Brief Description of DHSVM Updates at PNNL

Scott Waichler

February 3, 2003

This note briefly describes the most recent version of DHSVM as maintained by Scott Waichler and Mark Wigmosta at PNNL. In it I outline the nature of our recent major code changes, and the new input settings that are relevant to them. The complete tar file prepared for this tech transfer includes this file, all source files, makefile for Linux, and all input and output files for an example application for an arid watershed near the Hanford Site. My objective is to give you everything you need to compile, run, and check the model results.

1 DHSVM Changes

We modified DHSVM in several important ways to make it more suitable for the arid setting of Hanford. Infiltration capacity of the soil was limited by freezing temperatures and high soil moisture to simulate the ephemeral runoff-generating conditions at Hanford. Overland flow routing was modified to allow faster movement of runoff down the hillslope to simulate the ephemeral "flash floods" at Hanford. A "deep" groundwater layer was added to facilitate movement below the root zone and permit for slower lateral groundwater movement (Figure 1). This lower groundwater layer can be used to provide baseflow to the surface stream network in humid environments. In the arid Hanford application, it is not linked to the drainage network. Flow routing in stream channels was modified to permit infiltration into the channel bed and the simulation of "losing" streams. New variables were added to DHSVM to permit mapping of channel infiltration, recharge from the upper to lower groundwater layer, and deep recharge from the lower groundwater layer. I won't try to list all of the source files that have been changed, but I will highlight where major changes have been made and where the major new variables are set.

1.1 Limiting infiltration

New global variables `Options.FrozenSoil`: flag to potentially limit infiltration into soil. `FrozenSoil[y][x].airTemp`, `*.oldAirTemp`, `*.heatFlux`: frozen soil state and flux variables.

Source files *frozensoil.c*, *frozensoil.h*, *MassEnergyBalance.c*

Comments Various functions to limit infiltration based on 1) temperature/heat flux out of soil, 2) heat flux and upper layer soil moisture, or 3) snowpack presence. See Cary et al. (1978) for heat flux only method. Set compile flag to choose method via `ifdef` (see *makefile* and *MassEnergyBalance.c*). Input settings for heat flux method are under FROZEN SOILS INFORMATION SECTION.

1.2 Overland flow routing

The existence of overland flow and very sharp hydrographs in the arid setting require a faster overland flow routing than the default DHSVM mode. There are two basic considerations for moving surface water (Runoff) in grid-based model like DHSVM: (I) How much ponded water in a cell is to be moved in a timestep; (II) How frequently does water move from one cell to the next. Options implemented here for (I) are 1) all of the water in the cell is transferred; or 2) some of the water is transferred, according to a power law function. Options implemented here for (II) are a) water moves one cell per timestep; or b) water moves all the way down the hillslope, across all cells in a flowpath, as a cascade within the timestep. In summary:

1a. All water moved off the cell; moves one cell per timestep. `Options.HydraulicRouting` = **LINEAR**. Traditional DHSVM.

1b. All water moved off the cell; moves through all cells per timestep in order of decreasing elevation. `Options.HydraulicRouting` = **CASCADE_LINEAR**.

1c. All water moved off the cell if that cell was already wet at start of the timestep before upslope additions; moves through all cells per timestep in order of decreasing elevation. `Options.HydraulicRouting` = **CASCADE_LINEAR_LIMITED**.

2a. Some water moved off the cell; moves one cell per timestep. `Options.HydraulicRouting` = **POWERLAW**.

2b. Some water moved off the cell; moves through all cells per timestep in order of decreasing elevation. `Options.HydraulicRouting` = **CASCADE_POWERLAW**.

In the original DHSVM, water moved as 1a. Subsequently, Mark added the capability to move water as 2a (see Szilagyi and Parlange (1999)). Comparing the two options, 1a is the upper limit on water movement rate, while 2a slows it down. For both options, if the timestep or number of cells in a hillslope path is too large, the user may experience too-slow movement of water to the channels, and difficulty in reproducing peaks in observed hydrographs.

