HYDROLOGY LABORATORY-RESEARCH DISTRIBUTED HYDROLOGIC
MODEL (HL-RDHM) USER MANUAL V. 2.4.2
(Last Modified 8/15/08)

Table of Contents

1. INTRODUCTION

This User's Manual for the Hydrology Laboratory-Research Distributed Hydrologic Model (HL-RDHM) provides a description of the model and its capabilities and the information required to use the model. A separate HL-RDHM Developer's Manual is also available that describes the software structure in more detail and the common programming framework that HL-RDHM provides to model developers.

To facilitate research into the use of distributed models for hydrologic simulation and forecasting, the National Weather Service (NWS) Office of Hydrologic Development (OHD), Hydrology Laboratory (HL), Hydrologic Science and Modeling Branch (HSMB) developed HL-RDHM. HL-RDHM is a flexible, stand-alone tool for distributed hydrologic modeling research and development. It also serves as a prototype tool for validating techniques in NWS field offices prior to operational software development.

Scientific techniques validated using HL-RDHM are transferred to operational software. An example of this transition is the Distributed Hydrologic Model (DHM) operation recently developed for the National Weather Service River Forecast System (NWSRFS) in AWIPS OB7.2. There is naturally a transition period in moving proven science techniques from HL-RDHM or elsewhere into DHM or other operational software. Therefore, not all key capabilities available in HL-RDHM are available in DHM OB7.2. Specifically, the focus of software development for DHM OB7.2 was to integrate a basic distributed modeling capability with other NWSRFS operations in forecast mode. To set up models for use with DHM OB7.2, HL-RDHM and supporting utilities are necessary to generate routing parameter grids using local data and make calibration mode runs. To satisfy this need, HL-RDHM and the associated utility programs is made available to AWIPS users via the Local Applications Database (LAD).

HL-RDHM supports investigations into several promising areas for service improvement including (1) improved forecast accuracy at river forecast points, (2) improved flash flood information, and (3) new gridded products (e.g. soil moisture, soil temperature, snow, other) for water resources applications.

The features of HL-RDHM have evolved based on lessons learned from numerous scientific studies and practical applications since the mid 1990s. The HL-RDHM hydrologic system structure and some of its basic routing codes originated from the Nile Forecast System (Koren and Barrett, 1995). Several of the scientific subroutines accessible through HL-RDHM are identical to or slight modifications of National Weather Service River Forecast System (NWSRFS) operations. Koren et al., (2004) provide a description of the basic hydrologic system, a gridded implementation of the SAC-SMA model, the overland flow and channel routing algorithms, routing parameter estimation procedures, and representative results (Note that Koren et al., (2004) refer to the Hydrology Laboratory-Research Modeling System (HL-RMS), a precursor to HL-RDHM 2.4.2). The HL-RDHM code base has been used for numerous other scientific studies and prototype operational applications (Cooper, 2006; Koren et al., 2005; Moreda

et al., 2006; Reed, 2004; Reed et al., 2007; Reed et al., 2006; Schmidt et al., 2007; Shultz and Corby, 2006; Smith et al., 2004; Zhang et al., 2006). Cui et al. (2006) report on current software architecture features of HL-RDHM. Many more details on these features are provided in this User's Manual and HL-RDHM Developer's Manual.

As HL-RDHM has evolved, so has crucial work into distributed parameter estimation for various hydrologic modeling techniques supported by HL-RDHM. Based on this work, default a-priori grids for many required parameters and connectivity files required for cell-to-cell routing have been derived for CONUS and are delivered to RFCs with HL-RDHM. Chapter 3 and Appendix A in this manual summarizes the modeling techniques available in HL-RDHM, the required parameters for each, and whether or not default, spatially variable, a-priori parameter grids are provided for each parameter. Chapters 9 and 10 provide additional guidance on how to (1) estimate parameters for which no default values are provided, and (2) make adjustments to default parameter grids provided using manual and automatic calibration procedures.

Although HL-RDHM is the core tool for HL distributed model research and prototyping, there are also a number of related tools that have been developed to support various aspects of the distributed modeling process. This document also serves as the manual for some of these supporting tools, while other supporting tools are sufficiently complex or sufficiently independent of HL-RDHM to merit their own documentation. For example, Chapter 10 on model calibration strategy describes how the independent applications STAT-Q, ICP, and XDMS can be used as part of the model calibration process. Separate documentation is provided for these applications and the NWSRFS DHM operation (see Table 6.1).

2. HYDROLOGIC SYSTEM STRUCTURE

The availability of high resolution remote sensing data such as NEXRAD-based precipitation drives ongoing distributed hydrologic model research and, specifically, has driven the development of HL-RDHM. HL-RDHM uses a gridded model structure to provide an efficient interface to remote sensing-based products and atmospheric model outputs. HL-RDHM is a 'distributed' model in a relative sense. RFC lumped models are distributed relative to large river basins, and HL-RDHM models are distributed relative to RFC lumped models. To date, HL-RDHM applications have used either the Hydrologic Rainfall Analysis Project (HRAP) grid cells or ½ HRAP grid cells; however, the HL-RDHM structure can support any grid cell resolution. HRAP grid cells have a nominal side length of 4-km (2-km for ½ HRAP grid applications).

HL-RDHM can be run in either connected or unconnected mode. Unconnected mode runs are simpler because they exclude cell-to-cell routing. Unconnected mode is currently used for large area water balance and soil moisture studies and a prototype gridded Flash Flood Guidance (FFG) application at the Arkansas-Red Basin River Forecast Center (ABRFC). Connected mode runs include cell-to-cell routing and thus can produce channel state information for all model grid cells. Connected mode runs require a cell-to-cell connectivity file and channel routing parameter estimates as

described in Chapters 3 and 9.

Figure 2.1 describes the generic computational structure for HL-RDHM. The structure provides a flexible research and development tool to work with gridded and time series data (Cui et al., 2006; HL-RDHM Developer's Manual).

HL-RDHM assumes a main time loop and provides a generic template for executing functions "Before", "Inside", and "After" this loop. Model parameters and states are read in the "Before Loop". Model parameter and state grids can be output before, during, or after the time loop. Model time series data may be output during the time loop. The model user may choose from several modeling techniques (e.g. rainfall-runoff, routing) for which the gray box functionality in Figure 2.1 is already defined by the developers. Model developers write codes for the gray boxes when defining a new technique. The currently available techniques and their required inputs are described in Chapter 3.

The user also has flexibility in defining which grids and time series to output for a given model run. An input file (or 'input deck') is used to specify the selection of run period, modeling techniques, and output options. With the techniques available in HL-RDHM 2.4.2, a user could choose to output up to 102 different states in grid or time series format and up to 136 different parameter grids. Chapter 8 provides a complete description of the available run specification options (See also Appendix A for a list of data associated with each technique).

Unique rainfall-runoff and routing parameters may be defined for each model grid cell but this is not required. There are also options within the system that allow spatial averaging of inputs or model parameters over areas larger than a single pixel (e.g. a forecast basin).

Due to the availability of hourly, multi-sensor NEXRAD based precipitation grids from RFCs, most implementations of HL-RDHM to date have used an hourly time step. However, other time steps are allowed by the modeling techniques and can be specified to match the available forcing data. In doing so, however, users should be careful to understand model parameter sensitivity to the model time step.

Although the basic structure of HL-RDHM assumes only one time loop, this is not a strict limitation of the system structure. For example, a modeling technique within HL-RDHM can specify its own internal time loop. Two examples of this are a *HindCast* module that is being developed for flash flood model validation work and the internal calculations of the Sacramento model operation that reduce the computational time step if necessary to maintain the water balance.
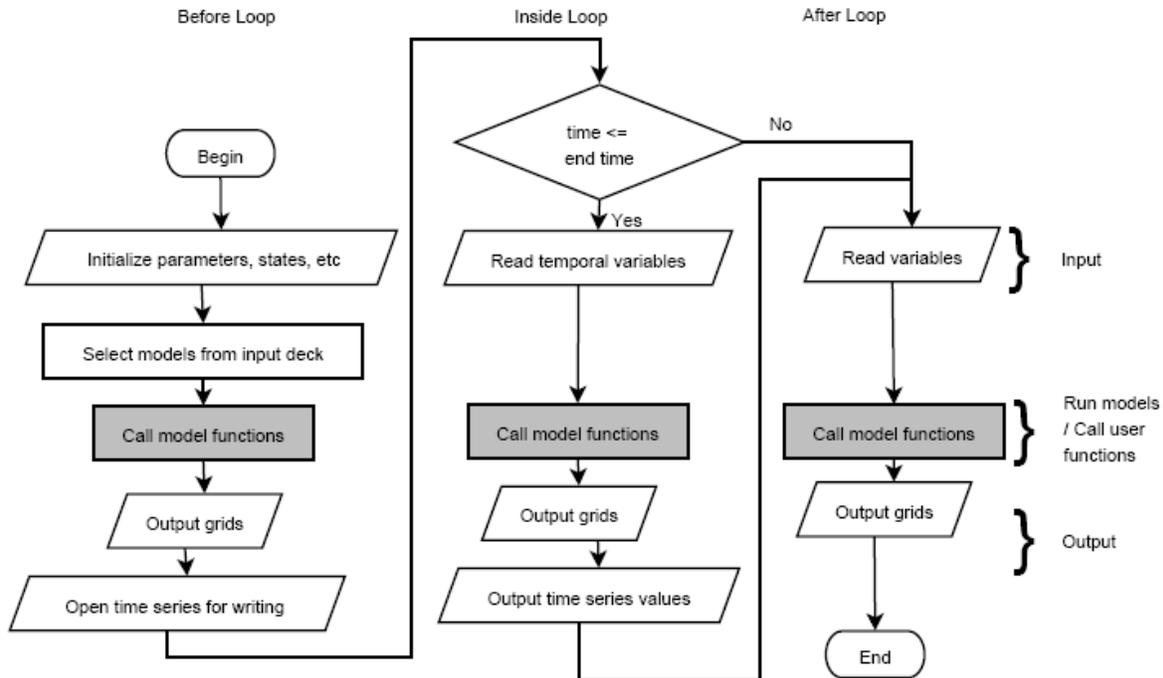
Figure 2.1 HL-RDHM generic computational flow chart

The basic HL-RDHM 2.4.2 without user defined modules runs in simulation mode only, reading historical data. Forecast mode features currently being developed.

3. HYDROLOGIC MODELING TECHNIQUES

3.1. Snow-17 (*snow17*)

3.1.1. Description

The NWS snow accumulation and ablation (Snow-17) developed by Anderson (1973) is available within HL-RDHM as the technique *snow17*. The *snow17* technique is run at each pixel with gridded precipitation inputs from multiple sensor products and gridded temperature inputs. In the distributed application of Snow-17, the areal depletion curve can be replaced by assuming snow or no snow at the pixel level. Another approach to represent the depletion curve is to assume a straight line relationship at a pixel level.

3.1.2. Inputs

3.1.2.1. Forcings

For *snow17*, precipitation and temperature grids are required. HL-RDHM assumes that grid files are provided in the NWS xmrg file format (See Chapter 5). The time step for the model run must be a multiple of the time step for the available input data.

3.1.2.2. Parameters

There are up to 25 Snow-17 parameters. Right now, four a-priori Snow-17 parameter grids are available: MFMIN, MFMAX, ALAT, and ELEV. In our initial application of the distributed Snow-17 model, other parameters are assigned as a single value for each basin in the model. However, for distributed applications the depletion curve can be set to a straight line with SI value greater than zero. Another alternative is to set SI=0. In the latter case the model assumes a snow or no snow for each pixel.

Snow parameters:
- *ALAT*, Altitude (HRAP grid available)
- *SCF*, Snow fall correction factor
- *MFMAX*, Maximum melt factor [mm (6 hr)$^{-1}$ °C^{-1}]
- *MFMIN*, Minimum melt factor [mm (6 hr)$^{-1}$ °C^{-1}]
- *NMF*, maximum negative melt factor [mm (6 hr)$^{-1}$ °C^{-1}]
- *UADJ*, The average wind function during rain-on-snow periods [mm mb^{-1}]
- *SI*, areal water-equivalent above which 100 percent areal snow cover [mm]
- *MBASE*, base temperature for non-rain melt factor [°C]
- *PXTMP*, temperature that separates rain from snow units [°C]
- *PLWHC*, maximum amount of liquid-water held against gravity drainage – decimal fraction [dimensionless]
- *TIPM*, antecedent snow temperature index parameter – range is 0.1 to 1.0
- *PGM*,   daily ground melt [mm day^{-1}]
- *ELEV*, mean elevation (HRAP grid available) [m]
- *LAEC* Snow–rain split Temperature, usually assumed to be zero [°C]
- Depletion Curve (11 values)

3.1.2.3. States

To run *snow17*, values of eight initial model states must be defined (we, neghs, liqw,tindex, accmax, sndpt, sntmp, and sb). The first time the model is run, initial state grids will not be available and constant values can be assigned to selected regions (i.e. basins or rectangular windows). In subsequent runs, states saved from earlier runs can be used as input

- *we*, snow water equivalent (mm)
- *neghs,* Initial heat deficit (mm),
- *liqw,* liquid water equivalent
- *tindex,* temperature index (DEGC)
- *accmax,* maximum water-equivalent that has occurred since snow began (mm)
- *sndpt,* snow depth (cm)
- *sntmp,* snow cover temperature (DEGC)
- *sb*, the last highest snow water equivalent before any snow fall (mm).

3.1.3. Outputs

In many cases, the main output of interest is gridded rain-plus-melt that would often pass to a rainfall-runoff model. Other possible outputs include snow water equivalent and snow depth.

3.2. <u>Sacramento Soil Moisture Accounting, SAC-SMA (*sac*)</u>

3.2.1. Description

The Sacramento Soil Moisture Accounting model with a Heat Transfer component (SAC-SMA-HT) is available within HL-RDHM and specified as the *sac* technique in the input deck. The science of the basic SAC-SMA model is well documented elsewhere (Anderson, 2002; Burnash, 1973; Smith et al., 2003) so it is not reproduced here. Within NWSRFS, the SAC-SMA model has most often been run for relatively large lumped basins (~ 300 – 4000 km^2). HL-RDHM facilitates gridded or lumped SAC-SMA-HT calculations. Gridded calculations can be made over large areas (e.g. CONUS).

A major advance that allows practical implementation of a gridded SAC-SMA model is the algorithm to derive a-priori parameters from soil and landuse data. This algorithm was first described by Koren et al. (2000) and later elaborated on by Koren et al. (2003), Anderson et al. (2006), and Zhang et al. (2006).

Another key advancement due to the addition of the Heat Transfer (HT) component is the ability to convert water depths in SAC-SMA conceptual storage zones to water contents in physical soil layers. This capability is described more fully in the description of the Frozen Ground technique below (*frz*). When used without the *frz* technique, the *sac* technique can still output computed soil moisture values on physical layers.

3.2.2. Inputs

3.2.2.1. Forcings

The *sac* technique requires rain or rain-plus-melt inputs. If run with *snow17*, the *snow17* rain-plus-melt output is automatically available to *sac*. When run by without *snow17*, *sac* reads gridded rainfall data in the NWS xmrg file format. A constant multiplier can be applied to input precipitation or temperature grids for the duration of the simulation if desired.

The *sac* technique also uses potential evaporation data. Implementations of HL-RDHM to date have used climatological monthly potential evaporation (PE) data. Twelve mean monthly PE grids at the HRAP resolution are available. The method for deriving these grids is described in unpublished work by V. Koren, J. Schaake, Q. Duan, M. Smith, and S. Cong (August 13, 1998). In this work, information from seasonal and annual free water surface evaporation maps (PE) in NOAA Technical Report 33 and mean monthly station data from NOAA Technical Report 34 were used to derive an equation that predicts long-term mean daily variability of PE. This equation was used to

derive the monthly PE grids. Summing the monthly values yields a consistent result with the annual and seasonal maps in NOAA Technical Report 33.

HL-RDHM 2.4.2 also had the capability to read daily PE grids.

3.2.2.2. Parameters

The basic SAC-SMA model uses 16 parameters. HL-RDHM can read these parameter values from grids or can read lumped values for specified basins. As described in Chapter 8, there is also an option to scale parameter values in a grid by a constant multiplier over selected regions (e.g. basins). Procedures to use this multiplier capability in the calibration process are discussed in Chapters 4 and 10.

Default a-priori grids for 11 of the SAC-SMA parameters are provided for the conterminous United States with HL-RDHM Version 2.4.2. The data table in Appendix A lists all parameter grids for which a-priori values are provided. Grids delivered with this version were derived using STATSGO soils data (Koren, 2000). Work is underway to derive improved a-priori grids using land use data with STATSGO and also using SSURGO data.

Although a-priori grids are not provided for 5 of the SAC-SMA parameters (sac_PCTIM, sac_ADIMP, sac_RIVA, sac_SIDE, sac_RSERV), constant values for these parameters can be defined in the model input deck for selected regions (e.g. basins or rectangular windows) using local knowledge.

To make potential evaporation demand calculations, the SAC-SMA model uses PE adjustment factors to account for the effects of vegetation. Potential evaporation demand is the product of PE and PE adjustment factors. It is common practice to use mean monthly values for PE adjustment factors, although in theory these adjustment factors can vary within a month. Twelve a-priori PE adjustment factor grids are provided with HL-RDHM version 2.4.2. Unpublished work by V. Koren, J. Schaake, Q. Duan, M. Smith, and S. Cong (August 13, 1998) describes the derivation of these grids. They derived the grids using an empirical function relating calibrated PE adjustment factors to satellite-derived, green vegetation fraction data.

3.2.2.3. States

To run *sac*, values of six initial model states must be defined (uztwc, uzfwc, lztwc, lzfsc, lzfpc, adimpc). The first time the model is run, spatially distributed state grids will not be available but constant values can be assigned to selected regions. In subsequent runs, states saved from earlier runs can be used as input.

3.2.3. Outputs

Traditionally, the principal output from the SAC-SMA technique is the runoff, which is a required input for the available routing techniques (*rutpix7, rutpix9*). With the

gridded SAC-SMA-HT, there are numerous other possible outputs that can be useful for water resources and flash flood applications. Any of the data grids listed in Appendix A can be output at the user-specified time interval. This includes the soil moisture content for different physical layers. Time series of spatially averaged states can also be output.

Note that the current version does not allow the user to output PE adjustment grids ("peadj_"). To save memory for large area runs, the 12 PE and 12 PE adjustment grids are read separately, but then multiplied together internally so only 12 sets of PE demand grids need to be stored in memory. If the user chooses to output "pe" grids, the data written to these files are actually PE demand.

3.3. Continuous API (*api*)

3.3.1. Description

The details of the Continuous API model are described in the NWSRFS User's Manual Section II.3. Moreda et al. (2006) describe a gridded implementation of the Coninuous API model and a technique to develop spatially distributed parameters. A summary of these issues is also provided here.

The CONT-API model uses a graphical technique consisting of four quadrants to compute surface runoff, baseflow, and a few additional features including an option to account for the effect of frozen ground on runoff (Figure 3.1). The four quadrants perform the following functions. The first quadrant accounts for the seasonal relationship between API and current soil-moisture conditions. The two curves in this quadrant represent wet and dry conditions of the watershed. A specific watershed condition is determined in three ways: a fixed week number, antecedent evaporation index, or antecedent temperature index. The second quadrant accounts for surface moisture conditions. The third quadrant computes the incremental surface runoff based on surface and overall soil-moisture conditions. The fourth quadrant computes the portion of the precipitation that does not become surface runoff and enters groundwater storage. Baseflow runoff is computed based on the total water in groundwater storage and the amount that has entered the storage in the recent past. The model also allows for impervious area runoff and riparian vegetation losses and frozen ground components. Table 3.1 describes the CONT-API parameters.
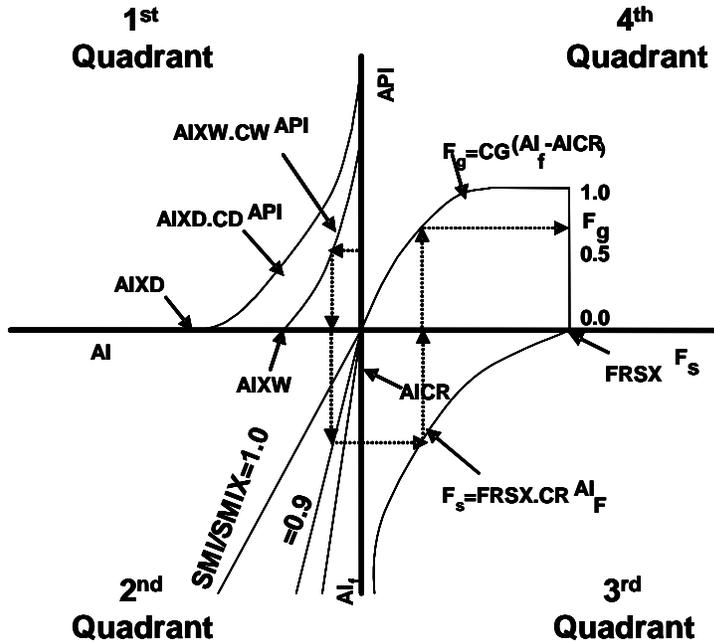
Figure 3.1 API model graphic.

The graphical method of the CONTI-API works as follows: Add the current precipitation amount to the antecedent precipitation, API. With the new API, start from the 1st quadrant and follow the dashed line as in Figure 3.1. Depending on the season (wet or dry), obtain the antecedent index, AI value. Then in the second quadrant obtain the adjusted or final antecedent index, AIF value depending on the soil moisture condition. With the AIF value, in the third quadrant obtain the surface flow component F_s. Finally, extend the Fs value into the fourth quadrant to get the ground water component F_g.

The CONT-API model produces surface and subsurface runoff in each grid cell. These flow components are often passed to the hillslope and channel routing algorithms. The main problem in the application of this model is to derive sensible grid level parameters.

3.3.2. Inputs

3.3.2.1. Forcings

Required forcings are precipitation and PE. In applications to date, the same forcings described for the *sac* application have been tested using HL-RDHM.

3.3.2.2. Parameters

Table 3.1.. API Parameters.

	Symbol	Description	Recommended default
		Basic API parameters	
1	AIXW	Intercept of wet curve (1st quadrant)	A-priori grid
2	AIXD	Intercept of dry curve (1st quadrant)	A-priori grid
3	CW	curvature of wet curve	A-priori grid
4	CD	Curvature of dry curve	A-priori grid
5	SMIX	Maximum value of SMIX	A-priori grid
6	PEX	Potential evaporation that occurs on July 15th max for week number option)	User defined
7	PEN	Potential evaporation that occurs on Jan 15th min for week number option)	User defined
8	FRSX	Maximum percent runoff	0.80
9	EFC	Effective forest fraction	A-priori grid
10	PIMPV	Fraction of watershed that acts as an impervious area	User defined
11	RIVA	Fraction of watershed covered by riparian area	User defined
12	RVAI	Antecedent Index, AI value above which riparian vegetation losses can occur	User defined
13	APIKS	Daily API recession when snow exists	0.93
14	BFPK	Daily primary base flow recession rate	A-priori grid
15	BFIK	Daily Base flow Index recession	A-priori grid
16	BFIM	Weighing factor for BFI	A-priori grid
17	AICR	Critical AIF value AIF value below which Fg=1.0	A-priori grid
18	CG	Curvature constant for groundwater inflow	User defined
		Commonly used option 1 for ET	
19	AEIX	Maximum allowed Antecedent Index of Evaporation value	User defined
20	AEIN	Minimum allowed Antecedent Index of Evaporation value	User defined
21	APIX	Maximum value that API contains	10 in
22	AEIK	Daily recession rate of AEI	User defined
23	APIK	Daily recession of API	User defined
		Option 0 for week number	
24	CS	Seasonal curvature Exponent (used only for week number option)	User defined
25	WKW	Week number when the wettest condition exhibits	User defined
26	WKD	Week number when the driest condition exhibits	User defined
		Option 2, for temperature	User defined
27	ATIR	Temperature weight factor	User defined
28	ATIX	Maximum allowed ATI	User defined
29	ATIN	Minimum allowed ATI	User defined
		Option for frozen ground	
30	CSOIL	Forest coefficient for bare ground	User defined
31	CSNOW	Accounts for insulating effect of a snow cover	User defined
32	GHC	Effect of heat transfer from below the forest layer	User defined
33	FICR	Value of forest index above which soil frost has no effect	User defined
34	CF	The EFI freezing coefficient controls the increase in FEI during cold periods	User defined
35	CP	The amount of rain that must freeze to fill soil pores FEI during cold periods	User defined
36	CT	The FEI thaw coeff.controls the decrease in FEI when thaws of the soil occurs	User defined
37	EFA	Effective forest area controls the portion of watershed affected by frozen ground	User defined

3.3.2.3. States

- API, Antecedent Precipitation Index
- SMI, Surface Moisture Index
- BFI, Base Flow Index
- BFSC, Base flow
- AEI, Antecedent Evaporation Index
- ATI, Antecedent Temperature Index
- FI, Frost Index
- FEI, Frost Efficiency Index

3.3.3. Outputs

The principal outputs of interest are surface and subsurface runoff for flood modeling. Any of the state grids listed in Appendix A can be output at a user specified time interval.

3.4. Frozen Ground (frz)

3.4.1. Description

The frozen ground technique (*frz*) works in conjunction with the SAC-SMA-HT model and therefore must be run with the *sac* technique. Koren (2005) describes the theory and validation of the gridded frozen ground model.

When the frozen ground model is run, the nodes (depths) where temperature computations are made are prescribed by the program in a way such that the numerical solution of the heat transfer equations is accurate. The heat transfer computational depths near the surface will tend to be closer together because the temperature gradients are higher near the surface. For conversion from SAC-SMA states to physical layers or from physical layers to SAC-SMA states the program adjusts the computational layer depths so that a selected set of layers are associated with the SAC-SMA upper zone (could be one or more) and a selected set of layers are associated with the SAC-SMA lower zone. If there is no frozen water, then all layers that correspond to a given SAC zone (upper or lower) will be assigned the same soil moisture value after SAC water balance calculations are made. If there is frozen water, then the allocation of water to layers in the each zone will vary depending on the amount of frozen water in that layer.

The program can output data for user-defined layers that do not necessarily correspond to the computational layers. Values for this output are computed via simple weighted averaging from the computational layers.

3.4.2. Inputs

3.4.2.1. Forcings

Required forcings are precipitation and PE.

3.4.2.2. Parameters

Default a-priori grids for two parameters required by *frz* are provided. These are:

Frz_STXT: dominant soil texture derived from STATSGO
Frz_TBOT: temperature at the bottom of the soil column estimated as the mean
annual temperature in ∘C.

The user does not have to specify any additional parameter values for the model to
run.

3.4.2.3. States

Model states include soil temperature in each layer as well as liquid and total water
contents in each layer. Appendix A provides a complete list of model states.

3.4.3. Outputs

The user can choose to output any gridded model parameters or state. When run in
conjunction with *sac*, the two models automatically share the necessary state information.

3.5. Overland and Channel Routing (*rutpix7, rutpix9*)

3.5.1. Description

As described by Koren et. al. (2004), there are two distinct routing regimes modeled
by the techniques available in HL-RDHM v. 2.4.2, hillslope and channel routing. The
rutpix7 and *rutpix9* techniques include both hillslope and channel routing.

Inputs to the routing techniques are runoff depths (e.g. mm/hr). The routing
techniques can ingest two types of runoff: fast (surface) and slow (subsurface/ground)
runoff. Within each cell, fast runoff is routed over a conceptual hillslope to a channel
(Figure 3.2), and then channel inflow from the hillslopes, combined with a slow runoff
component and upstream pixel outflows, is routed through a cell conceptual channel. A
conceptual hillslope consists of a number of uniform hillslopes (the number of uniform
hillslopes depends on the stream channel density specified for the cell and the cell area).
The conceptual channel that passes water from cell-to-cell usually represents the highest
order stream in a selected cell. A topographically defined cell-to-cell connectivity
sequence is used to move water from upstream to downstream cells and to basin outlets
(Figure 3.3). Figure 3.4 defines the SAC-SMA runoff components that are classified as
slow and fast response for routing purposes.

