

## A Brief Tutorial for Running DHSVM-RBM

### Files Structure

After unpacking the file, the source code and supporting files will be in **DHSVM\_RBM**:

The sub-directories within this directory are:

<b>Create</b>	- contains the Fortran 90 source and <b>makefile</b> that build the forcing file for <b>RBM</b>
<b>Output</b>	- contains the sample output for the examples provided here
<b>DHSVM</b>	- contains the source code and <b>makefile</b> for the hydrologic model, <b>DHSVM</b> , which generates the hydrologic data and meteorological data requires by the stream temperature model, <b>RBM</b>
<b>RBM</b>	- contains the source code <b>makefile</b> for the stream temperature model, <b>RBM</b>
<b>Scripts</b>	- contains the pre- and post-processing scripts
<b>tutorial DHSVM v3.1.pdf</b>	
<b>tutorial DHSVM-RBM.pdf</b>	

### Run the Model

The stream temperature model, **RBM**, simulates water temperatures using forcing data and topology created by the distributed hydrologic model, **DHSVM**. The files include:

#### **Run DHSVM, create RBM Forcing Input**

The hydrologic and meteorological forcing files are generated from the **DHSVM** model.

\*\*\*\*\*

For DHSVM beginners, follow the tutorial **tutorial DHSVM v3.1.pdf** to prepare DHSVM input and run the model.

\*\*\*\*\*

To turn on temperature modeling:

- I. In the configuration file, modify the section [**OPTIONS**]. Make

**Stream Temperature = TRUE**

If riparian vegetation is taken into account, then

**Canopy Shading = TRUE**

Otherwise,

**Canopy Shading = FALSE**

- II. If **Canopy Shading = TRUE**, create a **riparian network file** (e.g. **DHSVM/input/rveg.baseline.network**) that specifies the riparian parameter values for every stream segment, and enter the path to this network file in the DHSVM configuration file <stream network> section:

```
##### stream network # #####
```

```
# The following three fields are only used if Flow Routing = NETWORK
```

Stream Map File = ../input/stream.map.dat  
Stream Network File = ../input/stream.network.dat  
Stream Class File = ../input/adjust.classfile  
Riparian Veg File = ../input/ rveg.baseline.network

In the riparian network file, the parameter value from left to right (column) is:

- Stream Segment ID
- Tree Height (m)
- Vegetation Buffer Width (m)
- Monthly Extinction Coefficient (0 ~ 1) from Jan to Dec ( 12 columns)
- Overhang Coefficient that is a percentage of tree height used to represent overhanging canopy (0 ~ 1)
- Canopy Bank Distance indicating the distance from bank to canopy (m)
- Channel Width (m).

**Monthly light extinction coefficient (k)** is monthly varied in unit  $m^{-1}$ . In application, this parameter should be treated as a calibration parameter. In prior studies, we used LAI/64 as the initial guess of the k values.

Note from literature: *The value of k is typically between 0.001 and 0.1  $m^{-1}$ , representing a low-to-high canopy density (Federer, 1971; Baldocchi et al., 1984; Pomeroy and Dion, 1996; Link et al., 2004), and 0.05  $m^{-1}$  is used to represent an average condition.*

III. Compile DHSVM if use binary I/O:

`make -f makefile_for_binary`

If use netCDF I/O:

`make -f makefile_for_netcdf`

IV. Run DHSVM.

`DHSVM/sourcecode/DHSVM3.2 DHSVM/config/Template.Input.GriddedMet`

The output forcing files will be in the directory **DHSVM/output**.

Outputs of RBM forcing files include:

*\*\* Each file contains time series for each stream segment. The segment ID is consistent with the ID used in the [stream.network.dat](#) file. \*\**

NSW.Only	net shortwave radiation, $W/m^2$
ATP.Only	air temperature, $^{\circ}C$ , for each computational interval and
NLW.Only	net longwave radiation, $W/m^2$
VP.Only	vapor pressure, <i>Pascals</i>
WND.Only	wind speed, m/s
Inflow.Only	inflow, $m^3/s$
Outflow.Only	outflow, $m^3/s$

The format \*.Only file is as follows:

<MM/DD/YYYY-HH:MM:SS (start date)> <MM/DD/YYYY-HH:MM:SS (end date)>  
<segment number(n), n=1, number of segments>  
<MM.DD.YYYY-HH:MM:SS (initial time)>  
<MM.DD.YYYY-HH:MM:SS (data time), dhsvm\_output(n), n=1,number of segments>

where dhsvm\_output(n) are the simulated values of air temperature, shortwave radiation, longwave radiation, vapor pressure, wind speed, inflow and outflow, for each segment. A record is needed for each time step.

