Regional Simulation Model (RSM)
User's Manual

Part I of III: Hydrologic Simulation Engine (HSE) Components

RSM Version 2.2.2
Document Last Updated on March 1, 2005
Document Produced on March 17, 2005

South Florida Water Management District (SFWMD)
Office of Modeling (OoM),Model Development Division (4540)
3301 Gun Club Road, West Palm Beach, FL 33406

Revision History Shown Below For The Regional Simulation Model (RSM) User's Manual

Revision History Last updated on March 17, 2005

Table 1: *Revision history for the RSM User's Manual.*

Version	Name	Date	Comments
Initial	Ken Black	9/14/04	Several manual chapters and sections still untouched.
1.0	Ken Black	10/15/04	Updated boundary condition chapter, DTD and Schema revisions.
2.0	Ken Black	1/09/05	Incorporated revised pseudocell chapter, 2d_grid,2d_grid_data sections, exploded chapter 1 to 1,2,3.
2.1	Ken Black	1/20/05	Continued revision to Chapter 1, revised alternative 2d flow equation section.
2.2	Ken Black	1/26/05	Incorporated Joe Park's MSE Overview section revisions in Chapter 12.
2.3	Ken Black	1/28/05	Incorporated Kelsen GUI info, acknowledgments, lots of cleaning of Latex files.
3.0	Ken Black	2/14/05	Added Bibliography, additional XML content, final editing by Bob Yager, significant additions.
3.1	Ken Black	2/23/05	Updated and hyperlinked the benchmarks, formatting tables.
3.11	Tim Newton	2/24/05	Grammatical and editorial updates.
4.0	Tim,Bob, Ken	2/28/05	Major updates, rearranging, equation numbering, etc.

Contents

List of Figures

List of Tables

Preface

The hydrologic simulation engine (HSE) is a fully integrated groundwater and surface water model that can simulate a variety of hydrologic components such as overland flow, canal flow, lake storage, infiltration, evaporation, etc. Depending on the type of water bodies and water movers used in a model, 2-D overland flow, 2-D or 3-D groundwater flow, canal flow, lake flow or any combination of these flows can be simulated using the model. Local hydrology is simulated through the use of pseudocells, which calculate the local water balance on a cell by cell basis.

Acknowledgements

The South Florida Water Management District gratefully acknowledges the contributions of the professionals who have made this document a reality. The Regional Simulation Model has been developed over many years, and many people have contributed to this development. All contributors are South Florida Water Management District (SFWMD) staff except where noted.

Project Manager: Jayantha Obeysekera, Director of the Office of Modeling, wrote the original statement of work for this model in 1993. He has nurtured the technical staff all these years to bring the model to fruition, and we gratefully acknowledge his technical and managerial oversight. In the last few years, Jack Maloy has infused a new level of energy into the RSM project, fast-tracking the completion of the model by providing support in obtaining both human and financial resources.

Principal Contributors: Wasantha Lal, Lead Hydrologic Modeler, is the principal developer of the hydrologic/hydraulic tenets upon which the RSM is built. Randy VanZee, Chief Hydrologic Modeler, is the principal architect of the model, and developer of the majority of the object-oriented code. Wasantha Lal and Randy VanZee served as the principal authors of this manual. Other contributors include David Welter, Lead Hydrologic Modeler, Joseph Park, Lead Hydrologic Modeler, Eric Flaig, Senior Hydrologic Modeler, Clay Brown, Senior Hydrologic Modeler, and Mark Belnap, Senior Engineer at NTI/Verio and former SFWMD Engineer.

Editorial and Production Team: The authors' products were assembled, edited, and formatted by Ken Black, Robert Yager, and Timothy Newton of Jacobs Modeling Center, Jacobs Engineering.

FOR FURTHER INFORMATION PLEASE CONTACT:

For Part I of III technical questions:
Wasantha Lal, Ph.D., P.E.
Office of Modeling
South Florida Water Management District
3301 Gun Club Road, West Palm Beach, FL 33406
561-682-6826 wlal@sfwmd.gov

For additional copies of the RSM Documentation, please contact the District's Reference Center at 561-682-2850. The complete RSM report is also available on the World Wide Web at http://www.sfwmd.gov/org/pld/hsm/models/index.html

Acronyms

Table 2: *Acronyms used in RSM*

Acronym	**Description**
CMM	Capability Maturity Model
CVIT	Calibration, Verification, Integration and Testing
ERDC	Engineer Research and Development Center (formally individual research commands of WES (Waterways Experiment Station)
ET	Evapotranspiration
FC	Flood Control
FSF	Free Software Foundation
GIS	Geographic Information Systems
GNU	Gnu's Not Unix
GUI	Graphical Users Interface
GLPK	GNU Linear Programming Kit (part of MSE)
GMS	Groundwater Modeling System
HEC	USACE Hydrologic Engineering Center
HSE	Hydrologic Simulation Engine
LEC	Lower East Coast
LPM	Local Process Module (formerly called pseudocells)
MSE	Management Simulation Engine
NSRSM	Natural System Regional Simulation Model (the pre-drainage model under implementation)
OoM	Office of Modeling
ORM	Object Oriented Routing Model
PETSC	Portable, Extensible Toolkit for Scientific Computation (the RSM 'solver')
PC	Pseudo Cells (local hydrology in a regional model)
PID	Proportional Integral Derivative (type of MSE Controller)
POR	Period of Record
RAD	Rapid Application and Development
RSM	Regional Simulation Model
SFRSM	South Florida Regional Simulation Model (under implementation)
SFWMM	South Florida Water Management Model (the 2 X 2)
SFWMD	South Florida Water Management District
USACE	United States Army Corps of Engineers
WASH123	WAterSHed Systems of 1-D stream-river networks, 2-D overland regimes and 3-D subsurface media (USACE Regional Model under development)

