The NCAR WRF-Hydro Technical Description and User's Guide

Version 1.0

Originally Created:
April 14, 2013

Updated:

FORWARD

This User's Guide describes the WRF-Hydro model coupling architecture and physics options, released in April 2013. As the WRF-Hydro system is developed further, this document will be continuously enhanced and updated. Please send feedback to wrfhelp@ucar.edu.

This document is complementary to the main Weather Research and Forecasting (WRF) model User's Guide and technical document (http://www.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf), which describes the equations, numerics, boundary conditions, and nesting etc. of the WRF model in greater detail. To the degree practicable, this document parallels the structure of the WRF model documents.

For the latest version of this document, please visit the WRF-Hydro Users' Web site at http://www.mmm.ucar.edu/wrf/users/.

Prepared by:
David Gochis, Wei Yu, David Yates

The National Science Foundation and the National Center for Atmospheric Research
Baron Advanced Meteorological Services
National Aeronautics and Space Administration (NASA)
National Oceanic and Atmospheric Administration (NOAA) Office of Hydrological
        Development (OHD)

Table of Contents

6.1 Overview
6.2

APPENDICES:

A1. Noah HRLDAS model namelist description (namelist.hrldas)
A2. WRF-Hydro model namelist description (hydro.namelist)
A3. Vegetation parameter table (VEGPARM.TBL)
A4. Soil parameter table (SOILPARM.TBL)
A5. General parameters table (GENPARM.TBL)
A6. Channel parameters table (CHANPARM.TBL)
A7. Lake parameters table (LAKEPARM.TBL)
A8. Groundwater/baseflow bucket model parameters table
                (GWBUCKPARM.TBL)
A9. Terrestrial hydrological hydraulic parameters table
        (HYDRO.TBL)
A10. High-resolution terrain model netcdf file header
        (HYDRO_IN*)
A11. Forcing data netcdf file header (*LDASIN* and
        LDASIN_PRECIP)
A12. Land model output netcdf file header (*LDASOUT*)
A13. High resolution routing grid output netcdf file header
        (*RTOUT*)
A14. Channel observation point netcdf file header (*CHANOBS*)
A15. Channel network point netcdf file header (*CHRTOUT*)
A16. Lake point netcdf file header (*LAKES*)
A17. Forecast/observation point ASCII output file (frxst_pts.txt)
A18. Channel inflow ASCII output file (chan_inflow.txt)

1. Introduction

The purpose of this technical note is to describe the physical parameterizations, numerical implementation, coding conventions and software architecture for the NCAR Weather Research and Forecasting model (WRF) hydrological extension package, hereafter referred to as WRF-Hydro. Chapters 1-4 provide the overview historical development of the WRF-Hydro (Chapter 1), a technical description of the WRF-Hydro code and steps to install and execute the system (Chapter 2), description of model physics options (Chapter 3), WRF-Hydro pre-processing, initialization and output file descriptions (Chapter 4), description of example use cases (Chapter 5) and a catalog of utility programs that accompany the WRF-Hydro system (Chapter 6). Examples and descriptions of all major input, output, parameter and namelist files are provided in the Appendices. The system is intended to flexible and extensible and users are encouraged to develop, add and improve components to meet their application needs.

1.1 Brief History

The WRF-Hydro model coupling extension package provides a means to couple hydrological model components to atmospheric models and other Earth System modeling architectures. The system is intended to be extensible and is built upon a modular FORTRAN90 architecture. The code has also been parallelized for distributed memory, parallel computing applications. An initial suite of terrestrial hydrologic routing physics is contained within version 1.0 of WRF-Hydro but users are encouraged to add additional components to meet their application needs. Included in version 1.0 is a fully distributed, 3-dimensional, variably-saturated surface and subsurface flow model previously referred to as 'Noah-distributed' for the underlying land surface model upon which the original code was based. Initially, the implementation of terrain routing and, subsequently, channel and reservoir routing functions into the 1-dimensional Noah land surface model was motivated by the need to account for increased complexity in land surface states and fluxes and to provide physically-consistent land surface flux and stream channel discharge information for hydrometeorological applications. The original implementation of the surface overland flow and subsurface saturated flow modules into the Noah land surface model were described by Gochis and Chen (2003). In that work, a simple subgrid disaggregation-aggregation procedure was employed as a means of mapping land surface hydrological conditions from a 'coarsely' resolved land surface model grid to a much more finely resolved terrain routing grid capable of adequately resolving the dominant local landscape gradient features responsible for gravitational redistribution of terrestrial moisture. Since then numerous improvements to the Noah-distributed model have occurred including optional selection for 2-dimensional (in x and y) or 1-dimensional ('steepest descent' or so-called 'D8' methodologies) terrain routing, a 1-dimensional, grid-based, hydraulic routing model, a reservoir routing model, 2 reach-based hydrologic channel routing models, and a simple empirical baseflow estimation routine. In 2004, the entire modeling system, now referred to as the NCAR WRF-Hydro hydrological modeling extension package was coupled to the Weather Research and Forecasting (WRF) mesoscale meteorological model (Skamarock et al., 2005) thereby permitting a physics-based, fully coupled land surface hydrology-regional atmospheric modeling capability for use in hydrometeorological and hydroclimatological research and applications. The code has since been fully parallelized for high-performance computing

applications. During late 2011 and 2012, the WRF-Hydro code underwent a major re-configuration of its coding structures to facilitate greater and easier extensibility and upgradability with respect to the WRF model, other hydrological modeling components and other Earth system modeling frameworks. The new code and directory structure implemented is reflected in this document.

As additional changes and enhancements to the WRF-Hydro occur they will be documented in future versions of this document.

1.2 Model Requirements

The WRF-Hydro has been developed to facilitate improved representation of terrestrial hydrologic processes related to the spatial redistribution of surface, subsurface and channel waters across the land surface and to facilitate coupling of hydrologic models with atmospheric models. Switch-activated modules in WRF-Hydro enable treatment of terrestrial hydrological physics, which have either been created or have been adapted from existing distributed hydrological models. The conceptual architecture for WRF-Hydro is shown in Figure 1.1 where WRF-Hydro exists as a coupling architecture or 'middle-ware' layer between weather and climate models and terrestrial hydrologic models and land data assimilation systems.
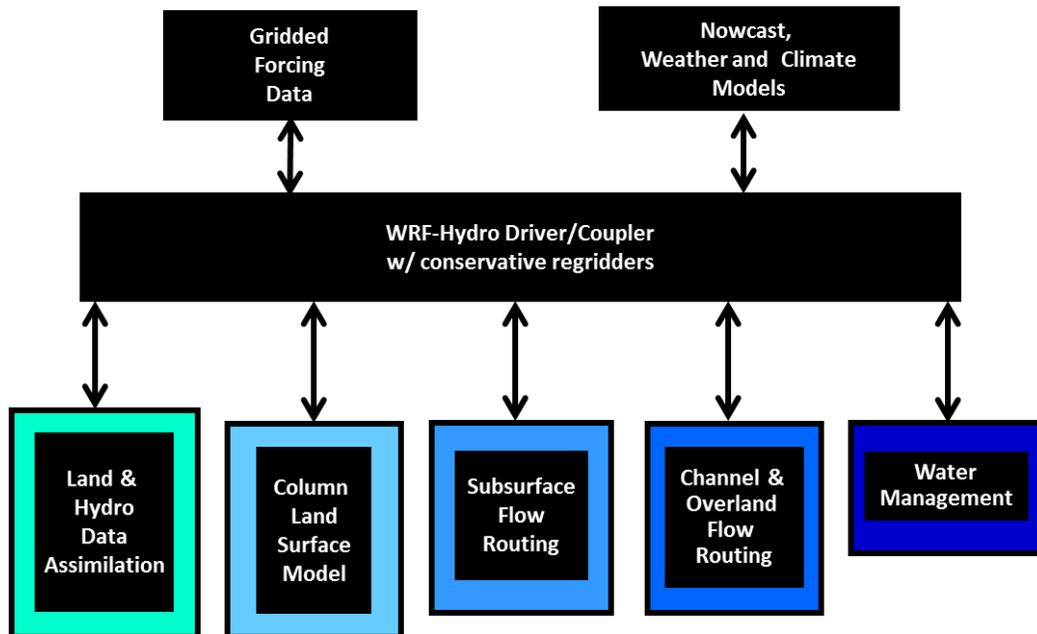
Figure 1.1. Generalized conceptual schematic of the WRF-Hydro architecture showing various categories of model components.
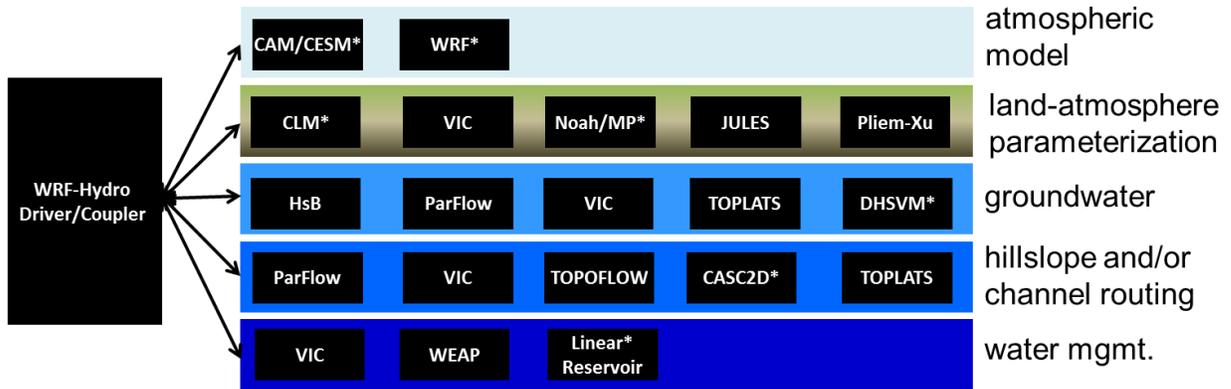
Figure 1.2. Model schematic illustrating where many existing atmosphere, land surface and hydrological model components would fit into the WRF-Hydro architecture. NOTE: Not all of these models are coupled into WRF-Hydro at this time. This schematic is meant to be illustrative. Components which are coupled have an asterisk (*) by their name.

The WRF-Hydro is designed to enable improved simulation of land surface hydrology and energy states and fluxes at a fairly high spatial resolution (typically 1 km or less) using a variety of physics-based and conceptual approaches. As such, it is intended to be used as either a land surface model in both stand-alone ('uncoupled' or 'offline') and fully-coupled (to an atmospheric model) configurations. Both time-evolving ('forcing') and static input datasets are required for model operation. The exact specification of both forcing and static data depends greatly on the selection of model physics and component options to be used. The principle model physics options in version 1.0 of WRF-Hydro include:

Table 1.1 Model physics options

    a. 1-dimensional (vertical) land surface parameterization
    b. surface overland flow
    c. saturated subsurface flow
    d. channel routing
    e. reservoir routing
    f. conceptual/empirical baseflow

Since only the Noah land surface model option is available for use in the current version of the WRF-Hydro, the rest of this document will focus on its implementation. Future versions will include other land surface model options.

Like nearly all current land surface models, the one-dimensional Noah land surface parameterization requires a few basic meteorological forcing variables including:

Table 1.2 Input forcing data for the Noah LSM

Incoming shortwave radiation $\qquad$ (W/m^2)
Incoming longwave radiation $\qquad$ (W/m^2)
Specific humidity $\qquad$ (kg/kg)
Air temperature $\qquad$ (K)
Surface pressure $\qquad$ (Pa)
Near surface wind in the u- and v-components $\qquad$ (m/s)
Liquid water precipitation rate $\qquad$ (mm/s)

[Different land surface models may require other or additional forcing variables or the specification of forcing variables in different units.]

When coupled to the WRF regional atmospheric model the forcing data is provided by the atmospheric model with a frequency dictated by the land surface model time-step specified in WRF. When run in a stand-alone mode, these forcing data must be provided as gridded input time series. External, third party, Geographic Information System (GIS) tools are used to delineate a stream channel network, open water (i.e., lake, reservoir, and ocean) grid cells and baseflow basins. Water features are mapped onto the high-resolution terrain-routing grid and post-hoc consistency checks are performed to ensure consistency between the coarse resolution Noah land model grid and the fine resolution terrain and channel routing grid. Further details on the preparation of forcing data for stand-alone WRF-Hydro execution is provided in Chapter 4.

The WRF-Hydro model components calculate fluxes of energy and moisture either back to the atmosphere or also, in the case of moisture fluxes, to stream and river channels and through reservoirs. Depending on the physics options selected, the primary output variables include:

Table 1.3 Output data from WRF-Hydro:

Surface latent heat flux
Surface sensible heat flux
Ground heat flux
Ground surface and/or canopy skin temperature
Surface evaporation components (soil evaporation, transpiration, canopy water
        evaporation, snow sublimation and ponded water evaporation)
Soil moisture
Soil temperature
Deep soil drainage
Surface runoff
Canopy moisture content
Snow depth
Snow liquid water equivalent
Stream channel inflow (optional with terrain routing)
Channel flow rate (optional with channel routing)
Channel flow depth (optional with channel routing)

Reservoir height and discharge (optional with channel and reservoir routing)

WRF-Hydro utilizes a combination of netcdf and flat ASCII file formats for input and output and therefore requires that netcdf libraries be installed on the local machine executing the simulations. For information regarding netcdf data structures and where to obtain netcdf libraries please visit the official netcdf website hosted by UNIDATA at: http://www.unidata.ucar.edu/software/netcdf/.

1.3 Computational/Hardware Requirements
The WRF-Hydro has been developed on a LINUX-based computing platform using Portland Group, gfort and ifort FORTRAN90 compilers. To date the model has been ported to the U.S. National Science Foundation IBM supercomputer 'yellowsone' located at NCAR and on the MAC/OS operating system using gfort compilers. It has not been rigorously tested on other platforms. When run in full hydrologic simulation mode, with all attendant process modules activated (e.g. overland and subsurface routing, channel and reservoir routing, and baseflow) the WRF-Hydro is a moderately computationally-intensive modeling system with respect to other physics-based environmental modeling systems (e.g. weather models, climate models, catchment hydrology models and other geophysical fluid dynamics models). However, computational requirements scale exponentially as functions of domain size and spatial resolution. Thus there are no firm rules on minimum computational requirements in terms of processor speed or memory allocation. Indeed it is possible and relatively straightforward to set up and execute simple runs over small basins using a single processor machine with a few hundred megabytes of disk space. However, the WRF-Hydro was designed for large domain, high-resolution applications which require significant computational, memory and disk storage resources. With these applications in mind, the WRF-Hydro has been fully parallelized to run on high performance computing systems. Information on the parallel computing schema is provided in Chapter 2.

The FORTRAN90 modular architecture of the WRF-Hydro allows for extensibility and compatibility of existing and newly developed parameterizations. Input/output is handled using netCDF data protocols, which enable easy visualization and analysis using an array of readily available software packages (e.g. GrADS, Unidata's IDV, IDL, MATLAB, ncl). The open-source, modular architecture is advantageous for community-based modeling systems where development occurs in geographically disparate locations.

2. Model Technical Description and User Guide

This portion of the document presents the technical description of the WRF-Hydro model code including:

1. Coding structure and programming conventions
2. Directory structures
3. Description of WRF-Hydro components
4. Setup and execution of uncoupled WRF-Hydro code
5. Setup and execution of coupled WRF/WRF-Hydro code
6. Brief description of WRF-Hydro namelists

2.1 Coding Structure and Programming Conventions
WRF-Hydro is written in a modularized, FORTRAN90 coding structure whose routing physics modules are switch activated through a model namelist file (hydro.namelist). The code has been parallelized for execution on high-performance, parallel computing architectures including LINUX operating system commodity clusters and multi-processor desktops as well as the IBM 'blue gene' class supercomputers.

The code has been compiled using the Portland Group FORTRAN compiler, the Intel 'ifort' compiler and the public license GNU Fortran compiler 'gfort' (for use with Linux-based operating systems on desktops and clusters) and the IBM AIX FORTRAN compilers (for supercomputers).

Because the WRF-Hydro relies on NETCDF input and output file conventions, NETCDF FORTRAN libraries must be installed and properly compiled on the system upon which the WRF-Hydro is to be executed. Not doing so will result in error numerous '*...undefined reference to netcdf library ...*' or similar messages upon compilation.

Parallelization of the WRF-Hydro code utilizes geographic domain decomposition and 'halo' array passing structures similar to those used in the WRF atmospheric model (Figures 2.1 and 2.2). Message passing between processors is accomplished using 'MPICH' protocols. Therefore the relevant 'mpich' libraries must be installed and properly compiled on the system upon which the WRF-Hydro is to be executed in parallel mode. Separate compilations, and therefore executables, of the WRF-Hydro code are required for single processor-sequential versus parallel simulations. See Sections 2.4 for the procedures required to install and run the WRF-Hydro in each sequential, parallel and WRF-Hydro modes.
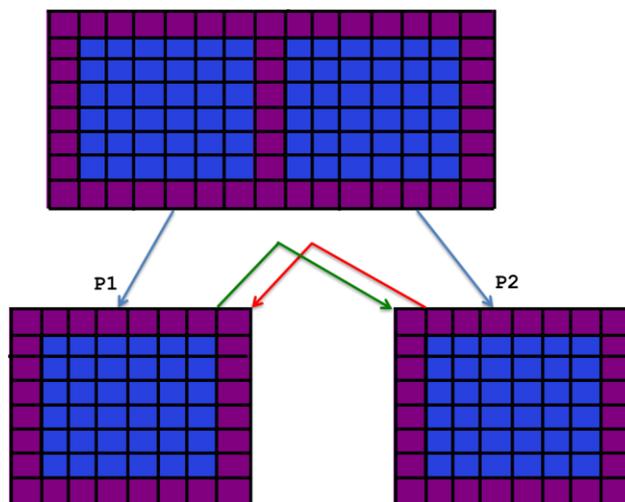
Figure 2.1 Schematic of parallel domain decomposition scheme in WRF-Hydro. Boundary or 'halo' arrays in which memory is shared between processors (P1 and P2) are shaded in purple.
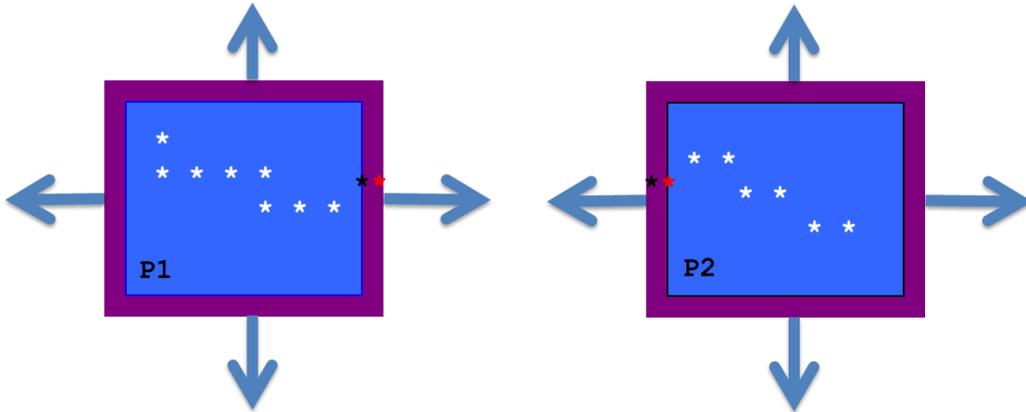
Figure 2.2 Schematic of parallel domain decomposition scheme in WRF-Hydro as applied to channel routing. Channel elements (stars) are communicated at boundaries via 'halo' arrays in which memory is shared between processors (P1 and P2). Black and red stars indicate overlapping channel elements used in the diffusive wave solver.

The WRF-Hydro extension package is essentially a group of modules and functions which handle the communication of information between atmosphere components (such as WRF, CESM or prescribed meteorological analyses) and sets of land surface components (See section 2.3 for a more complete description of the WRF-Hydro components.) From a coding perspective the WRF-hydro system is called from an existing architecture such as the WRF model, the HRLDAS, the CESM, NASA LIS, etc. Each new coupling effort requires some basic modifications to a general set of functions to manage the coupling. In WRF-Hydro, each new system WRF-Hydro is coupled into gets assigned to a directory indicating the name of the component WRF-Hydro is coupled to. For instance, the code which handles the coupling to the WRF model is contained in the WRF_cpl/ directory in the WRF-Hydro system. Similarly, the code which handles the coupling to the HRLDAS is contained within the HRLDAS_cpl/ directory and so on. Description of each directory is provided in Section 2.2 below.

This coupling structure is illustrated here, briefly, in terms of the coupling of WRF-Hydro into the WRF model. A similar approach is used for coupling the WRF-Hydro extension package into other modeling systems or for coupling other modeling systems into WRF-Hydro.

Example: For coupled WRF/WRF-Hydro runs the WRF-Hydro components get compiled as a single library function call with the WRF system. As such there is only a single executable (See Section 2.5 below for further details on the configuration and compilation procedure for coupled WRF/WRF-Hydro runs.) As illustrated in Figure 2.3 WRF-hydro is called directly from WRF in the WRF surface driver module (phys/ module_surface_driver.F). The code that manages the communication is the WRF_drv_Hydro.F interface module that is contained within the WRF_cpl/ directory.

The WRF_drv_Hydro.F interface module is the specific instance of a 'General WRF-Hydro Coupling Interface' for the WRF model which passes data, grid and time information between WRF and WRF-Hydro. Components within WRF-Hydro then manage the dynamic regridding ('data mapping') and sub-component routing functions (e.g. surface, subsurface and/or channel routing) within WRF-Hydro. Upon completion of the user-specified routing functions, WRF-Hydro will remap the data back to the WRF model grid and then pass the necessary variables back to the WRF model through the WRF_drv_Hydro.F interface module. Therefore, the key component of the WRF-Hydro system is the proper construction of the WRF_cpl_Hydro interface module (or more generally 'XXX_cpl_Hydro'). Users wishing to couple new modules to WRF-Hydro will need to create a unique 'General WRF-Hydro Coupling Interface' for their components. Some additional examples of this interface module are available upon request for users to build new coupling components.
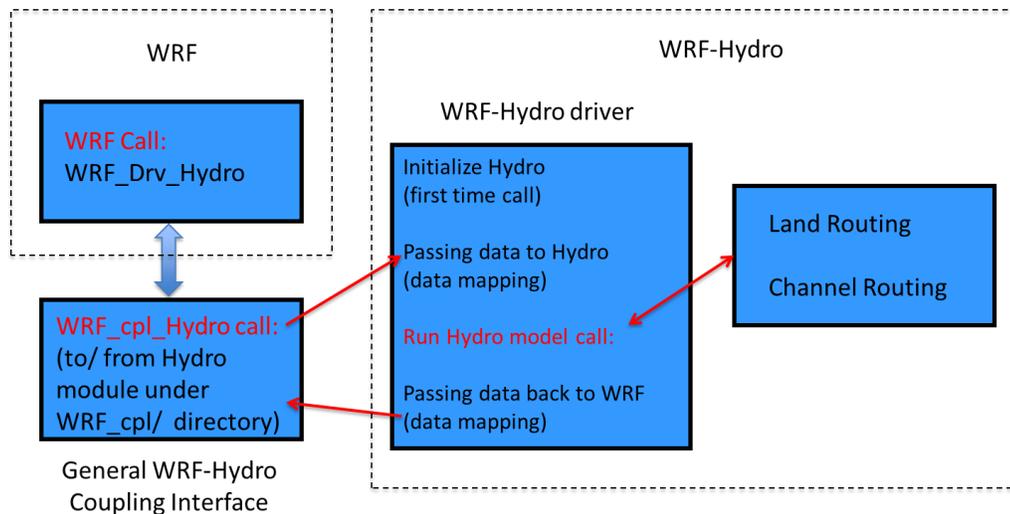
Figure 2.3 Schematic illustrating the coupling and calling structure of WRF-Hydro from the WRF Model.

2.2 Directory Structures

The top level directory structure of the code is provided below and Subdirectory structures are described thereafter. Code descriptions in bold italics indicate code that is relevant to routing modeling and/or more frequently modified routines.

