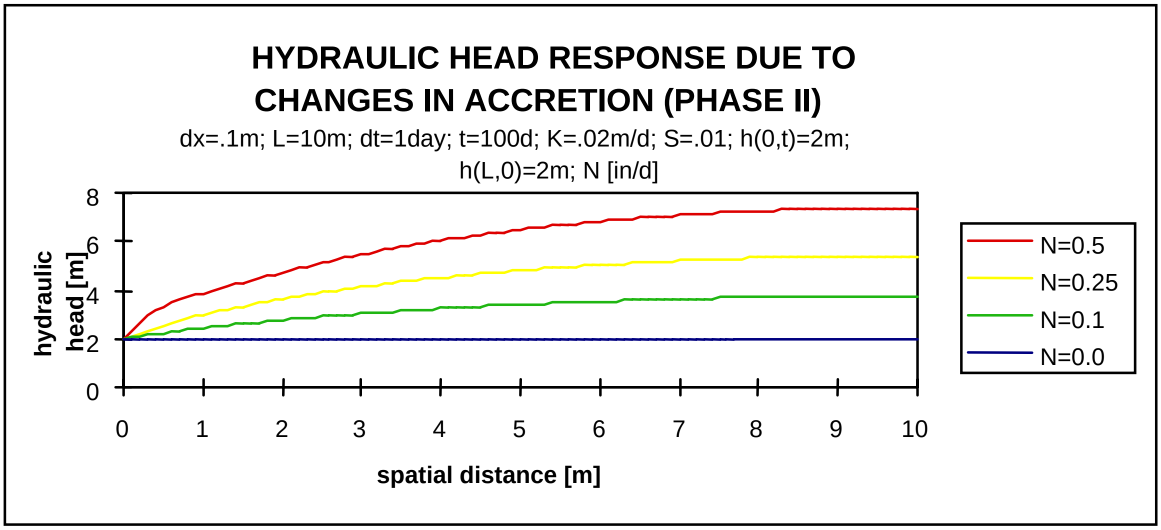

FIGURE 20: Hydraulic head response to changes in accretion under Phase II conditions.

PHASE III SENSITIVITY

Using the simulation approach, a sensitivity analysis for Phase III conditions was performed on the bedrock profile, boundary conditions, K, S, and N.

As discussed in the model formulation for Phase III, the whereabouts of bedrock elevations for the horizontal boundaries of the swale are unknown. Figure 21, displays the bedrock configurations for both the upper and lower sections of the swale which were used in the sensitivity analysis of the bedrock profile.

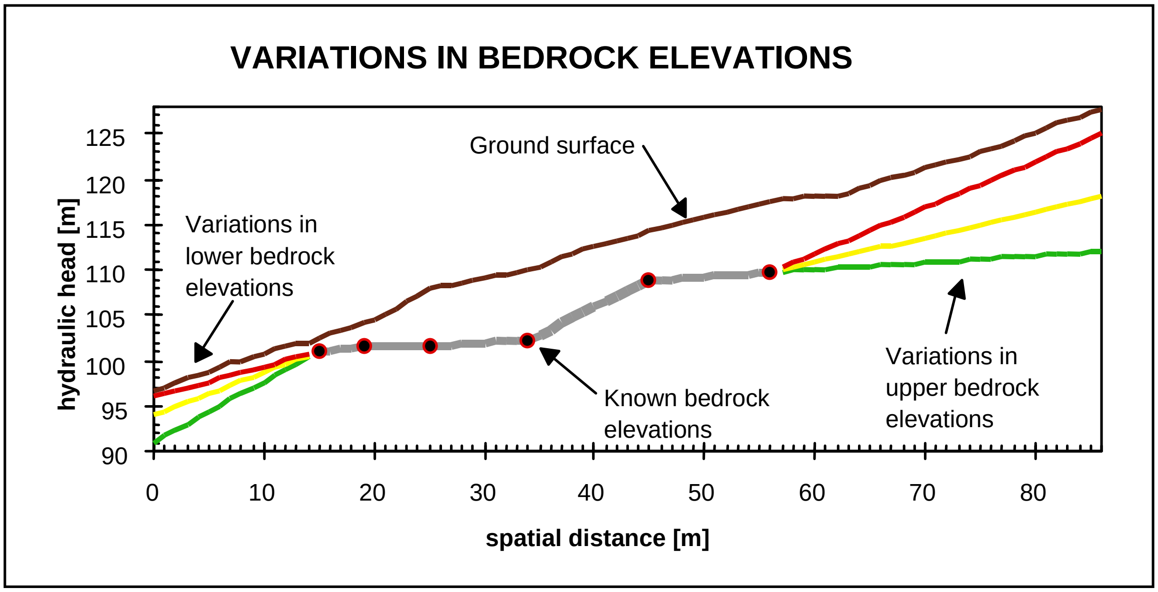

FIGURE 21: Variations in bedrock elevations

Simulations were made, shown in figure 22, which looked at the sensitivity of the model to bedrock configurations for the upper portion of the swale. Each run of the model was made with accretion set to zero in an attempt to simulate the drying out of the swale. By watching the phreatic surface propagate downward over time it was discovered that the model was severely limited in its ability to simulate a dry swale. Dry conditions were not permutable due to the models inability to model non-saturated swale conditions. A transition was needed between the Boussinesq equation for saturated flow and the Richards equation for unsaturated subsurface flow. The no-flux boundary condition allows for a vertical drop in hydraulic head at x = L, however, when bedrock was reached the phreatic surface was then limited in its ability to propagate downward any further. The natural progression of the phreatic surface down the bedrock slope towards x = 0 could not be simulated. The higher the elevation of bedrock at x = L the quicker the model was limited in its movement. For each of the three simulations made, limiting times were found at 7, 50, and 187 days.

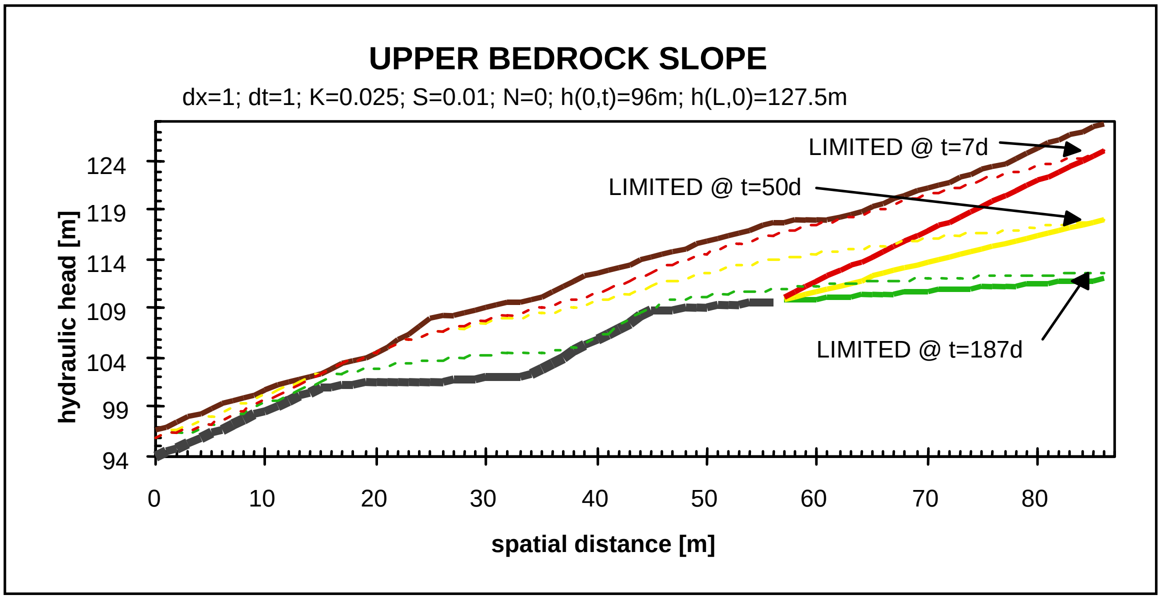

FIGURE 22: Variations in bedrock elevations within the upper portion of the swale

Results from a sensitivity analysis on the lower bedrock configuration are displayed in figure 23. Impacts associated with variations in lower swale bedrock depths were only detectable within the lower portion of the swale. In the lower swale, higher bedrock elevations forced the phreatic surface upward until contact was made with the ground surface. The shape of the phreatic surface was then dictated by the topography of the ground surface. Lower bedrock elevations resulted in a funneling effect of the groundwater through the lower portion of the swale.

