University of Nevada, Reno

A Model for Simulating Soil-Zone Processes at the Regional Scale

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Hydrogeology

by

Wesley R. Henson

Dr. Richard G. Niswonger/Thesis Advisor

May, 2011

University of Nevada, Reno
Statewide • Worldwide

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

WESLEY R. HENSON

entitled

A Model For Simulating Soil-Zone Processes At The Regional Scale

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Richard Niswonger, Ph. D., Advisor

Greg Pohll, Ph. D., Committee Member

David Prudic, Ph. D., Graduate School Representative

Marsha H. Read, Ph. D., Associate Dean, Graduate School

May, 2011

ABSTRACT:

The soil zone has been described as the upper most region of the vadose zone where plant and soil processes enhance storage and permeability, providing a fast pathway for water and solutes to streams. The soil zone connects the land surface to the deeper unsaturated and saturated zones. Modeling of soil-zone processes has been used to gain understanding of watershed hydrologic processes. Currently MODFLOW does not simulate dynamic near-surface hydrologic processes such as, infiltration, hortonian runoff, dunnian runoff, and return flow. The Soil-Zone Flow Package (SZF) for MODFLOW is being developed to address these near-surface components for simulating watershed processes in the context of basin-scale groundwater-flow modeling. especially those processes that partition rainfall into evapotranspiration, runoff, and deep percolation.

In a series of test simulations, Richards' equation (RE) was compared with MODFLOW-SZF. Across a range of hydraulic conductivities and applied precipitation rates, MODFLOW-SZF achieved a good infiltration and interflow solution (0.014-0.052 RMSE) with only 6 grid cells and 0.11 seconds computation time compared to the 6250 grid cells and 40 seconds of computation time required for a stable solution to RE. MODFLOW-SZF solutions had negligible errors due to grid effects (<0.01 RMSE) however, RE solutions were sensitive to grid resolution (up to 0.12 RMSE). Errors associated with using the groundwater flow equation to represent soil-zone flow instead of RE were much lower (<0.052 RMSE) than the errors caused by grid effects in RE (0.12 RMSE, 0.5 m cell). This research suggests that the SZF Package will be an effective tool for efficiently representing soil-zone processes at the basin scale.

ACKNOWLEDGEMENTS:

There are several institutions and people I want to thank for making my research possible: the U.S. Geological Survey Office of Groundwater for funding this research, the U.S. Geological Survey Nevada Water Science Center who helped to give me the resources I needed to be successful, my thesis advisor and mentor Richard Niswonger for the skills he helped me build and knowledge he imparted, David E. Prudic for getting me out of my cubicle and exploring Hydrogeology in the field, Greg Pohll for introducing me to hydrogeology and research methods, and Christian Fey who has always supported me in all I do.

Table of Contents

Table of Figures:

Tables:

CHAPTER 1: INTRODUCTION

Numerical models that use process based physics are needed to better understand the effects of water resources development, climate change, and hydrologic controls on ecosystems (Harter and Hopmans, 2005). Departing from the traditional view of groundwater modeling as only simulating the baseflow component of a hydrograph, The Soil-Zone Flow (SZF) Package for MODFLOW can simulate the entire streamflow hydrograph, including runoff, interflow, and baseflow using spatially distributed parameters and physics-based equations.

This thesis presents a new enhancement to MODFLOW (SZF Package) for simulating near-surface hydrologic processes coupled to the three-dimensional groundwater-flow equation that is solved by MODFLOW. Additionally, this thesis describes the conceptualization of the SZF Package and presents comparisons to another numerical model that has been used to simulate watershed flow processes.

PROBLEM STATEMENT

The soil zone partitions precipitation into runoff, interflow, ET, and deep percolation. Understanding the role of the soil zone in groundwater recharge can help improve hydrologic models. The SZF Package for MODFLOW will simulate soil-zone processes using physics-based equations by partitioning precipitation into runoff, infiltration, evapotranspiration, subsurface-storm flow (interflow), and deep percolation.

Although the SZF Package simulates all of these near-surface hydrologic processes, this thesis focuses on evaluating the interflow component of watershed drainage.

PREVIOUS WORK

IMPORTANCE OF SOIL-ZONE FLOW

The permeability of the soil zone typically is enhanced relative to underlying sediments and rocks due to biologic activity and root zone processes. Chemical leaching from the soil zone may also reduce the permeability of sediment and rock beneath the soil zone, further increasing the contrast in permeability. The vadose zone, which often includes the soil zone, regulates flow to and from groundwater (Harter and Hopmans, 2005). Soils serve several important functions. Soils act like sponges soaking up rainwater and limiting runoff, which in turn affect deep percolation, groundwater recharge and flood-control potential in urban and natural areas. Soils store and release water, air, and nutrients for plants and animals to use. Soils sequester carbon, filter water, as well as buffer, degrade, immobilize, detoxify, and trap pollutants (Scheyer & Hipple, 2005). More than 95% of water in streamflow has passed over or through soils before reaching the channel network (Kirkby, 1988).

WATERSHED HYDROLOGY

Watershed hydrology is the study of hydrologic processes that occur within watersheds such as infiltration, runoff, interflow, and subsurface flow. Knowledge of relevant processes is integral to understanding watershed hydrology (Beven, 1989). Therefore, simulation of soil-zone processes such as infiltration, runoff and interflow will improve the accuracy of basin-scale hydrologic models. The SZF Package will simulate these processes.

Developing an understanding of watershed processes is important in the context of basin-scale modeling. The relative importance of each hydrologic process in a particular region is affected by watershed characteristics, including the type of climate, geology, physiography, soils, vegetation and land use (Dunne, 1978).   For example, regions with semi-arid to arid climates, runoff processes are less common, but in wetter climates these processes dominate watershed drainage. In order to simulate watershed-hydrologic processes, it is important to delineate how watershed characteristics vary with, time, and space. The SZF Package is a physically-based distributed parameter model that considers areally distributed soil-zone properties and climate forcing to consider spatial variability in climate, geology, physiography, soils, vegetation and land use.

A conceptual cross section of a watershed (fig. 1) is divided into three hydrogeologic layers: a soil-zone layer, an unsaturated zone layer, and a groundwater layer. This thesis will focus on processes occurring in the soil zone. Precipitation is aerially distributed across the watershed and collects on the surface. Once precipitation (rain) contacts the soil surface, the processes driving infiltration and runoff generation begin. Precipitation is then partitioned into runoff and infiltration. Evapotranspiration (ET) of soil moisture is removed from the soil zone by vegetation and bare-soil evaporation. Infiltration and runoff are functions of potential ET, antecedent-soil conditions, soil type, soil composition, degree of vegetation, and macropore distribution. The relationship between runoff and infiltration directly impacts how precipitation is partitioned between the surface and subsurface, determining how water is transmitted between the surface and subsurface layers such as the soil zone, unsaturated zone, and groundwater.

Groundwater perches in the soil zone when deep percolation is less than infiltration at land surface. When groundwater perches in the soil zone, horizontal flow occurs downslope along the restrictive layer (interflow) and is routed to a stream channel or percolates downward to the regional aquifer where the restrictive layer is absent (Scheyer & Hipple, 2005; Dunne, 1978). Even though interflow is typically a temporary storm-related phenomena, it can represent a large component of watershed drainage in watersheds. The proportion of streamflow that is comprised of interflow has been estimated to be 2 to 34% of precipitation and 6 to 86% of total runoff, the average being 10 and 23%, respectively (Hewlett and Hibbert, 1967).

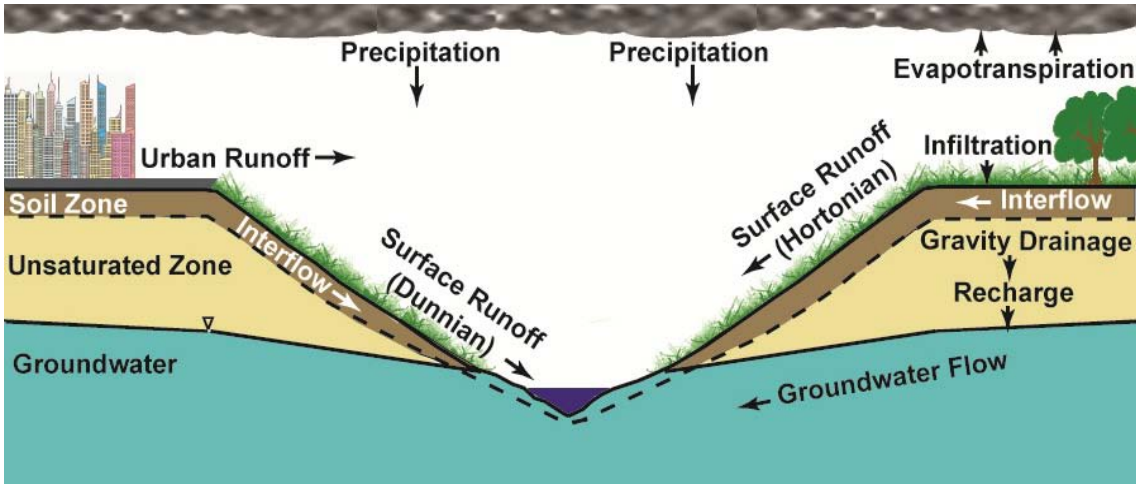

Figure 1: Conceptual watershed cross section

VARIABLY SATURATED FLOW

Richards (1931) derived an equation to describe unsaturated flow in three dimensions, using the Buckingham-Darcy formulation for unsaturated flow (Buckingham, 1907). Richards' equation calculates variably-saturated flow using a factor of proportionality relating the forces of gravity and capillary pressure gradients to the flow rate through porous media, called unsaturated hydraulic conductivity. Unsaturated hydraulic conductivity can be calculated as a function of either capillary pressure or water content. Celia et al. (1990) showed that the mixed form of Richards' equation with special treatment of the time derivative provided the best results for simulating variably-saturated flow. Many researchers have derived relationships between unsaturated hydraulic conductivity and water content or matric potential (e.g., Brooks and Corey, 1966; Mualem, 1976, Van Genuchten, 1980). The Brooks-Corey hydraulic conductivity function is used in the SZF Package to calculate the unsaturated hydraulic conductivity as

$$K(\theta) = K_s \left[\frac{\theta-\theta_r}{\theta_s-\theta_r}\right]^{\epsilon},$$

where K(θ) is the unsaturated hydraulic conductivity as a function of water content, θ, Ks is the saturated hydraulic conductivity, θ_r is the residual water content, θ_s is the saturated water content, ϵ is the Brooks-Corey shape factor.

Currently there is no unified, universally accepted mathematical representation of regional flow or transport in the unsaturated zone (Harter and Hopmans, 2005). Richards' equation is the universally accepted mathematical model but it has less physical meaning at the regional scale because the constitutive relationships that are used in the model were established for samples analyzed in the laboratory (<1 m length scale) (Vogel and Ippisch, 2008; Harter and Hopmans, 2005). Although Richards' equation has mostly been verified at the field scale, it is the current standard in the absence of a regional scale mathematical model to represent variably-saturated flow in three dimensions. The validity of Richards’ equation in the laboratory and small scale experiments does not guarantee that the same equation will describe the average behavior of a large scale system equally well (Mantaglou, 1992). Harter and Hopmans (2005) state that neither three dimensional numerical models based on field scale parameterization of Richards’ equation nor upscaled stochastic equations are practical because of the complexities and computational requirements associated with numerical solutions of these equations. Further, Jensen and Mantoglou (1992) showed unsaturated-flow paths essentially become vertical after upscaling to large grid cells (100 m), providing precedence for using 1-D unsaturated-zone flow equations for regional-scale simulations. Accordingly, this was the approach used in the development of the SZF Package for MODFLOW.

A major factor that contributes to the complexity of regional scale models is the large degree of spatial variability exhibited by soil formations. While we can discretize governing partial differential equations at a scale fine enough to account for local fluctuations in soil properties, adequate data is not available to represent intrinsic spatial variability (Montoglou, 1987). Length correlation scales in the soil zone are on the order of 5-10 cm (Jensen and Montoglou, 1992), infiltration solutions are stable in simulations with vertical discretizations on the order of 0.1-1cm (Downer and Ogden, 2003), and assumptions that define the spatial scale of Richards' equation break down at approximately 4 m or less (Vogel and Ippisch, 2008); therefore, creating a transient 3-D Richards' equation regional-scale model may require billions of model nodes, and the use of multiple computers to obtain results faster than real time (Harter and Hopmans, 2005).

Some models explicitly account for vadose-zone processes but the issue of field versus basin scale measurement and computational requirements for basin-scale models still persists. The SZF Package uses an analytical solution to Richards' equation (Smith and Parlange, 1978; Smith and Corradini, 1993) that does not require vertical discretization, reduces computation time, and increases numerical stability for regional scale models. Analytical models assume vertical homogeneity in the soil zone. However, this assumption is not typically limiting in regional-scale models due to a lack of data that describe soil layering that is laterally extensive to traverse an entire model cell.

INFILTRATION

Many models have been developed to simulate infiltration (Green and Ampt, 1911); Horton, 1933; Phillip, 1957; Smith and Parlange, 1978; and Smith and Cordini, 1993). Infiltration models typically consider both capillary and gravitational forces that affect the flow through unsaturated-porous media.

Infiltration is complex even under assumed conditions of homogeneity and uniform initial water content (Mein and Larson, 1973). Infiltration capacity is a function of the saturated hydraulic conductivity, capillarity of the unsaturated soil, initial water content, and the cumulative infiltration. Generally, the infiltration capacity is a function of rainfall characteristics such as intensity, duration, and drop size; and soil characteristics such as texture, structure, depth, and clay mineralogy (Dunne, 1978). Infiltration and runoff are interrelated by the time of ponding, which describes when the precipitation rate exceeds the infiltration capacity of the soil. Following the time of ponding, excess precipitation becomes runoff.

RUNOFF

Runoff can be modeled as sheet flow until it channelizes into riffles. When the rainfall intensity exceeds infiltration capacity, surface depressions are filled first. This is referred to as depression storage. Once depression storage has filled, excess precipitation then spills over into overland flow (Dunne, 1978). Two types of overland flow are Hortonian runoff and Dunnian runoff. Hortonian runoff was first described by Horton (1933) as rainfall or snowmelt rates that exceed the infiltration rate at the soil surface (Kirkby, 1988). Excess infiltration ponds on the surface, and eventually moves down slope to a stream channel if the runoff does not re-infiltrate before reaching a stream.

Dunnian runoff, or saturation excess runoff, occurs when the subsurface storage capacity is filled (Kirkby, 1988), or when precipitation lands on the saturated areas or groundwater discharge areas, both are typically adjacent to streams. The variable source area concept (fig. 2) has been used to explain some differences between the predicted volume of storm runoff and the actual runoff because infiltration capacity at any time is only exceeded in part of a drainage basin. Betson (1964), Hewlett and Hibbert (1967), Dunne and Black (1970), and Dunne (1978) have shown that the saturated zone (variable source area) adjacent to the stream increases as the storm duration increases.

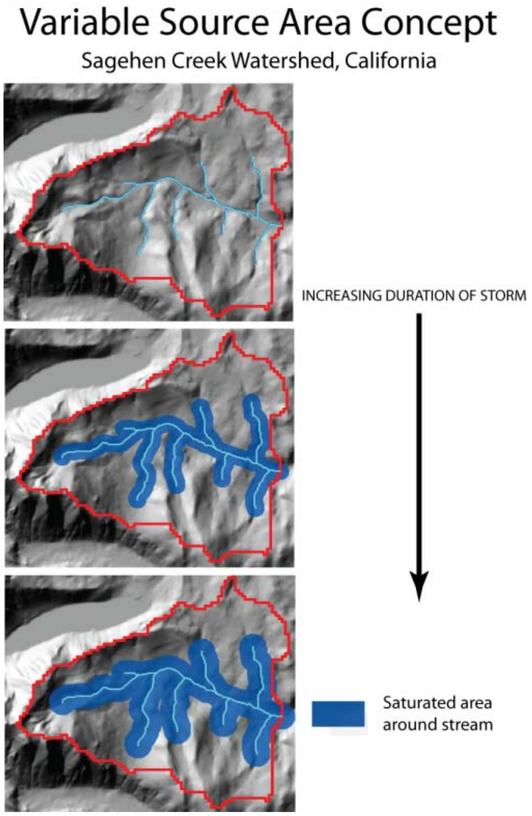

Figure 2: A hypothetical example of changes in the variable source area during a storm

The kinematic wave approximation to the Saint-Venant equations is widely used to describe open channel flow [MIKE-SHE (Refsgaard et al., 1995), HEC1 (US Army Core of Engineers, 1973), KINEROS (Woolhiser et al., 1990)]. Kinematic waves govern when internal and pressure forces are not important to the movement of the wave (i.e., the hydraulic slope is equal to the land surface slope). Typically, overland flow will be shallow, such that diffusive forces do not dominate. In the SZF Package, a quasi-2-D kinematic wave approximation is used to route overland flow according to predefined cell-by-cell pathways developed based on the slope of land surface (fig. 3).

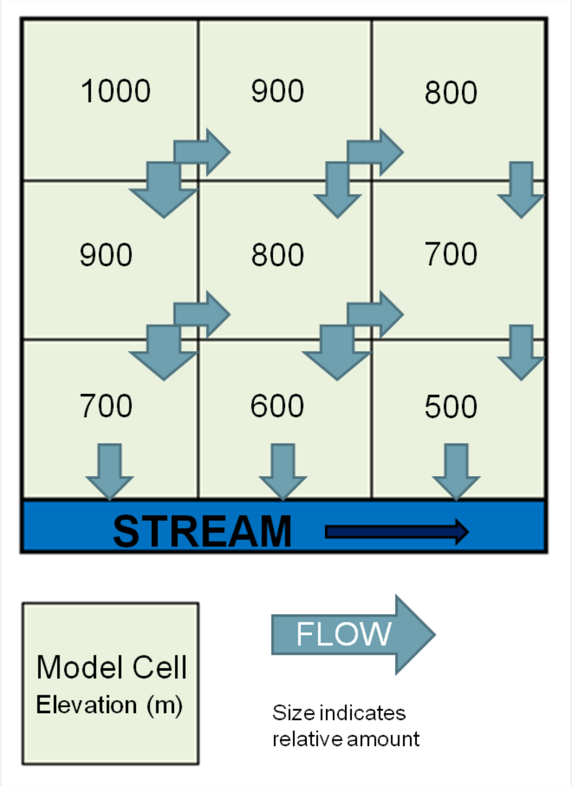

Figure 3: Example of quasi-2D cascading overland flow

DEVELOPMENT OF THE PERCHED SOIL ZONE AQUIFER

Development of a perched soil zone aquifer during a storm is illustrated in figure 4. This figure also shows macropores created by roots and faunal burrows. Development of macropores results in a higher effective hydraulic conductivity in the soil layer relative to the underlying sediment and rocks. A large fraction of infiltrating water flows into macropores, and reaches a stream through preferential-flow paths (Kirkby, 1988). Water can flow at a greater rate through macropores relative to water flowing through pores in the soil. Beven (1981) provides effective hydraulic conductivity values for soil zones in different regions that range between 1.74 m/day (Ohio) and 1,184 m/day (New Zealand).

A perched water table is a common occurrence in the soil zone because many types of restrictive layers tend to form beneath soil zones, including alluvial hardpan, unweathered rock, or a clay rich layer. The rate of rainfall or snowmelt is also an important factor that controls the formation of perched water tables, and the resulting amount of interflow. If the duration and rate of precipitation remains sufficiently high, such that the infiltration rate exceeds the percolation rate through the base of the soil zone, then interflow occurs (Dunne, 1978).

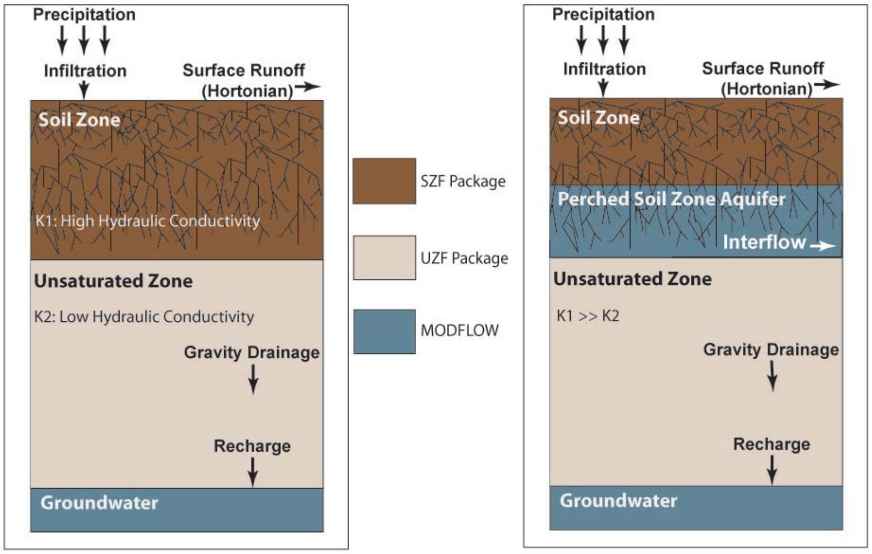

Figure 4: Conceptual model of the development of the perched soil-zone aquifer

CHAPTER 2: RESEARCH METHODS

Soil-zone flow is simulated by establishing a thin surface layer in MODFLOW with hydraulic properties representative of shallow soils (i.e., high effective hydraulic conductivity). The soil-zone layer is parameterized using available soil data including soil-zone thickness, as well as soil textural classifications that provide estimates of porosity and saturated, residual, and initial water contents, and air entry pressures. While connected macropores within the surrounding porous soil matrix may have different storage characteristics than the porous matrix itself, this model does not separately account for storage in these two domains. Hydraulic conductivity is increased to represent effective properties of both the macropores and the porous matrix in the soil zone. This enhanced permeability is what facilitates the high infiltration in the soil zone leading to interflow.