This is the top level directory present immediately following untarring of the WRF-Hydro file tar package:

This section provides a brief description of the file contents of each directory where the model code resides. ***Code descriptions in bold italics indicate code that is relevant to routing modeling and/or more frequently accessed or modified routines.***

arc/ - directory containing macro files reside which specify the compile configurations, compiler options, links to netcdf libraries, etc.

configure **- script to configure the WRF-Hydro compilation**

Data_Rec/ - directory containing some data declaration modules

HRLDAS/ - directory containing the Noah land surface model driver for offline or uncoupled applications (see documentation on the HRLDAS should you desire to make changes to it: http://www.ral.ucar.edu/research/land/technology/lsm.php)

HRLDAS_cpl/ **- directory containing the WRF-Hydro coupling interface for coupling WRF-Hydro components with the HRLDAS system**

HYDRO_drv/ **- directory containing the high level WRF-Hydro component driver:**

module_HYDRO_drv.F

lib/ - directory where compiled libraries are written

macros **- macro definition file created by the 'configure' script that specifies compilation settings**

Makefile - the top-level makefile for building and cleaning HRLDAS code

Makefile.comm - the top-level makefile for building and cleaning WRF-Hydro code

mod/ - directory where compiled .mod files are written upon compilation

15

MPP/   - directory containing parallel model code

README.hydro   **- WRF-Hydro README file**

Routing/   **- directory containing modules and drivers related to specific routing processes in WRF-Hydro:**

     Makefile   – Makefile for WRF-Hydro components

     module_channel_routing.F – module containing WRF-Hydro channel routing components

     module_date_utilities_rt.F – module containing various date/time utilities for routing

     module_GW_baseflow.F – module containing model physics module for simple baseflow model

     module_HYDRO_io.F – module containing WRF-Hydro input/output functions

     module_HYDRO_utils.F – module containing several WRF-Hydro utilities

     module_lsm_forcing.F – module containing the options for reading in different forcing data types

     module_noah_chan_param_init_rt.F – module containing routines to initialize WRF-Hydro routing grids

     module_RT.F – module containing the principle routing driver which calls all the WRF-Hydro routing components

     Noah_distr_routing.F – module containing overland flow and subsurface physics routines

     rtFunction.F – module containing additional routing drivers

Run/   **- directory containing the parameter tables and namelist files required to run the WRF-Hydro.** The contents of this directory need to be present for WRF-Hydro to execute. It is recommended to copy contents of directory into an alternate directory, separate from the code and

then link the compiled executable to the new directory from which the model will be executed.

WRF_cpl/   **- directory containing the WRF-Hydro coupling interface for coupling WRF-Hydro components with the WRF system**

wrf_hydro_config   - low-level configuration script for compiling WRF-Hydro (this script is NOT edited or directly called by the user.)

2.3 Description of WRF-Hydro Components:

The basic structure of the WRF-Hydro is diagrammatically illustrated in Figure 2.4. High-level job management is handled by the WRF-Hydro system unless WRF-Hydro is coupled into, and beneath, a different modeling architecture. The WRF-Hydro system can either call an independent land model driver such as NCAR High Resolution Land Data Assimilation System (HRLDAS) to execute column land surface physics or be called by a different modeling architecture such as WRF, the NCAR CESM or the NASA LIS. When run in an offline or 'uncoupled' mode, WRF-Hydro must read in the meteorological forcing data necessary to perform land surface model calculations and it contains the necessary routines to do this. When run in a coupled mode with WRF or another larger architecture, WRF-Hydro receives meteorological forcing or land surface states and fluxes from the parent architecture. The basic execution process is as follows:

1. Upon initialization static land surface physiographic data are read into the WRF-Hydro system and the model domain and computational arrays are established.
2. Depending on whether or not WRF-Hydro is run offline as a stand-alone system or whether it is coupled into another architecture, either forcing data is read in or land surface states and fluxes are passed in.
3. For offline simulations which require land model execution, the 1-D, gridded land surface model is executed.
4. Land surface states and fluxes are then disaggregated to the high resolution terrain routing grids if routing is activated and there is a difference between the land model grid and the routing grid.
5. If activated, sub-surface routing physics are executed.
6. If activated, surface routing physics are executed.
7. If activated, the conceptual baseflow model is executed.
8. If activated, channel and reservoir routing components are executed.
9. Updated land surface states and fluxes are then aggregated from the high resolution terrain routing grid to the land surface model grid.
10. Results from these integrations are then written to the model output files and restart files or, in the case of a coupled WRF/WRF-Hydro simulation, passed back to the WRF model.
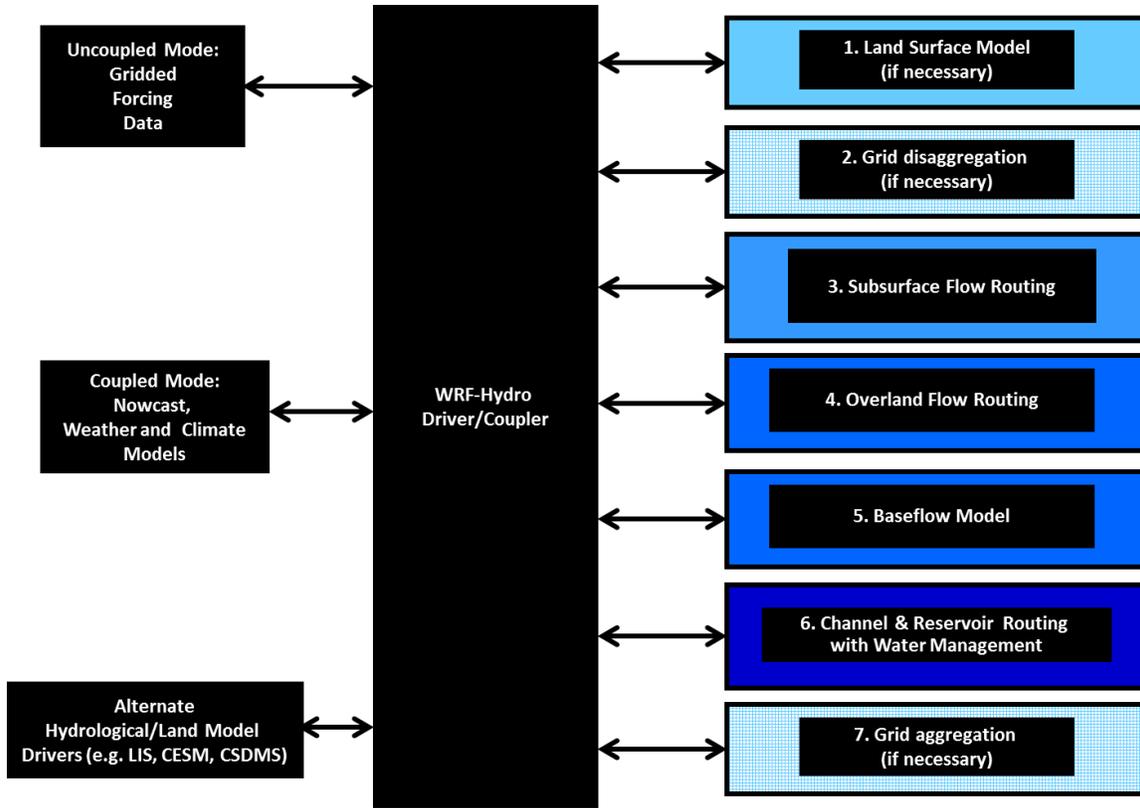
Figure 2.4 Modular calling structure WRF-Hydro.

2.4 Setup and execution of uncoupled WRF-Hydro
This section describes the step-by-step procedure for setting up and running WRF-Hydro in uncoupled and coupled-WRF/WRF-Hydro modes.

1. **Get code and set up necessary libraries:**
   All versions of WRF-Hydro code and documentation are available via the internet at: http://www.ral.ucar.edu/projects/wrf_hydro/

   a. *Unzip and/or Untar code as necessary*
   b. *Set the necessary WRF-Hydro environment variables:*

      setenv WRF_HYDRO 1   - A "1" will activate additional WRF-Hydro environment settings. "0" or no definition will default to the WRF model environment settings only when WRF is run.

      setenv HYDRO_D 1   - A "1" for HYDRO_D results in WRF-Hydro producing some run-time diagnostic information. When HYDRO_D is set to "0 "or not defined, the diagnostic information will not be produced during run-time.

      i. Set up the appropriate links to netcdf INCLUDE and LIB directories on the users system.   If the netcdf library and linclude files have not yet been created these can be set up in a local directory by the user and linked to in the proper macro file. You can explicitly set the "NETCDF_INC" and "NETCDF_LIB" environment variables or just set "NETCDF". If you only set "NETCDF" environment variable, the default NETCDF_INC and NETCDF_LIB inside WRF-Hydro will be "$NETCDF/include" and "NETCDF/lib".

               setenv NETCDF_INC   "$path/netcdf/include"
               setenv NETCDF_LIB   "$path/netcdf/lib"

          "NETCDF_INC" and "NETCDF_LIB" are defined for the WRF-Hydro only and can be different from those set for the WRF model. WRF-Hydro has two netcdf libraries for Fortran and C respectively: libnetcdff and ibnetcdf.

      ii. If the user's netcdf library combined them together (i.e. there is only one built netcdf library), the user will need to manually change this part in order to successfully compile WRF-Hydro. Refer to the README.hydro file on porting about how to change this.

      NOTE: WRF-Hydro v1.0 does not presently support parallel netcdf capabilities in netcdf 4.0 but it will in future versions.

2. Configuring and compiling WRF-Hydro for uncoupled WRF-Hydro and fully-coupled WRF/WRF-Hydro execution:

    a. Configure WRF-Hydro:

To configure the WRF-Hydro one needs to run the './configure' script that is contained within the top-level WRF-Hydro directory. There are several options for configuring the model build which have been created. New executables will be placed in the 'Run/' directory upon successful compilation.

IMPORTANT: If you desire to keep old executables you need to create a backup copy of the existing executable prior to running the configuration script. Run the 'configure' script by issuing the following command from the top-level 'WRF-Hydro' directory:

%./configure   - Executing this command will produce the following options:

Please select from following supported options.

    1. Linux PGI compiler sequential
    2. Linux PGI compiler dmpar
    3. IBM AIX compiler sequential, xlf90_r
    4. IBM AIX compiler dmpar
    5. Linux gfort compiler sequential
    6. Linux gfort compiler dmpar
    7. Linux ifort compiler sequential
    8. Linux ifort compiler dmpar
    0. exit only

    Enter selection [0-8] :

Options 1 (for single processor, sequential runs) or 2 (parallel processor runs) on typical LINUX cluster machines use the Portland Group FORTRAN (PGI) compilers. Options 3 and 4 are available if the IBM AIX compiler is used (Use this for NCAR IBM supercomputers such as 'yellowstone'). Options 5 and 6 designate configuration options for the open source GNU FORTRAN ('gfort') compiler. Options 7 and 8 designate configuration options for the open source Intel FORTRAN ('ifort') compiler. Upon hitting <RETURN> after selecting a configuration option, you will be returned to the command line and no other output to the screen is given.

At this point users should verify that all of the variables and pathways that need to be specified in the 'macros' file are properly setup or linked (e.g. COMPILER90, CPP, NETCDFINC, NETCDFLIB). Most compile-time errors or issues relate to the settings in the 'macros' file.

b. *Clean out all old model object files and executable files:*
Issue the following command:

%make clean

c. *Compiling the uncoupled WRF-Hydro version of the code:*
The configuring and compiling commands are quite basic after the user has set up the above four environment variables. The default supported platform is Linux with the PGI (Portland Group) compiler. The compiler options IBM AIX with xlf fortran and Linux with gfort and Intel-ifort compilers have also been successful tested on several systems. After checking the macros file for proper settings issue the following command to compile the code:

%make

If successful executable files created for uncoupled WRF-Hydro builds (i.e. when running 'offline' or 'uncoupled' to WRF) will be found in the Run/ directory. For example, a successful build of the uncoupled WRF-Hydro system using the default Noah land surface model will produce the following executable in the Run/ directory:

Noah_hrldas_beta

If there are compilation errors, oftentimes error messages will be provided with module names and line numbers. These module line numbers are ONLY relevant to lowercase (*.f) files and NOT uppercase (*.F) files. Since the *.f files are scrubbed within the Makefile upon compilation they are not available to view. To not scrub the .f files one needs to make the appropriate edit to the appropriate Makefile. This means determining which make file needs to be edited and commenting out the 'rm *.f' line under the appropriate module.

Be sure to check the date on the executable to make sure that you have compiled successfully. Again, if you experience problems compiling try typing 'make clean' to remove old object files.

It is recommended that users copy the contents of the Run/ directory to a new location to begin their work.

When compiling for coupled WRF/WRF-Hydro simulations, the WRF-Hydro system is called as a function inside the WRF model and thus only one executable is created. In the case of coupled WRF/WRF-Hydro compiling, a successful compilation will produce only a single "wrf.exe" file will be created and it will be placed in the main/ directory of the WRF model.

Further details on building the coupled WRF/WRF-Hydro code are provided next.

d. ***Compiling the coupled WRF-Hydro version of the code:***
In a fully-coupled mode, the 'WRF/WRF-Hydro' system serves as a hydrological extension package to the WRF atmospheric model for the purpose of performing fully-coupled hydrometeorological (i.e. rainfall, runoff, groundwater flow, streamflow) simulations and predictions. Before beginning setup of a coupled WRF-Hydro run, make sure that both uncoupled WRF model and the uncoupled version of WRF-Hydro have been properly installed, configured, compiled and executed independently on the desired system. It will be much easier to debug problems with the coupled model once each component of the uncoupled models has been properly installed compiled and executed. Also, if any changes or model developments are made to WRF-Hydro (such as routing functions or I/O modules) it is strongly suggested that you recompile and perform test executions of the uncoupled WRF-Hydro run in offline mode before attempting to directly perform a coupled WRF/WRF-Hydro model simulation. These steps will help ensure that the coupled version will compile and execute successfully and will help with troubleshooting any problems that may arise.

Upon first installation, or if you change something in the WRF-Hydro portion of the coupled model, recompile and test the uncoupled model. To install, compile and execute the coupled WRF/WRF-Hydro system do the following:

    i.   If necessary, download, unzip and untar the WRF-Hydro tar package in the top-level WRF directory.

    ii.   If you have not already done so, set the WRF-Hydro environment variable to specify that WRF-Hydro will be active. [setenv WRF_HYDRO 1]

    iii.   If you have not already done so run the WRF-Hydro configure script and compile the WRF-Hydro code as specified above. Although you will not use the executable created from this compilation this will guarantee that the WRF-Hydro part compiles correctly. If it does not compile correctly, then you will need to fix that compilation so it does before you can compile WRF-Hydro with WRF.

    iv.   Compile the WRF model as you normally would. The setting of the WRF-Hydro environment variable will force WRF-Hydro to be compiled with WRF. If successful a single 'wrf.exe' executable will be created in the WRF main/ directory.

It is highly recommended that users follow the example provided in the 'test cases' for a fully-coupled WRF run. Users should refer to the WRF documentation for questions regarding the setup of the WRF model or its data

requirements. For the latest version of this document, please visit the WRF-Hydro Users' Web site at: http://www.mmm.ucar.edu/wrf/users/.

3. **Model setup and execution:**
    a. ***Create a project Run/ directory:*** Once the proper model executable file has been created users will need to create and populate a run directory from which the model will be executed. As stated above, all of the appropriate parameter tables, model namelist files and model executable need to be placed in this directory and can be copied from the 'Run/' directory listed above and also have the model executable linked into it. Keeping voluminous terrain data, forcing data and model output data separate from the model code and executables can make backup, storage or transfer of WRF-Hydro model code between systems much simpler.

    b. ***Create/place routing files in a 'terrain/' directory.*** Place the land surface terrain data files (e.g. 'geogrid_d0X.nc' files) and, if routing is to be performed, the high-resolution netcdf terrain grid into a 'terrain/' directory. Construction of these netcdf files is described in Documentation Chapter 4.2.

    c. ***Create/place meteorological forcing data files in 'forcing/' directory.*** These are the data that "drive" the hydrologic simulation. WRF-Hydro uses a netcdf I/O convention similar to the Weather Research and Forecasting (WRF) model and, therefore, it is fairly easy to adapt WRF output to drive WRF-Hydro. (A simple script to extract and regrid WRF model output to a different WRF-Hydro domain is available in the utils/ directory of the WRF-Hydro tar package.) There are seven meteorological variables that are required: 2 m air temperature, 2 m specific humidity, 10 wind speed (u and v components), surface pressure, precipitation rate, incoming short and longwave radiation. These variables are stored within a single netcdf file where one file is specified for each model time-step. Specific details on the units and formats of these data at given in Documentation Chapter 4.3 and an example netcdf header from a forcing data file is provided in the Documentation Appendix (A11). These data are placed in within the 'forcing/' directory.

    d. ***Edit the namelists.*** Open and edit 'namelist.hrldas' and 'hydro.namelist' to set up a simulation to your specifications. In version 1.0 of WRF-Hydro the HRLDAS system which drives the Noah land surface model is the only uncoupled land model option. The namelist files are fairly well commented. The directory for input forcing data must be specified (INDIR), as must the pathway and filename to the GEO_STATIC file (GEO_STATIC_FLNM, aka a 'geogrid' file which is created from WRF pre-processing software) and type of forcing data (FORC_TYP). Similarly, for simulations where routing components are activated, a GEO_FINEGRID_FLNM must be specified. Be sure to activate only those routing switches for which you have the required data. Routing timestep 'DTRT' must be set in accordance with the routing grid spacing in order to satisfy Courant constraints (see the Documentation Chapter 3.5 for a discussion on Courant constraints). Be sure to include the full path and directory when specifying Routing Input Files. Description of the terms in the namelist are given in the Documentation Chapter 2.6 and

example namelists with descriptive comments are provided in Appendices A1 and A2.

e. ***Execute the model.*** Assuming the model was built to use the Noah land surface model, type './Noah_wrfdriver_beta' at the command line to execute the sequential version the model. For parallel runs the command may differ according the specifications of the parallel-processing software on individual machines but a common execution command may look like 'mpirun –np # Noah_wrfdriver_beta', where # is the number of processors to be used. If run successfully, output will be generated as a series of netcdf files with associated time and date information in the filenames. Depending on the runtime options selected a number of ASCII output files may also be created.

2.5 Setup and execution of the coupled WRF-Hydro System

In a fully-coupled mode, the 'WRF/WRF-Hydro' system serves as a hydrological extension package to the WRF atmospheric model for the purpose of performing fully-coupled hydrometeorological (i.e. rainfall, runoff, groundwater flow, streamflow) simulations and predictions. Before beginning setup of a coupled WRF-Hydro run, make sure that both uncoupled WRF model and the uncoupled version of WRF-Hydro have been properly installed, configured, compiled and executed independently on the desired system. It will be much easier to debug problems with the coupled model once each component of the uncoupled models has been properly installed compiled and executed. Also, if any changes or model developments are made to WRF-Hydro (such as routing functions or I/O modules) it is strongly suggested that you recompile and perform test executions of the uncoupled WRF-Hydro run in offline mode before attempting to directly perform a coupled WRF/WRF-Hydro model simulation. These steps will help ensure that the coupled version will compile and execute successfully and will help with troubleshooting any problems that may arise.

To install, compile and execute the coupled WRF/WRF-Hydro system do the following:

v.   Upon first installation or if you change something in the WRF-Hydro portion of the coupled model, recompile and test the uncoupled model:

    a. If necessary, download, unzip and untar the WRF-Hydro tar package in the top-level WRF directory.

    b. If you have not already done so, set the WRF-Hydro environment variable to specify that WRF-Hydro will be active. [setenv WRF_HYDRO 1]

    c. If you have not already done so run the WRF-Hydro configure script and compile the WRF-Hydro code as specified above in Section 2.5. Although you will not use the executable created from this compilation this will guarantee that the WRF-Hydro part compiles correctly. If it does not compile correctly, then you will need to fix that compilation so it does before you can compile WRF-Hydro with WRF.

    d. Compile the WRF model as you normally would. The setting of the WRF-Hydro environment variable will force WRF-Hydro to be compiled with WRF. If successful a single 'wrf.exe' executable will be created in the WRF main/ directory.

vi.   Setup and execute a coupled WRF/WRF-Hydro simulation:

    a. Create all necessary input files required by WRF-Hydro and place in the Run/ directory of the WRF model (i.e. place these where you will execute WRF).

b. Copy WRF-Hydro namelist, .TBL files and any restart files necessary for WRF-Hydro simulations from the hydro/ directory to your directory where you will run WRF.
c. Edit the WRF-Hydro namelist. **VERY IMPORTANT:** It is essential that the first option in the hydro.namelist file (sys_cpl = 2) be set to specify that the run is a coupled WRF/WRF-Hydro run. To do so, set 'sys_cpl' = 2. Proceed with setting up all other hydro.namelist variables for the desired configuration. An example hydro.namelist file is provided in Appendix A2.
d. Edit the WRF model 'namelist.input' file to specify all of the configurations required for the WRF model run.
e. Prepare all necessary data required by the WRF model (e.g. wrfinput and wrfbdry files)
f. Execute the WRF model executable 'wrf.exe'. Recall this single executable has compiled the WRF-Hydro components.

It is highly recommended that users follow the example provided in the 'test cases' for a fully-coupled WRF run. Users should refer to the WRF documentation for questions regarding the setup of the WRF model or its data requirements. For the latest version of this document, please visit the WRF-Hydro Users' Web site at: http://www.mmm.ucar.edu/wrf/users/.

2.6 Brief description of WRF-Hydro Namelists

There are two namelist files that users must edit in order to successfully execute the WRF-hydro system in an 'offline' mode or 'uncoupled' to the WRF. One of these namelist files is the 'hydro.namelist' file and in it are the various settings for operating all of the routing components of the WRF-Hydro system. The hydro.namelist file is well commented so that it should be very clear as to what is needed for each setting. A full printout of the hydro.namelist file is provided in Appendix A2.

The second namelist is the namelist which specifies the land surface model options to be used. This namelist can change depending on which land model is to be used in conjunction with the WRF-Hydro routing components. For example, a user would use one namelist when running the Noah land surface model coupled to WRF-Hydro but that user would need to use a different namelist file when running the CLM model or NASA LIS model coupled to WRF-Hydro. The reason for this is the WRF-Hydro is intended to be *'minimally-invasive'* to other land surface models or land model driver structures and not require significant changes to those systems. This minimal invasiveness facilitates easier coupling with new systems and helps facilitate easy supportability and version control with those systems.

In WRF-Hydro v1.0, the Noah land surface model is the main land surface model option and, when not coupled to WRF, it is driven by the NCAR HRLDAS. Therefore, the namelist to set up and run the NCAR HRLDAS system (namelist.hrldas) becomes the second namelist to be filled out by the user. For a run where WRF-Hydro is coupled to the WRF model, the WRF model input file (namelist.input) becomes the second namelist file. A full printout of the 'namelist.hrldas' file is provided in Appendix A1.

3. Model Physics Description

This chapter describes the physics behind each of the modules in version 1.0 of WRF-Hydro, which include the 1-dimensional land surface model, the overland routing routine, the saturated subsurface routing routine, the 1-dimensional channel routing routine, the reservoir routing module and a simple empirical baseflow routing routine.

3.1 Overview

[NOTE: As of this writing, only the Noah land surface model is formally supported within WRF-Hydro. Additional land surface models such as CLM or land model driver frameworks, such as the NASA Land Information System (LIS) have been coupled with WRF-Hydro but those efforts are yet in their prototype phases of development and are not yet formally supported. They will be released as soon as coupling and testing is complete.] The 1D Noah LSM calculates the vertical fluxes of energy (sensible and latent heat, net radiation) and moisture (canopy interception, infiltration, infiltration-excess, deep percolation, ponded water depth) and soil thermal and moisture states. Infiltration excess, ponded water depth and soil moisture are subsequently disaggregated from the 1D Noah LSM grid, typically of O(1–4 km) spatial resolution, to a high-resolution, O(30–100 m) routing grid using a time-step weighted method (Gochis and Chen, 2003) and are passed to the subsurface and overland flow terrain-routing modules. In typical U.S. applications, land cover classifications for the 1D Noah LSM are provided by the USGS 24-type land cover product of Loveland et al. (1995); soil classifications are provided by the 1-km STATSGO database (Miller and White, 1998); and soil hydraulic parameters that are mapped to the STATSGO soil classes are specified by the soil analysis of Cosby et al. (1984). The WRF model pre-processing system (WPS) also provides a fairly comprehensive database of land surface data that can be used to setup the Noah land surface model As discussed in Chapter 4, it is possible to use other land cover and soils datasets.

Subsurface lateral flow in WRF-Hydro is calculated prior to the routing of overland flow to allow exfiltration from fully saturated grid cells to be added to the infiltration excess calculated from the LSM. The current existing method used to calculate the lateral flow of saturated soil moisture is that of Wigmosta et al. (1994) and Wigmosta and Lettenmaier (1999), implemented in the Distributed Hydrology Soil Vegetation Model (DHSVM). It calculates a quasi-3D flow, which includes the effects of topography, saturated soil depth, and depth-varying saturated hydraulic conductivity values. Hydraulic gradients are approximated as the slope of the water table between adjacent grid cells in either the steepest descent or in both x- and y-directions. The flux of water from one cell to its down-gradient neighbor on each time-step is approximated as a steady-state solution.

The saturated subsurface routing methodology of Wigmosta et al. (1994) has no explicit information on soil layer structure: it treats the soil as a single homogeneous column. Typically, a minimum of four soil layers are used in a 2-meter soil column used in WRF-Hydro; the original Noah LSM has four. Additional discretization would permit improved resolution of a time-varying water table height although the user may vary the number and thickness of soil layers in the model namelist described in the Appendices A1 and

A2. WRF-Hydro specifies the water table depth according the depth of the top of the saturated soil layer that is nearest to the surface.