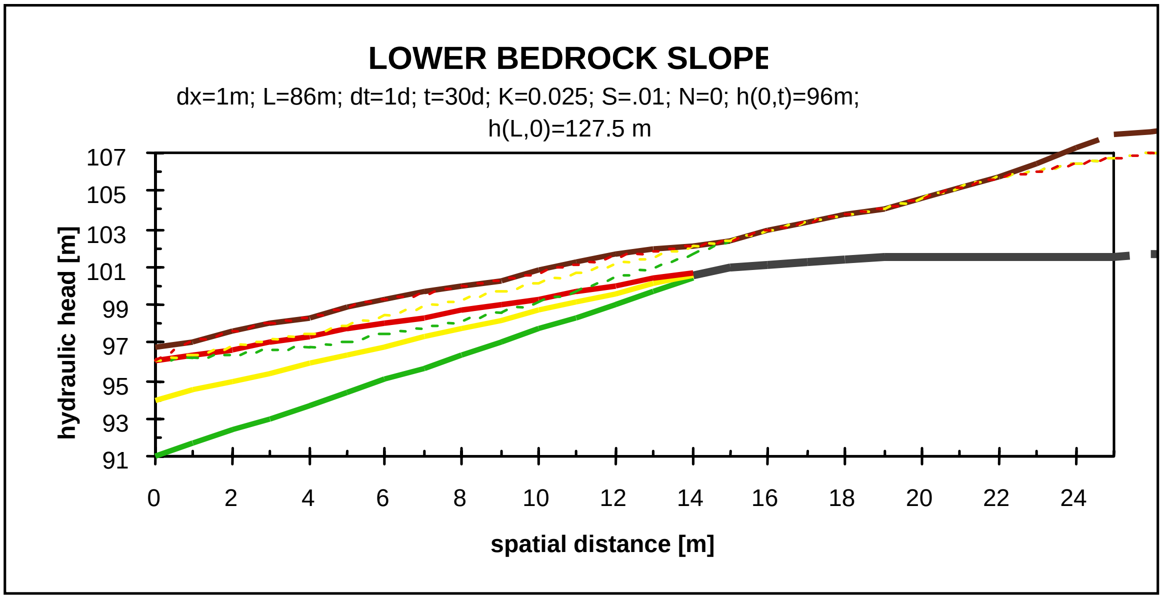

FIGURE 23: Variations in bedrock elevations within the lower portion of the swale

The sensitivity of the model to variations in lower boundary constant head elevations are displayed in figure 24. From the figure it was clear that the model was extremely sensitive to the magnitude of constant hydraulic head at x = 0. Lower constant head elevations resulted in the restriction of exiting flow which in turn caused a backup of waters within the system.

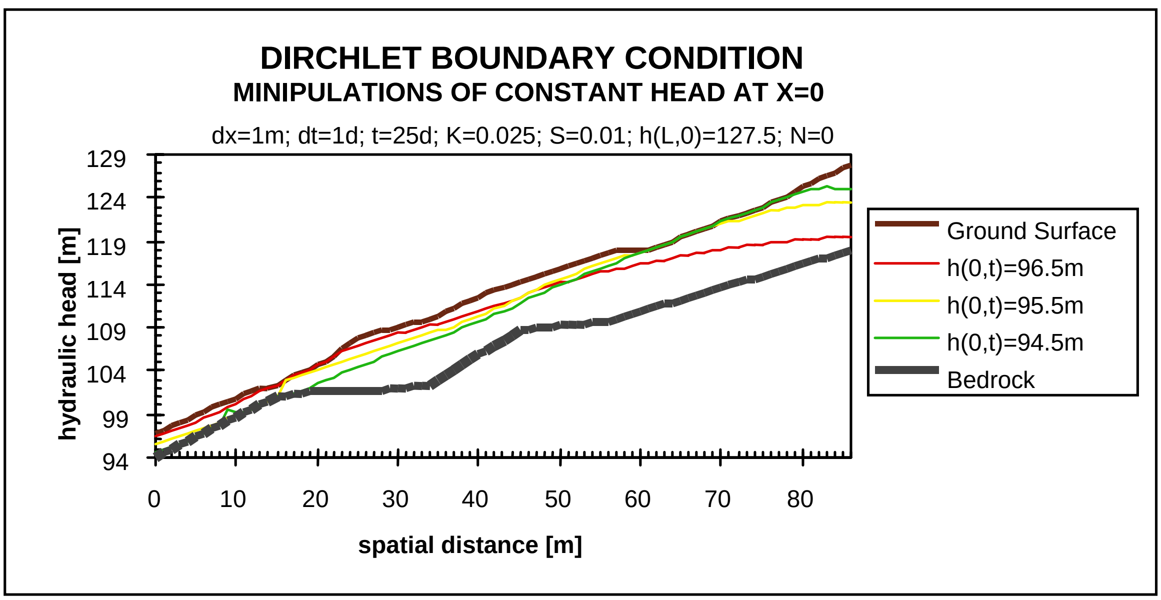

FIGURE 24: Manipulations of constant head at x = 0

An additional impact associated with having a lower elevation at the constant head boundary was the instability produced within the solution. This instability can be found in the spiky nature of the phreatic surface shown in figure 25. Within the lower portion of the swale, both the h(0,t) = 95.5 m and h(0,t) = 94.5 m simulations produced the spiky behavior. The majority of the solution, however, remained stable within the upper portions of the swale.

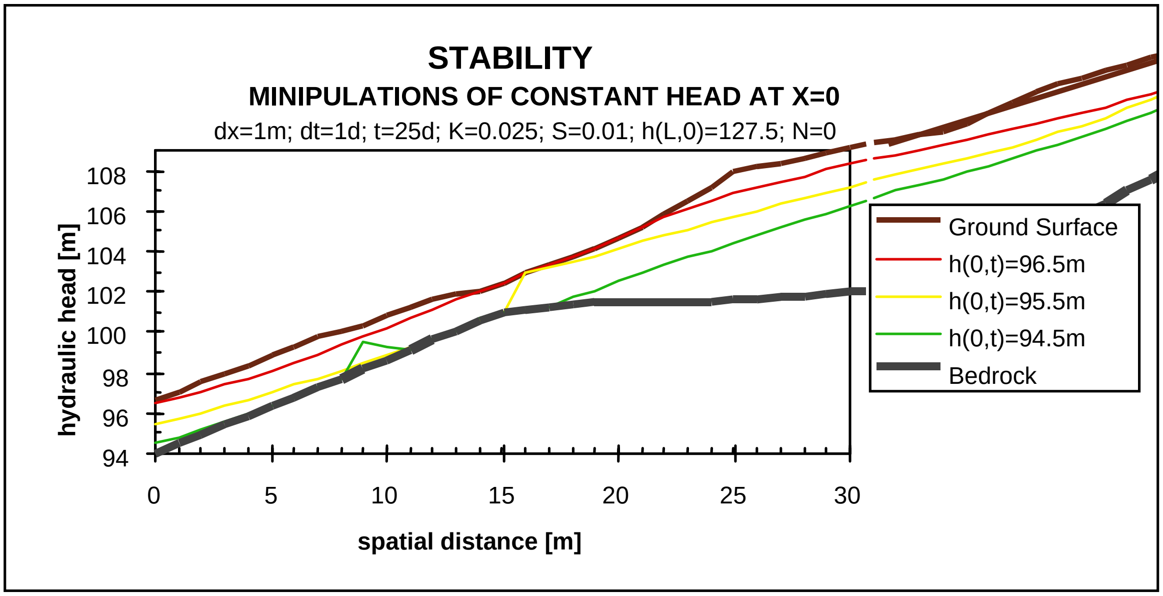

FIGURE 25: Impacts from manipulations of constant head at x = 0 for the lower portion of the swale.

