Session 1: Introduction

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

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

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

Unzip and untar the file

Right click on tutorial.tar.gz and select WinZip >> Extract to Here (or use alternative unzipping software).

This should expand into the following directory structure:

```
tutorial/
        arcinfo/
                rainysoil/
                rainyveg/
                rainydem/
                rainymask/
                roads/
                rainy10m/
                amlscripts/
                info/
        configfiles/
        metfiles/
        output/
                GMTscripts/
        programs/
        source/
                mwm/
                dhsvm_2.0.1.tar.gz
```

Run arc/info and view database files

Convert NODATA from -9999 to 0 in the mask file
1) Open the rainymask grid in arcmap
Open toolbox > spatial analyst tools > map algebra > single map output
(note requires spatial analyst extension to be turned on in tools=>extensions menu)
Paste the following instruction into the text box:

```
con (isnull (rainymask), 0, rainymask)
```

enter “rainymask2” in the output textbox (in the same directory as rainymask) hit ok

2) Open toolbox > conversion tools > from raster > raster to ascii
set input to “rainymask2”
set output to “rainymask.txt” (in the same directory)
hit ok

Repeat step 2 for rainysoil, rainyveg, and rainydem

Convert text grids to binary

This is a two step process. In the first step, the headers are removed from the *.txt files. Headers look like

```
ncols         67
nrows         52
xllcorner     394379.92247655
yllcorner     761247.95513744
cellsize      150
NODATA_value  -9999
```

The second step converts the remaining gridded information into binary files used by DHSVM.

open rainymask.txt in text editor (double click on file name)
cut out the header and paste into a new text file
save the header file as rainyhdr.txt
re-save rainymask.txt with no header (initial line should be a line of data)

cut headers out of rainysoil, rainyveg, and rainydem. These headers are identical to the one already cut out of rainymask, so there is no need to save these headers.

Use myconvert.exe to create dhsvm binary input grids:

usage: myconvert <source_format> <target_format> <source_file> <target_file> <number_of_rows> <number_of_columns>

Open a Dos window, in the appropriate directory, type
(myconvert.exe will need to be in that directory)

```
myconvert ascii char rainymask.txt rainymask.bin 52 67
myconvert ascii float rainydem.txt rainydem.bin 52 67
myconvert ascii char rainysoil.txt rainysoil.bin 52 67
myconvert ascii char rainyveg.txt rainyveg.bin 52 67
```

Create model states

Usage: MakeModelState <infofile>
The info file MUST contain the following information:

- path for output file
- 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

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

```
MakeModelStateBin InitialState.txt
```

This will create the Interception, Snow and Soil state files for the date specified in InitialState.txt

Move all files with names including the word “state” into `../modelstate`

Session 2: Stream, Soil and Road Inputs

Start a Dos Command Window

```
  cd arcinfo
  arc
```

Set the path for aml scripts:

```
Arc: &amlpath amlscripts\
```

Create stream and soil inputs

usage: CREATESTREAMNETWORK <dem> <wshed> <soildepth> <stream network> <MOUTH|MASK> {source area} {min depth} {max depth}

```
Arc: &run createstreamnetwork rainydem rainymask soild streams
MASK 100000 .76 1.5
```

Note: all grids should have same domain and resolution
- dem: a grid of basin elevations
- wshed: EITHER a basin mask file, or a grid designating the location of the basin mouth, as indicated by the keyword MOUTH | MASK, if MOUTH is specified, the mask file will be created.
- soild: grid of soil depth, if it doesn't exist it will be created as a function of cumulative drainage area and slope, min and max depth must then be specified.
- streams: an arc coverage of stream locations, if it doesn't exist the coverage will be created using the specified constant threshold area (source area)

You will be prompted to see if you want to continue. Type `y`.

This step creates the files stream.network.dat and stream.map.dat that are needed by DHSVM.

Create road inputs

This step creates the files road.network.dat and road.map.dat that are needed by DHSVM.

usage: CREATEROADNETWORK <dem> <soildepth> <road network>

```
Arc: &run createroadnetwork rainydem soild roads
```

You will be prompted to see if you want to continue. Type `y`.