The SZF Package is combined with MODFLOW-NWT (Niswonger et al., 2011) to simulate three dimensional saturated groundwater flow, the UZF1 Package to simulate vertical unsaturated flow beneath the soil zone (Niswonger et al., 2006), and the SFR2 Package (Niswonger and Prudic, 2005) to simulate flow in streams. MODFLOW-NWT is a Newton formulation of MODFLOW-2005 (Harbaugh, 2005) that provides robust simulations of aquifers drying and rewetting, which is necessary for simulating groundwater flow in the soil zone. The SZF Package was programmed in the FORTRAN 95 programming language to interface with MODFLOW and its associated packages.

SOIL ZONE FLOW COMPUTATION

INFILTRATION AND RUNOFF

Distributed precipitation is applied to the surface layer. Precipitation applied to each MODFLOW cell is partitioned into surface runoff and infiltration using equations developed by Smith and Parlange (1978). The infiltration excess is routed as overland flow to adjacent down-slope cells using the kinematic-wave equation. Infiltration is partitioned into interflow or deep percolation based on the solution of the groundwater flow equation. Interflow is simulated using the solution to the unconfined groundwater flow equation that is solved by MODFLOW. Evapotranspiration is simulated in the soil zone as a function of soil saturation.

The infiltration rate can be computed from the following equation by Smith and Parlange (1978):

$$f_c = K_S \left[1 + \frac{\propto}{\exp\left(\frac{I' \propto}{\Delta\theta G}\right) - 1} \right], \tag{1}$$

where f_c is the infiltration rate, K_S is the saturated hydraulic conductivity, G is the integral capillary drive across wetting front, α is the Smith-Parlange alpha dimensionless shape factor which ranges between 0.5 and 1.0, I' is the cumulative infiltration depth, and Δθ is the change in water content.

The integral of capillary drive at the wetting front (G) is calculated as:

$$G = \int_{\theta_i}^{\theta_0} K(\theta)\, d\theta = \frac{K_s(\theta-\theta_r)\left(\frac{\theta_r-\theta}{\theta_r-\theta_s}\right)^{\epsilon}}{1+\varepsilon}, \qquad (2)$$

where K(θ) is the unsaturated hydraulic conductivity as a function of water content (θ), θ_r is the residual water content, θ_i is the initial water content, θ_0 is the water content for the current iteration, θ_s is the saturated water content, ϵ is the Brooks-Corey shape factor.

Time of ponding is computed from the following equation (Smith-Parlange, 1978):

$$t_p = t - \left\{ \frac{(I-I_P) - \frac{\Delta\theta G K_s}{K_d} ln \left| \frac{exp\left(\frac{\alpha I'}{\Delta\theta G}\right)-1+\gamma}{\exp\left(\frac{{I_p}'\propto}{\Delta\theta G}\right)-1+\gamma} \right|}{K_s(1-\alpha)} \right\} \qquad (3)$$

where t is time, t_p is time of ponding, α is Smith-Parlange alpha, I is the cumulative infiltration, I_p is the ponded infiltration, K_d is the difference between $K(\theta_i)$ and $K(\theta_2)$, θ_2 is water content at time=t_2, γ is a dimensionless shape factor equal to $\alpha * K_s/K_d$, and I'_p has the following relationship;

$I'_p = (I_p) - K_i * t_p$.

The cumulative infiltration at time of ponding is computed as the rainfall rate multiplied by the time of ponding.

Soil moisture in the soil zone between rainfall events is redistributed using the following equation by Smith-Corradini (1993). This ensures that the next time-of-ponding calculation takes into account updated soil moisture condition. The equation can be written as:

$$\frac{d\theta_0}{dt} = \frac{\Delta\theta_{io}}{{I_z}'}\left\lfloor R_h - K_i - \left(K(\theta_0) + \frac{\beta p K_s \Delta\theta_{io} G}{I'_z}\right)\right\rfloor , \tag{4}$$

where θ_0 is water content at land surface, $\Delta\theta_{io}$is the difference between θ_0 at land surface and θ_i , I'_z is the infiltrated water during hiatus, R_h is the rainfall rate during hiatus, K_i is the hydraulic conductivity at θ_i (K(θ_i)), β is the Smith-Parlange Beta dimensionless shape factor which ranges between Π/4 and 1.0, p is the dimensionless wetting front shape factor which ranges from 1.0-2.0, K_s is the saturated hydraulic conductivity, and G is the integral capillary drive across the wetting front.

OVERLAND FLOW

Surface runoff for each MODFLOW finite-difference cell is calculated and routed to the nearest channel using a Quasi 2-D kinematic-wave approximation. Discrete 1-D pathways are defined based on relative elevations defined by a digital elevation model (DEM). Four pathways can be simulated for each grid cell, similar to a 2-D finite-difference formulation. Kinematic waves then route the overland flow downhill along these paths to stream channels. Runoff that flows from one model cell to multiple downslope cells is partitioned according to the relative slope among the downslope cells. If surface runoff flows to a downstream cell that has more infiltration capacity, then all or a portion of the surface runoff re-infiltrates.

Hortonian runoff is calculated using the time-of-ponding relationship between the precipitation rate and the infiltration rate. If precipitation is applied to a MODFLOW cell that is already saturated, Dunnian saturation excess runoff will be generated. If interflow moving through the soil zone encounters an area of saturation then return flow is routed to the surface and overland flow continues to the next model cell. If precipitation is applied to an impervious surface, such as pavement, the precipitation is automatically added to overland flow. The 1-D kinematic wave approximation (Lighthill and Whitham , 1955) is used to route overland flow for each downstream connection as:

$$\frac{1}{C}\frac{\partial q}{\partial t}+\frac{\partial q}{\partial x}=\frac{Q(x,t)}{l} \quad , \tag{5}$$

where C is Chezy’s coefficient, x is 1-D flow path direction, l is cell length along columns or row (depends on flow direction out of each cell), $Q(x,t)$ is volume flux, and q is the flow velocity. Chezy’s relation is used to calculate mean flow velocity as:

$$q = C\sqrt{RS_0} \ , \tag{6}$$

where R is the hydraulic radius, S_0 is land surface slope, and C is Chezy's coefficient. Equation 6 is solved using a 4-point finite-difference scheme (Preissmann, 1961)

INTERFLOW AND GROUNDWATER FLOW

Interflow and vertical flow beneath the base of the soil zone is calculated according to the groundwater-flow equation solved by MODFLOW-NWT. If the groundwater head is at or above the base of the soil zone, then the soil zone cell is treated as an aquifer. If the groundwater head is below the base of the soil zone, then water is routed through the deeper unsaturated zone by the UZF1 Package. Interflow through the soil zone and regional groundwater flow is modeled using the three-dimensional groundwater-flow equation (Harbaugh, 2005; Niswonger et al., 2011):

$$\frac{d}{dx}\left(K_{xx}\frac{dh}{dx}\right)+\frac{d}{dy}\left(K_{yy}\frac{dh}{dy}\right)+\frac{d}{dz}\left(K_{zz}\frac{dh}{dz}\right)+W=S_s\frac{dh}{dt}\,, \qquad (7)$$

where K_{xx}, K_{yy}, K_{zz} are hydraulic conductivity in the x, y, and z directions, h is the groundwater head, W represents any sources or sinks of water, and S_s is specific storage. Volume averaging of equation (7) results in the conservative form of the groundwater flow equation (Harbaugh, 2005; Markstrom et al., 2008):

$$\Sigma Q_L = S_s\frac{dh}{dt}V \qquad (8)$$

where Q_L is the volumetric flow rate into a cell (negative value is a volumetric flow out of cell, in cubic length per time), and V is the volume of the cell in cubic length.

IMPLEMENTATION OF INTERFLOW BOUNDARY INTO MODFLOW

Interflow to streams is implemented through a head-dependent seepage face in each cell that contains a stream. Interflow is calculated on the basis of the head in the model cell, the stream elevation, and the stream stage. The interflow calculation considers the feedback between stage, streamflow, and interflow; which provides numerical stability.

The interflow boundary is implemented into the MODFLOW-NWT as a nonlinear stress. Niswonger et al. (2011) provide a detailed explanation of how a nonlinear stress is added to MODFLOW-NWT. The function added to MODFLOW-NWT for the calculation of interflow is

$$f(h) = K_{if}\ Sn\ T\ (B - h), \tag{9}$$

where, K_{if} is the interflow conductance term, $Sn * T$ is a function which smoothes conductance as a function of saturated thickness in the cell; B is the bottom elevation of the soil-zone cell if the stream stage is not above the soil-zone base, or the stream stage elevation, if it is above the soil zone base; h is the head in the soil zone in the model cell which contains the stream. The interflow conductance term is

$$K_{if} = Hk\ \frac{S*h}{\frac{1}{4}*(delr+delc)}\ , \tag{10}$$

where Hk is the horizontal hydraulic conductivity of the soil zone, S is two times the stream length of the reach in the model cell for which interflow is calculated, h is the head in the soil-zone model cell that contains the seepage face to the stream, $delr$ is the model cell dimension in the row direction, and $delc$ is the model cell dimension in the column direction.

Twice the stream length is multiplied by h to calculate the facial area for interflow because the stream length is the distance over which interflow will occur on each side of the stream channel. Half of the average of the cell dimensions, $delr$ and $delc$ represents the distance over which the hydraulic gradient is calculated between the edge of the model cell and the model node. The average of $delr$ and $delc$ is used because streams meander within model cells, and are sub grid-scale features for which obtaining an exact distance from the model node to each point in the stream reach is not available.

The Newton formulation is

$$\frac{d}{dh}(f(h))\Delta h = -f(h) \text{ ,}$$

where in this example, $f(h)$ is equation (9), and $\Delta h = h_k - h_{k-1}$ where, the subscripts k and $k-1$ designate values of h for nonlinear iterations k and $k-1$. Implementing the Newton formulation, equation (9) becomes

$$\frac{\partial}{dh}\left[K_{if}\ Sn\ T\ (B-h)\right]\Delta h = -\left[K_{if}\ Sn\ T\ (B-h)\right]. \tag{11}$$

Evaluating the derivative and distributing Δh results in

$$\left[K_{if}\frac{\partial Sn}{dh}(B-h_{k-1})-K_{if}\,Sn\,T\right]h_k = \tag{12}$$

$$\left[K_{if}\frac{\partial Sn}{dh}(B-h_{k-1})-K_{if}\,Sn\,T\right]h_{k-1}-\left[K_{if}\,Sn\,T\,(B-h_{k-1})\right],$$

To implement the interflow into MODFLOW-NWT, equation (12) is added to the discretized groundwater flow equations.

TEST SIMULATION

A hypothetical model, similar to the model used by Vauclin et al. (1979), was used to compare results from a MODFLOW-SZF simulation with a two dimensional form of Richards' equation (VS2DT; Healy, 1990). The test simulation has two hydrogeologic units (table 1) represented in two model layers consisting of a highly permeable layer, which represents the soil zone, overlying a lower permeable clay layer, which represents the regional aquifer and unsaturated zone (fig. 5). Hydraulic conductivity of the soil zone ranged from a maximum value of 250 m/d to a minimum of 0.05 m/day in several simulations (table 2). The vertical and horizontal hydraulic conductivity were parameterized to represent the extensive preferential flow path networks typically present in the soil zone.

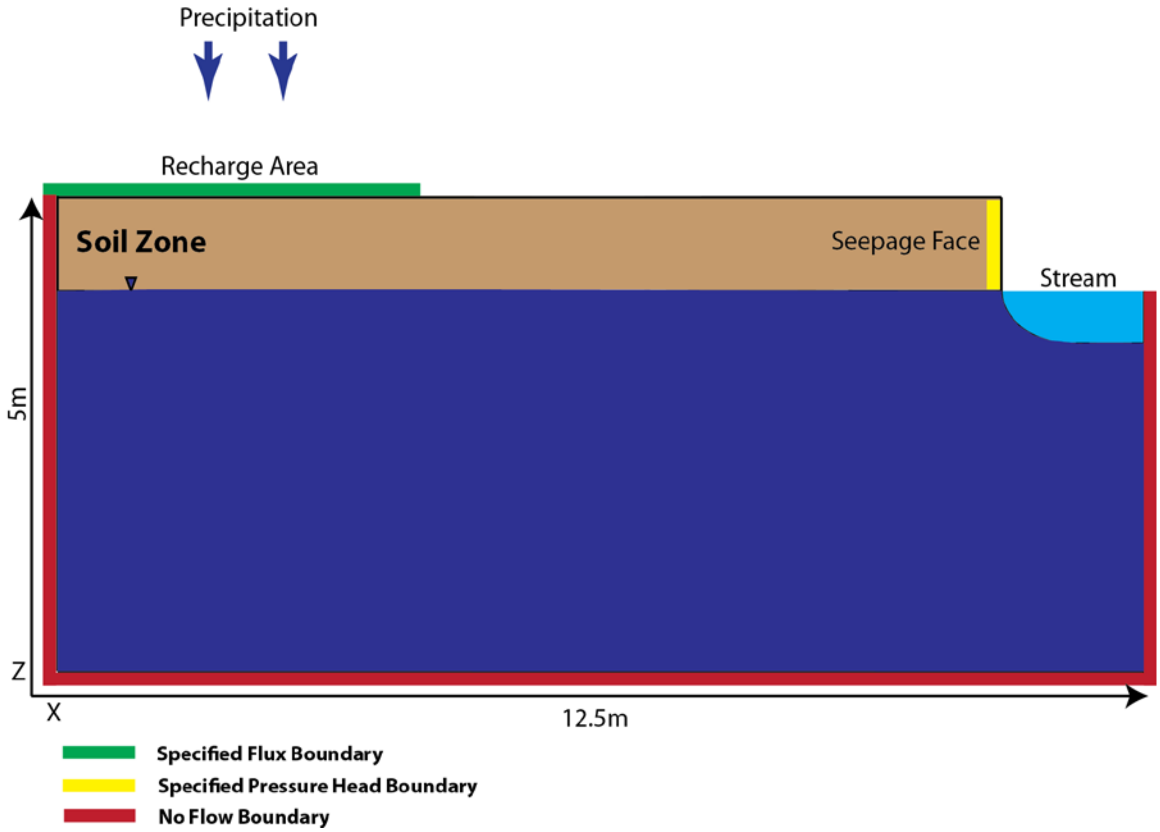

Figure 5: Model domain for test simulations comparing results from MODFLOW-SZF with VS2DT.

The cross-sectional model was 12.5 m in the x-direction, 5 m in the z-direction, and 1 m in the y-direction. At the distal edge of the cross section, interflow occurs as a seepage face to the stream channel. The same hydraulic properties were used for both simulations (table 1) using values typical for each hydrogeologic unit. The water table was set equal to the elevation at the base of the soil zone. Constant precipitation was applied on the soil-zone surface over a 5 m^2 area adjacent to the model boundary, opposite the interflow boundary for the entire 20-day simulation. In the model comparison, 0.25, 0.5, and 1.0 day timesteps were evaluated in MODFLOW-SZF and in

VS2DT adaptive time steps were used which range from a minimum size of 0.005 days to a maximum of 1 day.

Table 1: Input parameters for SZF and VS2DT in simulation 1

Parameter	Layer 1	Layer 2
Horizontal Hydraulic Conductivity (m/d)	250	0.002
Vertical Hydraulic Conductivity (m/d)	0.5	0.002
Specific Storage	1.0E -5	1.0E -5
Specific Yield	0.35	0.15
Porosity	0.45	0.45
Residual Moisture Content	0.10	0.30
Air Entry Pressure	-0.05	-0.05
Lambda	2	2
Epsilon	4	4

The focus for this thesis will be on model results from four simulations using the SZF Package combined with MODFLOW-NWT (referred to as MODFLOW-SZF). These simulations test the effects of varying: 1) grid dimensions, 2) hydraulic conductivity values, and 3) rainfall rates. Simulation 1 used the input parameters listed in table 1 to evaluate the effectiveness of using the groundwater-flow equation to represent interflow in MODFLOW-SZF by comparing with Richards' equation in VS2DT. Additionally, the effects of grid-cell size was varied by testing grid-cell dimensions ranging between 0.05 and 0.5m in VS2DT, and horizontal cell dimensions ranging between 0.25 and 5 m with vertical discretization ranging between 0.25 to 1 m in MODFLOW-SZF. Testing the effects of grid-cell dimensions in VS2DT provided a means of establishing a “correct” solution for comparison to MODFLOW-SZF for the hypothetical interflow problem. The grid-cell dimensions did not have an effect on the VS2DT simulation results for cell dimensions less than 0.1 m. Accordingly, the VS2DT model with a grid cell dimension equal to 0.1 m or smaller was used to represent the "correct" solution for evaluating the accuracy of MODFLOW-SZF. Three additional simulations were run to examine sensitivity to vertical and horizontal hydraulic conductivity and applied rainfall rate. VS2DT does not simulate runoff; therefore, runoff was not a part of the test simulation comparison. Table 2 shows a summary of the parameters that were varied in all of the simulations..

Table 2: Summary of parameters varied in simulations [HK: horizontal hydraulic conductivity; VK: vertical hydraulic conductivity]

Simulation	HK (m/day)	VK (m/day)	Precipitation rate (m/day)	Total Infiltration (m^3)	Time (Days)
1	250	0.50	0.04	4.00	20
2	25	0.05	0.04	4.00	20
3	250	0.50	0.08	8.00	20
4	250	0.50	0.12	12.00	20

The effects of grid-cell dimensions on the amount of interflow simulated by MODFLOW-SZF was evaluated by comparing results from the three different grid-cell configurations. Three different grid-cell configurations were used in MODFLOW-SZF to test the effects of grid-cell dimensions on the interflow solution. Configuration 1 was a coarse resolution model with uniform 5 m spacing in the x-direction for both layers. Layer 1 was 1 m thick and layer 2 was 4 m thick. Configurations 2 and 3 represent finer resolution models. Configuration 2 had uniform spacing of 0.25 m in the x-direction for both layers with the same vertical resolution as configuration 1. Configuration 3 had the finest resolution with a grid-cell dimension of 0.25 m, in the x and z directions in both layers.

Four different grid-cell configurations were used in VS2DT to test the effects of grid-cell dimensions on the interflow solution. Uniform grid-cell dimensions in the x and z directions were used in all VS2DT simulations equal to 0.05 m, 0.10 m, 0.25 m, and 0.50 m, configurations 1, 2, 3 and 4 respectively. MODFLOW-SZF and VS2DT solutions were compared based on timing of the vertical and horizontal wetting fronts, flow at the seepage face, and development of a water table.

Sensitivity analyses were done (referred to as simulations 2, 3, and 4) based on MODFLOW-SZF configuration 1 and VS2DT configuration 2. Sensitivity analyses were used to evaluate the effect of various precipitation rates and hydraulic-conductivity values on the timing and amount of interflow. The simulations were designed to analyze and compare the timing of interflow and percent difference of total interflow to the stream when varying hydraulic conductivity (simulation 2) and precipitation (simulations 3 and 4).

Model sensitivity to varying hydraulic conductivity was evaluated by comparing results between simulations 1 and 2, in which the ratio between vertical and horizontal hydraulic conductivity remained constant at 0.002 and horizontal and vertical hydraulic conductivity of the soil zone was decreased by an order of magnitude (table 2).

Sensitivity to precipitation rates was evaluated in simulations 1, 3, and 4, where applied precipitation rates increased from 0.04 m/d in simulation 1, to 0.08 m/d in simulation 2, and to 0.12 m/d in simulation 3.

In all of the test simulations, the interflow percent difference was calculated as the difference in total interflow between models divided by the average interflow. Root-mean-squared error (RMSE) was calculated as the square root of the average of the squared differences between model simulations.

CHAPTER 3: RESULTS AND DISCUSSION

DISCRETIZATION

Figures 6 and 7 show the effects of different grid-cell configurations. All MODFLOW-SZF grid-cell configurations had almost identical interflow breakthrough curves (fig. 6). This result indicates that the MODFLOW-SZF interflow solution is not sensitive to varying grid-cell dimensions over the range evaluated in these simulations. Additionally, time discretization had no effect on the MODFLOW solutions for time steps equal to and smaller than one day.

