Download the tutorial.tar.gz file from the dhsvm ftp site

ftp://ftp.hydro.washington.edu/pub/dhsvm/

Select *tutorial.3.1.tar.gz* and save it to your working directory.

Unzip and untar the file

```
unix% gunzip tutorial.3.1.tar.gz
unix% tar -xvf tutorial.3.1.tar
```

This should expand into the following directory structure:

tutorial3.1/
- *amlscripts/*
- *input/*
- *configfiles/*
- *metfiles/*
- *output/*
  - *GMTscripts/*
- *programs/*
- *source_codes/*
- *modelstate/*
- *DHSVM3.1_Test_Site_Intro.pdf/*

Create surface routing file for the urban module

Run the program 'find_nearest_channel.c'
usage:

```
find_nearest_channel nrows ncols binary_flowd_file
binary_mask_file stream_map_file n_header_map_file
```

```
unix% cd programs
unix% gcc find_nearest_channel.c -o find_nearest_channel
unix% cd ..
unix% programs/find_nearest_channel 213 164 input/flowdir.bin
input/mask.bin stream.map 9
```

where:

flowd_file is a binary flow direction file in the same format as the DHSVM mask file. Make sure that the flowd_file is free of sinks, etc, and flowdirection is assumed to be from ARC-INFO, i.e. 1 to 128.

binary mask_file and ***stream_map file*** are the DHSVM specific input files for mask and stream_map file, respectively.

n_header_map_file are the number of header lines in the stream map file, i.e. lines starting with # . Enter 0 if there are no header lines. Caution: make sure you are referring to the map file not the network file

2) Fill in the path to the flow_routing.txt file under the section of VEGETATION in the configuration file.

Create model states

Compile the MakeModelStateBin.c:

unix% `make -f Makefile.ModelState`

Create the Interception, Snow and Soil state files for the date specified in InitialState.txt

unix% `./MakeModelState modelstate/ initialstate.txt`

InititalState.txt provides a sample input file set up for Springbrook Creek, assuming no interception storage, no snow for 210 days and 35% volumetric soil moisture in all layers.

The 'initialstate' file MUST contain the following information:

- path for output file (change it if not ../modelstate)
- date for the model state, in mm/dd/yyyy-hh
- number of rows (ny) and number of columns (nx)
- maximum number of vegetation layers
- rain interception in m for each vegetation layer
- snow interception in m for top vegetation layer
- snow cover mask
- number of days since last snow fall
- snow water equivalent in m
- liquid water content in m of bottom layer of snowpack
- temperature in C of bottom layer of snow pack
- liquid water content in m of top layer of snowpack
- temperature in C of top layer of snow pack
- cold content of snow pack
- maximum number of root zone layers
- volumetric soil moisture content for each layer (including the layer below the lowest root zone layer)
- temperature in C at soil surface
- soil temperature in C for each root zone layer
- ground heat storage
- runoff

Create initial channel state files

*Also have to change the path to which the output file is stored.

If necessary, make the script MakeChannelState.scr executable:

unix% `chmod 755 MakeChannelState.scr`

usage: MakeChannelState.scr <StreamNetworkFile> <InitialDepth> <Output Date String: MM.DD.YYYY.hh.mm.ss>

To run the script (calculates volume of water in each channel segment, given an assumed uniform initial depth in meters):

```
unix% ./MakeChannelState.scr ../input/stream.network 0.25
10.01.1995.00.00.00
```

Create shadow and skyview files

If necessary, make the script MakeChannelState.scr executable:

```
unix% chmod 755 run_solar_programs_daily.scr
```

Edit *run_solar_programs_monthly.scr***,** change the parameter settings and the location of DEM map (should be in ../input/dem) and output location (../input).

```
unix% ./run_solar_programs_daily.scr
```

Run the model

```
unix% cd ../source_codes/
unix% make
unix% cd ..
unix% source_codes/DHSVM configfiles/INPUT.3.1.SpringBrook
```

Note that the sediment option is turned off in this sample configuration file for the Springbrook creek. If the sediment module or mass wasting option is going to be implemented, please follow the *Tutorial for DHSVM 3.0* (*ftp://ftp.hydro.washington.edu/pub/dhsvm/*)

Use GMT to view output hydrographs/basin average time series

```
unix% output/GMTscripts
```

 # Produces a graph of flow (plot_flow.ps)

```
unix% plot_flow.scr
```

 # Close ghostview window

```
unix% ctrl-c
```

Produces a graph of precipitation over the basin (plot_precip.ps)

```
unix% plot_precip.scr
```

Produces a graph of snow water equivalent (plot_swe.ps)

```
unix% plot_SWE.scr
```

Each of these can also be viewed in ghostscript, e.g.:

```
unix% gs  plot_flow.ps
```

Use GMT or ArcInfo to view output spatial images

The script **plot_modelmap.scr** allows you to plot maps, by changing the script slightly. It works if you designated only one date per map number. For each map you need to designate the map name, the name of the output file, the name of the input file and the name of the color plot table (cpt). Create the Soil Moisture Map first.

```
unix% xemacs &
```

Edit **plot_modelmap.scr** so that the Soil Moisture map name, output file, input file and cpt file do not have a "#" in front of it. Add a "#" in front of all the other map file names. Save the file, but leave it open. The paths to each file may require modification.

```
unix% plot_modelmap.scr
unix% ctrl-c        <to close ghostview window>
```

Repeat these steps for any other maps for which you created files.
Each of these can also be viewed in ghostscript, e.g.:

```
unix% gs Map.1.Soil.Moist.ps
```