HRAP Cell (~ 4 km x 4 km)

Uniform, conceptual hillslopes within a
modeling unit are assumed

- Drainage density illustrated is ~1.1 km/km2
- Number of hillslopes depends on drainage density

Overland flow routed
independently for each
hillslope

Conceptual
channel
provides cell-
to-cell link

(adapted from Chow et al., 1988)
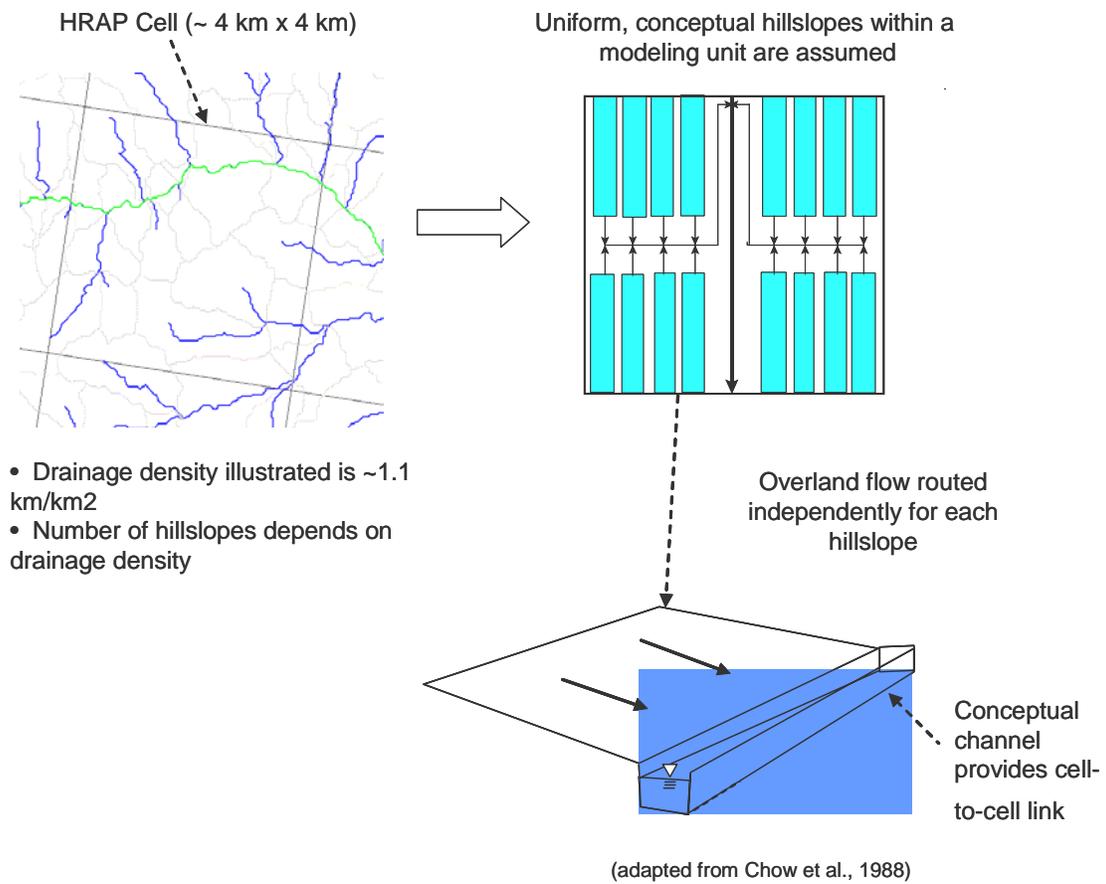
Figure 3.2. Conceptual hillslope representation
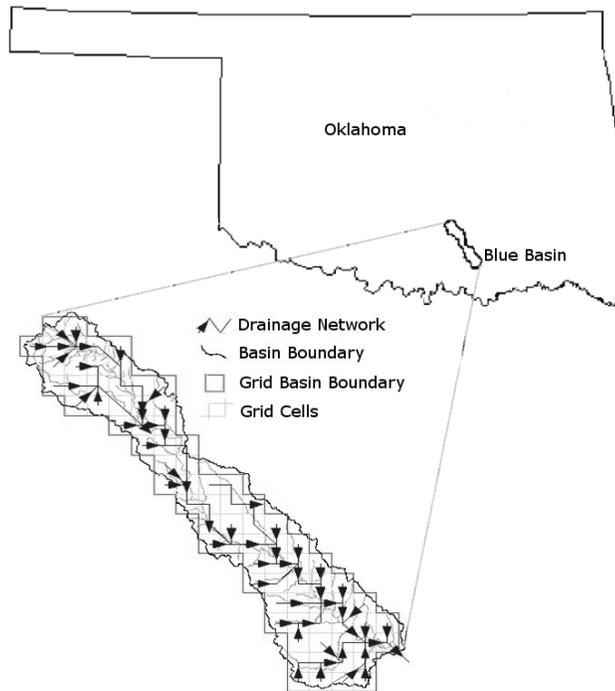
Figure 3.3. Cell-to-cell connectivity example for a headwater basin.
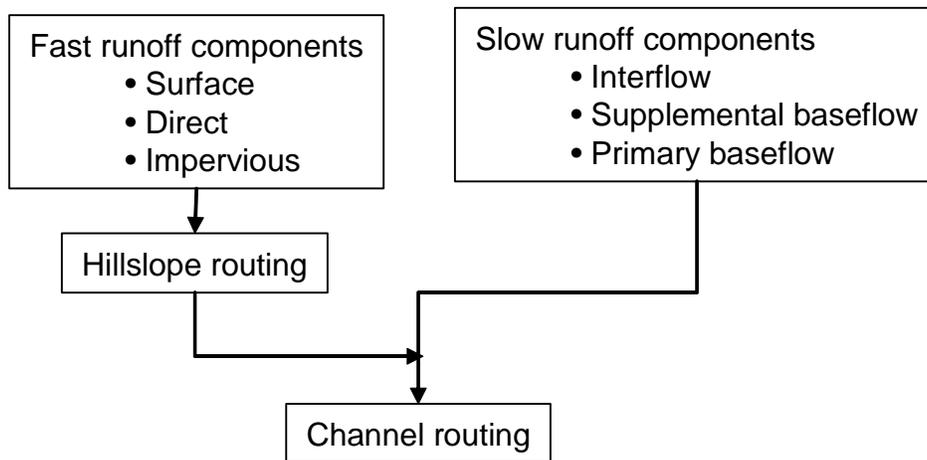
Figure 3.4. Treatment of SAC-SMA runoff states by routing model

Koren et. al. (2004) describe the mathematical formulas for hillslope and channel routing and the solution techniques used in HL-RDHM. Koren et al. (2004) also describe algorithms that can be used to derive distributed hillslope and channel routing parameters. The ability to derive reasonable distributed parameters from physical data is critical for practical implementation of the distributed model.

To solve the hillslope equations, a unique relationship is defined for each cell between the average hillslope water depth (h) and the discharge per unit area of hillslope

(q_h). Similarly, for the channel representation, a unique relationship between the discharge (Q_c) and cross sectional area (A) is defined for each cell. The kinematic model assumes that these relationships are the same for both rising and falling flood waves.

The equation relating q_h to h is,

$$q_h = 2k_q D \frac{\sqrt{S_h}}{n_h} h^{5/3} \tag{3.1}$$

where $k_q = 10^5$ is a unit transformation coefficient, and D is stream channel density in km^{-1}. To solve this equation, the hillslope routing method requires three input grids:

    (1) hillslope slope (S_h)
    (2) hillslope roughness coefficient (n_h)
    (3) stream channel density, D

For channel routing, there are currently two options available for computing the relationship between Q and A in each cell. One is referred to as the channel shape method (*rutpix7*) and the other as the rating curve method (*rutpix9*). For the channel shape method, there are four channel input grids required:

    (1) slope (S_c)
    (2) roughness coefficient (n_c)
    (3) shape parameter (β)
    (4) top width parameter (α)

Shape and top width parameters are defined based on an assumed relationship between channel top width (B) and depth (H):

$$B = \alpha H^{\beta} \tag{3.2}$$

With the channel shape method, the four input grids are converted into two basic channel parameter grids within the program, a specific channel discharge, q_o per unit channel cross-section area, and a power value, q_m, in the relationship between discharge and cross-section.

$$Q_c = q_0 A^{q_m} \tag{3.3}$$

$$q_0 = \alpha^{-\frac{2}{3(\beta+1)}} (\beta+1)^{-\frac{2\beta}{3(\beta+1)}} \frac{\sqrt{S_c}}{n_c} \tag{3.4}$$

$$q_m = \frac{\beta + \frac{5}{3}}{\beta + 1} \tag{3.5}$$

The units of channel specific discharge are $\text{m}^{3-2\text{m}}\text{s}^{-1}$ if A is in meters. It is useful to keep in mind special values for the parameter q_m when the channel shape of Equation 3.2 is assumed. The maximum value of q_m of 1.0 occurs as β approaches infinity (an infinitely wide channel). $q_m = 1.333$ when the channel has a triangular shape ($\beta=1$). $q_m=1.6667$ when the channel has a finite rectangular shape ($\beta=0$) and infinitely high banks.

When the rating curve method is used, grids of q_0 and q_m are defined explicitly by the user. The method to estimate channel shape and rating curve parameters at ungauged locations uses a combination of basin outlet streamflow measurements and geomorphological relationships.

3.5.2. Inputs

3.5.2.1. Forcings

Forcings required for the routing techniques are fast and slow response runoff. In HL-RDHM v. 2.4.2, routing techniques must be run in conjunction with one of the available rainfall-runoff models (e.g. *sac* or *api*). When this is done, the required runoff data are automatically made available to the routing model.

3.5.2.2. Parameters

The *rutpix7* and *rutpix9* techniques require a connectivity file and the parameters grids specified in Table 3.2.

Table 3.2. Routing parameter grid names.

No.	Grid Name	Description	Required for
1	rutpix_SLOPC	channel slope (S_c)	Rutpix7
2	rutpix_ROUGC	channel roughness (Manning's n, n_c)	Rutpix7
3	rutpix_BETAC	channel shape parameter (β in B = αH^β)	Rutpix7
4	rutpix_ALPHC	channel width parameter (α in B = αH^b)	Rutpix7
5	rutpix_SLOPH	hillslope slope (S_h)	Rutpix7, rutpix9
6	rutpix_DS	drainage density (D)	Rutpix7, rutpix9
7	rutpix_ROUGH	hillslope roughness (n_h)	Rutpix7, rutpix9
8	rutpix_Q0CHN	q_0 in $Q = q_0 A^{qm}$	Rutpix9
9	rutpix_QMCHN	q_m in $Q = q_0 A^{qm}$	Rutpix9

Initially, HRAP resolution, cell-to-cell connectivity files are provided to each CONUS RFC by HL. Reed (2003) describes the algorithm used to derive flow directions that are subsequently used to create connectivity files. The default, HRAP resolution

connectivity files were derived using a relatively coarse resolution DEM (15 arc-second), albeit with corrections for the digitized stream network. The default connectivity files will be adequate for many applications; however, for some areas and for higher resolution applications (e.g. ½ to ¼ HRAP), a higher resolution DEM should be used to generate the HL-RDHM connectivity and slope grids (discussed below). DEM processing programs have been developed for this, but they are not yet user friendly enough to deliver with HL-RDHM 2.4.2.

Each connectivity file delivered with HL-RDHM 2.4.2 stores a list of all HRAP cells in upstream to downstream order. Users customize the headers of the connectivity files to identify basin outlet cells of interest in a region. Chapter 9 provides a detailed description of the connectivity file format and its use. Hydrographs can be generated for any basin outlet identified in the connectivity file.

Default grids of channel slope (rutpix_SLOPC) and hillslope slope (rutpix_SLOPH) are also provided for CONUS RFCs. Hillslope slopes (rutpix_SLOPH) at the HRAP resolution were computed by averaging the slopes of individual 400-m resolution DEM cells within each HRAP cell. For hillslope calculations, the slopes of individual DEM cells were computed using the ESRI "slope" function. Representative channel slopes (rutpix_SLOPC) for HRAP cells were computed using a three step process: (a) define high resolution channels from the DEM with enough density so that at least one channel appears in each cell, (b) compute the slope of each stream segment from the DEM (dElevation/dLength), and (c) identify the stream location in the HRAP cell that best represents the drainage area of that HRAP cell within the overall drainage network and extract the channel slope for that point from the high resolution network defined in Step b. A similar procedure has been used to develop a local channel slope grid ("rutpix_SLOPCL") for possible use in flash flood modeling. For the local channel slope, Step c is modified so that the selected stream location is close to the drainage area of the local cell, independently of where it is in the overall stream network.

For the highlighted cell in Figure 3.5, the total upstream drainage area is 850 km^2. The representative channel slope for cell-to-cell routing (rutpix_SLOPC) is computed from segment A in the diagram and the local channel slope is calculated from the segment B tributary. For this cell, hillslope slope = 0.27, local channel slope = 0.18, and main channel slope = 0.005. As in this case, we expect that most cells will have hillslope slope > local slope > main channel slope.
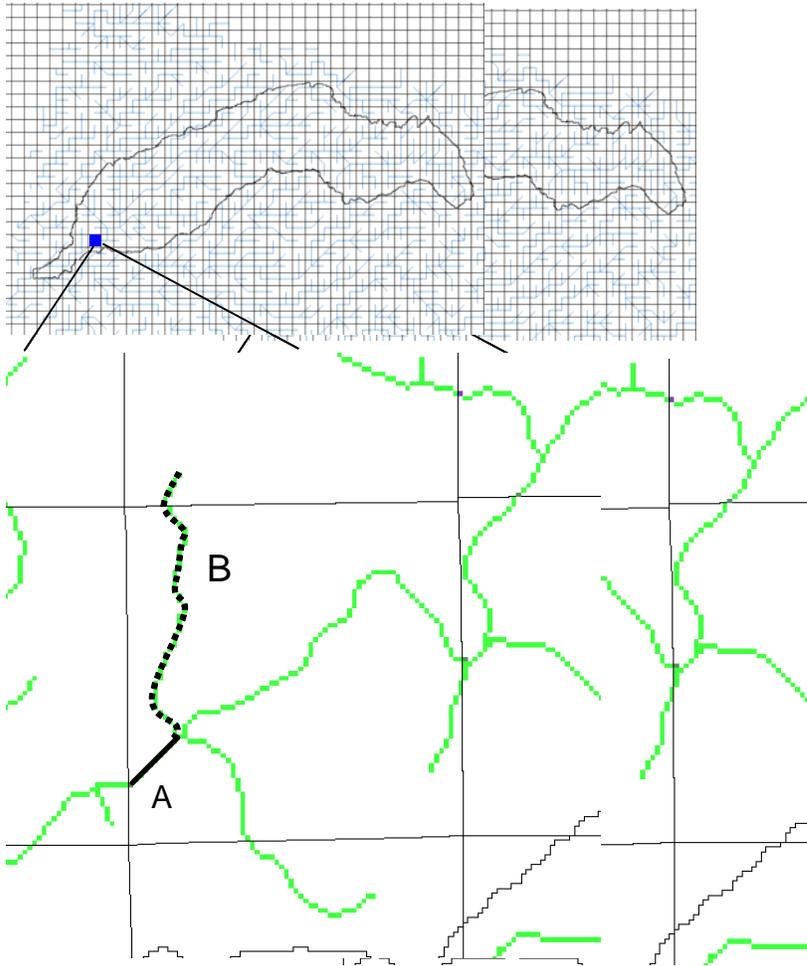
Figure 3.5. An HRAP cell in the N. Fork of the American R., CA, with stream segments for which slope is calculated. The slope of segment A is taken as the main channel slope for the HRAP cell and segment B is taken as the local channel slope.

The user is responsible for providing reasonable estimates of the hillslope parameter grids rutpix_ROUGH and rutpix_DS. If using the channel shape method, the user is responsible for populating the grids rutpix_ROUGC, rutpix_BETAC, and rutpix_ALPHC with reasonable values. If using the rating curve method, the user must generate reasonable grids for rutpix_Q0CHN and rutpix_QMCHN. Chapter 9 describes how to create these customized input grids.

3.5.2.3. States

To run *rutpix7* or *rutpix9*, values of two initial state variables must be specified: (1) *areac* – cross-sectional area of channel flow in m^2 and (2) *depth* – hillslope state in mm. The first time the model is run, initial state grids will not be available and constant values can be assigned for these states in modeled regions. A reasonable initial hillslope *depth* is 0 m in most cases. For a cold start estimate, we often use a small uniform value for

areac over a region (e.g. 0.5 m^2). When first running a model in an area, a warm-up period is required to generate reasonable spatial patterns of *areac*. In subsequent runs, states saved from earlier runs can be used as input.

Depending on the cell size, the channel routing algorithm subdivides the cell into sub-reaches for more accurate calculations. As reflected in Appendix A, this can result in multiple channel states per cell, denoted as *areac1*, *areac2*, *areac3*, etc. A typical HRAP resolution application has 3 sub-reaches in each cell. The state file with the highest index (i.e. 3 in this case) represents the cross-section state at the cell outlet.

3.5.3. Outputs
Any of the model states (e.g. *depth*, *areac3*, *discharge, etc.)* listed in Appendix A can be output as either time series or grids.

3.6. Threshold Frequency Techniques (DHM-TF)

3.6.1. *Overview*

The idea of using threshold frequencies along with distributed hydrologic models (DHM-TF) for flash flood forecasting at ungauged locations is discussed in detail by Reed et al. (2007). Two variations on the DHM-TF modeling concept are worth considering for NWS flash flood forecasting operations and techniques to test these two variations are described in this section: (1) a WFO DHM-TF approach, and (2) a Threshold Frequency - Gridded Flash Flood Guidance (GFFG-TF). The WFO DHM-TF approach is the theoretically more appealing of the two because it includes cell-to-cell routing and presents flood information across multiple scales. However, GFFG-TF is a simpler option to implement in the short term due to data requirements and current RFC/WFO operational roles and responsibilities. WFO DHM-TF requires multiple sources of precipitation compared with only one source for GFFG-TF. WFO DHM-TF also requires more model setup. Other operational scenarios could also be considered, such as running DHM-TF at an RFC or national center. The reason we focus on 'WFO' DHM-TF here is because we want to evaluate new high resolution HPE data streams and QPF data streams (HPN) that are only operationally available at WFOs. Figure 3.6 summarizes the inputs and outputs for each approach.
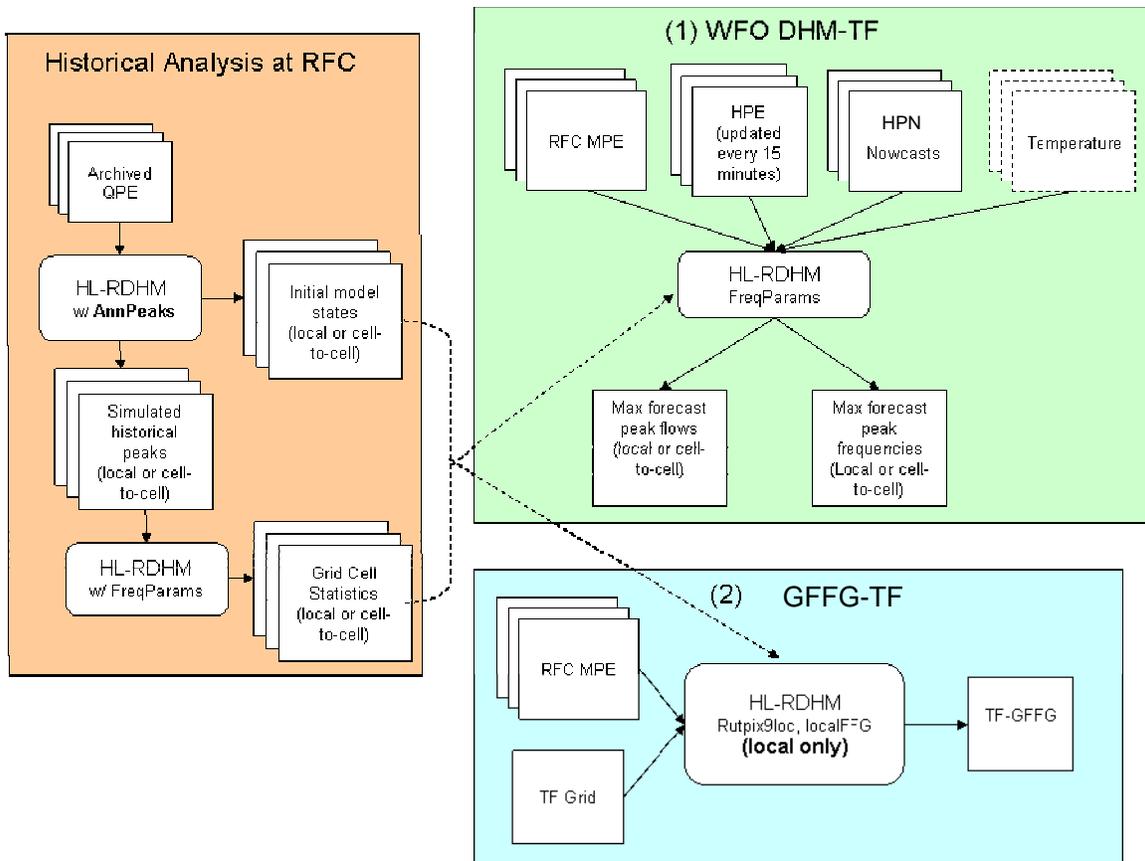
Figure 3.6. Inputs and Outputs for WFO DHM-TF approach and GFFG-TF approach.

Both threshold frequency (TF) based approaches require historical simulations to generate statistical information at ungauged locations. Reed et al. (2007) discuss advantages of the TF approach. A difficulty with the TF approach is that it requires an archive of relatively unbiased gridded precipitation data. Results from Reed et al. (2007) suggest that the quality and length of the hourly multi-sensor precipitation archive at ABRFC is adequate to support the methodology (at least in the wetter parts of the RFC where experiments were run). Recent work using MARFC MPE data indicate that a much shorter record of unbiased rainfall data is available in the MARFC area. To enable testing of the TF approach in MARFC, simple techniques to produce a bias-corrected archive of hourly precipitation grids were developed. Section 3.7 describes these techniques.

As of 8/4/2008, all testing of the DHM-TF and GFFG-TF techniques described in this section has been done without running a snow model. Incorporating a snow model into the procedures described below should work with no additional HL-RDHM enhancements, but has not yet been tested.

3.6.2. *WFO DHM-TF*

The proposed operations concept for WFO DHM-TF is somewhat similar to the current Site Specific Hydrologic Predictor (SSHP) model but with several advantages. The

vision for DHM-TF is to produce gridded flash flood information covering the entire WFO domain, not just within small basins where lumped river models are run. The techniques available to setup DHM-TF should make implementation of DHM-TF over large areas much simpler than SSHP.

We assume that RFC's would do DHM routing parameter estimation, model set up, and derivation of initial statistical parameters for each model grid cell. This is, in fact, less work then the full model calibration process used for river forecasting at gauged locations (the approach recommended for SSHP applications). The standard for improvement is to do better than the current lumped-model FFG for ungauged locations. Results from Reed et al. (2007) suggest that it is possible to get results better than existing lumped-model based FFG approaches without full calibration of the DHM against observed streamflow. Calibrated distributed model parameters may be used within DHM-TF as they become available; however, the historical statistics need to be re-computed any time an improved calibrated parameter set becomes available.

Initial states and statistical parameters would be transferred to WFOs to get the model going. After that, unlike SSHP, we don't expect updated states (including MODS) to be needed by the WFO on a regular basis. Experience with distributed hydrologic model simulations suggests that reasonable model states can be maintained as long as the quality-controlled MPE products are used to force the model. Therefore, instead of transferring model states from the RFC, we suggest transferring the best available, quality-controlled MPE data on a regular basis to update model states. Use of unmodified runs is recommended for DHM-TF so that real-time model behavior is consistent with historical statistics. Although current procedures only use observed MPE data to generate historical simulations, future procedures could include forecast precipitation data sets as extended archives become available to better account for uncertainty in the forecast data component.

During a given forecast period (e.g. 3 days), the primary output from the WFO DHM-TF model will be grids of the annual maximum return period associated with the maximum simulated flow in each cell during the period. These frequencies provide information about the relative severity of flooding expected in each grid cell. Forecast frequencies are compared to threshold frequency grids derived from local information such as information about channel morphology (bankfull frequencies), engineering design standards, and flood frequencies on gauged rivers. Maximum flow grids can also be produced. If desired, hydrograph outputs can be examined at selected model grid cells using standard HL-RDHM functionality. OHD is working on a real-time prototype of the DHM-TF concept for two study areas in Maryland.

DHM-TF simulations can be done with or without cell-to-cell channel routing. The option without cell-to-cell routing (referred to as 'local' routing) was developed primarily for the GFFG-TF approach (described further below); however, this local grid information may also be useful in the WFO DHM-TF applications where first order tributaries directly connected to big rivers are flooding, but are not resolved by our coarse resolution model grid cells. For example, a large river flowing through a non-headwater,

2 km-by-2 km cell may not be flooding, but a low order tributary channel within this cell may be flooding. The local routing technique (rutpix9loc) described below can highlight this local flooding risk. Although not explicitly modeling these local channels (e.g. draining less than 4 km^2) when using local routing, we derive locally representative hillslope and local tributary slopes from higher resolution DEMs (See Figure 3.5). If desired, forecasters could run two separate instances of DHM-TF, one that includes cell-to-cell routing and one that does not. This approach would also require two separate historical runs to generate both local and cell-to-cell statistics for each cell in the domain.

Figure 3.6 shows that both historical and real-time runs are required to implement DHM-TF. In fact, the scheme we have used to implement a DHM-TF prototype in MD requires two separate real-time runs (Figure 3.7). Run 1 is used to update model states using quality controlled precipitation from the RFC. Run 2 uses higher resolution data created at the WFO to more accurately capture flashy events. In the current setup, the
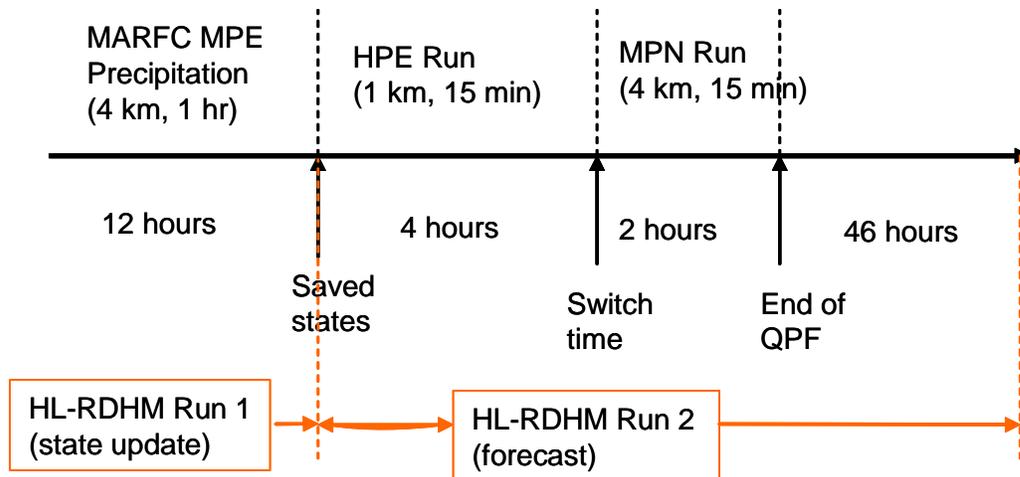
Figure 3.7. DHM-TF Forecast Mode Prototype Time Line

hydrologic model time step is 1 hr for both Run 1 and Run 2; however, the rainfall forcing in Run 2 is updated every 15 minutes and the model is rerun every 15 minutes to allow use of updated rainfall information. The time step for Run 2 is always kept at 1 hr to match the time step that is possible with the available historical precipitation data. Both Runs 1 and 2 are executed from a cron in the prototype.

Table 3.3 summarizes the HL-RDHM operations required to implement DHM-TF. Section 3.8 provides a more detailed description of the relevant operations.

