# A Brief Tutorial for Running DHSVM-RBM

## Download and Uncompress the Files

Unpack the compressed file **DHSVM_RBM.tar.gz**.

```
tar -xzvf DHSVM_RBM.tar.gz
```

After unpacking the file, the source code and supporting files will be in **DHSVM_RBM**:

The sub-directories within this directory are:

| | |
|---|---|
| **../Create** | - contains the Fortran 90 source and **makefile** that build the forcing file for **RBM** |
| **../Output** | - contains the sample output for the examples provided here |
| **../DHSVM3.1.2** | - contains the source code and **makefile** for the hydrologic model, **DHSVM**, which generates the hydrologic data and meteorological data requires by the stream temperature model, **RBM** |
| **../RBM** | - contains the source code **makefile** for the stream temperature model, **RBM** |
| **../Scripts** | - contains the pre- and post-processing scripts |
| **tutorial.pdf** | - contains the tutorial |

**NOTE: In what follows, there are nreach sub-basins. Each sub-basin has a headwaters and is divided into segments where each segment has a unique number. One set of segment numbers is assigned by executing the hydrologic model, DHSVM. A second set of numbers is created for purposes of running the stream temperature model, RBM. The process for creating the second set also creates a file, *.segmap, that maps the second set of segment numbers onto the first for purposes of plotting.**

## Run the Model

The stream temperature model, **RBM**, simulates water temperatures using forcing data and topology created by the distributed hydrologic model, **DHSVM3.1.2**. The files include:

Forcing Files

The hydrologic and meteorological forcing files are generated from the **DHSVM3.1.2** model in this directory, "***../DHSVM3.1.2***". To run **DHSVM3.1.2**, follow these steps:

I. From the directory, "**`../DHSVM3.1.2`**", modify the configuration file, **INPUT.Mercer.3.1.2**. In 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`, enter the parameter values that characterize the riparian vegetation in the section [**CONSTANS**]. The parameters include: *Tree Height* (m), *Vegetation Buffer Width* (m), *Overhang Coefficient* that is a percentage of tree height used to represent overhanging canopy (0 ~ 1), *Extinction Coefficient* (0 ~ 1), and *Canopy Bank Distance* indicating the distance from bank to canopy (m).

III. Run the model following the tutorial for **`DHSVM3.1.1`**, which can be downloaded at http://www.hydro.washington.edu/Lettenmaier/Models/DHSVM/index.shtml. The output forcing files will be in the directory "**`../DHSVM3.1.2/output`**".

Outputs include:

**`ATP.Only`** – air temperature, ℃, for each computational interval and each stream segment.
**`NSW.Only`** – net shortwave radiation, *Watts/m$^2$*, for each time step and each stream segment.
**`NLW.Only`** – net longwave radiation, *Watts/m$^2$*, for each time step and each stream segment.
**`VP.Only`** – vapor pressure, *pascals*, for each time step and each stream segment.
**`WND.Only`** – wind speed, m/s, for each times step and each stream segment.
**`Inflow.Only`** – inflow, m$^3$/second, for each computational interval and each stream segment.
**`Outflow.Only`** – outflow, m$^3$/second, for each time step and each stream segment.

The format for each of these files 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, `INPUT.Mercer.3.1.2`, and so is the start date of output stream temperature simulations.**

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 **DHSVM3.1.2**, and are stored in this directory, "***../DHSVM3.1.2/input***". **<outdir>** specifies the output directory, **<no_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 ../DHSVM3.1.2/input/stream.map
../DHSVM3.1.2/input/stream.network ./  80  9
```

The format for the topology file is as follows:

```
&lt;segment id&gt; <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.**

**NOTE: If more than one segment other than the basin outlet has the "SAVE" indicator in the 7th column in file, `stream.network`, 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 6th column, reduce the minimum contributing area and rerun the "createstreamnetwork" script to reduce the number of outlets in the stream network created.**

After creating the convergence file, follow these steps:

I. Create a working directory, "**`../Work_Space`**", and copy the topology file, <**`ProjectName`**>**`.dir`**, to this directory.