IF (Surface Head > Retention Depth) →
Route Water as Overland Flow

1- or 2-Dimensional
Diffusive Wave
Overland Flow Routing
Ogden, 1997
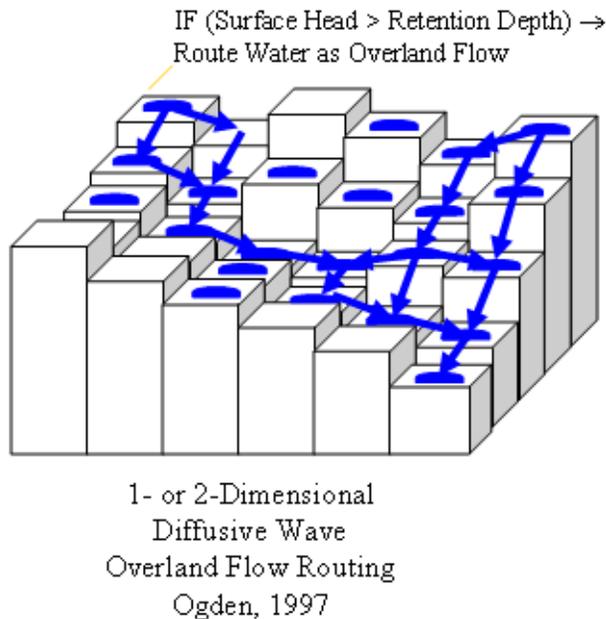
Figure 3.1: Overland flow outing modules in Noah-d

The fully unsteady, spatially explicit, diffusive wave formulation of Julien et al. (1995-CASC2D) with later modification by Ogden (1997) is the current option for representing overland flow, which is calculated when the depth of water on a model grid cell exceeds a specified retention depth. A schematic representation of the grid-cell routing process is shown in Figure 2.1. The diffusive wave equation accounts for backwater effects and allows for flow on adverse slopes (Ogden, 1997). As in Julien et al. (1995), the continuity equation for an overland flood wave is combined with the diffusive wave formulation of the momentum equation. Manning's equation is used as the resistance formulation for momentum and requires specification of an overland flow roughness parameter. Values of the overland flow roughness coefficient used in WRF-Hydro were obtained from Vieux (2001) and were mapped to the existing land cover classifications provided by the USGS 24-type land-cover product of Loveland et al. (1995), which is the same land cover classification dataset used in the 1D Noah LSM.

Additional modules have also been implemented to represent stream channel flow processes, lakes and reservoirs and stream baseflow. In WRF-Hydro v1.0 inflow into the stream network and lake and reservoir objects is a one-way process. This current formulation implies that stream and lake inflow from the land surface is always positive to the stream or lake element. There currently are no channel or lake loss functions where water can move from channels or lakes back to the landscape. Channel flow in WRF-Hydro is represented by routing a 1-dimensional diffusive wave through a gridded channel network. Water passing into and through lakes and reservoirs is routed using a simple level pool routing scheme. Baseflow, to the stream network, is represented using a simple, empirically-based bucket model which obtains 'drainage' flow from the spatially-distributed landscape. Discharge from baseflow buckets is input directly into the stream using an empirically-derived storage-discharge relationship. Each of these process options are enabled through the specification of switches in the model namelist file.

3.2 Land model description: The community Noah land surface model
The Noah land surface model is a state of the art, community, 1-dimensional land surface model that simulates soil moisture (both liquid and frozen), soil temperature, skin temperature, snowpack depth, snowpack water equivalent, canopy water content and the energy flux and water flux terms at the earth's surface (Mitchell et al., 2002; Ek et al., 2003). The model has a long heritage, with legacy versions extensively tested and validated, most notably within the Project for Intercomparison of Land surface Paramerizations (PILPS), the Global Soil Wetness Project (Dirmeyer et al. 1999), and the Distributed Model Intercomparison Project (Smith, 2002). Mahrt and Pan (1984) and Pan and Mahrt (1987) developed the earliest predecessor to Noah at Oregon State University (OSU) during the mid-1980's. The original OSU model calculated sensible and latent heat flux using a two-layer soil model and a simplified plant canopy model. Recent development and implementation of the current version of Noah has been sustained through the community participation of various agency modeling groups and the university community (e.g. Chen et al., 2005). Ek et al. (2003) detail the numerous changes that have evolved since its inception including, a four layer soil representation (with soil layer thicknesses of 0.1, 0.3, 0.6 and 1.0 m), modifications to the canopy conductance formulation (Chen et al., 1996), bare soil evaporation and vegetation phenology (Betts et al., 1997), surface runoff and infiltration (Schaake et al., 1996), thermal roughness length treatment in the surface layer exchange coefficients (Chen et al., 1997a) and frozen soil processes (Koren et al., 1999). More recently refinements to the snow-surface energy budget calculation (Ek et al., 2003) and seasonal variability of the surface emmissivity (Tewari et al., 2005) have been implemented.

The Noah land surface model has been tested extensively in both offline (e.g., Chen et al., 1996, 1997; Chen and Mitchell, 1999; Wood et al., 1998; Bowling et al., 2003) and coupled (e.g. Chen et el., 1997, Chen and Dudhia, 2001, Yucel et al., 1998; Angevine and Mitchell, 2001; and Marshall et al., 2002) modes. The most recent version of Noah is currently one of the operational LSP's participating in the interagency NASA-NCEP real-time Land Data Assimilation System (LDAS, 2003, Mitchell et al., 2004 for details). Gridded versions of the Noah model are currently coupled to real-time weather forecasting models such as the National Center for Environmental Prediction (NCEP) North American Model (NAM), and the community WRF model.

Users are referred to Ek et al. (2003) and earlier works for more detailed descriptions of the 1-dimensional land surface model physics of the Noah LSM.

3.3 *Subgrid disaggregation-aggregation*
The disaggregation-aggregation algorithms described below are found in:
Routing/ Noah_distr_routing.F

This section details the implementation of a subgrid aggregation/disaggregation scheme in WRF-Hydro. The disaggregation-aggregation routines are activated when routing of overland flow or subsurface flow is active and the specified routing grid increment is different from that of the land surface model grid. Routing in WRF-Hydro is 'switch-activated' through the declaration of parameter switches in the primary model namelist that are described in Appendix A1. In WRF-Hydro subgrid aggregation/disaggregation is used to represent overland and subsurface flow processes on grid scales much finer than the native land surface model grid. Hence, only routing is represented within a subgrid framework. It is certainly possible to use the same subgrid methodology to run the entire land surface model and routing schemes at finer resolutions than those at which forcing data, either from analyses or numerical models, is provided. This following section describes the aggregation/disaggregation methodology in the context of a 'subgrid' routing implementation.

In WRF-Hydro the routing portions of the code have been structured so that it is simple to perform both surface and subsurface routing calculations on gridcells that potentially differ from the native land surface model gridsizes provided that each land surface model gridcell is divided into integer portions for routing. Hence routing calculations can be performed on comparatively high-resolution land surfaces (e.g. a 25 m digital elevation model) while the native land surface model can be run at much larger (e.g. 1 km) grid sizes. (In this example, the integer multiple of disaggregation in this example would be equal to 40.) This capability adds considerable flexibility in the implementation of WRF-Hydro. However, it is well recognized that surface hydrological responses exhibit strongly scale-dependent behavior such that simulations at different scales, run with the same model forcing may yield quite different results.

The aggregation/disaggregation routines are currently activated by specifying either the overland flow or subsurface flow routing switches in the model namelist file and prescribing terrain grid dimensions (IXR,JXR) which differ from the land surface model dimensions (IX,JX). Additionally, the model sub-grid size (DXRT), the routing time-step (DTRT), and the integer divisor (AGGFACTR), which determines how the aggregation/disaggregation routines will divide up a native model grid square, all need to be specified in the model hydro.namelist file.

If IXR=IX, JXR=JX and AGGFACTR=1 the aggregation/disaggregation schemes will be activated but will not yield any effective changes in the model resolution between the land surface model grid and the terrain routing grid. Specifying different values for IXR, JXR and AGGFACTR≠1 will yield effective changes in model resolution between the land model and terrain routing grids.

[NOTE: As described in the Overland Flow Routing section below, DXRT and DTRT must always be specified in accordance with the routing grid even if they are the same as the native land surface model grid.]

The disaggregation/aggregation routines are implemented in WRF-Hydro as two separate spatial loops that are executed after the main land surface model loop. The disaggregation loop is run prior to routing of saturated subsurface and surface water. The main purpose of the disaggregation loop is to divide up specific hydrologic state variables from the land surface model grid square into integer portions as specified by AGGFACTR. An example disaggregation (where AGGFACTR=4) is given in Figure 3.2:

Figure 3.2 Example of the routing sub-grid implementation within the regular land surface model grid for an aggregation factor = 4.
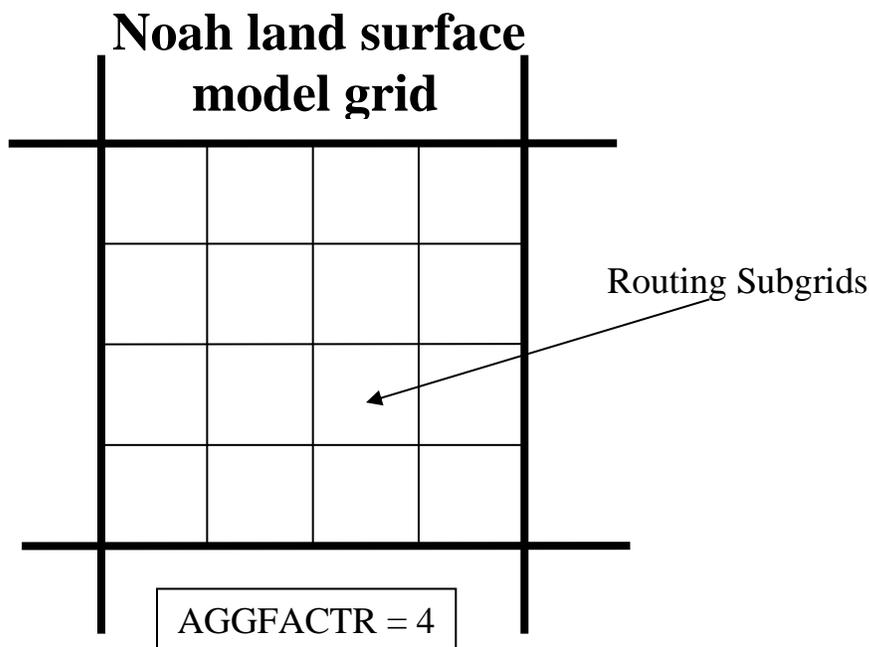
Four model variables are required to be disaggregated for higher resolution routing calculations:

SMCMAX   - maximum soil moisture content for each soil type
INFXS   - infiltration excess
LKSAT   - lateral saturated conductivity for each soil type
SMC   - soil moisture content for each soil layer

In the model code, fine-grid values bearing the same name as these with an 'R' extension are created for each native land surface model grid cell (e.g. INFXSR vs INFXS).

To preserve the structure of the spatial variability of soil moisture content on the sub-grid from one model time step to the next, simple, linear sub-grid weighting factors are assigned. These values indicate the fraction of the of total land surface model grid value that is partitioned to each sub-grid pixel.

After disaggregation, the routing schemes are executed using the fine grid values.

Following execution of the routing schemes the fine grid values are aggregated back to the native land surface model grid. The aggregation procedure used is a simple linear average of the fine gird components. For example the aggregation of surface head (SFHEAD) from the fine grid to the native land surface model grid would be:

$$SFHEAD_{i,j} = \frac{\sum\sum SFHEADR_{ir,jr}}{AGGFACTR^2} \qquad (1)$$

where, i_r and j_r are the indices of all of the gridcells residing within the native land model grid cell i,j. The following variables are aggregated and, where applicable, update land surface model variable values:

      SFHEAD   - surface head (or, equivalently, depth of ponded water)
      SMC   - soil moisture content for each soil layer

These updated values are then used on the next iteration of the land surface model.

3.4 Subsurface Routing
The subsurface flow routing algorithms are found in:
Routing/Noah_distr_routing.F

Subsurface lateral flow is calculated prior to the routing of overland flow. This is because exfiltration from a supersaturated soil column is added to infiltration excess from the land surface model, which, ultimately, updates the value of surface head prior to routing of overland flow. A supersaturated soil column is defined as a soil column that possesses a positive subsurface moisture flux which when added to the existing soil water content is in excess of the total soil water holding capacity of the entire soil column. Figure 3.3 illustrates the lateral flux and exfiltration processes in Noah-router.

In the current implementation of WRF-Hydro with the Noah land surface model, there are four soil layers. The depth of the soil layers in WRF-Hydro can be manually specified in the model namelist file under the 'ZSOIL' variable. Users must be aware that, in the present version of WRF-Hydro, total soil column depth and individual soil layer thicknesses are constant throughout the entire model domain. Future versions will attempt to relax this constraint.

However, the model is capable of using a different distribution of soil column layer depths and these simply need to be specified in the model namelist file. Assuming a 2 m soil profile the soil layer depths (and associated water table depths) are:

Table 3.1: Depths of 4 soil layers in Noah-router

Layer	z (depth to top of layer, mm)
1	0
2	100
3	400
4	1000
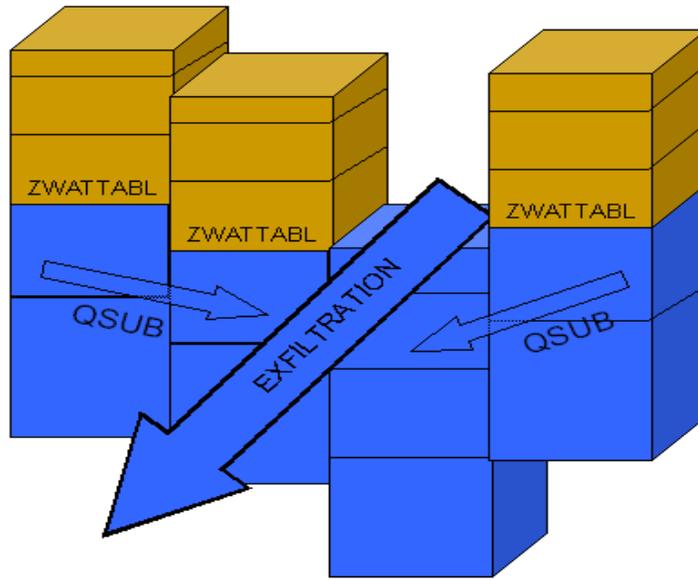
Figure 3.3 Conceptualization of saturated subsurface flow components.

The method used to calculate the lateral flow of saturated soil moisture employs a quasi three-dimensional flow representation, which include the effects of topography, saturated soil depth (in this case layers), and saturated hydraulic conductivity. Hydraulic gradients are approximated as the slope of the water table between adjacent gridcells in the x- and y-directions. In each cell, the flux of water from one cell to its down-gradient neighbor on each time-step is approximated as a steady-state solution. The looping structure through the model grid performs flux calculations separately the x- and y-directions. A second alternative, using the path of steepest descent, is also available in Noah-distributed.

Using Dupuit-Forcheimer assumptions the rate of saturated subsurface flow at time t can be calculated as:

$$q_{i,j} = \begin{cases} -T_{i,j} \tan \beta_{i,j} w_{i,j} & \beta_{i,j} < 0 \\ 0 & \end{cases} \tag{3.1}$$

where, $q_{i,j}$ is the flow rate from cell i,j, $T_{i,j}$ is the transmissivity of cell i,j, $\beta_{i,j}$ is the water table slope and $w_{i,j}$ is the width of the cell which is fixed for a regular grid. $\beta_{i,j}$ is calculated as the difference in water table depths between two adjacent gridcells divided by the grid spacing. The method by which the water table depth is determined is provided below. Transmissivity is a power law function of saturated hydraulic conductivity ($Ksat_{i,j}$) and soil thickness ($D_{i,j}$) given by:

$$T_{i,j} = \begin{cases} \dfrac{Ksat_{i,j}D_{i,j}}{n_{i,j}}\left(1-\dfrac{z_{i,j}}{D_{i,j}}\right) & z_{i,j} <= D_{i,j} \\ 0 & z_{i,j} > D_{i,j} \end{cases} \tag{3.2}$$

where, $z_{i,j}$ is the depth to the water table. $n_{i,j}$ in Eq. (6) is defined as the local power law exponent and is a tunable parameter that dictates the rate of decay of $Ksat_{i,j}$ with depth. When Eq. (2.2) is substituted into (2.1) the flow rate from cell i,j to its neighbor in the x-direction can be expressed as

$$q_{x(i,j)} = \gamma_{x(i,j)} h_{i,j} \quad when \quad \beta_{x(i,j)} < 0 \tag{3.3}$$

where,

$$\gamma_{x(i,j)} = -\left(\frac{w_{i,j}Ksat_{i,j}D_{i,j}}{n_{i,j}}\right)\tan\beta_{x(i,j)} \tag{3.4}$$

$$h_{i,j} = \left(1-\frac{z_{i,j}}{D_{i,j}}\right)^{n_{i,j}} \tag{3.5}$$

This calculation is repeated for the y-direction when using the two-dimensional routing method. The net lateral flow of saturated subsurface moisture (Q_{net}) for cell i,j then becomes:

$$Q_{net(i,j)} = h_{i,j}\sum_x \gamma_{x(i,j)} + h_{i,j}\sum_y \gamma_{y(i,j)} \tag{3.6}$$

The mass balance for each cell on a model time step (Δt) can then be calculated in terms of the change in depth to the water table (Δz):

$$\Delta z = \frac{1}{\phi_{(i,j)}}\left[\frac{Q_{net(i,j)}}{A} - R_{(i,j)}\right]\Delta t \tag{3.7}$$

where, ϕ is the soil porosity, R is the soil column recharge rate from infiltration or deep subsurface injection and A is the grid cell area. In WRF-Hydro, R, is implicitly accounted for during the land surface model integration as infiltration and subsequent soil moisture increase. Assuming there is no deep soil injection of moisture (i.e. pressure driven flow from below the lowest soil layer), R, in WRF-Hydro is set equal to 0.

The methodology outlined in Equations 2.2-2.7 has no explicit information on soil layer structure, as the method treats the soil as a single homogeneous column. Therefore, changes in water table depth (Δz) can yield water table depths, which fall within a

particular soil layer. WRF-Hydro specifies the water table depth according the depth of the top of the highest (i.e. nearest to the surface) saturated layer. The residual saturated water above the uppermost, saturated soil layer is then added to the overall soil water content of the overlying unsaturated layer. This computational structure requires accounting steps to be performed prior to calculating Q_{net}.

Given the timescale for groundwater movement and limitations in the model structure there is significant uncertainty in the time it takes to properly spin-up groundwater systems. The main things to consider include 1) the specified depth of soil and number and thickness of the soil vertical layers and 2) the prescription of the model bottom boundary condition. Typically, for simulations with deep soil profiles (e.g. > 10m) the bottom boundary condition is set to a 'no-flow' boundary (SLOPETYP = 8) in the GENPARM.TBL parameter file (see Appendix A5, for a description of GENPARM.TBL).

3.5 Surface overland flow routing

The terrain routing algorithms described below are found in:
Routing/Noah_distr_routing.F

Overland flow in WRF-Hydro is calculated using a fully-unsteady, explicit, finite-difference, diffusive wave formulation (Figure 3.1 and 3.4) similar to that of Julien et al. (1995) and Ogden et al. (1997). The diffusive wave equation, while slightly more complicated, is, under some conditions, superior to the simpler and more traditionally used kinematic wave equation, because it accounts for backwater effects and allows for flow on adverse slopes. The overland flow routine described below can be implemented in either a 2-dimensional (x and y direction) or 1-dimension (steepest descent or 'D8') method. While the 2-dimensional method may provide a more accurate depiction of water movement across some complex surfaces it is more expensive in terms of computational time compared with the 1-dimensional method. While the physics of both methods are identical we have presented the formulation of the flow in equation form below using the 2-dimensional methodology.
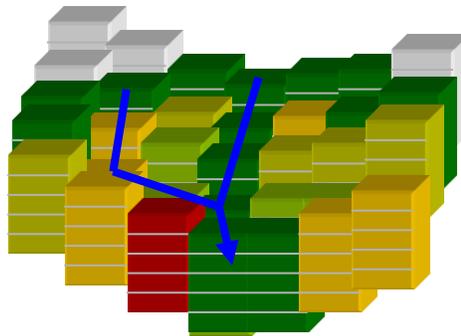
Figure 3.4: Conceptual representation of terrain elements. Flow is routed across terrain elements until it intersects a "channel" grid cell indicated by the blue line where it becomes 'in-flow' to the stream channel network.

The diffusive wave formulation is a simplification of the more general St. Venant equations of continuity and momentum for a shallow water wave. The two-dimensional continuity equation for a flood wave flowing over the land surface is

$$\frac{\partial h}{\partial t} = \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial x} = i_e \qquad (3.8)$$

where, h is the surface flow depth; q_x and q_y are the unit discharges in the x- and y-directions, respectively; and i_e is the infiltration excess. The momentum equation used in the diffusive wave formulation for the x-dimension is

$$S_{fx} = S_{ox} - \frac{\partial h}{\partial x} \qquad (3.9)$$

where, S_{fx} is the friction slope (or slope of the energy grade line) in the x-direction, S_{ox} is the terrain slope in the x-direction and $\partial h/\partial x$ is the change in depth of the water surface above the land surface in the x-direction.

In the 2-dimensional option, flow across the terrain grid is calculated first in the x- then in the y-direction. In order to solve Eq. 2.8 values for q_x and q_y are required. In most hydrological models they are typically calculated by use of a resistance equation such as Manning's equation or the Chezy equation, which incorporates the expression for momentum losses given in Eq. 2.9. In WRF-Hydro, a form of Manning's equation is implemented:

$$q_x = \alpha_x h^{\beta} \qquad (3.10)$$

where,

$$\alpha_x = \frac{S_{fx}^{1/2}}{n_{OV}}; \qquad \beta = \frac{5}{3} \qquad (3.11)$$

where, n_{OV} is the roughness coefficient of the land surface and is a tunable parameter and β is a unit dependent coefficient expressed here for SI units.

The overland flow formulation has been used effectively at fine terrain scales ranging from 30-1000 m. There has not been rigorous testing to date, in WRF-Hydro, at larger length-scales (> 250 m). This is due to the fact that typical overland flood waves possess length scales much smaller than 1 km. Micro-topography can also influence the behavior of a flood wave. Correspondingly, at larger grid sizes (e.g. > 500 m) there will be poor resolution of the flood wave and the small-scale features that affect it. Also, at coarser resolutions, terrain slopes between gridcells are lower due to an effective smoothing of topography as grid size resolution is decreased. Each of these features will degrade the performance of dynamic flood wave models to accurately simulate overland flow processes. Hence, it is generally considered that finer resolutions yield superior results.

The selected model time step is directly tied to the grid resolution. In order to prevent numerical diffusion of a simulated flood wave (where numerical diffusion is the artificial dissipation and dispersion of a flood wave) a proper time step must be selected to match the selected grid size. This match is dependent upon the assumed wave speed or celerity (*c*). The Courant Number, $C_n = c(\Delta t/\Delta x)$, should be close to 1.0 in order to prevent numerical diffusion. The value of the C_n also affects the stability of the routing routine such that values of C_n should always be less than 1.0. Therefore the following model time steps are suggested as a function of model grid size:

Table 3.2: Suggested routing time steps for various grid spacings

x (m)	t (s)
30	2
100	6
250	15
500	30
1000	60

The overland flow routing option is activated using a switch parameter (OVRTSWTCH) in WRF-Hydro model namelist. If activated the following terrain fields and model namelist parameters must be provided:

- Terrain grid or Digital Elevation Model (DEM) Note: this grid may provided at resolutions equal to or finer than the native land model resolution
- Channel network grid identifying the location of stream channel grid cells
- Specification of the routing grid cell spacing (DXRT), routing grid time step (DTRT) and subgrid aggregation factor (AGGFACTR-defined as the ratio of the subgrid resolution to the native land model resolution, see Section 2.3 above.)

3.6 Channel and lake routing description
The channel routing algorithms are found in:
Routing/ module_channel_routing.F

Currently there is no explicit sub-grid representation of overland flow discharging into a stream channel. Instead, a simple mass balance calculation is performed. Inflow to stream channels occurs when the ponded water depth (or surface head, 'SFHEADRT') of stream channel grid cells exceeds a pre-defined retention depth ('RETDEPRT'). As indicated above, the depth of surface head on any grid cell is a combination of the local infiltration excess, the amount of water flowing onto the grid cell from over land flow, and exfiltration from groundwater flow (described below). The quantity of surface head in excess of the retention depth is accumulated as stream channel inflow and is added to the channel routing routine (described below).

When routing is activated any overland flow in excess of the local retention depth, reaching a stream channel grid cell will be effectively 'discharged' into the stream channel network. In WRF-Hydro, values of 'channel inflow' are accumulated on the channel grid. Users can keep track of the total channel network channel inflow volume by setting the hydro.namelist file option 'HIRES_OUT' = 1, which produces an ASCII-formatted timeseries file of channel inflow volume in units of millions of cubic meters (Mm^3). The name of the file produced will be 'chan_inflow.txt' and it will be written to the local run directory. Users can also output the spatial distribution of channel inflow on a high-resolution, netcdf-formatted terrain grid along with the 'chan_inflow.txt' file by setting the HIRES_OUT = 2 in the model hydro.namelist file.

