A Brief Tutorial for running DHSVM-RBM

The **DHSVM-RBM** modeling system couples a spatially distributed land surface hydrologic model, **DHSVM**, (Wigmosta et al., 1994) with the distributed stream temperature model, **RBM** (Sun et al., 2015, Yearsley, 2009; Yearsley, 2012).

****

Note: Please cite the above publications if using this modeling system.

****

Cloning Files from GitHub

Assuming that **git** is installed, clone the repository, **DHSVM-RBM:**

git clone https://github.com/jyearsley/DHSVM-RBM.git

Note: This version of **DHSVM** is created with a modified version of **makefile_for_binary**. In addition, some of the scripting for preparing **RBM** files differs slightly from the version maintained by the Pacific Northwest National Laboratory at **https://github.com/pnnl/DHSVM-PNNL.**

****

File Structure

The source code for **DHSVM** version 3.1.3 and **RBM** and supporting files will be in the newly created (local) working directory, **DHSVM-RBM**, with the following structure:

DHSVM3.1.3/src	-source code and **Makefile** for **DHSVM3.1.3**.
DHSVM3.1.3/White	-archived input files for simulating hydrology and energy budget for the White River basin in Vermont.
Scripts	-various scripts used to build the network file for simulating stream temperature with **RBM**.
Create	-Fortran 90 source code and **makefile** to build forcing files from **DHSVM** output.
Test_Problem	- sample workspace for simulating water temperature in the sample basin (White River in Vermont) with **RBM**,

Tutorial DHSVM v3.1.pdf
Tutorial DHSVM-RBM.pdf

Executing the Model Chain

The stream temperature model, **RBM**, simulates water temperature using forcing data and topology created by the distributed hydrologic model, **DHSVM.** To obtain simulated stream temperatures, execute the following steps based on the test data set from the White River in Vermont:

Running DHSVM

→**I.** Prepare the required input files for **DHSVM** as described in the **Tutorial DHSVM v3.1.pdf**. The sample files for the White River are in the directory, **DHSVM3.1.3/White**. Sample input files, including **input.tar.gz**, **metfiles.tar.gz** and **modelstate.tar.gz** for **DHSVM.** The files can be unpacked with the **Unix tar** utility:

```
>>tar –xf <file name>
```

→**II.** In the folder, **DHSVM3.1.3/src**, create the executable, **DHSVM3.1.3:**

```
>>make
```

Note: This makefile creates the version of **DHSVM** that uses/creates binary files. Functions that require **Xwindows** have been removed.

→**III.** Create a configuration file similar to **DHSVM3.1.3/White/INPUT.White.Only.** For guidance on estimating parameters in sections **[SOILS]** and **[VEGETATION]** see Cuo et al (2011).

→**IV.** In the configuration file, (see **DHSVM3.1.3/White/INPUT.White.On**ly) enable stream temperature modeling in the section **[OPTIONS]:**

Stream Temperature = TRUE

If riparian shading is being considered, enable this feature in the section **[OPTIONS]:**

Canopy Shading = TRUE

When setting **Canopy Shading = TRUE**, it is necessary to create a riparian network file similar to **DHSVM3.1.3/White/input/rveg.baseline.network** and enter the path to this network file in the **DHSVM** configuration file (see **DHSVM3.1.3/White/INPUT.White.On**ly) in the section **[ROUTING]:**

Riparian Veg File = ../input/rveg.baseline.network

Type equation here.

In the riparian network file (e.g.,**rveg.baseline.network)**, the format for parameter values,on each line beginning in Column 1 is:

- Stream Segment ID
- Tree Height (m)
- Vegetation Buffer Width (m)
- Monthly Extinction Coefficient 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).

The Monthly Extinction Coefficient represents the parameter, k, in the attenuation, I, of solar radiation, I_o, through a canopy of riparian vegetation of height, z:

$$I = I_0 e^{-kz}$$

In previous studies using DHSVM, the parameter, k, was estimated as ***LAI/64*** where ***LAI*** is the leaf area index for the riparian vegetation type. See Aubin et al (2000) and Smith (1992) for estimates of the parameter, *k*.