A sensitivity analysis of K can be found in figure 26. Results from the analysis indicate that the larger the magnitude of K the quicker the swale drains. Furthermore, lower magnitudes of K will result in a backup of water within the system.

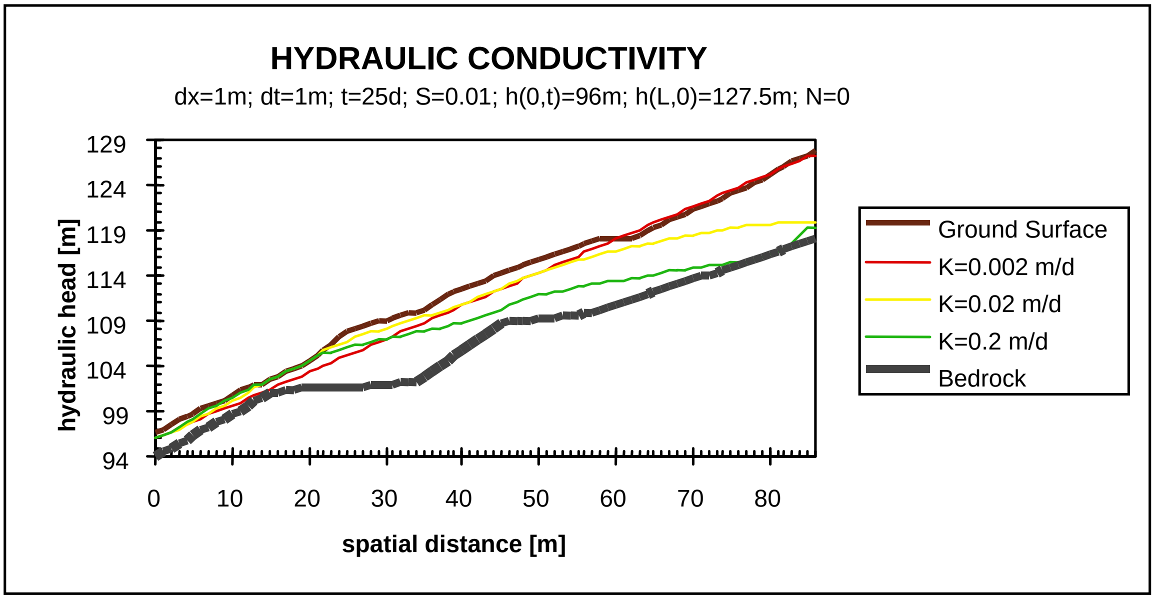

FIGURE 26: Sensitivity analysis for hydraulic conductivity, K, under phase III conditions.

Figure 27, shows a number of phreatic surfaces generated utilizing different magnitudes of the storage coefficient. Results from the sensitivity analysis indicate that small magnitudes of S allow for quick drainage while larger S values produce a backup of groundwater within the swale.

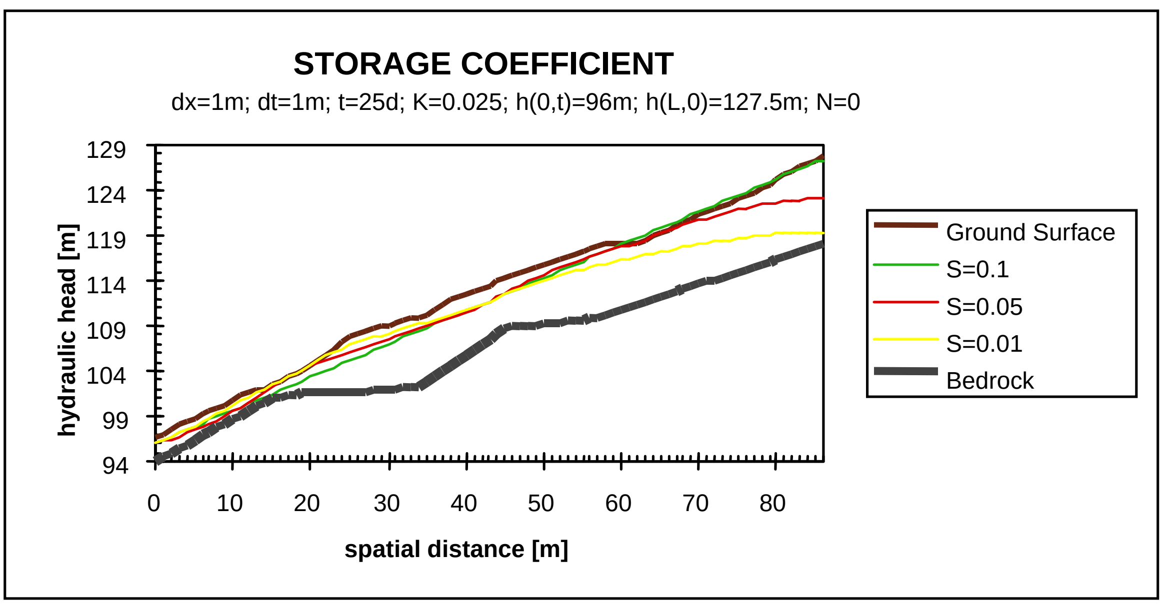

FIGURE 27: Sensitivity analysis for storage coefficient, S, under phase III conditions.

An analysis of the systems response to accretion was made with the construction of simulations which modeled both storm and drought conditions. Figure 28 gives the results for the first scenario which models a single storm event under dry initial conditions. For the first 50 days of the simulation accretion was set to zero allowing for dry swale conditions. After 50 days, accretion was then set to 0.05 m/d in an effort to simulate a storm event. In figure 28, phreatic surfaces are given for a number of temporal locations throughout the simulated storm event.

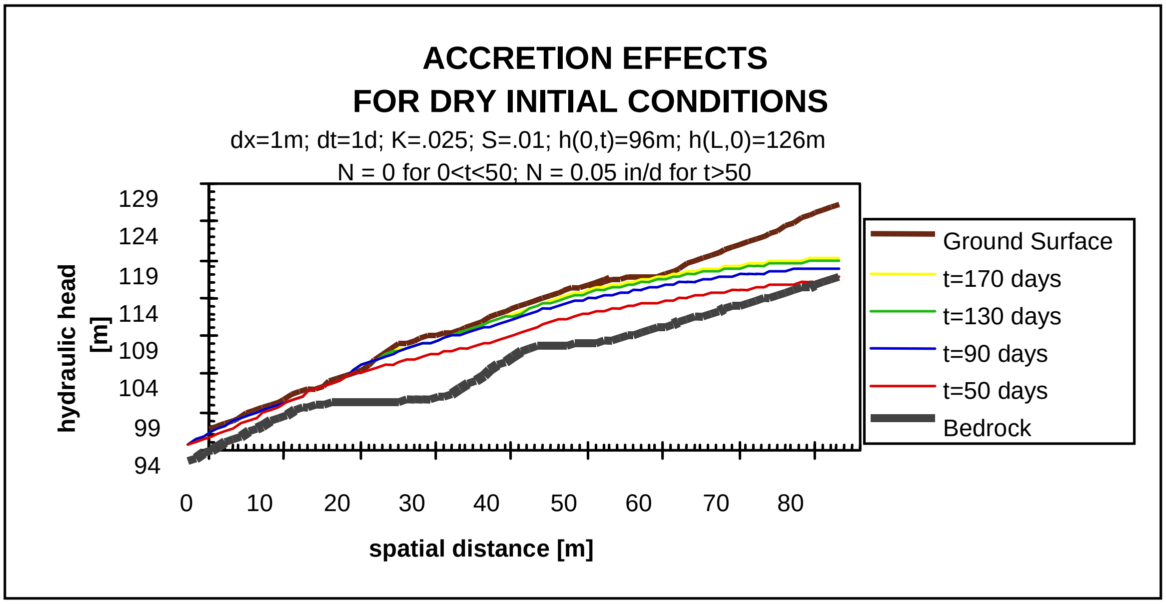

FIGURE 28: Accretion effects for dry initial conditions

The second scenario models drought conditions given an initially saturated system. For the first 50 days of the simulation accretion was set to 0.2 m/d in an attempt to completely saturate the system. After 50 days, a period of drought was simulated with accretion set to zero. In figure 29, phreatic surfaces are given for a number of temporal locations throughout the drought event. As time goes by the phreatic surface propagates downward through the soil matrix.