The channel routing module (module_channel_routing.F) allows for the one-dimensional, distributed routing of streamflow across the domain. An optional, switch-activated, level-pool lake/reservoir algorithm is also available and is described below. Channel routing is performed on a pixel-by-pixel basis along a predefined channel network grid that is input to the model within the high-resolution terrain routing grid file (see Chapter 4 for details). Within each channel grid cell there is an assumed channel reach of trapezoidal geometry as depicted in Figure 3.5. Channel parameters side slope, bottom width and roughness are currently prescribed as functions of Strahler stream order which is also input within the high-resolution terrain routing grid file. As discussed above in Section 3.5, channel elements receive lateral inflow from overland flow. There is currently no overbank flow so flow into the channel model is effectively one-way and the vertical dimension of the channel is effectively infinite. Future enhancements will attempt to relax these assumptions. Therefore, WRF-Hydro does not presently explicit represent inundation areas. Uncertainties in channel geometry parameters and the lack of an overbank flow representation result in significant uncertainty for users wishing to compare model flood stages versus those from observations. It is strongly recommended that users compare model versus observed streamflow discharge values and use observed stage-discharge relationships or 'rating curves' when wishing to relate modeled/predicted streamflow values to actual river levels and potential inundation areas.
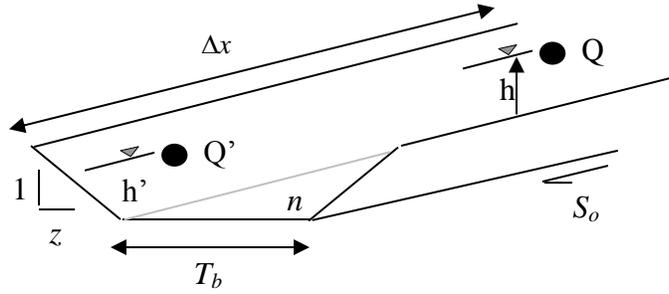
- Channel Slope, S_o
- Channel Length, Δx (m)
- Channel side slope, z (m)
- Constant bottom width, T_b (m)
- Manning's roughness coefficient, n

Figure 3.5 Schematic of Channel Routing Terms

Channel flow down through the gridded channel network is performed using an explicit, one-diemnsional, variable time-stepping diffusive wave formulation. As mentioned above in Sections 3.1 and 3.5, the diffusive wave formulation is a simplification of the more general St. Venant equations for shallow water wave flow. Similarly, for channel routing, the mass and momentum continuity equations are given as:

Continuity:
$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q_{lat} \tag{3.12}$$

Momentum:
$$\frac{\partial Q}{\partial t} + \frac{\partial (\beta Q^2 / A)}{\partial x} + gA\frac{\partial Z}{\partial x} = -gAS_f \tag{3.13}$$

Where, t is the time, x is the streamwise coordinate, A is in the flow area of the cross section, and q_{lat} is the lateral flow rate. In the momentum equation, Q is the flow rate, β is a momentum correction coefficient, Z is the water surface elevation, and S_f is the friction slope computed as:

$$S_f = \left(\frac{Q}{K}\right)^2 \tag{3.14}$$

where K is the conveyance, computed from the Manning's equation:

$$K = \frac{C_m}{n} AR^{2/3} \tag{3.15}$$

where n is the Manning's roughness coefficient, A is the cross-sectional area, R is the hydraulic radius (A/P), P is the wetted perimeter, and C_m is dimensional constant (1.486 for English units or 1.0 for SI units).

Ignoring the convective term in the momentum equation gives the diffusive wave approximation of open channel flow. The momentum equation then simplifies to:

$$Q = -\text{SIGN}\left(\frac{\partial Z}{\partial x}\right) K \sqrt{\left|\frac{\partial Z}{\partial x}\right|} \qquad (3.16)$$

where the substitution for friction slope has been made and the SIGN function is 1 for $\partial Z/\partial x > 0$ and -1 for $\partial Z/\partial x < 0$.

The numerical solution is obtained by discretizing the continuity equation over a raster cell as:

$$A^{n+1} - A^n = \frac{\Delta t}{\Delta x}\left(Q^n_{i+\frac{1}{2}} - Q^n_{i-\frac{1}{2}}\right) + \Delta t q^n_{lat} \qquad (3.17)$$

where $Q^n_{i+\frac{1}{2}}$ is the flux across the cell face between point i and $i+1$, and is computed as:

$$Q^n_{i+\frac{1}{2}} = -\text{SIGN}\left(\Delta Z^n_{i+1}\right) K_{i+\frac{1}{2}} \sqrt{\frac{\left|\Delta Z^n_{i+1}\right|}{\Delta x}} \qquad (3.18)$$

where:

$$\Delta Z^n_{i+1} = Z^n_{i+1} - Z^n_i \qquad (3.19)$$

$$K^n_{i+\frac{1}{2}} = 0.5\left[\left(1 + \text{SIGN}\left(\Delta Z^n_{i+1}\right)\right)K_i + \left(1 - \text{SIGN}\left(\Delta Z^n_{i+1}\right)\right)K_{i+1}\right] \qquad (3.20)$$

Presently a first-order, Newton-Raphson (N-R) solver is used to integrate the diffusive wave flow equations. Under certain streamflow conditions (e.g. typically low gradient channel reaches) the first-order solver method can produce some instabilities resulting in numerical oscillations in calculated streamflow values. To address this issue, higher order solver methods will be implemented in future versions of WRF-Hydro.

Variable time-stepping in the diffusive wave channel routing module in order to satisfy Courant constraints and avoid numerical dispersion and instabilities in the solutions. Unlike typical overland flow flood waves which have very shallow flow depths, on the order of millimeters or less, channel flood waves have appreciably greater flow depths and wave amplitudes, which can potentially result in strong momentum gradients and strong accelerations in a propagating wave. To properly characterize the dynamic propagation of such highly variable flood waves it is often necessary to decrease model time-steps in order to satisfy Courant conditions. Therefore WRF-Hydro utilizes a variable time-step methodology. The initial value of the channel routing time-step is set equal to that of the overland flow routing timestep which is a function of grid spacing. If, during model integration the N-R convergence criteria for upstream-dowsntream streamflow discharge values is not met, the channel routing time-step is decreased by a factor of one-half and the N-R solver is called again.

It is important to note that use of variable time-stepping can significantly affect model computational performance resulting in much slower solution times for rapidly evolving streamflow conditions such as those occurring during significant flood events. Therefore,

selection of the time-step decrease factor (default value set to 0.5) and the N-R convergence criteria can each affect model computational performance.

Uncertainty in channel routing parameters also can have a significant impact on the accuracy of the model solution which implies that model calibration is often required upon implementation in a new domain. Presently, all of the channel routing parameters are prescribed as functions of stream order in a channel routing parameter table 'CHANPARM.TBL'. The structure of this file is described in detail in Appendix A6. It should be noted that prescription of channel flow parameters as functions of stream order is likely to be a valid assumption over relatively small catchments and not over large regions. Future versions of WRF-Hydro will incorporate options to prescribe spatially distributed channel routing parameters (side slope, bottom width and roughness) within the high-resolution terrain routing grid file.

The channel flow routing option is activated using a switch parameter (CHRTSWTCH) in WRF-Hydro model hydro.namelist. If activated the following terrain fields and model hydro.namelist parameters must be provided:

- Terrain grid or Digital Elevation Model (DEM) Note: this grid may provided at resolutions equal to or finer than the native land model resolution
- Channel network grid identifying the location of stream channel grid cells
- Strahler stream order grid identifying the stream order for all channel pixels within the channel network
- Channel flowdirection grid. This grid explicitly defines flow directions along the channel network.
- Optional: Forecast point grid. This grid is a grid of selected channel pixels for which channel discharge and flow depth are to be output within a netcdf point file and an ASCII timeseries file.

3.7 Lake and Reservoir routing description
The lake and reservoir routing algorithms are found in:
Routing/ module_channel_routing.F

A simple mass balance, level-pool lake/reservoir routing module allows for an estimate of the inline impact of small and large reservoirs on hydrologic response. A lake/reservoir or series of lakes/reservoirs are identified in the channel routing network, and lake/reservoir storage and outflow are estimated using a level-pool routing scheme. The only conceptual difference between lakes and reservoirs as represented in WRF-Hydro is that reservoirs contain both orifice and weir outlets for reservoir discharge while lakes only contain weirs.

Fluxes into a lake/reservoir object occur through the channel network and when surface overland flow intersects a lake object. Fluxes from lake/reservoir objects are made only through the channel network and no fluxes from lake/reservoir objects to the atmosphere or the land surface are currently represented (i.e. there is currently no lake evaporation or subsurface exchange between the land surface and lakes and reservoirs). The Level Pool scheme tracks water elevation changes over time, *h(t)* where water from the reservoir can exit either through weir overflow (Q_w) and/or a gate-controlled flow (Q_o), where these outflows are functions of the water elevation and spillway parameters. Weir flow is given as $Q_w(t) = C_w L h^{3/2}$ when $h > h_{max}$ or $Qw(t) = 0.0$ when $h \leq h_{max}$ where, h_{max} is the maximum height before the weir begins to spill (m), C_w is a weir coefficient, and L is the length of the weir (m). Orifice flow is given as $Q_o(t) = C_o O_a \sqrt{2gh}$, where C_o is the orifice coefficient, O_a is the orifice area (m^2), and g is the acceleration of gravity (m/s^2). In addition, the level pool scheme is designed to track each reservoir's surface area, *Sa* (km^2) as a function of water depth and the area at full storage, *As* (km^2). Presently, lake/reservoir object is assumed to have vertical side walls, such that the surface area is always constant.

Figure 3.6 Schematic of Level Pool Routing

The following lake/reservoir parameters are required for level-pool routing and are defined in the 'LAKEPARM.TBL' parameter table:

- Weir and Orifice Coefficients (*Co, Cw*)
- Weir Length, L (m)
- Orifice Area, O_a (m^2)
- Reservoir Area, *As* (km^2)
- Maximum reservoir height at full storage, h_{max} (m)

The lake/reservoir flow routing option is activated when lake objects are defined and properly indexed as a data field in the high resolution terrain routing grid file. If lake/reservoir objects are present in the lake grid (and also within the channel network) then routing through those objects will occur. There are several special requirements for the lake grid and channel routing grids when lakes/reservoirs are to be represented and these are discussed in Chapter 4. The following input data variables and parameter files are required for level-pool routing:

- LAKEPARM.TBL parameter file (described in the Appendix A7)
- LAKEGRID
- CHANNELGRID

3.8 *Conceptual baseflow model description*
Routing/ module_GW_baseflow.F

Aquifer processes contributing baseflow often operate at depths greater than well below ground surface. As such, there are often conceptual shortcomings in current land surface models in their representation of groundwater processes. Because these processes contribute to streamflow (typically as 'baseflow') a parameterization is often used in order to simulate total streamflow values that are comparable with observed streamflow from gauging stations. Therefore, a switch-activated baseflow module 'module_GW_baseflow.F' has been created which conceptually (i.e. *not* physically-explicit) represents baseflow contributions to streamflow. This model option is particularly useful when WRF-Hydro is used for long-term streamflow simulation/prediction and baseflow or 'low flow' processes must be properly accounted for. Besides potential calibration of the Noah land surface model parameters the conceptual baseflow model does not directly impact the performance of the land surface model scheme. The new baseflow module is linked to WRF-Hydro through the discharge of 'deep drainage' from the land surface soil column (sometimes referred to as 'underground runoff').

The baseflow parameterization in WRF-Hydro uses spatially-aggregated drainage from the soil profile as recharge to a conceptual groundwater reservoir (Fig. 3.7). The unit of spatial aggregation is often taken to be that of a sub-basin within a watershed where streamflow data is available for the sub-basin. Each sub-basin has a groundwater reservoir ('bucket') with a conceptual depth and associated conceptual volumetric capacity. The reservoir operates as a simple bucket where outflow (= 'baseflow' or 'stream inflow') is estimated using an empirically-derived function of recharge. The functional type and parameters are determined empirically from offline tests using the estimated of baseflow from stream gauge observations and model-derived estimates of bucket recharge provided by WRF-Hydro. Presently, WRF-Hydro uses an exponential function for estimating the bucket discharge as a function of a conceptual depth of water in the bucket. Note that because this is a highly conceptualized formulation that the depth of water in the bucket in no way infers the actual depth of water in a real aquifer system. Estimated baseflow discharged from the bucket model is then combined with lateral inflow from overland flow from Noah-distributed and is input directly into the stream network as 'stream inflow'. Presently, the total basin baseflow flux to the stream network is equally distributed among all channel pixels within a basin. Lacking more specific information on regional groundwater basins, the groundwater/baseflow basins in WRF-Hydro are assumed to match those of the surface topography.
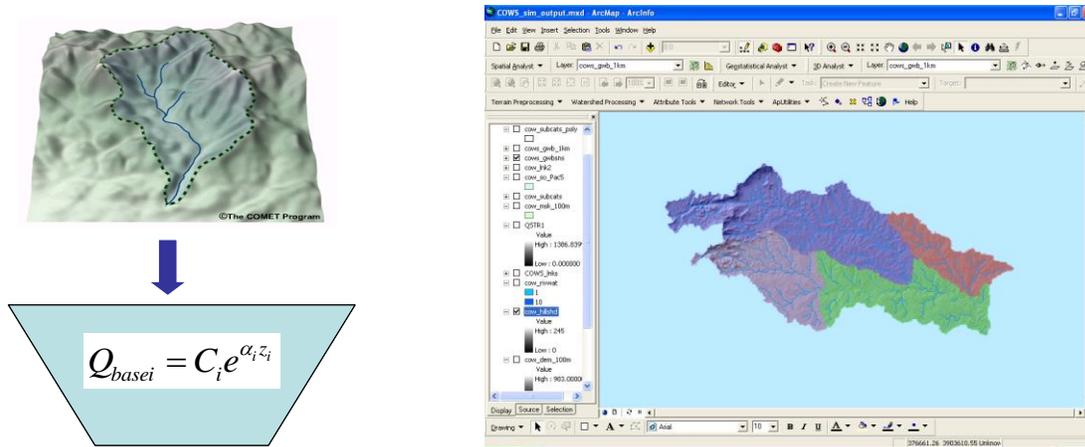
Figure 3.7: Conceptualization of baseflow 'bucket' parameterization in WRF-Hydro and hypothetical map of groundwater/baseflow sub-basins within a watershed.

A groundwater/baseflow bucket model parameter file (GWBUCKPARM.TBL) specifies the empirical parameters governing the behavior of the bucket model parameterization for each groundwater/baseflow basin specified within the model domain. An example parameter file with 4 groundwater basins will look like:

```
Basin,Coeff.,Expon.,Zmax,Zinit
1,0.7760, 3.144, 0.100, 0.0982
2,0.0400, 3.220, 0.070, 0.0358
3,0.4270, 2.813, 0.125, 0.0678
4,0.0140, 5.861, 0.055, 0.0358
```

where, 'Coeff.' is the bucket model coefficient, 'Expon.' is the bucket model exponent and 'Zinit' is the initial depth of water in the bucket model. It is important to remember that a simple bucket model is a highly abstracted and conceptualized representation of groundwater processes and therefore the depth of water values in the bucket have no real physical basis. As mentioned above, initial values of the groundwater bucket model parameters, including 'Zinit' are typically derived analytically or 'offline' from WRF-Hydro and then are fine-tuned through model calibration. A description of the procedure to derive initial groundwater bucket model parameters is provided in the Appendix A8.

To activate the simple baseflow bucket model in WRF-Hydro the user must do each of the following:

1. Set the model namelist variable GWBASESWC = 1
2. Properly assign the groundwater/baseflow bucket model parameters within the GWBUCKPARM.TBL file. Suggested steps for deriving the bucket model parameters are provided in Chapter 4.
3. Define groundwater/baseflow basins as data layers within the high resolution terrain routing grid file. Steps to generate these data layers assuming an identical match between surface and subsurface basins is provided in Chapter 4.

If activated, three ASCII-formatted, time-series, output files are generated by the bucket model parameterization that contain timeseries values of the flow into the bucket ('gw_inflow.txt'), flow out of the bucket ('gw_outflow.txt') and the conceptual depth of water in the groundwater/baseflow bucket ('gw_zlev.txt').

4. WRF-Hydro Input and Output

This chapter describes WRF-Hydro model input requirements and data structures as well as the various output formats. The chapter is divided into the following sections:

4.1 Overview
4.2 Domain processing and description of surface physiographic input files
4.2 Description of meteorological forcing data input files
4.3 Description of output files
4.4 Description of parameter files

4.1 Overview
Utilizing the netcdf data model, there are two files that need to be created in order to run WRF-Hydro. The two files specify the data layers that are used for modeling on the land surface model grid (or coarse grid) and the terrain routing grid (or high-resolution grid). Data contained in the land surface model grid is highly dependent upon the specific land surface model (LSM) used in WRF-Hydro. As of the preparation of this document only the Noah LSM (Ek et al., 2003, Mitchell et al. 2002) has been coupled into WRF-Hydro though future versions may incorporate additional LSMs such as the Community Land Model (CLM, Oleson et al., 2010). Conversely, because of the modular structure of WRF-Hydro, data for the terrain routing grid should remain fairly consistent even when different LSMs are coupled into the system. Lastly, preparation of the LSM grid and the terrain grid are required and these grids are the same, whether or not WRF-Hydro is executed in a coupled mode with the Weather Research and Forecasting (WRF) model. In the sections below, the data requirements for model execution are provided as are an example set of processing steps that would create the required LSM and terrain routing grids. The last two sections describe the required attributes that must be specified, via parameter tables, to enable channel routing, reservoir routing and a simple baseflow parameterization.

One key requirement in version 1.0 of WRF-Hydro in setting up the LSM and terrain grids is that the spatial extent of the two grids must be identical and that the spatial resolution of the terrain grid must be an integer multiple of the LSM grid. This is because the terrain grid operates on a fine mesh overlain onto the LSM and that in WRF-Hydro selected model state and flux variables are disaggregated/aggregated between the LSM and terrain grids. This internal nesting in Gochis and Chen (2003) allows the LSM to run at one spatial resolution while the terrain and stream channel routing routines are executed on a much more finely resolved grid. While the model can operate when both grids have equal spatial resolution (e.g. 100m or finer) this sub-grid nesting capability is particularly useful in minimizing computational demands when WRF-Hydro is coupled to WRF.

4.2 Domain processing and description of surface physiographic input files

The data required to execute a spatially-distributed, or gridded, 1-dimensional (vertical) LSM is specified on the Land Surface Model grid (hereafter referred to as the 'LSM grid' data file). The data contained within LSM grid data file are dependent on the specific choice of LSM used in WRF-Hydro. In this section, we will use the example of the Noah LSM as this model has already been coupled to WRF-Hydro and because this model is already coupled into the WRF model. First, use of the WRF pre-processor 'geogrid' to build the LSM grid data file will be discussed. The 'geogrid' tool is extremely useful since it automates the entire procedure of defining in space, geo-referencing and attributing most of the land surface parameter data required to execute the Noah LSM. Users familiar with the WRF model or desiring to run WRF-Hydro coupled to the WRF model will most likely use this approach. Second, we present a methodology to develop a 'custom' LSM grid data file in netcdf format using a set of tools known as netcdf command operators (or NCO commands). Users not familiar with the WRF modeling framework and who do not desire to use WRF-Hydro may choose to use this second method. Either method can produce a useable LSM grid data file.

i. WRF Pre-processor 'GEOGRID':
The WRF modeling system contains a set of data 'pre-processors' that prepare both land surface and atmospheric data for use in the WRF mesoscale atmospheric model. Combined the suite of pre-processors are referred to as the WRF Pre-processing System (WPS). The land surface model pre-processor component of WPS is known as 'geogrid' and it acquires and interpolates land surface terrain, soils and vegetation data from standard, readily available data products (such as the USGS National Elevation Dataset or the STATSGO Soils Database) to the WRF modeling grid. Complete documentation and user instructions for use of the WPS-geogrid system are provided online by NCAR and are updated regularly and, thus, are not discussed in great detail here (please visit the WPS Documentation Website: *http://www.mmm.ucar.edu/wrf/users/docs/user_guide_V3.1/users_guide_chap3.htm* or more recent versions for complete details). For installation and execution of the 'geogrid' WRF pre-processor users are referred to that documentation. However, here we will discuss the structure of the netcdf LSM grid data file that is produced by geogrid focusing on the data fields required by the Noah LSM. The following is a listing *only* of the data fields contained within the geogrid output file *that are required by the Noah LSM* (other fields contained in the file are not discussed here).

HGT_M : Topographic elevation (units of meters) on the 'mass grid'.
    (NOTE: WRF uses both centered and staggered grid systems but only the centered 'mass' grid is required by the land model.)
XLAT_M : Latitude coordinates, in decimal degrees, on the mass grid.
XLONG_M : Longitude coordinates, in decimal degrees, on the mass grid.
LANDUSEF : Land use fraction, in units of fraction. This is a 3-
    dimensional array in x, y, and land use category where each land use category is expressed as a fractional amount of area per spatial grid cell. During execution of the Noah LSM the dominant land

cover type for each spatial grid cell is determined and assigned as the single land cover type for the entire LSM grid cell (i.e. there is no sub-grid mosaicking of land cover type.)

SOILCTOP: Top layer soil texture category, in units of fraction. This is a 3-dimensional array in x, y, and soil texture category where each soil texture category is expressed as a fractional amount of area per spatial grid cell. During execution of the Noah LSM the dominant soil texture type for each spatial grid cell is determined and assigned as the single soil texture type for the entire LSM grid cell (i.e. there is no sub-grid mosaicking of soil texture type.)

GREENFRAC : Monthly mean green vegetation fraction values (units of fraction). This is a 3-dimensional array in x, y and time. During execution of the Noah LSM, the monthly values of green vegetation fraction are interpolated to daily values and are updated daily.

ALBEDO12M : Monthly mean surface albedo values (units of %) not including snow effects. This is a 3-dimensional array in x, y and time. During execution of the Noah LSM, the monthly values of land surface albedo are interpolated to daily values and are updated daily.

It is important to remember the above listed data fields are those that are required to run the Noah LSM. while the geogrid pre-processor creates an LSM grid data file that contains other data fields which may not be required to run the Noah LSM in an 'offline' or 'uncoupled' (i.e. un-coupled to the WRF model) mode. Therefore, users seeking to create their own LSM input datafiles only need to create those fields listed above. An example procedure to do this within the netdf framework is described below.

ii. Custom LSM grid netcdf file development:

As described above, the data required to run a gridded implementation of the Noah LSM within WRF-Hydro include topographic elevation (units of meters, 'HGT_M'), latitude of each grid cell (units of decimal degrees, 'XLAT_M'), longitude of each grid cell (units of decimal degrees, 'XLONG_M') land use fraction (units of fraction, 'LANDUSEF'), soil texture class (units of fraction, 'SOILCTOP'), monthly mean vegetation greenness fraction (units of fraction, 'GREENFRAC') and monthly mean albedo values (units of %, 'ALBEDO12M). Users creating their own LSM grids (i.e. those users not using the WRF geogrid pre-processors need to create netcdf files containing the proper data with the proper units and specified filenames (filenames are case sensitive). Once the user has created the individual netcdf data layers (either through code or a third party piece of software like ArcGIS or MATLAB), the individual datefiles con be concatenated together into a single netcdf datafile using netcdf command operators ('NCO'-commands – see http://nco.sourceforge.net/) the 'concatenate.csh' utility script contained within the /utils/ directory distributed with the WRF-Hydro tarfile. Use

of this script is described below towards the end of the section describing the terrain grid setup and is also contained in the comment lines of the script.

