US Army Corps
of Engineers®
Hydrologic Engineering Center

# Hydrologic Modeling System HEC-HMS

## *Validation Guide*

Version 4.0
December 2013

## Hydrologic Modeling System HEC-HMS, Release Notes

Use of the software described by this Manual is controlled by certain terms and conditions. The user must acknowledge and agree to be bound by the terms and conditions of usage before the software can be installed or used. For reference, a copy of the terms and conditions of usage are included in Appendix E of the software User's Manual so that they may be examined before obtaining the software.

This document contains references to product names that are used as trademarks by, or are federally registered trademarks of, their respective owners. Use of specific product names does not imply official or unofficial endorsement. Product names are used solely for the purpose of identifying products available in the public market place.

*Excel* is a registered trademark of Microsoft Corp.

*Mathcad* is a registered trademark of PTC Inc.

# Introduction

The Hydrologic Modeling System (HEC-HMS) is designed to simulate the hydrologic cycle. This broad simulation capability requires many different components: shortwave radiation, precipitation, potential evapotranspiration, snowmelt, infiltration, surface runoff, baseflow, and channel routing. Water control structures such as reservoirs and diversions also contain a number of simulation components for spillways, pumps, and culverts, to name a few. These simulation components require an extensive array of time-series data, paired data, and gridded data. The interaction of these various data and simulation components leads to a software package that is very complex.

Hydrologic engineers in Corps of Engineers' offices nationwide support planning, design, operation, permitting, and regulating activities by providing information about current and potential future runoff from watersheds, with and without water control structures. HEC-HMS can support these activities by simulating the behavior of watersheds, and computing estimates of runoff volumes, of peak flow rates, and of timing of flows. It is critical that these estimates be precise and accurate if they are to be useful. By precise we mean that every time the software computes a simulation, it must obtain the exact same numerical results for the same input data. By accurate we mean that the equations that are the foundation of each of the simulation components are solved correctly.

Code validation can be described as the process of determining that a complex software program such as HEC-HMS produces precise and accurate results. Code validation should not be confused with other types of testing. For example, model validation is the process of determining if a proposed model (often a mathematical model) accurately represents a physical process and provides predictive capability. In another example, verification is the process of determining if a set of model parameters estimated during calibration performs well under different but similar conditions. Code validation instead focuses on the very specific question of whether the software accurately solves the equations used to model or represent the physical process. Software should not be used for any of the tasks undertaken by a hydrologic engineer unless it first can demonstrate code validation of all its internal components.

The models and equations incorporated in HEC-HMS have been validated by the research community. The equations in each of the simulation components included in the program have been sourced from published hydrology textbooks and from papers published in peer-reviewed journals. The code validation process described here is designed to determine if the published models and equations have been incorporated correctly in the HEC-HMS software. However, code validation does not mean that a particular model can be applied in any and every watershed. A variety of simulation components are included in HEC-HMS because not every model is applicable in every watershed. The HEC-HMS user is responsible for selecting simulation components in the software appropriate to the selected watershed, and configuring them to appropriately represent the hydrological processes in the watershed. A calibration and verification process should be developed by the hydrologic engineer to determine if appropriate choices have been made in how the software is applied.

# The Test Suite

The team responsible for the development of HEC-HMS has created an extensive set of code validation tests. The tests are designed to determine if input data including time-series, paired data, and gridded data are processed correctly. This includes tests of unit system conversion and interpolation. The code validation tests also determine if the simulation results are accurate. The accuracy of the results is determined by one of two methods. In some cases it is possible, given the input data and parameters, to calculate the correct result by hand. In these cases the equations are simple enough that hand calculations with the aid of a calculator can determine that the results computed by HEC-HMS are correct. Other cases require independent software such as Parametric Technology Corporation Mathcad or Microsoft Excel. This second approach is required when the equations used in a simulation component are differential equations, or some other type of complex equation. Mathcad and Excel are developed independently by teams of industry specialists. The results from these independent software tools can be used to confirm that HEC-HMS is calculating accurate results.

The code validation test suite created for HEC-HMS includes 35 projects. Each of these projects focuses on a separate simulation component, as shown in Table 1. Each of these projects contains from several to many simulation runs. Each simulation run tests one component of the software in isolation from all other components. These tests have been carefully developed and have a known, accurate solution. The test suite also includes 13 projects from actual watershed applications, as shown in Table 2. These projects were selected because they are complex and require the interaction of many of the different simulation components included in HEC-HMS. These complex projects do not have rigorously developed accurate solutions. They are still useful because they can be used to detect any changes in simulated values that may arise from incremental changes in the software. A change in the software that causes a computed result to change can be detected with these 14 projects. If such a change occurs, then the source of the change must be found, investigated, and resolved.

The projects used in the code validation test suite compute thousands of time-series. A total of 1,902 of these time-series are compared against known standards. Each value of the time-series is compared and a test is deemed to have failed if even a single value is outside the allowable range. As a general rule, tests use an allowable error range of 0.02% of the correct value. This means that HEC-HMS is deemed to have failed a validation test if it makes an error greater than 0.02% of the correct value. The development team reviews the results from the entire test suite before certifying a new software version for general release. The validation suite is used to find errors so they can be repaired.

A new software version is not released for general use until it can successfully pass all of the code validation tests in the suite. We make the test suite available through the HEC website (www.hec.usace.army.mil) so that users may, if desired, review the data sets included in the suite, and optionally compute them and compare the results to the benchmark values. The projects do not need to be evaluated after downloading the software to know that the software is working correctly. The software is fully tested before it is released for use and therefore will pass all of the tests successfully after it is installed. The digital certificate included in the installation package guarantees that the software has not changed since it was certified as passing all validation tests. Any change that might occur in the software between the time it leaves the HEC, and the time it is actually installed, could cause errors during validation. Because a digital certificate is included in the installation package, the user would be alerted to any such changes during the installation process. Nevertheless, users have the option of independently confirming the results already achieved by HEC staff preparing the software for release.

