A Brief Tutorial for Running the VIC-RBM hydrologic and stream temperature model

These are the instructions for running the integrated modeling system comprised of the large-scale hydrologic model, VIC (Liang et al, 1994), a routing model based on the work of Lohmann et al (1996) and the semi-Lagrangian water temperature model, RBM (Yearsley; 2009,2012). Model development and implementation for the large-scale hydrologic model, VIC are described in detail on the Web site of the University of Washington Department of Civil and Environmental Engineering's UW Hydro|Computational Hydrology group:

(https://uw-hydro.github.io/code/)

Implementation of the routing model follows, for the most part, the description on the VIC Web site. However, there are some changes, as described below, that are required for the water temperature model.

Most of the requirements for data development can be satisfied by using the same data as that required for input to **VIC** and the routing model. Additional data requirements, as the model is presently configured, include determining the parameters for estimating headwater temperatures based on the method of Mohseni et al (1998) and the coefficients for estimating stream hydraulics based on the method of Leopold and Maddock (1953).

1 Overview

The **VIC-RBM** model is a coupled model that links three sub-models together: the **VIC** model; the routing model; and the **RBM** model. The **VIC** model is a hydrologic model that uses meteorological forcing data as input and simulates hydrologic variables such as runoff, evapotranspiration, and soil moisture, at each grid cell. The routing model takes the output from the **VIC** model along with flow network information as input, and output streamflow at specified locations along stream network. The **RBM** model takes both the streamflow results and the meteorological data as input and calculate stream temperature along stream network. Thus, the **VIC-RBM** model as a whole is able to simulate both streamflow and stream temperature at spatial scales determined by the basic **VIC** gridded network configuration. **VIC** gridded networks have been developed for the Continental United States (CONUS) at 1/16, 1/8, 1/4, 1/2 and 1 degree of latitude and longitude. As presently configured, the **VIC-RBM** model system simulates daily-averaged stream temperatures, only.

2 Model input

1) Input for the VIC hydrologic model

- Meteorological forcing data at each grid cell
  - Minimum requirement:
    daily precipitation
    daily maximum and minimum temperature
    wind speed
  - VIC implements algorithms which are able to generate a full set of meteorological variables (see model output section for a list of meteorological variables) at daily or sub-daily time steps. These variables can also be directly given as input.
- Soil properties at each grid cell
- Vegetation information at each grid cell

2) Input for the routing model

- Flow direction file – describes the topology of the river basin network.
- Unit hydrograph file - contains the grid cell impulse response function.
-

3) Input for the RBM stream temperature model

- Stream temperature as a function of time at headwaters is estimated using Mohseni method [Mohseni et al., 1998], a nonlinear regression of stream temperature on air temperature.
- Stream channel geometry charaacteristics are required to calculate flow depth and velocity using the method of Leopold and Maddock (1953).

3 Model output

Possible model output from the VIC-RBM model includes:

- Grid-cell-based meteorological data at each grid cell at daily or subdaily time step (calculated by the VIC model), including:
  - Precipitation;
  - Air temperature;
  - Wind speed;
  - Atmospheric pressure and density;
  - Vapor pressure (or vapor pressure deficit or relative humidity or specific humidity);
  - Incoming shortwave (solar) radiation;
  - Incoming longwave (or thermal) radiation;
- Grid-cell-based hydrologic data at each grid cell at daily or subdaily time step (calculated by the VIC model), such as:
  - runoff;
  - Snow cover;
  - Soil moisture;
  - Evapotranspiration;
  - ... ...
- Grid-cell-based energy data at each grid cell at daily or subdaily time step (calculated by the VIC model), such as:
  - Net downward shortwave radiaton;
  - Net downward longwave radiation;
  - Net upward sensible heat;
  - ... ...
- Routed streamflow at specified stream locations (can be any locations within the stream network, and the resolution is the same as grid cell, i.e., stream locations are specified by indicating the grid cell in which it falls), calculated by the routing model. Daily or subdaily time step.
- Stream temperature at specified stream locations. (calculated by the RBM model, at daily time steps.