The most critical thing to remember when creating a custom LSM grid data file is that the vegetation type index and the soils type index must be consistent with the indices that will be defined in the vegetation and soils parameter tables described below and in the Appendix. Native data for the LSM grid can come from any number of different sources. Therefore, it is also important to remember that in order to run a gridded implementation of the LSM all data must be mapped to single, consistent grid and that the spatial increment or resolution of the terrain grid must be an integer multiple of the resolution of the LSM grid. The reason for this requirement is discussed above in the General Overview section.

iii. Land surface model parameter specification:
   Land surface parameters in the Noah LSM are specified in three different files:
      VEGPARM.TBL : contains vegetation parameters indexed by land use/land cover categories.
      SOILPARM.TBL : contains soil physical parameters indexed by soil textural classes.
      GENPARM.TBL : contains miscellaneous model parameters that are applied globally. Examples of these files and description of their parameters is given in the Appendix.

iv. High-resolution terrain grid development
   The high-resolution terrain routing grid specifies the data that are necessary to route water across the landscape (via overland and saturated surbsurface flow) and through stream channels and lakes. Options also exist to specify baseflow values for stream channel routing based on an empirical bucket model formulation (see Section 3.8). The data layers contained within the high-resolution terrain grid include:

- Topography (required - in units of meters)
- Channel grid (required)
- Flow direction (required - using the ArcGIS direction convention)
- Stream Order (required for channel routing – using the Strahler (1952) stream order convention)
- Lakes (optional)
- Groundwater Basin Mask (optional)
- Grid of monitoring points or stream gauging stations (optional)
- Grid of latitude and longitude values (often required for proper geo-referencing of irregular gridded data in netcdf – in units of decimal degrees)

- Grid of overland flow roughness scaling parameters (set to = 1.0 until calibration is performed)
- Grid of surface retention depth scaling parameters (set to = 1.0 until calibration is performed)

Data is passed to/from the LSM grid to the terrain routing grid via a disaggregation/aggregation scheme described in Gochis and Chen (2003). The main requirement to enable the coarse-fine grid functionality is that terrain grid must exactly match the extent of the LSM grid and its dimensions must be integer multiples of the LSM grid. Given the need to interpolate, georeference and clip grids from different sources, resolutions and spatial projections, it is highly advantageous to use a Geographical Information System (GIS). In the following paragraphs we provide an example procedure to create these grids, in netcdf format using the ESRI ArcGIS system. Many other methods could be used as well so long as care is taken to accurately map the LSM and terrain grids to one another. NCAR and others in the hydro-informatics community are presently developing automated algorithms for developing these data layers using other GIS packages such as GRASS or MapWindow.
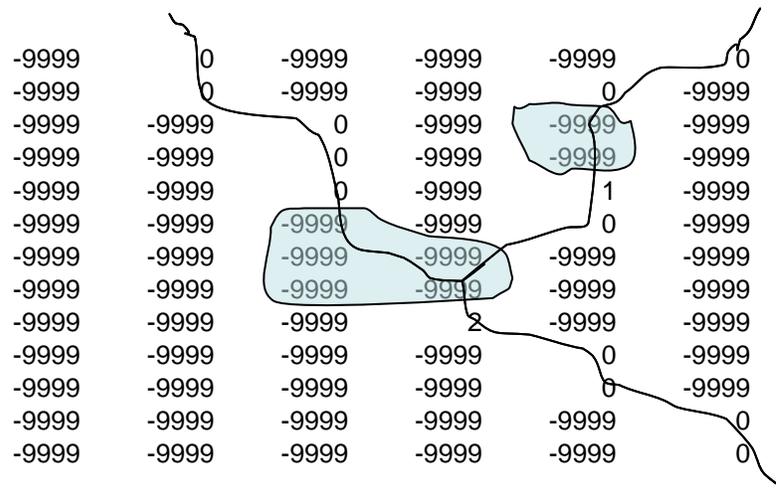
Figure 4.1 Example of final channel grid format with 2 lake objects present. Note that the main channel identifier index is 0 but that this number changes at lake/reservoir outlet to correspond with the index value of each reservoir shown in Figure 4.2 below. Also, all channel grid cells underneath a lake are assigned as -9999.

-9999	-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	1	-9999
-9999	-9999	-9999	-9999	1	-9999
-9999	-9999	-9999	-9999	-9999	-9999
-9999	-9999	2	-9999	-9999	-9999
-9999	-9999	2	2	-9999	-9999
-9999	-9999	2	2	-9999	-9999
-9999	-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	-9999	-9999
-9999	-9999	-9999	-9999	-9999	-9999
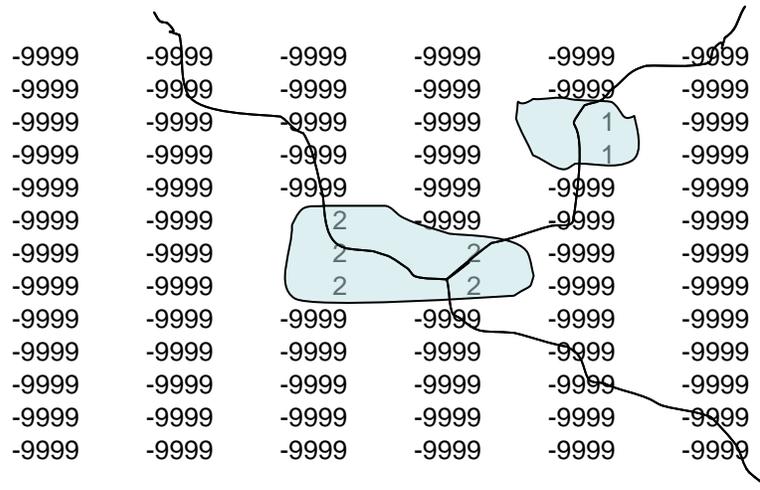
Figure 4.2 Example of final lake grid format with 2 lake objects present.

v. Import the LSM Grid into ArcGIS:
[A set of automated scripts for importing geogrid and geogrid-like netcdf files into ArcGIS and GRASS are being developed and will be made available with the WRF-Hydro code in future releases.]

A step-by-step procedure to ingest and georeference the LSM geogrid file created by the wrf pre-processors (described above) into ArcGIS is given as follows:

1. Using ArcGIS v9.3 or higher, import the geogrid netcdf file using the Spatial Analyst -> Multidimension Tools -> Create Raster Layer tool. Select the LSM terrain elevation values on the 'mass grid' (HGT_M) data layer from the geogrid file for import. Specify the 'x-dimension' to be 'east_west' and the 'y-dimension' to be 'north-south'.
2. Export/Save the elevation data layer as a raster (see Figure X for an example of how to set-up this export:
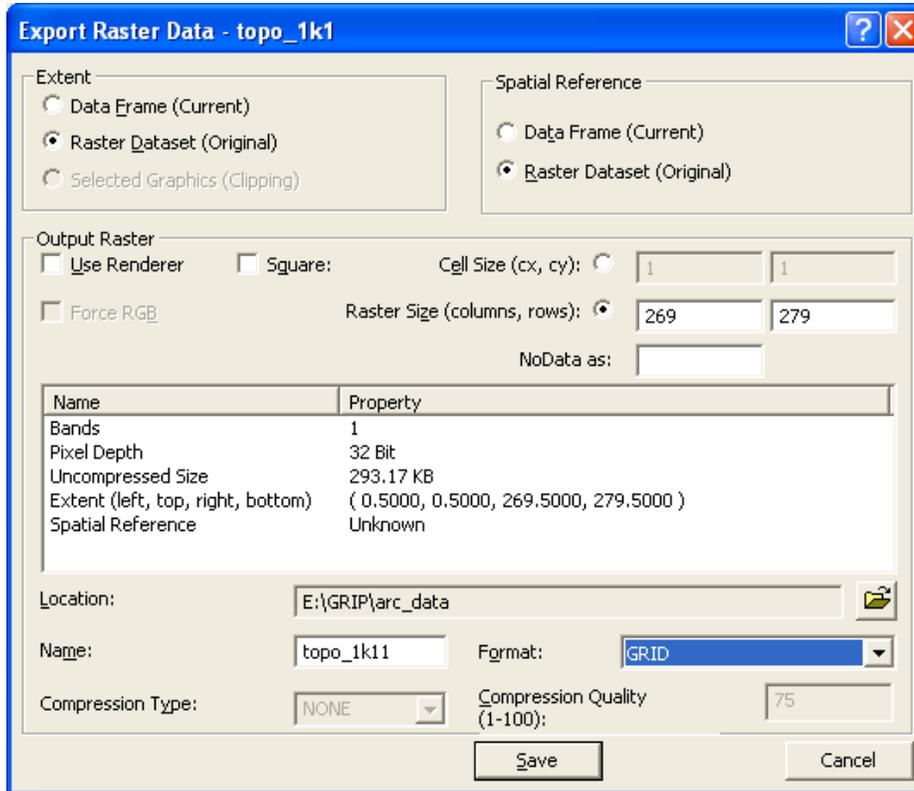
Figure 4.3. Export layer data to raster window in ArcGIS:

3. Export the raster to an ascii data file.
4. Obtain the lat/lon value (in decimal degrees) of the center point of the lower-left most point of the grid. This can be done a number of ways but the simplest method is to extract the first value in each of the 'corner_lats' and 'corner_lons' data arrays in the 'global attributes' section of the geogrid netcdf file header. This can easily be done using the 'ncdump' utility and scrolling down to the 'global attributes' section of the netcdf file header.
5. Create a single point data file in a spreadsheet for import into ArcGIS with the coordinates of the lower leftmost grid cell extracted from Step 4 above.
6. Import the point file intor ArcGIS and define the projection as geographic (e.g. WGS 84).
7. Project the point file to the desired projection specified in the WRF geogrid file (e.g. Lambert Conic Conformal) using the projection parameters provided in the 'global attributes' section of the netcdf file header.
8. Use the Data Management -> Features -> Add X/Y Data tool to add the x and y coordinates of the point to the point attribute table in the projected (e.g. Lambert Conic Conformal) coordinate system.

59

9. Edit the header in the ascii data file created in Step 3 above to specify the correct xllcenter and yllcenter values and the correct cell size values extracted from Step 4. Also set the cellsize to the appropriate values (e.g. 1000 for 1000 meters). An example based on the raster data from Fig. XX above is as follows:

```
ncols           269
nrows           279
xllcenter       -134332.118354
yllcenter       -138760.799122
cellsize        1000
NODATA_value    -9999
```

Note: The xll… and yll… parameter names have been changed to xllcenter and yllcenter to specify that the coordinate values given represent centerpoint values of the lower leftmost gridcell. The values listed here for xllcenter and yllcenter are in units of meters and are relative to the Lambert Conic Conformal projection specified in the geogrid netcdf file.

10. Import the ascii file back into ArcGIS as a raster datafile.
11. Define the projection of the raster using the 'Import' option in the Projection Definition menu and set the new grid projection to the desired projection specified in the WRF geogrid file (e.g. Lambert Conic Conformal) using the projection parameters provided in the 'global attributes' section of the netcdf file header.
12. Verify that the data is properly georeferenced by overlaying several additional datasets of known and verified projection.

Successful completion of these steps should create an ArcGIS raster of terrain elevation which is properly projected according to the geographic projection specified in the geogrid file and in the WRF/WPS pre-processing system.

vi.   Processing High Resolution (~100m or less) DEM:
Once the LSM grid has been defined within ArcGIS work can proceed on setting up the high-resolution grid. The steps to create the grids are as follows:

1. Obtain a high-resolution digital elevation model (DEM). A near-global (Arctic/Antarctic excepted) high resolution DEM developed through a joint project of the World Wildlife Fund and the U.S. Geological Survey called Hydrosheds (http://hydrosheds.cr.usgs.gov/) provides a reasonably high quality terrain dataset which is based on Shuttle Radar Topography Mission (SRTM) data. The resolution of

this dataset is 90m although gaps in SRTM data are filled coarser resolution data which results in some artifacts. Additionally, the Hydrosheds data set offers a 'hydrologically-conditioned' DEM which provides a more continuous flow field for use in hydrologic modeling. (See the Hydrosheds website and documentation for full details).

The Hydrosheds data is served in 5deg by 5deg lat/lon tiles. These tiles must be mosaicked to a single raster in ArcGIS and its spatial projection must be defined (the initial definition of spatial project must be that of the Hydrosheds DEM which is specified in the Hydrosheds documentation). Following definition of spatial projection, the data must be re-projected into the LSM grid projection which is specified in the geogrid file. Following projection one must then clip, interpolate and grid-match the high-resolution data to the LSM grid. Through much trial and error we have found the following process to be the most effective and efficient:

    a. In the Spatial Analyst drop-down menu, specify the domain 'extent' to that of the LSM grid and define the 'cell size' (resolution) to be the desired resolution of the high-resolution terrain grid. For example, if the geogrid datafile has a 1km spatial resolution a common resolution for the terrain routing grid is 100m. Thus, 100m would be set as the cell size in the Spatial Analyst options 'cell size' tab.

    b. Using the 'Raster Calculator' function in the Spatial Analyst drop-down menu, create the high resolution terrain grid using a simple equation:

        new_grid = 90m_DEM_grid
        (where the 90m_DEM_grid is the mosaicked and projected Hydrosheds DEM.)

        Because the extent and cell size have already been defined in the Spatial Analyst options sections the 'new_grid' will exactly match in spatial extent and grid cell edges the LSM grid. Although other methods in ArcGIS may be used (e.g. raster 'Clip' function) we have found those to be unreliable with regard to exactly matching cell size, grid cell edges and overall extent.

2. Once the high-resolution DEM has been defined one can proceed with deriving flow direction, channel grid and stream order grids. These tasks exist as pre-defined functions in the Spatial Analyst->Hydrology toolbox or as part of the ArcHydro toolbox available through (http://www.crwr.utexs.edu/giswr/ydro/ArcHOSS/index.cfm) The steps to derive the these fields using ArcGIS are as follows:

a. Create a 'Flow Direction' grid using the Spatial Analyst -> Hydrology -> FlowDirection tool. If successful, the output grid should have integer values of 1, 2, 4, 8, 16, 32, 64 and 128 and be oriented in the following directions:
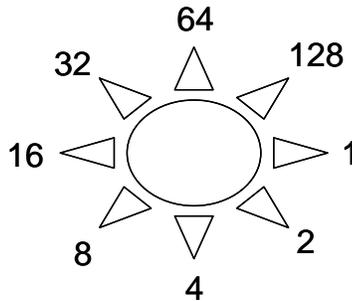
b. Create a 'Flow Accumulation' grid using the Spatial Analyst -> Hydrology -> FlowAccumulation

c. Define the channel grid using either the 'Stream Definition' tool from the ArcHydro toolbox or using a logical operation in the Spatial Analyst Raster Calculator or Spatial Analyst Raster toolbox. Essentially, one simply needs to specify the minimum threshold of flow accumulation upon which a stream channel will be defined. There is little reliable guidance for this value in a 'global' sense since climate, soils, geology, vegetation and geomorphic processes all combine to define channel networks. However, typical values often range between 1-10 sq. km for many temperate or humid climates.

d. (Optional for channel routing) If channel routing is to be performed, one must specify stream order as many channel parameters in WRF-Hydro are defined on the basis of stream order. To create a grid of stream order use the 'Stream Order' tool within the Spatial Analyst -> Hydrology toolbox.

e. (Optional for lake/reservoir routing) If specification of lakes and/or reservoirs are desired one must define these as a lake grid and, subsequently, update the channel grid to that the channel network accurately interacts with the specified reservoirs. The basic steps to do this include masking out the stream channel network where reservoirs exist (accomplished using spatial analyst logical operators and/or the 'Reclassify' tool) and defining the outlet of each reservoir on the channel grid as an integer index that

matches the associated reservoir. An example of what the lake grid and channel grid with a reservoir looks like is shown in Figs. 4.1 and 4.2 above. The steps to do this are as follows:

i. Obtain a good quality lake coverage (e.g. shapefile), project it to the proper projection (e.g. Lambert Conic Conformal) and assign a proper index for each lake ranging from 1 to n-# of lakes.

ii. Convert the lake coverage or shapefile to a grid using the Spatial Analyst -> Feature to Raster tool. Make sure that the proper spatial analyst options for domain extent and cell size have been properly set so that the output grid will perfectly match the other gridded data layers.

iii. Create a 'reverse' or 'negative' lake mask by re-classifying all 'lake grid cells' (e.g. those that have data values) to 'NoData' and all 'NoData' values equal to 1.

iv. In the Spatial Analyst -> Raster Calculator tool, multiply the existing channel grid by the 'negative' lake mask to create a new channel grid.

v. Convert the new channel grid file to a point dataset using the Raster to Point tool in the Conversions From Raster toolbox.

vi. Manually edit the output grid cell points to have a proper lake index value associated with the adjacent lake. Use the flow direction grid to help trace the proper flow path if it is not initially obvious.

vii. Convert the channel points file back to grid using the Point to Raster tool in the Conversions to Raster toolbox.

viii. Reclassify all no data values to -9999 and verify that all non-outlet channel gridcells have values of '0' and all channel outlet grid cells have values according to the adjacent lake.

f. (Optional for simple Groundwater/Baseflow implementation, additional details are provided in Section 2.8) To permit use of the groundwater/baseflow module in WRF-Hydro, one must define a groundwater basin that receives drainage from the LSM soil column (i.e. 'recharge') and feeds the stream channel network (i.e. 'discharge'). Generally, there is very little information on the spatial extent of aquifers or their actual connectivity with surface channel networks. Thus, we have adopted an overly-simple approach by defining a groundwater basin that is the same as the surface watershed defined solely by topography. (However, other methods can be used and users are encouraged to experiment.) To create a watershed function one needs to define a basin outlet or 'pour point' in ArcGIS. Typically, this point is a gauging station or some location where data is being or has been collected which allows for an empirical determination of baseflow values (see Section 2.8). Once a pour point has been defined this point bust be 'snapped' to the FlowAccumulation grid defined above in c) using the Spatial Analyst -> Hydrology toolbox. [One can create a grid of pour points either using the 'Snap Pour Points' tool or using the Spatial Analyst 'Feature to Raster' tool. In either case care must be taken to make sure that the proper spatial analyst options have been specified so that the grid of pour points created perfectly matches the flow accumulation grid.] Next the watershed can be defined using the 'Watershed' tool in the Spatial Analyst -> Hydrology toolbox. The process can be repeated for several watersheds to create multiple groundwater/baseflow basins within the modeling domain where each basin has an integer index value.

g. The final GIS processing step before data export to netcdf format is to re-classify or re-assign all grids so that they possess proper 'no-data' values. This can be accomplished using Spatial Analyst logical operators and/or the 'Reclassify' tool so that the grids EXACTLY follow the numbering convention shown in Figures 4.1 and 4.2. Not following these numbering conventions will likely result in runtime errors of WRF-Hydro such as a 'Segmentation Fault'.

h. Export of the newly created high-resolution terrain fields to netcdf is accomplished using the Spatial Analyst ->

Multidimensional Tools -> Raster to Netcdf tool. When using this tool the proper variable names for each data layer must be specified or else WRF-Hydro will not be able to properly input the data. The correct variable names for each data layer are as follows (these are case sensitive):

LATITUDE
LONGITUDE
TOPGRAPHY
FLOWDIRECTION
CHANNELGRID
STREAMORDER
LAKEGRID
frxst_pts
gw_basns
OVROUGHRTFAC
RETDEPRTFAC

[Note: OVROUGHRTFAC and RETDEPRTFAC default values are set to 1.0. These grids can either be created in the GIS or directly within the near final netcdf files using netcdf command operators ('nco') commands.]

Once all data layers have been properly exported to netcdf format, the final step is to merge or 'concatenate' all of the individual netcdf files into a single netcdf file. There are a variety of netcdf command operators (NCO commands) that can be utilized to do this and we have placed these into a simple c-shell script called '*concatenate.csh*' which is typically distributed with WRF-Hydro. Following the command line structure provided within the script the user simply needs to issue the script execution command followed by the list of individual netcdf files given as command line arguments and lastly an output file name. An example execution statement with command line arguments is given as follows:

csh concatenate.csh file1.nc file2.nc file3.nc outfile.nc

Upon successful completion of the *concatenate.csh* script the outfile.nc file will be created which contains all of the individual data layers specified in the command line. The most likely reason the *concatenate.csh* fails is because the individual netcdf files do not all have the exact same

dimension values. This serves as one valuable check to ensure that all of the high-resolution data layers have the same dimensions.

Helpful notes on high-resolution grid preparation:
1. The performance and computational stability of physically-based hydrologic simulation models is often dependent on the characteristics and quality of the underlying terrain fields upon which routing calculations are performed. Thus, anomalous artifacts in DEMs can have deleterious impacts on model stability, computational performance and/or simulation accuracy. Thus preparation of the high-resolution terrain fields can often be an iterative process where successive manipulations of the DEM can be made to reduce or eliminate large flat areas, change the density of the channel network or add/remove lakes/reservoirs. Thus when setting up a new domain, it is not uncommon to experiment with different thresholds and techniques during the DEM processing stages described above.
2. Large water bodies (the ocean, inland seas, etc) are somewhat distinct from typical lakes and reservoirs. For most applications, one large inland seas are treated as water sinks and not necessarily lakes/reservoirs and the oceans are always treated as sinks. These large water bodies are typically defined as 'water' grid points in the LSM. However, in the processing of the DEM data we have adopted a convention for specifying elevation where all ocean and large inland water bodies have either an elevation of 0 meters or a constant elevation of the mean lake surface elevation. Assigning these water body elevation values will help ensure the model does not crash since the model expects to find valid terrain elevation values everywhere in the simulation domain (i.e. there are no 'No Data' or -9999 values in the topography data layer).

vii. Channel Attributing

In addition to preparing the high-resolution data layers as described above, if one desires to perform channel routing in their simulation then parameters characterizing the channel network must be defined. These parameters are defined in a parameter file called CHANPARM.TBL which is placed in the run directory. These parameters are described in more detail in Sections 3.6 and 3.7 which describe the channel and lake routing schemes but they are also briefly defined here. The values are related, or 'indexed', to the channel network via stream order values which are defined above. Thus, in the present version of the model all stream pixel values of similar stream order will have identical channel parameter values. This assumption is likely not valid in many regions, and future

versions of the model will likely include spatially-distributed specifications of channel parameters that are assigned on the full high-resolution terrain grids. Nevertheless, the parameters in the CHANPARM.TBL file along with a range of possible values are defined as follows and in the Appendix A6:

```
Channel Parameters
StreamOrder
10,1,  'Bw      HLINK   ChSSlp   MannN'
1,     5.,     0.02,    1.0,      0.14
2,     10.,    0.02,    0.6,      0.12
3,     20.,    0.02,    0.3,      0.09
4,     30.,    0.03,    0.18,     0.09
5,     40.,    0.03,    0.05,     0.07
6,     60.,    0.03,    0.05,     0.06
7,     60.,    0.03,    0.05,     0.03
8,     60.,    0.10,    0.05,     0.03
9,     60.,    0.30,    0.05,     0.03
10,    60.,    0.30,    0.05,     0.03
```

where, the first column is the Strahler stream order, 'Bw' is the channel bottom width (unit of meters), 'HLINK' is the initial depth of water in the channel (unit of meters), 'ChSSlp' is the channel side slope (units of rise/run) and 'MannN' is the Manning's roughness coefficient for that stream order.

It is important to keep in mind that there is large uncertainty associated with these parameters. Therefore, model calibration is almost always warranted.

Also, because fully-distributed estimates of flow depth (HLINK) are not available for model initialization, it is almost always necessary to use a small initial value of HLINK and let the model come to its own equilibrium (i.e. 'spin-up') after several hours of integration.

viii.   Reservoir Attributing
In addition to preparing the high-resolution data layers as described above, if one desires to perform level-pool /lake reservoir routing in their simulation then parameters characterizing each reservoir must be defined. These parameters are defined in a parameter file called LAKEPARAM.TBL which is placed in the run directory. These parameters are described in more detail in Chapters 3.6 and 3.7 which describes the channel and lake routing schemes but they are also briefly defined here. If lake/reservoirs are not being simulated one can just assign bogus values for each parameter in the LAKEPARAM.TBL file. Failure to put any value in the file may result in crashing the model and receiving an I/O error message.

As described above and as shown in Fig. 4.2 each reservoir in the lake/reservoir data layer is assigned an integer index value ranging from 1-# of lake objects. The rest of the lake/reservoir parameters required for level-pool reservoir routing are as follows and in the Appendix:

```
lake        lake index (consecutively from 1 to n # of lakes)
LkArea      lake area (square meters)
LkMxH       elevation of maximum lake height(in meters MSL)
WeirC       weir coefficient
WeirL       weir length (units of meters)
OrificC     orifice coefficient
OrificeA    orifice area (units of square meters)
OrificeE    orifice elevation (units of meters MSL)
Lat         latitude of center of mass of lake (decimal degrees)
Long        latitude of center of mass of lake (decimal degrees)
Elevation   mean elevation of the lake surface (units of meters
            MSL)
```

These lake parameter values are specified for each one of the lake objects defined in the lake grid data layer contained within the high resolution terrain grid. Typically, several of these parameters are derived within the high-resolution terrain pre-processing stages described above using tools such as ArcGIS. Values for the weir and orifice coefficients and sizes can be drawn from standard engineering hydraulics textbooks (e.g. Chow et al., 1964). Weir parameters are specified for reservoir 'overflow' or 'spill' and orifice parameters are specified for design operations. Obviously, the behavior of the reservoir to store and release water is highly dependent on these parameters and that parameter values and reservoir operations data are often not available.

ix.   Groundwater/Baseflow Basin Attributing:

In addition to preparing the high-resolution data layers as described above, if one desires to include the representation of groundwater discharge/baseflow in their simulation then parameters characterizing the simple baseflow bucket model must be defined. These parameters are defined in a parameter file called GWBUCKPARM.TBL which is placed in the run directory. These parameters are described in more detail in Chapter 3.8 which describes the groundwater/baseflow scheme but they are also briefly defined here. The values in the table are related, or 'indexed', to each groundwater basin which is specified in the high-resolution terrain grid described above. The parameter values in GWBUCKPARM;TBL are defined as follows and in the Appendix A8:

```
Basin,Coeff.,Expon.,Zmax,Zinit
1,0.7760, 3.144, 0.100, 0.0982
2,0.0400, 3.220, 0.070, 0.0358
3,0.4270, 2.813, 0.125, 0.0678
4,0.0140, 5.861, 0.055, 0.0358
```

- this example assumes there are 4 individual groundwater basins or 'buckets' defined for this simulation domain

where, 'Coeff.' is the bucket model coefficient, 'Expon.' is the bucket model exponent, 'Zmax' is the conceptual maximum depth of the bucket and 'Zinit' is

the initial depth of water in the bucket model. It is important to remember that a simple bucket model is a highly abstracted and conceptualized representation of groundwater processes and therefore the depth of water values in the bucket have no real physical basis. Initial values of the groundwater bucket model parameters, particularly 'Zmax' and 'Zinit' are typically derived analytically or 'offline' from the WRF-Hydro and then are fine-tuned through model calibration. Full description of the procedure to derive initial groundwater bucket model parameters are presented in Chapter 3.8.