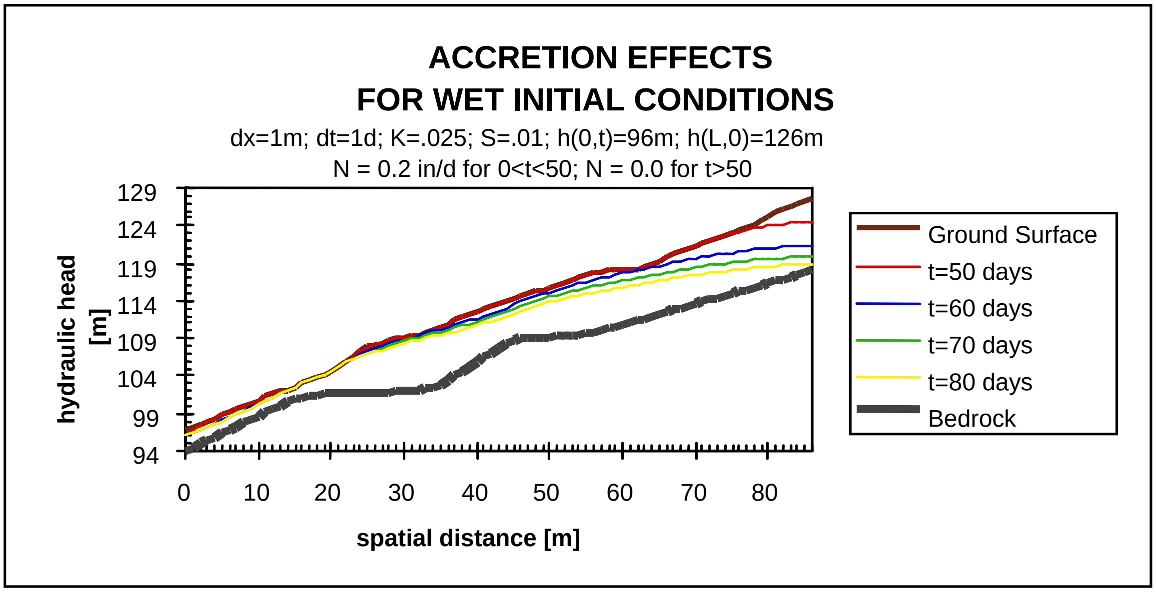

FIGURE 29: Accretion effects for wet initial conditions

CONCLUSIONS

- The models over prediction of hydraulic head was the result of an inaccurate representation of the bedrock within the swale.
- The 1-d model was unable to successfully simulate the drying out of the swale due to the models inability to simulate unsaturated soil conditions.
- Describing the E-Road swale as a homogeneous aquifer oversimplifies the system and was believed to produce abnormally high drainage rates.
- The magnitude of the Dirchlet B.C. at x = 0 made substantial impacts on the phreatic surface throughout the aquifer.
- Smaller spatial and temporal step sizes resulted in greater accuracy within the solution.
- Larger magnitudes of the hydraulic conductivity resulted in quicker drainage rates within the swale.
- Larger magnitudes of the storage coefficient resulted in slower drainage rates within the swale.

SUGGESTIONS FOR FURTHER RESEARCH

FIELD RESEARCH

- Further modeling endeavors require that a detailed map of bedrock elevations be constructed for the E-Road swale.
- Efforts should be made to better describe the soil properties within the swale.
- A tracer study would help to determine the interaction between piezometric holes and piping.

COMPUTATIONAL RESEARCH

- To better understand the impacts associated with alternative bedrock configurations additional simulations of the 1-d model should be run and compared to historical data.
- Modeling the E-Road system with an existing 3-d finite element model (3DFEMWATER).
- The development of a 2-d finite difference model over the horizontal and vertical directions which would allow for a simplified representation of non-homogeneity, surface runoff, and piping within the aquifer system.

REFERENCES

Anderson, M. G., and T. P. Burt, The role of topography in controlling throughflow generation, *Earth Surf. Process*, 3, 331-334, 1978.

Bear, J., *Dynamics of Fluids in Porous Media*, Dover Publications, Inc., New York, pp. 361-437, 1972.

Beven, K. J., The Grendon Underwood Field drainage Experiment, *Inst. Hydrol. Rep. 65*, Inst. of Hydrology, Wallingford, U. K., 1980.

Boussinesq, J., Recherches theoretiques sur l'ecoulement des nappes d'eau infiltrees dans le sol et sur le debit des sources, *J. Math. Pures Appl*., 5, 5-78, 1904.

Brown, D. L., *An Analysis of Transient Flow in Upland Watersheds: Interactions between Structure and Process*, unpublished dissertation submitted to the University of California at Berkeley, 1995.

Chu, W. S., and R. Willis, An explicit finite difference model for unconfined aquifers, *Ground Water*, 22(6), 728-734, 1984.

Douglas, J., and B. F. Jones, On predictor-corrector methods for nonlinear parabolic differential equations, *J. Soc. Indust. Appl. Math*, 11(1), 195-204, 1963.

Dunne, T., and R. D. Black, An experimental investigation of runoff production in permeable soils, *Water Resour. Res*., 6, 478-490, 1970.

Fetter, C. W., *Applied Hydrogeology*, Prentice-Hall: New Jersey, 1994.

Hanberg, W.C., and A. O. Gokce, Rapid water level fluctuations in a thin colluvium landslide west of Cincinnati, Ohio: *U.S. Geol. Surv. Bull.*, 2059-C, 16 p., 1994.

Harr, R. D., Water flux in soil and subsoil on a steep forested slope, *J. Hydrol* ., 33, 37-58, 1977.

Hewlett, J. D., and W. L. Nutter, The varying source area of streamflow from upland basins, paper presented at the Symposium on Interdisciplinary Aspects of Watershed management, Montana State Univ., Bozeman, 1970.

Huff, T. L., and D. W.Smith, W. R. Powell, Tables for the soil-vegetation map, Jackson State Forest, Mendocino County, Soil-Vegetation Survey, California Department of Forestry, The Resources Agency, Sacramento, California, USA. 1985.

Killbourne, R. T., Geology and slope stability of the Fort Bragg area, *California Geol*., 39(3), 58-68, 1986.

Mosely, M. P., Streamflow generation in a forested watershed, New Zealand, *Water Resour. Res*., 15(4), 795-806, 1979.

Reddi, L. N., and I. M. Lee, T. H. Wu, A comparison of models predicting groundwater levels of hillside slopes, *Water Res. Bull*., 26(4), 657-667, 1990.

Remson, I., and G. M. Hornberger, F. J. Molz, *Numerical Methods in Subsurface Hydrology*, John Wiley & Sons, Inc., New York, pp. 59-121, 1971.

Singh, R. N., and S. N. Rai, D. V. Ramana, Water table fluctuation in a sloping aquifer with transient recharge, *J. Hydrol.*, 126, 315-326, 1991.

Sloan, P. G., and I. D. Morre, Modeling subsurface stormflow on steeply sloping forested watersheds, *Water Resources Research*, 20(12), 1815-1822, 1984.

Swanston, D. N., Soilwater piezometry in a southeast Alaska landslide area, *Res. Note PNW-68*, 17 pp., For. Serv., U.S. Dep. of Agric., Portland, Oreg., 1967.