Following describes the steps for simulating water temperature with the grid-based semi-Lagrangian model, RBM, assuming one has downloaded version 4.2.d of **VIC** and prepared all the necessary input files.

****

- *

* **NOTE: The temperature of the distributed (base) flows is set at 10.0 ^{o}C at lines 230 & 232** *

- **In Systmm.f90** *

* *

****

Download and Uncompress the Files

Unpack the compressed file "**VIC_RBM.tar.gz**" and at the Unix prompt, type:

tar –xzvf VIC_RBM.tar.gz

After unpacking the file, the source code and supporting files will be in the following directory:

VIC_RBM

The file:

README - a text file describing the contents of each sub-directory.

The sub-directories within this directory are:

../Perl_Scripts	- contains pre- and post-processing scripts written in Perl.
../RBM	- contains the Fortran 90 source and **Makefile** that builds the executable. **RBM** and places it in the directory, **../Test_Case**. Users may wish to change this by editing the makefile.
../rout_DA	- contains the source code and **Makefile** for the modified routing model, **rout_RBM.**
../Test_Case	- contains the input files and executable for running the example problem, using input data for the Salmon River basin at a gridded resolution of 1/2° lat/long.
../Tutorial	- contains this tutorial.
../UH_Test	- contains the sample output from executing the routing model, **rout_RBM**, and the input data for the sample problem.
../VIC_Forcing	- contains the daily precipitation (mm), max/min air temperatures (°C), and wind speed (m/sec) that provides forcings for each **VIC** grid cell.
../VIC_Input	- contains the snowbands, soil, veg-param files and the world_veg_lib files
../VIC_Output	- contains the meteorologic output files, full_data_lat_long, and the hydrologic output - files, flux_lat_long, from the VIC simulations.

Create the RBM Executable

1. Navigate to the folder, ***../RBM***, and type:

make

This will create the executable, **RBM**, and copy a version to the folder, ***../Test_Case***, where the example problem is found.

Run the Model

Forcing Files

1. Build the forcing function files (flow and meteorology).

In the example problem, I used **VIC4.2.d** with the two (2) global parameter files in the directory, **run_VIC**. To create the necessary files, I copied **vicNl,** the executable for **VIC4.2.d** to the folder, **run_VIC**, renamed it as **vicNl_flux** for generating base flow and runoff and **vicNl_full_data** to generage the heat budget forcing. The two were implemented as follows:

./vicNl_flux –g global_param_Salmon_0.5_flux - generates base flow and runoff

and
./vicNl_full_data –g global_param_Salmon_0.5_full_data - generates heat budget

****
• *
* **NOTE: Output files for the heat budget using VIC4.2.d and example global_param file have** *
* **a different format than those created with VIC.4.1.2. The routing program that creates the** *
* **the direct access files for use by RBM assumes the format is the one from VIC4.2.d.** *
• *
****

In the example, the outputs from this process are copied to the folder, ***../VIC_Output***.

The routing program that uses these results requires that the HEADER option in the **VIC** parameter files be set to TRUE. The output format specified in the **VIC** parameter files is also the format used by the routing program.

2. In the folder, ***../Test_Case***, run the perl script, **build_network_beta.pl**, (copied from ***../Perl_Scripts***) using the direction file that was created as described in the VIC model development (see the VIC model page on **https://uw-hydro.github.io/code/**). All the grid cells surrounding the basin of interest must contain a negative one (-1) for purposes of determining the headwaters segments. The number of basins and sub-basins in a river system is limited only by the amount of computer memory, but there must be only a single outlet. For example, the Columbia River system could be modeled in its entirety. Modeling the Columbia River system and another river system, for example, the Fraser River, would require two separate simulations. Also, simulated river system cannot contain braided networks. The following example is from the Salmon River in Idaho:

perl build_network_beta.pl Salmon_flowd_0.5.dir Salmon_0.5.Topology

This script requires two (2) files in the command line, the direction file (**Salmon_flowd_0.5.dir**, in the example problem) and the output file, **Salmon_0.5.Topology** (in the example problem).