4.3 Description of meteorological forcing data input files
Modern land surface hydrology models require meteorological forcing data to simulate land-atmosphere exchanges and terrestrial hydrologic processes. Most land models use more or less the same variables with some variations for units, spectral bandwidths of radiation, phase of precipitation or biogeochemical constituents. Most commonly these variables include: incoming short and longwave radiation, humidity, temperature, pressure, wind speed and precipitation. Each land model will have some specific requirements regarding the format and exact units and variables. For Version 1.0 of WRF-Hydro the system requires those variables that are required to drive the Noah land surface model and those variables along with their units are listed in Table 4.1. When WRF-Hydro is coupled into other modeling architectures such as the NCAR Community Earth System Model (CESM) or the NASA Land Information system, those systems will set the requirements for the forcing data. Here we simply describe the requirements and options that are available in the stand-alone version of WRF-Hydro. As new hydrological components are added to WRF-Hydro, this section will be updated to provide model specific requirement and availability information.

Table 4.1 Input forcing data for the Noah LSM

Incoming shortwave radiation	(W/m2)
Incoming longwave radiation	(W/m2)
Specific humidity	(kg/kg)
Air temperature	(K)
Surface pressure	(Pa)
Near surface wind in the u- and v-components	(m/s)
Liquid water precipitation rate	(mm/s)

[NOTE: Different land surface models may require other or additional forcing variables or the specification of forcing variables in different units.]

When coupled to the WRF regional atmospheric model the forcing data is provided by the atmospheric model with a frequency dictated by the land surface model time-step specified in WRF. Therefore when running WRF-Hydro in a 'coupled' mode with WRF, there is no need to prepare forcing data.

When run in a stand-alone mode, these forcing data must be provided as gridded input data. Presently, there are 6 forcing data input options in WRF-Hydro. Because it is untenable to support a large variety of input file formats and data types within the model WRF-Hydro requires that most processing of forcing data be handled external to the model (i.e. as a 'pre-process') and that users get their forcing into one of the required formats. This includes performing tasks like, gridding of station observations, making sure forcing data is on the appropriate grid and has the correct variable name and units, getting data into the prescribed netcdf format, etc. To facilitate these pre-processing activities we have developed numerous scripts which can be executed to help in the

forcing data preparation process. These scripts are located in the 'utils/' directory and are described below.

The input forcing data type is specified in the 'namelist.hrldas' input file with the parameter name 'FORC_TYP' as follows:

!Specification of forcing data: 1=HRLDAS-hr format, 2=HRLDAS-min format, 3=WRF, 4=Idealized, 5=Ideal w/ Spec.Precip., 6=HRLDAS-hrly format w/ Spec. Precip
 FORC_TYP = 4

(in this example, the forcing data type is set to 4, which is the 'idealized' forcing data option.)

The six forcing data input options are as follows:

1 – HRLDAS hourly input files: All meteorological variables are packed into one netcdf input file for each time with a filename of the form: 2011071300.LDASIN_DOMAIN2

2 - HRLDAS minute format input files: All meteorological variables are packed into one netcdf input file for each time with a filename of the form: 201107130025.LDASIN_DOMAIN2. This format is often used when there is high-time resolution data available.

3 – WRF: This option simply reads a WRF model output file ('wrfout' file) and extracts the appropriate fields for driving the offline WRF-Hydro model. The necessary fields are available in a default wrf output file but users should verify their existence if modifications have been made to the wrf output files. The names of the variables in the wrfout file differs from those of the standard HRLDAS input file. Users need not worry about this as the WRF-Hydro code knows what variable name to look for in wrfout files. Lastly, this option requires that the wrfout grid be exactly the same as the WRF-Hydro grid. The WRF-Hydro code will not remap or spatially-subset the wrfout data in any way.

4 – Idealized: This option is the most simple method to force the model and requires no input files. A simple rainfall event is prescribed (i.e. 'hardwired') in the model of 25.4 inches per hour (1 inch per hour) for 1 hour duration. The event starts on timestep (hour) The rest of the forcing data variables are set to have either constant values (in space and time) or, in the case of temperature and radiation variables, a fixed diurnal cycle. This option is mainly used for simple testing of the model and is convenient for checking whether or not components besides the forcing data are properly being read into the model and working. Version 1.0 of WRF-Hydro has hardwired values of these forcing data terms. Future version will allow the user to input default values for the precipitation event and the other meteorological variables.

5 – Idealized with Specified Precipitation: This option is identical to option 4 except that the WRF-Hydro system will look for a gridded netcdf precipitation file. The filename format of this file is: 201107141705.LDASIN_PRECIP_DOMAIN2.   When using this

option, the WRF-Hydro system will look for a new precipitation input file based on the FORCING DATA TIMESTEP namelist parameter in the namelist.hrldas file.

6 – Hourly HRLDAS input file with Specified precipitation: This option combines options 1 and 5 in that an hourly HRLDAS input file is used for all meteorological forcing variables except precipitation and that precipitation is read in from a precipitation input file as described in option 5 above. This option is very useful when combining atmospheric analyses from re-analysis products or other models with a separate analysis of precipitation (e.g. a gridded gauge product, radar QPE, nowcasts, satellite QPE, etc). The model reads in each meteorological forcing data field on each hour and then holds those values constant for the entire hour. Precipitation data is then read in based on the user-specified FORCING DATA TIMESTAMP namelist parameter in the namelist.hrldas file. Thus, for example, the user can have 'hourly' meteorology with '5-minute' precipitation analyses. The filename formats for these two different input files are:

2011071300.LDASIN_DOMAIN2 and
201107141705.LDASIN_PRECIP_DOMAIN2

An example of what the netcdf file headers for the HRLDAS input file and the 'specified precipitation' input file look like are provided in Appendix A11.

4.4 Description of Output Files from WRF-Hydro

       i. Land surface model output (YYYYMMDDHHMM.LDASOUT_DOMAINX)

When the Noah LSM is used (current option available), model output on the land surface model grid is written to a multi-dimensional netcdf data file using the file naming convention **'YYYYMMDDHHMM.LDASOUT_DOMAINX'.**

where:
YYYY – year
MM – month
DD – day
HH – hour
MM – minutes
DOMAINX – the domain number that is specified in the hydro.namelist input file (also matches the domain number of the geogrid input file).

The west_east and north_south dimensions of the output file match those of the geogrid input file. Model output is created for every model time step. However, the length of the time dimension in each netcdf output data file can vary depending on the value of 'SPLIT_OUTPUT_COUNT' specified in the namelist.hrldas input file.

The names and definitions for each output variable in the LSM output file are generally consistent with those output from standard Noah LSM coupled to WRF. An example header of the netcdf output file is provided in the Appendix A12.

       ii. Output file description Routing output

Terrain Routing
When routing modules are activated additional output datasets are created. Here we distinguish these datasets between those that are created when only the terrain (overland and/or subsurface) routing is activated versus those datasets that are created when channel routing is activated. For all datasets, one output is provided for each LSM time-step, not each routing model time-step. These output data include the following:

    a. 'chan_inflow.txt': This is an ASCII formatted, timeseries data file of the channel network total accumulated, channel inflow (in units of cubic meters). Essentially, this value is the volume of water that is moving into the entire channel network from overland flow. A switch in the noah_namelist file (HIRES_OUTPUT = 1) activates the generation of this output. An example of 'chan_inflow.txt' is provided in the Appendix A18.

    b. (Optional) Netcdf high-resolution terrain grid output file: A gridded dataset of selected variables on the high-resolution terrain grid output to a netcdf file using the following file naming convention:

YYYYMMDGHHMM.RTOUT_DOMAINX, where the convention terms are defined above in the LSM output file description. Due to the shear size of these data layers, care should be used in deciding when to output high-resolution terrain data and which variables to output. Users desiring this output presently need to edit the source code and remove the comments blocking the call to the subroutine to perform this output. Default output variables provided in this file are listed in the 'RTOUT_header.txt' file in the Appendix (A13) and include:

LATITUDE
LONGITUDE
SOIL_M : Volumetric soil moisture content (units of m^3/m^3)
ZWATTABLRT : Depth to saturated layers where saturated subsurface
       routing may be occurring. This value will equal the total soil
       column depth (typically 2m) when no saturation is occurring.
       (units of m)
QSTRMVOLRT : Accumulated depth of stream channel inflow (units of
       mm)
SFCHEADSUBRT : Instantaneous value of depth of water ponded on the
       surface (units of mm)

Additional variables can be added to this file through changes in source code. As with the LSM output datafiles, the number of time slices per datafile are controlled through specification of the 'SPLIT_OUTPUT_COUNT' parameter specified in the hydro.namelist input file.

Channel Routing (only output when channel routing is active)
a. 'frxst_pts.txt': This is an ASCII-formatted data file that provides time series of streamflow discharge at selected 'forecast' points along the channel network. The forecast points are specified within the 'frxst_pts' data layer contained within the high-resolution terrain netcdf data file. Points are listed in sequential order by station index (numbered 1 to n # of stations) and by time. Many users will find this data file and format useful for streamflow forecasting and model calibration. An example of 'frxst_pts.txt' is provided in the Appendix (A17). The format for each data record is as follows:

column 1 : time (in seconds) into simulation
column 2 : date and time as YYYY-MM-DD_HH:MM:SS
column 3 : station number index (numbered 0 to n-1 stations)
column 4 : station longitude (in decimal degrees)
column 5 : station latitude (in decimal degrees)
column 6 : streamflow discharge (in cubic meters per second)
column 7 : streamflow discharge (in cubic feet per second)

column 8 : flow depth/river stage (in meters above channel bottom)

b. Netcdf forecast point output :
(YYYYMMDGHHMM_CHANOBS_DOMAINX)
This is a netcdf point file which contains streamflow discharge, flow depth (i.e. 'stage' or 'head'), longitude, latitude, forecast point index and stream order value for each forecast point specified in the 'frxst_pts' data layer of the high-resolution terrain data file. It ONLY contains data from the forecast points and, thus, is small in file size. Users wishing to overlay numeric streamflow values on top of other data layers such as topography or precipitation in data visualization tools (e.g. ArcGIS, matlab, IDV, etc.) will find this format useful. As with the LSM output datafiles, the number of time slices per datafile are controlled through specification of the 'SPLIT_OUTPUT_COUNT' parameter specified in the hydro.namelist input file. An example of the netcdf data file header is provided in the Appendix (A14).

c. Netcdf full channel point output
(YYYYMMDGHHMM_CHRTOUT_DOMAINX)
This is a netcdf point file which is identical to the forecast point netcdf data file except that it contains streamflow discharge, flow depth (i.e. 'stage' or 'head'), longitude, latitude, forecast point index and stream order value for all channel pixels within the high-resolution terrain data file. Because it contains data from all channel pixels within the high-resolution terrain grid, file sizes can become quite large. Users wishing to overlay spatially continuous numeric streamflow values on top of other data layers such as topography or precipitation in data visualization tools (e.g. ArcGIS, matlab, IDV, etc.) will find this format useful. As with the LSM output datafiles, the number of time slices per datafile are controlled through specification of the 'SPLIT_OUTPUT_COUNT' parameter specified in the hydro.namelist input file. An example of the netcdf data file header is provided in the Appendix (A15).

Lake/Reservoir Output
Netcdf lake point output (YYYYMMDGHHMM_LAKES_DOMAINX)
When one or more lakes/reservoirs are specified within the LAKEGRID data layer of the high-resolution netcdf input file, a netcdf point data file is created which contains values of several state and flux variables to/from the lake/reservoir. An example of the header from the LAKEGRID netcdf file is provided in the Appendix (A16) and some of the most commonly used variable names are defined as follows:

ELEVATION : elevation of the lake/reservoir water surface (units of meters above MSL)

INFLOW : total inflow to the reservoir from all channel tributaries intersecting the lake/reservoir (units of m^3/s)

OUTFLOW : total outflow from the reservoir to a specified outlet on the channel network (units of m^3/s)

STATION_ID : integer index of the lake numbered from 0 to n-1 lakes/reservoirs as specified in the high-resolution terrain input file.

Groundwater flux files:

When the groundwater/baseflow bucket model is activated, three additional ASCII data files are created which help characterized the state of (bucket depth) and fluxes (input and output) to/from the conceptual bucket. Each datafile contains time series of values where output values are provided for each groundwater basin. The format of each datafile is described by the following:

a. GW_inflow.txt : Contains time-step values of drainage fluxes from the soil column to the groundwater bucket integrated over each groundwater basin specified in the high-resolution terrain input file. Units of these fluxes are in meters^3. See Appendix (A19).

b. GW_outflow.txt : Contains time-step values of groundwater/baseflow bucket discharge fluxes from the bucket to the channel network. Discharge flux values are a single value for each groundwater basin specified in the high-resolution terrain input file. As described in the Section 2.8 the bucket discharge values are spatially distributed across all channel pixels contained within each groundwater basin. Units of these fluxes are m^3/s. See Appendix (A20)

c. GW_zlev.txt : Contains time-step values of the conceptual depth of water within each bucket model. For the purposes of mass conservation units of these fluxes are meters. However, as mentioned previously, these values are conceptual, are specific to the calibrated parameters of each groundwater/baseflow bucket and do not reflect an actual depth of water in any 'natural' aquifer system. See Appendix (A21).

5. Example Use Cases:

5.1 Overview

In this section we will briefly describe the test cases that are distributed with the WRF-Hydro system. Presently we have 3 test cases as follows:

1. Fourmile Creek: This is a small domain (20km x 20km) test case that is useful for testing the installation of the model and in doing model benchmarking and mass balance checking after installation and model development.

2. Colorado Front Range: This is a medium-scale (260km x 268km) test case which illustrates the application of the WRF-Hydro system over a fairly extensive and heterogeneous hydrological environment, namely the mountain front region of the Colorado Rocky Mountains. This test case provides example implementations of nearly all model options. Both offline (i.e. not coupled to WRF) and fully-coupled implementations are shown.

3. Genoa-Italy Flood: This is another medium-scale (450km x 450km) test case that illustrates the application of the WRF-Hydro system, in an offline mode, for hydrologic prediction of an extreme rainfall event over Genoa, Italy in Nov. of 2011. This test shows implementations of an international (i.e. non-U.S.) domain, and in the use of WRF model output as forcing for an offline implementation of WRF_Hydro.

Each of these test cases, and future test cases to be developed, are contained as gzipped tarfiles in the /test_cases/ directory of the WRF_Hydro extension package or can be downloaded as an individual tarfile from the WRF_Hydro 'User Support' web site:

http://www.ral.ucar.edu/projects/wrf_hydro/support.php

5.2 Uncoupled simple single catchment benchmark with idealized forcing

As mentioned above, this is a small domain (20km x 20km) test case that is useful for testing the installation of the model and in doing model benchmarking and mass balance checking after installation and model development. All data for executing idealized runs over this domain are contained in the tarfile:

/test_cases/Fourmile_test_case.tar

The contents of this tarfile are as follows:

CHANPARM.TBL	: Channel parameter table
GENPARM.TBL	: Noah LSM general parameter table
GWBUCKPARM.TBL	: Groundwater bucket model parameter table
HYDRO.TBL	: Hydraulic parameters used in WRF-Hydro
LAKEPARM.TBL	: Lake model parameter table
SOILPARM.TBL	: Noah LSM soil parameter table
URBPARM.TBL	: Noah LSM urban canopy parameter table
VEGPARM.TBL	: Noah LSM vegetation parameter table
Fulldom_hires_hydrofile_4mile_benchmark.nc	: High resolution routing data file
geo_em.d01.nc.4mile_100m.nc	: Noah LSM surface data (100m grid spacing)
geo_em.d01.nc.4mile_1km.nc	: Noah LSM surface data (1km grid spacing)
wrfinput_d01	: Noah LSM initialization file
hydro.namelist	: Routing namelist file
hydro.namelist.100mLSM	: Routing namelist file (100m grid spacing)
hydro.namelist.1kmLSM	: Routing namelist file (1km grid spacing)
namelist.hrldas	: HRLDAS namelist file
namelist.hrldas.100mLSM	: HRLDAS namelist file (100m grid spacing)
namelist.hrldas.1kmLSM	: HRLDAS namelist file (1km grid spacing)
Noah_hrldas_beta	: Noah LSM/WRF_Hydro executable file
HYDRO_RST.2011-08-12_12:00_DOMAIN1_init.nc	: Sample routing grid restart file
RESTART.2011081212_DOMAIN1_init.nc	: Sample HRLDAS restart file
test_torque.csh	: Sample parallel job submit script
water_budget_Noah_LSM_only.ncl	: ncl mass balance/budget analysis script (no routing)
water_budget_Noah_LSM_terrain_routing.ncl	: ncl mass balance/budget analysis script (with routing)
filename_chngr.sh	: script to append '.nc' to model output for ncl script

The default implementation of the Fourmile Creek test case is a 30 day simulation using the 'idealized' forcing data specification (FORC_TYP=4) in the namelist.hrldas file with both the Noah LSM and the terrain and channel routing processes executed on identical 100m grid spacing grids. The idealized meteorological forcing is described in the main WRF_Hydro Technical Document and User Guide in Chapter 4.3. The model is set to run from a 'restart' condition since the namelist options for both the HRLDAS namelist file and the routing namelist file are uncommented (i.e. there is no '!' before the filename.) In this default case all data paths are specified to be the local directory where the model is executed. The default simulation has 1-d surface overland flow, saturated sub-surface flow and channel routing flow all activated in the routing namelist file. For descriptions of the namelist parameters in both files please refer to the User Guide, Chapter 2.6 and Appendices A1 and A2.

The commands to execute the simulation are:

For sequential, single processor compilations of the model:

% Noah_hrldas_beta

and for parallel, distributed memory compilations of the model: (specific formats of this command will vary depending on operating system configurations and parallel job management software)

%mpirun –np 16 Noah_hrldas_beta

When the model is completed running there will be a host of output files created for several of the different processes activated. For this simple default test case, all output options are active, including the production of a routing grid output file. Options to control output are described in the main User Document in Chapter 2.6 and Appendices A1 and A2.

A simple mass balance/water budget analysis can be performed on the output from the pre-configured model run. Running through the simple analysis is a good way to ensure that the model is producing credible output and that the mass balance closure over the whole domain is reasonable.

The table below lists the mass balance/water budget terms from the simple Fourmile Creek test case:

Variable	No routing	Overland Flow	Subsurface flow	Overland and Subsurface Flow
ACRAIN (mm)	25.4	25.4	25.4	25.4
DEWFALL (mm)	0	0	0	0
SFCEVP (mm)	15.48683	18.58462	15.48684	18.58462
qstrmvol (hires - mm)		1.359645		1.359645
sfcrunoff (mm)	17.66775		17.66775	
ugrdrnff (mm)	6.274211	9.564176	6.27422	9.564176
intrflow (mm)	0	0	0	0
del soilwat (mm)	-14.02878	-7.272209	-14.0288	-7.272209
del canwat (mm)	0	0	0	0
del weasd (m)	0	0	0	0
del sfc head (hires - mm)		8.24E-06	0	8.24E-06
qbdry sfc (hires - mm)		3.172269	0	3.172269
qbdry subsfc (hires - mm)				
Script value resid	-1.33E-05	-0.00850749	-0.00001	-0.0085075

The variables listed in the variable column are the principle water balance terms output from the model and the mass balance/water budget script. Note the channel routing component is not included in this simple analysis. Nevertheless, this simple simulation and analysis shows how a 1-hr rainfall event of 25.4 mm (i.e. 1 inch per hour for one hour) is partitioned into various runoff components when different model options are selected. All units are shown in mm and the residual values shown in the bottom line of the table indicate that the model conserves water to a few thousandths of a millimeter in a basin average sense but that this residual or 'closure' error does change with respect to the model options selected.

Finally, we would like to reiterate that this simple test case is useful for those seeking to do model development work as it provides a baseline implementation and mass balance check on the modeling system. New model enhancements should be verified against this or similarly constructed water balance analyses to ensure conservation of water mass is maintained.

5.3 Uncoupled real world flash flood event with a continuous spin-up

[UNDER DEVELOPMENT]

Colorado Front Range: This is a medium-scale (260km x 268km) test case which illustrates the application of the WRF-Hydro system over a fairly extensive and heterogeneous hydrological environment, namely the mountain front region of the Colorado Rocky Mountains. This test case provides example implementations of nearly all model options. Both offline (i.e. not coupled to WRF) and fully-coupled implementations are shown.

5.4 Fully-coupled real-world event

[UNDER DEVELOPMENT]

June 11-12, 2010 Front Range…

6. WRF-Hydro Utility Scripts: [UNDER DEVELOPMENT]

6.1 Overview

6.2 Catalog of Scripts:

REFERENCES   [UNDER DEVELOPMENT]

APPENDICES

The appendices below contain examples of the namelist files, parameter files, input files and output files used in WRF-Hydro. Where relevant short descriptions of what is contained within the files is provided.

A1. Noah HRLDAS model namelist description (namelist.hrldas)

&NOAHLSM_OFFLINE

!!!! MODEL INITIALIZATION DATA FILE !!!
 HRLDAS_CONSTANTS_FILE = "wrfinput_d01"

!!!! MODEL FORCING DATA INPUT DIRECTORY !!!
 INDIR = "./forcing/FRNG/2010_2011/CHILL_QPE_grids"

!!!! MODEL OUTPUT DIRECTORY (OPTIONAL) !!!
 ! OUTDIR = "./hrldas_output/"

!!!! MODEL START DATE & TIME !!!
 START_YEAR = 2011
 START_MONTH = 07
 START_DAY   = 13
 START_HOUR = 12
 START_MIN   = 00

!!!! MODEL RESTART FILE (OPTIONAL) !!!
RESTART_FILENAME_REQUESTED = "./RESTART.2011081212_DOMAIN1_init.nc"

!!!! MODEL DURATION !!!
! KDAY = 720
  KHOUR = 720

!!!! MODEL TIMESTEP INFORMATION !!!
 FORCING_TIMESTEP = 3600
 NOAH_TIMESTEP   = 3600
 OUTPUT_TIMESTEP = 43200

!!!! MODEL RESTART FILE WRITE FREQUENCY (9999=MONTHLY) !!!
! RESTART_FREQUENCY_HOURS = 99999 ! 480
 RESTART_FREQUENCY_HOURS = 24 ! 480

!!!! NUMBER OF OUTPUT TIMES PER OUTPUT FILE !!!
! Split output after split_output_count output times.
! SPLIT_OUTPUT_COUNT = 240
 SPLIT_OUTPUT_COUNT = 1

!!!! SUBWINDOW OF FULL MODEL DOMAIN (OPTIONAL) !!!
! SUBWINDOW_XSTART = 32
! SUBWINDOW_XEND = 32
! SUBWINDOW_YSTART = 60
! SUBWINDOW_YEND = 60

!!!! SOIL LAYER INFORMATION !!!

```
 NSOIL=4
 ZSOIL(1) = -0.10
 ZSOIL(2) = -0.40
 ZSOIL(3) = -1.00
 ZSOIL(4) = -2.00

!!!! HEIGHT OF FORCING DATA: ZLVL = TEMPERATURE & HUMIDITY !!!
 ZLVL = 2.0
 ZLVL_WIND = 10.0

!!!! NOAH MODEL OPTIONS (SEE NOAH LSM DOCUMENTATION) !!!
 IZ0TLND = 0
 SFCDIF_OPTION = 0
 UPDATE_SNOW_FROM_FORCING = .FALSE.

!!!! FORCING DATA OPTION !!!
!Specification of forcing data:  1=HRLDAS-hr format, 2=HRLDAS-min format, 3=WRF,
4=Idealized, 5=Ideal w/ Spec.Precip., 6=HRLDAS-hrly format w/ Spec. Precip
 FORC_TYP = 4

!!!! OPTION TO UPDATE MODEL SNOWPACK FROM FORCING DATA !!!
!Switch for snow data assimilation: 0=no, 1=yes
SNOW_ASSIM = 0

!!!! WRF GEOGRID FILE FOR SURFACE INPUT INFORMATION !!!
! for extract greenfrac
GEO_STATIC_FLNM = "./geo_em.d01.nc.4mile_100m.nc"

!!!! SPECIFY WHERE TO GET INITIALIZATION DATA FROM !!!
!HRLDAS_ini_typ 1: initial and parameters from forcing else from wrfinput.
HRLDAS_ini_typ = 1

/

!!!! NOAH URBAN MODEL OPTIONS (ONLY USED WITH NOAH URBAN CANOPY !!!
&URBAN_OFFLINE
 SF_URBAN_PHYSICS = 0
 ZLVL_URBAN = 15.0
/
```

A2. WRF-Hydro model namelist description (hydro.namelist)

&HYDRO_nlist

!!!! SYSTEM COUPLING !!!!
!Specify what is being coupled: 1=HRLDAS (offline Noah-LSM), 2=WRF, 3=NASA/LIS, 4=CLM
 sys_cpl = 1

!!!! MODEL INPUT DATA FILES !!!
!Specify land surface model gridded input data file...(e.g.: "geo_em.d03.nc")
 GEO_STATIC_FLNM = "./geo_em.d01.nc.4mile_100m.nc"

!Specify the high-resolution routing terrain input data file...(e.g.: "Fulldom_hires_hydrofile.nc"
 GEO_FINEGRID_FLNM = "./Fulldom_hires_hydrofile_4mile_benchmark.nc"

!Specify the name of the restart file if starting from restart...comment out with '!' if not...
 RESTART_FILE = 'HYDRO_RST.2011-08-12_12:00_DOMAIN1_init.nc'

!!!! MODEL SETUP AND I/O CONTROL !!!!
!Specify the domain or nest number identifier...(integer)
 IGRID = 1

!Specify the restart file write frequency...(minutes)
 rst_dt = 14400

!Specify the output file write frequency...(minutes)
 out_dt = 1440 ! minutes

!Specify if output history files are to be written...(.TRUE. or .FALSE.)
 HISTORY_OUTPUT = .TRUE.