Weyman, D. R.,Throughflow on hillslopes and its relation to the stream hydrograph, *Bull. Int. Assoc. Sci. Hydrol*., 15(2), 25-33, 1970.

Whipkey, R. Z., Subsurface stormflow on forested slopes, *Bull. Int. Assoc. Sci. Hydrol*., 10(2), 74-85, 1965.

Yeh, W. W., and G. W. Tauxe, Optimal Identification of Aquifer Diffusivity Using Quasilinearization, *Water Resources Research*, 7(4), 955-962, 1971.

APPENDIX A

PHASE III COMPUTER CODE

```
PROGRAM PHASE3
C
C  Unconfined subsurface flow, non-horizontal bottom with constant
C  B.C. at x = 0 and a no flow B.C. for x = L.
C______________________________________________________________________
C  VARIABLE DICTIONARY
C
C  arrc()  = center tridiagonal coef.
C  arrl()  = left hand tridiagonal coef.
C  arrr()  = right hand tridiagonal coef.
C  B()     = dimensionless base information
C  BL      = L.H.S. piez. boundary condition (m)
C  BR      = R.H.S. piez. boundary condition (m)
C  bound() = known information
C  dxx     = spatial step size which remains constant (m)
C  dX      = dimensionless spatial step
C  dT      = dimensionless temporal step
C  dtt     = temporal step size which remains constant (day)
C  fileout = file which contains output information
C  half()  = those piez. heads produced after the PREDICTOR step (m)
C  i,j     = index counter
C  K       = hydraulic conductivity (m/day)
C  L       = length of base (m)
C  lbound  = left hand piez. boundary condition (m)
C  n       = spatial nodes
C  new()   = those piez. heads produced after the CORRECTOR step (m)
C  node    = spatial node place holder
C  nt      = temporal nodes
C  ntime   = temporal node place holder
C  old()   = piez. heads introduced at begining of time step (m)
C  out     = spatial distance from L.H.S. boundary (m)
C  period  = period of time which to simulate over (day)
C  rrr()   = rainfall rate (in/day)
C  R()     = dimensionless rainfall rate information
C  rbound  = right hand piez. boundary conditions (m)
C  surf()  = elevations of ground surface (m)
C  S       = porosity
C  TRIDIAG = subroutine called to solve Ax=b
C  xx()    = inital inner piez. heads (m)
C______________________________________________________________________
C
       integer i,j,n,nt
       real lbound,rbound,BL,BR,dxx,dX,dT,dtt,L,period,S,node,ntime
       real rrr,K
       real arrc(1000),arrl(1000),arrr(1000),bound(1000)
       real half(1000),new(1000),old(1000),xx(1000),surf(1000)
       real R(1000),B(1000)
       character fileout*30
C
       write(*,*) '**************************************************'
       write(*,*) '*    A subsurface model of one-dimensional      *'
       write(*,*) '* unconfined flow based on Darcian assumptions. *'
       write(*,*) '*           Non-horizontal bottom               *'
       write(*,*) '**************************************************'
       write(*,*)
```

```
C
C  Open output file
C
       open(17,file='bedrock',status='old')
       open(16,file='surface',status='old')
       write(*,*)
C
C  Spatial information
C
       write(*,'(A,$)') ' Enter the spatial step size (m):'
       read(*,'(f5.2)') dxx
       write(*,'(A,$)') ' Enter the length of the aquifer (m):'
       read(*,'(f7.2)') L
       write(*,*)
       node=L/dxx
       n=INT(node)
C
C  Temporal information
C
       write(*,'(A,$)') ' Enter the temporal step size (day):'
       read(*,'(f5.2)') dtt
       write(*,'(A,$)') ' Enter the period of time to simulate (day):'
       read(*,'(f7.2)') period
       period=REAL(period)
       write(*,*)
       ntime=period/dtt
       nt=INT(ntime)
C
C  Physical parameters
C
       write(*,'(A,$)') ' Enter the hydraulic conductivity (m/day):'
       read(*,'(f5.4)') K
       write(*,'(A,$)') ' Enter the storage coefficient:'
       read(*,'(f5.4)') S
       write(*,*)
C
C  Boundary Conditions
C
       write(*,'(A,$)') ' Enter the left-hand constant B.C. (m):'
       read(*,'(f5.2)') lbound
       write(*,'(A,$)') ' Guess the right-hand B.C. (m):'
       read(*,'(f5.2)') rbound
       write(*,*)
C
C  Transformations/Initializing array's
C
       write(*,'(A,$)') ' Enter the rainfall rate [in/day]:'
       read(*,'(f5.2)') rrr
C
       Do 20 i=1,n+1
         read(17,*) B(i)
         B(i)=B(i)/L
         read(16,*) surf(i)
         surf(i)=surf(i)/L
         xx(i)=((rbound-lbound)/L)*(dxx*(i-1)) + lbound
         old(i)=xx(i)/L
```

```
half(i)=1.
```

```
20     Continue
C
C  Rainfall is converted from inches to meters and made dimensionless
C
       Do 21 j=1,nt
         R(j)=(rrr/39.37)/K
21     Continue
C
C  Dimensionless variables
C
       BL=lbound/L
       BR=rbound/L
       half(1)=BL
       half(n+1)=BR
       dX=dxx/L
       dT=(K*dtt) / (S*L)
C
C  Start temporal loop
C
       Do 30 j=1,nt
C
C  Start spatial loop
C
         Do 40 i=2,n
C
C  Devlop PREDICTOR tridiagonal
C
           If (i.EQ.2) then
             arrl(i)=0.0
             arrc(i)=-1.*(2 + (2*dX**2)/((old(i)-B(i))*dT))
             arrr(i)=1.0
             bound(i)=((-2.*(dX**2)*old(i))/(dT*(old(i)-B(i))))
     2                 + (((old(i)-old(i-1))/(old(i)-B(i))))
     3                 * (old(i-1)-old(i)+B(i)-B(i-1))
     4                 + ((-1.*(dX**2)*R(j))/(old(i)-B(i))) - BL
           Elseif (i.EQ.n) then
             arrl(i)=1.0
             arrc(i)=1.+(-1.*(2 + (2*dX**2)/((old(i)-B(i))*dT)))
             arrr(i)=0.0
             bound(i)=((-2.*(dX**2)*old(i))/(dT*(old(i)-B(i))))
     2                 + (((old(i)-old(i-1))/(old(i)-B(i))))
     3                 * (old(i-1)-old(i)+B(i)-B(i-1))
     4                 + ((-1.*(dX**2)*R(j))/(old(i)-B(i)))
           Else
             arrl(i)=1.0
             arrc(i)=-1.*(2 + (2*dX**2)/((old(i)-B(i))*dT))
             arrr(i)=1.0
             bound(i)=((-2.*(dX**2)*old(i))/(dT*(old(i)-B(i))))
     2                 + (((old(i)-old(i-1))/(old(i)-B(i))))
     3                 * (old(i-1)-old(i)+B(i)-B(i-1))
     4                 + ((-1.*(dX**2)*R(j))/(old(i)-B(i)))
           Endif
C
40       Continue
```

```
C
C  Call tridiagnal solver
C
         Call TRIDIAG (n,arrl,arrc,arrr,bound,half)
C
         half(1)=BL
         half(n+1)=half(n)
C
C  Start spatial loop again.
C
         Do 41 i=2,n
C
C  Develop CORRECTOR tridiagonal
C
           If (i.EQ.2) then
             arrl(i)=0.0
             arrc(i)=-1.*(2 + (2*dX**2)/((half(i)-B(i))*dT))
             arrr(i)=1.0
             bound(i)=((-2.*(dX**2)*old(i))/((half(i)-B(i))*dT))
     2                + ((2.*(half(i)-half(i-1)))/(half(i)-B(i)))
     3                * (-1.*half(i)+half(i-1)+B(i)-B(i-1))
     4                - ((2.*(dX**2)*R(j))/(half(i)-B(i)))
     5                - (old(i+1)-2.*old(i)+old(i-1)) - BL
           Elseif (i.EQ.n) then
             arrl(i)=1.0
             arrc(i)=1.+(-1.*(2 + (2*dX**2)/((half(i)-B(i))*dT)))
             arrr(i)=0.0
             bound(i)=((-2.*(dX**2)*old(i))/((half(i)-B(i))*dT))
     2                + ((2.*(half(i)-half(i-1)))/(half(i)-B(i)))
     3                * (-1.*half(i)+half(i-1)+B(i)-B(i-1))
     4                - ((2.*(dX**2)*R(j))/(half(i)-B(i)))
     5                - (-1.*old(i)+old(i-1))
           Else
             arrl(i)=1.0
             arrc(i)=-1.*(2 + (2*dX**2)/((half(i)-B(i))*dT))
             arrr(i)=1.0
             bound(i)=((-2.*(dX**2)*old(i))/((half(i)-B(i))*dT))
     2                + ((2.*(half(i)-half(i-1)))/(half(i)-B(i)))
     3                * (-1.*half(i)+half(i-1)+B(i)-B(i-1))
     4                - ((2.*(dX**2)*R(j))/(half(i)-B(i)))
     5                - (old(i+1)-2.*old(i)+old(i-1))
           Endif
C
41       Continue
C
C  Call tridiagnal solver
C
         Call TRIDIAG (n,arrl,arrc,arrr,bound,new)
C
C  Save the peiz. heads from t into an old() array
C
         Do 35 i=2,n
           if(new(i).GT.B(i).AND.new(i).LT.surf(i)) then
             old(i)=new(i)
           elseif(new(i).LE.B(i)) then
             old(i)=B(i)+.0000001
```

```
elseif(new(i).GE.surf(i)) then
  old(i)=surf(i)
```

```
          endif
35      Continue
        old(1)=BL
        old(n+1)=old(n)
        if(old(n+1).LT.B(n+1)) then
          old(n+1)=B(n+1)
        endif
C
30    continue
C
      write(*,*)
      write(*,*)
      write(*,*) 'SOLUTION: hydraulic head [m]'
      Do 50 i=1,n+1
        out=(i*dxx)-dxx
        write(*,*) old(i)*L
50    Continue
C
      stop
      end
C
C***********************************************************************
C*                       TRIDIAG SUBROUTINE                            *
C***********************************************************************
      subroutine TRIDIAG (ndist1,dl1,di1,du1,b1,Ht1)
      implicit none
      integer i,ndist1
      real dl1(1000),di1(1000),du1(1000)
      real b1(1000),Ht1(1000),xmult
C
      Do 10 i=3,ndist1
        if(dl1(i).ne.0.0)then
          xmult=di1(i-1)/dl1(i)
          di1(i)=du1(i-1)-xmult*di1(i)
          du1(i)=-xmult*du1(i)
          b1(i)=b1(i-1)-xmult*b1(i)
        endif
10    Continue
      Ht1(ndist1)=b1(ndist1)/di1(ndist1)
      Do 31 i=ndist1,2,-1
        Ht1(i)=(b1(i)-du1(i)*Ht1(i+1))/di1(i)
31    Continue
      return
      end
```