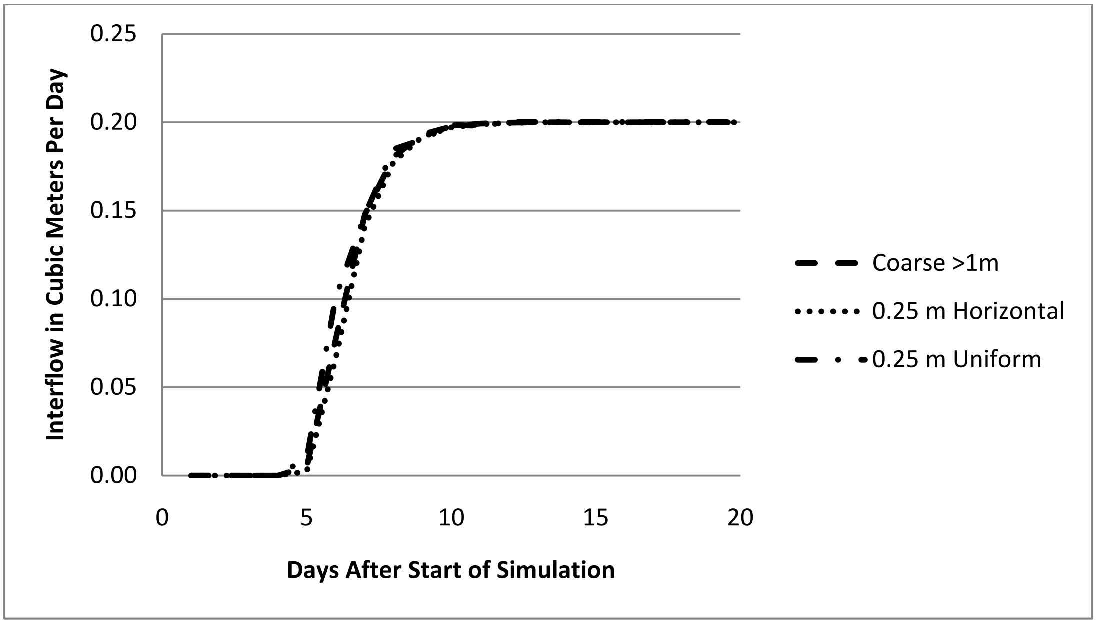

Figure 6: Effects of varying grid-cell dimensions on interflow simulated by MODFLOW-SZF

The interflow percent difference and RMSE for the smaller grid-cell configurations (0.25 x 1 m, and 0.25 x 0.25 cm) was calculated relative to the largest grid-cell configuration (5 x 1 m) because evaluation of MODFLOW-SZF interflow solutions at coarser grid resolution is the focus of this research. The interflow percent differences were low for all MODFLOW-SZF configurations, ranging between 0.92 and 1.10%. RMSE values were also low, ranging between 0.003 and 0.01.

Table 3: Summary of effects of varying grid-cell dimensions on interflow simulated by MODFLOW-SZF

Configuration	Discretization	Total Interflow (m^3)	Interflow % Difference	RMSE
1	5 x 1 m	2.80	-	-
2	0.25 X 1 m	2.77	0.92%	0.003
3	0.25 x 0.25 m	2.83	1.10%	0.01

The effects of varying grid-cell dimensions were much more pronounced in VS2DT (fig. 7 and table 4). Grid-cell dimensions affected both the timing and the shape of the interflow-breakthrough curves.. Simulations that used uniform grid-cell dimensions of 0.25 m and 0.5 m resulted in a 5 and 10-day delay, respectively, in the onset of interflow relative to the smaller grid-cell configurations (0.1 and 0.05 m; fig. 7). Reducing the grid-cell dimensions lower than 0.1 m had a negligible effect on the interflow simulated by VS2DT.

The interflow percent difference and RMSE for each VS2DT grid-cell configuration (uniform 0.1, 0.25, and 0.50 m) were calculated relative to the model with the smallest grid-cell dimension (0.05 x 0.05 m). The interflow percent difference ranged between 0.02% and 82.6%; the RMSE ranged between 0.008 and 0.12. As expected. the error in the interflow was larger for larger grid-cell dimensions.

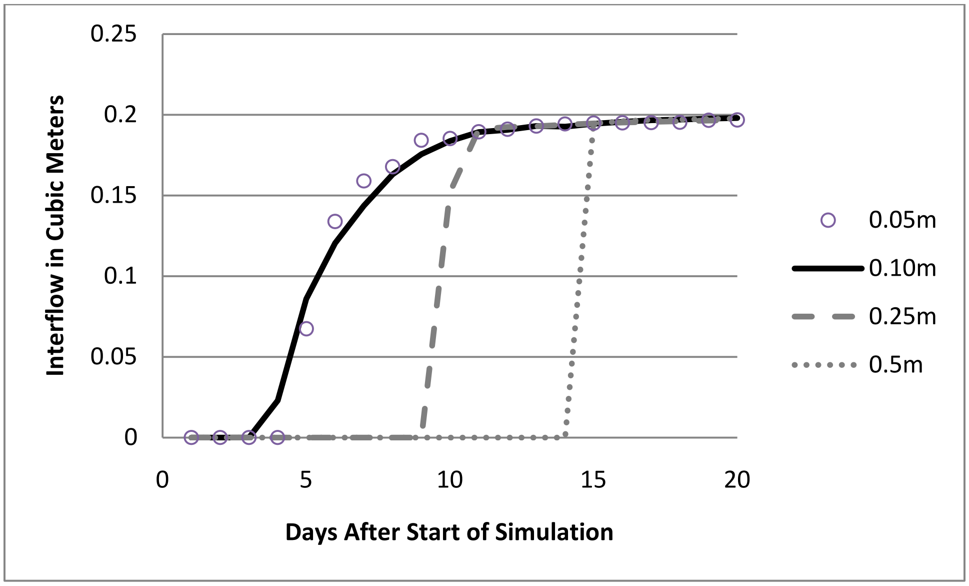

Figure 7: Effects of grid-cell dimensions on interflow simulated by VS2DT

Table 4: Summary of effects of grid-cell dimensions on interflow simulated by VS2DT

Configuration	Discretization	Total Interflow (m^3)	Interflow % Difference	RMSE
1	0.05 x 0.05 m	2.83	-	-
2	0.1 x 0.1 m	2.84	0.02%	0.008
3	0.25 x 0.25 m	2.09	30.09%	0.074
4	0.50 x 0.50 m	1.18	82.60%	0.12

Grid-cell dimensions of 0.05 and 0.1 m provided the most stable VS2DT solutions, and these results are in agreement with results presented by Jensen and Montglou (1992). The 0.1 m discretization was the largest cell size that preserves a stable solution. For these reasons, the 0.1 m uniform grid VS2DT model is the model that was compared with MODFLOW-SZF for all of the sensitivity analyses.

COMPARISON OF MODFLOW-SZF AND VS2DT

The effects of grid-cell dimensions on model computation time were compared between MODFLOW-SZF and VS2DT (table 5). VS2DT with a 0.1 m grid-cell dimension required a total of 6,250 grid cells and completed computation in 40 seconds. In contrast, MODFLOW-SZF required only 6 grid cells to obtain an accurate solution and completed computation in 0.11 seconds

Table 5: Summary of model computation times for MODFLOW-SZF and VS2DT

	MODFLOW		**VS2DT**	
Configuration	1	3	2	3
Discretization	5 x 1 m	0.25 x 0.25 m	0.1 x 0.1 m	0.25 x 0.25 m
# Model Nodes	6	1020	6,250	1,000
Computation Time (seconds)	0.11	22.5	40	10

Figure 8 shows the comparison between MODFLOW-SZF configuration 1 and VS2DT configuration 2. For simulation 1, interflow simulated by VS2DT begins one day earlier and initially has a steeper curve than interflow simulated by MODFLOW-SZF; however, interflow from VS2DT approaches steady state slower (fig. 8). The total fluxes and storage agree well with the VS2DT model over the duration of the simulation with a low RMSE of 0.022 between models.

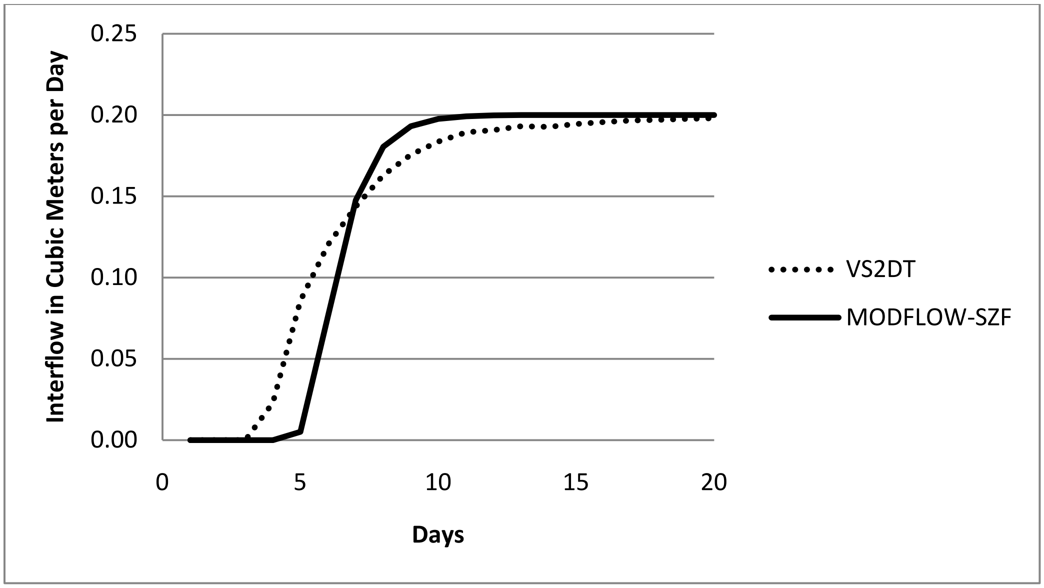

Figure 8: Comparison of interflow from simulation 1 using the best fit discretizations of MODFLOW-SZF and VS2DT with horizontal hydraulic conductivity of 250 m/day and vertical hydraulic conductivity of 0.50 m/day for the soil zone.

WETTING FRONT

Arrival of the wetting front at a depth of 1 m (i.e. the base of the soil zone) occurred at nearly the same time for the MODFLOW-SZF and VS2DT simulations (fig. 9). However, the wetting front simulated by VS2DT is more diffuse than that simulated by MODFLOW-SZF, owing to the effects of capillary pressure gradients at the wetting front. The UZF1 Package is used to route wetting fronts vertically in MODFLOW-SZF, and the UZF1 Package does not consider capillary pressure gradients. VS2DT simulations showed that the hydraulic conductivity averaging method used in VS2DT also affected the shape of the wetting front.

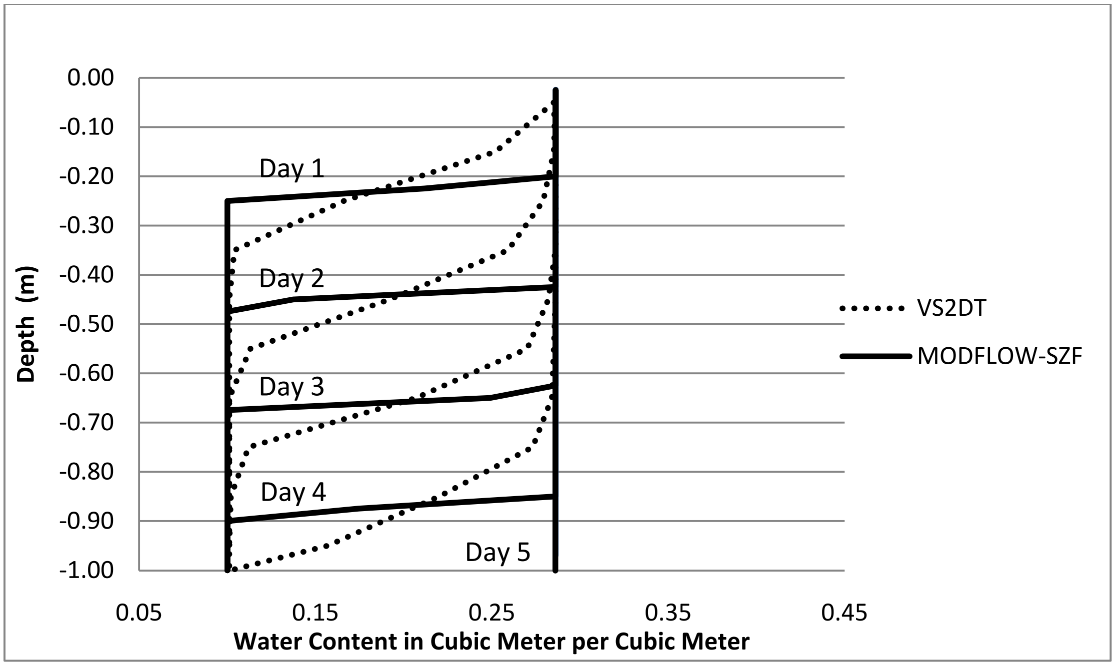

Figure 9: Comparison of vertical wetting fronts from simulation 1 between MODFLOW-SZF and VS2DT

WATER TABLE

Changes in the water table caused by precipitation and infiltration on the left side of the model domain (fig. 5) were compared between MODFLOW-SZF and VS2DT after 5 and 7 days and at steady state (fig. 10). The 0.25-m horizontal grid spacing (configuration 2) for MODFLOW-SZF was used because it provided a better horizontal resolution of the water table and reduced the need to interpolate water table elevations between largely-spaced model nodes. Comparison with MODFLOW-SZF configuration 2 is acceptable in this case because interflow results from MODFLOW-SZF were independent of the horizontal grid spacing (fig. 6).

The water table in VS2DT was calculated based upon pressure head at the lowest soil-zone model node and the difference between model node elevation and the base of the soil zone. The water table was assumed equal to the node elevation when the pressure head at the node was equal to zero. If the elevation of the zero pressure condition occured between nodes, then the location of the water table was interpolated between the nodes.

The water table was higher in MODFLOW-SZF because of the differences in how unsaturated storage is handled in each model. MODFLOW-SZF resolves the water table at a sub-grid scale using the vertical equilibrium assumption, which assumes that all water occupies the bottom of the cell, whereas in Richards' equation the volume of water contained in a cell is averaged over the volume of the cell. Capillarity also affects the water table height. In VS2DT, water is pulled from the saturated zone into the capillary fringe, whereas in MODFLOW water is not stored in a capillary fringe. The wetting front reaches the base of the soil zone at five days in both models and the steady state water table occurs at ten days in each model.

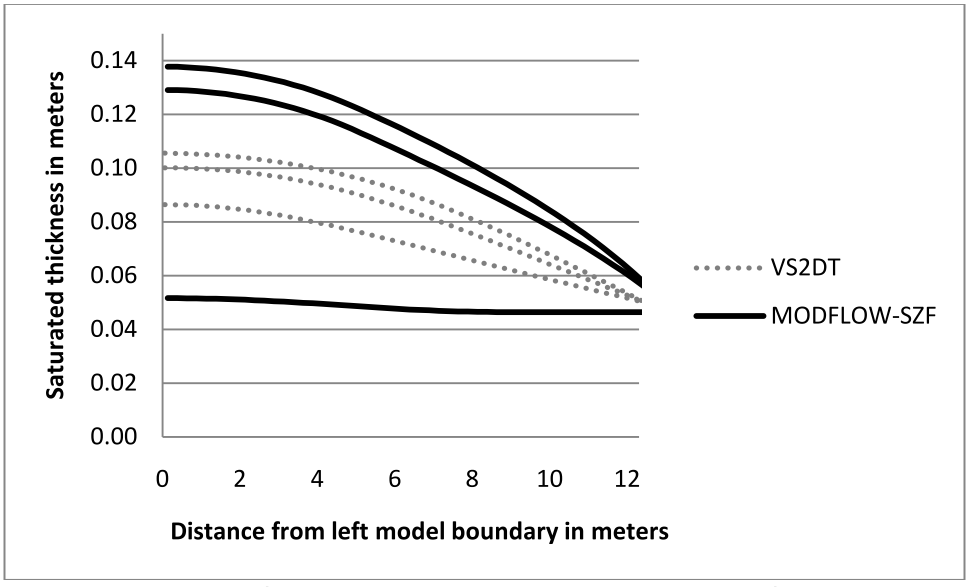

Figure 10: Comparison of rising water tables during recharge for 0.25 m spacing in MODFLOW-SZF and 0.1 m spacing in VS2DT and for simulation 1 at 5 days, 7 days, and steady state.

SENSITIVITY ANALYSES

HYDRAULIC CONDUCTIVITY

The effect of varying the hydraulic conductivity on interflow was evaluated by decreasing the horizontal hydraulic conductivity an order of magnitude from 250 m/d to 25 m/d while keeping the vertical to horizontal hydraulic conductivity ratio (0.002) the same (table 2). For this simulations with a lower hydraulic conductivity, VS2DT simulated the onset of interflow after 6 days as compared with 7 days for MODFLOW-SZF, although interflow does not increase as rapidly for VS2DT (fig. 11). Interflow curves between MODFLOW-SZF and VS2DT are similar with a RMSE of 0.014.

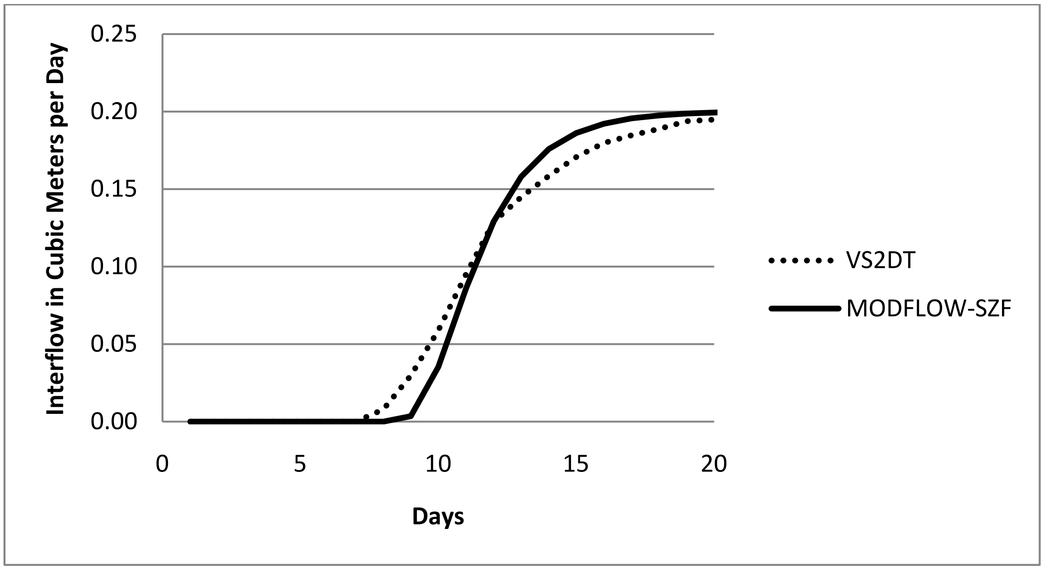

Figure 11: Comparison of interflow simulated by MODFLOW-SZF and VS2DT with horizontal hydraulic conductivity of 25 m/day and vertical hydraulic conductivity of 0.05 m/day

PRECIPITATION RATE

Figures 12 and 13 show model sensitivity to varying precipitation rates. The applied precipitation rate was 0.08 m/day in simulation 3, and the applied precipitation rate was 0.12 m/day in simulation 4 (table 2). Interflow occurred one day earlier in VS2DT compared with MODFLOW-SZF in simulations 3 and 4, and both VS2DT simulations reached steady state about 1 day after MODFLOW-SZF reached steady state. The shapes of the interflow curves are similar in simulations 3 and 4 with low RMSE of 0.045 and 0.052, respectively.

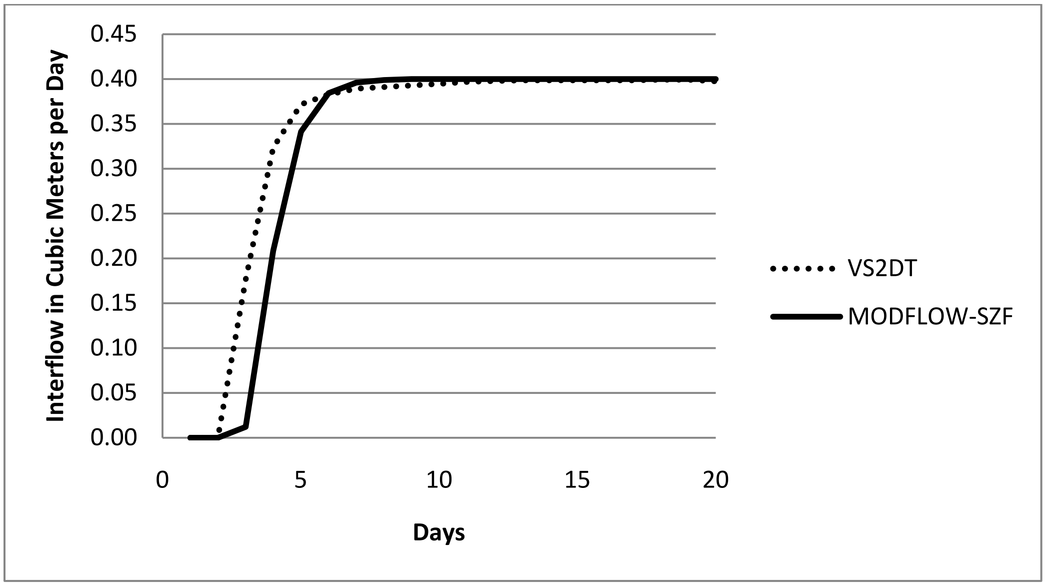

Figure 12: Comparison of interflow simulated by MODFLOW-SZF and VS2DT for an applied precipitation rate of 0.08 m/day

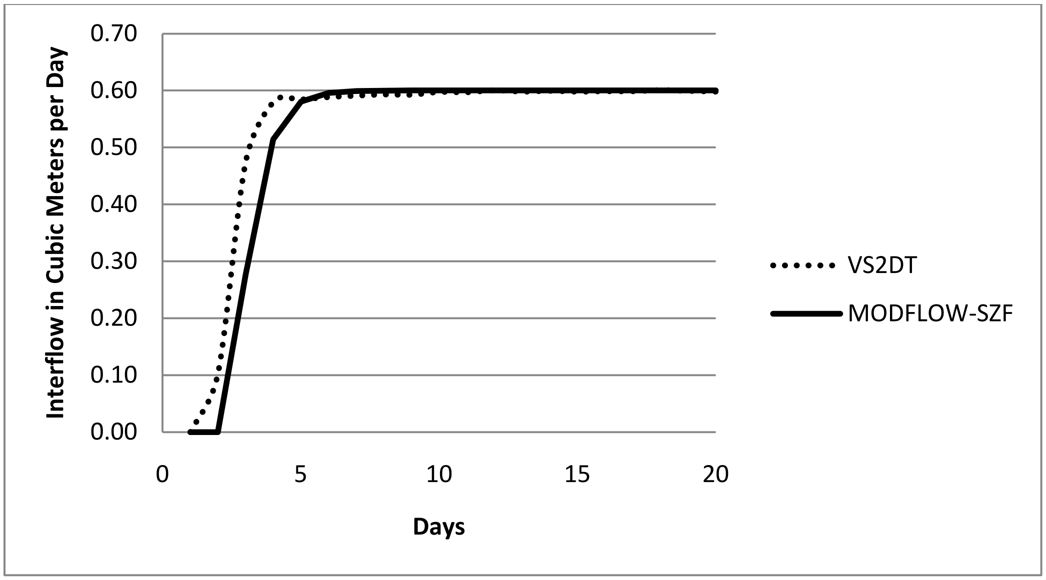

Figure 13: Comparison of interflow simulated by MODFLOW-SZF and VS2DT for an applied precipitation rate of 0.12 m/day

Table 6 summarizes the total interflow, percent differences, and RMSE values for the different simulations that were analyzed in this sensitivity analysis. The MODFLOW-SZF and VS2DT simulations had similar results for soil storage and interflow. The difference in total fluxes can be attributed to the timing delay of steady-state flux through the seepage face. The percent differences between MODFLOW-SZF and VS2DT were all less than 1.5% in all of the simulations with small RMSE ranging from 0.014-0.052.