Table 3.3. Example sequence of steps to setup and run WFO-DHM-TF

Step	HL-RDHM Techniques ('operations')*	Notes
1. Historical simulations to generate annual	Calsac, calrutpix9, AnnPeaks	The first year can be a warm-up period but, if so, delete the annual peak grid for that year so it is not used by subsequent programs. Alternatively,

peak grids		you could run a warm-up period without the AnnPeaks option and save the states for use at the beginning of your historical simulation run. *Example input card: dhmtfexample/oct07/adjannpeaks.card*
2. One time step simulation to generate statistical parameters for each cell from the annual peak grids	Calsac, calrutpix9, FreqParams	Creates the logmean, logstd, and wlogskew. *Example input card: dhmtfexample/oct07/freqparams.card*
3. Real-time simulations to maintain model states	sac, rutpix9	Maintain model states with highest quality precipitation forcing. Use relative dating. *Example input card: dhmtfexample/auto_run/update_states.card*
4. Real-time simulations to generate flow and frequency grids	calsac, calrutpix9, getMaxRet	Real-time forecasts with high resolution inputs and gridded outputs. Use the relative dating. *Example input card: dhmtfexample/auto_run/read_states_and_forecast.card*

* Step 3 requires sac rather than calsac because states are output at multiple time steps inside the time loop. All other steps could use either sac or calsac, rutpix9 or calrutpix9, etc.

3.6.3. *GFFG-TF*

GFFG-TF is a variation on the DHM-TF concept. Similar to the ABRFC-GFFG, each grid cell is treated as a headwater watershed and the amount of rain required to cause flooding is estimated. Although cell-to-cell routing like that used in the DHM-TF approach is desirable for more complete flood information over a range of scales, the GFFG-TF approach (without cell-to-cell routing) is still worth pursuing because it also has advantages over the current operational FFG and it is much simpler to implement operationally than the full WFO DHM-TF. GFFG-TF is simpler to implement because (1) It fits within the existing operational paradigm and infrastructure, and (2) is amenable to use with advective-statistical forecasts that provide probabilistic information about the occurrence of future rainfall in a cell but do not provide a continuous spatial field of rainfall.

The drawback of the local approach is that it only provides information at pre-defined spatio-temporal scales, and cannot be easily extended to intermediate scales between a model cell and an RFC basin. It cannot account for flooding that occurs very far downstream from where heavy rainfall occurs. However, this local approach, like the ABRFC-GFFG approach, does account for local (cell-level) variations in soil properties, slope, and antecedent moisture conditions. Relative to ABRFC-GFFG, the GFFG-TF

approach has the benefits of (1) inherent bias correction, (2) automatically accounting for the channel states, (3) kinematic routing compared to the unit hydrograph, (4) SAC-SMA compared to SCS for runoff generation, (5) flexibility in defining frequency thresholds across a region, and (6) ease with which Snow-17 can be incorporated. The biggest drawback relative to GFFG-TF is the requirement for a historical data archive. Implementation of GFFG-TF also require some work to estimate local routing parameters.

The steps to implement GFFG-TF are described in Table 3.4.

Table 3.4. Example sequence of steps to setup and run GFFG-TF

Step	HL-RDHM Techniques ('operations')	Notes
Generate local routing parameters (Generate local values for the specific discharge coefficient in rutpix9loc – the rutpix_Q0CHN grid)	None	I've done this only using the rating curve (rutpix9) method by (1) running the standard HL-RDHM routing parameter estimation procedures for enough gauges to cover the region of interest, (2) using GIS (e.g. GRASS) to query the values of rutpix_Q0CHN in headwater cells (they should be nearly identical unless there are slight differences in drainage area due to the map projection), (3) using GIS to assign the same Q0CHN values to all pixels in the basin, and (4) converting GIS grids into a new rutpix_Q0CHN to be used for local routing.
Generate historical peaks	Calsac, rutpix9loc, AnnPeaks	As delivered, the standard name for the channel slope grids is rutpix_SLOPC and for the local slope grids is rutpix_SLOPC1. Currently, the rutpix9loc algorithm does not know this. Therefore, before using local routing with rutpix9loc, rename the ruptix_SLOPC1 to rutpix_SLOPC or create a symbolic link to the correct slope grid. *Example input card: dhmtfexample/abrfc/local_states.card*
Compute the grid statistics.	*calsac rutpix9loc FreqParams*	Can run just for one time step to output gridded statistics. The calsac and rutpix models are run only to be sure that the analysis domain is defined

		properly for the FreqParams calculations. *dhmtfexample/abrfc/freq_params.card*
Generate initial states for the time period of interest (optional)	*Calsac, rutpix9loc*	*dhmtfexample/abrfc/local_states_07_2 004.card*
Simulate GFFG-TF	*Calsac, rutpix9loc, localFFG*	*dhmtfexample/abrfc/Ffg.card*

3.7. <u>Simple Precipitation Bias Correction</u>

In MARFC, we know that the archived multi-sensor MPE precipitation grids (hourly, 4-km) tend to have a low bias from approximately 1998 to 2003, causing severe underestimates in runoff simulation if used without forecaster correction. This conclusion is based on unpublished OHD analysis and technical report prepared at MARFC (Cognitore et al.). The OHD analysis compared monthly gridded precipitation from MPE with monthly gauge-based results from the PRISM project (Reference) and annual runoff simulations to observed streamflow data in two basins. Runoff simulations forced by 2004-2006 MPE data appear much improved over those from the 1998-2003 period. We suspect that MPE archives at other RFCs may also exhibit periods of lower quality data, and the variation in quality will not necessarily correspond to the same time periods in different RFCs. We know that ABRFC started scrutinizing MPE output for hydrologic modeling purposes much earlier and therefore, have a more unbiased archive back to 1996.

The theory behind the DHM-TF procedure depends on the availability of an unbiased precipitation archive. The archive length requirement is unknown, but the longer the better. Therefore, we have developed a simple procedure to create a bias corrected MPE data set using MPE precipitation archives and monthly PRISM data as input. In using this bias correction, the goal is to extend the usable archive length for DHM-TF statistical calculations and traditional hydrologic model calibration. The procedure assumes that recent data will not need as much correction due to advances in RFC tools and procedures.

Two HL-RDHM techniques have been developed to implement this. The "monthlySum" technique adds up hourly MPE grids to create grids of monthly totals. For each cell, the "xmrgAdjust" technique computes the ratio of the monthly PRISM to the monthly MPE values and multiplies each hourly value in that month by the ratio. This forces the sum for the month to equal the PRISM data. More detail on these techniques is provided in the next section.

3.8. <u>HL-RDHM Operations</u>

3.8.1. *monthlySum*

The `monthlySum` operation reads xmrg formatted hourly precipitation data and generates grid files containing monthly total precipitation. The time period for analysis is specified using the standard HL-RDHM key word "`time-period`". The program outputs a rainfall grid in xmrg format for each month of the analysis period. If the analysis period starts in the middle of the month and runs into the next month, the grid for the first month will contain partial month totals. If the analysis period ends in the middle of a month, no output file will be created for the last partial month. The grids are always written to the directory where the program was executed, not the standard HL-RDHM location specified by the "`output-path`" key word. The operation will typically be used by itself without any other operations.

The algorithm adds all precipitation values that are greater than –0.0001 to compute a monthly total. If more than 72 hours of data are missing in a cell, the program writes –1 as the monthly sum for that cell. The program also creates a grid for each month containing the number of missing values in that month (also in the xmrg file format).

The file name conventions for program outputs are:

```
monthsumMMYYYY.gz
nmissingMMYYYY.gz
```

3.8.2. *xmrgAdjust*

The `xmrgAdjust` operation reads xmrg formatted hourly precipitation data, monthly PRISM grids, and monthly MPE grids, and outputs adjusted hourly precipitation grids in xmrg format. The time period for analysis is specified using the standard HL-RDHM key word "time-period". When the `xmrgAdjust` operation is selected, the user should output the adjusted hourly grids by specifying "`output-grid-inside-timeloop=xmrg`". In this case, the hourly output grids will be written to the standard xmrg output directory underneath the standard HL-RDHM output path specified by the "`output-path`" key word. The operation will typically be used by itself without any other operations.

For each hour and each pixel, `xmrgAdjust` will check to make sure that the MPE monthly value is greater than zero and the PRISM monthly value is greater than zero. If so, an adjustment factor, the ratio of PRISM monthly value to the MPE monthly value, is calculated and multiplied by the original hourly value to compute a new hourly value. By doing so, the monthly sums of new hourly values is forced to equal the PRISM monthly sum.

The file name conventions for program outputs are the standard hourly xmrg filenames.

3.8.3. *AnnPeaks*

`AnnPeaks` keeps track of annual maximum peak discharges for each cell and stores a grid of annual maximum peaks at the end of each water year in the simulation. An output

grid of annual maximum peaks is written any time a simulation for September 30 at 23z is completed. If you run a simulation starting after October 1 at 00z, the first annual grid saved will not be for a complete year. The operation should be run after another operation that generates gridded discharge. Operations that generate gridded discharge include `rutpix9`, `rutpix7`, `calrutpix9`, `calrutpix7`, and `calrutpix9loc`.

`AnnPeaks` outputs files with the following naming convention:

        `pkYYYY.gz`

Output files are written in xmrg-like format.

3.8.4. *FreqParams*

Before the time loop, read the annual maximum peak grids generated from the `AnnPeaks` operation, compute the frequency distribution for each cell and write the distribution parameters (logmean, logstd, and wlogskew) to HL-RDHM memory (the `PixelGraph` object) for possible use in other programs. Inside the loop, convert discharge to a frequency.

This version uses the Bulletin 17B method (log-pearson Type III frequency distribution) suggested by the Water Resources Council (1982) (http://water.usgs.gov/osw/bulletin17b/bulletin_17B.html) to compute the probability that a computed discharge will exceed the annual maximum discharge in any given year. The routines to compute map skew for the Bulletin 17B method were taken directly from the PeakFQ software distributed by the USGS; however, most other statistical calculations were implemented using C++ and the GNU C Scientific Libraries for easier integration within HL-RDHM. In implementing the Bulletin 17B method, we include the low outlier test, but not the high outlier test since we do not have historic data to judge whether or not to keep high outliers. Calculations were checked against examples in Bulletin 17B and in Hydrology Textbooks (Hydrology and Hydraulic Systems by Ram S. Gupta; Applied Hydrology, Chow et al., 1988).

The module should be run after a function that generates discharge. For example,

    `operations = sac rutpix9 FreqParams`

It will also work with the `calsac` and `calrutpix9` operations.

Although, the `FreqParams` operation can run for multiple time steps and output exceedance probability and return period grids at the request of the user, a major use of `FreqParams` is simply to output statistical parameter grids for use in other operations (see `getMaxRet`, and `localFFG`). To do this efficiently you can run HL-RDHM with the `FreqParams` operation for only a single time step and output the statistics grids using standard HL-RDHM output options. For example, you can specify the following:

```
output-grid-before-timeloop =  logmean logstd wlogskew
error_code
```

If there is insufficient data to calculate the statistical parameters in a given grid cell, then an `error_code` is written to the `error_code` grid. Possible error codes include:

1. Less than 5 years of data makes the approximation for the Kn variable invalid (Handbook of Hydrology, 1993, Eq. 18.7.5).
2. Bulletin 17B, Appendix 5, Outlier detection procedures are invalid due to removing too large of a fraction of the record. We have seen this in dry areas where some years the maximum flow is zero or very low.
3. Bulletin 17b, p.5-3; logskew is out of range for a good approximation

`FreqParams` computes the following grids inside the time loop:

probam: stores the probability that the computed flow will exceed the annual maximum flow in any given year

returnp: stores the return period associated with a given probability level, returnp = 1/probam

These grids are available for standard output when, e.g.

```
output-grid-inside-timeloop             = probam, returnp
```

The user must specify the years for which annual peak flow grids are available. This is specified using the HL-RDHM 'user-data' option. For example:

```
#------------------
#----- user-data
#------------------
user-data = startyear 1998
user-data = endyear 2006
```

3.8.5. *getMaxRet*

When HL-RDHM is run in forecast mode, the `getMaxRet` option may be used to compute the maximum discharge and return period grids produced during the forecast period. This operation should only be used in forecast mode when a switch time is defined. During the forecast time period, the operation keeps track of the maximum discharge in each cell. At the end of the forecast loop, the program converts discharge into frequency and then return period. In order to do this, it needs to read in parameters of the logPearson Type III (logmean, logstd, and wlogskew) distribution defined from previous runs using the `FreqParams` operation.

Input files:

```
logmean.gz
logstd.gz
wlogskew.gz
```

* For the first version, this operation does not search all of the HL-RDHM input paths. The program looks for these files only the directory where the program is being executed. Note also that <u>the user must create subdirectories</u> "`maxdsch`" and "`returnp`" to hold the output grids.

Output files:

```
../maxdsch/maxdschMMDDYYYYHHz.gz
../returnp/returnpMMDDYYYYHHz.gz
```

The date stamp on the output files will be equal to the switch time for the forecast run.

3.8.6. *Rutpixloc9*

This algorithm allows routing in individual grid cells but no routing to the next downstream cell. The technique can only be used with *calsac* (not *sac*) because it uses global variables for computational efficiency. The `rutpix_SLOPC1` delivered with HL-RDHM data sets should be used as input; however, the program still looks for a file named `rutpix_SLOPC`, so the user should change the name or provide a symbolic link to the rutpix_SLOPC1 grid.

A five year hourly simulation on 33,036 pixels with `calsac` and `rutpixloc9` took 4 hours. As of 8/4/2008, there is currently no `rutpixloc7` technique to allow local routing with the 'channel shape' parameterization method, but this could easily be added.

3.8.7. *localFFG*

The FFG in each grid cell is estimated through repeated FFG guesses to match a target threshold probability level. A grid of target annual maximum exceedance probability levels can be supplied by the user (e.g. /fs/hsmb5/hydro/users/sreed/dhmtf/abrfc/threshF.gz). Figure 3.8 shows an example of the inverse of this grid, the average recurrence interval. `LocalFFG` iteratively runs the rainfall-runoff and routing models out for a period of interest (currently 6 hours), the maximum forecasted flows during this period are stored and converted to a probability level, a check is made to see if the result is close to the target probability level, and then an updated guess is made. The updated guess is based on the bisection root finding algorithm. The initial FFG guesses for the iterative procedure can be from either a saved grid from a previous run or a constant value over the domain.
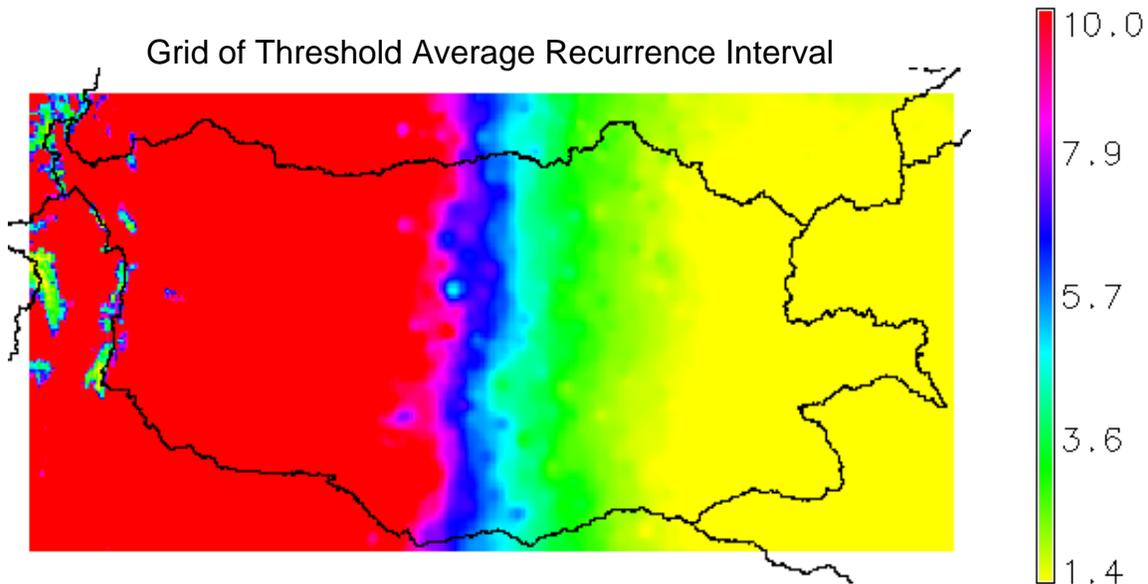
Figure 3.8. Example of the average recurrence interval thresholds. The inverse of this, the annual maximum exceedance probability (threshF), is supplied to the `localFFG` program.

The tolerance for how close the solution needs to come to the input gridded frequency value is hard-coded (beforeLoopInside). The current 'tolerance' value is 0.01. Thus, if the user specifies a threshold exceedance probability of 0.5, the `localFFG` function will iterate to find the precipitation that will produce a flow with an exceedance probability somewhere between 0.49 and 0.51. Also, the program has a hardcoded limit on the maximum number of iterations (10) allowed to converge on an acceptable GFFG solution (`localFFGInside`). If a solution cannot be obtained with these defined constraints, then the output value will obtain NODATA. Some cells may not have valid input statistics for logmean, logstd, and wlogskew (see description of the `FreqParams` algorithm for reasons). In these cases, NODATA is also written to the output grid.

It might be possible to improve the speed of the `localFFG` algorithm using a more sophisticated optimization routine. However, it does use global variables to optimize speed and **therefore runs only with** `calsac` **and** `calrutpix9`. GFFG-TF can be generated over the entire ABRFC area (33,036 cells) on one of our 'lx-nhdr' workstations in about 8 seconds.

As of 8/4/2008, GFFG-TF only computes 1 hour GFFG; however, other durations can be easily added as needed.

4. AUTOMATIC CALIBRATION

4.1. <u>Calibration technique</u>

    While tremendous advances have been made in recent years in estimation of
parameters in lumped hydrologic models and assessment of their uncertainty, the
currently available automatic calibration techniques are generally based on global
optimization which requires a very large number of function evaluations. As such, they
are not very amenable to estimation of distributed parameters in fine-scale hydrologic
modeling. In addition to computational considerations, automatic calibration based on
global optimization does not necessarily transfer the spatial patterns of pedologic and
physiographic patterns observed in the spatial data to the hydrologic model parameters
very well, and hence is not very conducive to hydrologic modeling over a large area
where interdependence of hydrologic model parameters among adjacent basins may be
important. Given the above observations, a simple optimization algorithm is
incorporated into HL-RDHM which is based on reasonable estimates of a priori
parameters and 'limited' optimization of them.

    This local search technique, referred to herein as the Stepwise Line Search (SLS), is
essentially a successive minimization along coordinate directions (Press et al., 1986) but
with a fixed step size along each coordinate and one-step propagation at a time.
Algorithmically, SLS is made of the following steps: 1) start with the a priori estimates of
the hydrologic model parameters, 2) with the rest of the parameter estimates fixed to the a
priori, increase or decrease the value of the first parameter by one step in the direction of
decreasing objective function value, 3) with the first parameter now set to the new (or
old, if the objective function value did not decrease) value, decrease or increase the value
of the second parameter by one step in the direction of decreasing objective function
value, 4) repeat Step 3 until the objective function is minimized with respect to each of
all remaining parameters, 5) repeat Steps 2 through 4 until no further reduction in the
objective function is realized. To improve SLS time performance, a simplification has
been added to the above basic algorithm: if a given parameter value remains the same in
three consecutive loops, where a loop represents an iteration of stepwise line search of all
parameters being optimized, the parameter is eliminated from further consideration. This
simplification usually results in as much as a four-fold reduction in computational
amount. The set of such non-sensitive parameters varies depending on the basin and the
time period, a reflection that interdependence among parameters varies in space and time.
The main benefits of SLS are physically realistic posterior model parameter estimates
(given reasonable a priori estimates), algorithmic simplicity, and computational
efficiency. SLS requires an average of 50 times fewer function evaluations compared to
the SCE algorithm (Duan et al., 1992). For example, Fig. 4.1 compares SCE and SLS
iteration processes considering a number of function evaluations needed to converge to
the optimum and overall distance of optimal parameters from a priori parameter space.
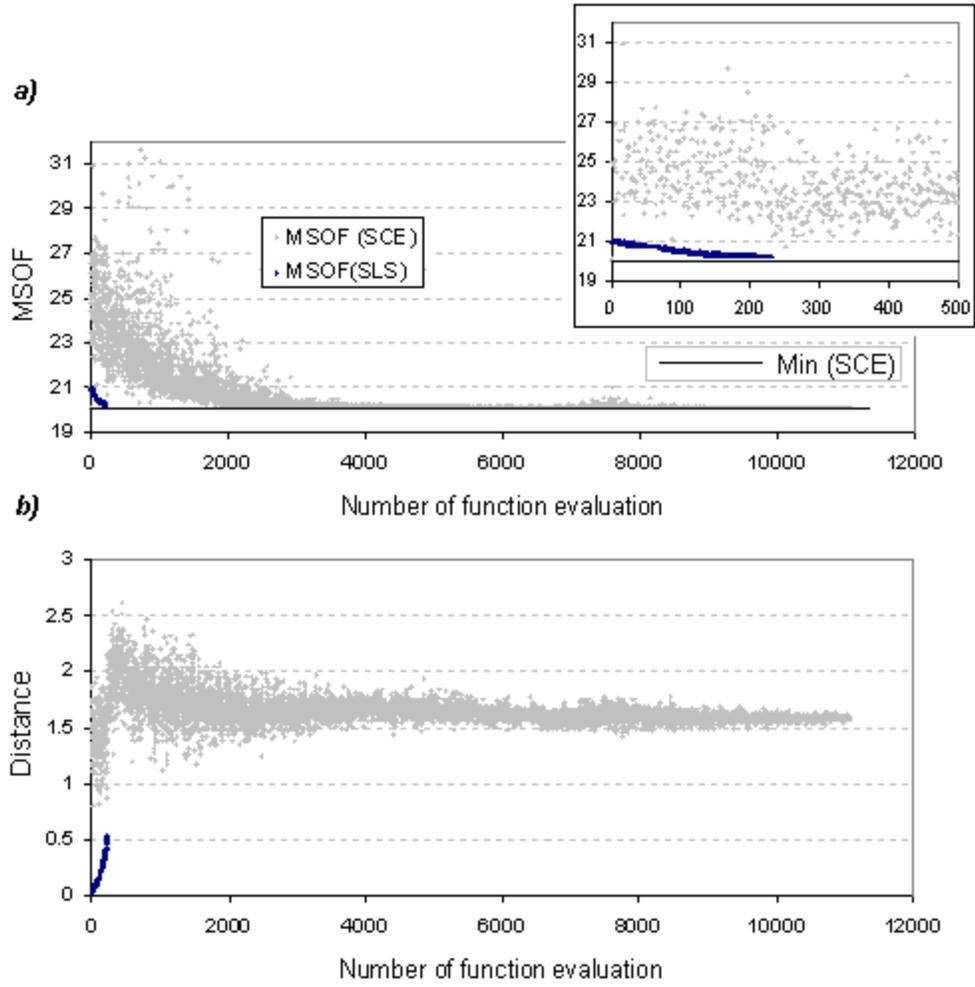
Fig. 4.1. (a) Multi-scale objective function vs. the number of function evaluations in the optimization sequence. The upper-right plot is a zoom-up near the origin. (b) Distance from the a priori parameters vs. the number of function evaluations. The light and dark markers denote the multi-scale objective function value and the distance associated with the SCE- and SLS-optimized parameters, respectively.

To perform calibration in a reasonable time frame, another simplification is introduced. Scale factors assumed to be uniform over basin are calibrated instead of actual model parameters which may vary over basin. It is assumed that a spatial distribution of a priori parameters is realistic. One parameter should be defined to run SLS, the parameter step size. Preliminary tests suggest that step size changes do not significantly affect calibration results. So, in this program version, the step size is estimated as a ratio of defined parameter ranges to a constant number of steps equal 40.

4.2. <u>Optimization criteria</u>

One of the most important aspects of manual calibration of hydrologic models is that human beings are very good at assessing the quality of a model simulation

simultaneously at many temporal scales of aggregation. To emulate this multi-scale nature of manual calibration, an objective function composed of contributions from a wide range of time scales of aggregation is selected. The rationale behind the approach is similar to Parada et al. (2003) and is not elaborated on here. The particular objective function used in this work has the following form:

$$ J = \sqrt{ \sum_{k=1}^{n} \left(\frac{\sigma_1}{\sigma_k} \right)^2 \sum_{i=1}^{m_k} \left(q_{o,k,i} - q_{s,k,i}(X) \right)^2 } $$

(4.1)

where $q_{o,k,i}$ and $q_{s,k,i}$ denote the observed and simulated flows averaged over time interval k (i.e. the k-th aggregation scale), σ_k denotes the standard deviation of the observed discharge at that scale, n denotes the total number of scales used, and m_k is a number of ordinates at the scale k. Note in the above equation that the weight associated with each term is given by the inverse of the standard deviation of the flow at the respective scales. This weighting scheme assumes that the uncertainty in modeled streamflow at each scale is proportional to the variability of the observed flow at that scale. Another important motivation for using the multi-scale objective function (MSOF) is that it smoothes the objective function surface and hence reduces the likelihood of the search getting stuck in small 'pits'.

Selection of specific scales and their number depends on the river regime. Too many low frequency scales can bias calibration to fit most low flow. Tests for a range of basins in Oklahoma and Texas suggest that reasonable results can be obtained using four time scales: 1, 24, 240, and 720 hours.

5. DATA TYPES AND FORMATS

5.1. Grid

5.1.1. XMRG, XMRG-like

HL-DHMS reads OHD XMRG formatted (http://www.nws.noaa.gov/oh/hrl/pps/pps.htm) precipitation files produced by MPE (typically one file for each hour).

A format similar to XMRG (XMRG-like) but with a small difference in the header is use for other data types (e.g. model parameters or saved model states from a previous run). XMRG-like grids are binary files with two header records. The first record is the same as the XMRG record and consists of four integers: X- and Y-origins of HRAP, and numbers of columns and rows in the file. The second record differs from XMRG records. It includes two variables: an integer scale factor to convert integer*2 values into real values (to get the true values, divide the value stored in the file by this factor), and number of bytes per value (2 for integer*2 format and 4 for real*4 format). This allows keeping all grids in either integer*2 or real*4 formats. As with the XMRG format, data

values are written to the file in row-column format, with the southernmost row written to the file first and the northernmost row written last.

Detailed format of XMRG-like grids are as following.

First record:

(4bytes Int)	X origin (4 byte float)	Y origin (4 byte float)	num of cols (4 byte int)	num of rows (4 byte int)	(4bytes Int)

Second record:

(4bytes Int)	Scale factor (4 byte int)	Num of bytes per value (4 byte int)	Cell Size (4 byte float)	No Data Value (4 byte float)	(4bytes Int)

Or

(4bytes Int)	Scale factor (4 byte int)	Num of bytes per value (4 byte int)	(4bytes Int)

Data values for each row (repeat the (num of row) times) :

(4bytes Int)	Cell value 0 (2/4 byte short/float)	......	Cell value (cols – 1) (2/4 byte short/float	(4bytes Int)

5.1.2. 1D XMRG

When HL-RDHM is run for research and calibration purposes, it is very common to make repeated runs with the same temporal input data, such as precipitation and air temperature, but different parameter data set. Because the temporal variables are stored in XMRG files which are individual binary gridded data files for each hour, expensive operations such as opening/closing files, retrieving values for selected pixels, and accumulating and averaging over the user specified time step, are involved. To avoid such expensive operations in repeated runs, and therefore improve the performance, '1D XMRG' files are an optional program output. When the program starts, it first checks if 1D XMRG files exist. If the 1D XMRG files exist and contain valid values, the program

will read temporal variables from the 1D XMRG files, otherwise, the program will create an 1D XMRG file for each temporal variable and 'independent' basin from the data it reads from normal XMRG grids.

1D XMRG files are binary files containing only retrieved values for selected pixels. There is one 1D XMRG file for each temporal variable and 'independent' basin/window. An 'independent' basin is a basin that has no downstream sub-basins. An 'independent' basin may contain upstream basins however. The 1D XMRG file name has the format of 'YYYYY_XXXXX.1d', where YYYYY is the temporal variable name and XXXXX is the independent basin id. If the simulation is over a window without routing, the 'independent' basin is the default name for the window, that is '___WHOLE_WINDOW___'.

Each 1D XMRG file contains two parts, a header part and a data part. The header is composed of begin time, end time and time step of the data. It has 38 characters. The format is 'YYYYMMDDTHHMMSSYYYYMMDDTHHMMSSHH:MM:SS'. The first 'YYYYMMDDTHHMMSS' is the begin time; the second is the end time. The 'HH:MM:SS' is time step. The data part contains binary float numbers (4 bytes on most machines). The order in which the pixel values are written is pre-order, that is, the downstream pixel is written before all its upstream pixels.