The following sections serve as documentation of the procedures used by HEC staff during testing. All of these steps are followed by HEC staff during the testing required before a new version of HEC-HMS is released for use. A user could follow these same steps if they wished to independently validate the software.

*Table 1. Listing of the projects for testing each of the HEC-HMS simulation components in isolation from other components.*

| Project | Simulation Component |
|---|---|
| Bench1 | Sources, junctions, and sinks |
| Bench2 | Diversions |
| Bench3 | Lag reach routing |
| Bench4 | Kinematic wave reach routing |
| Bench5 | Modified Puls reach routing |
| Bench6 | Muskingum reach routing |
| Bench7 | Muskingum-Cunge reach routing |
| Bench8 | Muskingum-Cunge reach routing |
| Bench9 | Reservoir routing and structures |
| Bench9A | Reservoir specified release routing |
| Bench9B | Reservoir culvert outlets |
| Bench10 | Subbasin loss rate |
| Bench11 | Subbasin transform |
| Bench12 | Subbasin baseflow |
| Bench13 | Straddle stagger reach routing |
| Bench14 | Specified hyetograph precipitation |
| Bench15 | Inverse distance precipitation |
| Bench16 | Gage weights precipitation |
| Bench17 | SCS storm precipitation |
| Bench18 | Standard project storm precipitation |
| Bench19 | Frequency storm |
| Bench20 | Gridded precipitation |
| Bench21 | Monthly and Priestley-Taylor evapotranspiration |
| Bench22 | Temperature index snowmelt |
| Bench23 | Reach nutrient water quality |
| Bench24 | Reservoir nutrient water quality |
| Bench25 | Subbasin canopy and surface |
| Bench26 | Source, junction, diversion, sink nutrient water quality |
| Bench27 | Source, junction, diversion, sink sediment transport |
| Bench28 | Optimization trial simulation component |
| Bench29 | Depth-area analysis simulation component |
| Bench30 | Forecasting alternative simulation component |
| Bench31 | Subbasin surface erosion |
| Bench32 | Reach sediment transport |
| Bench33 | Reservoir sediment settling |

*Table 2. Listing of the projects from watershed applications with HEC-HMS for testing complex interactions of simulation components.*

| Project | Primary Testing Focus |
|---|---|
| American | Gridded snowmelt and time interpolation |
| Bald Eagle Creek | Continuous simulation with deficit constant method |
| Bald Eagle Dam Break | Dam failure and flood wave routing |
| Culvert 111P | Pond with culvert outlet |
| H100 PE REV | Green Ampt loss rate, Modified Puls reach routing |
| hms trmm | Unit hydrographs and save states |
| Mill Creek Event | Gridded deficit constant loss rate |
| Mincio | Soil moisture accounting, save states |
| Monongahela | Forecasting and blending |
| Pistol Creek | Gridded deficit constant |
| Rogue | Gridded precipitation, monthly ET, ModClark transform |
| Taro | Canopy, surface, soil moisture accounting |
| UNBR Reservoirs | Sediment in reaches and reservoirs |
| UNBR Reservoirs 2 | Sediment in reaches and reservoirs |

## Project Data

The code validation test suite includes 35 projects for individual components plus an additional 13 projects for testing complex interactions. All of the files in each project directory are set with "read-only" file permission. The permission must be changed to "read-write" file permission before the files can be used with HEC-HMS. Additionally, the project DSS file in each project does not contain any simulation results. If the user plans to compute the simulation runs included in these projects, a backup copy of these projects should be kept. Each time the benchmark validation tests will be computed, a fresh copy of the backup projects should be used. Any changes to the data in the validation projects will cause the tests to fail.

## Established Accurate Results

Hand calculations and independent software have been used to establish known accurate results for each code validation test. These known results are stored in a DSS file. There is a separate DSS file for each of the corresponding projects used in validation testing. The simulation results in each DSS file have been checked very carefully. Some of the methods used to check the results include HEC-RAS for culvert calculations, Mathcad®, Excel®, and hand calculations by calculator. These files should never be modified.

## Automated Comparison

It would be onerous to compare all of the 1,890 time-series manually. The development team has created a small utility program to automate this task. The utility program reads the time-series of computed results from the project DSS file. It also reads the time-series from the known result. The utility program then compares value by value the two time-series. If two values do not agree within the error range, the error is recorded in a log file. A successful passage of the test is recorded in the log file if all values match within the necessary error range. The log file is the only place to determine the success or failure of each test.

The configuration information for the comparison utility is stored in text files that have "in" as a prefix. These files contain the configuration data necessary for the automated comparison of the simulated results with the established standard results. These files are designed to be used by the "Compare" utility. The utility reads the configuration file and according to the instructions, compares simulated results against the established standard results. A CMD script file is provided for each validation project. The script file can be executed by double-clicking on it in Windows Explorer. The script file will cause the Compare program to start, read a configuration file, and then perform certain comparisons. The results of the comparisons will be written to an output report file.

## Application Steps

The following steps should be used to execute the validation tests:

1) Place a fresh copy of the benchmark projects in the directories and make sure the file permission is set to "read-write."

2) Start HEC-HMS and open the first project.

3) Compute all of the simulation runs in the project.

4) Open each of the additional projects and also compute all of the simulation runs.

5) Close HEC-HMS.

6) Using Windows Explorer, double-click on each of the CMD script files except for the one called "compare.cmd".

7) Open each of the output report files and examine the results.