Table 6: Summary of results of sensitivity analyses [storage and total interflow values in cubic meters]

	MODFLOW		VS2DT			
Simulation	**Storage**	**Total Interflow**	**Storage**	**Total Interflow (m^3)**	**Interflow % difference**	**RMSE**
1	1.20	2.80	1.22	2.78	0.72%	0.022
2	2.32	1.68	2.32	1.68	0.00%	0.014
3	1.46	6.53	1.38	6.62	-1.37%	0.045
4	1.63	10.37	1.57	10.43	-0.58%	0.052

IMPLICATIONS

The errors associated with using the groundwater flow equation to represent flow instead of Richards' equation were much lower (table 6) than the errors caused by grid effects (table 4), across a range of hydraulic conductivities and applied precipitation rates. In order to obtain a stable solution for Richards’ equation, the model resolution had to be 0.1 m or less. When you consider that regional models encompass hundreds to thousands of square kilometers, using a grid resolution of 0.1 m quickly becomes intractable, especially if results are desired in less than real time. In basin to regional scale applications, the spatial scale will be even greater than was used in this comparison, indicating that the groundwater flow equation is likely to be more accurate at representing interflow through the soil-zone when applied to scales greater than an experimental plot.

Recent research in this area of hydrology has been focused developing efficient numerical methods for solving Richards' equation in the context of integrated basin-scale models. These results suggest that more evaluation of simple governing equations is needed to determine how to best to represent variably saturated flow processes at the basin scale.

CHAPTER 4: SUMMARY AND CONCLUSIONS

Models can be improved by considering subsurface conditions in the soil zone because infiltration and runoff are functions of soil-zone saturation. The SZF Package uses Smith-Parlange infiltration equations, quasi 2-D kinematic wave overland flow routing, and the groundwater flow equation to simulate infiltration, runoff, and interflow because these equations are generally not as sensitive to grid-cell dimensions as Richards' equation.

- The Soil-Zone Flow Package for MODFLOW effectively simulates infiltration and interflow through the soil zone; verified through comparison to VS2DT including: water table development, timing of the wetting front from the land surface to the base of the soil zone, timing of interflow to the stream, soil-zone storage, total infiltration, and total interflow for a variety of applied precipitation rates and hydraulic conductivities. Timing of vertical and horizontal wetting fronts in both models was consistently less than 1 day of each other.
- The MODFLOW-SZF simulations achieved a reasonable infiltration and interflow solution with only 6 grid cells and 0.11 seconds of computation time, requiring a fraction of the cells and computation time required for VS2DT.
- The MODFLOW-SZF interflow solution was independent of discretization, whereas the VS2DT interflow solution was strongly sensitive to the discretization with percent differences between the 0.05-m and corresponding 0.1-m, 0.25-m

and 0.5-m discretizations equal to 0.02%, 0.30%, and 82.6% respectively and RMSE between 0.008 and 0.012.

- MODFLOW-SZF simulations correlated well with the VS2DT simulations, with simulation total percent differences of 1.5% or less and RMSE between 0.014 and 0.052 in all of the hydraulic conductivity and precipitation sensitivity simulations.
- The errors associated with discretization in Richards equation (up to 82.6% total interflow percent difference , RMSE up to 0.12) far outweigh the errors associated with using the groundwater flow equation to represent soil-zone flow (<1.5% total interflow percent difference, RMSE <0.052).

The SZF Package was developed for MODFLOW to incorporate soil-zone processes at spatial scales useful for basin-scale models. Results suggest that the Smith-Parlange analytical solution to Richards' equation and the groundwater flow equation can be used to accurately represent soil-zone processes at a variety of spatial scales, precipitation rates, and hydraulic conductivities independent of discretization.

The SZF Package effectively simulates infiltration and interflow at a coarser grid resolution than is usually required to obtain a stable solution to Richards' equation, indicating that it will be an effective framework for modeling soil zone processes at the regional scale.

REFERENCES CITED

Beven, K., (1981). Kinematic Subsurface Stormflow. Water Resources Research 17(5): 1419–1424

Beven, K., (1989). Changing Ideas in Hydrology - The Case of Physically-Based Models. Journal of Hydrology 105: 157-172.

Betson, R.P., (1964). What is Watershed Runoff?. Journal of Geophysical Research, Vol. 69 No. 8: 1541-1552

Brooks, R.H., and Corey, A.T., (1966). Properties of porous media affecting fluid flow. Journal of Irrigation and Drainage, v. 101, p. 85-92.

Buckingham E., (1907). Studies on the movement of soil moisture. Bureau of Soils Bulletin No. 38, USDA, p. 1–61.

Celia, M. A., Bouloutas, E. T., and Zarba, R. L., (1990), A general mass-conservative numerical solution for the unsaturated flow equation, Water Resources Research, 26(7), 1483–1496, doi:10.1029/WR026i007p01483.

Downer, C. W., and Ogden, F. L., (2004). Appropriate vertical discretization of Richards' equation for two-dimensional watershed-scale modeling. Hydrologic Processes, 18(1), 1–22.

Dunne, T., (1978). Field studies of hillslope flow processes. Hillslope Hydrology. M. J. Kirby. New York, John Wiley: 227-293.

Dunne, T., and Black, R.G., (1970). An experimental investigation of runoff production in permeable soils: Water Resources Research, v. 6, p. 478-490.

Green, W. A. and Ampt, G. A., (1911). Studies in soil physics, I, The flow of air and water through soils. Journal of Agricultural Science, 4: 1-24.

Harter, T. and Hopmans, J. W., (2004). Role of vadose-zone flow processes in regional scale hydrology-review, opportunities and challenges. Unsaturated Zone Modeling--Progress, Challenges and Applications. R. A. Feddes, G. H. De Rooij and J. C. Van Dam. Dordrecht, The Netherlands, Kluwer Academic Publisher, Wageningen Frontis Series: 179-208.

Healy, R.W., (1990). Simulation of solute transport in variably saturated porous media with supplemental information on modifications to the U.S. Geological Survey's Computer Program VS2D: U.S. Geological Survey Water-Resources Investigations Report 90-4025, 125 p.

Hewlett, J.D. and Hibbert A.R., (1967). Factors Affecting the Response of Small Watersheds to Precipitation in Humid Areas. Forest Hydrology 1967: 275-290

Horton, R. E.,(1933). The role of infiltration in the hydrological cycle. American Geophysical Union Transactions, 23: 479-482.

Jensen, K. H., and Mantoglou, A., (1992), Application of stochastic unsaturated flow theory, numerical simulations, and comparisons to field observations, Water Resources Research, 28(1), 269–284, doi:10.1029/91WR02233.

Kirby, M. J., (1988). Hillslope runoff processes and models. Journal of Hydrology, 100: 315-329.

Lighthill, M. J. and Whitham, G. B., (1955). On kinematic floods—flood movements in long rivers. Proceedings. R. Soc. London A220: 281-316.

Mantoglou, A., (1992), A theoretical approach for modeling unsaturated flow in spatially variable soils: Effective flow models in finite domains and nonstationarity, Water Resources Research, 28(1), 251–267, doi:10.1029/91WR02232.

Mantoglou, A., and Gelhar, L. W., (1987). Effective hydraulic conductivities of transient unsaturated flow in stratified soils. Water Resources Research, 23(1), 57-67, doi:10.1029/WR023i001p0005

Markstrom, S.L., Niswonger, R.G., Regan, R.S., Prudic, D.E., and Barlow, P.M., 2008, GSFLOW—Coupled ground-water and surface-water flow model based on the integration of the Precipitation-Runoff Modeling System (PRMS) and the Modular Ground-Water Flow Model (MODFLOW-2005): U.S. Geological Survey Techniques and Methods 6-D1, 240 p.

Mualem, Y., (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. Water Resources Research, 12, pp. 513–522.

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, Chapter A-1, 586 p.

Mein, R.G and Larson, C.L.(1973). Modeling Infiltration during a Steady Rain. Water Resources Research Vol. 9, No. 2: 384-394.

Niswonger, R.G., Panday, Sorab, and Ibaraki, Motomu, (2011). MODFLOW-NWT, A Newton formulation for MODFLOW-2005: U.S. Geological Survey Techniques and Methods 6–A37, 44 p.

Niswonger, R. G. and Prudic, D.E., (2005). Documentation of the Streamflow-Routing (SFR2) Package to include unsaturated flow beneath streams a modification to SFR1, U.S. Geological Techniques and Methods Book 6, Chapter A13: 62.

Niswonger, R.G., Prudic, D.E., and Regan, R.S., (2006a) Documentation of the Unsaturated-Zone Flow (UZF1) Package for modeling unsaturated flow between the land surface and the water table with MODFLOW-2005: U.S. Geological Survey Techniques and Methods 6-A19, 74 p.

Philip, J. R., (1957). The theory of infiltration: 2. The profile at infinity. Soil Science, 83: 435-448.

Preissmann, A. (1961). Propagation of translatory waves in channels and rivers. Proc., 1st Congress of French Association for Computation, AFCAL, Grenoble, France, 433–442.

Refsgaard, J.C., and Storm, B., and Singh, V.P.,(1995). MIKE-SHE. Computer models of Watershed Hydrology, vol. 1, 809-846

Richards, L.A., (1931). Capillary conduction of liquids through porous mediums. Physics, 1 (5): 318–333. doi:10.1063/1.1745010

Scheyer, J.M., and Hipple, K.W., (2005). Urban Soil Primer. United States Department of Agriculture, Natural Resources Conservation Service, National Soil Survey Center, Lincoln, Nebraska (http://soils.usda.gov/use).

Smith, R. E. and Parlange, J. Y., (1978). A parameter-efficient hydrologic infiltration model:. Water Resources Research 14(3): 533-538.

Smith, R. E., Corradini ,C., (1993). Modeling infiltration for multistorm runoff events. Water Resources Research 29: 133-134.

United States Department of Defense. Army Core of Engineers.(1973). Introduction and Application of Kinematic wave routing techniques Using HEC1. Davis: USACE Institute for Water Resources, Print.

Van Genuchten, M. T., (1980). A closed form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Science Society of America Journal 44: 892-898.

Vauclin, M., Khanji, D., and Vachaud, G., (1979). Experimental and numerical study of a transient, two-dimensional unsaturated-saturated water table recharge problem, Water Resources Research, 15(5), 1089–1101, doi:10.1029/WR015i005p01089.

Vogel, H.J., and Ippish, O., (2008). Estimation of a Critical Spatial Discretization Limit for Solving Richards' Equation at Large Scales. Vadose Zone Hydrology, Vol. 7, no. 1, 112-114

Whipkey, R. Z., (1965). Subsurface Stormflow from Forested Slopes. Bull., Int. Assoc. Sci. Hydrol., 10(2): 74-85

Woolhiser, D.A., Smith, R.E. and Goodrich, D.C., (1990). KINEROS, Kinematic Runoff and Erosion Model: Documentation and User Manual. USDA-ARS, No. 77.