Designate stream save indicator

Routing results for this stream segment will be placed in the stream output file if the keyword SAVE appears in the last column of the stream.network.dat file, located in the arcinfo directory. Using the text editor, add

SAVE "RAINY CREEK"

to the last column of Rainy Creek, stream segment 1.

Export soildepth grid and convert to binary

Open toolbox > conversion tools > from raster > raster to ascii
set input to soild
set output to soild.txt (in the same directory)
hit ok

open soild.txt in text editor (double click on file name)
cut out the header

use myconvert.exe to change soild grid to binary

```
myconvert ascii float soild.txt soild.bin 52 67
```

Edit road and stream class files

These are ascii files that contain a look-up table of channel hydraulic properties for each road or stream class. The following columns are necessary for each channel class: <Channel Class> <Hydraulic Width (m)> <Hydraulic Depth (m)> <Manning's friction coefficient> <Max Infiltration (m/s)>. The road class file requires one additional column <CrownType> to describe the flow characteristics of the surface. The type must be coded in ALL CAPS as "INSLOPED", "OUTSLOPED" or "CROWNED"

The channel hydraulic width and depth are currently hard-wired in the createroadnetwork and createstreamnetwork scripts, as follows:

Streams:

#Class	Width	Depth
1	1.5	1.0
2	2.0	1.5
3	3.0	5.0
4	5.0	7.5
5	2.0	1.5
6	5.0	4.0
7	7.5	7.0
8	10.0	10.0
9	15.0	13.0

Roads:

#Class	Width	Depth
1	0.5	0.5
2	0.5	0.5
3	0.25	0.25
4	0.25	0.25
5	0.25	0.25
6	0.25	0.25
7	0.25	0.25
8	0.25	0.25
9	0.25	0.25

Edit these tables (road.class.dat and stream.class.dat) in text editor to add the friction, infiltration, and crown type columns.

The maximum infiltration rate is not used for the stream class file, but it must be specified as a place holder. Manning's roughness can vary from 0.025 to 0.15 for natural channels. Infiltration into the roads may be assumed to be 0 m/s for a starting point (i.e. 100% impervious).

Create initial channel state files

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

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

From the modelstate directory type

```
 MakeChannelState ..\arcinfo\stream.network.dat
..\arcinfo\stream.class.dat 0.25 10.01.1990.03.00.00
```

Session 3: Running the model

Complete the configuration file

Using the text editor on the file \tutorial\configfiles\INPUT.rainycr
Look through the file and fill out:
path names
Model Start = 10/01/1990-03
Model End = 10/01/1994-03
Number of Model States: 1
Model State Date 1 = 10/01/1994-03

Run the model

From the arcinfo directory type

```
DHSVM ..\configfiles\INPUT.rainycr
```

Session 4: Model Output and Calibration

Copy the model state files and backup previous output

```
unix% cd tutorial/
unix% cp output/*1994* modelstate/
unix% mv output output1
unix% mkdir output
unix% mv output1/GMTscripts/* output
```

Edit the configuration file

```
unix% xemacs &
```

Change:
Model Start = 10/01/1994-03
Model End = 10/01/1996-03
Model State Date 1 = 10/01/1995-03
Fill out the Model Maps Section. Refer to the handouts for the Map Variables. (Map 512 is not working.)

Run the model

```
unix% cd source/dhsvm_2.0.1
unix% DHSVM ../../configfiles/INPUT.rainycr
```

Use GMT to view output hydrographs/basin average time series

```
unix% cd ../../output/GMTscripts

unix% plot_flow.scr              # Produces a graph of the Rainy Creek flow
                                 (plot_flow.ps)
unix% ctrl-c <to close ghostview window>
unix% plot_precip.scr            # Produces a graph of precipitation over the basin
                                 (plot_precip.ps)
unix% plot_SWE.scr               # Produces a graph of snow water equivalent
                                 (plot_swe.ps)
```

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 three types of maps, by changing the script slighty. 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.

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

Repeat these steps for the other 2 types of maps.