When the program starts to run, it first reads input deck to get simulation period, time step and selected basins in case it runs over a connected domain. Then it will check to see if 1D XMRG files exist. For a particular temporal variable and basin,

- If the 1D XMRG file doesn't exist, the program will create an 1D XMRG file using the corresponding variable and basin id. Then it writes the header with the begin time, end time, and the time step of the simulation. Within the time loop, pixel values of the independent basin will be written to the file.

- If the 1D XMRG exists. The program will read in the header.

  o If the time step in the header differs from the time step in the input deck, it removes the content of the 1D XMRG file and rewrites the 1D XMRG file.

  o If the time steps are same. Then it checks the time period.

    ▪ If the time period in the 1D XMRG file doesn't overlap with the simulation period, it removes the content of the 1D XMRG file and rewrites the 1D XMRG file.

    ▪ If the time periods overlap, but the simulation period has an earlier begin time then the time period in the 1D file, it removes the content of the 1D XMRG file and rewrites the 1D XMRG file.

- Otherwise, the program reads temporal inputs from the 1D XMRG file and appends data to the file if the simulation end time time exceeds the end time of the 1D XMRG file.

For a particular temporal variable, if one selected independent basin is missing 1D XMRG, input data for this variable will be read from normal XMRG grids for all independent basins, even though other basins have 1D XMRG files.

For a particular independent basin, if one temporal variable is missing 1D XMRG, input data for this variable will be read from normal XMRG grids for all independent basins, even though other basins have 1D XMRG files for this variable.

The reason we read in all basins from normal XMRG grids if 1D XMRG file is missing for only one variable or independent basin is that the normal XMRG grid will be open anyway.

Caution should be taken when the simulation is over a window. Because the 1D XMRG file name is a generic name, '___WHOLE_WINDOW___', when the simulation window has been changed to other window between the simulations, the program will never be able to know that. Therefore you should delete the 1D XMRG files created by the previous runs manually.

You can specify a multiplication factor for a temporal variable in the input deck. However, since only independent basins are used when reading from 1D XMRG files, the multiplication factor for dependent basins will not take effects. Things become worse when the reading is mixed with 1D XMRGs and normal XMRGs, because when reading from normal XMRGs multiplication factors for dependent basins will take effect.

The 1D XMRG feature can be turned off using the option "ignore-1d-xmrg" in the input deck (see Chapter 8).

5.2. GRASS Scripts for Displaying and Managing HL-RDHM Output Grids (XMRG like)

A set of C++ programs and scripts are available to convert HL-RDHM grids (XMRG like) into GRASS format and assist in displaying loops of these grids within GRASS. The file docs/readme_grass_programs.txt describes these programs. The most useful is perhaps the 'xg' program which provides a lot of flexibility in importing grids into and deleting files from GRASS. For example, the program can be used to import a time sequence of any grid type (e.g. discharge, runoff, soil moisture, ffg) or a list of filenames.

5.3. Time series

Time series data are output in the NWSRFS Calibration System DATACARD File Format (NWSRFS User's Manual, VII.2), hereafter referred to as the OHD Data Card format.

6. OVERVIEW OF IMPLEMENTATION STEPS

Table 6.1 summarizes the basic steps required to use HL-RDHM (and DHM) with a connected domain. Not all of these steps are required to run the model using an unconnected domain. The level of calibration effort required to improve upon existing models will vary with application (e.g. flash flood vs. river forecast) and location. Column 2 identifies specific tools that are available to accomplish these steps and Column 3 specifies where documentation on these tools is provided.

Table 6.1. Basic Steps in Distributed Hydrologic Modeling with HL-RDHM and DHM

Simulation/Calibration Mode			
Step	**Available Tool**	**More information**	**Comments**
1. Organize input forcings and parameter files	Linux file system commands	Chapters 7 and 8	
2. Customize connectivity file for specific basins	XDMS and *cellarea* (stand-alone executable); (could also use ESRI or GRASS GIS)	Chapter 9	Not required for unconnected mode
3. Derive basic a-priori SAC parameters, PE adjustment factors, connectivity files, and slope grids	None required; initial data sets are centrally processed at OHD		Default data are delivered with HL-RDHM; re-computation using local data or higher resolution DEM data may be required (particularly for connectivity) but procedures for this are not described in this document.
4. Customize channel routing parameters	*Preprocess.R,, outletmeas_manual. R* (R scripts); *genpar* (stand-alone executable)	Chapter 9	Not required for unconnected mode
5. Create input deck • Define run period • List basins to model • Define values for non-gridded parameters • Define initial states (names of grids or constants) • Define scalar multipliers	Text editor	Chapter 8	

6. Run model	HL-RDHM (*rdhm*)	Chapter 8	
7. Examine time series results	*ICP, stat_q (stand-alone executable)*	NWSRFS Documentation for ICP; STAT_Q User's Manual	
8. Examine gridded inputs/results	*XDMS* or export for use with a GIS using *xmrg2asc, asc2xmrg* (stand-alone executables)	XDMS Documentation	
9. Re-run model/adjust scalars	*rdhm*	Chapters 8, 5, and 9	

Forecast Mode			
Step	**Available Tool**	**More information**	**Comments**
1. Organize input forcings (XMRG PCP, PE, a priori parameter files)	Linux file system commands, Apps_defaults	NWSRFS Manual I.2 UHIX-DHM	
2. Prepare initial state grids	DHM Grid Editor	DHM Grid Editor User Manual	
3. Edit operations table	Text editor	NWSRFS Manual V.3.3 DHM-OP	
4. Run model	NWSRFS DHM-OP	NWSRFS Manual	
5. View time series output	IFP or XDMS	NWSRFS Manual; XDMS Documentation	
6. View gridded output	D2D or XDMS	XDMS Documentation	
7. Make mods	IFP, DHM	NWSRFS Manual	This option is not available in OB7.2; it will be available in OB8.x
8. Re-run model	IFP, DHM	NWSRFS Manual	

7. INSTALLATION

7.1. What is provided with HL-RDHM Release 2.4.2?

Table 7.1 summarizes the HL-RDHM programs and utilities described in this manual and delivered with HL-RDHM 2.4.2.

Table 7.1. List of HL-RDHM 2.4.2 binary executables, scripts, and example input files

File Names	**Description**

Rdhm	Main HL-RHDM executable
Genpar	Program to generate customized routing parameter grids
Preprocess.R,, outletmeas_manual.R, reg.plot.2pan.R	R scripts used to assist in routing parameter estimation
Plot.vel.q.R, sub.vel.q.R	R scripts used to adjust routing parameters during calibration
cellarea	Stand-alone executable to adjust cell areas in the connectivity file to reflect USGS areas
xmrg2asc	Converts XMRG-like files to ESRI style ASCII grid format.
Asc2xmrg	Converts ESRI style ASCII grid format to XMRG-like format.
Rdhm.card, genpar.card, cellareas.card	Example input cards.
DHM-TF examples	Example input cards for DHM-TF operations, include dhmtfexample/abrfc/freq_params.card, dhmtfexample/abrfc/ffg.card. dhmtfexample/abrfc/local_states.card, dhmtfexample/abrfc/local_states_07_2004.card, dhmtfexample/auto_run/read_states_and_forecast.card, dhmtfexample/auto_run/update_states.card, dhmtfexample/oct07/adjannpeaks.card, dhmtfexample/oct07/freqparams.card
Grass scripts	GRASS Scripts for Displaying and Managing HL-RDHM Output Grids (XMRG like)
xg	Load or remove a collection of HL-RDHM output grid files from a GRASS MAPSET. When loading, a pre-defined GRASS color file can also be specified.
xmrgtograss	Program converts any xmrg-like formatted file(s) used with HL-RDHM into GRASS. The program should be run at the GRASS prompt.
xmrgtograss2	Program converts any xmrg-like formatted file(s) used with HL-RDHM into GRASS. The program should be run at the GRASS prompt.
dxmrg	This is a simple wrapper program to run a shell script that will automatically display GRASS grids and automatically loop through sequential time steps at the user's prompting.
dxmrg2	This is a simple wrapper program to run a shell script that will automatically display two GRASS grids (in separate windows) and automatically loop through the same grids for sequential time steps at the user's prompting.

Table 7.2 lists default parameter files provided to assist RFCs in implementing HL-RDHM and DHM.

Table 7.2. List of parameter grids delivered with HL-RDHM 2.4.2.

frz_STXT.gz	peadj_DEC.gz	*rutpix_ROUGH.gz	snow_ALAT.gz
frz_TBOT.gz	peadj_FEB.gz	rutpix_SLOPH.gz	snow_ELEV.gz
pe_APR.gz	peadj_JAN.gz	rutpix_SLOPC.gz	snow_MFMAX.gz
pe_AUG.gz	peadj_JUL.gz	rutpix_SLOPC1.gz	snow_MFMIN.gz
pe_DEC.gz	peadj_JUN.gz	rutpix_SLOPH.gz	
pe_FEB.gz	peadj_MAR.gz	sac_LZFPM.gz	
pe_JAN.gz	peadj_MAY.gz	sac_LZFSM.gz	
pe_JUL.gz	peadj_NOV.gz	sac_LZPK.gz	
pe_JUN.gz	peadj_OCT.gz	sac_LZSK.gz	
pe_MAR.gz	peadj_SEP.gz	sac_LZTWM.gz	
pe_MAY.gz	*rutpix_ALPHC.gz	sac_PFREE.gz	
pe_NOV.gz	*rutpix_BETAC.gz	sac_REXP.gz	
pe_OCT.gz	*rutpix_DS.gz	sac_UZFWM.gz	
pe_SEP.gz	*rutpix_Q0CHN.gz	sac_UZK.gz	
peadj_APR.gz	*rutpix_QMCHN.gz	sac_UZTWM.gz	
peadj_AUG.gz	*rutpix_ROUGC.gz	sac_ZPERC.gz	

With the exception of the grids marked with an asterisk, all grids contain a-priori parameter estimates covering CONUS. Grids marked with an asterisk should be filled in using local customization as described in Chapter 9. These grids contain only default values except for some study areas where OHD has done R&D. Default values for these grids are:

rutpix_ALPHC: -1 (nodata)
rutpix_BETAC: 1
rutpix_DS: 2.5
rutpix_Q0CHN: -1 (nodata)
rutpix_QMCHN: 1.333
rutpix_ROUGC: -1 (nodata)
rutpix_ROUGH: 0.15

HRAP-scale connectivity files are provided for CONUS RFCs. These files are named
xxrfc4k.con
where xx is a two character RFC specific identifier. The rutpix_SLOPC, rutpix_SLOPC1, and rutpix_SLOPH grids are also RFC specific in that there are separate files covering the domain of each RFC. These grids are delivered in separate subdirectories named by RFC.

HRAP scale flow direction and flow accumulation grids are also provided for each RFC. These grids are used by XDMS to graphically display the cell-to-cell connectivity.

These files are in the XMRG-like grid format and the filenames are:

hrapfa10.xmrg
hrapfd10.xmrg

The XDMS documentation tells the location where these files should be placed.

7.2. System requirements

To run the pre-compiled executable, the recommend hardware requirements are Intel Pentium or newer processor and 512 MB of memory. The recommended operating system is Red Hat Enterprise Linux 4.0. It does not necessarily have to be an AWIPS OBx machine

7.3. Recommended location for input XMRG files

The relationship between where the *rdhm* executable is run and the XMRG forcing files are stored can have a big impact on performance due to I/O and network considerations. This has the biggest impact in simulation mode when many files are being read. For a re-run using 1D-XMRG files, this issue is less important. HL-RDHM accesses XMRG rainfall grids in the following way:

If the XMRG files are compressed with gzip (.gz) or are already uncompressed, *rdhm* reads and processes the file directly. If the XMRG files are compressed with the Unix compress utility (.Z), HL-RDHM does the following:

1) copies the compressed .Z XMRG file to the /tmp directory
2) decompresses the file in the new location (/tmp)
3) reads and processes the file
4) deletes the file
5) repeats the process for the next XMRG file.

Performance tests with an earlier version of HL-RDHM indicated much faster execution in simulation mode when the XMRG radar data files were located on a drive that was local to the CPU running HL-RDHM, instead of being accessed over a network. To determine which directories are local to the Linux workstation, you can execute the *df* Linux command. All local directories are on the /dev file system.

8. RUNNING THE MODEL

8.1. Program Execution

The program has three options: `--help`, `--version`, and `-I <inputdeck>`. If run without option, the default is `--help`. In the last option, the `-I` is optional.

- With `help` or without option, it prints the following message:

Command line options:

-v [--version]   *print version string*

--help   *produce help message*

-I [--input-deck] arg   *Input Deck filename*

Input Deck:

--time-period arg   *simulation time period in ISO format*

--time-step arg   *time step in HH:MM:SS.XXXX format*

--connectivity arg   *the connectivity filename*

--window-in-hrap arg   *Simulaion window in HRAP, left-x,*

     right-x, bottom-y, top-y

--pixel-size-hrap arg (=1)   *pixel size in hrap unit*

--subwindows arg   *list of subwindows, subwindow id*

     followed by left-x, right-x, bottom-y,

     top-y

--output-path arg   *the output directory*

--input-path arg   *the input directory*

--input-path-priority arg   *the input directory priority, higher*

     valuepath will be searched before lower

     ones

--ignore-1d-xmrg arg (=0)   *whether use/read/create 1d xmrg,true or*

     false, default is false

--calibration arg   *the calibration algorithm*

--calib-time-period arg   *the calibration period*

--operations arg   *list of operations*

--output-grid-before-timeloop arg   *grid data to be output before timeloop*

--output-grid-inside-timeloop arg   *grid data to be output inside timeloop*

--output-grid-after-timeloop arg   *grid data to be output after timeloop*

--output-grid-step arg (=1)   *output grid in number of timesteps*

--output-grid-last-step arg   *data grid to be output at the last step*

--output-timeseries-basin-average arg   *timeseries data averaged over every*

                                                                       basin

--output-timeseries-basin-outlet arg   *timeseries data at the outlet*

--window-input arg   *input data for the whole window*

--input-data arg   *basin id followed by name=value pair*

--mask arg   *the mask layer, XMRG format*

--user-data arg   *other data needed by specific models*

--observed arg   *the observed stream flow time series in*

                                                           OHD data card format

--timescale-interval arg   *select timescale interval options*

--calib-parameters arg   *parameter name followed by name=min,max pair*

--DSINT arg   *list of soil layers*

--DSINTW arg   *list of soil layers water*

--user-comments arg   *these go to the timeseies file*

--normalize-soil-moisture arg (=0)   *option ot normalize soil moisture"true"*

                                                - normalize or "false" - not normalize,

                                                default is false

--disaggregation arg   *Input forcing ID followed by one of*

                                                disaggregation method, uniform, linear,

                                                vector. Default is uniform. for example

                                                disaggregation = pe=linear,xmrg=uniform

--obvgrids arg   *Input forcing ID followed by the*

                                                filename, frequency, and start time of

                                                the day of the grid. Default is to use

	the DataDescription defined by the
	particular model.
--fcstgrids arg	*Input forcing ID followed by the*
	filename, frequency, and start time of
	the day of the grid. Default is to use
	the DataDescription defined by the
	particular model.
--simple-forecast arg (=0)	*is this a real forecast or a*
	simple-forecast run "true" or "false",
	default is false, ignored for
	calibration mode
--global-output arg (=0)	*whether or not to output nested(global)*
	basin time series "true" or "false",
	default is false
--warnings-off arg (=0)	*turn off warnings or not "true" or*
	"false", default is false

- With `--version`, print the version information

```
Reseach Distributed Hydrologic Model, version 2.4.2
```

- With `-I <inputdeck>`, run simulation using the given input deck

8.2. Input deck

The input deck is a plain text file that contains entries defined as a key word plus an equal sign (=) followed by values for this entry. There is no particular order required for the entries. The input deck entries can be categorized into five groups according to their functionalities, (1) run setup, (2) inputs, (3) outputs, (4) calibration, (5) real time run, (6) PE disaggregation, and (7) other.

See Appendix B for an example input deck.

8.2.1. Run setup

This category has three entries: time period, time step, and operations.

time-period

specifies simulation time period in the ISO standard format of YYYYMMDDTHH. The simulation steps include both the start and the end time, i.e. [start, end]. For example, the first simulation has a time period of [19980526T00, 19980527T00]. States are saved at the last time step of the first simulation. To resume the first simulation using the saved states, the second simulation could be [19980527T00, 19980528T00].

```
time-period  = 19980526T00 19980531T23
```

Note that simulation time stamp will be rounded backward to the nearest input grid when the time-period and time-step are not synchronized.

For real time run, three formats could be used.
I. time-period = 'start time' 'end time' 'switch time', where 'switch time' is the time at which to read forecast forcing instead of observed. For example,
```
    time-period = 19941231T00 19950101T12 19950101T003300
```
II. time-period = '*days before switch time' 'days after switch time' 'switch time'. The '*' indicates this is relative time period. For example,
```
    time-period = *2 2 19950101T003300
```
III. time-period = '*days before current time' 'days after current time'. The '*' indicates this is relative time period. In this case the switch time is missing; the default switch time (current system time) is used. For example,
```
  time-period = *2 2
```

time-step
simulation time step, format is HH:MM:SS.XXXX
```
time-step  = 6:00:00
```

operations

list of operations (models) to be run, they will be performed in the order specified here. Current available operations are:
- `sac` for the SAC_SMA runoff model
- *calsac* for SAC-SMA model with calibration
- `snow17` for the Snow17 model
- `calsnow17` for the Snow17 model with calibration
- `snow17_mcp3` mimics NWSRFS version of SNOW17. In this version liquid snow water do not physically refreeze but model keeps track on negative snow heat during cold periods.
- `calsnow17_mcp3` the calibration version of snow17_mcp3

- *frz* for the frozen ground model
- *api* for the Continuous Antecedent Precipitation Index (CONT-API) model
- *rutpix7* for the Kinematic Wave Routing Model
- *calrutpix7* for Kinematic Wave Routing Model with calibration
- *rutpix9* for the Kinematic Wave Routing Model using
- rating curves to derive routing parameters
- *calrutpix9* for Kinematic Wave Routing Model using rating curves to derive routing parameters with calibration
- *func_opt* for the calibration objective function
- *monthlySum,* The monthlySum operation reads xmrg formatted hourly precipitation data and generates grid files containing monthly total precipitation.
- xmrgAdjust, The xmrgAdjust operation reads xmrg formatted hourly precipitation data, monthly PRISM grids, and monthly MPE grids, and outputs adjusted hourly precipitation grids in xmrg format.
- AnnPeaks. AnnPeaks keeps track of annual maximum peak discharges for each cell and stores a grid of annual maximum peaks at the end of each water year in the simulation.
- FreqParams, Before the time loop, read the annual maximum peak grids generated from the AnnPeaks operation, compute the frequency distribution for each cell and write the distribution parameters (logmean, logstd, and wlogskew) to HL-RDHM memory (the PixelGraph object) for possible use in other programs. Inside the loop, convert discharge to a frequency.
- getMaxRet, When HL-RDHM is run in forecast mode, the getMaxRet option may be used to compute the maximum discharge and return period grids produced during the forecast period.
- Rutpixloc9, This algorithm allows routing in individual grid cells but no routing to the next downstream cell.
- localFFG, The FFG in each grid cell is estimated through repeated FFG guesses to match a target threshold probability level.

Example:

```
operations = sac rutpix9
```

This runs the sac model followed by rutpix9 for each time step.

NOTE:

snow17 source code was adjusted to account for the latest Eric's changes to the lumped NWSRFS version. The new code actually has two versions of snow model which differ only on the treatment of refreezing liquid water in snow pack:

1. operation 'snow17_mcp3' that mimics NWSRFS version of SNOW17. In this version liquid snow water do not physically refreeze but model keeps track on negative snow heat during cold periods.

2. operation 'snow17' that explicitly accounts for liquid water refreezing.

In many cases, simulation results are rather close from the both operations, and they are the same if snow holding capacity (parameter PLWHC) =0. However, in cases of prolong snow accumulation periods with frequent returning cycles of melting-freezing, 'snow17' usually accumulate more SWE at the end of accumulation period than 'snow17_mcp3'. SWE difference depends on PLWHC value and on melting-freezing cycle frequency.

8.2.2. Inputs

The HL-RDHM can be run in two modes, connected, and unconnected. When run in connected mode, a connectivity file is given. Therefore, runoff from each pixel can be routed. While run in unconnected mode, the flow connectivity of the grid cells is either unknown or irrelevant. Therefore, there is no connectivity file. Only an HRAP rectangular bounding box is given. In this case, no routing can be performed. This mode is particularly useful for a water balance and soil moisture modeling applications.

Connected Mode Entries

connectivity:

The connectivity file name with full path, this should not be used when a window is given, for example, `connectivity = /fs/hsmb5/abrfc4x4.con`

Unconnected Mode Entries

window-in-hrap:

The simulation window in HRAP units, this should not conflict with "connectivity". There are 4 HRAP coordinates for this option, left (west boundary), right (east boundary), bottom (south boundary) and top (north boundary), and they must be provided in this order. Please note that the window includes all the boundaries. For example,

`window-in-hrap  = 500   503   300   303`

has 4 x 4 = 16 pixels if the pixel size is 1 HRAP unit, the most west coordinate of the window is 500, the most east coordinate is 504, the most south coordinate is 300 and the most north is 304.

pixel-size-hrap:
pixel size, default is 1 HRAP unit. If the connectivity also specifies a pixel size, the pixel size in the connectivity file has a higher priority.

pixel-size-hrap = 0.5

subwindows:

The subwindows inside the window defined above, it is meaningful only when a window is defined. This option has the same format as the `window-in-hrap` option except that there is a subwindow identifier before each subwindow bounding box. For example

subwindows = A 500 501   300   301
subwindows = B 502 502   302   302

The above entries define two subwindows. Subwindow B has only 1 pixel.

Mask:

The mask XMRG file excludes pixels having negative values. It is only meaningful when window is defined. The resolution of the mask must be coarser than or equal to the pixel size defined for the window. If the mask layer is finer than the window resolution, multiple values will be retrieved from the mask for each pixel in the window, and a random value will be used.
Example:

```
mask = /fs/parameterslx/sac_LZFSM
```

window-input:

The input data for the entire simulation window, which consists of "name=value" pairs. Note that there is no space before and after the "=". For example,
window-input = sac_LZFPM=-2.0 sac_UZFWM=-.5e-5
window-input = sac_PCTIM=+3.

Entries Applicable to Both Modes

Entries listed here are applicable to both connected and unconnected modes.

input-path:

List of input paths. *rdhm* will search in these paths for input data, such as precipitations, soil parameters, and states. However, it will not recursively search subdirectories in these paths. For example,

input-path = /fs/parameterslx
input-path = /fs/arkansas/nasa_pcp

For every input forcing, *rdhm* will search all directories for the XMRG file at every time step. In most cases, the input XMRG files are put in the same directory contiguously in time. As the program progresses, it is most likely that the XMRG forcing will be found in the same directory as the previous on. To improve performance, the program remembers the last found input directory for each input forcing. Then for the following searches, the program will check this directory first. If a forcing grid is not found, it will continue searching the list of input directories; otherwise the search is done. If multiple files are found in the directory list, the program will print a warning message and the last found file is going to be used.

In addition, specifically for the 'snow17' operations, one of the required XMRG forcings, 'psfrac', will be missing for the entire simulation period. It will be wasting time to search for it in the whole list of input directories and print out the 'psfrac' missing messages. To turn off the search and warning message for 'psfrac', the user can add adding the 'psfrac=9' to the 'input-data' entry for each basin or subwindow in the input card. Here, the number 9 can be replaced by other numbers but it has to be greater than 1.

input-path-priority:

This option sets the priorities of the input paths. This is used when there are multi-sets of input data, and the user wants to specify a particular set of data should be used. When this option is not set, the program searches the entire list of paths for data and remembers the last found directory. For the next search, the search will start from the last found directory until the first file is found. When this option is set for each input path a priority level, the paths with higher priorities will be searched before the last found directory. Thus the data in higher priority paths will be used. Please note that higher priority values mean higher priority. If one of the paths is set a priority value, all other paths should be set a value also. Otherwise, priorities should be omitted for all paths.

input-data:

The input data of each basin or subwindow, which consists of repetitive occurrences "name=value" pairs following the basin or window ID. For subwindows, the values specified in "window-input" for the entire window will be overwritten.

The program looks for the basin ID in the connectivity file which then determines which cells are modeled.

For example,

input-data = TEST1
input-data = sac_LZFPM=-2.0 sac_UZFWM=-.5e-5 sac_PCTIM=+3.

ignore-1d-xmrg:

This option sets the switch that turns on or off the 1D XMRG feature. The value is either true or false and is optional. The default value is false. If it is true, the program will ignore 1D XMRGs even they exist and will not create any new 1D XMRGs. For example,

ignore-1d-xmrg = true

DSINT:

This option sets depths of soil layers for soil temperature inputs. Default values are 0.1, 0.4, 0.6, 0.75, and 1.5, 5 layers in total. The unit is in meters. For example,

DSINT = 0.15 0.3 0.6 0.75 1.5

DSINTW:

This option sets user defined depths of soil layers for soil moisture and temperature outputs. Default values are 0.1, 0.4, 0.6, 0.75, and 1.5, 5 layers in total. The unit is in meters. For example,

DSINTW = 0.15 0.3 0.6 0.75 1.5

8.2.3. Outputs

Output rules – outputs are only created for the domain of the model run. Users select which model variables to output using entries in this category. The model variables could be parameters, states or intermediate results. Appendix A contains a complete list of currently available model variables that can be output for each model. Every variable could be output as a time series or an XMRG grid, according to user's selection. If a variable is selected to be output as a time series, the variable is either averaged over each sub-basin/sub-window, or the value at the basin outlet pixel is output for each time step according to the user's selection. If an XMRG grid is selected for output, the variable could be output at three specific run points, 1) *before time loop*, 2) *inside time loop*, and 3) *after time loop*. If *inside time loop* is chosen, the XMRG grids will be output at every particular number of time steps which is specified in the `output-grid-step` entry described below.

Table 8.1 summarizes grid output options Table 8.2 summarizes of time series options.

Table 8.1 Summary for grid output

Output Point Entry	Entry
Before Loop	`output-grid-before-timeloop`
Inside Loop	`output-grid-inside-timeloop`
After Loop	`output-grid-after`-timeloop
Last Time Step	output-grid-last-step

Table 8.2. Summary for time series output

Type Entry	Entry
Basin Average	`output-timeseries-basin-average`
Basin Outlet	`output`-timeseries-basin-outlet

output-path:

This entry specifies the output directory with full path. For example,

```
output-path  = /tmp/abrfc
```

output-grid-step:

This entry gives a number. Output grids of selected variables output every 'n' time steps inside the time loop. Thus it is possible to output grids at regular time intervals coarser than the model time step. The default value is 1. For example,

```
output-grid-step  = 2
```

Grids will be output at every other time step.

output-grid-before-timeloop:

This lists variable names to be output as grids before time loop. It is optional. For example,

```
output-grid-before-timeloop = sac_LZFPM sac_PCTIM
```

output-grid-inside-timeloop:

This lists variable names to be output as grids inside time loop. It is optional. For example,

```
output-grid-inside-timeloop  = xmrg
output-grid-inside-timeloop = surfaceFlow subsurfaceFlow
```

output-grid-after-timeloop:

This lists variable names to be output as grids after time loop. It is optional. For example,

```
output-grid-after-timeloop  = xmrg surfaceFlow
output-grid-after-timeloop  = subsurfaceFlow
```

output-grid-last-step:

This lists variables/states to be output as grids at the last simulation step. It is optional. For example,
```
output-grid-last-step = uztwc uzfwc
output-grid-last-step = lzfsc lzfpc adimpc lztwc
```

output-timeseries-basin-average

This lists variables/states averaged by basin to be output as time series. It is optional. For example,
```
output-timeseries-basin-average = xmrg
```

output-timeseries-basin-outlet

This lists variables/states at basin outlet pixel to be output as time series. It is optional.
```
output-timeseries-basin-outlet = discharge
```

normalize-soil-moisture

This option sets whether or not to output soil moisture as relative value. A value of 'true' means the output soil moisture values are relative, 'false' means not relative values.

global-output:

By default, when output basin average time series, the values are averaged over the sub-basin (local) which doesn't include the upper nested sub-basins. Sometimes the average value over the entire basin (global) including upper nested sub-basins is desired. When this entry is 'true', this global averaged time series is turned on.