3. Prepare a control file (see Figure 1 for an example) that describes:

Starting Date:	
Ending Data:	
Time Steps per Day:	Note: This version of model simulates only daily averages, so this is always "1".
Input directory:	
Output directory:	
Topology File:	
Network File:	Note: The Network File must have the suffix, **_Network**.
Flow File:	Note: This is the name of the file that will be created by the routing program
Heat File:	Note: This is the name of the file that will be created by the routing program
Mohseni Parameters:	Note: When the characters, " grid" (note the space) are missing, constant values for the Mohseni parameters are obtained from a file with the same format as the example file, **Salmon_Parameters**.
Heat Dump:	Note: FALSE if there is no advective heat source (Power Plants, for example)
Ndelta:	Note: Usually "2", but can be larger, particularly in the case of slower streams

The control file must have the suffix, **.Control**, and the colon (:) after the descriptive characters in each line is required.

4. Run the Perl script, **build_input.pl**, using the control file,(**Salmon_0.5.Control**) creating the stream temperature network file (**Salmon_0.5_Network** in this example) and the ordered input file names for the routing scheme. (**Rout.Cells.init** and **Rout.Cells**). See Appendix A for a description of elements in the network file. Here again, the project name (**Salmon_0.5)** is required on the command line. The suffix, **.Control**, is appended by the Perl script:

```
perl build_input.pl Salmon_0.5
```

5. Build the forcing function files for flow and heat budget using **rout_RBM**

Create control file, **salmon.inp_DA**, to run the modified Lohmann routing model using the file, **Rout.Cells.init**, the first time and the file, **Rout.Cells**, if the routing model, **rout_RBM** is run again for the same set of unit hydrographs. The input file (in this example, **salmon.inp_DA**) is similar to that described on the VIC model Web site with the exception of the addition the Leopold parameters. The routing model requires a file with unit hydrographs for each cell, or, in this case, a file, **UH_ALL,** with a single unit hydrograph that is applied to all grid cells.

6. Run **rout_RBM** from the directory, **rout_RBM**, directory using the file, **salmon.inp_DA**

```
./rout_RBM salmon.inp_DA
```

creating the direct access files for flow and heat budget, (**Salmon_DA_flow** and **Salmon_DA_heat** in the example) . See Appendix A for a description of the state variables in the flow and heat budget files.

7. All necessary files have been created at this point (see Appendix A for a description of the output). Simply

run the temperature model with command line files as follows:

./RBM_VIC Salmon_0.5 Salmon_0.5.Temp

where **Salmon_0.5** refers to **Salmon_0.5_Network** ("**_Network**" is appended by the model software) and "**Salmon_0.5.Temp**" is the output file. "**Salmon_0.5.Spat**" is also created and is a file that cross-references element numbers in the output file "**Salmon_0.5.Temp**" to lat/long for post-processing.

References

Liang, X., D.P. Lettenmaier, E.F. Wood, and S.J. Burges, 1994, A simple hydrologically based model of land surface water and energy fluxes for GSMs, *J. Geophys. Res.*, 99(D7), 14,415-14,428.

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

Lohmann, D., R. Nolte-Holuber, and E. Raschke, 1996: A large-scale horizontal routing model to be coupled to land surface parameterization schemes, *Tellus*, 48(A), 708-721.