APPENDIX B

OUTPUT FROM PHASE III MODEL

```
***********************************************
*    A subsurface model of one-dimensional      *
* unconfined flow based on Darcian assumptions. *
*           Non-horizontal bottom               *
***********************************************

Enter the spatial step size (m):1.
Enter the length of the aquifer (m):86.

Enter the temporal step size (day):1.
Enter the period of time to simulate over (day):50.

Enter the hydraulic conductivity (m/day):.025
Enter the storage coefficient:.01

Enter the left-hand constant B.C. (m):96.
Guess for the right-hand B.C. (m):126.

Enter the rainfall rate [in/day]:.02

SOLUTION: hydraulic head [m]
   96.00000
   96.46834
   96.92689
   97.37561
   97.81886
   98.25777
   98.69318
   99.12352
   99.55211
   99.97929
   100.4054
   100.8281
   101.2509
   101.6739
   102.0300
   102.4200
   102.9300
   103.3800
   103.7300
   104.1000
   104.6100
   105.1900
   105.8000
   106.3344
   106.7214
   107.0772
   107.4079
   107.7208
   108.0181
   108.3018
   108.5728
   108.8323
   109.0812
   109.3208
   109.5511
```

109.7733
110.0031
110.2417
110.4892
110.7466
111.0144
111.2929
111.5832
111.8863
112.2030
112.5339
112.8798
113.1961
113.4878
113.7595
114.0140
114.2528
114.4783
114.6913
114.8933
115.0852
115.2673
115.4407
115.6116
115.7797
115.9452
116.1077
116.2675
116.4242
116.5781
116.7287
116.8762
117.0204
117.1612
117.2984
117.4321
117.5618
117.6877
117.8096
117.9269
118.0397
118.1473
118.2498
118.3463
118.4366
118.5197
118.5951
118.6612
118.7167
118.7587
118.7835
118.7835

APPENDIX C

SPLUS - FORTRAN 77 CODE

GRAPHICAL REPRESENTATION OF HISTORICAL AND SIMULATED DATA

SPLUS DRIVER FUNCTION: "submodel()"

```
function(horiz = T, K, S, lbound, rbound, hole, sd, ed, percentr)
{
      print("************************************************")
      print("*    A subsurface model of one-dimensional      *")
      print("* unconfined flow based on Darcian assumptions. *")
      print("************************************************")
      print("")    #
# Horizontal/non-horizontal conditions
      if(horiz == T) {
            dxx <- 1
            L <- 10
            n <- (L/dxx) + 1
            B <- rep(0, n)
      }
      else {
            aquifer <- "einfo2"
            aquif <- scan(paste(c("/home/rsl2/jfisher/Model1d/", aquifer),
                  collapse = ""), what = list(distance = 0, base = 0,
                  surface = 0), flush = T)
            aquif <- as.data.frame(aquif)
            B <- aquif$base
            surf <- aquif$surface
            dxx <- aquif$distance[2] - aquif$distance[1]
            L <- max(aquif$distance) - 1
      }
      K <- as.numeric(K)
      S <- as.numeric(S) #
# Boundary Conditions
      lbound <- as.numeric(lbound)
      rbound <- as.numeric(rbound)    #
# Piezometric hole to be observed
      piez <- dist2piez(hole)
      filename <- paste(c("subsk2", hole), collapse = "")#
# Spatial and temporal intormation
      dtt <- 1
      stime <- 0
      etime <- 2400
      ms <- as.numeric(substring(sd, 1, 2))
      ds <- as.numeric(substring(sd, 4, 5))
      ys <- as.numeric(substring(sd, 7, 8)) + 1900
      sdate <- julian(ms, ds, ys, origin = c(12, 31, 1983)) - 100
      me <- as.numeric(substring(ed, 1, 2))
      de <- as.numeric(substring(ed, 4, 5))
      ye <- as.numeric(substring(ed, 7, 8)) + 1900
      edate <- julian(me, de, ye, origin = c(12, 31, 1983))
      sd <- dates(sd)
      ed <- dates(ed)
      period <- as.numeric(edate - sdate) + 1      #
# Rainfall Rate
      days <- c(rain$cd[1]:rain$cd[length(rain$cd)])
      norain <- c(rep(0, length(days)))
      subframe <- as.data.frame(list(days = days, norain = norain))
      comb <- merge(subframe, rain, by = 1, all.x = T)
```

```
      comb$precip[comb$precip == "NA"] <- c(rep(0, length(comb$precip[comb$
             precip == "NA"])))
      rain <- comb[, c("days", "precip")]
      segm <- seq(along = rain$days)[rain$days > (sdate - 1) & rain$days < (
             edate + 1)]
      minsegm <- min(segm)
      maxsegm <- max(segm)
      rain <- rain[minsegm:maxsegm,  ]
      percentr <- percentr/100
      R <- rain$precip * percentr     #
# Download historical piezometric hole information
      hist <-
scan(paste(c("/user15/water/data/subsurface/er/analysis/recon/",
             filename), collapse = ""), what = list(Day = 0, Time = 0, Elev
              = 0, Code = 0), flush = T)     #
# Run numerical fortran model
      span <- dates(sdate:edate, origin = c(12, 31, 1983))      #
      model <- list(day = span, head = rep(0, length(span)))
      model <- list(day = span, head = rep(0, 200))
      model$head <- stepplus.fortran(dxx, L, dtt, period, K, S, lbound,
             rbound, piez, R, B, surf)
      model$head <- model$head[101:length(span)]
      model$day <- span #
# Graph boundaries
      ymin <- 999
      ymax <- -999
      lrain <- 666
      hrain <- -666      #
# Graphics window set up
      if(is.null(dev.list()))
             openlook()
      par(mfrow = c(1, 1), mar = c(4, 5, 3, 2))
      colors <- xgetrgb("lines")
      background <- xgetrgb("background")
      images <- xgetrgb("images")
      ps.options.send(colors = colors, background = background, image.colors
              = images)   #
#
      sdate <- format(sd)
      edate <- format(ed)#
#
      top <- 1
      mid <- 0.2
      bot <- 0     #
# Graph everything up
      X <- subdump2(hist, filename, sdate, stime, edate, etime, ymin, ymax,
             top, mid, bot, lrain, hrain, model, rain)
}
```