The latest version here by Scott and Mark allows movement by 1b or 2b. In the case of 1b, all of the Runoff in the watershed will be moved down to the stream channel and be intercepted at the next timestep (linear cascade). In the case of 2b, some of the water on each cell will remain according to the power law function, and will ultimately infiltrate (power law cascade). For both 1b and 2b, the movement of water in a cascade downslope to the receiving channel or basin pour point is accomplished by processing the grid cells in descending order of elevation. This results in moving water out of the highest cells first, then down through the lowest cells, all within the same timestep. Option 1c, a limited linear cascade, is an empirical compromise between 1b and 2b, wherein all ponded water moves off the cell if and only if that cell was already wet before upslope contributions flowed into it.

If a grid cell contains a channel, no movement out of that cell is allowed, with the result that the water (Runoff) will remain on the cell until RouteChannel() is called the next timestep. A better solution perhaps would be to have the water enter the stream channel in the same timestep, and have the channel routing process all of the water intercepted from the subsurface and surface in the same timestep. This would require moving the call to RouteChannel() after RouteSurface() in main(). It is not there now for some good reason, which is now forgotten. It probably has to do with avoiding flow loops when road channels are involved.

Option 1b will probably rarely be used, as it results in little difference between input hydrograph (generation of distributed Runoff on the cells) and the output hydrograph (channel interception). These hydrographs would be exactly the same under 1b if channel interception took place after surface routing in the main program. Option 2b is a true kinematic wave with a delayed rise, peak, and gradual recession. Numerical stability of the wave should be checked in practice. Remember that normally the kinematic wave is solved on the order of seconds or minutes, so some unusual values of the power law coefficients might be needed to slow down flow across the hillslope to realistic rates for a given dx, dt, and hillslope length. Both 1b and 2b are a big departure from the original DHSVM logic. Be careful! Ultimately, none of these solutions are satisfactory for quick runoff in humid environments, where the conceptual model is usually macropore flow and translatory flow dependent on moisture content in the hillslope soil.

New global variables `Options.HydraulicRouting`: flag for type of overland flow routing. `Map.NumCells`: number of cells within basin. `Map.OrderedCellsRowCol`: matrix that is `NumCells` x 2 in size, contains the row and col of all basin cells sorted by decreasing elevation.

Source files *InitTerrainMaps.c*, *RouteSurface.c*

Comments Overland flow routing options are set in Hydraulic Routing field in main input file. See Szilagyi and Parlange (1999) for information about the algorithm and possible parameter values.

1.3 Channel routing and infiltration

Channel routing may be done via three methods: 1) **LINEAR** for linear reservoir method for routing streamflow; 2) **POWERLAW** for powerlaw storage-discharge method; 3) **FTABLE** for flow table (ftable) method. Only the ftable method allows channel infiltration, or loss from the stream to the soil column below it. The new ftable method allows the simulation of losing streams. If this option is set, then the ftable and multiplier files defined under FTABLE SECTION in the main input file are read and used to do all channel flow routing, which involves a simultaneous solution of inflow, outflow, and infiltration. Stream segment properties that are relevant to routing and independent of water depth are defined in the multipliers file. Routing properties that depend on water depth are defined in the ftable file. Each stream segment (channel reach) has a unique listing in the multipliers file, which includes a cross-reference to the correct flow table to use in the ftable file. For a given depth of water in the stream segment, factors for outflow and infiltration are read or interpolated from the flow table, then multiplied by the corresponding values in the multipliers file to obtain the potential flows. These potential flows are then prorated to the available volume of water in the reach, if necessary, to obtain the final outflow and infiltration.

In the multipliers file, Number of Tables is the number of stream segments in the stream network, and number of rows in this file. TableNum is the channel reach number, or segment ID in the stream network file. Type is the number of the ftable to use for look-up. Depth is set to unity. SurfArea, Vol, and Infiltration are equal to the length of the channel segment (=unit-width area). DownstreamQ is the square root of the gradient, S_0.