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, doi:10.1029/98WR01877

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

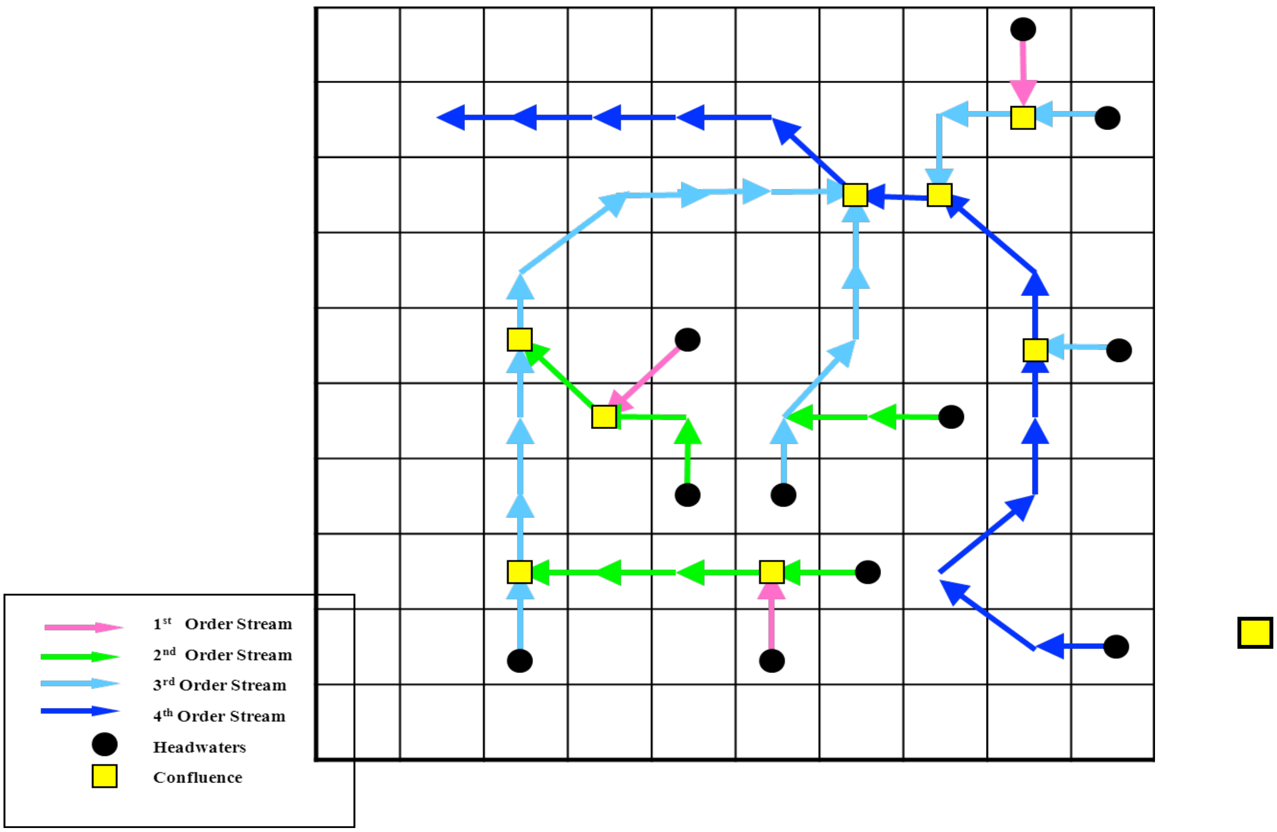

Figure 1. Network Example

```
Test Example for Input to RBM10_VIC modeling system:
Salmon River 0.5 degree grid
#
#Temperature Simulation
Starting Date:              20000101
Ending Data:                20011231
Time Steps per Day:         1
Input directory:            .//
Output directory:           .//
Topology File:              .//Salmon_0.5.Topology
#
#
Network File:               .//Salmon_0.5_Network
#
#
Flow File:                  .//Salmon_DA_flow
#
#
Heat  File:                 .//Salmon_DA_heat
#
#
Mohseni Parameters:         .//Salmon_Mohseni:grid
#
#
Heat Dump:                  FALSE
#
#
Ndelta:                     2
```