\*\*\*\*\*  
NOTE: The start date of above forcing files, \*.Only, is one day behind the start date specified in the DHSVM configuration file, and so is the start date of output stream temperature simulations.  
\*\*\*\*\*

### Create Stream Topology

In this directory, “**Scripts**”, compile the program, **make\_stream\_connectivity.c**, to create the stream topology file, **convergence.txt**.

```
gcc make_stream_connectivity.c -o make_stream_connectivity
```

Copy the executable, **make\_stream\_connectivity**, to “**Work\_Space**”.

Execute the program::

```
./make_stream_connectivity <map> <network> <outdir> <no_segments> <skip>
```

Where **<map>** is the stream map file, **<network>** is the stream network file. Both files are input files required by **DHSVM**, and are stored in this directory, “**DHSVM/input**”. **<outdir>** specifies the output directory, **<#segments>** is the total segment numbers, **<skip>** is the number of lines in the header of the stream map file, **<map>**.

Example:

```
./make_stream_connectivity ../DHSVM/input/stream.map  
../DHSVM/input/stream.network ./ 1 9
```

The format for the topology file is as follows:

<segment id> <destination segment id> <length-meters> <depth-meters> <avg  
azimuth> <upstream segment id>

\*\*\*\*\*  
NOTE: This file, **convergence.txt**, must be renamed to: **<ProjectName>.dir**, where **<ProjectName>** is a unique name given to this set of simulations. The name will be used to identify a number of additional files required for the simulation.

If more than one segment other than the basin outlet has the “SAVE” indicator in the 7<sup>th</sup> column in file, **stream.network.dat**, REMOVE the “SAVE” indicators for these segments before using the network file in this program!! Also, make sure that ONLY the outlet segment has a

value of -1 in the sixth column. If more than one segment has the value of -1 in the 6<sup>th</sup> column, reduce the minimum contributing area and rerun the "createstreamnetwork" script to reduce the number of outlets in the stream network created.

\*\*\*\*\*

### Create RBM Parameter Files

Create a working directory, "**Work\_Space**", and copy the topology file, **<ProjectName>.dir**, to this directory.

#### I. Create the RBM river network file

From the directory, "**Scripts**", copy the file, **build\_DHSVM\_network.pl**, to the working directory, "**Work\_Space**", and run this script as shown below:

1) Execute the Perl script

```
perl build_DHSVM_network.pl
```

2) Enter the **<ProjectName>**. In this example, we use **Mercer** as the project name.

```
Mercer
```

Executing this Perl script creates the network text file, **<Project>.net** and **<Project>.segmap**, required by the stream temperature model, **RBM**.

The file, **<Project>.net**, has the following structure:

Group 1: **<Title>**

Group 2: **<Forcing File>**

Group 3: **<alpha> <beta> <gamma> <mu>** (not used)

Group 4: **<D\_a> <D\_b> <D\_min>** (not used)

Group 5: **<U\_a> <U\_b> <U\_min>** (not used)

Group 6: **<nreach>** Headwater **<nseg>** stream segments in Project **<Project>**

Headwaters Data:

```
#_Segments <#segments> Headwaters <temp> TribCell <next_id_RBM>  
<trib_id_DHSVM> <next_id_DHSVM>
```

Segment Data:

If **#segments = 2**, there should be 3 lines in this section.

```
Seq <seq_id> Path <trib_id_DHSVM> X_0 <seg_x0> X_1 <seg_x1>  
Elevation <elevation in ft>
```

```
Seq <seq_id> Path <head_id_DHSVM> X_0 <seg_x0> X_1 <seg_x1>  
Elevation <elevation in ft>
```

The file, **<Project>.segmap** contains a mapping from the sequence numbers used by the **RBM** to the segment ID created by **DHSVM** in stream.network.data. This file has the following format:

Group 1: **<nreach> <no\_segments>** (where **<nreach>** - number of reaches, **<#segments>** - number of stream segments)  
 Group 2: Sequence **<RBM\_sequence>** Path **<DHSVM\_Sequence>** (where **<RBM\_sequence>** - segment sequence number for RBM, **<DHSVM\_Sequence>** - segment sequence number for DHSVM. There are **<#segments>** lines for Group 2 data.)

## II. Make Mohseni parameter file **<Project>.Mohseni**

The file format is:

Line 1: **<θ>** smoothing parameter <headwater is closer to air temperature with a large θ>

From line 2 after: **<reach #> <α><β><γ><μ>**

$$T_{smooth} = (1 - \theta)T_{smooth} + \theta T_{air}$$

$$T_{head} = \mu + \frac{\alpha - \mu}{1 + e^{\gamma(\beta - T_{smooth})}}$$

where  $T_{smooth}$  is smoothed stream temperature by air temperature. The coefficient  $\alpha$  is the estimated **maximum stream temperature**,  $\mu$  is **minimum stream temperature**,  $\beta$  is a measure of the steepest slope of the function, and  $\gamma$  represents the air temperature at the inflection point. See Mohseni et al. (1998) for details of this equation.