SPLUS FUNCTION: “stepplus.fortran()”

```
function(dxx, L, dtt, period, K, S, lbound, rbound, piez, R, B, surf)
{
# Calls fortran function for piez information
      if(!is.loaded(Fortran.symbol("_stepplus_"))) dyn.load(
                    "/user15/water/data/subsurface/er/analysis/stepplus.o")
      nt <- 200
      result <- numeric(nt)
      .Fortran("stepplus",
            as.double(result),
            as.double(dxx),
            as.double(L),
            as.double(dtt),
            as.double(period),
            as.double(K),
            as.double(S),
            as.double(lbound),
            as.double(rbound),
            as.integer(piez),
            as.double(R),
            as.double(B)
            as.double(surf))[[1]]
}
```

FORTRAN77 SUBROUTINE: "stepplus.f"

```
    subroutine
stepplus(result,dxx,L,dtt,period,K,S,lbound,rbound,piez,R,B,surf)
C
C  Unconfined subsurface flow, non-horizontal bottom with constant boundary
C  conditions.  Designed to link with Splus.
C
       integer i,j,n,nt,piez
       double precision lbound,rbound,BL,BR,dxx,dX,dT,dtt,K,L
       double precision period,S,node,ntime
       double precision arrc(200),arrl(200),arrr(200),bound(200),half(200)
       double precision new(200),old(200),xx(200)
       double precision R(200),B(200)
       double precision result(200),surf(200)
C
       node=L/dxx
       n=INT(node)
       ntime=period/dtt
       nt=INT(ntime)
C
C  Initial Conditions
C
       xx(1)=lbound
       xx(n+1)=rbound
       Do 10 i=2,n
         xx(i)=(((rbound-lbound)/L)*i)+lbound
10     Continue
C
C  Account for initial result
C
       result(1)=xx(piez)
C
C  Transformations/Initializing array's
C
       Do 20 i=1,n+1
         old(i)=xx(i)/L
         half(i)=1.
          B(i)=B(i)/L
          surf(i)=surf(i)/L
20     Continue
C
       BL=old(1)
       BR=old(n+1)
       half(1)=BL
       half(n+1)=BR
       dX=dxx/L
       dT=(K*dtt) / (S*L)
C
       Do 33 j=1,nt+1
         R(j)=(R(j)/39.37)/K
33     Continue
C
C  Start temporal loop
C
```

```
      Do 30 j=2,nt+1
C
C  Start spatial loop
C
        Do 40 i=2,n
C
C  Devlop PREDICTOR tridiagonal
C
          If (i.EQ.2) then
            arrl(i)=0.0
            arrc(i)=-1.*(2 + (2*dX**2)/((old(i)-B(i))*dT))
            arrr(i)=1.0
            bound(i)=((-2.*(dX**2)*old(i))/(dT*(old(i)-B(i))))
     2               + (((old(i)-old(i-1))/(old(i)-B(i))))
     3               * (old(i-1)-old(i)+B(i)-B(i-1))
     4               + ((-1.*(dX**2)*R(j))/(old(i)-B(i))) - BL
          Elseif (i.EQ.n) then
            arrl(i)=1.0
            arrc(i)=1. + (-1.*(2 + (2*dX**2)/((old(i)-B(i))*dT)))
            arrr(i)=0.0
            bound(i)=((-2.*(dX**2)*old(i))/(dT*(old(i)-B(i))))
     2               + (((old(i)-old(i-1))/(old(i)-B(i))))
     3               * (old(i-1)-old(i)+B(i)-B(i-1))
     4               + ((-1.*(dX**2)*R(j))/(old(i)-B(i)))
          Else
            arrl(i)=1.0
            arrc(i)=-1.*(2 + (2*dX**2)/((old(i)-B(i))*dT))
            arrr(i)=1.0
            bound(i)=((-2.*(dX**2)*old(i))/(dT*(old(i)-B(i))))
     2               + (((old(i)-old(i-1))/(old(i)-B(i))))
     3               * (old(i-1)-old(i)+B(i)-B(i-1))
     4               + ((-1.*(dX**2)*R(j))/(old(i)-B(i)))
          Endif
C
40      Continue
C
C  Call tridiagnal solver
C
        Call TRIDIAG (n,arrl,arrc,arrr,bound,half)
C
        half(1)=BL
        half(n+1)=half(n)
C
C  Start spatial loop again.
C
        Do 41 i=2,n
C
C  Develop CORRECTOR tridiagonal
C
          If (i.EQ.2) then
            arrl(i)=0.0
            arrc(i)=-1.*(2 + (2*dX**2)/((half(i)-B(i))*dT))
            arrr(i)=1.0
            bound(i)=((-2.*(dX**2)*old(i))/((half(i)-B(i))*dT))
     2               + ((2.*(half(i)-half(i-1)))/(half(i)-B(i)))
     3               * (-1.*half(i)+half(i-1)+B(i)-B(i-1))
```

```
     4                - ((2.*(dX**2)*R(j))/(half(i)-B(i)))
     5                - (old(i+1)-2.*old(i)+old(i-1)) - BL
           Elseif (i.EQ.n) then
             arrl(i)=1.0
             arrc(i)=1. + (-1.*(2 + (2*dX**2)/((half(i)-B(i))*dT)))
             arrr(i)=0.0
             bound(i)=((-2.*(dX**2)*old(i))/((half(i)-B(i))*dT))
     2                + ((2.*(half(i)-half(i-1)))/(half(i)-B(i)))
     3                * (-1.*half(i)+half(i-1)+B(i)-B(i-1))
     4                - ((2.*(dX**2)*R(j))/(half(i)-B(i)))
     5                - (-1.*old(i)+old(i-1))
           Else
             arrl(i)=1.0
             arrc(i)=-1.*(2 + (2*dX**2)/((half(i)-B(i))*dT))
             arrr(i)=1.0
             bound(i)=((-2.*(dX**2)*old(i))/((half(i)-B(i))*dT))
     2                + ((2.*(half(i)-half(i-1)))/(half(i)-B(i)))
     3                * (-1.*half(i)+half(i-1)+B(i)-B(i-1))
     4                - ((2.*(dX**2)*R(j))/(half(i)-B(i)))
     5                - (old(i+1)-2.*old(i)+old(i-1))
           Endif
C
41       Continue
C
C  Call tridiagnal solver
C
         Call TRIDIAG (n,arrl,arrc,arrr,bound,new)
C
C  Save the peiz. heads from t into an old() array
C
         Do 35 i=2,n
           if(new(i).LT.B(i)) then
             old(i)=B(i)+0.00000001
           elseif(new(i).GT.surf(i)) then
             old(i)=surf(i)
           else
             old(i)=new(i)
           endif
35       Continue
         old(1)=BL
         old(n+1)=old(n)
         result(j)=old(piez)*L
C
30     continue
C
       return
       end
C
```

```
C***********************************************************************
C*                    TRIDIAG SUBROUTINE                               *
C***********************************************************************
      subroutine TRIDIAG (ndist1,dl1,di1,du1,b1,Ht1)
      implicit none
      integer i,ndist1
      double precision dl1(200),di1(200),du1(200)
      double precision b1(200),Ht1(200),xmult
C
      Do 10 i=3,ndist1
        if(dl1(i).ne.0.0)then
          xmult=di1(i-1)/dl1(i)
          di1(i)=du1(i-1)-xmult*di1(i)
          du1(i)=-xmult*du1(i)
          b1(i)=b1(i-1)-xmult*b1(i)
        endif
10    Continue
      Ht1(ndist1)=b1(ndist1)/di1(ndist1)
      Do 31 i=ndist1,2,-1
        Ht1(i)=(b1(i)-du1(i)*Ht1(i+1))/di1(i)
31    Continue
      return
      end
```