APPENDIX A: SZF FORTRAN CODE

```fortran
C-------SUBROUTINE GWF2PSZ1AR
C
      SUBROUTINE GWF2PSZ1AR(IN,IUNITUZF,IUNITSFR,IGRID)
C
      USE GLOBAL,       ONLY: NCOL, NROW, NLAY, IOUT, ITRSS, ISSFLG,
     +                        DELR, DELC, IBOUND, BOTM, HNEW
      USE GWFPSZMODULE
      USE GLOBAL,       ONLY:ITMUNI,LENUNI
      IMPLICIT NONE
C     ------------------------------------------------------------------
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ARGUMENTS
C     ------------------------------------------------------------------
      INTEGER IN, IGRID, IUNITUZF, IUNITSFR
      CHARACTER*24 ANAME(12)
      DATA ANAME(1)/' PERCH  VERT. HYD. COND.'/
      DATA ANAME(2)/' SZ AIR ENTRY PRESSURE  '/
      DATA ANAME(3)/' SZ SPECIFIC RETENTION  '/
      DATA ANAME(4)/' SZ POROSITY            '/
      DATA ANAME(5)/' SZ B-K EPSILON        '/
      DATA ANAME(6)/' SZ SAT.VERT. HYD. COND.'/
      DATA ANAME(7)/' SZ HYD.COND. ALONG COL.'/
      DATA ANAME(8)/' SZ HYD.COND. ALONG ROWS'/
      DATA ANAME(9)/' SZ PARLANGE ALPHA      '/
      DATA ANAME(10)/' SZ INIT. WATER CONTENT '/
      DATA ANAME(11)/' CASCADE SURFACE       '/
      DATA ANAME(12)/' CHEZY"S COEFFICIENT   '/
C     ------------------------------------------------------------------
C     LOCAL VARIABLES
C     ------------------------------------------------------------------
      INTEGER IUZM, LLOC, ISTART, ISTOP, ICONFLAG, NUMCHECK
      INTEGER IC,IR,L
      CHARACTER*200 LINE
      REAL R
C     ------------------------------------------------------------------
      ALLOCATE (PERCHFLG, IPSZCB1, IPSZCB2, IFILLFLG, IUNITPSZ )
      IUNITPSZ=IN
C1-----IDENTIFY PACKAGE AND INITIALIZE.
C2-----PERCHFLG.GT. ZERO VKS FROM UZF/PERCHFLG.LE.0 VKS FROM PSZ
      WRITE (IOUT, 9001) IN
 9001 FORMAT (1X, /' PSZ1 -- PERCHED FLOW AND SOIL ZONE PACKAGE, ',
     +      'VERSION 1, 10/03/2005', /, 9X, 'INPUT READ FROM UNIT', I3)
      CALL URDCOM(IN, IOUT, LINE)
      LLOC = 1
      CALL URWORD(LINE, LLOC, ISTART, ISTOP, 2, PERCHFLG, R, IOUT, IN)
      IF ( PERCHFLG.GT.0 .AND. IUNITUZF.EQ.0 ) THEN
        WRITE (IOUT, 9002)
 9002   FORMAT (//' VERTICAL HYDRAULIC CONDUCTIVITY ',
     +          'OF THE PERCHING LAYER IS TO BE READ FROM UZF '
     +          //, ' AND UZF IS INACTIVE  --  SETTING PSZ PACKAGE TO
',
     +          'INACTIVE'///)
```

```
        IN = 0
        RETURN
      ELSE IF ( PERCHFLG.LE.0 ) THEN
        WRITE (IOUT, 9003)IN
 9003   FORMAT (//' READING  VERTICAL HYDRAULIC CONDUCTIVITY  ',
     +          //'OF THE PERCHING LAYER FROM UNIT ',I5///)
      ELSE
        WRITE (IOUT, 9004)
 9004   FORMAT (//' THE VERTICAL HYDRAULIC CONDUCTIVITY (K) ',
     +          //'OF THE PERCHING LAYER IS BEING SET ',
     +          'EQUAL TO THE VERTICAL K FROM UZF'///)
      END IF
      CALL URWORD(LINE, LLOC, ISTART, ISTOP, 2, IPSZCB1, R, IOUT, IN)
      CALL URWORD(LINE, LLOC, ISTART, ISTOP, 2, IPSZCB2, R, IOUT, IN)
      CALL URWORD(LINE, LLOC, ISTART, ISTOP, 2, IFILLFLG, R, IOUT, IN)
      IF ( IFILLFLG.GT.0 ) THEN
        WRITE (IOUT, 9665)
 9665   FORMAT (//' ADDITIONAL DEM FILL FOR SURFACE ROUTING CASCADES ',
     +    'WILL BE DONE '///)
      ELSE IF ( IFILLFLG.LE.0 ) THEN
        WRITE (IOUT, 9666)IN
 9666   FORMAT (//' ADDITIONAL DEM FILL FOR SURFACE ROUTING CASCADES ',
     +          ' WILL NOT BE DONE '///)
      END IF
      ALLOCATE (IPSZBOUND(NCOL,NROW))
      IPSZBOUND=0
      CALL ACTIVECELL
C
C3-----ALLOCATE SPACE FOR SOIL ZONE FLOW.
      IUZM = NROW*NCOL
      ALLOCATE (NUMACTIVE)
      ALLOCATE (DEM2(NCOL,NROW),DIFF(NCOL,NROW))
      ALLOCATE (HASZ(NCOL,NROW),THRSZ(NCOL,NROW))
      ALLOCATE (STRMCELL(NCOL,NROW))
      ALLOCATE (CHEZC(NCOL,NROW), SEEPOLD(NCOL, NROW))
      ALLOCATE (THSSZ(NCOL,NROW),EPSSZ(NCOL,NROW))
      ALLOCATE (FKZSZ(NCOL,NROW),HROFF(NCOL,NROW))
      ALLOCATE (ALPHASZ(NCOL,NROW),THTOSZ(NCOL,NROW))
      ALLOCATE (THTISZ(NCOL,NROW),RUN2STRM(NCOL,NROW))
      ALLOCATE (RSZ(NCOL,NROW), UZF_REJ(NCOL,NROW))
      ALLOCATE (RSZOLD(NCOL,NROW),R2SZ(NCOL,NROW))
      ALLOCATE (RITR(NCOL,NROW), TOT2CELL(NCOL,NROW))
      ALLOCATE (CUM_INF(NCOL,NROW),CUM_INF_P(NCOL,NROW))
      ALLOCATE (DYNCUM_H(NCOL,NROW),PRCHSTRM(NCOL,NROW))
      ALLOCATE (TIME_H(NCOL,NROW),PONDED(NCOL,NROW))
      ALLOCATE (PETSZ(NCOL,NROW),RUNOFFSZ(NCOL,NROW))
      ALLOCATE (TIME_INFILT(NCOL,NROW),TIME_POND(NCOL,NROW))
      ALLOCATE (IORDER(10,IUZM))
      ALLOCATE (IFLWSTRM(NCOL,NROW),IORDERCHK(IUZM))
      ALLOCATE (FLOWOTPSZ(NCOL,NROW))
      ALLOCATE (HA(IUZM,11),PIVOT(IUZM))
      ALLOCATE (CP(3),RP(4),RUNOLD(NCOL,NROW))
      ALLOCATE (RUNIN(NCOL,NROW,4),RUNOUT(NCOL,NROW,4),
     +          HITR(NCOL,NROW,NLAY))
      ALLOCATE (QSTORM(NCOL,NROW))
```

```
ALLOCATE (ETOUT(NCOL,NROW), PERCHDIS(NCOL,NROW))
ALLOCATE (SUMFLOW(NCOL,NROW), STRMCHD(NCOL,NROW))
ALLOCATE (RUNOLDT(NCOL,NROW))
STRMCELL=0
QSTORM=0.0
SEEPOLD=0.0
STRMCHD=0.0
HROFF=0.0
PETSZ = 0.0
TOT2CELL = 0.0
PONDED = 0.0
CUM_INF = 0.0
CUM_INF_P = 0.0
DYNCUM_H = 0.0
FLOWOTPSZ = 0.0
TIME_H = 0.0
RUNOFFSZ = 0.0
PRCHSTRM = 0.0
TIME_INFILT = 0.0
TIME_POND = 1.0E20
RSZOLD = 0.0
R2SZ = 0.0
RSZ = 0.0
RITR = 0.0
IFLWSTRM = 0
IORDERCHK = 0
IORDER = 0
THTOSZ = 0.0
THTISZ = 0.0
CP = 0.0
RP = 0.0
RUNOLD = 0.0
HA = 0
NUMACTIVE=0
RUNIN = 0.0
RUNOUT = 0.0
ETOUT = 0.0
PERCHDIS = 0.0
RUN2STRM = 0.0
SUMFLOW = 0.0
DEM2=0.0
DIFF=0.0
UZF_REJ=0.0
NUMCHECK=0
ICONFLAG = 0
HITR=HNEW
RUNOLDT = 0.0
CALL U2DREL(HASZ, ANAME(2), NROW, NCOL, 0, IN, IOUT)
CALL U2DREL(THRSZ, ANAME(3), NROW, NCOL, 0, IN, IOUT)
CALL U2DREL(THSSZ, ANAME(4), NROW, NCOL, 0, IN, IOUT)
CALL U2DREL(EPSSZ, ANAME(5), NROW, NCOL, 0, IN, IOUT)
CALL U2DREL(FKZSZ, ANAME(6), NROW, NCOL, 0, IN, IOUT)
CALL U2DREL(ALPHASZ, ANAME(9), NROW, NCOL, 0, IN, IOUT)
CALL U2DREL(THTISZ, ANAME(10), NROW, NCOL, 0, IN, IOUT)
CALL U2DREL(CHEZC, ANAME(12), NROW, NCOL, 0, IN, IOUT)
```

```
C4-----INTIALIZE THETA OLD AND DEM2 FOR FILL
      DO IR = 1, NROW
        DO IC = 1, NCOL
          DEM2(IC,IR)= BOTM(IC,IR,0)
          IF ( IPSZBOUND(IC,IR).GT.0 ) THEN
            THTOSZ(IC,IR) = THTISZ(IC,IR)+1.0E-16
            NUMCHECK = NUMCHECK + 1
          END IF
        END DO
      END DO
C-----SAVE POINTERS FOR GRID AND RETURN
      CALL SGWF2PSZ1PSV(IGRID)
C
C5----RETURN
      RETURN
      END SUBROUTINE GWF2PSZ1AR
C
C-------SUBROUTINE GWF2PSZ1RP
C
      SUBROUTINE GWF2PSZ1RP(IN, KPER, KSTP, IGRID, IUNITSFR)
      USE GLOBAL,       ONLY: NCOL, NROW, IOUT, IBOUND, DELR, DELC
      USE GWFPSZMODULE, ONLY: PETSZ, RSZ, NUMACTIVE, PERCHFLG,
IFLWSTRM,
     + HROFF
      USE GWFUZFMODULE, ONLY: VKS
      USE GWFSFRMODULE, ONLY: NSTRM, ISTRM
      IMPLICIT NONE
C     -----------------------------------------------------------------
C     SPECIFICATIONS:
C     -----------------------------------------------------------------
C     ARGUMENTS
C     -----------------------------------------------------------------
      INTEGER IN, KPER, IGRID, KSTP, IUNITSFR
C     -----------------------------------------------------------------
C     LOCAL VARIABLES
C     -----------------------------------------------------------------
      INTEGER NPSZ, IC, IR, L
      CHARACTER*24 ANAME(2)
      DATA ANAME(1)/' WATER APPLICATION RATE '/
      DATA ANAME(2)/'       POTENTIAL ET RATE'/
C     -----------------------------------------------------------------
C
C1------SET POINTERS FOR THE CURRENT GRID.
      CALL SGWF2PSZ1PNT(IGRID)
C
C----DETERMINE ORDER OF RUNOFF TO CELLS.
      IF (KPER.EQ.1) THEN
        CALL SOILZONE_ORDER()
C
C----DETERMINE WHAT SOIL ZONE CELLS FLOW TO A STREAM.
        IF ( IUNITSFR.GT.0 ) THEN
          DO L = 1 , NSTRM
            IR = ISTRM(2, L)
            IC = ISTRM(3, L)
            IFLWSTRM(IC, IR) = L
```

```
          END DO
        END IF
      END IF
C
C2-----READ IN ARRAY FOR WATER APPLICATION RATE.
      READ (IN, *) NPSZ
      IF ( NPSZ.LT.0 ) THEN
        WRITE (IOUT, *) 'USING WATER APPLICATION RATE FROM PREVIOUS'
        WRITE (IOUT, *) 'STRESS PERIOD.', 'CURRENT PERIOD IS: ', KPER
      ELSE
        CALL U2DREL(RSZ, ANAME(1), NROW, NCOL, 0, IN, IOUT)
      END IF
C
C2-----READ IN ARRAY FOR PET RATE.
      READ (IN, *) NPSZ
      IF ( NPSZ.LT.0 ) THEN
        WRITE (IOUT, *) 'USING POTENTIAL ET RATE FROM PREVIOUS STRESS '
        WRITE (IOUT, *) 'PERIOD.', 'CURRENT PERIOD IS: ', KPER
      ELSE
        CALL U2DREL(PETSZ, ANAME(2), NROW, NCOL, 0, IN, IOUT)
      END IF
C
C27-----RETURN.
      RETURN
      END SUBROUTINE GWF2PSZ1RP
C
C--------SUBROUTINE GWF2PSZ1FM
C
      SUBROUTINE GWF2PSZ1FM(KKPER, KKSTP, KKITER, IUNITSFR, IUNITLAK,
     +                      IUNITUZF, IGRID)
      USE GLOBAL,       ONLY: NCOL, NROW, NLAY, IOUT, DELR, DELC,CV,
     +                        IBOUND, BOTM, ISSFLG, BUFF, RHS, HNEW,
     +                        HCOF, HOLD
      USE GWFBASMODULE, ONLY: DELT, TOTIM, ICBCFL, TOTIM, PERTIM
      USE GWFPSZMODULE
      USE GWFUZFMODULE, ONLY: FINF, IUZFBND, SEEPOUT, REJ_INF
      USE GWFSFRMODULE, ONLY: NSS, NSTRM, ISTRM, SEG, STRM
      IMPLICIT NONE
C     -----------------------------------------------------------------
C     SPECIFICATIONS:
C     -----------------------------------------------------------------
C     FUNCTIONS
C     -----------------------------------------------------------------
      DOUBLE PRECISION FUNC3,FUNC9,FUNC1, FUNC2,FUNC4, FUNC5, FUNC7
      EXTERNAL FUNC1
      EXTERNAL FUNC2
      EXTERNAL FUNC3
      EXTERNAL FUNC4
      EXTERNAL FUNC5
      EXTERNAL FUNC7
      EXTERNAL FUNC9
C     -----------------------------------------------------------------
C     ARGUMENTS
C     -----------------------------------------------------------------
```

```
      INTEGER KKPER, KKSTP, IUNITSFR, IUNITLAK, IGRID, IUNITUZF,
KKITER,
     + ICONFLAG
C     -----------------------------------------------------------------
C     LOCAL VARIABLES
C     -----------------------------------------------------------------
      DOUBLE PRECISION DYNCUM_P, DYNCUM , BETA,
     +     ALPHA, FKS,FLOWIN, R, FKI, FKD,
     +     TIME, DELTH, SURF, TP, ROLD, DELT_TEMP,
     +     CLOSEZERO,CLENGTH,CWIDTH, DELH,
     +     R2,FKO,TIME_END,DYNCUM_SAVE,DIFFSEEP,
     +     CUM_INF_TEMP,CUM_INF_P_TEMP,
     +     TIME_TEMP,CP1,THOCHECK,DYNCUM_H_TEMP,ETACT,
     +     TIME_H_TEMP,DEPTHCHECK,FA,FB,FTEMP,
     +     TIME_TEMPA,TIME_TEMPB,CUM_INFA,CUM_INFB, CUM_INF_TEMP_OLD,
     +     TIME_TEMP1, F1, F2, TIME_TEMP2, FDERIV, TOL2, ERRORP, SQD
      DOUBLE PRECISION RUNERR, DLR, THI, AA, BB, BIGG, THS, THR, THO,
     +     THO_TM1, A, B
      INTEGER I, J, K, II, IC, IR, IK, IBD1, IBD2, IBDPSZ, IRUN,
     +        ISS, IFR, IJK, LL, IL, ILL, ICE, IRE
      CHARACTER*16 TEXTPRCH, TEXTET, TEXTINF, TEXTTHO, TEXTRUN
      DATA TEXTRUN/' SURF. RUNOFF  '/
      DATA TEXTINF/' INFILT. RATE  '/
      DATA TEXTTHO/' SURF. SAT.    '/
      DATA TEXTET/' SOILZONE ET  '/
      DATA TEXTPRCH/'PERCH THCKNESS '/
C     -----------------------------------------------------------------
C
C1------SET POINTERS AND INTITIALIZE VALUES FOR THE CURRENT GRID.
      CALL SGWF2PSZ1PNT(IGRID)
      CLOSEZERO = 1.0E-20
      DLR = 1.0E-5
      ISS = ISSFLG(KKPER)
      RUNERR = 0.0
      ICONFLAG = 0
      DO I=1, NSTRM
        STRM(31,I)=0.0
      END DO
      A = 0.0
      B = 0.0
      IF( KKITER.EQ.1 )RITR = 0.0
      LL = 1
      IFR = 0
      IJK = 1
      R = 0.0
      HROFF=0.0
      UZF_REJ=0.0
      FLOWOTPSZ=0.0
      QSTORM=0.0
C
C2-----INDIVIDUAL CELL INFILTRATION, TP, RUNOFF CALCULATION
C
      CONVERGE2: DO WHILE( LL.LE.100 )
        IF ( ICONFLAG.EQ.0) THEN
          DO IR = 1, NROW
```

```
            DO IC = 1, NCOL
            IL=IPSZBOUND(IC, IR)
       !NOTE: IF SZF ALLOWED IN MULTIPLE LAYERS THEN CHANGE
TO(IBOUND(IC,IR,IPSZBND(IC, IR))
       !HAVE TO ADJUST SO THAT PERCHED AQUIFERS CAN
       !BE SIMULATED IF THERE ARE ACTIVE CELLS BELOW AN INACTIVE CELL
              IF ( IBOUND(IC,IR,1).NE.0 ) THEN
                DIFFSEEP=0.0
                !INTIALIZE DIFFERENCE BETWEEN SEEPOUT ITERATIONS
                RP(1)=THRSZ(IC,IR)
                RP(2)=THSSZ(IC,IR)
                RP(3)=EPSSZ(IC,IR)
                RP(4)=FKZSZ(IC,IR)
                CP(1)=THRSZ(IC,IR)
                CP(2)=THSSZ(IC,IR)
                CP(3)=HASZ(IC,IR)
                THS=CP(2)
                THR=RP(1)
                BETA = 1.0
                ALPHA = ALPHASZ(IC,IR)
                FKS=RP(4)
                R = RITR(IC,IR) + RSZ(IC,IR)
                ROLD = RSZOLD(IC,IR)
                THI = THTISZ(IC,IR)
                THO = THTOSZ(IC,IR)+1.0E-7
                DELTH = THO - THI
                IF ( R.GE.FKS ) DELTH = THS - THI
                FKI = FUNC3(THI)
                FKO = FUNC3(THO)
                FKD = FKS - FKI
                TIME = TIME_INFILT(IC,IR)
                TIME_END = TIME + DELT
                TP = TIME_POND(IC,IR)
                DYNCUM = CUM_INF(IC,IR) - FKI*TIME
                DYNCUM_SAVE = DYNCUM
                CUM_INF_TEMP = CUM_INF(IC,IR)
                CUM_INF_P_TEMP = CUM_INF_P(IC,IR)
                DYNCUM_H_TEMP = DYNCUM_H(IC,IR)
                TIME_H_TEMP = TIME_H(IC,IR)
C
C3----CALCUATE MATRIC FLUX POTENTIAL
C
C4-----HIATUS IN STORM.
                IF ( (R-FKS).LT.0.0 ) THEN
                    B=THO
                    A=THI
                    BIGG = FUNC9(A,B) !NOTE: INTEGRATE B&C USING FUNC9
                    TP = 1.0E20
                    THOCHECK = FUNC7(FKS,R)
                    !NOTE: BC WATER CONTENT AT FKS
C
C5-----R<ROLD
C
                  IF ( R-ROLD.LT.CLOSEZERO .OR.
     1                 THOCHECK-THO.LT.-CLOSEZERO ) THEN
```

```
                    TIME_H(IC,IR) = TIME
                    DYNCUM_H(IC,IR) = DYNCUM
                    TIME_H_TEMP = TIME
                    DYNCUM_H_TEMP = DYNCUM
                    THO_TM1 = THO
                    THO = FUNC5(ALPHA,FKS,THI,THO_TM1,R,TIME,
     1                        TIME_H_TEMP,BETA,DYNCUM_H_TEMP,
     2                        IC,IR)
                  ELSE
                    THO = THOCHECK
                  END IF
                  CUM_INF_TEMP = CUM_INF_TEMP + R * DELT + FKI * DELT
                  R2 = R
                  ROLD = R
                  TIME = TIME + DELT
C6----DURING STORM EVENT.
                ELSE
C7----IF NOT SATURATED AT BEGINING OF TIMESTEP (ROLD<FKS)
                  IF( ROLD-FKS.LT.-CLOSEZERO ) THEN
                    CUM_INF_TEMP = 0.0
                    IF(THO-THS.LT.-CLOSEZERO)THI = THO
                    THO = THS
                    DELTH = THO - THI

                    FKI = FUNC3(THI)
                    FKO = FUNC3(THO)
                    FKD = FKS - FKI
                    TIME = 0.0
                    CUM_INF_TEMP = 0.0
                    TIME_END = DELT
                    TP = 1.0E20
                    DYNCUM = 0.0
                    DYNCUM_SAVE = 0.0
                  END IF
                  BIGG = FUNC9(THI,THO)
C8----CALCULATE PONDING TIME FOR LATEST R.
                  IF ( TIME.LT.TP.OR.TIME.LT.CLOSEZERO ) THEN
                    TP = FUNC4(FKS,ALPHA,R,BIGG,FKI,TIME,DELTH,
     +                       CUM_INF_TEMP,CLOSEZERO)
                    IF ( TIME-TP.LT.-CLOSEZERO ) THEN !NOTE: TIME<TP
C9----INCREMENT TIME BEFORE PONDING.
                      IF ( TIME_END.GE.TP ) THEN
                        CUM_INF_TEMP = CUM_INF_TEMP + R * TP +
     1                               FKI * TP
                        CUM_INF_P_TEMP = CUM_INF_TEMP
                        CUM_INF_P(IC,IR) = CUM_INF_P_TEMP
                        TIME = TP
                        DYNCUM = CUM_INF_TEMP - FKI * (TP-TIME)
                      ELSE                           !NOTE:TIME END<TP
                        CUM_INF_TEMP = CUM_INF_TEMP + R * DELT +
     1                                 FKI * DELT
                        DYNCUM = CUM_INF_TEMP - FKI * DELT
                        TIME = TIME + DELT
                      END IF
                    END IF
```

```
                  END IF
                  IF ( TIME-TIME_END.LT.-CLOSEZERO ) THEN
C
C10----INCREMENT TIME AFTER PONDING.
C
                    CUM_INF_TEMP = FUNC2(ALPHA,DELTH,BIGG,FKS,FKD,FKI,
     1                         CUM_INF_TEMP,CUM_INF_P_TEMP,TP,
     2                         TIME_END,DYNCUM,IC,IR,LL, THO, THI)

                    TIME = TIME_END
                  END IF
                  R2 = (DYNCUM-DYNCUM_SAVE)/DELT
                  HROFF(IC,IR)=(R-R2)*(DELR(IC)*DELC(IR))
                END IF
                R2SZ(IC,IR) = (R2*DELR(IC)*DELC(IR))
                FLOWOTPSZ(IC,IR)= HROFF(IC,IR)+ SEEPOUT(IC,IR)
                !NOTE: ADD BACK IN REJ_INF ONCE UZF SET UP FOR LAYERS
                IF ( ICONFLAG.EQ.1 .OR. LL.EQ.100 ) THEN
                  THTISZ(IC,IR) = THI
                  THTOSZ(IC,IR) = THO
                  TIME_INFILT(IC,IR) = TIME
                  TIME_POND(IC,IR) = TP
                  CUM_INF(IC,IR) = CUM_INF_TEMP
                END IF
              END IF
              SEEPOLD(IC,IR)=SEEPOUT(IC,IR)
            END DO
          END DO
C
C11-----END OF INDIVIDUAL CELL COMPUTATION
C
          CALL RUNOFF(IUNITSFR,IFR,IJK,RUNERR,ICONFLAG, KKPER, KKITER,
     +                LL, ICE, IRE)
        ELSE
          EXIT CONVERGE2
        END IF
        LL = LL + 1
      END DO CONVERGE2
C
C12-----END OF SOIL ZONE CONVERGENCE LOOP DISTRIBUTE SZ FLUXES
C
      CALL ADDFINF(IUNITUZF, KKPER)
      IF ( LL.GE.100) THEN
        WRITE (IOUT, 9002) RUNERR
        WRITE (IOUT,*) ICE,IRE
 9002   FORMAT (1X, 'CONVERGENCE FAILURE IN RUNOFF (ERROR = ',
     +        E20.10, ') -- SIMULATION ABORTING')
      END IF
      DO IC = 1, NCOL
        DO IR = 1, NROW
          IRUN = IFLWSTRM(IC, IR)
          IF ( IRUN.GT.0 .AND. IUNITSFR.GT.0 ) THEN
            STRM(31, IRUN) = STRM(31, IRUN)+ RUN2STRM(IC,IR)
          END IF
          IF ( ICONFLAG.EQ.1 ) THEN
```

```
            IF ( IBOUND(IC,IR,1).GT.0 ) THEN
              RSZOLD(IC,IR)  = RITR(IC,IR) + RSZ(IC,IR)
              RUNOLDT(IC,IR) = RUNOFFSZ(IC,IR)
            END IF
          END IF
        END DO
      END DO
C
      END SUBROUTINE GWF2PSZ1FM
C
C--------SUBROUTINE GWF2PSZ1BD
C
      SUBROUTINE GWF2PSZ1BD(KKPER, KKSTP, IUNITUZF, IUNITSFR, IGRID)

      USE GLOBAL,       ONLY: NCOL, NROW, NLAY, IOUT, IBOUND,CV, CC,
CR,
     +                        BUFF, DELR, DELC, BOTM, HNEW, RHS
      USE GWFBASMODULE, ONLY:MSUM,ICBCFL,DELT,PERTIM,TOTIM,
     1                      VBVL, VBNM
      USE GWFSFRMODULE, ONLY: NSS, NSTRM, ISTRM, SEG, STRM, HSTRM
      USE GWFUZFMODULE, ONLY: FINF, IUZFBND, VKS, REJ_INF
      USE GWFPSZMODULE, ONLY: IPSZCB1, IPSZCB2, R2SZ, IPSZBOUND,
     + FLOWOTPSZ, UZF_REJ, RUNOFFSZ, STRMCELL, STRMCHD, QSTORM,
     + RUN2STRM, IFLWSTRM, HROFF
      USE GWFNWTMODULE, ONLY: A, IA, ICELL
      USE GWFUPWMODULE, ONLY: HKUPW, SN
      IMPLICIT NONE
      CHARACTER*19 TEXT
      DOUBLE PRECISION RIN, ROUT, RATIN, RATOUT, CELBOT, DELH, QQ, HH,
     +               CHECK, TTOP, BBOT, THICK, CONDIF, DH, CHECK2,
     +                SSTRLEN, QCHECK,STAGELEV, DDDHORIZ
      INTEGER IBD1, IBD2, IL, IR, IC, IGRID, KKPER, KKSTP, IUNITUZF, I,
     +        IUNITSFR, IRUN,IJ, L
      REAL ZERO, CLOSEZERO,DUM7
      DATA TEXT /'INTERFLOW TO SFR'/
C     -----------------------------------------------------------------
C     ARGUMENTS
C     -----------------------------------------------------------------
      CALL SGWF2PSZ1PNT(IGRID)

C1------CLEAR RATIN AND RATOUT ACCUMULATORS, AND SET CELL-BY-CELL
C1------BUDGET FLAG.
      CLOSEZERO=1.0E-2
      ZERO=0.
      RATIN=ZERO
      RATOUT=ZERO
      IBD1=0
      IBD2=0
      QQ=0
      QSTORM=0.0
      QCHECK=0.0
      DUM7=0.0 !ALLOW MULT SZ LAY
      IF(IPSZCB1.GT.0 .AND. ICBCFL.NE.0) IBD1=1
      IF(IPSZCB2.GT.0 .AND. ICBCFL.NE.0) IBD2=1
C
```

```
C3------CLEAR THE BUFFER.
      DO 50 IL=1,NLAY
      DO 50 IR=1,NROW
      DO 50 IC=1,NCOL
      BUFF(IC,IR,IL)=ZERO
50    CONTINUE
C4------SUBTRACT LEAKAGE TO UZF FROM PERCHED WATER TABLE.
      !NOTE: NEED TO ADD RATOUT FOR SZ DISCHARGE TO LS
      DO IR=1, NROW
        DO IC=1, NCOL
          CHECK=0.0
          IF (KKPER.GT.3)THEN
          CHECK=0.0
          END IF
          IF (IL.GT.0) THEN
            IF (IL+1.LE.NLAY) THEN
              IF(IUZFBND(IC,IR).NE.0) THEN
              FINF(IC,IR)=R2SZ(IC,IR)/(DELC(IR)*DELR(IC))
                IF (FINF(IC,IR).GT.VKS(IC,IR)) THEN
                  UZF_REJ(IC,IR)=FINF(IC,IR)-VKS(IC,IR)
                  FINF(IC,IR)=VKS(IC,IR)