In the ftable file, Number of Tables is the number of discrete channel types you wish to describe for purposes of flow routing and infiltration; each type has its own look-up table. Area (m) is the width of the stream channel class and should match the variable in the stream class file. Vol is the water depth (m), DownstreamQ is Manning's equation less the slope term (see below), and Infiltration is an empirical factor for infiltration. During simulation, the model linearly interpolates between these depth points (rows in each table) to compute the potential outflow and infiltration for the channel segment. Outflow Q is defined as Manning's Equation:

$$Q = \mathsf{Ftable.DownstreamQ} \times \mathsf{Multiplier.DownstreamQ}, \tag{1}$$

$$\mathsf{Ftable.DownstreamQ} = \frac{R^{\frac{2}{3}}A}{n}, \tag{2}$$

$$\mathsf{Multiplier.DownstreamQ} = \sqrt{S_0}; \tag{3}$$

where R =hydraulic radius, A =cross-sectional area, S_0 =slope, and n = roughness coefficient. In practice, the channel is assumed to have a rectangular cross-section. DownstreamQ values in the ftable file must be computed as a preprocessing step. Infiltration values are set to zero for an impermeable streambed.

New global variables `Options.ChannelRouting`: streamflow routing method. `segment.outflow`: outflow to next channel segment. `segment.recharge`: infiltration through channel bed. `SoilMap[y][x].CumChannelLoss`: cumulative infiltration through the channel bed.

Source files *channel.c*, *ftable5.cpp*, *ftable.h*, *MainDHSVM.cpp*, *RouteSubSurface.c*

Comments Channel flow routing options are set in Channel Routing field in main input file.

1.4 Recharge and Deep Groundwater

In the new DHSVM, the user may allow some of the water in the shallow system to percolate below the bottom soil layer and enter a deep groundwater system. The rate at which this shallow recharge happens is governed by Darcy's law and Groundwater Conductivity set in the soil table. Once the amount of leakage is computed for the timestep, the water is transferred to the lower groundwater layer in the same timestep, with no additional rate- dependent vertical percolation. The thickness of the lower aquifer zone is not defined, and the deep groundwater height (saturated thickness) is the height of the water column unadjusted for porosity. Potential outflow of deep groundwater from a cell is computed as the sum of deep lateral outflow, a function of the layer's transmissivity, and deep leakage, or ultimate recharge. Deep recharge is given preference over lateral outflow if the available deep groundwater is insufficient to provide for both. If no deep recharge is desired, Base Layer Conductivity in the input is set to zero. The deep layer requires its own DEM and flowpath grids to be input; these may be set to the surface/shallow subsurface flowpath map if you can't justify using different grids, as would usually be the case. A stream map file must also be specified for the deep layer, and should either be the stream map file for the surface stream channels, or a dummy file with one row entered for a cell outside the mask. The first option is used in most applications, where the motivation to use the deep layer is to provide baseflow for streams. The second option was used for Hanford, where all streams are ephemeral and lose water.

New global variables `SoilMap[y][x].GwRecharge`, `*.CumGwRecharge`: percolation downward from the upper saturated zone. `Groundwater[y][x].depth`, `*.depth`, `*.Dem`, etc.: parameters for groundwater (shallow) recharge. `Groundwater[y][x].deepLoss`, `*cumDeepLoss`: parameters for ultimate (deep) recharge.

Source files *Groundwater.c*, *groundwater.h*, *RouteSubSurface.c*.

Comments Input settings for recharge from shallow to deep layer are under SOIL tables in main input file: Groundwater Conductivity, Groundwater Conductivity Lat. Input settings for deep groundwater layer are under GROUNDWATER INFORMATION SECTION and SOIL tables in main input file: Gwater DEM File, Gwater Flow Direction File, Gwater To Channel File, Initial Groundwater Depth, Base Layer Conductivity.

1.5 Soil Evaporation Based on Resistance

Source files `SoilResistanceEvap.c`, `MassEnergyBalance.c`.

Comments Choose old or new soil evaporation function by commenting out in `MassEnergyBalance.c`. See de Silans et al. (1989); Peters-Lidard et al. (1997) for information about the algorithm and possible parameter values.