SPLUS FUNCTION: “axis.time2()”

```
function(xdump, filename, sdate, stime, edate, etime, ymin, ymax, top, mid,
bot,
      lrain, hrain, model, rain, ...)
{
# "subdump" is a plotting function which is called upon by "multisub".
#
# Parameters:
#      xdump = a list object containing subsurface information
#      filename = the filename which was origionally scaned in "multisub"
#      sdate = start date, character string (##/##/##)
#      stime = start time, numeric (#)
#      edate = end time date, character string (##/##/##)
#      etime = end time, numeric (#)
#      ylimit = logical, if true the user determines the y-axis limits
#      flag = logical, if true precipitation is plotted
#      top,mid,bot = format parameters used by subplot()
#
      x <- xdump
      y <- filename
      sprint <- sdate
      eprint <- edate
      stp <- zfill(stime, 4)
      etp <- zfill(etime, 4)
      stprint <- paste(c(substring(stp, 1, 2), ":", substring(stp, 3, 4)),
            collapse = "")
      etprint <- paste(c(substring(etp, 1, 2), ":", substring(etp, 3, 4)),
            collapse = "")
      syear <- as.numeric(substring(sdate, first = 7, last = 8))
      eyear <- as.numeric(substring(edate, first = 7, last = 8))
      file <- substring(y, first = 7, last = 10)
      year <- substring(y, first = 11, last = 12)
      if(top == 1) {
            title <- paste("PIEZOMETER HOLE ", file)
      }
      else title <- ""
      ms <- as.numeric(substring(sdate, 1, 2))
      ds <- as.numeric(substring(sdate, 4, 5))
      ys <- as.numeric(substring(sdate, 7, 8)) + 1900
      sd <- julian(ms, ds, ys, origin = c(12, 31, 1983)) +
mt2msm(stime)/1440
      me <- as.numeric(substring(edate, 1, 2))
      de <- as.numeric(substring(edate, 4, 5))
      ye <- as.numeric(substring(edate, 7, 8)) + 1900
      ed <- julian(me, de, ye, origin = c(12, 31, 1983)) +
mt2msm(etime)/1440
            #
# Establish base
      if(file == "r1p2") {
            L <- 1.37
            E <- 102.38
      }
      if(file == "r2p2") {
            L <- 3.6
```

```
                E <- 105.13
        }
        if(file == "r3p1") {
                L <- 3.71
                E <- 107.94
        }
        if(file == "r3p2") {
                L <- 6.4
                E <- 107.97
        }
        if(file == "r4p1") {
                L <- 1.79
                E <- 110.81
        }
        if(file == "r4p2") {
                L <- 8.39
                E <- 110.7
        }
        if(file == "r5p2") {
                L <- 5.72
                E <- 114.51
        }
        if(file == "r6p2") {
                L <- 7.77
                E <- 117.41
        }
        if(file == "r6p3") {
                L <- 6.39
                E <- 117.39
        }
        bottom <- E - L
        ystuff <- c(bottom, bottom)
        xstuff <- c(sd, ed)#
# Convert Time list from min to days and put into a seperate array
        wholedate <- dates(x$Day, origin = c(12, 31, 1983))
        comb <- wholedate + (mt2msm(x$Time)/1440)
        sdate <- dates(sdate)
        edate <- dates(edate)
        ndays <- as.numeric(edate - sdate)
        Days <- x$Day
        x$Day <- x$Day + (mt2msm(x$Time)/1440)
        seg <- seq(along = x$Day)[x$Day > sd & x$Day < ed]
        minseg <- min(seg)
        maxseg <- max(seg) #
# Add the newly formed array to the x list
        x <- append(x, list(Sum = comb), after = 2)  #
# Convert the "x" list into a data.frame
        x <- as.data.frame(x)
        x <- x[minseg:maxseg,  ]  #
# Y-axis set if limits are not predetermined
        ymin <- min(min(x$Elev), min(model$head))
        ymax <- max(max(x$Elev), max(model$head))
        subplot(x = c(0, 1), y = c(mid, top), fun = {
                plot(x$Day, x$Elev, type = "n", xlab = "", ylab = "", main =
                        title, xlim = c(sd, ed), ylim = c(ymin, ymax), axes = F,
                        ...)
```

```
            axis(2, las = 2, mgp = c(2.2, 0.4, 0), cex = 0.5)
            lines(model$day, model$head, type = "l", lty = 3, col = 5)
            mtext(paste("BASE AT", bottom, "m  "), side = 3, line = -0.6,
                  cex = 0.4, adj = 1)
            mtext("Water Elev. (m)", side = 2, line = 2.5, cex = 0.5)
            if(top == 1) {
                  mtext(date(), side = 3, adj = 1, line = 0.2, cex = 0.5)
            }
            mtext(paste("START =", stprint, "on", sprint), side = 3, adj =
                  0, line = 0.5, cex = 0.4)
            mtext(paste("END     =", etprint, "on", eprint), side = 3, adj
                   = 0, line = 0.1, cex = 0.4)    #
# Fill in gaps in data
            diffs <- diff(x$Sum)
            startgaps <- seq(along = diffs)[diffs > 2]
            starts <- 1 + c(0, startgaps)
            ends <- c(startgaps, length(comb))
            for(i in seq(along = starts)) {
                  series <- starts[i]:ends[i]
                  xseries <- x$Day[series]
                  yseries <- x$Elev[series]
                  cseries <- x$Code[series]
                  lines(xseries[cseries != 3], yseries[cseries != 3], col
                         = 4)
            }
# Indicate manual measurements
            points(x$Day[x$Code == 3], x$Elev[x$Code == 3], pch = 3, cex =
                  0.5, col = 2)      #
# Box graph
            box()  #
# Add a legend
            print("Click to place upper-left corner of legend")
            legend(locator(1), c("tracer", "observed", "model"), marks = c(
                  -1
                  , 3, -1), mkh = 0.06, lty = c(1, -1, 3), col = c(4, 2,
                  5), background = -999999)
      }
      )
      subplot(x = c(0, 1), y = c(bot, mid), fun = {
            if(hrain == -666) {
                  plot(rain$days, rain$precip, axes = F, xlab = "", ylab
                         = "", type = "n", xlim = c(sd, ed))
            }
            else {
                  plot(rain$days, rain$precip, axes = F, xlab = "", ylab
                         = "", type = "n", xlim = c(sd, ed), ylim = c(
                        lrain, hrain))
            }
            mtext("Precip. (in)", side = 2, line = 2.5, cex = 0.5)
            segments(rain$days, rep(0, length(rain$days)), rain$days, rain$
                  precip, col = 5)   #
# Recontruct the x-axis
            if(ndays < 61)
                  axis.time2(sdate, edate)
            else axis.time3(sdate, edate, ndays, syear, eyear, flag)
            axis(2, las = 2, mgp = c(2.2, 0.4, 0), cex = 0.5)
```

```
                box()
        }
        )
}
```