→**V.** Copy the executable, **DHSVM3.1.3** and the configuration file (**/DHSVM3.1.3/White/configfiles/INPUT.White.Only** in the example) to the directory that contains the various input files (**/DHSVM3.1.3/White**, in the example) and execute, as in the example problem:

```
>>DHSVM3.1.3 INPUT.White.Only
```

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

Outputs of RBM forcing files include:

••*Each file contains the time series for each Stream Segment.* ***Note:*** *The Segment ID is consistent with the ID used in the* **stream.network.dat** *file.* ••

ATP.Only	air temperature, °*C*
NSW.Only	net shortwave radiation, W/m^2
NLW.Only	net longwave radiation, W/m^2
VP.Only	vapor pressure, *Pascals*
WND.Only	wind speed, m/s
Inflow.Only	stream inflow, m^3/s
Outflow.Only	segment outflow, m^3/s

The format of each of the *.Only files is as follows:

<MM/DD/YYYY-HH:MM:SS (start date)> <MM/DD/YYYY-HH:MM:SS (end date)>

<seg#(n), n=1, number of segments>

<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 and are output for each time step.

NOTE: The start date in the 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.

→**VI.** In the directory, **Scripts**, compile the program, **make_stream_connectivity.c,** to create the stream topology file, **<Project>.dir** where **<Project>** is the name assigned to the project. In the example, the project name is **White:**

```
>>gcc make_stream_connectivity.c –o make_stream_connectivity
```

Copy **DHSVM3.1.3/<Project>/input/stream.map.dat**, **DHSVM3.1.3/<Project>/input/stream.network.dat**, and the executable, **make_stream_connectivity**, to the working directory (**Test_Problem** is the working directory in the sample problem). Execute the program:

```
>>make_stream_connectivity  <#seg>  <Project>
```

The **<#seg>** is the total number of segments and **<Project>** specifies the project name (**White** in the example). The resulting output file, **<Project>.dir**, has the following format for each of the **<#seg>** lines:

<RBM seg#><DHSVM seg#> <seg length><seg height> <seg azimuth> <tribs>

****

<u>Note:</u> If more than one segment other than the basin outlet has the "SAVE" indicator in the 7th 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 -1 in the 6th column, reduce the minimum contributing area and rerun the **create_stream_network** script such that the above criterion is met.

****

→**VII.** Copy the topology file, **<Project>.dir**, created previously, and the script, **Scripts/build_DHSVM_network.pl** to the working directory (**Test_Problem** in the example) and run the script:

```
>>perl build_DHSVM_network.pl
NOTE:  This is an updated version of build_DHSVM_network
        that builds the network file for the version of RBM
        that incorporates variable Mohseni  and Leopold parameters.

       The number of reach cells, ndelta, is set at 2. Specific changes,
       where necessary, must be made in the *.net file.

       Input Project Name for topology file: <Project>
       This script will build a network file:  <Project>.net and a <Project>.segmap file.
```

At the prompt, enter the project name, as in the example:

```
>>White
```

Executing the Perl script creates the network text file, **<Project>.net** and **<Project>.segmap**, required by the stream temperature model, **RBM**. The resulting river basin network is comprised of **<#seg>** stream segments and each stream segment is divided into **<ndelta>** cells for computational purposes. **Note:** In what follows, **<#seg>** refers to the total number of stream segments and **<seg#>** refers to a specific segment. Similarly, **<#nreach>** refers to the total number of stream reaches and **<reach#>** refers to an individual reach.

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

Group 1: **<Title>**
Group 2: **<Forcing File>**

Group 3: **<#nreach>** Headwater **<#seg>** stream segments in **<Project**

There are **<#nreach>** of the following two Groups:
Group 4: #_Segments **<#seg>** Headwaters **<dmmy>** Tribcell **<dmmy><dmmy><dmmy>**
(**<dmmy>** refers to segment numbers not used by **RBM**)
Group 5: Seq **<seq_id>** Path **<DHSVM_id>** X_0 **<seg_x0>** X_1 **<seg_x1>**
Elevation **<elev>** **<#ncells>**

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

Group 1: **<#nreach>** **<#seg>**

There are **<#seg>** of the following Group:
Group 2: Sequence **<RBM seg#>** Path **<DHSVM seg#>**

→**VIII.** Create the file, **<Project>.Mohseni**, that has the parameters for estimating headwaters temperatures from air temperature based on the method described by Mohseni et al. (1998) :

$$T_{smooth}(reach\#) = (1 - \theta)^{*}T_{smooth}(reach\#) + \theta^{*}T_{air}(reach\#)$$

where,

$T_{smooth}(reach\#)$ = the smoothed air temperature in the headwaters (most upstream segment of the reach#th branch, °C,

$T_{air}(reach\#)$ = the air temperature in the headwaters (most upstream segment of the reach#th branch, °C,

θ = a smoothing parameter

The smoothed value of the air temperature, T_{smooth}(reach#), is used to estimate the headwaters temperature, T_{head}(reach#) with the following equation:

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

where,

a = the estimated maximum stream temperature, °C,

b = measure of the steepest slope of the function,

g = the air temperature at the inflection point, °C.