2 Example Application

This example application involves upper Cold Creek west of Hanford, named Basin 1 (Figure 2). Input files are listed in Table 1. DHSVM input grids have 200 m resolution. Two soil types were specified: one with an impermeable base layer boundary to represent areas where basalt lies above the water table, and one with essentially unlimited base layer hydraulic conductivity to represent areas where the water table is the lower boundary. All other properties of the two soil types were defined alike. Vegetation consisted on one type, sagebrush-steppe, with 50% coverage of the ground surface.

Some of the output files are provided for you check your output against. If you find significant discrepancies, please let me know. I've also provided an R script, *postproc1c.R*, that you may be interested in (R is essentially an open-source version of S-plus, www.r-project.org). It reads in the map output files from DHSVM and plots it up with basin boundaries and stream lines. I wrote this to give me an alternative to Arc for displaying and printing grids. The files *stdir_1.Rdat* and *maskbndry_1.Rdat* are the streamlines and boundaries, respectively. These are included in the plots to give the reader more visual references. These ascii files were created using Arc's "ungenerate" command for printing out a text file with point listings from a line coverage, then using the Perl script *process_ungenerate1.pl* to convert the Arc output to a format that R could handle more readily.

References

Cary, J., G. Campbell, and R. Papendick: 1978, Is the soil frozen or not? an algorithm using weather records. *Water Resources Research*, **14**, 1117–2244.

de Silans, A. P., L. Bruckler, J. Thony, and M. Vauclin: 1989, Numerical modeling of coupled heat and water flows during drying in a stratified bare soil—comparison with field observations. *Journal of Hydrology*, **105**, 109–138.

Peters-Lidard, C., M. Zion, and E. Wood: 1997, A soil-vegetation-atmosphere transfer scheme for modeling spatially variable water and energy balance processes. *Journal of Geophysical Research*, **D4**, 4303–4324.

Szilagyi, J. and M. B. Parlange: 1999, A geomorphology-based semi-distributed watershed model. *Advances in Water Resources*, **23**, 177–187.