Note: When operations 'calsac' and 'calrutpix7/9' are selected, you can only output model parameters, precipitation and discharge. All other variables are not available for output because these two operations skip some time consuming steps to obtain better performance.

8.2.4. Calibration

This category provides options for the automatic calibration functions.

calibration:

This selects the calibration algorithm. When it is set, *rdhm* will run in automatic calibration mode, otherwise, it runs in simulation mode. Only one optimization algorithm can be specified in this version. Also, in calibration mode, only one basin can be calibrated. That is, you must specify data for only one basin in the `input-data` entry. For example,

```
calibration = sls
```

calibration-time-period:

This sets the calibration time period, which could be different from the simulation period specified in the `time-period` entry. The calibration-time-period is used by the calibration objective function to calculate criteria. It uses the same format as the `time-period` entry. For example,

```
calibration-time-period  = 19980526T00 19980531T23
```

observed:

This sets the observed time series (the full path including the file name should be defined here), which is used to calculate the calibration objective function. It must be a text file in OHD Data Card format. For example,

```
observed  = /fs/hsmb5/hydro/rms/doc/statq_test_sim.dat
```

timescale-interval:

This sets the time scale intervals in hours, which are used by the calibration objective function to calculate multi-scale criteria. For example,

```
timescale-interval  = 24 240 720
```

```
With the above entry, the criteria will be estimated as a
combined error of statistics from 24 hrs, 240 hrs, 720 hrs,  and
the simulation time interval defined by 'time-step' entry.
```

calib-parameters:

This lists the names of parameters to be calibrated with their lower bound and upper bound in the format of "`parameter_name=lower_bound,upper_bound`". Please note that there is no space before and after the '=' and ','. You can specify multiple parameters to be calibrated separated by spaces. Note that these values are multiplication factors instead of actual parameter values. For example,

```
calib-parameters = sac_UZTWM=0.1,1.0 sac_UZFWM=1.0,2.5
calib-parameters = rutpix_Q0CHN=0.5,1.5
```

Calibration starts from the initial parameter set defined in 'input-data' statements.

Note: To perform calibration faster, operations 'calsac' and 'calrutpix7/9' should be selected instead of 'sac' and 'rutpix7/rutpix9'.

8.2.5. Real Time Run Mode

The program recognizes the 'Real Time Run Mode' by looking at the 'time-period' option format. If it is one of the three formats listed in 8.2.2, the program knows that this is a real time run. The following additional entries are need for real time run model.

obvgrids

This entry specifies how to resolve the observed forcing grids. The default observed forcing grid file name is the variable name, such as, 'xmrg' followed by a time string, such as 0520200702z. This entry lets the user specify a different variable name and time string format for the input forcing. The syntax is,

```
obvgrids = <input name> filename=<grid file name convention>
freq=<HH:MM:SS> start=<HH:MM:SS> resolution=<in HRAP> nodata=<NO
DATA VALUE> relativecoord=<true|false>
```

<obvered grid file name convension> is the file name for the given input data. Acceptable syntaxes are,
    1) <any char>yyyyMMddhhmm<any char>
    2) <any char>MMddyyyyhh<any char>
    3) <any char>MMddyyhh<any char>
'resolution' is the grid cell resolution in HRAP unit. 'relativecoord' is whether the XMRG file using relative HRAP coordinate. Some XMRG or XMRG like file use absolution HRAP coordinate. However, some XMRG file's coordinate change's as resolution changes. 'nodata' is the no data value in the forcing grid.

For example,

```
obvgrids = xmrg filename=EMMOSAICM60yyyyMMddhhmmz freq=0:15:00
obvgrids = start=0:00:00 resolution=0.25 nodata=-999
relativecoord=true
```

by using this specification, the program will read the precipitation grid, EMMOSAICM60yyyyMMddhhmmz, instead of the default precipitation grid, xmrgMMddyyyyhhz.

Note: the entry is applicable to normal calibration runs.

fcstgrids

this entry specifies how to find and read the forecasting forcing grids. The syntax is,

```
fcstgrids = <input name> filename=<grid file name convention>
issuingfreq=<HH:MM:SS> fcstfreq=<HH:MM:SS>start=<HH:MM:SS>
resolution=<in HRAP> nodata=<NO DATA VALUE
>relativecoord=<true|false>
```

Forecasted grids have two frequencies, issuing frequency and forecast frequency. The <grid file name conversion> syntax is the same as 'obvgrids' except an extra term is needed to specify the order of the forecast at the particular time. The extra term should be specified using the C standard printf syntax, such as 'f%02d' for a two digits integer. Other items are the same as observed grids.

For example,

```
fcstgrids = xmrg filename=mpn15_yyyyMMddhhmmf%02d ssuingfreq=0:15:00
fcstgrids = fcstfreq=0:15:00 start=0:00:00 resolution=1
fcstgrids = nodata=-999 relativecoord=false
```

simple-forecast

The entry specifies whether this is a real forecast run or a simple forecast run. In the simple forecast run, the program will not read forecast data before the switch time even there are forecast data available before the switch time. The value is either 'true' or 'false'.

For example,

```
simple-forecast = true
```

Note: The order of each item in 'obvgrids' and 'fcstgrids' must be followed exactly. That means the order of each item is not interchangeable.

8.2.6. PE Disaggregation

The PE disaggregation, disaggregation, lets the user disaggregate input forcing when the simulation time step is smaller than the input forcing time step.

1. Syntax
The 'disaggregation' OPTION has the following syntax:

disaggregation = <variable name> interpolation=<interpolation method>

Where the <variable name> is the input forcing name defined by the specific operations. For example, 'xmrg' defined in the 'sac' operation, 'ped' defined in the 'sacped' operation, 'tair' defined in the 'snow17' operation.

<interpolation method> is either 'uniform', 'linear' or 'vector'.

If 'vector' is selected, a list of vector values must be given using the following format:

interpolation=vector(value1,value2,...,valueN-1, valueN)

where value1, value2,...,valueN are positive numbers. The number of values equals to (grid frequency) / (simulation time step). For example, for daily grids, if simulation time step is 6 hourly, then 24/6 = 4 values are needed. If more then 4 are given, the rest are discarded. The interpolated values are calculated as:

grid value * (value[i] / (value1 + value2 + ... + valueN))

For 'linear' interpolation, the values are linearly interpolated between current grid and previous grid. For example, for daily grids, the hourly values of May 5th, are interpolated using grid values of May 4th and May 5th.

The 'start=<HH:MM:SS>' and 'freq=<HH:MM:SS>' in the 'obvgrids' option (see 8.2.5) specifies a time in the format of hours:minutes:seconds. The 'freq=' option gives the frequency, or time step of the data, which should be greater than 0. It's assumed that potential evaporation and precipitation are cumulative values per time interval, but temperature is an interval average value. The 'start=' option gives the data's start time in a day, which ranges from 0 - 23 hours. For instance, 'freq=3:00:00 start=0:00:00' means that the data grids are 3 hourly, starting from midnight, that is data grids appear at hour 0, 3, 6, 9, 12, 15, 18, and 21.

Note that there is no space between the equal signs, '=' after 'interpolation', freq', and 'start'. The example,

```
disaggregation = ped interpolation=uniform
```

says that using the uniform interpolation method for 'ped'. 'ped's are daily grids. They are saved at every day midnight.

In summary, to run 'uniform' and 'vector', only one grid is needed for each time step, that is the accumulated grid for current time step. However, to run 'linear' option, two grids are needed, the previous grid and the current grid. For example, if you have daily grids, and you run 6 hourly simulation, to use 'linear' for simulation period of one day (19950102T00 to 19950102T23), two grids are needed, the grid of previous day, 19950101, and the grid of 19950102. Please see section B on how to create these grids using the XMRG naming convention.

2. Gridded Forcing Data Naming Convention

HL-RDHM uses the same naming convention for gridded forcing data as the XMRG precipitation.

XMRG filenames look like this 'xmrgMMDDYYYYHHz'. For example, for hourly precipitation, 'xmrg0501199901z' is the accumulated precipitation of the first hour in ay 1st, 1999, that is from 0:00 to 1:00am on May 1st, 1999.

Similarly, for 2 hourly precipitations, 'xmrg0501199902z' is the accumulated precipitation from 0 to 2am on May 1st, 1999.

Please pay particular attention to XMRG filenames when the date changes. For instance, the XMRG file, 'xmrg0501199900z', is the accumulated precipitation of the last hour of previous day (April 30th, 1999), although the filename is May 1st, 1999.

Using this logic, for daily precipitation, the filename 'xmrg0502199900z' is the accumulated precipitation of May 1st, 1999, although the filename says May 2nd, 1999.

When you are creating your customized input forcing data, you should follow the same convention.

8.2.7. Others

There currently are two other entries in this category, comment and `user-data`.

(Comment)

Everything after the "# " sign is a comment. For example,

```
# this is a comment line
```

user-data:

This entry provides extra data needed by model developers. A convenient function is provided to access these data. You should not use this entry unless the model developer tells you to do so. For example,

```
user-data = user-data could by any thing,
user-data = every thing here is
user-data = read in an array of C strings
```

warnings-off:

By default, when a forcing is missing at a particular time step is missing, the program prints a warning message by saying that the forcing is missing at this time step. This behavior can be turned off by setting this entry to "true".

8.3. Where to Find Outputs

The output files are put in the output directory specified by the *output-path* entry of the input deck. These files have two formats: time series and gridded data. Time series output files have a suffix of `.ts`. Gridded data are further grouped into two categories, static that do not change with time and temporal that change with time. For static variables, such as parameters, the grid filename is the same as the variable name and they are put in the root of the output directory. For temporal variables such as states and precipitation, the grid filename is the name of the variable appended by a time string. The time string uses the same format as that of the XMRG file name. The *rdhm* also creates a subdirectory whose name is the variable name for each output temporal variable and puts the corresponding grids in that subdirectory.

The gridded data are in the XMRG grid format that is a binary file. The time series data are in OHD Data Card format that are ASCII text files.

Note that because the XMRG time string uses a different convention from the input deck, the time in the XMRG file time string is 1 hour later than the time in the input deck. For example, if you are looking for the first hour of May 1st, 1998, in the input deck, then you need look at the XMRG whose file name is `xmrg0501199801z` instead of `xmrg0501199800z` as shown in Figure 8.1. All temporal gridded data in the HL-RDHM
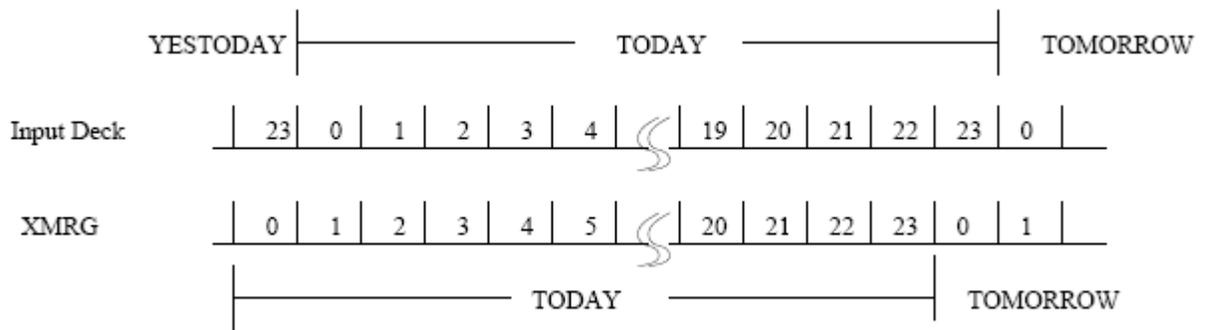
use the XMRG naming convention.

Figure 8.1. Time in the XMRG time string vs. time in the input deck.

9. PARAMETER SPECIFICATION AND CUSTOMIZATION

Use of *rdhm* with a connected domain in a new area requires local customization to develop reasonable initial routing parameters. This requires several steps including:

(1) identifying the HRAP outlet coordinates for the basin or basins of interest (9.1)

(2) adding outlets of interest to the connectivity file (9.2)

(3) analyzing USGS flow measurement data to estimate outlet parameters (9.3)

(4) using the genpar program to generate spatially variable routing parameters (9.4)

Users may also wish to customize parameter grids through manual editing or other procedures in a GIS software. Therefore Section 9.5 describes how to export and import *rdhm* parameter grids to and from a GIS compatible format.

9.1. Determining the correct HRAP coordinates for outlet points of interest

A few simple procedures in XDMS can facilitate outlet selection. These procedures are summarized here. More extensive documentation of XDMS is provided separately. For knowledgeable GIS users, the same information can be obtained using ESRI or GRASS GIS.

The key is that information such as streams and basin boundaries can be displayed along with the cell flow direction data so the correct HRAP outlet cell can be selected. This is necessary because the latitude/longitude coordinates recorded for a gage are not precisely consistent with digitized stream lines (RF1) and the derived HRAP flow direction grids. This means that in some cases simply converting latitude/longitude values for a gauge location into HRAP coordinates may not provide the correct outlet HRAP cell. Figure 9.1 shows an example of one of these cases. In most cases, incorrect outlet HRAP coordinates will become obvious when HL-RDHM is run because the program will associate an incorrect number of cells with the basin and the drainage area will be wrong.

Features of XDMS that are useful to idenfity the correct HRAP outlet cell for a basin:

- Overlays → Flow Direction (displays data from hrapfd10.xmrg)

   It is best to be zoomed to the basin or basins of interest before adding this overlay. A line network connecting cells based on the flow direction grid is displayed. The numbers plotted represent the flow direction (1 = E, 2=SE, 3=S, 4=SW, 5=W, 6=NW, 7=N, 8=NE).

- Overlays → Flow Accumulation (displays data from hrapfa10.xmrg)

- Overlays → Basins

   The level of detail and accuracy of your basin boundaries and rivers can make a difference in some situations. This is where use of a GIS software may be advantageous.

- Overlays → Rivers

- Locator → Basins

- Control → Zoom

- Use the middle mouse button to display of the cursor location in latitude/longitude and HRAP coordinates.

- Use the middle mouse button to query values of the flow accumulation overlay. This is useful in determining if a cell is on a main channel or tributary.

- Display → A Priori Routing Parameters → Orig SLOPC

  This displays the rutpix_SLOPC grid. The middle mouse button can then be used to the query the channel slope value for any grid cell.   This slope estimate is useful for implementing the channel shape routing parameter method described in Section 9.3.

Figure 9.1. Arkansas River at Lamar, CO, gauge reported lat/lon (green triangle) fallsclose to the center of HRAP cell (451, 387), but cell (451, 386) should be selected as the outlet to model flows in the main channel with HL-RMS. Note that the green lines connect the center points of HRAP cells and black dots are shown at center points for selected cells.

9.2. Customize the connectivity file
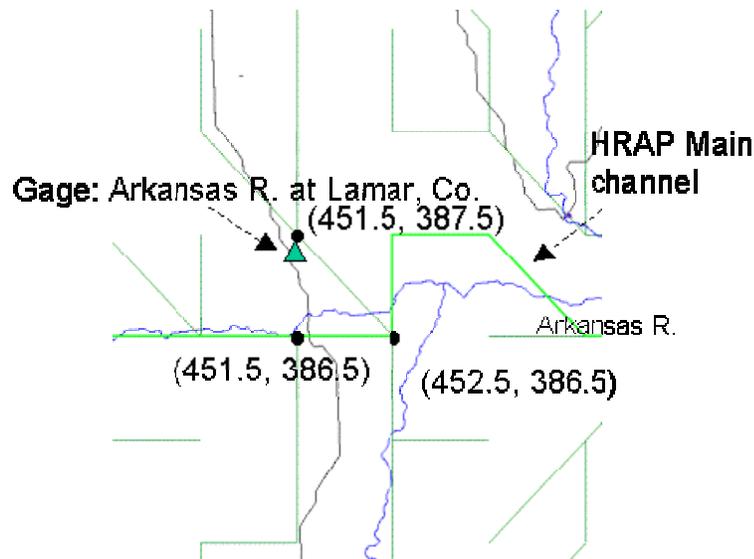
9.2.1. Adding outlets to the connectivity file

The *rdhm* and *genpar* input decks allow the user to specify basins for which time series data will be output. These basins must be identified in the connectivity file header. Figure 9.2 shows the top portion of an example connectivity file displayed with *nedit* (the option to show line numbers is turned on).

```
example.con - /fs/hsmb5/hydro/rms/sequence/

File   Edit   Search   Preferences   Shell   Macro   Windows              Help

/fs/hsmb5/hydro/rms/sequence/example.con 2067241 bytes            L: 5  C: 13

    1 TEXT_SEQ
    2 NUM_HEADER_REC    8 36259
    3 COL    337
    4 ROW    163
    5 LLX   364.0000
    6 LLY   261.0000
    7 URX   700.0000
    8 URY   423.0000
    9 DXY     1.0000
   10 DATA_HRAP
   11 ELDO2       1   32064      6.3911   625.0 347.0
   12 ELMSP       1   31181      5.6069   636.0 357.0
   13 KNS02       1   31663      5.9090   627.0 355.0
   14 SAVOY       1   31125      8.2608   635.0 352.0
   15 TAL02       1   32088      6.2750   623.0 346.0
   16 TEST1       1   31998      6.2179   625.0 350.0
   17 TIFM7       1   29913      6.4359   627.0 366.0
   18 WTT02       1   31573      4.6412   630.0 353.0
   19        0   -5 Rv   1 184 162   6.0089   548.0000   261.0000
   20        1   -3 Rv   1 181 162   6.0123   545.0000   261.0000
   21        2    4 Rv   1 136 154   6.0598   500.0000   269.0000
   22        3    5 Rv   1 133 155   6.0558   497.0000   268.0000
   23        4    7 Rv   1 136 155   6.0549   500.0000   268.0000
   24        5    7 Rv   1 134 155   6.0560   498.0000   268.0000
   25        6    9 Rv   1 134 156   6.0528   498.0000   267.0000
   26        7   11 Rv   1 135 155   6.0568   499.0000   268.0000
   27        8   12 Rv   1 134 159   6.0408   498.0000   264.0000
   28        9   13 Rv   1 134 157   6.0495   498.0000   266.0000
   29       10   14 Rv   1 137 155   6.0551   501.0000   268.0000
   30       11   15 Rv   1 135 156   6.0502   499.0000   267.0000
```

Figure 9.2. Example connectivity file.

The connectivity file contains 10 lines of general header information, followed by a list of basin outlets defined for the model, followed by an ordered list of pixels covering the entire modeling domain.

Line 2: Lists the number of outlets identified in this file (8) and the total number of cells listed in the file (36259)

Lines 3, 4: Number of columns and rows in the domain.

Lines 5, 6: Lower-left X and Y coordinates.

Lines 7, 8: Upper-right X and Y coordinates.

Line 9: Model cell size (1 = HRAP, 0.5 = half HRAP, etc.)

Lines 10-18: Basin outlet specifications. Information included in each line:

Col. 1: ASCII identifier. Must be the same as identifier used in HL-RDHM input deck with the 'input-data' option.

Col. 2: Ignore this column.

Col. 3: Cell line number associated with the outlet of this basin

Col. 4: Representative area (mi^2) of cells in this basin. Computed as the USGS area divided by the number of HRAP cells in the basin. Program does not currently use this value so any floating point number can be here.

Col. 5: hrapx coordinate of outlet cell

Col. 6: hrapy coordinate of outlet cell

Lines 19-36278: Cell List

Col. 1: Cell line number

Col. 2: Cell line number of the next downstream cell

Col. 3: Ignore

Col. 4: Ignore

Cols. 5-6: Local reference coordinates used internally by the program (Ignore)

Col. 7: Area represented by the cell in mi^2. Used for routing calculations.

Cols. 8-9: Globally referenced Hrapx and Hrapy (the true hrapx, hrapy coordinates)

To add a new outlet point, the following steps are required:

- Add a new basin outlet specification line (e.g. lines 10 – 18 in the example).

  As defined above, the information required for the added line includes 6 entries: a basin identifier of **eight** characters or less, a dummy value of 1, **the cell line number in the connectivity file corresponding to the basin outlet**, a representative cell area, and the HRAP x and y coordinates of the outlet cell for this point. These values should all be entered on one line in free format, separated by at least one space.

  To determine the cell line number corresponding to a new outlet point, do a text string search on the connectivity file for the true hrapx and hrapy coordinates of the point you are adding (note the number format must be precise and match entries in the cell list). For example, search for "630.0000 353.0000" to identify the outlet line number for WTTO2 in the above example. The search should find "630.0000 353.0000" in Columns 8-9 somewhere in the Cell List (lines 19-36278). The value in Column 1 at this cell line number is the third number in your basin specification line (e.g. 31573 for WTTO2 here). The fourth number, a representative cell area, can be taken from column 7 of the cell line number found in your search.

  The program uses the cell line number in the third column to index the cells and ignores of the hrapx and hrapy coordinates. Therefore, when adding a new basin outlet cell to the header, make sure the cell line number matches the corresponding entry in the full cell list.

- Increment the value defining the number of outlets in line 2 (8 in this example) by 1.

9.2.2. Adjusting cell areas in the connectivity file to reflect USGS defined drainage areas

The HL-RMS algorithms use the drainage areas defined for each cell in the connectivity file (cell list lines, Column 7) to convert rainfall-runoff depths to volumes. The program sums the individual cell areas to get basin area estimates. When the basic connectivity files are first generated, the drainage areas for each cell are computed based on the definition of the HRAP coordinate system. Cell area estimates vary depending on the latitude of the cell center point. Because of the coarse resolution of the HRAP cell network, it is sometimes possible that the total area estimate for basins from the raw connectivity file can be significantly different from the USGS estimate of the drainage area (on the order of 10% or more in some cases, particularly for basins smaller than 1000 km^2) (see Reed (2003) for more details). The reason for this is not just that curves in the basin boundaries cannot be precisely captured by a fixed origin, coarse resolution grid, but also that a single HRAP cell may contain the confluence of multiple stream tributaries. If one of these tributaries is downstream of the gage and another upstream, then it is not possible to select an HRAP cell in the cell connectivity network that includes one tributary but not the other.

For situations when significant discrepancies between the USGS drainage area and the drainage area derived from the raw connectivity file exist, a utility program called *cellarea* is provided that will create adjust the cell areas in the connectivity file. In this initial release, the *cellarea* utility is a rather inflexible program; however, it serves its purpose. The syntax for running the program is simply:

cellareas <control_file_name>

An example control file is shown here:

> *input-connectivity = /fs/hsmb5/hydro/rms/sequence/areaadjust/abrfc_var_adj.con*
> *output-connectivity =*
> */fs/hsmb5/hydro/rms/sequence/areaadjust/abrfc_var_adj2.con*
> *basin-area = POWELL=365*
> *basin-area = LANAG=619*
> *basin-area = CAVESP=90*
> *basin-area = SSILOAM=49*
> *basin-area = SLOA4=719*
> *basin-area = SAVOY=433*
> *basin-area = ELMSP=247*

The first two lines specify the input and output files. The output file is a new connectivity file with adjusted cell areas. The data lines are indicated by the key word

"basin-area", the basin identifier (must be from 5 to 8 characters in length), and the local USGS area for the basin in km^2.

IMPORTANT NOTE: For basins containing sub-basins, only the local area should be entered into the control file if the subbasins are also included in the same control file. In other words, rather than entering the total area (km^2) reported by the USGS for a large basin with nested/interior forecast points, the sum of areas for nested subbasins should be subtracted from the total basin area and that value should be entered in the cellarea control file. If your control file includes a parent basin but not the subbasin, then the areas for cells in the subbasin will be overwritten. If you want the subbasin area to be overwritten, you should enter the total area for the parent basin and don't include the subbasin in the control file. If you add a new basin to the control file that is a subbasin of an existing parent basin, the area of the existing parent basin in the control file should be changed to reflect only the local area. It is a good idea to keep all defined basins in one control file.

9.3. <u>Analyze flow measurement data to determine at-a-station routing parameters</u>

9.3.1. Overview

Two methodologies are provided for deriving routing parameters. The first one is the rating curve method, which entails quantifying the discharge-cross section relationship (Equation 3.3). The second is the channel shape method based on the relationship between cross-section area and channel width. Once this relationship is established, it is used by the distributed model to derive the discharge-cross section relationship for internal calculations. This section describes tools provided to estimate at-a-station routing paremeters. The next section describes a tool to generate distributed routing parameters using these derived outlet parameters and geomorphological relationships.

The steps to estimate at-a-station routing parameters from flow measurement data are implemented using R scripts. R is an open-source statistical package that can be downloaded from www.r-project.org. The scripts have been tested against R 2.4 and they only require the R base package.

Deriving routing parameters involves the following procedures: i) downloading raw USGS cross-section and flow measurement data; ii) preprocessing (optional); iii) running R script to plot data, calculate weighted regression relationships deriving the at-a-station parameters.

9.3.2. Download Flow Measurement Data

(1) Go to http://water.usgs.gov/nwis

(2) From the main page: Click Surface Water. Click Measurements. At this point there are many search options to identify the stations you are interested in. Choose whichever

option you prefer. The final product that you need to download is a Text file in **Tab-Separated** format. Also, the date format is YYYY-MM-DD. The top of the file that you obtain should look something like the following (Note: Data lines are cut-off on this example. Actual data lines contain many more columns.). One can download multiple stations as a single file and use the preprocessing script described in the subsequent section to generate separate files for each station.

```
#
# Surface water measurements
#
# Further descriptions of the columns and codes used
can be found at:
#
http://waterdata.usgs.gov/nwis/help?output_formats_hel
p#streamflow
#
# Stations in this file include:
#  USGS 07337000 Red River at Index, AR
#
#
site_no    agency_cd measurement_nu measurement_dt
      party_nm
15s  5s    5s    19d   12s  12s  12s   12s  12s  12s  12s
07337000   USGS 62    1952-07-11      k/m  5.40 1.03 0.43
      1.68
07337000   USGS 63    1952-07-23      w    5.80 1.33 0.72
      1.70
07337000   USGS 64    1952-08-06      w    35.0 38.5 0.79
      2.07
07337000   USGS 65    1952-08-19      w    9.50 2.96 0.39
      1.70
07337000   USGS 66    1952-09-03      g    7.60 2.14 0.34
      1.69
```

The measurements relevant to routing parameter derivation are i) discharge ('discharge_1'), ii) width of wetted area of a channel ('channel_width_va'), iii) wetted x-section area ('xsec_area_va'), and iv) measurement quality flag ('measured_rating_diff'). i), iii) and iv) form the basis for the rating curve method and ii), iii) and iv) are needed for the channel shape method.

(3) Run Preprocess.R (optional)

     If you choose to downloaded measurement data for multiple gauges in a single file, the script "preprocess.R" can be used to take the multi-gauge data as input and generate a separate file for each individual gauge. Before running the script, one needs to first edit it to set correct values to run control variables. Three variables require change:

i) the name of the directory that contains input files (given by variable "dir.in")
ii) the input file name ("file.in"); and
iii) the output directory ("dir.out").
Then one can run it in by issuing the following command at linux/unix prompt:

R CMD BATCH preprocess.R

This generates output files under designated output directory. Each file is named after the gauge number and has an extension of ".d".

9.3.3. Data analysis

Data analysis is done interactively using an R script, "outletmeas_manual.R" (This script calls another script called "reg.plot.2pan.R.") This script reads in a file that contains measurement for a single station, and produces initial estimate of routing parameters via both rating curve and channel shape methods. The user can further refine the parameters by adjusting the discharge-dependent, and measurement quality-dependent weighting factors. For a user, there are four steps involved in the estimation process and the details follow below.

(1) Modify the user inputs.

a. Input directory name, which is stored in variable "dir.in". A default is the following:

#--
#---(1)--- input file list
dir.in<-"../flow_data/Processed/"

Change the right hand side to the actual directory locationon your system.

b. Assign discharge weights.

Discharge weights allow a higher emphasis on high flow measurements for the regression models used. This is particularly important for the common case where there are many low flow observations and few high flow observations. It is desirable not to give to many similar observations and we also want to consider higher weights to higher flows that are associated with flooding. Discharge weights (w) are exponents.