!Specify the number of output times to be contained within each output history file...(integer)
!   SET = 1 WHEN RUNNING CHANNEL ROUTING ONLY/CALIBRATION SIMS!!!
!   SET = 1 WHEN RUNNING COUPLED TO WRF!!!
 SPLIT_OUTPUT_COUNT = 1

! rst_typ = 1 : overwrite the soil variables from routing restart file.
 rst_typ = 1

!Restart switch to set restart accumulation variables = 0 (0-no reset, 1-yes reset to 0.0)
 RSTRT_SWC = 1

!Output high-resolution routing files...0=none, 1=total chan_inflow ASCII time-series, 2=hires grid and chan_inflow...
 HIRES_OUT = 2

!Specify the minimum stream order to output to netcdf point file...(integer)
!Note: lower value of stream order produces more output.
 order_to_write = 1

!!!! PHYSICS OPTIONS AND RELATED SETTINGS !!!!
!Switch for terrain adjustment of incoming solar radiation: 0=no, 1=yes
!Note: This option is not yet active in Verion 1.0...
!   WRF has this capability so be careful not to double apply the correction!!!
 TERADJ_SOLAR = 0

!Specify the number of soil layers (integer) and the depth of the bottom of each layer (meters)...
! Notes: In Version 1 of WRF-Hydro these must be the same as in the namelist.input file
!   Future versions will permit this to be different.
 NSOIL=4
 ZSOIL8(1) = -0.10
 ZSOIL8(2) = -0.40
 ZSOIL8(3) = -1.00
 ZSOIL8(4) = -2.00

!Specify the grid spacing of the terrain routing grid...(meters)
 DXRT = 100

!Specify the integer multiple between the land model grid and the terrain routing grid...(integer)
 AGGFACTRT = 1

!Specify the routing model timestep...(seconds)
 DTRT = 6

!Switch activate subsurface routing...(0=no, 1=yes)
 SUBRTSWCRT = 1

!Switch activate surface overland flow routing...(0=no, 1=yes)
 OVRTSWCRT = 1

!Switch to activate channel routing Routing Option: 1=Seepest Descent (D8) 2=CASC2D
 CHANRTSWCRT = 1
 rt_option   = 1

!Specify channel routing option: 1=Muskingam-reach, 2=Musk.-Cunge-reach, 3=Diff.Wave-gridded
 channel_option =3

!Specify the reach file for reach-based routing options...
 route_link_f = ""

!Switch to activate baseflow bucket model...(0=none, 1=exp. bucket, 2=pass-through)
 GWBASESWCRT = 0

!Specify baseflow/bucket model initialization...(0=cold start from table, 1=restart file)
 GW_RESTART = 0

!Groundwater/baseflow mask specified on land surface model grid...
 !Note: Only required if baseflow bucket model is active
 gwbasmskfil = "./gw_basn1k.txt"

/

A3. Vegetation parameter table (VEGPARM.TBL)

User's needing information about the data in the VEGPARM.TBL file need to refer to the documentation for the Noah land surface model.

http://www.ral.ucar.edu/research/land/technology/lsm.php

A4. Soil parameter table (SOILPARM.TBL)

User's needing information about the data in the SOILPARM.TBL file need to refer to the documentation for the Noah land surface model.

http://www.ral.ucar.edu/research/land/technology/lsm.php

A5. General parameters table (GENPARM.TBL)

User's needing information about the data in the GENPARM.TBL file need to refer to the documentation for the Noah land surface model.

http://www.ral.ucar.edu/research/land/technology/lsm.php

A6. Channel parameters table (CHANPARM.TBL)

```
CHANPARM.TBL file:


    Channel Parameters
    StreamOrder
    10,1,  'Bw      HLINK    ChSSlp    MannN'
    1,      5.,     0.02,     1.0,       0.14
    2,     10.,     0.02,     0.6,       0.12
    3,     20.,     0.02,     0.3,       0.09
    4,     30.,     0.03,    0.18,       0.09
    5,     40.,     0.03,    0.05,       0.07
    6,     60.,     0.03,    0.05,       0.06
    7,     60.,     0.03,    0.05,       0.03
    8,     60.,     0.10,    0.05,       0.03
    9,     60.,     0.30,    0.05,       0.03
    10,    60.,     0.30,    0.05,       0.03
```

where, the first column is the Strahler stream order, 'Bw' is the channel bottom width (unit of meters), 'HLINK' is the initial depth of water in the channel (unit of meters), 'ChSSlp' is the channel side slope (units of rise/run) and 'MannN' is the Manning's roughness coefficient for that stream order.

It is important to keep in mind that there is large uncertainty associated with these parameters. Therefore, model calibration is almost always warranted.

Also, because fully-distributed estimates of flow depth (HLINK) are not available for model initialization, it is almost always necessary to use a small initial value of HLINK and let the model come to its own equilibrium (i.e. 'spin-up') after several hours of integration. The necessary time required to spin up the channel network is a direct function of how dense and long your channel network is. Larger, more dense networks will take substantially longer to spin up. Estimates of total travel time from the furthest channel element to the basin outline are a reasonable initial approximation of the time it will take to spin up the channel elements.

A7. Lake parameters table (LAKEPARM.TBL)

```
LAKEPARM.TBL
```

lake	LkArea	LkMxH	WeirC	WeirL	OrificC	OrificeA	OrificeE
	lat	long		elevation			
1	9.67	1752.1	0.4	12.1	0.1	1.0	1664.4
	40.5580	-105.1586		1752.1			
2	3.07	1530.8	0.4	3.8	0.1	1.0	1519.6
	40.4407	-105.0586		1530.8			
3	1.61	1537.7	0.4	2.0	0.1	1.0	1528.7
	40.4158	-105.0903		1537.7			
4	1.11	1554.6	0.4	1.4	0.1	1.0	1544.4
	40.3876	-105.1441		1554.6			
5	3.82	1785.1	0.4	4.8	0.1	1.0	1758.2
	40.3377	-105.2196		1785.1			
6	1.36	1569.5	0.4	1.7	0.1	1.0	1565.6
	40.3378	-105.1278		1569.5			
7	1.47	1571.1	0.4	1.8	0.1	1.0	1565.3
	40.3297	-105.1167		1571.1			

- this example assumes there are 7 lakes defined within the simulation domain (note column wrapping…)

where,

```
lake        lake index (consecutively from 1 to n # of lakes)
LkArea      lake area (square meters)
LkMxH       elevation of maximum lake height(in meters MSL)
WeirC       weir coefficient
WeirL       weir length (units of meters)
OrificC     orifice coefficient
OrificeA    orifice area (units of square meters)
OrificeE    orifice elevation (units of meters MSL)
Lat         latitude of center of mass of lake (decimal degrees)
Long        latitude of center of mass of lake (decimal degrees)
Elevation   mean elevation of the lake surface (units of meters
            MSL)
```

These lake parameter values are specified for each one of the lake objects defined in the lake grid data layer contained within the high resolution terrain grid. Typically, several of these parameters are derived within the high-resolution terrain pre-processing stages described above using tools such as ArcGIS. Values for the weir and orifice coefficients and sizes can be drawn from standard engineering hydraulics textbooks (e.g. Chow et al., 1957). Weir parameters are specified for reservoir 'overflow' or 'spill' and orifice parameters are specified for design operations. Obviously, the behavior of the reservoir to store and release water is highly dependent on these parameters and that parameter values and reservoir operations data are often not available.

A8. Groundwater/baseflow bucket model parameters table (GWBUCKPARM.TBL)

GWBUCKPARM.TBL

```
Basin,Coeff.,Expon.,Zmax,Zinit
1,0.7760, 3.144, 0.100, 0.0982
2,0.0400, 3.220, 0.070, 0.0358
3,0.4270, 2.813, 0.125, 0.0678
4,0.0140, 5.861, 0.055, 0.0358
```

- this example assumes there are 4 individual groundwater basins or 'buckets' defined for this simulation domain

where, 'Coeff.' is the bucket model coefficient, 'Expon.' is the bucket model exponent, 'Zmax' is the conceptual maximum depth of the bucket and 'Zinit' is the initial depth of water in the bucket model. It is important to remember that a simple bucket model is a highly abstracted and conceptualized representation of groundwater processes and therefore the depth of water values in the bucket have no real physical basis. Initial values of the groundwater bucket model parameters, particularly 'Zmax' and 'Zinit' are typically derived analytically or 'offline' from WRF-Hydro and then are fine-tuned through model calibration.

A9. Terrestrial hydrological hydraulic parameters table (HYDRO.TBL)

HYDRO.TBL file:

```
    27 USGS for OV_ROUGHSFC_ROUGH'
    0.025,    'Urban and Built-Up Land'
    0.035,    'Dryland Cropland and Pasture'
    0.035,    'Irrigated Cropland and Pasture'
    0.055,    'Mixed Dryland/Irrigated Cropland and Pasture'
    0.035,    'Cropland/Grassland Mosaic'
    0.068,    'Cropland/Woodland Mosaic'
    0.055,    'Grassland'
    0.055,    'Shrubland'
    0.055,    'Mixed Shrubland/Grassland'
    0.055,    'Savanna'
    0.200,    'Deciduous Broadleaf Forest'
    0.200,    'Deciduous Needleleaf Forest'
    0.200,    'Evergreen Broadleaf Forest'
    0.200,    'Evergreen Needleleaf Forest'
    0.200,    'Mixed Forest'
    0.005,    'Water Bodies'
    0.070,    'Herbaceous Wetland'
    0.070,    'Wooded Wetland'
    0.035,    'Barren or Sparsely Vegetated'
    0.055,    'Herbaceous Tundra'
    0.055,    'Wooded Tundra'
    0.055,    'Mixed Tundra'
    0.055,    'Bare Ground Tundra'
    0.010,    'Snow or Ice'
    0.010,    'Playa'
    0.100,    'Lava'
    0.010,    'White Sand'
19, for SATDK
SATDK      MAXSMC     REFSMC     WLTSMC    QTZ      '
1.07E-6,   0.339,     0.236,     0.010,    0.92,  'SAND'
1.41E-5,   0.421,     0.383,     0.028,    0.82,  'LOAMY SAND'
5.23E-6,   0.434,     0.383,     0.047,    0.60,  'SANDY LOAM'
2.81E-6,   0.476,     0.360,     0.084,    0.25,  'SILT LOAM'
2.81E-6,   0.476,     0.383,     0.084,    0.10,  'SILT'
3.38E-6,   0.439,     0.329,     0.066,    0.40,  'LOAM'
4.45E-6,   0.404,     0.314,     0.067,    0.60,  'SANDY CLAY LOAM'
2.04E-6,   0.464,     0.387,     0.120,    0.10,  'SILTY CLAY LOAM'
2.45E-6,   0.465,     0.382,     0.103,    0.35,  'CLAY LOAM'
7.22E-6,   0.406,     0.338,     0.100,    0.52,  'SANDY CLAY'
1.34E-6,   0.468,     0.404,     0.126,    0.10,  'SILTY CLAY'
9.74E-7,   0.468,     0.412,     0.138,    0.25,  'CLAY'
3.38E-6,   0.439,     0.329,     0.066,    0.05,  'ORGANIC MATERIAL'
   0.0,    1.0,       0.0,       0.0,      0.60,  'WATER'
1.41E-4,   0.20,      0.170,     0.006,    0.07,  'BEDROCK'
1.41E-5,   0.421,     0.283,     0.028,    0.25,  'OTHER(land-ice)'
9.74E-7,   0.468,     0.454,     0.030,    0.60,  'PLAYA'
```

```
1.41E-4,  0.200,    0.170,    0.006,  0.52, 'LAVA'
1.07E-6,  0.339,    0.236,     0.01,  0.92, 'WHITE SAND'
```

The HYDRO.TBL parameter table file contains 2 parts. The first part contains the Manning's roughness coefficients for overland flow as a function of the USGS vegetation types as that data is used in the Noah land surface model. The roughness values are strictly indexed to the USGS vegetation classes so that if one wanted to use a different vegetation index dataset (e.g. the MODIS/IGBP option in the Noah land surface model) a user would need to remap these roughness values to those new vegetation indices.   Users can alter the values of overland flow roughness here for a given vegetation type. However, users may also 'scale' these initial values of roughness by changing the gridded values of the overland flow roughness scaling factor (OVROUGHRTFAC) that are contained within the high resolution routing data netcdf file. Because hydrological models are often calibrated over a particular region or watershed as opposed to a specific vegetation type it is recommended that users modify the OVROUGHRTFAC scaling factor as opposed to altering the roughness values in HYDRO.TBL.

The second part of the HYDRO.TBL parameter table contains several soil hydraulic parameters that are classified as functions of soil type. The values listed here are:

```
SATDK      - saturated soil hydraulic conductivity (m/s)
MAXSMC     - maximum volumetric soil moisture value (m^3 / m^3)
REFSMC     - reference volumetric soil moisture value (m^3 / m^3)
WLTSMC     - 'wilting point' for volumetric soil water (m^3 / m^3)
QTZ         - quartz fraction of the soil
```

These soil parameters are copied from the SOILPARM.TBL parameter table from the Noah land surface model. They are provided in HYDRO.TBL to allow the user to modify those parameters as needed during model calibration activities without modifying the SOILPARM.TBL file and thus is just done for convenience. In effect, when routing options in WRF-Hydro are activated the code will read the soil hydraulic parameters from HYDRO.TBL. If the Noah land surface model is run within WRF-Hydro without any of the routing options active, the code will simply use the parameter values specific in HYDRO.TBL.

A10. High-resolution terrain model netcdf file header

```
netcdf Fulldom_hires_hydrofile_4mile_benchmark {
dimensions:
      y = 200 ;
      x = 200 ;
variables:
      float OVROUGHRTFAC(y, x) ;
             OVROUGHRTFAC:_FillValue = -3.402823e+38f ;
             OVROUGHRTFAC:coordinates = "x y" ;
             OVROUGHRTFAC:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
             OVROUGHRTFAC:grid_mapping = "lambert_conformal_conic"
;
             OVROUGHRTFAC:long_name = "ovroughrtfac" ;
             OVROUGHRTFAC:missing_value = -3.402823e+38f ;
             OVROUGHRTFAC:units = "Meter" ;
      short RETDEPRTFAC(y, x) ;
             RETDEPRTFAC:_FillValue = 0s ;
             RETDEPRTFAC:coordinates = "x y" ;
             RETDEPRTFAC:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
             RETDEPRTFAC:grid_mapping = "lambert_conformal_conic" ;
             RETDEPRTFAC:long_name = "retdeprtfac" ;
             RETDEPRTFAC:missing_value = 0s ;
             RETDEPRTFAC:units = "Meter" ;
      int lambert_conformal_conic ;
             lambert_conformal_conic:grid_mapping_name =
"lambert_conformal_conic" ;
             lambert_conformal_conic:longitude_of_central_meridian
= -105.459999084 ;
             lambert_conformal_conic:latitude_of_projection_origin
= 40.0380058289 ;
             lambert_conformal_conic:false_easting = 0. ;
             lambert_conformal_conic:false_northing = 0. ;
```

```
                    lambert_conformal_conic:standard_parallel = 39., 41. ;
        short CHANNELGRID(y, x) ;
                CHANNELGRID:long_name = "CHANNELGRID" ;
                CHANNELGRID:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
                CHANNELGRID:coordinates = "x y" ;
                CHANNELGRID:grid_mapping = "lambert_conformal_conic" ;
                CHANNELGRID:units = "Meter" ;
                CHANNELGRID:missing_value = -32768s ;
                CHANNELGRID:_FillValue = -32768s ;
        double y(y) ;
                y:long_name = "y coordinate of projection" ;
                y:standard_name = "projection_y_coordinate" ;
                y:units = "Meter" ;
        double x(x) ;
                x:long_name = "x coordinate of projection" ;
                x:standard_name = "projection_x_coordinate" ;
                x:units = "Meter" ;
        short FLOWDIRECTION(y, x) ;
                FLOWDIRECTION:long_name = "flowdirection" ;
                FLOWDIRECTION:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
                FLOWDIRECTION:coordinates = "x y" ;
                FLOWDIRECTION:grid_mapping = "lambert_conformal_conic"
;
                FLOWDIRECTION:units = "Meter" ;
                FLOWDIRECTION:missing_value = 0s ;
                FLOWDIRECTION:_FillValue = 0s ;
        short frxst_pts(y, x) ;
                frxst_pts:long_name = "frxst_pts" ;
                frxst_pts:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
```

```
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
        frxst_pts:coordinates = "x y" ;
        frxst_pts:grid_mapping = "lambert_conformal_conic" ;
        frxst_pts:units = "Meter" ;
        frxst_pts:missing_value = -32768s ;
        frxst_pts:_FillValue = -32768s ;
    short basn_msk(y, x) ;
        basn_msk:long_name = "basn_msk" ;
        basn_msk:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
        basn_msk:coordinates = "x y" ;
        basn_msk:grid_mapping = "lambert_conformal_conic" ;
        basn_msk:units = "Meter" ;
        basn_msk:missing_value = -32768s ;
        basn_msk:_FillValue = -32768s ;
    short LAKEGRID(y, x) ;
        LAKEGRID:long_name = "LAKEGRID" ;
        LAKEGRID:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
        LAKEGRID:coordinates = "x y" ;
        LAKEGRID:grid_mapping = "lambert_conformal_conic" ;
        LAKEGRID:units = "Meter" ;
        LAKEGRID:missing_value = -32768s ;
        LAKEGRID:_FillValue = -32768s ;
    float LATITUDE(y, x) ;
        LATITUDE:long_name = "latitude" ;
        LATITUDE:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
```

```
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
        LATITUDE:coordinates = "x y" ;
        LATITUDE:grid_mapping = "lambert_conformal_conic" ;
        LATITUDE:units = "Meter" ;
        LATITUDE:missing_value = -3.402823e+38f ;
        LATITUDE:_FillValue = -3.402823e+38f ;
    float LONGITUDE(y, x) ;
        LONGITUDE:long_name = "longitude" ;
        LONGITUDE:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
        LONGITUDE:coordinates = "x y" ;
        LONGITUDE:grid_mapping = "lambert_conformal_conic" ;
        LONGITUDE:units = "Meter" ;
        LONGITUDE:missing_value = -3.402823e+38f ;
        LONGITUDE:_FillValue = -3.402823e+38f ;
    short STREAMORDER(y, x) ;
        STREAMORDER:long_name = "str_order" ;
        STREAMORDER:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
        STREAMORDER:coordinates = "x y" ;
        STREAMORDER:grid_mapping = "lambert_conformal_conic" ;
        STREAMORDER:units = "Meter" ;
        STREAMORDER:missing_value = -32768s ;
        STREAMORDER:_FillValue = -32768s ;
    short TOPOGRAPHY(y, x) ;
        TOPOGRAPHY:long_name = "topography" ;
        TOPOGRAPHY:esri_pe_string =
"PROJCS[\"North_America_Lambert_Conformal_Conic\",GEOGCS[\"GCS_No
rth_American_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GR
S_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT
[\"Degree\",0.0174532925199433]],PROJECTION[\"Lambert_Conformal_C
```

```
onic\"],PARAMETER[\"false_easting\",0.0],PARAMETER[\"false_northi
ng\",0.0],PARAMETER[\"central_meridian\",-
105.459999084],PARAMETER[\"standard_parallel_1\",39.0],PARAMETER[
\"standard_parallel_2\",41.0],PARAMETER[\"latitude_of_origin\",40
.0380058289],UNIT[\"Meter\",1.0]]" ;
        TOPOGRAPHY:coordinates = "x y" ;
        TOPOGRAPHY:grid_mapping = "lambert_conformal_conic" ;
        TOPOGRAPHY:units = "Meter" ;
        TOPOGRAPHY:missing_value = -32768s ;
        TOPOGRAPHY:_FillValue = -32768s ;

// global attributes:
        :Conventions = "CF-1.0" ;
        :Source_Software = "Esri ArcGIS" ;
        :history = "Thu Feb 21 19:55:26 2013: ncap2 -s
        :nco_openmp_thread_number = 1 ;
}
```

A11. Forcing data netcdf file header

netcdf \201111040900 {
dimensions:
        Time = UNLIMITED ; // (1 currently)
        south_north = 475 ;
        west_east = 475 ;
variables:
        float Q2D(Time, south_north, west_east) ;
                Q2D:FieldType = 104 ;
                Q2D:MemoryOrder = "XY " ;
                Q2D:description = "QV at 2 M" ;
                Q2D:units = "kg kg-1" ;
                Q2D:stagger = "" ;
                Q2D:coordinates = "XLONG XLAT" ;
        float T2D(Time, south_north, west_east) ;
                T2D:FieldType = 104 ;
                T2D:MemoryOrder = "XY " ;
                T2D:description = "TEMP at 2 M" ;
                T2D:units = "K" ;
                T2D:stagger = "" ;
                T2D:coordinates = "XLONG XLAT" ;
        float SWDOWN(Time, south_north, west_east) ;
                SWDOWN:FieldType = 104 ;
                SWDOWN:MemoryOrder = "XY " ;
                SWDOWN:description = "DOWNWARD SHORT WAVE FLUX AT
GROUND SURFACE" ;
                SWDOWN:units = "W m-2" ;
                SWDOWN:stagger = "" ;
                SWDOWN:coordinates = "XLONG XLAT" ;
        float LWDOWN(Time, south_north, west_east) ;
                LWDOWN:FieldType = 104 ;
                LWDOWN:MemoryOrder = "XY " ;
                LWDOWN:description = "DOWNWARD LONG WAVE FLUX AT
GROUND SURFACE" ;
                LWDOWN:units = "W m-2" ;
                LWDOWN:stagger = "" ;
                LWDOWN:coordinates = "XLONG XLAT" ;
        float U2D(Time, south_north, west_east) ;
                U2D:FieldType = 104 ;
                U2D:MemoryOrder = "XY " ;
                U2D:description = "U at 10 M" ;
                U2D:units = "m s-1" ;
                U2D:stagger = "" ;
                U2D:coordinates = "XLONG XLAT" ;
        float V2D(Time, south_north, west_east) ;

```
        V2D:FieldType = 104 ;
        V2D:MemoryOrder = "XY " ;
        V2D:description = "V at 10 M" ;
        V2D:units = "m s-1" ;
        V2D:stagger = "" ;
        V2D:coordinates = "XLONG XLAT" ;
float PSFC(Time, south_north, west_east) ;
        PSFC:FieldType = 104 ;
        PSFC:MemoryOrder = "XY " ;
        PSFC:description = "SFC PRESSURE" ;
        PSFC:units = "Pa" ;
        PSFC:stagger = "" ;
        PSFC:coordinates = "XLONG XLAT" ;
double RAINRATE(Time, south_north, west_east) ;
        RAINRATE:FieldType = 104 ;
        RAINRATE:MemoryOrder = "XY " ;
        RAINRATE:coordinates = "XLONG XLAT" ;
        RAINRATE:description = "ACCUMULATED TOTAL GRID SCALE
PRECIPITATION" ;
        RAINRATE:stagger = "" ;
        RAINRATE:units = "mm" ;

// global attributes:
            :TITLE = " OUTPUT FROM WRF V3.3 MODEL" ;
            :START_DATE = "2011-11-04_00:00:00" ;
            :SIMULATION_START_DATE = "2011-11-04_00:00:00" ;
            :WEST-EAST_GRID_DIMENSION = 476 ;
            :SOUTH-NORTH_GRID_DIMENSION = 476 ;
            :BOTTOM-TOP_GRID_DIMENSION = 84 ;
            :DX = 1000.f ;
            :DY = 1000.f ;
            :GRIDTYPE = "C" ;
            :DIFF_OPT = 1 ;
            :KM_OPT = 4 ;
            :DAMP_OPT = 0 ;
            :DAMPCOEF = 0.2f ;
            :KHDIF = 0.f ;
            :KVDIF = 0.f ;
            :MP_PHYSICS = 8 ;
            :RA_LW_PHYSICS = 1 ;
            :RA_SW_PHYSICS = 2 ;
            :SF_SFCLAY_PHYSICS = 1 ;
            :SF_SURFACE_PHYSICS = 1 ;
            :BL_PBL_PHYSICS = 1 ;
            :CU_PHYSICS = 0 ;
            :SURFACE_INPUT_SOURCE = 1 ;
```

```
:SST_UPDATE = 1 ;
:GRID_FDDA = 0 ;
:GFDDA_INTERVAL_M = 0 ;
:GFDDA_END_H = 0 ;
:GRID_SFDDA = 0 ;
:SGFDDA_INTERVAL_M = 0 ;
:SGFDDA_END_H = 0 ;
:SF_URBAN_PHYSICS = 0 ;
:FEEDBACK = 1 ;
:SMOOTH_OPTION = 0 ;
:SWRAD_SCAT = 1.f ;
:W_DAMPING = 0 ;
:MOIST_ADV_OPT = 1 ;
:SCALAR_ADV_OPT = 1 ;
:TKE_ADV_OPT = 1 ;
:DIFF_6TH_OPT = 0 ;
:DIFF_6TH_FACTOR = 0.12f ;
:OBS_NUDGE_OPT = 0 ;
:BUCKET_MM = -1.f ;
:BUCKET_J = -1.f ;
:PREC_ACC_DT = 0.f ;
:OMLCALL = 0 ;
:ISFTCFLX = 0 ;
:ISHALLOW = 0 ;
:DFI_OPT = 0 ;
:SHCU_PHYSICS = 0 ;
:WEST-EAST_PATCH_START_UNSTAG = 1 ;
:WEST-EAST_PATCH_END_UNSTAG = 475 ;
:WEST-EAST_PATCH_START_STAG = 1 ;
:WEST-EAST_PATCH_END_STAG = 476 ;
:SOUTH-NORTH_PATCH_START_UNSTAG = 1 ;
:SOUTH-NORTH_PATCH_END_UNSTAG = 475 ;
:SOUTH-NORTH_PATCH_START_STAG = 1 ;
:SOUTH-NORTH_PATCH_END_STAG = 476 ;
:BOTTOM-TOP_PATCH_START_UNSTAG = 1 ;
:BOTTOM-TOP_PATCH_END_UNSTAG = 83 ;
:BOTTOM-TOP_PATCH_START_STAG = 1 ;
:BOTTOM-TOP_PATCH_END_STAG = 84 ;
:GRID_ID = 2 ;
:PARENT_ID = 1 ;
:I_PARENT_START = 36 ;
:J_PARENT_START = 72 ;
:PARENT_GRID_RATIO = 5 ;
:DT = 0.2f ;
:CEN_LAT = 43.74775f ;
:CEN_LON = 8.732391f ;
```

```
            :TRUELAT1 = 42.894f ;
            :TRUELAT2 = 42.894f ;
            :MOAD_CEN_LAT = 42.894f ;
            :STAND_LON = 9.137f ;
            :POLE_LAT = 90.f ;
            :POLE_LON = 0.f ;
            :GMT = 0.f ;
            :JULYR = 2011 ;
            :JULDAY = 308 ;
            :MAP_PROJ = 1 ;
            :MMINLU = "USGS" ;
            :NUM_LAND_CAT = 24 ;
            :ISWATER = 16 ;
            :ISLAKE = -1 ;
            :ISICE = 24 ;
            :ISURBAN = 1 ;
            :ISOILWATER = 14 ;
}
```