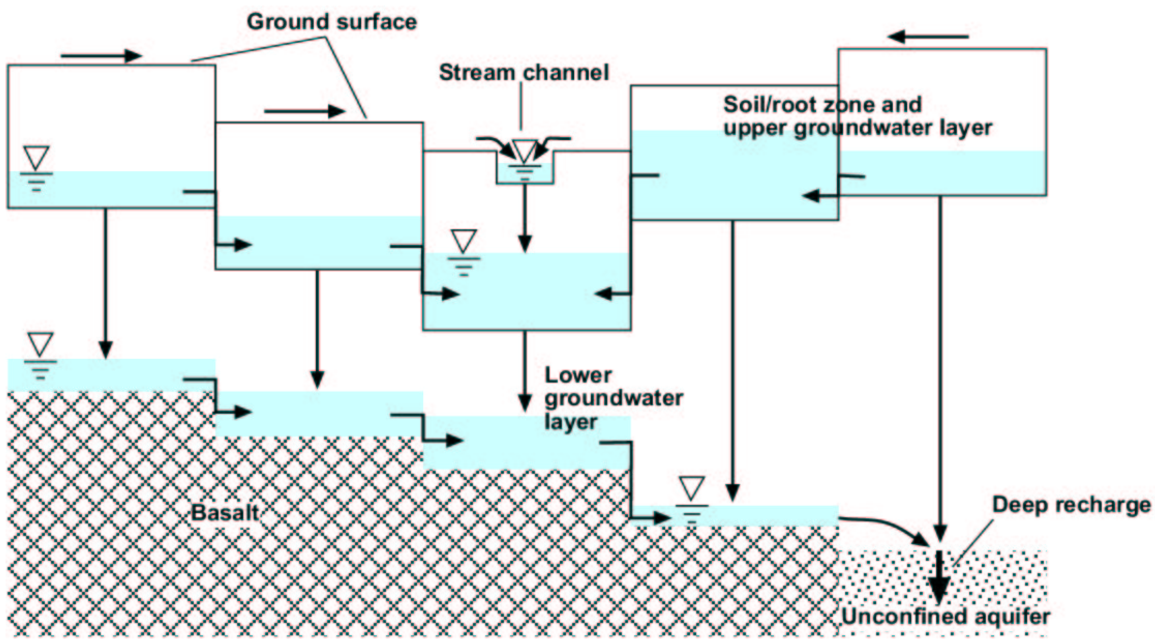

Figure 1: Water pathways in DHSVM. Overland flow, if any, is subject to infiltration in lower grid cells and interception by stream channels. Subsurface flow moves in an upper ("shallow") layer and a lower ("deep") layer. If the shallow water table is below the streambed, the stream may lose water to the shallow water table. Travel time from the streambed to the water table is neglected and the infiltrated water is added immediately to the shallow water table. Similarly, water percolates through the upper groundwater layer as Darcian flow, but vertical travel time within the lower layer is neglected, and recharge from the upper layer is added immediately to the lower layer. For the current recharge estimates, hydraulic conductivity of zero is assigned to the basalt, and all deep groundwater follows a lateral flowpath until a cell defined as unconfined aquifer is encountered. A very large hydraulic conductivity for recharge is defined for the unconfined aquifer, resulting in recharge of all lower groundwater entering such a cell. In this application, the flowpath map for the deep groundwater was set equal to the surface and shallow flowpath map, but the model allows separate definition, as depicted here.

Lower Boundary Type, Greater Cold and Dry Creek Basin (200 m grid)

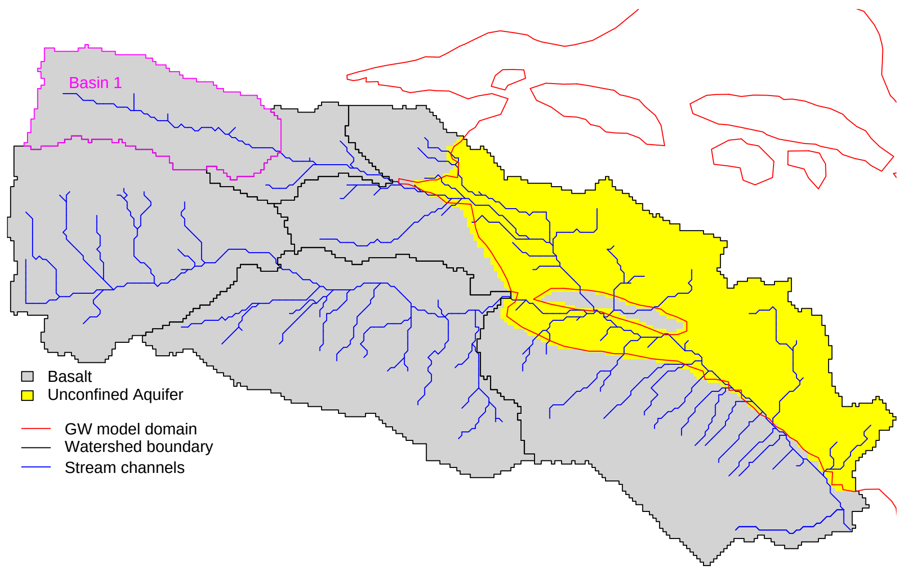

Figure 2: Lower boundary specification for watershed modeling. Example application simulates Basin 1, upper Cold Creek.

File	Purpose
input.dat	Main ascii input file
dem_1.bin	DEM for ground surface and deep groundwater layer
mask_1.bin	Mask grid
flwdir_1.bin	Flow direction grid for surface and both groundwater layers
soildepth_1.bin	Soil depth grid
soiltypelb_1.bin	Soil type grid
vegtype_1.bin	Vegetation type grid
stream-map_1.dat	Stream map table
multipliers_200m_1.dat	Multipliers table to be used with flow tables
ftables4.dat	Flow tables for channel routing
stream-net_1.dat	Stream network table
stream-class.dat	Stream classes table
hms_final_hourly_wy95.dat	Meteorology timeseries

Table 1: Input files for example application.