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

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

Experiment with adjusting calibration parameters and re-run

Open the configuration file in emacs and edit the parameters

```
unix% xemacs &
```

Save the previous output and rerun DHSVM

```
unix% mv output output2
unix% mkdir output
unix% cd ../../source/dhsvm_2.0.1
unix% DHSVM ../../configfiles/INPUT.rainycr
```

Run alternate road or vegetation scenario

No Roads:

Open configuration file in emacs and edit Road Network Section by commenting out the three path and file names (place a # before each `Road`).

```
unix% xemacs &
```

New Vegetation Scenario:

Open the rainyveg.txt in emacs and change all the vegetation to type 7 (Woodland Grass) or type 8 (Closed Shrub).

```
unix% xemacs &
```

Remove the header:

```
unix% cd /arcinfo
unix% tail +7 rainyveg.txt >! newrainyveg.txt
```

Convert to binary:

```
 unix% /programs/myconvert ascii char newrainyveg.txt
rainyveg.bin 52 67
```

Open the configuration file in emacs and edit the file names

```
unix% xemacs &
```

Save previous output and rerun DHSVM.

Give colleen locations of your final hydrographs

Session 5: Sediment Model

Rerun DHSVM with sediment output turned on:

```
unix% xemacs configfiles/INPUT.rainycr &
```

Adjust:

```
Sediment = TRUE
Model Start  = 10/01/1995-03
Model End = 06/01/1996-03
```

Rerun DHSVM. Saturated fraction and runoff files will now be output to your output directory. Meanwhile:

Export 10 m dem grid and convert to binary

```
Arc: gridascii rainy10m rainy10.txt
```

Remove the header

On freebsd machine:

```
unix% cd ../arcinfo
unix% tail +7 rainy10.txt >! temp1
unix% rm rainy10.txt
unix% mv temp1 rainy10.txt
```

Convert to binary

```
 unix% ../programs/myconvert ascii float rainy10.txt
rainy10.bin 52 67
```

Adjust the time period, input and output paths in tutorial/configfiles/INPUT.mwm in emacs

```
Model Start   =  11/27/1995-00     # Model start time (MM/DD/YYYY-HH)
Model End     =  11/30/1995-21     # Model end time (MM/DD/YYYY-HH)
Maximum Iterations =  1            # Max iterations of 1 means deterministic mode
Dem File      = ../../arcinfo/rainy10.bin
```

```
Basin Mask File = ../../arcinfo/rainymask.bin
Soil Map File = ../../arcinfo/rainysoil.bin
Soil Depth File = ../../arcinfo/soildepth.bin
Soil Saturation Path  = ../../output/
Vegetation Map File = ../../arcinfo/rainyveg.bin
Output Directory     =  ../../output/
```

Untar and compile the source code

```
unix% cd source/
unix% gunzip mwm_0.99.tar.gz
unix% tar -xvf mwm_0.99.tar
unix% cd mwm/
unix% make
```

Run the model in deterministic mode

```
unix%: MWM ../../configfiles/INPUT.mwm
```

View output sediment map & failures

Sample plotting scripts similar to the ones used above are available in tutorial/output/GMTscripts. The date to be plotted is currently set to 11/28/1995 in each of these scripts. If desired, change the date by editing the script in xemacs to look for a different file.

```
unix% cd ../../output/GMTscripts
unix% Plot.SatFrac.scr         # Produces a map of the saturated soil fraction as
                               output from DHSVM (Map.SatFrac.ps)
unix% Plot.FailureMap.scr      # Produces a map showing failure locations
                               (Map.Failure.ps)
unix% Plot.FailureTrack.scr    # Produces a map showing change in soil depth
                               due to failure (Map.Soildepth.ps)
```

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

```
unix% gs Map.Failure.ps        # Stochastic mode
```

Adjust mode in tutorial/configfiles/INPUT.mwm

```
Maximum Iterations       =  100
```

Run the model in stochastic mode

```
unix% MWM ../../configfiles/INPUT.mwm
```

View failure risk maps

More GMT scripts, same as above:

```
unix% Plot.Probability.scr     # Produces a map of the probability of landslide
                               occurrence (Map.probability.ps)
```

```
unix% Plot.CumDepth.scr          # Produces a map showing track of all possible
                                 failures, as reflected in the change in soil depth
                                 (Map.cumdepth,ps)
unix% gs Map.probability.ps
```