A12. Land model output netcdf file header

```
netcdf \2011072312 {
dimensions:
        Time = UNLIMITED ; // (1 currently)
        DateStrLen = 19 ;
        west_east = 200 ;
        south_north = 200 ;
        west_east_stag = 201 ;
        south_north_stag = 201 ;
        soil_layers_stag = 4 ;
variables:
        char Times(Time, DateStrLen) ;
        int IVGTYP(Time, south_north, west_east) ;
                IVGTYP:MemoryOrder = "XY" ;
                IVGTYP:description = "Dominant vegetation category" ;
                IVGTYP:units = "category" ;
                IVGTYP:stagger = "-" ;
        int ISLTYP(Time, south_north, west_east) ;
                ISLTYP:MemoryOrder = "XY" ;
                ISLTYP:description = "Dominant soil category" ;
                ISLTYP:units = "category" ;
                ISLTYP:stagger = "-" ;
        float SKINTEMP(Time, south_north, west_east) ;
                SKINTEMP:MemoryOrder = "XY" ;
                SKINTEMP:description = "Skin temperature" ;
                SKINTEMP:units = "K" ;
                SKINTEMP:stagger = "-" ;
        float CANWAT(Time, south_north, west_east) ;
                CANWAT:MemoryOrder = "XY" ;
                CANWAT:description = "Canopy water content" ;
                CANWAT:units = "mm" ;
                CANWAT:stagger = "-" ;
        float SOIL_T(Time, soil_layers_stag, south_north, west_east) ;
                SOIL_T:MemoryOrder = "XYZ" ;
                SOIL_T:description = "soil temperature" ;
                SOIL_T:units = "K" ;
                SOIL_T:stagger = "Z" ;
        float SOIL_M(Time, soil_layers_stag, south_north, west_east) ;
                SOIL_M:MemoryOrder = "XYZ" ;
                SOIL_M:description = "volumetric soil moisture" ;
                SOIL_M:units = "m{3} m{-3}" ;
                SOIL_M:stagger = "Z" ;
        float SOIL_W(Time, soil_layers_stag, south_north, west_east) ;
                SOIL_W:MemoryOrder = "XYZ" ;
                SOIL_W:description = "liquid volumetric soil moisture" ;
                SOIL_W:units = "m{3} m{-3}" ;
                SOIL_W:stagger = "Z" ;
        float SOIL_MX(Time, south_north, west_east) ;
                SOIL_MX:MemoryOrder = "XY" ;
                SOIL_MX:description = "total column soil moisture" ;
                SOIL_MX:units = "mm" ;
                SOIL_MX:stagger = "-" ;
        float SFCRNOFF(Time, south_north, west_east) ;
                SFCRNOFF:MemoryOrder = "XY" ;
```

```
        SFCRNOFF:description = "Accumulatetd surface runoff" ;
        SFCRNOFF:units = "mm" ;
        SFCRNOFF:stagger = "-" ;
float UGDRNOFF(Time, south_north, west_east) ;
        UGDRNOFF:MemoryOrder = "XY" ;
        UGDRNOFF:description = "Accumulated underground runoff" ;
        UGDRNOFF:units = "mm" ;
        UGDRNOFF:stagger = "-" ;
float INTRFLOW(Time, south_north, west_east) ;
        INTRFLOW:MemoryOrder = "XY" ;
        INTRFLOW:description = "Accumulated interflow runoff" ;
        INTRFLOW:units = "mm" ;
        INTRFLOW:stagger = "-" ;
float SFCEVP(Time, south_north, west_east) ;
        SFCEVP:MemoryOrder = "XY" ;
        SFCEVP:description = "Accumulated evaporation from surface" ;
        SFCEVP:units = "mm" ;
        SFCEVP:stagger = "-" ;
float ETPND(Time, south_north, west_east) ;
        ETPND:MemoryOrder = "XY" ;
        ETPND:description = "Accumulated evaporation from PONDED Water" ;
        ETPND:units = "mm" ;
        ETPND:stagger = "-" ;
float ETAKIN(Time, south_north, west_east) ;
        ETAKIN:MemoryOrder = "XY" ;
        ETAKIN:description = "Evapotranspiration" ;
        ETAKIN:units = "mm" ;
        ETAKIN:stagger = "-" ;
float CANEVP(Time, south_north, west_east) ;
        CANEVP:MemoryOrder = "XY" ;
        CANEVP:description = "Accumulated canopy evaporation" ;
        CANEVP:units = "mm" ;
        CANEVP:stagger = "-" ;
float EDIRX(Time, south_north, west_east) ;
        EDIRX:MemoryOrder = "XY" ;
        EDIRX:description = "Accumulated direct soil evaporation" ;
        EDIRX:units = "mm" ;
        EDIRX:stagger = "-" ;
float ETTX(Time, south_north, west_east) ;
        ETTX:MemoryOrder = "XY" ;
        ETTX:description = "Accumulated plant transpiration" ;
        ETTX:units = "mm" ;
        ETTX:stagger = "-" ;
float ALBEDX(Time, south_north, west_east) ;
        ALBEDX:MemoryOrder = "XY" ;
        ALBEDX:description = "Albedo -- What kind? (I.e., including what effects?)" ;
        ALBEDX:units = "fraction" ;
        ALBEDX:stagger = "-" ;
float WEASD(Time, south_north, west_east) ;
        WEASD:MemoryOrder = "XY" ;
        WEASD:description = "Water equivalent snow depth" ;
        WEASD:units = "m" ;
        WEASD:stagger = "-" ;
float ACRAIN(Time, south_north, west_east) ;
        ACRAIN:MemoryOrder = "XY" ;
        ACRAIN:description = "Accumulated precipitation" ;
```

ACRAIN:units = "mm" ;
ACRAIN:stagger = "-" ;
float ACSNOM(Time, south_north, west_east) ;
ACSNOM:MemoryOrder = "XY" ;
ACSNOM:description = "Accumulated snow melt" ;
ACSNOM:units = "mm" ;
ACSNOM:stagger = "-" ;
float ESNOW(Time, south_north, west_east) ;
ESNOW:MemoryOrder = "XY" ;
ESNOW:description = "Accumulated evaporation of snow" ;
ESNOW:units = "mm" ;
ESNOW:stagger = "-" ;
float DRIP(Time, south_north, west_east) ;
DRIP:MemoryOrder = "XY" ;
DRIP:description = "Accumulated canopy drip" ;
DRIP:units = "mm" ;
DRIP:stagger = "-" ;
float DEWFALL(Time, south_north, west_east) ;
DEWFALL:MemoryOrder = "XY" ;
DEWFALL:description = "Accumulated dewfall" ;
DEWFALL:units = "mm" ;
DEWFALL:stagger = "-" ;
float SNODEP(Time, south_north, west_east) ;
SNODEP:MemoryOrder = "XY" ;
SNODEP:description = "Snow depth" ;
SNODEP:units = "m" ;
SNODEP:stagger = "-" ;
float VEGFRA(Time, south_north, west_east) ;
VEGFRA:MemoryOrder = "XY" ;
VEGFRA:description = "Green vegetation fraction" ;
VEGFRA:units = "fraction" ;
VEGFRA:stagger = "-" ;
float Z0(Time, south_north, west_east) ;
Z0:MemoryOrder = "XY" ;
Z0:description = "Roughness length" ;
Z0:units = "m" ;
Z0:stagger = "-" ;
float HFX(Time, south_north, west_east) ;
HFX:MemoryOrder = "XY" ;
HFX:description = "Upward surface sensible heat flux" ;
HFX:units = "W m{-2}" ;
HFX:stagger = "-" ;
float QFX(Time, south_north, west_east) ;
QFX:MemoryOrder = "XY" ;
QFX:description = "Upward surface latent heat flux" ;
QFX:units = "W m{-2}" ;
QFX:stagger = "-" ;
float GRDFLX(Time, south_north, west_east) ;
GRDFLX:MemoryOrder = "XY" ;
GRDFLX:description = "Ground heat flux at surface" ;
GRDFLX:units = "W m{-2}" ;
GRDFLX:stagger = "-" ;
float SW(Time, south_north, west_east) ;
SW:MemoryOrder = "XY" ;
SW:description = "Downward shortwave radiation flux" ;
SW:units = "W m{-2}" ;

```
              SW:stagger = "-" ;
      float LW(Time, south_north, west_east) ;
              LW:MemoryOrder = "XY" ;
              LW:description = "Downward longwave radiation flux" ;
              LW:units = "W m{-2}" ;
              LW:stagger = "-" ;
      float FDOWN(Time, south_north, west_east) ;
              FDOWN:MemoryOrder = "XY" ;
              FDOWN:description = "Radiation forcing at the surface" ;
              FDOWN:units = "W m{-2}" ;
              FDOWN:stagger = "-" ;
      float XLAI(Time, south_north, west_east) ;
              XLAI:MemoryOrder = "XY" ;
              XLAI:description = "Leaf area index" ;
              XLAI:units = "dimensionless" ;
              XLAI:stagger = "-" ;
      float SNOTIME(Time, south_north, west_east) ;
              SNOTIME:MemoryOrder = "XY" ;
              SNOTIME:description = "Snow age" ;
              SNOTIME:units = "s" ;
              SNOTIME:stagger = "-" ;
      float EMBRD(Time, south_north, west_east) ;
              EMBRD:MemoryOrder = "XY" ;
              EMBRD:description = "Background Emissivity" ;
              EMBRD:units = "dimensionless" ;
              EMBRD:stagger = "-" ;
      float SNOALB(Time, south_north, west_east) ;
              SNOALB:MemoryOrder = "XY" ;
              SNOALB:description = "Maximum albedo over deep snow" ;
              SNOALB:units = "fraction" ;
              SNOALB:stagger = "-" ;
      float NOAHRES(Time, south_north, west_east) ;
              NOAHRES:MemoryOrder = "XY" ;
              NOAHRES:description = "Residual of surface energy balance" ;
              NOAHRES:units = "W m{-2}" ;
              NOAHRES:stagger = "-" ;
      float CH(Time, south_north, west_east) ;
              CH:MemoryOrder = "XY" ;
              CH:description = "Heat Exchange Coefficient" ;
              CH:units = "-" ;
              CH:stagger = "-" ;

// global attributes:
              :TITLE = "OUTPUT FROM HRLDAS v20110427" ;
              :missing_value = -1.e+33f ;
              :START_DATE = "2011-07-13_12:00:00" ;
              :MAP_PROJ = 1 ;
              :LAT1 = 39.94843f ;
              :LON1 = -105.5768f ;
              :DX = 100.f ;
              :DY = 100.f ;
              :TRUELAT1 = 39.f ;
              :TRUELAT2 = 41.f ;
              :STAND_LON = -105.46f ;
              :MMINLU = "USGS" ;
              :IZ0TLND = 0 ;
```

```
                :SFCDIF_OPTION = 0 ;
                :UCMCALL = 0 ;
}
```

A13. High resolution routing grid output netcdf file header

```
netcdf \201107231200 {
dimensions:
      time = UNLIMITED ; // (1 currently)
      x = 200 ;
      y = 200 ;
      depth = 4 ;
variables:
      int time(time) ;
            time:units = "seconds since 2011-07-23 12:00 UTC" ;
      double x(x) ;
            x:long_name = "x coordinate of projection" ;
            x:standard_name = "projection_x_coordinate" ;
            x:units = "Meter" ;
      double y(y) ;
            y:long_name = "y coordinate of projection" ;
            y:standard_name = "projection_y_coordinate" ;
            y:units = "Meter" ;
      float LATITUDE(y, x) ;
            LATITUDE:long_name = "LATITUDE" ;
            LATITUDE:standard_name = "LATITUDE" ;
            LATITUDE:units = "deg N" ;
      float LONGITUDE(y, x) ;
            LONGITUDE:long_name = "LONGITUDE" ;
            LONGITUDE:standard_name = "LONGITUDE" ;
            LONGITUDE:units = "deg e" ;
      float depth(depth) ;
            depth:units = "cm" ;
            depth:long_name = "depth of soil layer" ;
      float SOIL_M(time, depth, y, x) ;
            SOIL_M:units = "m^3/m^3" ;
            SOIL_M:description = "moisture content" ;
            SOIL_M:long_name =
"èVF¨ÿ\177\000\000ã¾.ñŒ+\000\000É\000\000\000\000\000\000\000€" ;
            SOIL_M:coordinates = "x y z" ;
            SOIL_M:grid_mapping = "lambert_conformal_conic" ;
            SOIL_M:missing_value = -9.e+15f ;
      float ZWATTABLRT(time, y, x) ;
            ZWATTABLRT:units = "m" ;
            ZWATTABLRT:long_name = "water table depth" ;
            ZWATTABLRT:coordinates = "x y" ;
            ZWATTABLRT:grid_mapping = "lambert_conformal_conic" ;
            ZWATTABLRT:missing_value = -9.e+15f ;
      float QSTRMVOLRT(time, y, x) ;
            QSTRMVOLRT:units = "mm" ;
            QSTRMVOLRT:long_name = "channel inflow" ;
            QSTRMVOLRT:coordinates = "x y" ;
            QSTRMVOLRT:grid_mapping = "lambert_conformal_conic" ;
            QSTRMVOLRT:missing_value = -9.e+15f ;
      float SFCHEADSUBRT(time, y, x) ;
            SFCHEADSUBRT:units = "mm" ;
            SFCHEADSUBRT:long_name = "surface head" ;
            SFCHEADSUBRT:coordinates = "x y" ;
            SFCHEADSUBRT:grid_mapping = "lambert_conformal_conic" ;
            SFCHEADSUBRT:missing_value = -9.e+15f ;
```

```
     float QBDRYRT(time, y, x) ;
             QBDRYRT:units = "mm" ;
             QBDRYRT:long_name = "accumulated value of the boundary
flux, + into domain, - out of domain" ;
             QBDRYRT:coordinates = "x y" ;
             QBDRYRT:grid_mapping = "lambert_conformal_conic" ;
             QBDRYRT:missing_value = -9.e+15f ;
     int lambert_conformal_conic ;
             lambert_conformal_conic:grid_mapping_name =
"lambert_conformal_conic" ;
             lambert_conformal_conic:longitude_of_central_meridian = -
105.46f ;
             lambert_conformal_conic:latitude_of_projection_origin =
40.03801f ;
             lambert_conformal_conic:false_easting = 0.f ;
             lambert_conformal_conic:false_northing = 0.f ;
             lambert_conformal_conic:standard_parallel = 39.f, 41.f ;

// global attributes:
             :missing_value = -9.e+15f ;
             :Conventions = "CF-1.0" ;
             :time_coverage_start = "2011-07-13_12:00:00" ;
             :output_decimation_factor = 1 ;
             :time_coverage_end = "2011-07-23_12:00:00" ;
}
```

A14. Channel observation point netcdf file header

```
netcdf nc200606300030 {
dimensions:
      recNum = UNLIMITED ; // (288 currently)
      station = 6 ;
      id_len = 11 ;
variables:
      float latitude(station) ;
            latitude:long_name = "Observation latitude" ;
            latitude:units = "degrees_north" ;
      float longitude(station) ;
            longitude:long_name = "Observation longitude" ;
            longitude:units = "degrees_east" ;
      float altitude(station) ;
            altitude:long_name = "Observation altitude" ;
            altitude:units = "meters" ;
      int parent_index(recNum) ;
            parent_index:long_name = "index of the station for this
record" ;
      int prevChild(recNum) ;
            prevChild:long_name = "record number of the previous record
for the same station" ;
      int lastChild(station) ;
            lastChild:long_name = "latest report for this station" ;
      float streamflow(recNum) ;
            streamflow:units = "meter^3 / sec" ;
            streamflow:long_name = "River Flow" ;
      float head(recNum) ;
            head:units = "meter" ;
            head:long_name = "River Stage" ;
      char station_id(station, id_len) ;
            station_id:long_name = "Observation id" ;
      int time_observation(recNum) ;
            time_observation:units = "seconds since 2006-06-01 00:00
UTC" ;
            time_observation:long_name = "time of observation" ;

// global attributes:
            :Conventions = "Unidata Observation Dataset v1.0" ;
            :cdm_datatype = "Station" ;
            :geospatial_lat_max = "90.0" ;
            :geospatial_lat_min = "-90.0" ;
            :geospatial_lon_max = "180.0" ;
            :geospatial_lon_min = "-180.0" ;
            :time_coverage_start = "2006-06-01_00:00:00" ;
            :stationDimension = "station" ;
            :missing_value = -9.e+15f ;
            :time_coverage_end = "2006-07-01_00:00:00" ;
}
```

A15. Channel network point netcdf file header

```
netcdf nc200606300030 {
dimensions:
        recNum = UNLIMITED ; // (45504 currently)
        station = 948 ;
        id_len = 11 ;
variables:
        float latitude(station) ;
                latitude:long_name = "Station latitude" ;
                latitude:units = "degrees_north" ;
        float longitude(station) ;
                longitude:long_name = "Station longitude" ;
                longitude:units = "degrees_east" ;
        float altitude(station) ;
                altitude:long_name = "Station altitude" ;
                altitude:units = "meters" ;
        int parent_index(recNum) ;
                parent_index:long_name = "index of the station for this record" ;
        int prevChild(recNum) ;
                prevChild:long_name = "record number of the previous record for the same station" ;
        int lastChild(station) ;
                lastChild:long_name = "latest report for this station" ;
        float streamflow(recNum) ;
                streamflow:units = "meter^3 / sec" ;
                streamflow:long_name = "River Flow" ;
        float head(recNum) ;
                head:units = "meter" ;
                head:long_name = "River Stage" ;
        char station_id(station, id_len) ;
                station_id:long_name = "Station id" ;
        int time_observation(recNum) ;
                time_observation:units = "seconds since 2006-06-01 00:00 UTC" ;
                time_observation:long_name = "time of observation" ;

// global attributes:
                :Conventions = "Unidata Observation Dataset v1.0" ;
                :cdm_datatype = "Station" ;
                :geospatial_lat_max = "90.0" ;
                :geospatial_lat_min = "-90.0" ;
                :geospatial_lon_max = "180.0" ;
                :geospatial_lon_min = "-180.0" ;
                :time_coverage_start = "2006-06-01_00:00:00" ;
                :stationDimension = "station" ;
                :missing_value = -9.e+15f ;
                :time_coverage_end = "2006-07-01_00:00:00" ;
}
```

A16. Lake point netcdf file header

```
netcdf nc200606300030 {
dimensions:
      recNum = UNLIMITED ; // (96 currently)
      station = 2 ;
      id_len = 6 ;
variables:
      float latitude(station) ;
            latitude:long_name = "Lake latitude" ;
            latitude:units = "degrees_north" ;
      float longitude(station) ;
            longitude:long_name = "Lake longitude" ;
            longitude:units = "degrees_east" ;
      float altitude(station) ;
            altitude:long_name = "Lake altitude" ;
            altitude:units = "meters" ;
      int parent_index(recNum) ;
            parent_index:long_name = "index of the lake for this
record" ;
      int prevChild(recNum) ;
            prevChild:long_name = "record number of the previous record
for the same lake" ;
      int lastChild(station) ;
            lastChild:long_name = "latest report for this lake" ;
      float elevation(recNum) ;
            elevation:units = "meters" ;
            elevation:long_name = "Lake Elevation" ;
      float inflow(recNum) ;
            inflow:units = "meter^3 / sec" ;
      float outflow(recNum) ;
            outflow:units = "meter^3 / sec" ;
      char station_id(station, id_len) ;
            station_id:long_name = "Station id" ;
      int time_observation(recNum) ;
            time_observation:units = "seconds since 2006-06-01 00:00
UTC" ;
            time_observation:long_name = "time of observation" ;

// global attributes:
            :Conventions = "Unidata Observation Dataset v1.0" ;
            :cdm_datatype = "Station" ;
            :geospatial_lat_max = "90.0" ;
            :geospatial_lat_min = "-90.0" ;
            :geospatial_lon_max = "180.0" ;
            :geospatial_lon_min = "-180.0" ;
            :time_coverage_start = "2006-06-01_00:00:00" ;
            :stationDimension = "station" ;
            :missing_value = -9.e+15f ;
            :time_coverage_end = "2006-07-01_00:00:00" ;
}
```

A17. Forecast/observation point ASCII output file (frxst_pts.txt)

```
ASCII Forecast/Observation Point Output File ('frxst_pts_out.txt')


                                                  Flowrate   Flowrate  Head
Time (sec)     Date              Stn.   Lon.    Lat.    cms        cfs     m

      3600   2005-01-01_01:00:00    0  -94.37383 36.58603   16.038    566.378  1.236
      3600   2005-01-01_01:00:00    1  -94.44971 36.59901   20.239    714.749  1.281
      3600   2005-01-01_01:00:00    2  -94.18576 36.62368   11.989    423.380  1.182
      3600   2005-01-01_01:00:00    3  -94.58810 36.63018   76.155   2689.410  0.983
      7200   2005-01-01_02:00:00    0  -94.37383 36.58603   15.794    557.772  1.226
      7200   2005-01-01_02:00:00    1  -94.44971 36.59901   20.030    707.355  1.276
      7200   2005-01-01_02:00:00    2  -94.18576 36.62368   11.770    415.665  1.172
      7200   2005-01-01_02:00:00    3  -94.58810 36.63018   75.694   2673.123  0.980
     10800   2005-01-01_03:00:00    0  -94.37383 36.58603   15.530    548.436  1.214
     10800   2005-01-01_03:00:00    1  -94.44971 36.59901   19.877    701.947  1.271
     10800   2005-01-01_03:00:00    2  -94.18576 36.62368   11.566    408.458  1.161
     10800   2005-01-01_03:00:00    3  -94.58810 36.63018   75.250   2657.438  0.978
     14400   2005-01-01_04:00:00    0  -94.37383 36.58603   15.213    537.264  1.201
     14400   2005-01-01_04:00:00    1  -94.44971 36.59901   19.687    695.234  1.266
     14400   2005-01-01_04:00:00    2  -94.18576 36.62368   11.344    400.615  1.149
     14400   2005-01-01_04:00:00    3  -94.58810 36.63018   74.722   2638.802  0.974
```

A18. Channel inflow ASCII output file (chan_inflow.txt)

ASCII-formatted total accumulated stream channel inflow
   (Mm3)

   0.000000
   0.000000
   0.000000
   1.482668
   1.613662
   2.987877
   6.950461
   22.66269
   50.36945
   91.84914
   143.1280
   292.6869
   448.1576
   584.5008
   709.0095
   813.7799
   890.2136

A19. Groundwater inflow ASCII output file (GW_inflow.txt)

A20. Groundwater outflow ASCII output file (GW_outflow.txt)

A21. Groundwater level ASCII output file (GW_zlevtxt)