                  REJ_INF(IC,IR)=REJ_INF(IC,IR)+UZF_REJ(IC,IR)
                END IF
              END IF
            END IF
          END IF
          IF(STRMCELL(IC,IR).GT.0)THEN
            L=STRMCELL(IC,IR)
            STAGELEV=HSTRM(L,1)
C5-----CALCULATE QSTORM AND UPDATE RATOUT
           DO IL=1,1 !NOTE IF .GT. 1 ALLOW MULTIPLE SZ LAYER
           IF (STAGELEV.LT.HNEW(IC,IR,IL)) THEN
             IF (IBOUND(IC, IR, IL).GT.0)THEN
               IF (HNEW(IC,IR,IL).GT.BOTM(IC,IR,IL))THEN
                 IF (STAGELEV.GT.BOTM(IC,IR,IL)) THEN
                   HH= HNEW(IC,IR,IL)
                   TTOP=(BOTM(IC,IR,IL-1))
                   BBOT=STAGELEV
                   THICK=TTOP-BBOT
                   SSTRLEN=2.0*STRM(1,L)
                   CONDIF= (HKUPW(IC,IR,IL)*SSTRLEN)/
     +                        ((DELC(IR)+DELR(IC))/4)
                   DH=DDDHORIZ(HH,TTOP,BBOT,IL) !DSN/DH
                   IJ=ICELL(IC,IR,IL)
                   QCHECK=(CONDIF*SN(IJ)*THICK*(STAGELEV-HH))
                 ELSE
                   HH= HNEW(IC,IR,IL)
                   TTOP=(BOTM(IC,IR,IL-1))
                   BBOT=(BOTM(IC,IR,IL))
                   THICK=TTOP-BBOT
                   SSTRLEN=2*STRM(1,L)
                   CONDIF= (HKUPW(IC,IR,IL)*SSTRLEN)/
     +                       ((DELC(IR)+DELR(IC))/4)
                   DH=DDDHORIZ(HH,TTOP,BBOT,IL) !DSN/DH
                   IJ=ICELL(IC,IR,IL)
```

```
                   QCHECK=(CONDIF*SN(IJ)*THICK*(BBOT-HH))
                   DUM7=DUM7+QCHECK
                 END IF
               !NOTE: PREVENTS BACKFLOW INTO SZ; FLOW INTO SZ HANDLED
BY GFE AND CONDUCTANCE THROUGH MF FLOW PKG
               IF (QSTORM(IC,IR).LT.0.0) QSTORM(IC,IR)=0.0
               END IF
             END IF
           END IF
           END DO
           QSTORM(IC,IR)=QSTORM(IC,IR)+DUM7
           QQ=QSTORM(IC,IR)
           RATOUT=RATOUT-QQ
         END IF
        END DO
       END DO
C6-------SET VALUE FOR Q AND UPDATE BUFFER
      IF ( IBD1.GT.0) THEN
         DO IR=1, NROW
           DO IC=1, NCOL
            !NOTE: NEED TO RE-EVAL IF SZF NOT L1
            IF(IPSZBOUND(IC, IR).EQ.0)THEN
              IL=1
            ELSE
              IL=IPSZBOUND(IC,IR)
            END IF
            BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+ R2SZ(IC,IR)
          END DO
        END DO
        CALL UBUDSV(KKSTP, KKPER, TEXT, IPSZCB1, BUFF,
     +                                NCOL, NROW, NLAY, IOUT)
      END IF
      IF ( IBD2.GT.0) THEN
        DO IR=1, NROW
          DO IC=1, NCOL
            IF(IPSZBOUND(IC, IR).EQ.0)THEN
              IL=1
            ELSE
              IL=IPSZBOUND(IC,IR)
            END IF
            BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+ R2SZ(IC,IR)
          END DO
        END DO
        CALL UBDSV3(KKSTP, KKPER, TEXT, IPSZCB2,
     +                                   BUFF, IPSZBOUND, 1, NCOL, NROW,
     +                                   NLAY, IOUT, DELT, PERTIM, TOTIM,
     +                                   IBOUND)
      END IF
C7------MOVE RATES, VOLUMES & LABELS INTO ARRAYS FOR PRINTING.
!THIS SETS UP BUDGET PRINTING TEXT=BUDGET ITEM
      VBVL(4,MSUM)=RATOUT
      VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
      VBNM(MSUM)=TEXT
C
C8------INCREMENT BUDGET TERM COUNTER(MSUM).
```

```
      MSUM=MSUM+1
C
C9------WRITE STATEMENT TO PRINT OUT INTERFLOW AND SEEPAGE TO UZF
       WRITE (81,*) KKPER
       WRITE (81,*)
       DO IR=1,NROW
         DO IC=1,NCOL
           IF(QSTORM(IC,IR).GT.0.0) THEN
           WRITE(81,*)KKPER, QSTORM(3,1)
           END IF
         END DO
       END DO
       WRITE (82,*) KKPER
       WRITE (82,*)
       DO IR=1,NROW
         DO IC=1,NCOL
           IF(FINF(IC,IR).GT.0.0) THEN
           WRITE(82,999) FINF(IC,IR)
           END IF
         END DO
       END DO
  999  FORMAT(1000E10.5)
 1111  FORMAT(81F8.2)
      RETURN
      END SUBROUTINE GWF2PSZ1BD
C
C--------SUBROUTINE RUNOFF
C
      SUBROUTINE RUNOFF(IUNITSFR,IFR,IJK,FERROR,ICONFLAG,KKPER,KKITER,
     +  LL, ICE,IRE)!WH ADD KKITER,LL FOR DEBUG
      USE GLOBAL,       ONLY: NCOL, NROW, NLAY, IOUT, DELR, DELC,
     +                        IBOUND, BOTM
      USE GWFBASMODULE, ONLY: DELT, TOTIM, ICBCFL, TOTIM, PERTIM
      USE GWFPSZMODULE, ONLY: NUMACTIVE, IORDER, DEM2, FLOWOTPSZ,
     +                        CHEZC, RUNIN, RUNOUT, RITR, RUNOFFSZ,
     +                        IFLWSTRM, RUN2STRM, RUNOLD
      USE GWFSFRMODULE, ONLY: NSS, NSTRM, ISTRM, STRM
C     -----------------------------------------------------------------
      IMPLICIT NONE
C     -----------------------------------------------------------------
C     ARGUMENTS
C     -----------------------------------------------------------------
      DOUBLE PRECISION CLOSEZERO, FERROR, QA, QB, QC, QD, SLOPE, WIDTH,
     +                 C, DELX, TOL, FPART,DELZ, DELY, FTEST
      INTEGER IFR, NUMLEFT,NUMITR, II, IC, IR, TOTRUN, ICM, ICP, IRM,
     +        IRP, IFLIC, IFLIR,ICONFLAG, ICE, IRE, IUNITSFR, IJK,
     +        KKPER,KKITER, IRUN, RUNCON,LL,DUM
      REAL AREA, SLOPEMIN, ELEVLOW
C     -----------------------------------------------------------------
      IFR = 0
      NUMLEFT = NCOL*NROW
      NUMITR = 1
      CLOSEZERO = 1.0E-20
      TOL = 5.0E-3
      SLOPEMIN = 1.0E-6
```

```
      RUNCON=0
      FTEST=0.0
C1----LOOP THROUGH CALCULATION ORDER.
      FERROR=0.0
      RITR=0.0 !NOTE: RESET EVERY FORMULATE
      DO II = 1, NUMACTIVE
        IC = IORDER(1,II) !NOTE:THESE ARE SET IN SZ ORDER SBRT
        IR = IORDER(2,II)
        AREA = DELR(IC)*DELC(IR)
        TOTRUN = IORDER(7,II)+IORDER(8,II)+IORDER(9,II)+IORDER(10,II)
        ICM = 0
        ICP = 0
        IRM = 0
        IRP = 0
        ICE = IC
        IRE = IR
        IFLIC = IC
        IFLIR = IR
        ELEVLOW = DEM2(IC, IR)
        RUNOFFSZ(IC,IR)=0.0
        IF ( TOTRUN.GT.0 ) THEN
          FPART = FLOWOTPSZ(IC,IR)/TOTRUN
          IF ( FPART.GT.CLOSEZERO ) THEN
            IF( IC.GT.1 .AND. IORDER(7,II).GT.0 )THEN
              WIDTH = DELC(IR)
              DELX =  (DELR(IC)+DELR(IC-1))/2.0
              DELZ = (DEM2(IC,IR) - DEM2(IC-1,IR))
              SLOPE = DELZ/DELX
              IF ( SLOPE.LT.SLOPEMIN ) SLOPE = SLOPEMIN
              C = (CHEZC(IC,IR)+CHEZC(IC-1,IR))/2.0
              QA = RUNIN(IC-1,IR,1)
              QB = RUNOUT(IC-1,IR,1)
              QC = IORDER(7,II)*FPART
              CALL ROUTE_CHAN2(QA, QB, QC, QD, DELT, SLOPE, WIDTH, C,
     +                         DELX, IC, IR)
              RITR(IC-1,IR) = RITR(IC-1,IR) + QD/AREA
              RUNOFFSZ(IC-1,IR) = RUNOFFSZ(IC-1,IR) + QD
              IF ( ICONFLAG.EQ.1 ) RUNIN(IC-1,IR,1) =
IORDER(7,II)*FPART
              IF ( ICONFLAG.EQ.1 ) RUNOUT(IC-1,IR,1) = QD
            END IF
            IF( IC.LT.NCOL .AND. IORDER(8,II).GT.0 ) THEN
              WIDTH = DELC(IR)
              DELX = (DELR(IC)+DELR(IC+1))/2.0
              DELZ = (DEM2(IC,IR) - DEM2(IC+1,IR))
              SLOPE = DELZ/DELX
              IF ( SLOPE.LT.SLOPEMIN ) SLOPE = SLOPEMIN
              C = (CHEZC(IC,IR)+CHEZC(IC+1,IR))/2.0
              QA = RUNIN(IC+1,IR,2)
              QB = RUNOUT(IC+1,IR,2)
              QC = IORDER(8,II)*FPART
              CALL ROUTE_CHAN2(QA, QB, QC, QD, DELT, SLOPE, WIDTH, C,
     +                         DELX)
              RITR(IC+1,IR) = RITR(IC+1,IR) + QD/AREA
              RUNOFFSZ(IC+1,IR) = RUNOFFSZ(IC+1,IR) + QD
```

```
              IF ( ICONFLAG.EQ.1 ) RUNIN(IC+1,IR,2) =
IORDER(8,II)*FPART
              IF ( ICONFLAG.EQ.1 ) RUNOUT(IC+1,IR,2) = QD
            END IF
            IF( IR.GT.1 .AND. IORDER(9,II).GT.0 ) THEN
              WIDTH = DELR(IC)
              DELY = (DELC(IR)+DELC(IR-1))/2.0
              DELZ = (DEM2(IC,IR) - DEM2(IC,IR-1))
              SLOPE = DELZ/DELY
              IF ( SLOPE.LT.SLOPEMIN ) SLOPE = SLOPEMIN
              C = (CHEZC(IC,IR)+CHEZC(IC,IR-1))/2.0
              QA = RUNIN(IC,IR-1,3)
              QB = RUNOUT(IC,IR-1,3)
              QC = IORDER(9,II)*FPART
              CALL ROUTE_CHAN2(QA, QB, QC, QD, DELT, SLOPE, WIDTH, C,
     +                        DELY)
              RITR(IC,IR-1) = RITR(IC,IR-1) + QD/AREA
              RUNOFFSZ(IC,IR-1) = RUNOFFSZ(IC,IR-1) + QD
              IF ( ICONFLAG.EQ.1 ) RUNIN(IC,IR-1,3) =
IORDER(9,II)*FPART
              IF ( ICONFLAG.EQ.1 ) RUNOUT(IC,IR-1,3) = QD
            END IF
            IF( IR.LT.NROW .AND. IORDER(10,II).GT.0 ) THEN
              WIDTH = DELR(IC)
              DELY = (DELC(IR)+DELC(IR+1))/2.0
              DELZ = (DEM2(IC,IR) - DEM2(IC,IR+1))
              SLOPE = DELZ/DELY
              IF ( SLOPE.LT.SLOPEMIN ) SLOPE = SLOPEMIN
              C = (CHEZC(IC,IR)+CHEZC(IC,IR+1))/2.0
              QA = RUNIN(IC,IR+1,4)
              QB = RUNOUT(IC,IR+1,4)
              QC = IORDER(10,II)*FPART
              CALL ROUTE_CHAN2(QA, QB, QC, QD, DELT, SLOPE, WIDTH, C,
     +                        DELY)
              RITR(IC,IR+1) = RITR(IC,IR+1) + QD/AREA
              RUNOFFSZ(IC,IR+1) = RUNOFFSZ(IC,IR+1) + QD
              IF ( ICONFLAG.EQ.1 )
     +        RUNIN(IC,IR+1,4) = IORDER(10,II)*FPART
              IF ( ICONFLAG.EQ.1 ) RUNOUT(IC,IR+1,4) = QD
            END IF
          END IF
        ELSE
          RUNOFFSZ(IC,IR) = FLOWOTPSZ(IC,IR)
        END IF
      END DO
        DO IR = 1, NROW
          DO IC = 1, NCOL
          IF ( IBOUND(IC,IR,1).NE.0 ) THEN !NOTE:CHANGED TO NE
            IRUN = IFLWSTRM(IC, IR) !IRUN=STRMCELL OF DISCHARGE
            IF ( IRUN.GT.0 .AND. IUNITSFR.GT.0 ) THEN
              RUN2STRM(IC,IR) = RUNOFFSZ(IC,IR)
            END IF
            FTEST = ABS(RUNOFFSZ(IC,IR) -
RUNOLD(IC,IR))!NOTE:RUNOLD=LAST ITER RUNOFF
            IF ( FTEST.GT.FERROR+1.0E-7 ) THEN
```

```
              !NOTE:DETERMINE WHICH CELL IS MOST PROBLEMATIC
              FERROR = FTEST
              ICE = IC
              IRE = IR
            END IF
          END IF
          RUNOLD(IC,IR) = RUNOFFSZ(IC,IR)
        END DO
      END DO
      IF ( FERROR.LT.TOL ) ICONFLAG = 1
      IFR = 1
      RETURN
      END SUBROUTINE RUNOFF
C
C--------SUBROUTINE ROUTE_CHAN2
C
      SUBROUTINE ROUTE_CHAN2(QA, QB, QC, QD, DELTINC, SLOPE, WIDTH1, C,
     +                       DEL, IC, IR)
C***********************************************************************
*
C     IMPLICIT FINITE-DIFFERENCE SCHEME TO ROUTE OVERLAND FLOW
C     VERSION 2.6: DECEMBER 6, 2005
C***********************************************************************
*
      USE GLOBAL,       ONLY: IOUT
      USE GWFPSZMODULE  !PERHAPS CONSIDER MAKLING AN ONLY CLAUSE
C     ------------------------------------------------------------------
-
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
-
C     ARGUMENTS
C     ------------------------------------------------------------------
-
      IMPLICIT NONE
      DOUBLE PRECISION SLOPE, WIDTH1, QA, QB, QC, QD, C, DEL, WIDTH
      REAL DELTINC
      INTRINSIC ABS, DABS
C     ------------------------------------------------------------------
-
C     LOCAL VARIABLES
C     ------------------------------------------------------------------
-
      INTEGER I, MAXITER, IC, IR
      REAL W_1, WEIGHT
      DOUBLE PRECISION QDERIV, DQ, DELQ, TOL, AB, AC, AA, AD1,
CLOSEZERO
      DOUBLE PRECISION DEPTH, QD2, F11, F12, F1, F2, AD2, CONST
C     ------------------------------------------------------------------
      !NOTE: QA QB RUNIN RUNOUT PREV ITER
      MAXITER = 100
      WEIGHT = 0.9
      CLOSEZERO = 1.0D-20
      W_1 = 1.0 - WEIGHT
      DQ = 1.0E-12
```

```
      TOL = 1.0E-6
      I = 1
      WIDTH = 0.1*WIDTH1
C
C2------MAKE AN INITIAL GUESS AT QD.
      QD = (QC+QB)/2.0
C
C3------INITIALIZE CONSTANTS.
      CONST = C*WIDTH*SLOPE**0.5
      DELQ = 0.0D0
      IF ( CONST.GT.CLOSEZERO ) THEN
        DEPTH = (QA/CONST)**(2.0/3.0)
        AA = DEPTH*WIDTH
        DEPTH = (QB/CONST)**(2.0/3.0)
        AB = DEPTH*WIDTH
        DEPTH = (QC/CONST)**(2.0/3.0)
        AC = DEPTH*WIDTH
      ELSE
        AA = WIDTH
        AB = WIDTH
        AC = WIDTH
      END IF
C
C4------CALCULATE FLOW IN CHANNELS--MAXIMUM ITERATIONS IS 50.
      CONVERGE: DO WHILE( I.LT.MAXITER )
C
C5------LOOP THROUGH UNTIL DELQ LESS THAN TOLERANCE.
        IF ( DABS(DELQ).GT.TOL .OR. I.EQ.1 ) THEN
          QD2 = QD + DQ
C
C6------CALCULATE VARIABLES AD1 AND AD2.
          IF ( CONST.GT.CLOSEZERO ) THEN
            DEPTH = (QD/CONST)**(2.0/3.0)
            AD1 = DEPTH*WIDTH
            DEPTH = (QD2/CONST)**(2.0/3.0)
            AD2 = DEPTH*WIDTH
          ELSE
            AD1 = WIDTH
            AD2 = WIDTH
          END IF
C
C7------CALCULATE FLOW.
          F11 = (WEIGHT*((QD)-QC)+(W_1)*(QB-QA))/DEL
          F12 = (AD1-AB)/(DELTINC)
          F1 = F11+F12
          F11 = (WEIGHT*((QD+DQ)-QC)+(W_1)*(QB-QA))/DEL
          F12 = (AD2-AB)/(DELTINC)
          F2 = F11+F12
          QDERIV = (F2-F1)/DQ
          DELQ = -F1/QDERIV
          IF ( (QD + DELQ).LT.0.0 ) THEN
            IF ( QD-TOL.GT.CLOSEZERO ) THEN
              DELQ = -QD
              QD = 0.0D0
            ELSE
```

```
              QD = 0.0D0
              DELQ = 2.0*TOL
            END IF
          ELSE
            QD = QD + DELQ
          END IF
C
C8------EXIT LOOP IF DELQ LESS THAN TOLERANCE.
        ELSE
          EXIT CONVERGE
        END IF
          I = I + 1
      END DO CONVERGE
 C
      IF ( I.GE.MAXITER ) WRITE(IOUT,*)
     +               'NON-CONVERGENCE IN OVERLAND FLOW ROUTING',I,QD,
     +                   DELQ
C
C9------RETURN.
      RETURN
      END SUBROUTINE ROUTE_CHAN2

      DOUBLE PRECISION FUNCTION FUNC1(BIGG,DELTH,ALPHA,FKS,DYNCUM)
      IMPLICIT NONE
C1------CALCULATE FC
      DOUBLE PRECISION F1, DYNCUM, ALPHA, FKS, DELTH, BIGG
      INTRINSIC EXP
      F1=DYNCUM*ALPHA/(DELTH*BIGG)
      FUNC1=FKS*(1.0+ALPHA/(EXP(F1)-1.0))
      END FUNCTION FUNC1
C
      DOUBLE PRECISION FUNCTION FUNC2(ALPHA,DELTH,BIGG,FKS,FKD,FKI,
     +               CUM_INF_TEMP,CUM_INF_P,TP,
     +               TIME_TEMP,DYNCUM,IC, IR,LL, THO, THI)
      USE GLOBAL,       ONLY: IOUT
      USE GWFBASMODULE, ONLY: DELT
      IMPLICIT NONE
      DOUBLE PRECISION BIGG, F1,F2, CHECK, DCUM,CLOSEZERO,TOL,
     + CUMINC, F3, DELTH, FKS, FKD, F4,ALPHA, GAMMA,DYNCUM_P,
     + CUM_INF_P,FKI, TP, CUM_INFA, CUM_INF_TEMP, CUM_INFB,
     + TIME_TEMP, DYNCUM, D, CTOP,C, FNUM, FA, FB,F11, THO, THI
      INTEGER IC, IR, LL, I
      INTRINSIC LOG, ABS
C
C1-----CALCULATE CUMULATIVE INFILTRATION AFTER PONDING
      CHECK=0.0 !WH ADD
      DCUM = 1.0E-8
      CLOSEZERO = 1.0E-20
      TOL = 1.0E-5
      I = 1
      CUMINC = 0.0
      F3=(DELTH*BIGG*FKS)/FKD  !NOTE: EQN 22 SMITH CORDINI
      F4=-FKS*(1.0-ALPHA)
      GAMMA=ALPHA*FKS/FKD  !SP GAMMA SHAPE FACTOR
      DYNCUM_P=CUM_INF_P-FKI*TP
```

```
      !NOTE: BOUNDING SOLN BETWEEEN A&B FOR BISECTION METHOD
      CUM_INFA = 1.01*DYNCUM_P + FKI*TIME_TEMP
      CUM_INFB = CUM_INF_TEMP + 5.0*FKS*DELT + FKI*DELT
      CUM_INF_TEMP = (CUM_INFA + CUM_INFB)/2.0
C2-----CALCULATE FA AND FB
      !FA CALCULATION
      IF(CUM_INFA-(FKI*TIME_TEMP).GT.CLOSEZERO)THEN
        DYNCUM=CUM_INFA - FKI*TIME_TEMP
      ELSE
        DYNCUM=0.0
        CUM_INF_TEMP = (CUM_INFA + CUM_INFB)/2.0
      END IF
      F1=ALPHA*DYNCUM/(DELTH*BIGG)
      F2=ALPHA*DYNCUM_P/(DELTH*BIGG)
      D=CUM_INFA-CUM_INF_P !NOTE: I-IP
      CTOP = CLOSEZERO
      IF ( F1.LT.70. .AND. F1.GT.CLOSEZERO )
      !NOTE: ANYTHING HIGHER THAN 70 TOO BIG
     +      CTOP = (EXP(F1)-1.0+GAMMA)
      C=(CTOP)/(EXP(F2)-1.0+GAMMA)
      IF ( C.LT.CLOSEZERO )C = CLOSEZERO
      FNUM=D-F3*LOG(C)
      FA=F4*((TIME_TEMP)-TP)+(FNUM) !NOTE: FA CAN BE NEG
      !FB CALCULATION
      IF((CUM_INFB - FKI*TIME_TEMP).GT.CLOSEZERO)THEN
      DYNCUM=CUM_INFB - FKI*TIME_TEMP
      ELSE
      DYNCUM= 0.0
      CUM_INFB=FKI*TIME_TEMP
      END IF
      F1=ALPHA*DYNCUM/(DELTH*BIGG)
      F2=ALPHA*DYNCUM_P/(DELTH*BIGG)
      D=CUM_INFB-CUM_INF_P
      CTOP = 2.51544E30
      IF ( F1.LT.70. .AND. F1.GT.CLOSEZERO )
     +      CTOP = (EXP(F1)-1.0+GAMMA)
      C=(CTOP)/(EXP(F2)-1.0+GAMMA)
      IF ( C.LT.CLOSEZERO )C = CLOSEZERO
      FNUM=D-F3*LOG(C) !NOTE: EQN 22 SMITH CORRADINI
      FB=F4*((TIME_TEMP)-TP)+(FNUM)
C3-----INITIAL F11 CALCULATION-CONVERGENCE CRITERIA
      IF((CUM_INF_TEMP - FKI*TIME_TEMP).GT.CLOSEZERO)THEN
        DYNCUM=CUM_INF_TEMP - FKI*TIME_TEMP
      ELSE
        DYNCUM=0.0
        CUM_INF_TEMP=FKI*TIME_TEMP
      END IF
      F1=ALPHA*DYNCUM/(DELTH*BIGG)
      F2=ALPHA*DYNCUM_P/(DELTH*BIGG)
      D=CUM_INF_TEMP-CUM_INF_P
      CTOP = 2.51544E30
      IF ( F1.LT.70. .AND. F1.GT.CLOSEZERO ) !NOTE: EXP(70) TOO BIG
     +  CTOP = (EXP(F1)-1.0+GAMMA)
        C=(CTOP)/(EXP(F2)-1.0+GAMMA)
      IF ( C.LT.CLOSEZERO )C = CLOSEZERO
```

```
      FNUM=D-F3*LOG(C)
      F11=F4*((TIME_TEMP)-TP)+(FNUM) !NOTE: FUNC EVAL WITH INTIAL GUESS
C4-----CHECK RESIDUAL AND ITERATE BISECTION AS NEEDED.
      CONVERGE: DO WHILE( I.LT.50 )
        IF ( ABS(F11).GT.TOL ) THEN !NOTE: CONVERGENCE CHECK
          IF ( F11*FB.GT.0.0 ) THEN
            CUM_INFB = CUM_INF_TEMP
            FB=F11
          ELSE
            CUM_INFA = CUM_INF_TEMP
            FA=F11
          END IF
          !NOTE: BISECT VALUE
          CUM_INF_TEMP=(CUM_INFB+CUM_INFA)/2.0
          !NOTE: RECALCULATE F11 FOR BISECTED VALUE.
          IF((CUM_INF_TEMP - FKI*TIME_TEMP).GT.CLOSEZERO)THEN
            DYNCUM=CUM_INF_TEMP - FKI*TIME_TEMP
          ELSE
            DYNCUM=0.0
            CUM_INF_TEMP=FKI*TIME_TEMP
          END IF
          F1=ALPHA*DYNCUM/(DELTH*BIGG)
          F2=ALPHA*DYNCUM_P/(DELTH*BIGG)
          D=CUM_INF_TEMP-CUM_INF_P
          CTOP = 2.51544E30
          IF ( F1.LT.70. .AND. F1.GT.CLOSEZERO )
          !NOTE:SETS CTOP IF.LT.70
     +    CTOP = (EXP(F1)-1.0+GAMMA)
          C=(CTOP)/(EXP(F2)-1.0+GAMMA)
          IF ( C.LT.CLOSEZERO )C = CLOSEZERO
          FNUM=D-F3*LOG(C)
          F11=F4*((TIME_TEMP)-TP)+(FNUM) !NOTE: FUNCTION EVALUATION
        ELSE                             !NOTE: IF F11.LE.TOL
          DYNCUM=CUM_INF_TEMP - FKI*(TIME_TEMP)
          EXIT CONVERGE
        END IF
        I = I + 1
      END DO CONVERGE
      IF ( I.GE.100 ) THEN
        WRITE(IOUT,*)'NON-CONVERGENCE FOR TIME AFTER PONDING  ',
     1                'ERROR = ',LL,IC,IR,F11
      END IF
      FUNC2 = CUM_INF_TEMP
      END FUNCTION FUNC2
C
      DOUBLE PRECISION FUNCTION FUNC3(DUM1)
      USE GWFPSZMODULE, ONLY: RP
      IMPLICIT NONE
      DOUBLE PRECISION:: DUM1
C1-----USE BROOK-COREY TO CALCULATE UNSATURATED K.
      IF(DUM1.LT.RP(2))THEN
        FUNC3=RP(4)*(((DUM1-RP(1))/(RP(2)-RP(1)))**RP(3))
      ELSE
        FUNC3=RP(4)
      END IF
```

```
      END FUNCTION FUNC3

      DOUBLE PRECISION FUNCTION FUNC4(FKS,ALPHA,R,BIGG,FKI,TIME,DELTH,
     1               CUM_INF_TEMP,CLOSEZERO)
      IMPLICIT NONE
      DOUBLE PRECISION BIGG, R, FKS, ALPHA, FKI, TIME, DELTH,
     + CUM_INF_TEMP, CLOSEZERO, F1, F2, F3
      INTRINSIC LOG, ABS
C1-----CALCULATE TP FOR GIVEN R
      IF ( ABS(R-FKS).GT.CLOSEZERO ) THEN
        F1=R*TIME-CUM_INF_TEMP
        F2=BIGG*DELTH/ALPHA
        F3=LOG((R-FKS+ALPHA*FKS)/(R-FKS))
        FUNC4=(F2*F3+F1)/(R-FKI)
      ELSE
        FUNC4 = 1.0E20
      END IF
      END FUNCTION FUNC4
C
      DOUBLE PRECISION FUNCTION FUNC5(ALPHA,FKS,THI,THO_TM1,R,TIME,
     1               TIME_H,BETA,DYNCUM_H,IC,IR)
      USE GLOBAL,       ONLY: IOUT
      USE GWFPSZMODULE, ONLY: CP
      USE GWFBASMODULE, ONLY: DELT
      IMPLICIT NONE
      EXTERNAL FUNC3, FUNC9
      DOUBLE PRECISION A, B, BIGG, FUNC3, THO, THI, THO_TM1,THS,
     +                 K1,THOK3,THOK4,THOK5,THOK6, CLOSEZERO,
     +                 ALPHA,FKS,TIME,TIME_H,H,TE,FCT,
     +                 K2,K3,K4,YIP1,X,EPS,HNEW,K6,K5, FUNC9,
     +                 HMIN,HMAX,BETA, DELTHO, FKI, DYNCUM_Z, DYNCUM_H,
     +                 R, FKO, FUNC, F1, F2, F11
      REAL P
      INTEGER L,I,MAXITER,KK, IC, IR
C1-----CALCULATE TIME AFTER PONDING
      THS=CP(2)
      P = 2.0
      EPS = 1.0E-6
      I = 1
      MAXITER = 100
      BETA = 0.90 !SMITH PARLANGE BETA
      DELTHO = 0.0
      CLOSEZERO = 1.0E-16
      THO = THO_TM1
      IF ( ABS(THO-THI).LT.CLOSEZERO )THO = THI-5.0*EPS
      FKI = FUNC3(THI)
      X = DELT/3.0
      FCT = 0.9
      H = DELT
      HMAX = DELT
      HMIN = DELT/1000.
C2-----BEGIN RUNGE-KUTTA
      !K1
      CONVERGE: DO WHILE ( X.LT.DELT )
        IF ( I.GE.MAXITER ) THEN
```

```
          EXIT CONVERGE
        ELSE
          I = I + 1