Figure 1. Network Example

Figure 2. Sample control file to run RBM for the Salmon River basin at a resolution 0.5° latitude/longitude. (Note:This example makes use of a gridded set of Mohseni parameters)

```
# INPUT FILE FOR PROGRAM CREATING FLOW AND ENERGY BUDGET FILES-RBM
# NAME OF FLOW DIRECTION FILE
./Salmon_flowd_0.5.dir
# Name of velocity file
.false.
2.0
# Name of diffusion coefficient file
.false.
800
# Name of xmask file
.false.
12500
# Name of fraction file
.false.
1.0
# Leopold coefficients
F
0.34 0.341 1.22 0.557
# Name of station file
./Rout.Cells
.true.   If true, read the path for flow and weather data
# Path of input files for flow and energy budget from VIC simulations
../VIC_output/fluxes_
../VIC_output/full_data_
2  1  8  (=2 for 1/2 deg,=3 for 1/4 deg,etc),# flux values/day,# full_data
values per day
# Path of direct access files for flow and energy budget
./Salmon.DA_flow
./Salmon.DA_heat
# Year,month,day (start & end) of VIC flux,VIC full_data,output
2000 01 01 2001 12 31    VIC flux
2000 01 01 2001 12 31    VIC full_data
2000 01 01 2001 12 31    write output
# This file contains the grid cell impulse response function
./UH_ALL
# Name of directory with response function files (*.uh_s)
../UH_Test/
```

Figure_3 Input control file for running the routing model, **rout_RBM**. This will produce the flow and thermal energy forcings for **RBM**, as well as the unit hydrographs for additional simulations.

APPENDIX A: RBM I/O instruction

Notation:

Reach: stream reaches can include several cells
Cell: a cell is a grid cell.
Cell path: each reach is first divided into parts, which we call cell path here, based on cells; for each reach, if it covers "ncell" number of cells, then it has (ncell - 1) number of cell paths
Node: A cell can have one or more nodes. If a cell is not a confluence, then it has one node; if a cell is a confluence cell, then this cell has more than one node, depending on how many tributaries converge at this cell. Cell paths are the channel parts that are between two adjacent nodes.
Segment: a cell path is further divided into multiple segments (default is that each river cell is divided into two segments).
Segment points: the division points of segments. Note that the most upstream segment point would be the upstream edge of the headwater cell.

Example:

Some variables below are explained using the topology of the Salmon example:

Network file (input):

Line 1: not read
Line 2: directory of flow file
Line 3: directory of heat file
Line 4: <start_date> <end_date>

date format is "YYYYMMDD". Extra arguments will not be read.

Line 5: <nreach> <flow_cells> <heat_cells> <source>

<nreach>: # reaches. A reach (Figure 1) is comprised of segments from the headwaters (circles) to the confluence with the next longest reach (squares). In Figure 1, a 1st order reach is the color, magenta, a 2nd order reach is the color, chartreuse, and the longest (main stem) reach is the color, blue. Note that the term, “order”, is not the same as the term used in the Strahler classification of stream order. In the example from from the Salmon River, nreach = 7.

<flow_cells>: # total cell paths. In the example, flow_cells = 19.

<heat_cells>: # nodes. For non-confluence cells, each cell has one node. For confluence cells, one cell has more than one nodes (depending on how many tributaries it has).

<source>: whether there are point source inputs. TRUE for yes, FALSE for no.

(**Line 6:** <souce_file>)

If <source>=TRUE, this line is the path of point source file.

For each reach:

Line 1: <no_cells> Headwaters <head_name> TribCell <trib_cell> Headwaters <main_stem> R.M. = <rmile0>

<no_cells>: # nodes in the reach (at least 2)

<head_name>: the headwater number. Since one headwater corresponds to one reach, headwater number is also the indicator of reach number.