II. 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`**

3) Enter the **`<smooth>`** parameter for smoothing the air temperatures. In this example, 0.1:

   **`0.1`**

4) Enter the Mohseni nonlinear regression parameters of smoothed air temperature on initial headwaters temperatures:

   **`17.0, 16.0, 0.3, 0.1`**

5) Enter the Leopold coefficients for stream speed in $ft^3/s$:

   **`0.9, 0.21, 0.5`**

6) Enter the Leopold coefficients for stream depth in *ft*:

   **`0.2, 0.4, 0.5`**

The work flow is also shown in Figure 1.

```
[[illegible] Work_Space]$ perl build_DHSVM_network.pl
Input ProjectName for topology file: <ProjectName>.dir
This script will build a network file: <ProjectName>.net
Mercer
Input parameters for initial (headwaters) temperatures
and hydraulic parameters, depth and stream speed
Input values are separated by commas

Input parameter, <smooth>, for smoothing daily air temperatures
0.1
Input Mohseni nonlinear regregression parameters
<alpha>,<beta>,<gamma>,<mu>
17, 16,0.3,0.1
Leopold coefficients,<U_a>, <U_B> + u_min for stream speed, u
where u = <U_a>*Q**<U_b> and u_min is a threshold speed (English units
0.9 0.21 0.5
Leopold coefficients,<D_a>, <D_b> + d_min for stream depth, D
where D = <D_a>*Q**<D_b> and D_min is a threshold depth (English units)
0.2, 0.4, 0.5
```

Figure 1. Screen shot when the Perl script that generates the network file is executed.

III. Executing this Perl script creates the network text file, **`<Project>.net`** (**`Mercer.net`**, in this example), 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>** (Parameters for nonlinear regression)
(See Mohseni et al, 1998, for description of parameters)
Group 4: **<D_a> <D_b> <D_min>** (Leopold coefficients for depth and threshold depth) (See Leopold and Maddock [1953] or Yearsley [2012] for description of parameters)

Group 5: **<U_a> <U_b> <U_min>** (Leopold coefficients for speed and threshold speed)
(See Leopold and Maddock [1953] or Yearsley [2012] for description of parameters)

Group 6: **<nreach>** (Number of reaches: see **NOTE** at the beginning for definition)
Group 7: (There are **nreach** groups describing the topology of each sub-basin)
Headwaters Data:
#_Segments **<no_segments**> Headwaters **<head_no>** TribCell **<trib_cell>**
Segment Data:
Seq **<seq_no>** Path **<dhsvm_no>** X_0 **<seg_x0>** X_1 **<seg_x1>**
Elevation **<seg_elevation>**
(For each headwaters, there is a Segment line for the total number of stream segments, **no_segments**, in that reach.

The file, **<Project>.segmap** (**Mercer.segmap**, in the example) is also created. This file contains a mapping from the sequence numbers used by the stream temperature model, **RBM**, to the segment numbers created by the hydrologic model, **DHSVM**. This file has the following format:

Group 1: **<nreach> <no_segments>** (where **<nreach>** - number of reaches, **<no_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 **<no_segments>** lines for Group 2 data.)

IV. In the directory, "***../Create***", create the executable, **Create_File**, by executing the make file:

```
make
```

Copy the executable, **Create_File**, to the working directory, "***../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.

In the example:

```
./CreateFile ../DHSVM3.1.2/output Mercer
```

This will create the file with hydrologic and meteorological forcings **<ProjectName>.forcing**. The file in the example will have the name **Mercer.forcing**.

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

```
make
```

Then copy the resulting executable, **RBM**, to the working directory. If **RBM** is not executable, modify its access permission:

```
chmod 755 RBM
```

VI. In the working directory, 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> <day> <segment sequence#> <in-reach sequence#> <simulated temperature> <headwaters temperature> <air temperature> <depth>**

## Process the Model Results

Compile the script, **Extract.Segment.Temp.scr**, to reformat the 3-hourly stream temperature output and compute daily average stream temperature for the selected segments.

Before running the script, change the hard coded parameters in this script including *the selected segment #*, *path to the input file* **<ProjectName>.temp**, *output directory*, *the time step of the input file*, **<ProjectName>.temp**, and *the 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:

```
chmod 755 Extract.Segment.Temp.scr
./Extract.Segment.Temp.scr
```

The output files are stored in the designated directory, in this example, "***../Output***".

The file, **seg??.temp.txt**, has the following structure:

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

The file, **seg??.daily.temp.txt**, has the following structure:

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

## References

Leopold, L.B and T. Maddock, Jr. (1953). The hvdraulic geometrv of stream channels and some physiographic irn plica tions, Geological Survey Professional Paper 252, United States Government Printing Office,eological Survey, 64 pp.

Mohseni, O ., H. G. Stefan,and T. R. Erickson (1998). A nonlinear regression model for weekly stream temperatures, *Water Resour. Res*., *34*(10), 2685-2693.

Yearsley, J. R. (2012). A grid-based approach for simulating stream temperature, *Water Resour. Res.*, *48*(W03506).