          IF ( DELT-X.LT.H )H = DELT-X
          DYNCUM_Z = DYNCUM_H + (R-FKI)*(TIME-TIME_H)
          IF ( DYNCUM_Z.LT.EPS ) DYNCUM_Z = EPS
          DELTHO = THO-THI
          B=THO
          FKO=FUNC3(B)
          BIGG = FUNC9(THI,B)
          IF (BIGG.LT.CLOSEZERO)BIGG = CLOSEZERO
          F1 = DELTHO/DYNCUM_Z
          F2 = BETA*P*FKS*DELTHO*BIGG/DYNCUM_Z
          F11 = F1*(R-FKI-FKO-F2)
          K1 = H*F11
          !K2
          DYNCUM_Z = DYNCUM_H + (R-FKI)*(TIME+H/4.0-TIME_H)
          DELTHO = THO+K1/4.0-THI
          B=THO+K1/4.0
          FKO=FUNC3(B)
          BIGG = FUNC9(THI,B)
          IF (BIGG.LT.CLOSEZERO)BIGG = CLOSEZERO
          F1 = DELTHO/DYNCUM_Z
          F2 = BETA*P*FKS*DELTHO*BIGG/DYNCUM_Z
          F11 = F1*(R-FKI-FKO-F2)
          K2 = H*F11
          !K3
          THOK3=THO+(3.0/32.0)*K1+(9.0/32.0)*K2
          DYNCUM_Z = DYNCUM_H + (R-FKI)*(TIME+(3.0/8.0)*H-TIME_H)
          DELTHO = THOK3-THI
          B=THOK3
          FKO=FUNC3(B)
          BIGG = FUNC9(THI,B)
          IF (BIGG.LT.CLOSEZERO)BIGG = CLOSEZERO
          F1 = DELTHO/DYNCUM_Z
          F2 = BETA*P*FKS*DELTHO*BIGG/DYNCUM_Z
          F11 = F1*(R-FKI-FKO-F2)
          K3 = H*F11
          !K4
          THOK4 = THO+(1932./2197.)*K1-
(7200./2197.)*K2+(7296./2197.)*K3
          DYNCUM_Z = DYNCUM_H + (R-FKI)*(TIME+(12./13.)*H-TIME_H)
          DELTHO = THOK4-THI
          B=THOK4
          FKO=FUNC3(B)
          BIGG = FUNC9(THI,B)
          IF (BIGG.LT.CLOSEZERO)BIGG = CLOSEZERO
          F1 = DELTHO/DYNCUM_Z
          F2 = BETA*P*FKS*DELTHO*BIGG/DYNCUM_Z
          F11 = F1*(R-FKI-FKO-F2)
          K4 = H*F11
          !K5
          THOK5 = THO+(439./216.)*K1-8.0*K2+(3680./513.)*K3-
     +            (845./4104.)*K4
          DYNCUM_Z = DYNCUM_H + (R-FKI)*(TIME+H-TIME_H)
```

```
          DELTHO = THOK5-THI
          B=THOK5
          FKO=FUNC3(B)
          BIGG = FUNC9(THI,B)
          IF (BIGG.LT.CLOSEZERO)BIGG = CLOSEZERO
          F1 = DELTHO/DYNCUM_Z
          F2 = BETA*P*FKS*DELTHO*BIGG/DYNCUM_Z
          F11 = F1*(R-FKI-FKO-F2)
          K5 = H*F11
          !K6
          THOK6 = THO-(8./27.)*K1+2.0*K2-(3544./2565.)*K3+
     +            (1859./4104.)*K4-(11./40.)*K5
          DYNCUM_Z = DYNCUM_H + (R-FKI)*(TIME+0.5*H-TIME_H)
          DELTHO = THOK6-THI
          B=THOK6
          FKO=FUNC3(B)
          BIGG = FUNC9(THI,B)
          IF (BIGG.LT.CLOSEZERO)BIGG = CLOSEZERO
          F1 = DELTHO/DYNCUM_Z
          F2 = BETA*P*FKS*DELTHO*BIGG/DYNCUM_Z
          F11 = F1*(R-FKI-FKO-F2)
          K6 = H*F11
          TE = ABS((1./360.)*K1-(128./4275.)*K3-(2197./75240.)*K4
     +               +(1./50.)*K5+(2./55.)*K6)
          HNEW = FCT*H*(H*EPS/TE)**0.25
          IF ( TE.LE.H*EPS )THEN
            THO = THO_TM1 +(16./135.)*K1+(6656./12825.)*K3+
     +                (28561./56430.)*K4-(9.0/50.0)*K5+(2.0/55.0)*K6
            THO_TM1 = THO
            H = HNEW
            IF ( H.LT.HMIN ) H = HMIN
            X = X + H
          END IF
          IF ( TE.GT.H*EPS )THEN
            H = HNEW
            IF ( H.GT.HMAX ) H = HMAX
          END IF
        END IF
      END DO CONVERGE
      IF (I.GE.200) WRITE(IOUT,*) "FAILED IN FUNC5", IC, IR,X,DELT
      DYNCUM_Z = DYNCUM_H + (R-FKI)*(TIME-TIME_H)
      FUNC5 = THO
      END FUNCTION FUNC5
C
      DOUBLE PRECISION FUNCTION FUNC7(FKS,R)
      USE GWFPSZMODULE, ONLY: RP
      IMPLICIT NONE
      DOUBLE PRECISION R, FKS
C1-----CALCULATE THETA WHEN R>ROLD AND R<FKS.
      FUNC7 = (RP(2)-RP(1))*(R/FKS)**(1.0/RP(3)) + RP(1)
      END FUNCTION FUNC7
C
      DOUBLE PRECISION FUNCTION FUNC9(A,B)
      USE GWFPSZMODULE, ONLY: RP
      IMPLICIT NONE
```

```
      DOUBLE PRECISION:: A, B, TERM1, TERM2, TERM_DIFF
C1-----RETURNS INTEGRAL OF BROOKS COREY RELATIVE PERMEABILITY
       EQUATION(SMITH PARLANGE G)
        TERM1=((((B-RP(1))*((RP(1)-B)/(RP(1)-
RP(2)))**(RP(3)))/(1+RP(3))))
        TERM2=((((A-RP(1))*((RP(1)-A)/(RP(1)-
RP(2)))**(RP(3)))/(1+RP(3))))
        TERM_DIFF=TERM1-TERM2
        FUNC9=RP(4)*TERM_DIFF
      END FUNCTION FUNC9
C
C--------SUBROUTINE SOILZONE_ORDER
C
      SUBROUTINE SOILZONE_ORDER
      USE GWFPSZMODULE, ONLY: IORDER, IORDERCHK, NUMACTIVE, DEM2, DIFF,
     +                        STRMCELL, IFILLFLG
      USE GLOBAL,       ONLY: NCOL, NROW, NLAY, IBOUND, BOTM, IOUT
      IMPLICIT NONE
C     -----------------------------------------------------------------
C     LOCAL VARIABLES
C     -----------------------------------------------------------------
      INTEGER IALLCELL, IDD, NUMCELL, IL, I, K, IGRID
      INTEGER ICOUNT,ICM,ICP,IRM,IRP,ITRACK, IDD2, III, NTOTRCH, IC, IR
      INTEGER IDONE,IFLAG,IFLAG2,IFILL,IFILL2,OUTIT,OUTITMAX, INIT
      INTEGER INITMAX
      INTEGER II, J, HRUFLG, NUMACT,IDUM1,IDUM2
      REAL CLOSEZERO, PCT, TOL, DPIT !INCREMENT TO RAISE ELEVATION
C     -----------------------------------------------------------------
C1------ORDER ALL STORMFLOW CELLS BASED ON GRAVITY FLOW PATHS (TOP OF
        HILLS DOWN)
       !CASCADE_FILL SUBROUTINE
       !NOTE: ****CONSIDER MOVING THESE VARIABLES TO INPUT FILE***
       INITMAX=100000000
       IDONE=0
       DPIT=0.1
       TOL=0.5*DPIT
       OUTIT=0 !NUMBER OF OUTER ITERATIONS
       OUTITMAX=100000 !MAX NUMBER OF OUTER ITERATIONS
       INIT=0
       IF (IFILLFLG.EQ. 1) THEN
         DO WHILE (IDONE.EQ.0)
      !NOTE: IDONE=1 THE DEM COMPLETELY REPROCESSED AND NO SWALES EXIST
         IFLAG2=0
      !NOTE: NUMIT=0 ARRAY FOR STORING NUM OF ITERATIONS/CELL ***MOVE
      TO  MODULE IF DECIDE TO KEEP***
         DO IR=1,NROW
           DO IC=1,NCOL
           IF (IBOUND(IC,IR,1).EQ.1) THEN
             IFLAG=1
            !NOTE: IFLAG=0 A LOWER CELL EXISTS AROUND CURRENT CELL
C
C2-----CHECK TO SEE IF CASCADE OPERATION WILL BE OUT OF BOUNDS
             IF (((IR+1).GT.NROW) .OR. ((IC+1).GT. NCOL)) THEN
               WRITE(IOUT,*)"CHECK MODEL DIMENSIONS-",
     +                      "EXITING CASCADE FILL"
```

```
               EXIT
             ELSE IF (((IR-1).LT.NROW) .OR. ((IC-1).LT. NCOL))THEN
               WRITE(IOUT,*)"CHECK MODEL DIMENSIONS-",
     +                      "EXITING CASCADE FILL"
               EXIT
             END IF
C
C3-----CASCADE CELL BY CELL FILL PROCEDURE
             CHECKSWALE:DO WHILE(IFLAG.EQ.1)
               IF(STRMCELL(IC,IR).GT.0) EXIT CHECKSWALE
               !INSIDE CELLS
               IF ( ((IC.GT.1) .AND. (IC.LT.NCOL)) .AND. ((IR.GT.1)
     +           .AND. (IR.LT.NROW)) ) THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR-1)).GT.TOL).AND.
     +              (IBOUND(IC,(IR-1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSEIF ( (DEM2(IC,IR)-DEM2(IC,(IR+1)).GT.TOL).AND.
     +                  (IBOUND(IC,(IR+1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSEIF ( (DEM2(IC,IR)-DEM2((IC-1),IR).GT.TOL).AND.
     +                  (IBOUND((IC-1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 ELSE IF ( (DEM2(IC,IR)-DEM2((IC+1),IR).GT.TOL).AND.
     +                   (IBOUND((IC+1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 END IF
               ENDIF
               !TOP LEFT CORNER
               IF ( ((IR.EQ.1) .AND. (IC.EQ.1)) ) THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR+1)).GT.TOL).AND.
     +              (IBOUND(IC,(IR+1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSE IF ( (DEM2(IC,IR)-DEM2((IC+1),IR).GT.TOL).AND.
     +                   (IBOUND((IC+1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 END IF
               ENDIF
               !UPPER EDGE
               IF ( ((IC.NE.1) .AND. (IC.NE.NCOL)) .AND. (IR.EQ.1))
THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR+1)).GT.TOL).AND.
     +                  (IBOUND(IC,(IR+1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSEIF ( (DEM2(IC,IR)-DEM2((IC-1),IR).GT.TOL).AND.
     +                  (IBOUND((IC-1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 ELSE IF ( (DEM2(IC,IR)-DEM2((IC+1),IR).GT.TOL).AND.
     +                   (IBOUND((IC+1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 END IF
               ENDIF
               !TOP RIGHT CORNER
               IF ((IC.EQ.NCOL) .AND. (IR.EQ.1)) THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR+1)).GT.TOL).AND.
     +              (IBOUND(IC,(IR+1),1).NE. 0) ) THEN
```

```
                   IFLAG=0
                 ELSEIF ( (DEM2(IC,IR)-DEM2((IC-1),IR).GT.TOL).AND.
     +                      (IBOUND((IC-1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 END IF
               ENDIF
               !LEFT HAND SIDE
               IF ((IC.EQ.1) .AND. ((IR.NE.1) .AND. (IR.NE.NROW))) THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR-1)).GT.TOL).AND.
     +              (IBOUND(IC,(IR-1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSEIF ( (DEM2(IC,IR)-DEM2(IC,(IR+1)).GT.TOL).AND.
     +                  (IBOUND(IC,(IR+1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSE IF ( (DEM2(IC,IR)-DEM2((IC+1),IR).GT.TOL).AND.
     +                   (IBOUND((IC+1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 END IF
               ENDIF
               !LOWER LEFT CORNER
               IF ((IC.EQ.1) .AND. (IR.EQ.NROW)) THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR-1)).GT.TOL).AND.
     +              (IBOUND(IC,(IR-1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSE IF ( (DEM2(IC,IR)-DEM2((IC+1),IR).GT.TOL).AND.
     +                   (IBOUND((IC+1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 END IF
               ENDIF
               !LOWER EDGE
               IF (((IC.NE.1) .AND. (IC.NE.NCOL)).AND.(IR.EQ.NROW))
THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR-1)).GT.TOL).AND.
     +              (IBOUND(IC,(IR-1),1).NE. 0) ) THEN
                   IFLAG=0
                ELSEIF ( (DEM2(IC,IR)-DEM2((IC-1),IR).GT.TOL).AND.
     +                       (IBOUND((IC-1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 ELSE IF ( (DEM2(IC,IR)-DEM2((IC+1),IR).GT.TOL).AND.
     +                   (IBOUND((IC+1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 END IF
               ENDIF
              !LOWER RIGHT CORNER
               IF ((IC.EQ.NCOL) .AND. (IR.EQ.NROW)) THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR-1)).GT.TOL).AND.
     +                  (IBOUND(IC,(IR-1),1).NE. 0) ) THEN
                 IFLAG=0
                 ELSEIF ( (DEM2(IC,IR)-DEM2((IC-1),IR).GT.TOL).AND.
     +                  (IBOUND((IC-1),IR,1).NE. 0) ) THEN
                 IFLAG=0
                 END IF
               ENDIF
               !RIGHT HAND SIDE
```

```
               IF ((IC.EQ.NCOL).AND.((IR.NE.1) .AND. (IR.NE.NROW)))
THEN
                 IF ( (DEM2(IC,IR)-DEM2(IC,(IR-1)).GT.TOL).AND.
     +               (IBOUND(IC,(IR-1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSEIF ( (DEM2(IC,IR)-DEM2(IC,(IR+1)).GT.TOL).AND.
     +                  (IBOUND(IC,(IR+1),1).NE. 0) ) THEN
                   IFLAG=0
                 ELSEIF ( (DEM2(IC,IR)-DEM2((IC-1),IR).GT.TOL).AND.
     +                      (IBOUND((IC-1),IR,1).NE. 0) ) THEN
                   IFLAG=0
                 END IF
               ENDIF
               IF (IFLAG.EQ.1) THEN
                 DEM2(IC,IR)=DEM2(IC,IR)+ DPIT
                 IFLAG2=1 !NOTE: INDICATES THAT A CELL WAS UPDATED
                 !NOTE: IDONE DOES NOT GET SET TO 1 UNTIL IFLAG2=0
                 !NUMIT(IC,IR)=NUMIT(IC,IR)+1
                 !NOTE: STORES NUMBER OF INCREMENTS FOR EACH CELL
               ENDIF
               INIT=INIT+1
               IF (INIT.GT.INITMAX) THEN
                  PRINT*, "THERE WERE GREATER THAN", INITMAX,
     +                   "INNER ITERATIONS...EXITING CASCADE_FILL"
                  EXIT CHECKSWALE
                ENDIF
             END DO CHECKSWALE
            INIT=0
            ENDIF
            END DO
          END DO
          OUTIT=OUTIT+1
          IF (OUTIT.GT.OUTITMAX) THEN
            PRINT*, "THERE WERE GREATER THAN", OUTITMAX,
     +      "OUTER ITERATIONS. RESETTING DEM TO ORIGINAL SURFACE.
     +       EXITING CASCADE FILL."
            DO IR=1, NROW
              DO IC=1, NCOL
              DEM2(IC,IR)=BOTM(IC,IR,0)
              END DO
            END DO
            EXIT
          ENDIF
          IF (IFLAG2.EQ.0) THEN
            IDONE=1
          ENDIF
        END DO
      ENDIF
C
C4-----WRITE OUT FILLED SURFACE TO OUTPUT FILE
      WRITE (IOUT, *)"FILLED DEM FOR SURFACE CASCADES"
      DO IR=1,NROW
        DO IC=1,NCOL
          WRITE (IOUT,'(50F7.2)') DEM2(IC,IR)
        END DO
```

```
      END DO
C
C5-----BEGIN CASCADE EVALUATION
      IALLCELL=0
      IDD=1
      NUMCELL=NROW*NCOL
      CLOSEZERO = 1.0E-7
      I=1
      IL = 1
      DO IR = 1,NROW
        DO IC = 1,NCOL
          IF( IBOUND(IC, IR, IL).LE.0 )THEN
            IORDERCHK(I)=-999
          END IF
          I=I+1
        END DO
      END DO
      II = 1
      IDD = 1
      ORDERCELL: DO WHILE(IALLCELL.EQ.0)
          IF ( II.LE.NUMCELL ) THEN
          IDD2 = 1
          DO IR = 1,NROW
            DO IC = 1,NCOL
              ICOUNT=0
              IF(IORDERCHK(IDD2).EQ.0)THEN
                DO III = 3,10
                  IORDER(III,IDD) = 0
                END DO
                ICOUNT=0
                ICM=IC-1
                ICP=IC+1
                IRM=IR-1
                IRP=IR+1
C
C6------DETERMINE THE NUMBER OF SURROUNDING CELLS THAT HAVE A
C-------LOWER ELEVATION.
                IF( IC.GT.1 ) THEN
                  IF ( IBOUND(ICM, IR, 1).GT.0 )THEN
                    IF(DEM2(IC, IR).GE.DEM2(ICM, IR))THEN
                      ICOUNT=ICOUNT+1
                      IF(DEM2(IC, IR)-DEM2(ICM, IR).GT.CLOSEZERO)
     +                  IORDER(7,IDD)=1
                    ELSE IF(IORDERCHK(IDD2-1).GT.0)THEN
                      ICOUNT=ICOUNT+1
                    END IF
                    IF(DEM2(IC, IR).LT.DEM2(ICM, IR))IORDER(3,IDD)=1
                  ELSE
                    ICOUNT=ICOUNT+1
                    IORDER(3,IDD)=0
                    IORDER(7,IDD)=0
                  END IF
                ELSE
                  ICOUNT=ICOUNT+1
                  IORDER(3,IDD)=0
```

```
  IORDER(7,IDD)=0
END IF
IF( IC.LT.NCOL ) THEN
  IF ( IBOUND(ICP, IR, 1).GT.0 )THEN
    IF(DEM2(IC, IR).GE.DEM2(ICP, IR))THEN
      ICOUNT=ICOUNT+1
      IF(DEM2(IC, IR)-DEM2(ICP, IR).GT.CLOSEZERO)
        IORDER(8,IDD)=1
    ELSEIF(IORDERCHK(IDD2+1).GT.0)THEN
      ICOUNT=ICOUNT+1
    END IF
    IF(DEM2(IC, IR).LT.DEM2(ICP, IR))IORDER(4,IDD)=1
  ELSE
    ICOUNT=ICOUNT+1
    IORDER(4,IDD)=0
    IORDER(8,IDD)=0
  END IF
ELSE
  ICOUNT=ICOUNT+1
  IORDER(4,IDD)=0
  IORDER(8,IDD)=0
END IF
IF( IR.GT.1 ) THEN
  IF ( IBOUND(IC, IRM, 1).GT.0 )THEN
    IF(DEM2(IC, IR).GE.DEM2(IC, IRM))THEN
      ICOUNT=ICOUNT+1
      IF(DEM2(IC, IR)-DEM2(IC, IRM).GT.CLOSEZERO)
        IORDER(9,IDD)=1
    ELSEIF(IORDERCHK(IDD2-NCOL).GT.0)THEN
      ICOUNT=ICOUNT+1
    END IF
    IF(DEM2(IC, IR).LT.DEM2(IC, IRM))IORDER(5,IDD)=1
  ELSE
    ICOUNT=ICOUNT+1
    IORDER(5,IDD)=0
    IORDER(9,IDD)=0
  END IF
ELSE
  ICOUNT=ICOUNT+1
  IORDER(5,IDD)=0
  IORDER(9,IDD)=0
END IF
IF( IR.LT.NROW ) THEN
  IF ( IBOUND(IC, IRP, 1).GT.0 )THEN
    IF(DEM2(IC, IR).GE.DEM2(IC, IRP))THEN
      ICOUNT=ICOUNT+1
      IF(DEM2(IC, IR)-DEM2(IC, IRP).GT.CLOSEZERO)
        IORDER(10,IDD)=1
    ELSEIF(IORDERCHK(IDD2+NCOL).GT.0)THEN
      ICOUNT=ICOUNT+1
    END IF
    IF(DEM2(IC, IR).LT.DEM2(IC, IRP))IORDER(6,IDD)=1
  ELSE
  ICOUNT=ICOUNT+1
  IORDER(6,IDD)=0
```

```
                  IORDER(10,IDD)=0
                  END IF
                ELSE
                  ICOUNT=ICOUNT+1
                  IORDER(6,IDD)=0
                  IORDER(10,IDD)=0
                END IF
C
C7------IF OUTFLOW TO ALL SURROUNDING CELLS HAS BEEN DETERMINED
C-------OR IF SURROUNDING CELLS HAVE LOWER ELEVATION.
                IF(ICOUNT.EQ.4)THEN
                  IORDER(1,IDD)=IC
                  IORDER(2,IDD)=IR
                  IORDERCHK(IDD2) = 1
                  IDD=IDD+1
                END IF
              END IF
              IDD2=IDD2+1
            END DO
          END DO
          ITRACK=0
          I = 1
          FILTER: DO WHILE ( ITRACK.EQ.0 )
            IF ( I.LE.NUMCELL ) THEN
              IF(IORDER(1,I).EQ.0)THEN
                ITRACK=1
              END IF
            ELSE
              EXIT FILTER
            END IF
            I = I + 1
          END DO FILTER
          IF(ITRACK.EQ.0)IALLCELL=1
        ELSE
          EXIT ORDERCELL
        END IF
        II = II + 1
      END DO ORDERCELL
      NUMACTIVE = IDD - 1
      END SUBROUTINE SOILZONE_ORDER
C
      SUBROUTINE ACTIVECELL

      USE GLOBAL, ONLY: IBOUND, NLAY, NCOL, NROW
      USE GWFPSZMODULE, ONLY: IPSZBOUND
      IMPLICIT NONE
C     -----------------------------------------------------------------
C     LOCAL VARIABLES
C     -----------------------------------------------------------------
      INTEGER::IC,IR, ILL
C     -----------------------------------------------------------------
C1------NEW SUBROUTINE HERE TO FIND ACTIVE CELL AND ADD TO IPSZBOUND
ARRAY
      ILL = 1
      DO IR= 1, NROW
```

```
        DO IC= 1, NCOL
          ILL=1
          ACTIVECHECK:DO WHILE ( ILL.LE.NLAY )
            IF ( IBOUND(IC, IR, ILL).NE.0 ) THEN
              EXIT
            END IF
            ILL = ILL + 1
          END DO ACTIVECHECK

        IF(ILL.LE.NLAY) IPSZBOUND(IC,IR) = ILL
        END DO
      END DO
      END SUBROUTINE ACTIVECELL
C
      SUBROUTINE ADDFINF(IUNITUZF,KKPER)
      USE GLOBAL, ONLY: IBOUND, NLAY, NCOL, NROW, HNEW, BOTM, CV,RHS,
     +                  HCOF, DELR, DELC, IOUT
      USE GWFPSZMODULE, ONLY: IPSZBOUND,R2SZ, UZF_REJ
      USE GWFUZFMODULE, ONLY: FINF, IUZFBND, VKS, REJ_INF
      IMPLICIT NONE
C     LOCAL VARIABLES
C     -----------------------------------------------------------------
      INTEGER::IC, IR, IL, IUNITUZF, KKPER
      DOUBLE PRECISION:: CELBOT, DELH
C     -----------------------------------------------------------------
C1-----SUBROUTINE TO CALCULATE WATER BALANCE AND ADD TO FINF ARRAY IN
UZF
      DO IR= 1, NROW
        DO IC= 1, NCOL
          IL=IPSZBOUND(IC, IR)
          IF (IL.GT.0) THEN
            IF (IL+1.LE.NLAY) THEN
              IF(IUZFBND(IC,IR).NE.0) THEN
!NOTE: THIS MAY PRESENT PROBLEMS WHEN CONSTANT HEAD IUZFBND SET TO ZERO
!WHICH MEANS ET, AND GROUNDWATER CONNECTION CANNOT BE ESABLISHED IF
!GW HEAD AT BASE OF SZ FINF NOT ADDED TO UZF
                FINF(IC,IR)=R2SZ(IC,IR)/(DELC(IR)*DELR(IC))
                IF (FINF(IC,IR).GT.VKS(IC,IR)) THEN
                  UZF_REJ(IC,IR)=(FINF(IC,IR)-VKS(IC,IR))
                  FINF(IC,IR)=VKS(IC,IR)
                  REJ_INF(IC,IR)=REJ_INF(IC,IR)+UZF_REJ(IC,IR)
                END IF
              END IF
            END IF
          END IF
        END DO
       END DO
      END SUBROUTINE ADDFINF
C
      FUNCTION DDDHORIZ(HUP, TTOP, BBOT, IL)
      USE GWFNWTMODULE
      USE GWFUPWMODULE, ONLY: LAYTYPUPW
      IMPLICIT NONE
C     -----------------------------------------------------------------
-
```

```
C     SPECIFICATIONS:
C     ----------------------------------------------------------------
C     ARGUMENTS
C     ----------------------------------------------------------------
      DOUBLE PRECISION HUP, TTOP, BBOT
      DOUBLE PRECISION DDDHORIZ
C     ----------------------------------------------------------------
C     LOCAL VARIABLES
C     ----------------------------------------------------------------
      DOUBLE PRECISION FACTOR, X, S, V, COF1, COF2, EPS, ACOF, Y
      DOUBLE PRECISION EPSQD, Z
      INTEGER IL
C     ----------------------------------------------------------------
C1----RETURNS DERIVATIVE OF HORIZONTAL CONDUCTANCE BASED ON SMOOTH
      FUNCTION
C     FUNCTION IS CALCULATED IN UPW PACKAGE IN SUBROUTINE SAT_THICK
      DDDHORIZ = 0.0D0
      IF ( LAYTYPUPW(IL).LE.0 ) RETURN
C2----STRAIGHT LINE WITH PARABOLIC SMOOTHING
      EPS = THICKFACT
      ACOF = 1.0 / (1.0 - EPS)
      X = (HUP-BBOT)/(TTOP-BBOT)
      IF ( X.LT.1.0D-9 )  X = 1.0D-9
      IF ( X.LT.0.0 ) THEN
        Y = 0.0
      ELSEIF(X.LT.EPS)THEN
        Y = ACOF * X / (EPS*(TTOP-BBOT))
      ELSEIF(X.LT.1.0D0-EPS)THEN
        Y = ACOF /(TTOP-BBOT)
      ELSEIF(X.LT.1.0D0)THEN
        X = 1.0 - X
        Y = - ACOF * X / (EPS * (TTOP - BBOT))
        Y = 1.0-Y
      ELSE
        Y = 0.0
      ENDIF
      FACTOR = Y
      DDDHORIZ = FACTOR
      END FUNCTION DDDHORIZ
C     ----------------------------------------------------------------
```