<trib_cell>: the node number in the next higher order reach where the current reach enters into. If the current reach is the highest order reach, then <trib_cell> is equal to the node number of the outlet node.

<main_stem>: the headwater number (thus also the reach number) of the next higher order reach it enters. If the current reach is the highest order reach, then <main_stem> is equal to 0.

<rmile0>: length of the reach + length of one extra grid cell upstream of the headwater cell.

Line 2: <alphaMu> <beta> <gamma> <mu> <smooth_param>

Mohseni parameters for this reach (alphaMu is 'alpha - mu')

For each cell in the reach (# loops = <no_cells>):

Line 1: Node <node> Row <row> Column <col> Lat <lat> Long <lon> R.M. = <rmile1> <ndelta>

<node>: node number of this node in this reach (note that the same cell can have multiple nodes if it is a confluence cell)

<row>: row number of the node

<col>: column number of the node

<lat>: lat of the node

<lon>: lon of the node
<rmile1>: river length (in miles) between this node and the most downstream node of this reach (so if the current node is the most downstream node, its <rmile1> would be 0)
<ndelta>: # segments of this cell in this reach (each cell reach is further diveded into <ndelta> number of segments for calculation); default: 2.

Flow file (input):

For each time step (currently daily):

For each reach:

for each cell in this reach (except the most downstream node):

Line: <nnd> <ncell> <Q_in> <Q_out> <Q_diff> <depth> <width> <u>

<nnd>: index of time step (currently daily, i.e., 1, 2, ..., last day)
<ncell>: node number **(must be in order, and must skip the last node in each reach)**
<Q_in>: inflow discharge of the node [cfs]
<Q_out>: outflow discharge of the node [cfs]
<Q_diff>: lateral flow directly from land surface [cfs] (usually 0 if flow is already routed)
<depth>: flow depth [ft]
<width>: channel width [ft] (not used)
<u>: flow velocity [ft/s]

Note: Changing line is not necessary

Heat file (input):

For each time step (currently daily):

For each reach:

for each cell in this reach (including the last cell also):

Line: <ncell> <dbt> <ea> <Q_ns> <Q_na> <rho> <press> <wind>
<ncell>: node number **(must be in order)**
<dbt>: air temperature [degC]
<ea>: air vapor pressure at surface [mb]
<Q_ns>: net incoming shortwave radiation [Kcal/sec/meter**2] (see modified routing model instruction for details in units)
<Q_na>: net incoming longwave radiation [Kcal/sec/meter**2] (see modified routing model instruction for details in units)
<rho>: air density [kg/m3] (not used)
<press>: air pressure at surface [mb=100 Pa] (not used)
<wind>: wind speed [m/s]

Output file (*.Spat):

For each divided stream segment:

Line: <reach_ind> <cell_ind> <row> <col> <lat> <lon> <delta_ind>
<reach_ind>: index of reach (i.e., which reach it is)
<cell_ind>: index of node (i.e., which node in this reach it is)
<row>: row number of the cell
<col>: column number of the cell
<lat>: lat of the cell
<lon>: lon of the cell
<delta_ind>: index of the divided segment point (i.e., which segment point in this cell and this reach it is). Note that the first segment point in the cell would be the upstream edge of the cell; the last segment point would be the last segment point upstream from the downstream edge of the cell (but not the downstream edge). For example, if using the default segment # =2 for each cell, the first segment point would be the upstream edge of the cell; the second segment point would be the middle point of the cell. (is it correct?)

Output file (*.Temp):

For each node:

Line: <time> <nd> <nr> <ncell> <ns> <T_0> <T_head> <dbt>

<time>: decimal year
<nd>: day of year
<nr>: reach index
<ncell>: node index in this reach
<ns>: segment index in this reach (note that it is index in this reach, but not just in this cell)

<T_0>: stream temperature
<T_head>: headwater temperature (should be the same for a certain reach)
<dbt>: air temperature (should be the same for the nodes which are in the same cell)