→**IX.** Create the file, **<Project>.Leopold**, containing the variable parameters for estimating stream speed and depth in the directory, **./Test_Problem**, or similar working (local) directory, using the relationships described in Leopold and Maddock (1953) with the following format:

There are **<#seg>** of the following two Groups:

Group 1: **<seg#> <u_a> <u_b> <u_{min}>**
Group 2: **<d_a> <d_b> <d_{min}>**

These parameters are used in the following equations to estimate stream speed, U(ft/sec) and stream depth, D(ft):

$$U = u_a Q^{u_b}, U > u_{min}$$
$$D = d_a Q^{d_b}, D > d_{min}$$

The parameters of Group 1 and Group 2, above, can be estimated from data collected by the US Geological Survey at gaging stations.

→**X.** Create the **RBM** forcing file, **<Project>.forcing** by first navigating to the directory, **./Create**, and executing the **make** file to create the executable, **Create_File**:

→**X.** Create the **RBM** forcing file, **<Project>.forcing** by first navigating to the directory, **./Create**, and executing the **make** file to create the executable, **Create_File**:

```
>>make
```

Copy the executable, **Create_File**, to the working directory,(**./Test_Problem** in the example) and execute it:

./CreateFile **<Input Files Directory> <Project>**

where **<Input Files Directory>** is the directory with the ***.only** files, **<Project>** is the project name.

→**XI.** Run **RBM** by first navigating to the directory, **./RBM** and creating the executable, **RBM:**

```
>>make
```

Copy the executable, **RBM**, to the working directory,(**./Test_Problem** in the example). The working directory should now have all the following files required for performing stream temperature simulations:

RBM	-one-dimensional, time-dependent stream temperature model
<Project>.net	-stream network file
<Project>.forcing	-hydrologic and meteorologic forcings
<Project>.Leopold	-Leopold parameters for each stream segment
<Project>.Mohseni	-Mohseni parameters of each stream headwaters

execute it:

./RBM **<Project> <Result>**

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

Each line of the file, **<Result>.temp**, has the following structure:

<time> <year> <day><seg#> <ndelta> <simulated temperature> <headwaters temperature> <air temperature> <depth> <speed> <flow>

References

Aubin, I., M. Beaudet, and C. Messier (2000). Light extinction coefficients specific to the understory vegetation of the southern boreal forest, Quebec, *Can J. For. Res.*,30: 168-177.

Ca, Q., N., Sun, J. Yearsley, B. Nijssen, and D.P. Lettenmaier (2016). Climate and land cover effects on the temperature of Puget Sound streams, *Hydrol. Process.* 30 (13), 2286–2304, DOI: 10.1002/hyp.10784

Cuo, L., T.W. Giambelluca and A.D. Ziegler (2011). Lumped parameter sensitivity analysis of a distributed hydrological model within tropical and temperate catchments, *Hydrol. Process.* 25, 2405–2421, DOI: 10.1002/hyp.8017

Leopold, L. B., and T. Maddock (1953), The hydraulic geometry of channels and some physiographic implications, U.S. Geol. Surv. Prof. Pap., 252.

Mohseni, O., H.G. Stefan and T.R. Erickson (1998). A nonlinear regression model for weekly stream temperatures, *Water Resour. Res.*, 34(10), 2685-2692. https://doi.org/10.1029/98WR01877.

Smith, N.J. (1992). Estimating leaf area index and light extinction coefficients in stands of Douglas-fir (*Pseudotsugamensiesii*), *Can. J. For. Res.*, 23: 317-321.

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

Wigmosta, M.S., L.W. Vail and D.P. Lettenmaier (1994). A distributed hydrology model for complex terrain, *Water Resour. Res.*, 30(6), 1665-1679.

Yearsley, J. R. (2009). A semi-Lagrangian water temperature model for advection-dominated river systems, *Water Resour. Res.*, 45, W12405, doi:10.1029/2008WR007629.

Yearsley, J.R. (2012). A grid-based approach for simulating stream temperature, *Water Resour. Res.*, 48(3), W03506, doi:10.1029/2011WR011515.