The weights assigned to each measurement point in the regressions is a function of the flow and a user specified exponent.

$$w_q = \frac{Q^a}{\sum Q^a} \qquad (9.1)$$

As seen below, parameters are estimated using relationships between discharge (Q) and cross sectional area (A), cross sectional area (A) and width (B), and Manning's n and Q

(Qwt.n). A different exponent (a) in Equation 9.1 can be applied depending on the relationship being considered. Different exponents for weights may be specified in the Q-A, A-B, and Manning's n relationships using the variables Qwt.qa, Qwt.ab, and Qwt.n respectively. The default exponents equal to 1 are shown below. An exponent of 0 would mean equal weight for all data points. Note that in R the symbol <- is used to assign values to variables.

> *#---(2)--- user specified weight exponent for regression*
> *Qwt.qa<-1 # for Q-A*
> *Qwt.ab<-1 # for A-B*
> *Qwt.n <-1 # for Manning's n*

One can increase the value of a weight to produce better regression fits at higher flows, and vice versa.

c. The user can also assign weights to measurements based on the data quality flag. The default is to assign equal weights. An example of non-equal weights is shown below:

> *#---(3)--- User specified relative weights for each of the USGS data quality flags*
> *ws<-c(8,4,2,1,0)*

This expression assigns a weight of 8 to E (excellent), 4 to G (good), 2 to F (fair), 1 to P(poor) and 0 to -1 (missing). The factors are normalized later in the script so that the relative weights assigned to each quality flag would be 8/15, 4/15, 2/15, 1/15, and 0 in this example.   One may reset the numbers in according to their understanding of the data quality.

     If both discharge and data quality weights are specified, the weights are combined in such a way that the weights of all points still sum to one.

d. plotting options.

> *#---(4)--- graph options*
> *plot_quality=TRUE*
> *new_graphics=T*

"plot_quality" is for specifying whether to plot the measurement quality. If it is set true (T), measurement quality flags will be shown and different colors will be given to data points with different quality flags. If false, then data plots will be shown as open circles.

"new_graphics" is a binary variable. If set to false (F), two sets of regression lines will be shown on graphics devices 3 and 4, one based on the previously specified weights and one with the current set of weights. Dotted lines are the previous set of results and solid lines are the current set. If set to true (T), then all graphics are refreshed and only the current set of regression lines (based on the current set of weights) are shown on all plots.

e. Slope needed for the channel shape method.

> *#----(5)--- slope*
> *slope=0.002*

As shown above, a default of "0.002" is given.

d. output file name

> *#--- (6)--- output file*
> *file.out <- "../routing_params/" #final comparison*

The final parameter are saved in a file for future reference.

(2) Run R by typing "R" at the linux/unix prompt (this is assuming the R is installed in a directory on the default path). This would provide a ">" prompt. At this prompt, type

> source ("outletmeas_manual.R").

This generates five graphic windows (devices 2-6). Device 2 shows the Discharge (Q) vs Area (A) for both rating curve and channel shape methods. Shown below is an example.
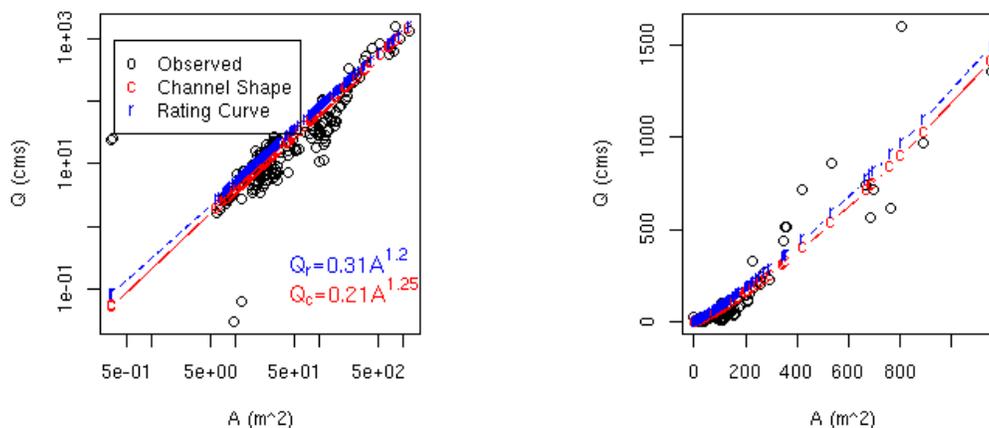
Figure 9.3. Device 2 plots. Predicted discharge via the rating curve and channel slope methods vs. cross-section area, in logarithmic (left) and arithmetic scales (right).

Device 3 shows the regression results for observed Q vs. A on logarithmic scale (left panel) and arithmetic scale(right panel), with data quality shown in each panel if quality_flag is set to true. An example is shown below.

Rating Curve Method 80-07
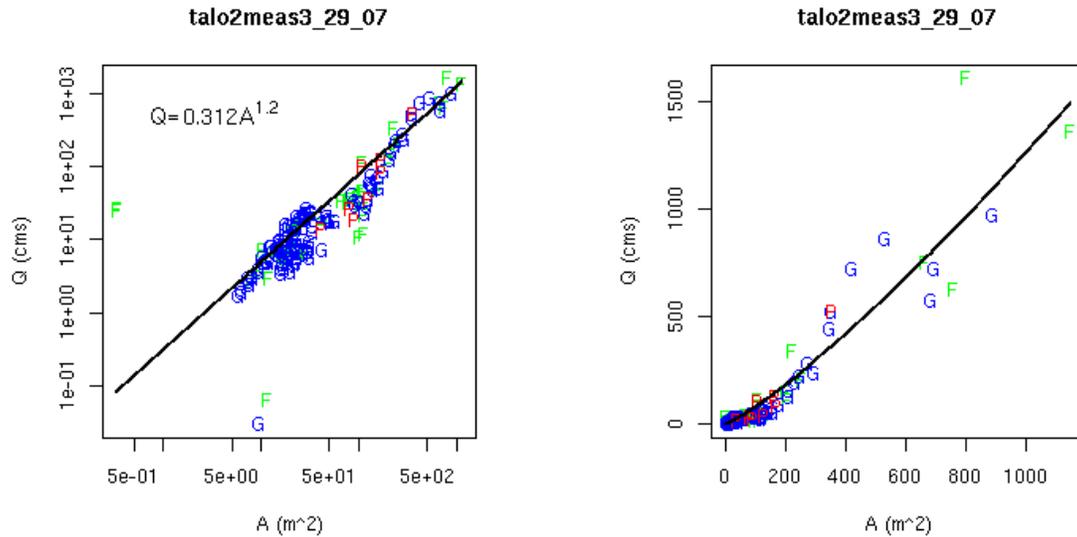
Figure 9.4. Device 3 plots. Regression results for the rating curve methods. Shown are discharge vs. cross-section area in logarithmic (left) and arithmetic scales (right).

Device 4 is similar to 3 except that it is showing the results of regressing Area(A) aginst Width (B). An example is shown below.

Channel Shape Method 80-07
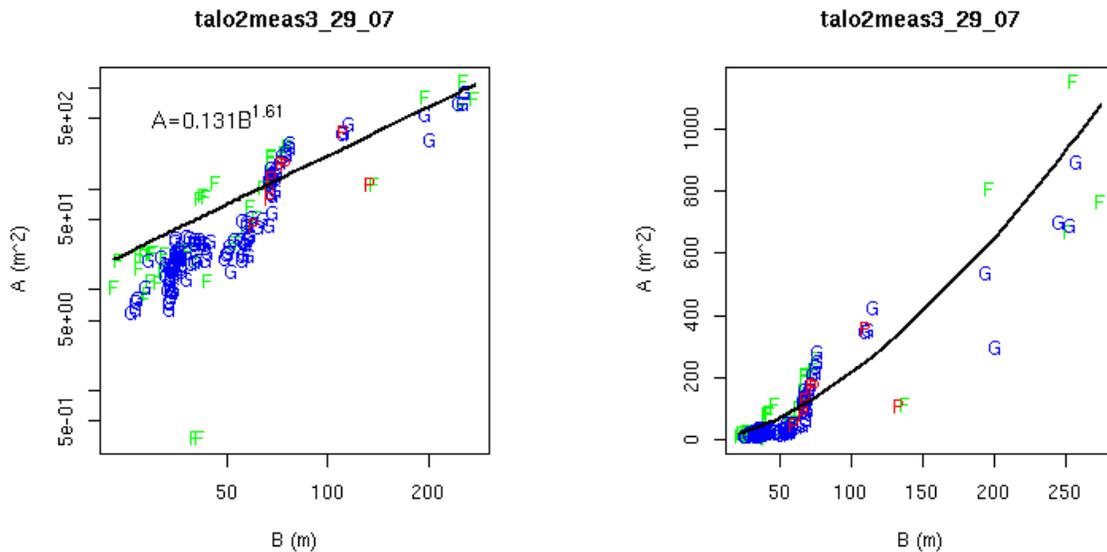
Figure 9.5. Regression results for the channel shape methods. Shown are cross-section area vs. width (of wetted portion of a channel) in logarithmic (left) and arithmetic (right) scales.

Device 5 shows the Manning's n versus discharge and the weighted average value (with weigths determined by Qwt.n).

Manning's n vs. Q: talo2meas3_29_07

n= 0.09

Q (cms)
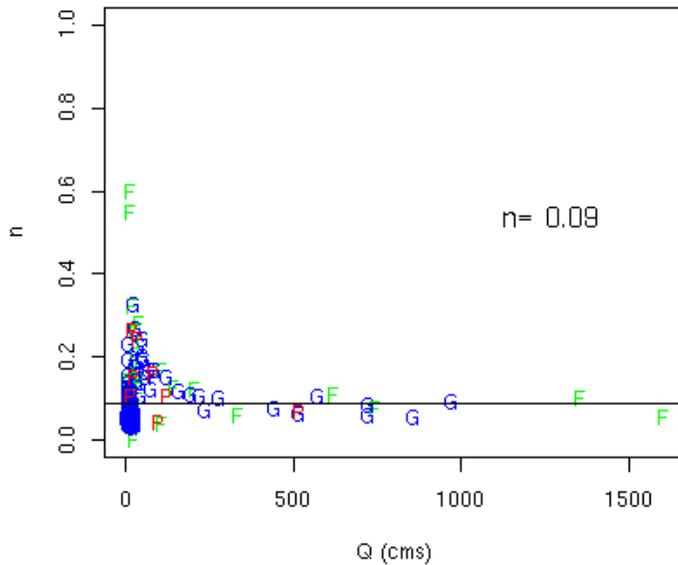
Figure 9.6. Manning's n vs. discharge. The weighted average of n is superimposed as a straight line.

Device 6 shows the width (B) vs. height (H) plot for channel shape method.

H vs. B in Channel-Slope Method

$B = 5.72 H^{1.65}$

B (m)
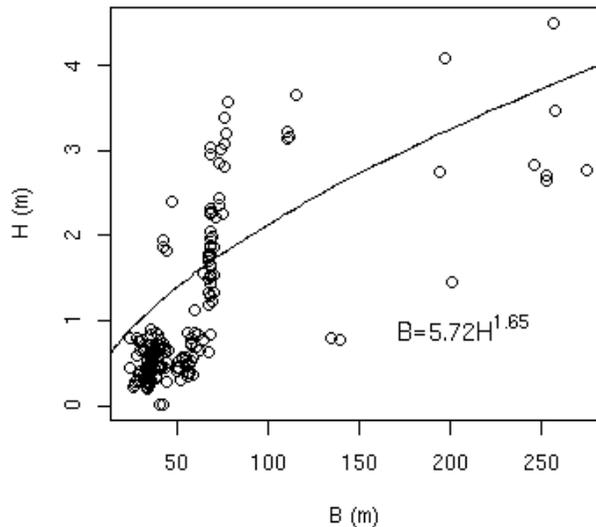
Figure 9.7. Width vs. stage. Superimposed is the predicted width via the channel width method.

(3) If one is satisfied with the results, exit R by typing
> Quit()

74

The coefficients for the Q-A relationship (Equation 3.3) and the B-H relationship (Equation 3.2) that are required for the rating curve and channel shape routing methods respectively, can be found in the specified output file. The example output file shown below contains all the information required for *genpar* with the exception of A_o which is explained below.

```
-----------------------------------------------
q0chn  qmchn   alphac betac  n
0.312  1.2       5.72   1.65   0.09
 -----------------------------------------------
```

Q0chn and qmchn are rating curve-based routing parameters. "alphac" and "betac" are channel-shape based routing parameters. "n" is the Manning's n. Another value needed for the channel shape method to generate distributed alphac values is a representative cross-sectional area at the outlet (Ao). The exact value for this parameter is not critical. To get a reasonable value for Ao, it is helpful to list the n values and corresponding A values for the station. Then, simply select one of the A values corresponding to high flows and the representative Manning's n value from above (0.09 in this case).

To do this in R type:

```
> a_narray=rbind(data.sub$A,data.sub$n)
> write(file="a_n.txt",ncol=2,a_nbind)
```

Two columns are written to the file (A and n). The values should be sorted in order of increasing A values. To determine a reasonable value for A_o, scroll down to the bottom of the file and pick a relatively large value corresponding to the weighted mean value for "n" determined above.

In the case where results are unsatisfactory, stay in R, and edit "outletmeas_manual.R" to change the weighting exponents. By increasing the weighting exponents, one can achieve better match at higher flow rates. After making the changes, type source("outletmeas_manual.R") to rerun the script. Iterate until satisfactory results are achieved.

9.4. Generate customized routing parameter grids

9.4.1. Overview

The *genpar* program is used to update and localize values in existing HL-RDHM routing parameter grids. The program assumes that all required parameter grid files already exist; however, the values in these grids may all be NODATA or the default values as specified in Table 3.2 and Section 7.1.

Genpar can be used to update only one type of parameter at a time. Multiple runs of *genpar* are required to update multiple parameters. However, values in multiple basins can be updated in one run of *genpar*. The input deck that specifies the basins for which parameters will be updated is very similar to that used for the main *rdhm* program. The file *genpar.card* provided is an example input deck used to generate routing parameter grids.

9.4.2. How to run

The *genpar* program has three command line options: `--help`, `--version`, and `-I` `<inputdeck>`. If run without option any options, the default is `--help`. For the last option, the `-I` is optional. The basic syntax is simply:

Genpar <input_deck>

By typing *genpar –help* or simply *genpar*, the following message detailing the input deck options is printed:

9.4.3. Input Deck Specifications

Like *rdhm*, the input deck entries in genpar.card file are specified with the following syntax:

Input_entry = parameter_value

The input deck entry options include the following. These are listed at the terminal when you simply type genpar or genpar –help.

Table 9.1 Genpar input options

No.	Input Key Word	Explanations	Format
1	Connectivity	name of connectivity file	string
2	Output-path	The output directory	string
3	input-path	Input location for parameter grid	string
4	overwrite-existing-grid	Whether to overwrite current grid (will ignore output-path if true)	boolean
4	Create-new-grid	Whether to create new grid (mutually exclusive with 3	boolean
5	windows-in-hrap	Boundary of new grid in x_left x_right y_bottom y_top, needed when 4 is true	int int tint int
6	genpar-id	Name of the parameter	string
7	genpar-data	Basin name and parameter values	string int ….

In a single run, **one and only one** of the following 9 parameters strings must be assigned as the genpar-id (#6 in Table 9.1): SLOPC, ROUGC, BETAC, ALPHC, SLOPH, DS, ROUGH, Q0CHN, QMCHN

As an example,
genpar-id=SLOPC

would make *genpar* compute the grid for channel slope. The first argument required for genpar-data (#7) is always the basin name. The rest of the arguments vary in number depending on the parameter being specified (genpar-id entry, #6).

The nature of the *genpar* calculations vary depending on which parameter string is specified. Table 9.2 describes these differences.

<div align="center">Table 9.2 Genpar computational algorithms</div>

Parameter string	Algorithm
BETAC SLOPH DS ROUGH QMCHN	If any of these strings are specified as the "genpar-id", a constant value is assigned to the grid cells in selected basins based on the genpar-data specification (see Table 9.3). In our applications, we have used constant values for all of these parameters except rutpix_SLOPH for which values can easily be computed from the DEM. A pre-computed grid of rutpix_SLOPH is delivered with the HL-RDHM software. Rutpix_DS and rutpix_ROUGH grids are initially assigned default values of 2.5 and 0.15. Default values for BETAC and QMCHN in the grids provided are 1 and 1.333 respectively; however, these values should be overwritten using locally derived values.
SLOPC	Default grids of channel slope are pre-computed at HL and provided to each RFC. In the grids delivered, there are inaccuracies in computing channel slopes, particularly for low order channel cells. Two reasons for inaccuracies are (1) the coarse resolution of the input DEM (400-m resolution in this case) and (2) the difficulty in precisely defining where lower order channels initiate. With the SLOPC genpar-id specified, *genpar* can be used to make adjustments to slopes for low order channels based on an empirical relationship (Strahler, 1964) between hillslope slopes and channel slopes. In the *genpar* routine, the relationship is only applied if the value predicted by the relationship is significantly different from the value computed from the DEM.

	The default arguments listed in Table 9.3 for this parameter are based on the Strahler (1964) relationship.
ROUGC.	Spatially distributed values of channel roughness are estimated using an empirical relationship that is a function of drainage area, slope, and a representative channel roughness (n_o) at the outlet: $$n_i = n_o S^{0.272} F^{-0.00011}$$ (Tokar and Johnson, 1995) n_o and the two constants 0.272 and -0.00011 are specified in the input deck as described in Table 9.3.
ALPHC	ALPHC values are estimated as a function of drainage area, SLOPC, and ROUGC (Koren, 2004).
Q0CHN	Q0CHN values are estimated as a function of drainage area (Koren, 2004).

Table 9.3 provides a summary of the number of arguments needed for each parameter (excluding the parameter name) and the default values.

Table 9.3. Number of arguments required to generate each routing parameter.

No.	genpar-id	# of arguments needed	Arguments	Comments
1	SLOPC	2	0.178 1.23	Use defaults
2	ROUGC	3	n_o 0.272 -0.00011	The user should specify the first argument and use defaults for arguments 2 and 3. The first argument is the Manning's n at the outlet derived as described in Section 9.3.
3	BETAC	1	Betac	Derived as described in Section 9.3.
4	ALPHC	3	*-1 A_o alphac*	Enter -1 for the first argument since it is no longer used. A_o is a

				representative cross sectional area at the outlet (See section 9.3)
5	SLOPH	1	constant	Typically, this option is not needed since reasonable values of SLOPH can be derived from the DEM and are provided.
6	DS	1	Constant	
7	ROUGH	1	Constant	
8	Q0CHN	2	q0chn qmchn	
9	QMCHN	1	qmchn	

As an example, when
genpar-id = Q0CHN

three arguments need to be supplied to "genpar-data". The following expression serves this purpose:

genpar-data = TALO2 0.31 1.2

The first argument "TALO2" is the basin name. Two additional arguments are needed according to Table 9.3. The values 0.31 and 1.2 derived in Section 9.3 are used here.

Here is a complete example of *genpar* input card named "genpar.card".

###
#genpar.card

═══

#enter the connectivity file name
connectivity =
/fs/hsmb5/hydro/users/zhangy/RDHM/Genpar/sequence/abrfc_var_adj.con

#specify an input location for parameter grids
input-path   = /fs/hsmb5/hydro/rms/parameterslx/abrfc

#specificy an output location
output-path = /fs/hsmb5/hydro/users/zhangy/RDHM/Genpar/output

#replace the existing grid or output the grid to the output-path, true or false
#
overwrite-existing-grid = false

#
#create a new grid instead of modify existing grid, the boundary in this
case is the boundary of all selected basins, true or false
#
create-new-grid = true

#
#if the create-new-grid is true, the grid will be created in this window.
#if this window is not consistent with the window from the connectivity,
#the windows are combined into a big window that contains both subwindows.
#
#window-in-hrap =   480   505 298   306
#
Name of the parameter to be created, available names are:
slopc rougc betac alphc sloph ds rough Q0CHN QMCHM
They are case insensitive
#
#genpar-id = slopc
#genpar-id = rougc
#genpar-id = alphc
#genpar-id = betac
#genpar-id = sloph
#genpar-id = ds
#genpar-id = rough
#the next line specifies the parameter for which values will be generated
genpar-id = q0chn

#genpar-id = qmchn

#next line is an example input information for q0chn grid generation
genpar-data = TALO2 0.31 1.2

###

Values for any of the parameter grids listed in Table 3.2 can be updated using *genpar*.
The first seven grids are required to run HL-RDHM using the channel shape method.
Only the last two grids are required for the rating curve method.

9.4.4. Output

If you choose the create-new-grid option, you must replace your previous parameter file
in your *rdhm* parameter input directory path in order for *rdhm* to find the new grid values.

9.4.5. Rules and limitations

1. The order in which parameter grids are generated is important! The rutpix_ROUGC grid and rutpix_BETAC grids must be generated before the rutpix_ALPHC grid because the rutpix_ALPHC grid depends on ROUGC and BETAC. For the same reason, the rutpix_QMCHN grid should be created before the rutpix_Q0CHN grid.

2. Each time *genpar* is executed for a basin, values for all cells upstream of basin outlets specified in the input deck are modified. If a downstream outlet and a nested basin outlet are specified in the same input deck, the inter-basin area between the two gages will be assigned values depending on the downstream outlet parameters and cells upstream of the nested outlet will depend only on parameters specified for the nested outlet. If in a subsequent run, only parameters for a downstream outlet are specified, values for cells inside the nested basin will be overwritten in the *genpar* output file.

3. *Genpar* only updates parameter grid files. Basic template files (provided) must exist in order for the program to work.

9.5. Edit parameter grids using a GIS

Programs to convert HL-RDHM grids to and from a GIS compatible ASCII format will also be provided and documentation will be added here later.

10. CALIBRATION STRATEGY

10.1. Overview

Anderson (2002) and Smith et al. (2003) describe comprehensive strategies for river basin calibration with NWS lumped hydrologic models. Distributed hydrologic model calibration strategies are not as well developed; however, the simple calibration strategies described in this chapter have been shown to produce (1) improved outlet hydrographs relative to a-priori (uncalibrated) runs, and (2) outlet simulations comparable to or better than simulations from lumped models for basins in ABRFC (Koren et al., 2004; Reed et al., 2004). There is also evidence to suggest that efforts to calibrate HL-RDHM-SAC at basin outlets will improve interior point simulations in most cases (Reed et al., 2007). Calibration of distributed models continues to be an active area of research.

As described by Anderson (2002), the river basin calibration process for lumped models can be divided into the following major steps:

River Basin Calibration Steps
1. Gather information and data
2. Assess spatial variability
3. Select flow points and period of record
4. Historical data analysis and processing
5. Calibration of hydrologic models

6. Operational implementation of calibration results

Although some of the details differ, the same basic steps and principles should apply to distributed hydrologic models. This chapter focuses only on Step 5 of the overall process (where many of the differences exist). For Step 2, many data sets needed to assess spatial variability of soil properties, land use characteristics, and a-priori parameters are provided with HL-RDHM (as described in Chapter 7).

With regard to Step 4, here we assume that an archive of reasonable gridded forcing data is available. In most applications of HL-RDHM to date, we have used hourly, HRAP scale precipitation and climatological monthly PE-demand estimates as input. We have also developed hourly, HRAP-scale temperature forcings for Snow-17 applications. Although not without limitations, HL-RDHM model runs using hourly, radar-based, multi-sensor precipitation grids from ABRFC and WGRFC have shown promising results. For winter applications in MARFC and ongoing applications in the Sierra Nevada Mountains, more complex procedures have been used to derive the required gridded forcings for model calibration. As with lumped modeling, it is assumed that the gridded forcings are unbiased and stationary.

A simple strategy that has proven effective in calibrating HL-RDHM (with the sac and *rutpix7* or *rutpix9* models) includes the following basic steps.

Recommended Distributed Model Calibration Steps
1. Obtain the available a-priori gridded SAC-SMA estimates.
2. Derive a-priori routing parameter estimates using local data (see Chapter 9)
3. Calibrate a lumped model for the basin at a 1-hour time step using standard NWS tools and strategies (SAC-SMA, UHG) (This step is recommended but not required.)
4. Derive initial SAC-SMA scalar multipliers for individual parameters as: (lumped model parameter)/(basin average of gridded a-priori parameter values)
5. Iteratively make adjustments to the sac and rutpix7 and rutpix9 scalar multipliers using similar strategies and objectives used for lumped model calibration.

The use of scalar multipliers in Step 5 assumes that the *a priori* parameters are spatially correct relative to each other, yet their average magnitudes may be incorrect. Although there are many reasons why this may not be true in reality, initial experience shows that this technique is an effective way of improving upon *a priori* model simulations. With this approach, the scalar multipliers become the calibration parameters, reducing the dimensionality of the problem to that of a lumped model. Research into more sophisticated parameter adjustment techniques is ongoing.

The automatic calibration techniques described in Chapter 4 are provided to assist with Step 5. Experience to date suggests that some combination of manual and automatic techniques may be the most efficient path to accomplish Step 5 above.

10.2. A Manual Calibration Example

Although manual calibration is possible for distributed models, graphical displays to assist with manual calibration (such as ICP) are not yet tailored to distributed models (although ICP is still useful). The example here describes methods using existing tools. Since a distributed model has longer run times for a single model execution than a lumped model, manual calibration requires more patience (It may take several minutes to complete a multi-year run for a headwater). Therefore, parameter adjustments should be well thought out. This suggests that it is wise to develop some experience with lumped model calibration prior to attempting distributed model calibration. An expert with lumped model calibration may be able to skip lumped model calibration (Step 3 in the above list) and still derive useful results.

The basic calibration steps for a headwater described in Anderson (2002), Chapter 7-1, apply.

    Headwater Calibration Steps
    A. Remove large errors
    B. Obtain reasonable simulation of baseflow
    C. Adjust major snow model parameters, if snow is included\
    D. Adjust tension water capacities
    E. Adjust parameters that primarily affect storm runoff
    F. Make final parameter adjustments

For DMIP 2, the following steps were taken to calibrate a 2484 km^2 basin in Oklahoma, the Illinois River above Tahlequah (TALO2). Only key trials where improvements were made are described here. Note from the calibrator: "Although I have little calibration experience, I found that for TALO2, starting with the lumped model calibration and then going through the steps outlined by Anderson (2002) was fairly straightforward and satisfactory results were obtained."

Tables 10.1, 10.2, and 10.3 summarize statistics from key trials in the calibration process.

    A. Remove large errors

    Trial 1. Ran HL-RDHM using a-priori parameter grids. This includes a-priori PE and PE demand grids.

    Analysis: The average values for a-priori parameters in a basin required for the next step can be found in the 'basin.info' output file. To assess results, outlet hydrographs are plotted using ICP with the PLOT-TS option. We also use the STAT-QME operation to evaluate bias statistics. Although STAT-QME

calculations are based on daily flows, outputs of monthly, annual, and flow interval bias statistics are still useful.

Observations: Compared to lumped model results, a-priori results have a reasonable overall bias, but the seasonal biases are large (Table 10.1). Examination of the hydrographs shows the recessions and baseflows from the the lumped model are much better and the simulations of event peaks are more consistent.

Trial 2. Computed 11 SAC multipliers based on the (lumped/apriori ratio) and re-ran. For this and subsequent cases, I used spatially uniform values for the PE demand based on lumped calibration results.

Analysis: At this time, a spreadsheet or similar tool is used to compute these ratios. After each HL-RDHM run, output files you want to save should be renamed so they are not overwritten.

Observations: Comparing scaled results to the lumped model, overall bias has increased (Table 10.1). This is not surprising knowing that with a distributed model, higher rainfall intensities at the cell scale will not be averaged out as they are in the lumped model. Comparing the scaled results to the raw a-priori results, the baseflow representation is much improved and visual inspection shows that the event simulations are also much improved. The overall, seasonal, and flow interval bias statistics are also improved over the a-priori results.

Trial 3. To compensate for the increase overall bias that comes with a change in scale, I increased both of the tension water storages by about 20%.

Observation: This dropped the overall bias from 8.0 to 2.7%

B. Obtain reasonable simulation of baseflow

Trial 4. Baseflow tends to be high based on the flow interval statistics from trial 3 and visual inspection of the hydrographs using the log scale. Therefore, the LZFPM scale factor was lowered from 2.32 to 2.1 to reduce base flow.