APPENDIX B: SZF CODE IMPLEMENTED INTO STREAM FLOW ROUTING PACKAGE.

```
      DOUBLE PRECISION FUNCTION SEEPFACE(L,DEP, IC, IR)
      USE GLOBAL,       ONLY: NCOL, NROW, NLAY, IOUT, DELR, DELC, CR,
     +                        IBOUND, BOTM, HCOF, RHS, CC, CV, HNEW
      USE GWFPSZMODULE, ONLY: IPSZBOUND, STRMCELL, STRMCHD,
     +                        QSTORM
      USE GWFSFRMODULE, ONLY: NSTRM, ISTRM, STRM, HSTRM
      USE GWFNWTMODULE, ONLY: ICELL, IA, A
      USE GWFUPWMODULE, ONLY: HKUPW, SN
!     -----------------------------------------------------------------
      IMPLICIT NONE
      EXTERNAL DDHORIZ
C     -----------------------------------------------------------------
      !ARGUMENTS
C     -----------------------------------------------------------------
      DOUBLE PRECISION HH, TTOP, BBOT, THICK, CONDIF, DH,
     + DDHORIZ, SSTRLEN, CHECK, SSTRTOP,STAGELEV,DEP, DUM,
     + DUM2
      REAL DUM3
      INTEGER IC, IR,IL, L, IJ, N
C     -----------------------------------------------------------------
C1-----THIS SUBROUTINE SETS EACH STREAM CELL TO HAVE HEAD DEPENDENT
       BOUNDARIES TO ALLOW STAGE DEPENDENT FLOW FROM SOIL ZONE INTO
       STREAM
          DUM3=0.0
          CHECK=0.0
          IL=IPSZBOUND(IC,IR)
          STRMCELL(IC,IR)=L !NOTE: NEED TO MOVE STMT TO SZF
          SSTRTOP = STRM(3, L)
          STAGELEV=DEP+SSTRTOP
          DUM=HNEW(IC,IR,2)
          DO IL=1,1
          IF (STAGELEV.LT.HNEW(IC,IR,IL)) THEN
          IF (IBOUND(IC, IR, IL).GT.0)THEN
           IF (HNEW(IC,IR,IL).GT.BOTM(IC,IR,IL))THEN
           IF (STAGELEV.GT.BOTM(IC,IR,IL)) THEN
            HH= HNEW(IC,IR,IL)
            TTOP=(BOTM(IC,IR,IL-1))
            BBOT=STAGELEV
            THICK=TTOP-BBOT
            SSTRLEN=2*STRM(1,L)
            CONDIF=(HKUPW(IC,IR,IL)*SSTRLEN)/((DELC(IR)+DELR(IC))/4)
            DH=DDHORIZ(HH,TTOP,BBOT,IL) !DSN/DH
            IJ=ICELL(IC,IR,IL)
            CHECK=(CONDIF*SN(IJ)*THICK*(HH-STAGELEV))
           ELSE
            HH= HNEW(IC,IR,IL)
            TTOP=(BOTM(IC,IR,IL-1))
            BBOT=(BOTM(IC,IR,IL))
            THICK=TTOP-BBOT
            SSTRLEN=2*STRM(1,L)
            CONDIF= (HKUPW(IC,IR,IL)*SSTRLEN)/((DELC(IR)+DELR(IC))/4)
```

```
            DH=DDHORIZ(HH,TTOP,BBOT,IL) !DSN/DH
            IJ=ICELL(IC,IR,IL)
            CHECK=(CONDIF*SN(IJ)*THICK*(HH-BBOT))
           END IF
          END IF
          END IF
          END IF
          DUM3=DUM3+CHECK
          END DO
          SEEPFACE=DUM3
      END FUNCTION SEEPFACE
C     -----------------------------------------------------------------
      SUBROUTINE SEEPFACENWT(L, DEPNWT,KKPER, KKSTP)
C
      USE GLOBAL,       ONLY: NCOL, NROW, NLAY, IOUT, DELR, DELC, CR,
     +                        IBOUND, BOTM, HCOF, RHS, CC, CV, HNEW
      USE GWFPSZMODULE, ONLY: IPSZBOUND, STRMCELL, STRMCHD,
     +                        QSTORM, HITR
      USE GWFSFRMODULE, ONLY: NSTRM, ISTRM, STRM, HSTRM
      USE GWFNWTMODULE, ONLY: ICELL, IA, A, BB
      USE GWFUPWMODULE, ONLY: HKUPW, SN
C     -----------------------------------------------------------------
      IMPLICIT NONE
      EXTERNAL DDHORIZ
C     -----------------------------------------------------------------
      !ARGUMENTS
C     -----------------------------------------------------------------
      DOUBLE PRECISION DELH, HH, TTOP, BBOT, THICK, CONDIF, DH, DCDH,
     + DDHORIZ, SSTRLEN, CHECK2, SSSTRTOP,STAGELEV, FINTERFLOW, DEPNWT,
     + DUMM1,DUMM2, FIFCHECK, DUMM3, DUMM4,DUMM5, HOLD,DBDH, AO, RHSO
      REAL DUMM6
      INTEGER IC, IR,IL, L, IJ, KKPER, KKSTP
C     -----------------------------------------------------------------
C1-----THIS SUBROUTINE UPDATES THE RHS AND A NEWTON MATRIX WITH SZ
       INTERFLOW VALUES
          CHECK2=0.0
          DUMM6=0.0
          FIFCHECK=0.0
          SSSTRTOP = STRM(3, L)
          STAGELEV=DEPNWT+SSSTRTOP
          IC=ISTRM(3, L)
          IR=ISTRM(2, L)
        DO IL=1,1 !NOTE: IF.GT.1 WILL ALLOW FOR LAYERS IN SZ
          IF (STAGELEV.LT.HNEW(IC,IR,IL)) THEN
            IF (IBOUND(IC, IR, IL).NE.0)THEN
              IF (HNEW(IC,IR,IL).GT.BOTM(IC,IR,IL))THEN
              IF (STAGELEV.GT.BOTM(IC,IR,IL)) THEN
                BBOT=STAGELEV
              ELSE
                BBOT=(BOTM(IC,IR,IL))
              END IF
                HH= HNEW(IC,IR,IL)
                TTOP=(BOTM(IC,IR,IL-1))
                BBOT=(BOTM(IC,IR,IL))
                DBDH=0.0
```

```
           THICK=TTOP-BBOT
           SSTRLEN=2*STRM(1,L)
           CONDIF= (HKUPW(IC,IR,IL)*SSTRLEN)/
+                  ((DELC(IR)+DELR(IC))/4.0)
           DH=DDHORIZ(HH,TTOP,BBOT,IL) !DSN/DH
           IJ=ICELL(IC,IR,IL)
           CHECK2=(CONDIF*SN(IJ)*THICK*(BBOT-HH))
                 DUMM1=A(IA(IJ))
                 DUMM2=RHS(IC,IR,IL)
           DUMM3=(CONDIF*DH*THICK*(BBOT-HH))-(CONDIF*SN(IJ)*THICK)
           DUMM4=(CONDIF*DH*THICK*BBOT)-(CONDIF*SN(IJ)*THICK)-
+                (CONDIF*DH*THICK*DUMM5)
            !NOTE: DUMM4=DUMM3 PREV ITER; DUMM5=HOLD
                 A(IA(IJ))= A(IA(IJ))+DUMM3
                 BB(IJ)= BB(IJ) + DUMM3*HH
                 RHS(IC,IR,IL)= RHS(IC,IR,IL) - CHECK2
                 DUMM5=DUMM3*(HH-HOLD)+CHECK2
            !NOTE: DUMB5 SHOULD APPROACH FINTERFLOW
           FIFCHECK= (A(IA(IJ))-DUMM1)*HNEW(IC,IR,IL)-
                     RHS(IC,IR,IL)
           DUMM6=DUMM6+CHECK2
         END IF
      END IF
    END IF
  END DO
 QSTORM(IC,IR)= QSTORM(IC,IR)+DUMM6
  IF (QSTORM(IC,IR).GT.0.0) QSTORM(IC,IR)= 0.0
 !NOTE: THIS  PREVENTS FLOW BACK INTO SZ
 HITR=HNEW
 END SUBROUTINE SEEPFACENWT
```