## III. Make Leopold parameter file **<Project>.Leopold**

The file format is:

**<reach #> <ua><ub><umin>**  
**<da><db><dmin>**

The first 3 numbers are two Leopold coefficients ( $u_a$ ,  $u_b$ ) for speed ( $U$  in  $ft^3/s$ ) and threshold speed that sets the minimum velocity ( $u_{min}$ ) as used in the equation below.

The latter 3 numbers are two Leopold coefficients ( $d_a$ ,  $d_b$ ) for water depth ( $D$  in  $ft$ ) and threshold depth that sets the minimum depth as used in the equation below.

$$U = u_a Q^{u_b}$$

$$D = d_a Q^{d_b}$$

See Yearsley (2012) for description of parameters

## IV. Create RBM forcing file **<Project>.forcing**

In the directory, **Create/**, create the executable, **Create\_File**, by executing the make file. Then copy the executable, **Create\_File**, to the working directory, **RBM/Work\_Space**. Then execute **Create\_File**:

```
./CreateFile <Input Files Directory> <ProjectName>
```

where **<Input Files Directory>** is the directory with the **\*.Only** files, **<ProjectName>** is the project name.

### **Run RBM**

I. In the directory, **"RBM"**, create the executable, **RBM**, by typing:

```
Make
```

Then copy the resulting executable, **RBM**, to the working directory **RBM/Work\_Space**.

In the working directory **RBM/Work\_Space**, type

```
./RBM <ProjectName>
```

The simulation results will be in the file, **<ProjectName>.temp**. There is output for each segment for every simulation period.

The file **"<ProjectName>.temp"** has the following structure:

```
<time> <year> <day> <segment sequence#> <in-reach sequence#> <simulated  
temperature> <headwaters temperature> <air temperature> <depth> <flow  
velocity> <incoming flow>
```

### **Process the Model Results**

In **Scripts/**, Compile the script, **Extract.Segment.Temp.scr**, to reformat the stream temperature time series and compute daily average stream temperature for the selected segments.

Before running the script, change the hard coded parameters in this script including:

- Selected segment # **<segID>**
- Path to the input file **<ProjectName>.temp**
- Output directory
- Time step of the input file, **<ProjectName>.temp**
- Start date as in all input forcing files **\*.Only**.

```
*****  
NOTE: The segment sequence # is the one used by RBM NOT by DHSVM as in the stream  
network file. If the end date doesn't have a full record from MM/DD/YYYY-00:00 to  
MM/DD/YYYY-21:00, the averaging script will exclude the last day in the computations.  
*****
```

In the directory, **"Scripts"**, execute the program:

./Extract.Segment.Temp.scr

The output files are stored in the designated directory.

The file, <segID>.temp.txt, has the following structure:

<mm/dd/yyyy-hh:mm> <simulated temperature> <headwaters temperature> <air temperature> <depth>

The file, <segID>.daily.temp.txt, has the following structure:

<mm/dd/yyyy> <averaged daily stream temperature>

\*\*\*\*\*

**Please note: if you decide to use this model, please cite the following model development publications:**

\*\*\*\*\*

### **Model Development Publications**

Sun, N., Yearsley, J., Voisin, N., & Lettenmaier, D. P. (2015). A spatially distributed model for the assessment of land use impacts on stream temperature in small urban watersheds. *Hydrological Processes*, 29(10), 2331–2345. <https://doi.org/10.1002/hyp.10363>

Wigmosta, M. S., Vail, L. W., & Lettenmaier, D. P. (1994). A distributed hydrology-vegetation model for complex terrain. *Water Resources Research*, 30(6), 1665–1679.

Yearsley, J. (2012). A grid-based approach for simulating stream temperature. *Water Resources Research*, 48(3), W03506. <https://doi.org/10.1029/2011WR011515>

Yearsley, J. R. (2009). A semi-Lagrangian water temperature model for advection-dominated river systems. *Water Resources Research*, 45(12). <https://doi.org/10.1029/2008WR007629>

### **Relevant Publications**

Cao, Q., Sun, N., Yearsley, J., Nijssen, B., & Lettenmaier, D. P. (2016). Climate and land cover effects on the temperature of Puget Sound streams. *Hydrological Processes*, 2304(March), 2286–2304. <https://doi.org/10.1002/hyp.10784>

Leopold, L. B., & Maddock, T. (1953). *The hydraulic geometry of channels and some physiographic implications*. Washington, D.C.

Mohseni, O., Stefan, H. G., & Erickson, T. R. (1998). A nonlinear regression model for weekly stream temperatures. *Water Resources Research*, 34(10), 2685–2692. <https://doi.org/10.1029/98WR01877>