Observation: This change resulted in visual improvements and a better flow interval bias for the lowest three flow intervals in STAT-QME (Table 10.2).

Trial 5. Based on visual hydrograph inspection using the log scale and a 365 day window, it looks like more baseflow is needed during dry periods but not during wet periods. Thus, I increased the PFREE scale factor from 0.14 to 0.3.

Observation: Improvements are seen through visual inspection but the bias statistics do not change much.

c. Adjust parameters that primarily affect storm runoff and routing

Trial 6. Visual inspection of the hydrographs shows that trial 5 under-simulates medium sized events and over-simulates large events, I increased the UZK scalar from 0.52 to 0.7.

Analysis: Impacts of this change were examined using PLOT-TS and also using the STAT_Q program to analyze selected events. To help to distinguish the impacts of UZK, UZFWM, and other changes, separate statistics are calculated for large and medium sized events. 'Large' events were identified as events which are not impacted much by a change in UZK. Medium events are selected as the next 9 highest events with flows of 250 cms or higher in this example. Flood flow is 304 cms in this basin.

Visual examination and statistics in Table 10.3 indicate that this change improved the simulation of medium sized events.

Trial 7. Routing parameter adjustments were made to improve hydrograph timing for events. The goal of the routing parameter adjustment was to slow down the bigger events and speed up (to the degree possible with the current model) the smaller events. New scale factors of 1.8 and 0.92 were chosen for Q0CHN and QMCHN respectively.

Analysis: Two simple R scripts are provided to examine the effects of various scale factors on the wave velocities for different flow levels: plot.vel.q.R (this script calls sub.vel.q.R). Figure 10.1 illustrates the effects of parameter adjustments in this example.
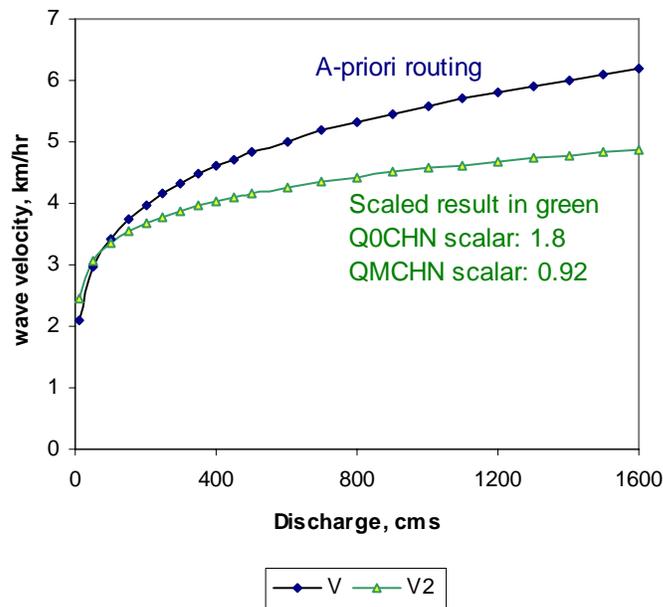
Figure 10.1. Relationship between wave velocity and discharge.

Observation: Event statistics improved from trial 6 to trial 7 (Table 10.3), particularly the Flood RMS.

Trial 8. Visual inspection and the average maximum peak flow ratio statistics indicate that trial 7 results over-simulate large event peaks (average peak ratio is 1.13). The UZFWM scale factor was adjusted from 0.69 to 0.75. .

Trial 9. Final adjustments were made to improve large peaks (increase UZFWM scale factor again from 0.75 to 0.8) and also reduce the overall bias (increased LZTWM scale factor from 0.72 to 0.82).

Observation: Nearly all statistics in Table 10.3 from trial 9 are superior to the best lumped model simulation.

Table 10.1.
Monthly Percent Bias for Different Calibration Trials on TALO2 (STAT-QME)

Month	Apriori (1)	Scaled Apriori (2)	3	4	5	6	7	8	9	Lumped
Oct	26.8	15.0	-0.7	-2.8	3.7	5.0	4.9	5.1	0.6	-3.9
Nov	12.0	6.8	-8.1	-7.2	-7.5	-5.5	-4.6	-3.9	-10.7	-3.7
Dec	10.9	14.1	3.5	5.1	3.9	4.1	3.8	4.3	-2.9	8.8
Jan	1.8	10.9	5.1	5.8	5.4	4.9	5.1	4.7	0.7	7.2
Feb	12.4	2.8	0.3	1.7	1.5	2.5	2.9	2.3	0.0	0.1
Mar	5.8	7.9	5.4	6.7	6.3	6.4	6.2	6.8	5.0	4.4
Apr	-25.0	2.6	0.3	0.0	0.0	-0.7	-0.8	-1.1	-2.6	-4.5
May	-26.4	16.5	14.3	14.3	14.7	15.9	16.1	15.9	14.9	7.4
June	-39.4	3.4	-0.1	-0.4	0.5	0.7	0.4	-0.3	-2.3	-11.9
July	-39.1	-1.5	-3.9	-6.2	-4.7	-6.8	-7.6	-6.8	-7.6	-10.6
Aug	3.2	7.2	4.5	-0.5	5.9	6.2	5.6	4.1	3.5	-22.4
Sep	34.2	24.3	19.9	13.3	26.0	28.8	29.1	29.7	30.6	3.9
Mean	-5.3	8.0	2.7	2.9	3.5	3.9	3.8	3.9	1.0	-0.4
Stdev	23.7	7.0	7.3	6.5	8.4	9.1	9.2	9.2	10.5	9.4

Table 10.2.
Flow Interval Bias for Different Calibration Trials on TALO2

Interval		No of Cases	Lumped	Apriori (1)	Scaled Apriori (2)	3	4	5	6	7	8	9
0.00 -	0.92	NO CASES										
0.92 -	2.87	9	18.5	11.9	42.6	52.4	32.4	37.5	33.2	30.0	28.9	29.9
2.87 -	9.20	550	0.4	-18.1	10.6	5.1	-2.4	5.1	3.0	2.6	2.4	1.2
9.20 -	28.74	993	1.3	-17.7	10.7	2.0	-0.1	1.1	-0.9	-1.1	-0.9	-5.7
28.74 -	91.96	482	2.1	-3.2	10.5	5.6	6.5	6.4	6.0	5.0	6.2	4.2
91.96 -	287.38	91	-0.5	14.3	5.9	2.9	5.3	5.1	9.7	10.7	10.1	7.9
287.38 AND ABOVE		16	-12.5	-12.3	-4.5	-6.5	-3.8	-4.0	-2.5	-0.3	-3.3	-6.7

Table 10.3

Statistics for Large Events (9) from STAT_Q

	Trial:					
	5	6	7	8	9 (final)	lumped
Overall event bias	-0.3	1.4	2.2	0.3	-3.2	-6.5
Flood bias	10.2	10.5	10.9	9.9	9.6	12.6
Flood RMS	50.5	49.2	42.2	38.5	36.4	35.5
Mean max flow ratio	1.15	1.15	1.13	1.08	1.02	0.81
Mean peak time diff (hr)	-1.56	-1.33	-1.00	-0.44	0.33	-0.89
Mean abs. Peak time diff. (hr)	3.11	2.89	2.56	2.00	1.89	2.44

Statistics for Medium Events (9) from statq_lx.exe

	Trial:					
	5	6	7	8	9 (final)	lumped
Overall event bias	-0.626	1.35	1.56	1.13	-0.706	-5.11
Flood bias	16.2	16	16	15.7	15.7	15.5
Flood RMS	33.4	32.4	31.7	31.1	31.6	36.4
Mean max flow ratio	0.9302	0.9566	0.9626	0.9292	0.9015	0.8601
Mean peak time diff (hr)	1.89	0.67	0	1	1	2.78
Mean abs. Peak time diff. (hr)	4.11	3.11	2.89	2.33	2.1	3.44

11. LINKING WITH NEW ICP

11.1. Introduction

The new ICP can be used together with HL-RDHM to streamline the calibration process. Originally, the ICP is design to facilitate lumped model calibrations. Very recently, a special version of ICP has been made so that the HL-RDHM can be run within the new ICP program just the same way as the MCP3 program does. The version that can run HL-RDHM is called new ICP sacht hl-rdhm.

The ICP sacht hl-rdhm has the following capabilities related to HL-RDHM

-Run HL-RDHM
-Edit HL-RDHM input deck
-View the optimized parameter scalars after a calibration run of HL-RDHM
-Automatically insert the optimized scalars into the HL-RDHM input deck.

11.2. Installation

The new ICP sacht hlrdhm is deployed in a tar file, the file name is *icp_testpackage.tar.ob83.hlrdhm.gz.*

First, create a directory which the new ICP sacht hlrdhm will be installed. Then unpack the tar file to the installation directory.

The new ICP sacht hlrdhm requires at least two *Apps_defaults* tokens, '*rdhm_exe*' and '*mcp_decks*'. The *rdhm_exe* token points to the HL-RDHM executable. The *mcp_decks* is the

directory where the program will be looking for input decks. Both HL-RDHM and MCP3 input decks are looked up in the same way. The tokens can be set in the *~/.profile_dir/Apps_defaults* file.

11.3. How to run
1) First go to the new ICP sacht hl-rdhm installation directory, then change to the '*test_scripts/*' directory. For example, '*cd icpnew_hlrdhm/test_scripts*'.
2) Run the '*run_icpnew_userdata*' shell script. The new ICP main window will pop up.
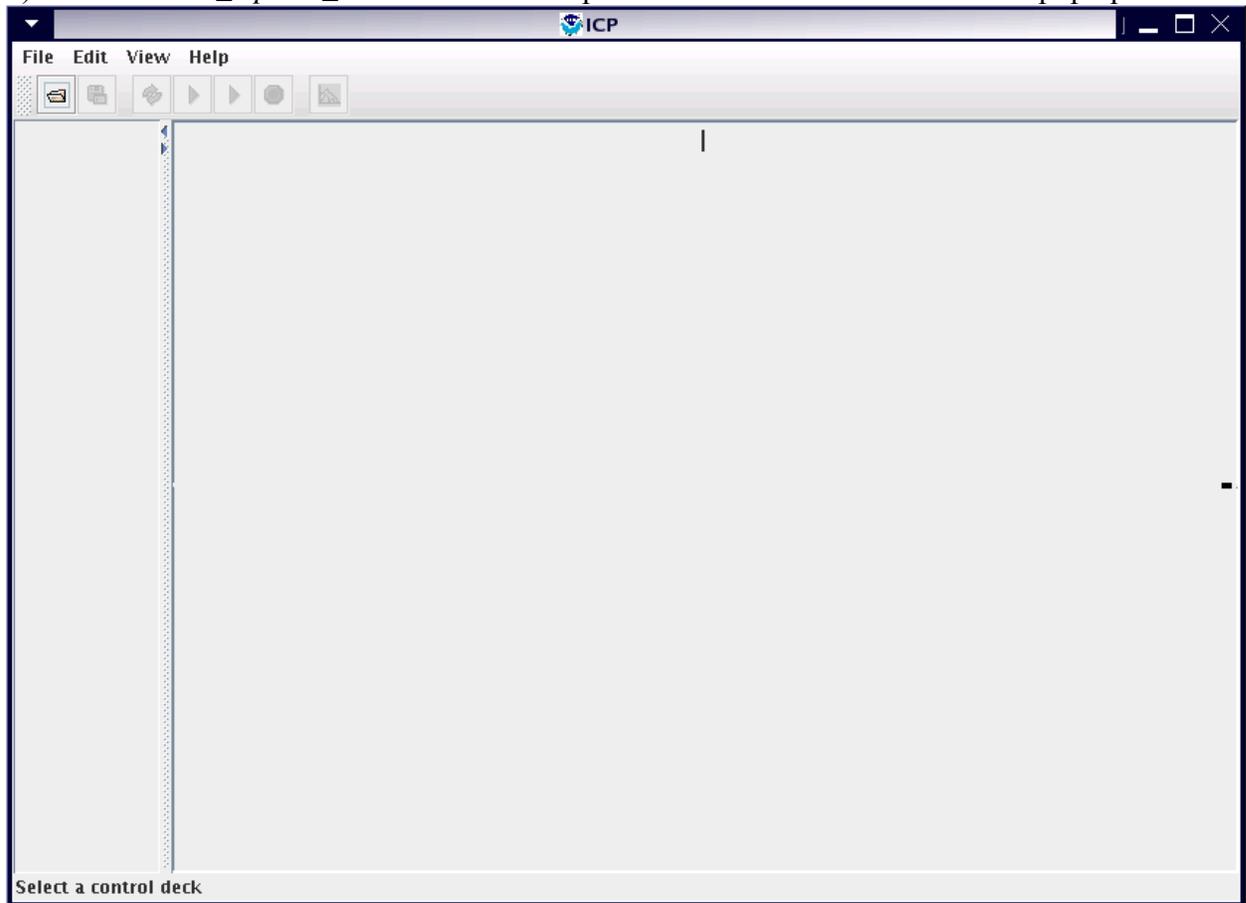
3) Click the *File* menu, select *Open*, the *File Selection* dialog will appear.
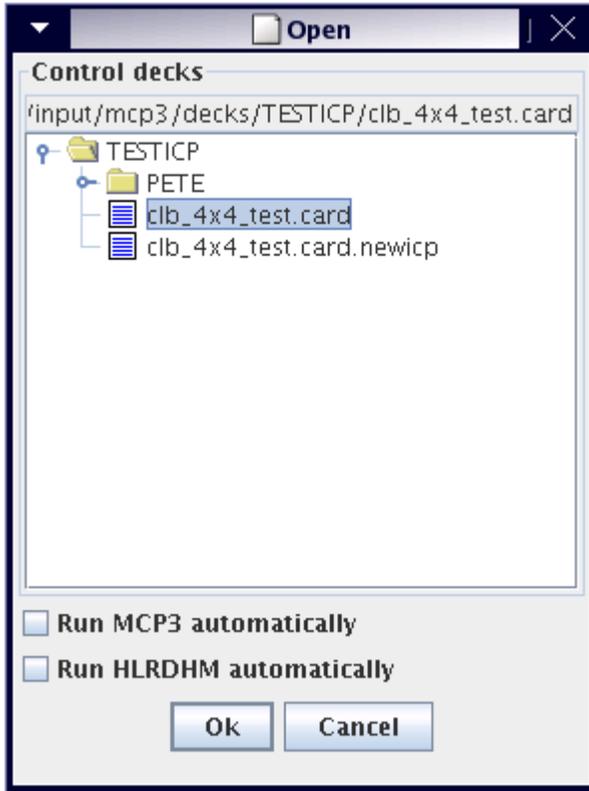
4) Browse the directories and select the input deck. The program will automatically detect whether the type of the selected deck is a HL-RDHM deck or a MCP3 deck. The radio buttons on the bottom can be checked to run HL-RDHM or MCP3 using the selected directly. If you select a HL-RDHM deck and the '*Run MCP3 automatically*' radio button is checked, you will get an error message after click on the '*OK*' button. The same thing will happen if you select a MCP3 deck, but the '*Run HL-RDHM automatically*' is checked

5) If no radio button is checked, after selecting the input deck, the program HL-RDHM or MCP3 can be run using the run tool button (arrow button on the tool bar). The arrow on the left side is for MCP3 and the arrow on the right side is for HL-RDHM.

Screenshot showing ICP window with title "ICP: clb_4x4_test.card" and status bar text:
Deck loaded: /fs/hsmb5/hydro/icpnew_rdhm/calb/input/mcp3/decks/TESTICP/clb_4x4_test.card
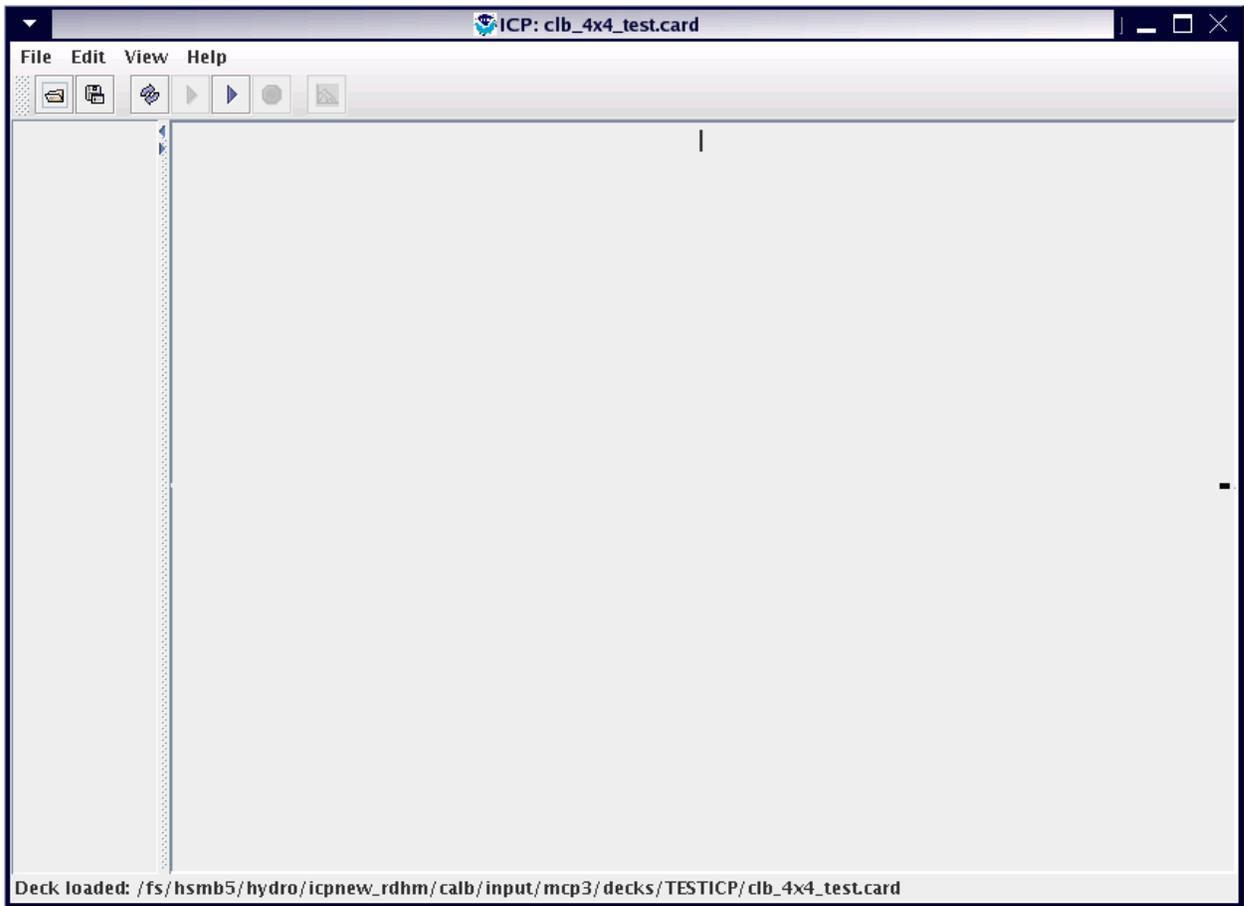
6) At this time, the selected deck can be viewed and edited within the new ICP. Select '*Edit*' from the menu, and then select '*Control Deck*'.

```
┌─────────────────────────────────────────────────────────────────────────┐
│ ▼          ▢ Deck Editor: clb_4x4_test.card        」─ □ ✕ │
├─────────────────────────────────────────────────────────────────────────┤
│ File   Edit                                                               │
│ ┌──┬──┬──┬──┬──┐                                                          │
│ │ 🖾 │ 🖫 │ ✂ │ 🗋 │ 📋 │                                                  │
│ └──┴──┴──┴──┴──┘                                                          │
│ #input deck of gorms                                                  ▲  │
│ #                                                                     ▤  │
│ #simulaion time period                                                   │
│ #                                                                        │
│ #time-period    = 19871001T00 19901001T00                                │
│ #time-period    = 19821001T00 19830901T00                                │
│ time-period    = 19821001T00 19821003T00                                 │
│ #time-period    = 19871001T00 20060930T23                                │
│ #                                                                        │
│ #simulation time step in the format of HH:MM:SS.XXXX                     │
│ #                                                                        │
│ time-step      = 1:00:00                                                 │
│ #                                                                        │
│ #the connectivity file                                                   │
│ #                                                                        │
│ #connectivity =  /fs/hsmb5/hydro/rms/sequence/cars4k_adj.con             │
│ connectivity = /fs/hsmb5/hydro/users/zcui/devl/src/hydrolibtesting/carson4x4.con │
│ #                                                                        │
│ #window                                                               ▼  │
│ ◄│          ▓▓▓                                                     │►    │
│ ┌──┐                                                                      │
│ │  │ Run MCP3 automatically                                              │
│ └──┘                                                                      │
│           ┌──────────┐  ┌──────┐  ┌────────┐                            │
│           │  Apply   │  │  Ok  │  │ Cancel │                            │
│           └──────────┘  └──────┘  └────────┘                            │
└─────────────────────────────────────────────────────────────────────────┘
```

7) After clicking on the run tool button, the new ICP runs the HL-RDHM or MCP3 with the selected input deck.

8) If the selected deck is an automatic calibration HL-RDHM input deck using the SLS algorithm, the calibration results will be displayed in the popup dialog windows.

```
┌─────────────────────────────────────────────────────────────────────────┐
│ ▼                    ▢ Calibration Results:                     」 ✕    │
├─────────────────────────────────────────────────────────────────────────┤
│ ┌───┐                                                                     │
│ │ ? │   Optimum found: icall=167 3365.261475 3365.261475                 │
│ └───┘                                                                     │
│                                                                           │
│         sac_LZTWM 2.750000                                                │
│         sac_LZFSM 2.000000                                                │
│         sac_LZFPM 2.000000                                                │
│         sac_LZSK 2.000000                                                 │
│         sac_LZPK 2.000000                                                 │
│         sac_PFREE 1.975001                                                │
│                                                                           │
│                                                                           │
│         Create a new input deck?                                          │
│                                                                           │
│         New deck will be created with extension .newicp in the same directory. │
│                                                                           │
│              ┌──────────────┐   ┌────────────┐                          │
│              │ Yes, please  │   │ No, thanks │                          │
│              └──────────────┘   └────────────┘                          │
└─────────────────────────────────────────────────────────────────────────┘
```
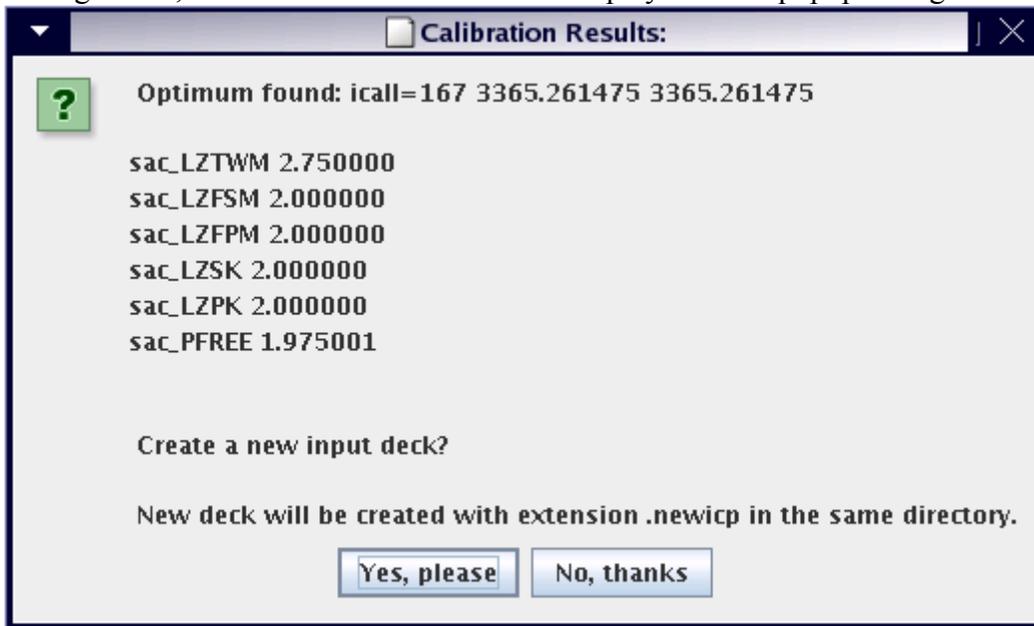
9) In the calibration result popup dialog, you will be asked whether or not to create a new input deck using the calibration results. If '*Yes, please*' is selected, a new HLRDHM input deck will be created; the parameter values listed in the dialog will be used. The new file name is the old file name plus '*.newicp*' in the same directory as the old deck.

10) If the selected deck is not an automatic calibration HL-RDHM deck, the message, '*Simulation finished successfully!*' will be displayed.
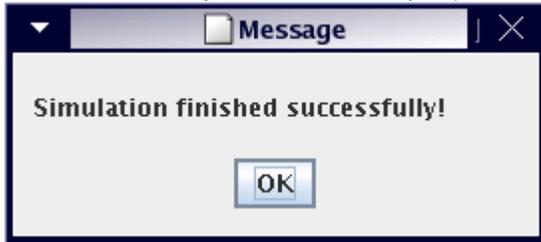
11) An MCP3 deck is needed to display the HL-RDHM simulation results in new ICP. Please see the ICP manual for how to create MCP3 decks. After a MCP3 deck has been created, go to the *File* menu and select the MCP3 deck from the *File Selection* dialog. Then run the MCP3 input deck. The HL-RDHM simulation time series can be plotted by selecting '*View*' and then '*Plot-TS*' from the menu or clicking on the "*View Plot-TS Time Series*" tool button.
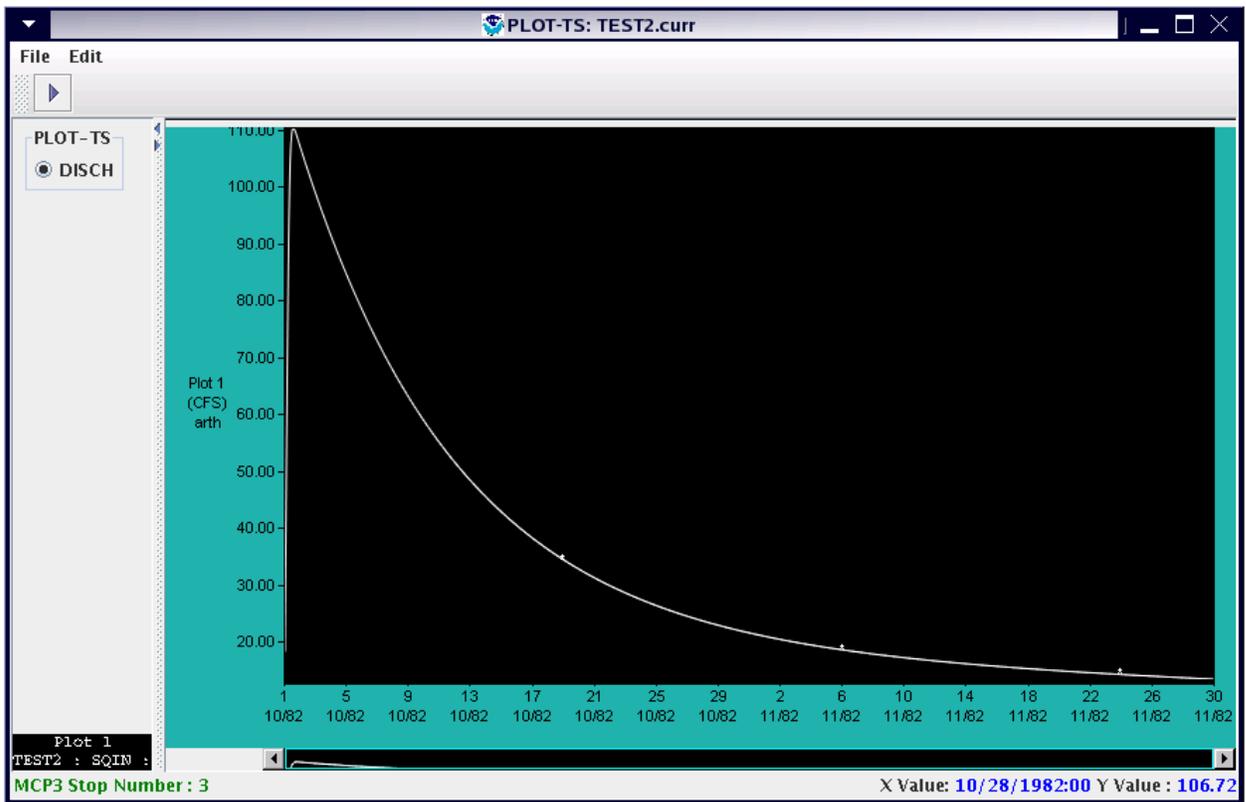
11.4. Limitations

Because MCP3 can not use long file names, for ICP to display HL-RDHM time series, the time series files produced by HL-RDHM have to be renamed to shorter names that are acceptable to MCP3 either manually or using a program.

In addition, please note that the new ICP can not display 2D gridded data created by HL-RDHM.

11.5. Examples

There are two example input decks comes with the installation package for HL-RDHM and MCP3 respectively. The HL-RDHM deck is

<newicp dir>/calb/input/mcp3/decks/TESTICP/clb_4x4_test.card.

The MCP3 deck is

<newicp_dri>/calb/input/mcp3/decks/TESTICP/PETE/TEST2.curr

which is used to display the HL-RDHM simulation results.

12. APPENDIX A. MODEL DATA

Row No.	Data Name	C++ Type	Unit	Thre shol d	Description	Para m (P) Or State (S)	A-priori Grid Provided
Data Available to All Models							
1	Location	HRAP float HRAP	L		HRAP location	P	n/a
2	Representative Area	Float	Mile2		Representative area of this pixel	P	n/a
3	xmrg	float	MM/hr	0	Precipitation	S	n/a
Data Available to SAC-SMA-HT (sac)			0				
4	sac_UZTWM	float	MM	0	SAC SMA parameter	P	Y
5	sac_UZFWM	float	MM	0	SAC SMA parameter	P	Y
6	sac_UZK	float	1/day	0	SAC SMA parameter	P	Y
7	sac_PCTIM	float	DL	0	SAC SMA parameter	P	N
8	sac_ADIMP	float	DL	0	SAC SMA parameter	P	N
9	sac_RIVA	float	DL	0	SAC SMA parameter	P	Y
10	sac_ZPERC	float	MM	0	SAC SMA parameter	P	Y
11	sac_REXP	float	DL	0	SAC SMA parameter	P	Y
12	sac_LZTWM	float	MM	0	SAC SMA parameter	P	Y
13	sac_LZFSM	float	MM	0	SAC SMA parameter	P	Y
14	sac_LZFPM	float	MM	0	SAC SMA parameter	P	Y
15	sac_LZSK	float	1/day	0	SAC SMA parameter	P	Y
16	sac_LZPK	float	1/day	0	SAC SMA parameter	P	Y
17	sac_PFREE	float	DL	0	SAC SMA parameter	P	Y
18	sac_SIDE	float	DL	0	SAC SMA parameter	P	Y
19	sac_RSERV	float	DL	0	SAC SMA parameter	P	y
20	sac_EFC	float	DL	0	SAC SMA parameter - fraction of forest cover	P	n
21	pe_JAN	float	MM/day		mean monthly PE	P	y
22	pe_FEB	float	MM/day		mean monthly PE	P	y
23	pe_MAR	float	MM/day		mean monthly PE	P	y
24	pe_APR	float	MM/day		mean monthly PE	P	y
25	pe_MAY	float	MM/day		mean monthly PE	P	y
26	pe_JUN	float	MM/day		mean monthly PE	P	y
27	pe_JUL	float	MM/day		mean monthly PE	P	y
28	pe_AUG	float	MM/day		mean monthly PE	P	y
29	pe_SEP	float	MM/day		mean monthly PE	P	y

30	pe_OCT	float	MM/day		mean monthly PE	P	y
31	pe_NOV	float	MM/day		mean monthly PE	P	y
32	pe_DEC	float	MM/day		mean monthly PE	P	y
33	peadj_JAN	float			PE adjustment factor	P	y
34	peadj_FEB	float			PE adjustment factor	P	y
35	peadj_MAR	float			PE adjustment factor	P	y
36	peadj_APR	float			PE adjustment factor	P	y
37	peadj_MAY	float			PE adjustment factor	P	y
38	peadj_JUN	float			PE adjustment factor	P	y
39	peadj_JUL	float			PE adjustment factor	P	y
40	peadj_AUG	float			PE adjustment factor	P	y
41	peadj_SEP	float			PE adjustment factor	P	y
42	peadj_OCT	float			PE adjustment factor	P	y
43	peadj_NOV	float			PE adjustment factor	P	y
44	peadj_DEC	float			PE adjustment factor	P	y
45	real_uztwc	float	MM	0	SAC SMA state	S	
46	real_uzfwc	float	MM	0	SAC SMA state	S	
47	real_lztwc	float	MM	0	SAC SMA state	S	
48	real_lzfsc	float	MM	0	SAC SMA state	S	
49	real_lzfpc	float	MM	0	SAC SMA state	S	
50	real_adimpc	float	MM	0	SAC SMA state	S	
51	uztwc	float	%	0	SAC SMA state	S	
52	uzfwc	float	%	0	SAC SMA state	S	
53	lztwc	float	%	0	SAC SMA state	S	
54	lzfsc	float	%	0	SAC SMA state	S	
55	lzfpc	float	%	0	SAC SMA state	S	
56	adimpc	float	%	0	SAC SMA state	S	
57	surfaceFlow	float	MM/(dt)sec	0	surface runoff	S	
58	subsurfaceFlow	float	MM/(dt)sec	0	subsurface runoff	S	
59	tsint1, tsint2, . . .		DL	0	soil ? For selected layer	S	
60	swint1, swint2, . . .	vector float [n]	DL	0		S	
61	swhint1, swhint2, . . .	vector float [n]	DL	0		S	
62	tet	float	DL	0		S	
63	totalWater1	float	DL	0		S	
64	totalWater2	float	DL	0		S	
65	upperTsint	float	DL	0		S	
66	lowerTsint	float	DL	0		S	

67	totalTsint	float	DL	0		S	
68	upperSwint	float	DL	0		S	
69	lowerSwint	float	DL	0		S	
70	totalSwint	float	DL	0		S	
71	upperSwhint	float	DL	0		S	
72	lowerSwhint	float	DL	0		S	
73	totalSwhint	float	DL	0		S	
Data Available Frozen Ground (frz)							
74	frz_STXT	float	DL	0	Frozen ground parameter - soil texture	P	y
75	frz_TBOT	float	C		Frozen ground parameter - temperature of the lower boundary	P	y
76	frz_RSMAX	float	DL		Frozen ground parameter - residual porosity	P	n
77	frz_CKSL	float	DL		Frozen ground parameter - ratio of frozen to non frozen surface	P	n
78	frz_ZBOT	float	M		Frozen ground parameter - lower boundary depth	P	n
79	frz_RTUP	float	DL		Frozen ground parameter - will be calculated from other parameters parametric data	P	n
80	frz_RTLW	float	DL		Frozen ground parameter - will be calculated from other parameters parametric data	P	n
81	frz_PSISAT	float	N/A		Frozen ground parameter - will be calculated from other parameters parametric data	P	n
82	frz_SWLT	float	DL		Frozen ground parameter - will be calculated from other parameters parametric data	P	n
83	frz_Z0	float	M		Frozen ground parameter - will be calculated from other parameters parametric data - layer depth	P	n

84	frz_Z1	float	M		Frozen ground parameter - will be calculated from other parameters parametric data - layer depth	P	n
85	frz_Z2	float	M		Frozen ground parameter - will be calculated from other parameters parametric data - layer depth	P	n
86	frz_Z3	float	M		Frozen ground parameter - will be calculated from other parameters parametric data - layer depth	P	n
87	frz_Z4	float	M		Frozen ground parameter - will be calculated from other parameters parametric data - layer depth	P	n
88	smax	float	M		Frozen ground parameter - will be calculated from other parameters parametric data	P	n
89	ts0	float	C		Frozen ground state - soil temperature at depth Z0	S	
90	ts1	float	C		Frozen ground state - soil temperature at depth Z1	S	
91	ts2	float	C		Frozen ground state - soil temperature at depth Z2	S	
92	ts3	float	C		Frozen ground state - soil temperature at depth Z3	S	
93	ts4	float	C		Frozen ground state - soil temperature at depth Z4	S	
94	uztwh	float	MM		Frozen ground state - liquid water	S	
95	uzfwh	float	MM		Frozen ground state - liquid water	S	
96	lztwh	float	MM		Frozen ground state - liquid water	S	
97	lzfsh	float	MM		Frozen ground state - liquid water	S	
98	lzfph	float	MM		Frozen ground state - liquid water	S	

99	real_uztwh	float	MM		Frozen ground state real value - liquid water	S	
100	real_uzfwh	float	MM		Frozen ground state real value - liquid water	S	
101	real_lztwh	float	MM		Frozen ground state real value - liquid water	S	
102	real_lzfsh	float	MM		Frozen ground state real value - liquid water	S	
103	real_lzfph	float	MM		Frozen ground state real value - liquid water	S	
104	uztwc_prv	float	MM	0	previous SAC SMA state	S	
105	uzfwc_prv	float	MM	0	previous SAC SMA state	S	
106	lztwc_prv	float	MM	0	previous SAC SMA state	S	
107	lzfsc_prv	float	MM	0	previous SAC SMA state	S	
108	lzfpc_prv	float	MM	0	previous SAC SMA state	S	
109	adimpc_prv	float	MM	0	previous SAC SMA state	S	
110	tair	float	C	-99	Air temperature - input grid is in F, converted to C	S	
111	frzd_up	float	CM	0	Frozen ground result	S	
112	frzd_bt	float	CM	0	Frozen ground result	S	
113	frost	float	C	0	Frozen ground result	S	
114	nsoil	int	DL	0	parametric data,number of soil layers	P	
115	nupl	int	DL	0	parametric data, number of soil layers of upper zone	P	
116	nsac	int	DL	0	parametric data, number of soil layers to reflect SAC-SMA storages	P	
117	smc1, smc2 . . .	vector float [n]	N/A	0	soil profile total moisture states, n is number of soil layers	S	
118	sh2o1, sh2o2 . . .	vector float [n]	N/A	0	soil profile liquid moisture states, n is number of soil layers	S	
119	dtfrz	float	N/A	0	parametric data, simulation time step of the frozen ground component	P	

120	liquidWater1	float	N/A	0		S	
121	liquidWater2	float	N/A	0		S	
Data Available to Snow17							
122	tair	float	C	-99	Air temperature - input grid is in F, converted to C	S	
123	psfrac	float	N/A	-1	fraction of snow and rain	P	n
124	snow_ALAT	float	N/A		Snow17 parameter	P	y
125	snow_SCF	float	N/A		Snow17 parameter	P	n
126	snow_MFMAX	float	MM/C/6hr		Snow17 parameter input as MM/6hr, but later it is converted to MM/C/dthr and output as MM/C/dthr where dthr is the model time step	P	y
127	snow_MFMIN	float	MM/C/6hr		Snow17 parameter input as MM/6hr, but later it is converted to MM/C/dthr and output as MM/C/dthr where dthr is the model time step	P	y
128	snow_NMF	float	MM/C/6hr		Snow17 parameter input as MM/6hr, but later it is converted to MM/C/dthr and output as MM/C/dthr where dthr is the model time step	P	n
129	snow_UADJ	float	MM/mb		Snow17 parameter	P	n
130	snow_SI	float	MM		Snow17 parameter	P	n
131	snow_MBASE	float	C		Snow17 parameter	P	n
132	snow_PXTMP	float	C		Snow17 parameter	P	n
133	snow_PLWHC	float	DL		Snow17 parameter	P	n
134	snow_TIPM	float	DL		Snow17 parameter	P	n
135	snow_PGM	float	MM/day		Snow17 parameter input as MM/day, but later it is converted to MM/C-dthr output as MM/C-dthr	P	n
136	snow_ELEV	float	M		Snow17 parameter	P	y
137	snow_LAEC	float	DL		Snow17 parameter	P	n
138	snow_ADC1	float	DL		Snow17 parameter	P	n
139	snow_ADC2	float	DL		Snow17 parameter	P	n
140	snow_ADC3	float	DL		Snow17 parameter	P	n
141	snow_ADC4	float	DL		Snow17 parameter	P	n

142	snow_ADC5	float	DL		Snow17 parameter	P	n
143	snow_ADC6	float	DL		Snow17 parameter	P	n
144	snow_ADC7	float	DL		Snow17 parameter	P	n
145	snow_ADC8	float	DL		Snow17 parameter	P	n
146	snow_ADC9	float	DL		Snow17 parameter	P	n
147	snow_ADC10	float	DL		Snow17 parameter	P	n
148	snow_ADC11	float	DL		Snow17 parameter	P	n
149	we	float	MM		Snow17 state	S	
150	neghs	float	MM		Snow17 state	S	
151	liqw	float	MM		Snow17 state	S	
152	tindex	float	C		Snow17 state	S	
153	accmax	float	MM		Snow17 state	S	
154	sndpt	float	CM		Snow17 state	S	
155	sntmp	float	C		Snow17 state	S	
156	sb	float	MM		Snow17 state - can be computed	S	
157	sbaesc	float	fraction		Snow17 state - can be computed	S	
158	sbws	float	MM		Snow17 state - can be computed	S	
159	storge	float	MM		Snow17 state - can be computed	S	
160	aeadj	float	MM		Snow17 state - can be computed	S	
161	sxlag1	float	MM		Snow17 state - can be computed	S	
162	sxlag2	float	MM		Snow17 state - can be computed	S	
163	sxlag3	float	MM		Snow17 state - can be computed	S	
164	sxlag4	float	MM		Snow17 state - can be computed	S	
165	sxlag5	float	MM		Snow17 state - can be computed	S	
166	sxlag6	float	MM		Snow17 state - can be computed	S	
167	sxlag7	float	MM		Snow17 state - can be computed	S	
168	tet	float	MM/dt		ET	S	
169	rmlt	float	MM/dt	0	precipitation + snow17 melt	S	
170	snsg	float	CM	0	snow depth	S	
171	twe	float	MM	0	water equivalent	S	
172	cover	float	DL	0	snow cover	S	
173	tair_prev	float	C	-99	previous Air temperature	S	
174	snof	float	N/A	0		S	

Data Available to api							
175	api_AIXW	float	in.		API parameter	P	
176	api_AIXD	float	in.		API parameter	P	
177	api_CW	float	frac		API parameter	P	
178	api_CD	float	frac		API parameter	P	
179	api_SMIX	float	in		API parameter	P	
180	api_PEX	float	in.		API parameter	P	
181	api_PEN	float	in.		API parameter	P	
182	api_FRSX	float	frac		API parameter	P	
183	api_EFC	float	frac		API parameter	P	
184	api_PIMPV	float	frac		API parameter	P	
185	api_RIVA	float	frac		API parameter	P	
186	api_RVAI	float	in.		API parameter	P	
187	api_APIKS	float	frac		API parameter	P	
188	api_BFPK	float	1/day		API parameter	P	
189	api_BFIK	float	1/day		API parameter	P	
190	api_BFIM	float	N/A	0	API parameter	P	
191	api_AICR	float	in.		API parameter	P	
192	api_CG	float	frac		API parameter	P	
193	api_AEIX	float	in.		API parameter	P	
194	api_AEIN	float	in.		API parameter	P	
195	api_APIX	float	in.		API parameter	P	
196	api_AEIK	float	1/day		API parameter	P	
197	api_APIK	float	1/day		API parameter	P	
198	api_CS	float	frac		API parameter	P	
199	api_WKW	float	N/A		API parameter	P	
200	api_WKD	float	N/A		API parameter	P	
201	api_ATIR	float	frac		API parameter	P	
202	api_ATIX	float	degF		API parameter	P	
203	api_ATIN	float	degF		API parameter	P	
204	api_CSOIL	float	frac		API parameter	P	
205	api_CSNOW	float	frac		API parameter	P	
206	api_GHC	float	N/A		API parameter	P	
207	api_FICR	float	degF		API parameter	P	
208	api_CF	float	1/degF		API parameter	P	
209	api_CP	float	in.		API parameter	P	
210	api_CT	float	1/degF(6hr)		API parameter	P	
211	api_EFA	float	frac		API parameter	P	
212	API	float	in.		API state	S	

213	SMI	float	in.		API state	S	
214	BFI	float	in.		API state	S	
215	BFSC	float	in.		API state	S	
216	AEI	float	in.		API state	S	
217	ATI	float	degF		API state	S	
218	FI	float	degF		API state	S	
219	FEI	float	1/degF		API state	S	
220	surfaceFlow	float	MM/(dt)sec	0	surface runoff		
221	subsurfaceFlow	float	MM/(dt)sec	0	subsurface runoff		
222	tet	float	DL	0			
Data Available to rutpix7							
223	rutpix_SLOPC	float	DL		routing parameter	P	
224	rutpix_ROUGC	float	DL		routing parameter	P	
225	rutpix_BETAC	float	N/A		routing parameter	P	
226	rutpix_ALPHC	float	N/A		routing parameter	P	
Data Available to rutpix9							
227	rutpix_Q0CHN	float	N/A		routing parameter	P	
228	rutpix_QMCHN	float	DL		routing parameter	P	
Data Available to rutpix7 and rutpix9							
229	rutpix_SLOPH	float	N/A		routing parameter	P	
230	rutpix_DS	float	N/A		routing parameter	P	
231	rutpix_ROUGH	float	N/A		routing parameter	P	
232	rutpix_QSC	float	N/A		routing parameter	P	
233	rutpix_POWC	float	N/A		routing parameter	P	
234	rutpix_QSH	float	N/A		routing parameter	P	
235	areac	float	M		channel state	S	
236	areac1	float	M		channel state of 1st sub-reaches	S	
237	areac2	float	M		channel state of 2nd sub-reaches	S	
238	areac3	float	M		channel state of 3rd sub-reaches	S	
239	areac4	float	M		channel state of 4th sub-reaches	S	
240	depth	float	MM		hillslope state	S	
241	ndx	int	N/A		number of sub-reaches	S	
242	discharge	float		0	routed discharge	S	

13. APPENDIX B. EXAMPLE HL-RDHM INPUT DECK

#--

```
# Example input deck for HL-RDHM User Manual
#----------------------------------------------------------------------------
#
#simulation time period
time-period                    = 19951001T00 20060930T23
#
ignore-1d-xmrg = false
#
#simulation time step in the format of HH:MM:SS.XXXX
#
time-step            =  1
#
#the connectivity file
#
connectivity = /fs/hsmb5/hydro/rms/sequence/abrfc_var_adj2.con
#
#output path
#
output-path       = /fs/hsmb5/hydro/dmip2/talo2/ws1a
#
#input paths
#
input-path        =  /fs/hsmb5/hydro/rms/parameterslx
input-path        =
/fs/hsmb5/hydro/Hydro_Data/ABRFC/PRECIPITATION/RADAR/STAGE3/dmip2
#
#select operations
#available snow17, calsnow17, sac, calsac, frz, api, rutpix7, calrutpix7, rutpix9, calrutpix0
operations = calsac calrutpix9
#
#data to be output before, inside and after timeloop in grid format
#
#output-grid-before-timeloop        = sac_LZFPM
#
#In number of timestep, example, 2 means output every 2 timestep
#
#output-grid-step             = 1
#
#data to be output at the last time step
#
#output-grid-last-step          = uztwc uzfwc lztwc lzfsc lzfpc adimpc
output-grid-last-step          = areac areac1 areac2 areac3 areac4 depth
output-grid-last-step          = real_uztwc real_uzfwc real_lztwc real_lzfsc real_lzfpc
real_adimpc
#output-grid-after-timeloop           = discharge
```

```
#
#Time series to be averaged over every basin
#
#output-timeseries-basin-average      = xmrg
output-timeseries-basin-average      = surfaceFlow
#
#Time series at the outlet
#
output-timeseries-basin-outlet      = discharge
#
#basin id and factors for input data
# basin id followed by "name=value" pairs
#------------------
#----- TALO2 ------
#------------------
input-data = TALO2
#SAC parameters
input-data =      sac_PCTIM=0.005
input-data =      sac_ADIMP=0.1
input-data =      sac_RIVA=0.03
input-data =      sac_EFC=0.5
input-data =      sac_SIDE=0.0
input-data =      sac_RSERV=0.3
input-data =      sac_UZTWM=-0.5
input-data =      sac_UZFWM=-0.80
input-data =      sac_UZK=-0.7
input-data =      sac_ZPERC=-3.49
input-data =      sac_REXP=-0.74
input-data =      sac_LZTWM=-0.82
input-data =      sac_LZFSM=-1.24
input-data =      sac_LZFPM=-2.1
input-data =      sac_LZSK=-0.51
input-data =      sac_LZPK=-0.45
input-data =      sac_PFREE=-0.30
input-data =      pe_JAN=0.9
input-data =      pe_FEB=1.0
input-data =      pe_MAR=1.70
input-data =      pe_APR=2.7
input-data =      pe_MAY=3.7
input-data =      pe_JUN=5.2
input-data =      pe_JUL=5.6
input-data =      pe_AUG=5.3
input-data =      pe_SEP=4.1
input-data =      pe_OCT=2.4
input-data =      pe_NOV=1.3
```

```
input-data =    pe_DEC=1.0
input-data =    peadj_JAN=1.0
input-data =    peadj_FEB=1.0
input-data =    peadj_MAR=1.0
input-data =    peadj_APR=1.0
input-data =    peadj_MAY=1.0
input-data =    peadj_JUN=1.0
input-data =    peadj_JUL=1.0
input-data =    peadj_AUG=1.0
input-data =    peadj_SEP=1.0
input-data =    peadj_OCT=1.0
input-data =    peadj_NOV=1.0
input-data =    peadj_DEC=1.0
#SAC states
input-data =    uztwc=0.66    uzfwc=0.0   lztwc=0.69
input-data =    lzfsc=0.0    lzfpc=0.57   adimpc=0.69
#rutpix parameters
input-data =    rutpix_Q0CHN=-1.8 rutpix_QMCHN=-0.92
#rutpix states
input-data =    areac=5.0 depth=0.0
#-------------------------------------------------------------------------------
# End of example input deck
#-------------------------------------------------------------------------------
```

14. APPENDIX C. REFERENCES

Anderson, E.A., 1973. National Weather Service River Forecast System-Snow Accumulation and Ablation Model. Technical Memo. NOAA, Silver Spring, MD, pp. 217.

Anderson, E.A., 2002. Calibration of Conceptual Hydrologic Models for Use in River Forecasting, Hydrology Laboratory.

Anderson, R.M., Koren, V.I., Reed, S.M., 2006. Using SSURGO Data to Improve Sacramento Model a-priori Parameter Estimates. Journal of Hydrology, 320(1): 103-116.

Burnash, R.J., Ferral, R.L., McGuire, R.A., 1973. A Generalized Streamflow Simulation System: Conceptual Modeling for Digital Computers, U.S. Department of Commerce National Weather Service and State of California Department of Water Resources.

Cooper, D., 2006. An operational forecast office perspective of the National Weather Service Distributed Hydrologic Modeling System (HDMS), 3rd Federal Interagency Hydrologic Modeling Conference, Reno, NV.

Cui, Z., Koren, V.I., Moreda, F. and Smith, M.B., 2006. Common Programming Framework for Distributed Hydrologic Modeling Research: An Overview of the Architecture 3rd Federal Interagency Hydrologic Modeling Conference, Reno, NV.

Duan, Q., Sorooshian, S. and Gupta, V.K., 1992. Effective and Efficient Global Optimization for Conceptual Rainfall-runoff Models. Water Resources research, 28(4): 1015-1031.

Koren, V.I., 2005. Parameterization of frozen ground effects: sensitivity to soil properties, Seventh IAHS Scientific Assembly. IAHS Publication No. 303, Foz do Iguacu, Brazil, pp. 125-133.

Koren, V.I. and Barrett, C.B., 1995. Satellite based, distributed monitoring, forecast, and simulation (MFS) system for the Nile River. In: G.W. Kite and A. Pietroniro (Editors).

Koren, V.I., Moreda, F., Reed, S., Smith, M.B. and Zhang, Z., 2005. Evaluation of a grid-based distributed hydrological model over a large area, Seventh IAHS Scientific Assembly. IAHS Publ. 303, pp. 47-56.

Koren, V.I., Reed, S., Smith, M., Zhang, Z., Seo, D.J., 2004. Hydrology laboratory research modeling system (HL-RMS) of the US national weather service. Journal of Hydrology, 291: 297-318.

Koren, V.I., Smith, M.B. and Duan, Q., 2003. Use of a priori parameter estimates in the derivation of spatially consistent parameter sets of rainfall-runoff models. In: Q. Duan, S. Sorooshian, H. Gupta, H. Rosseau and R. Turcotte (Editors), Calibration of Watershed Models. Water Science and Applications. AGU, pp. 239-254.

Koren, V.I., Smith, M., Wang, D., Zhang, Z., 2000. Use of Soil Property Data in the Derivation of Conceptual Rainfall-Runoff Model Parameters, American Meteorological Society 15th Conference on Hydrology, Long Beach, CA, pp. 103-106.

Moreda, F., Koren, V., Zhang, Z., Reed, S. and Smith, M., 2006. Parameterization of distributed hydrological models: learning from the experiences of lumped modeling. Journal of Hydrology (Amsterdam), 320(1-2): 218-237.

Parada, L.M., Fram, J.P. and Liang, X., 2003. Multi-resolution calibration methodology for hydrologic models: application to a sub-humid catchment. In: Q. Duan, H.V. Gupta, S. Sorooshian, A.N. Rousseau and R. Turcotte (Editors), Calibration of Watershed Models. American Geophysical Union, Washington, D.C.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., 1986. Numerical Recipes. Cambridge Univ. Press, 818 pp.

Reed, S., Koren, V., Smith, M., Zhang, Z., Moreda, F., Seo-D.J., DMIP Participants, 2004. Overall distributed model intercomparison project results. Journal of Hydrology, 298: 27-60.

Reed, S., Schaake, J. and Zhang, Z., 2007. A distributed hydrologic model and threshold frequency-based method for flash flood forecasting at ungauged locations. Journal of Hydrology, 337(3-4): 402-420.

Reed, S.M., 2003. Deriving flow directions for coarse-resolution (1 - 4 km) gridded hydrologic modeling. Water Resources Research, 39(9).

Reed, S.M., Fulton, R., Zhang, Z. and Guan, S., 2006. Use of 4 km, 1 hr, Precipitation Forecasts to Drive a Distributed Hydrologic Model for Flash Flood Prediction, American Meteorological Society 20th Conference on Hydrology, Atlanta, GA.

Schmidt, J.A., Anderson, A.J. and Paul, J.H., 2007. Spatially-variable, physically-derived flash flood guidance, AMS 21st Conference on Hydrology, San Antonio, TX, pp. 6B.2.

Shultz, M.J. and Corby, R.J., 2006. Development, calibration, and implementation of a distributed hydrologic model for use in real-time river forecasting, 3rd Federal Interagency Hydrologic Modeling Conference, Reno, NV.

Smith, M.B., Laurine, D.P., Koren, V.I., Reed, S.M. and Zhang, Z., 2003. Hydrologic model calibration in the National Weather Service. In: Q. Duan, H.V. Gupta, S. Sorooshian,

A.N. Rousseau and R. Turcotte (Editors), Calibration of Watershed Models. American Geophysical Union, Washington, D.C.

Smith, M.B. et al., 2004. The distributed model intercomparison project (DMIP): motivation and experimental design. Journal of Hydrology, 298: 4 - 26.

Strahler, A.N., 1964. Quantitative geomorphology of drainage basins and channel networks. In: V.T. Chow (Editor), Handbook of Applied Hydrology. McGraw-Hill, New York, pp. 4-39-4-76.

Tokar, A.S. and Johnson, P., 1995. Optimization of river characteristics in the Blue Nile watershed: NFS 3.0 Reference Manual, NOAA, Silver Spring, MD.

Zhang, Z., Koren, V.I., Reed, S., Smith, M.B. and Moreda, F., 2006. Comparison of Simulation Results Using SSURGO-based and STATSGO-based Parameters in a Distributed Hydrologic Model, 3rd Federal Interagency Hydrologic Modeling Conference, Reno, NV.