Acronym	**Description**
WCA	Water Conservation Area
WCU	Water Control Unit
WS	Water Supply
XML	Extensible Markup Language

Chapter 1

Introduction To RSM

Welcome to the South Florida Water Management District's (SFWMD) Regional Simulation Model (RSM). The RSM is a general hydrologic computer model developed over the past 10 years by the SFWMD in West Palm Beach, Florida. RSM is capable of simulating a wide range of hydrologic conditions, although it has been developed principally for application in South Florida. The RSM is developed on a sound conceptual and mathematical framework that allows the RSM to be applied in a wide range of hydrologic situations.

The RSM simulates the coupled movement and distribution of groundwater and surface water throughout the model domain. The RSM currently has two principal components including the Hydrologic Simulation Engine (HSE) and the Management Simulation Engine (MSE). The HSE is capable of simulating the natural hydrology, water control features, water conveyance systems and the storage systems of South Florida. The HSE solves the governing equations of water flow through both the natural hydrologic system and man-made structures. Future versions of RSM will also be able to simulate water quality.

The MSE is a platform that provides a wide range of operational capability to the HSE. The MSE is capable of simulating a wide range of management operations for the water control features of the South Florida system. There are two levels of management. At the lower level, control algorithms or "controllers" carry out local control functions to achieve specified criteria. At the higher level, supervisors using rules or linear programming and other methods, manage the controllers to achieve certain system wide objectives. Considering that there is no unique way operations can be executed, MSE is designed to be able to simulate a variety of options including ones used in the past and many planned for the future under many normal and extreme conditions. This User's Guide, Part I of III, is focused on the HSE. The MSE is described in two separate manuals including the Controller User's Guide [1] and the Supervisor User's Guide [2].

---

[1] http://gwmftp.jacobs.com/manuals/mse_controller.pdf

[2] http://gwmftp.jacobs.com/manuals/mse_supervisor.pdf

1.1 Background

The Regional Simulation Model (RSM) has been developed to provide a tool to simulate the hydrology and man-made water control features of South Florida. This model represents the next generation of integrated water management modeling and is intended to eventually replace the existing South Florida Water Management Model (SFWMM)[3]. The RSM is being designed and built to provide flexibility in simulating the physical and operational complexities of the South Florida hydrologic system. This model will be able to support hydrology-related decisions well into the future.

One of the primary goals of the SFRSM is that it must be both flexible and adaptable to changing conditions within South Florida. With the expansive planned changes to the South Florida basins under the Comprehensive Everglades Restoration Program (CERP) and new water supply strategies, it is necessary to develop a model that can be adapted to simulate whatever new conditions develop. It is imperative that this model be easier to use than the SFWMM, with shorter learning curves and improved documentation and examples. The object-oriented program design of the RSM allows the SFRSM to consist of an assembly of different water management objects that can be interchanged as the model evolves. There will be very little hard-wiring of site or operational conditions within the SFRSM to allow maximum flexibility in model application. The SFRSM will also model hydraulic management activities to allow the simulation and assessment of environmental, urban, agriculture, and flood control demands of the water distribution system. The SFRSM could be used to provide for the resolution of many issues related to water management in South Florida.

1.2 Model Purpose

The RSM is a new generation model developed using recent advances in computer technology, computational methods, and information technology (IT) related developments. It's primary purpose is to provide a robust computational platform for solving a wide range of complex hydrology problems. It was needed in South Florida to address the competing demands for water, and the growing complexity of the water management system. The complexity of the water management system exists because of the large number of man-made structures, the operational decisions imposed on these structures, and the complexities of the surface and groundwater systems.

The HSE algorithms were developed considering the need to balance the accuracy of a solution with the efficiency of computations. RSM is implemented using the object-oriented computer language C++ which can use high level abstractions to handle some of the com-

---

[3] http://www.sfwmd.gov/org/pld/hsm/models/sfwmm/

plexity. For example, abstractions such as water bodies and water movers are central to the organizational hierarchy of the HSE. The property of abstraction, along with the ability to inherit from base types allow plenty of opportunities for the code to grow. HSE uses the PETSC external sparse linear system solver[4] to obtain the solution efficiently.

HSE can simulate two-dimensional overland flow, two-dimensional or three-dimensional groundwater flow, one-dimensional canal flow, and flow in and out of lakes in an integrated system. Figure 1.1 is a definition sketch showing some of the hydrological components simulated in the model. In the model, overland flow and canal flow are simulated using the diffusion flow model. The overland and groundwater flow domains are discretized using unstructured triangular grids. The canal network system is discretized using canal segments and junctions. A semi-implicit finite volume method is used as the computational engine. The method is unconditionally stable.

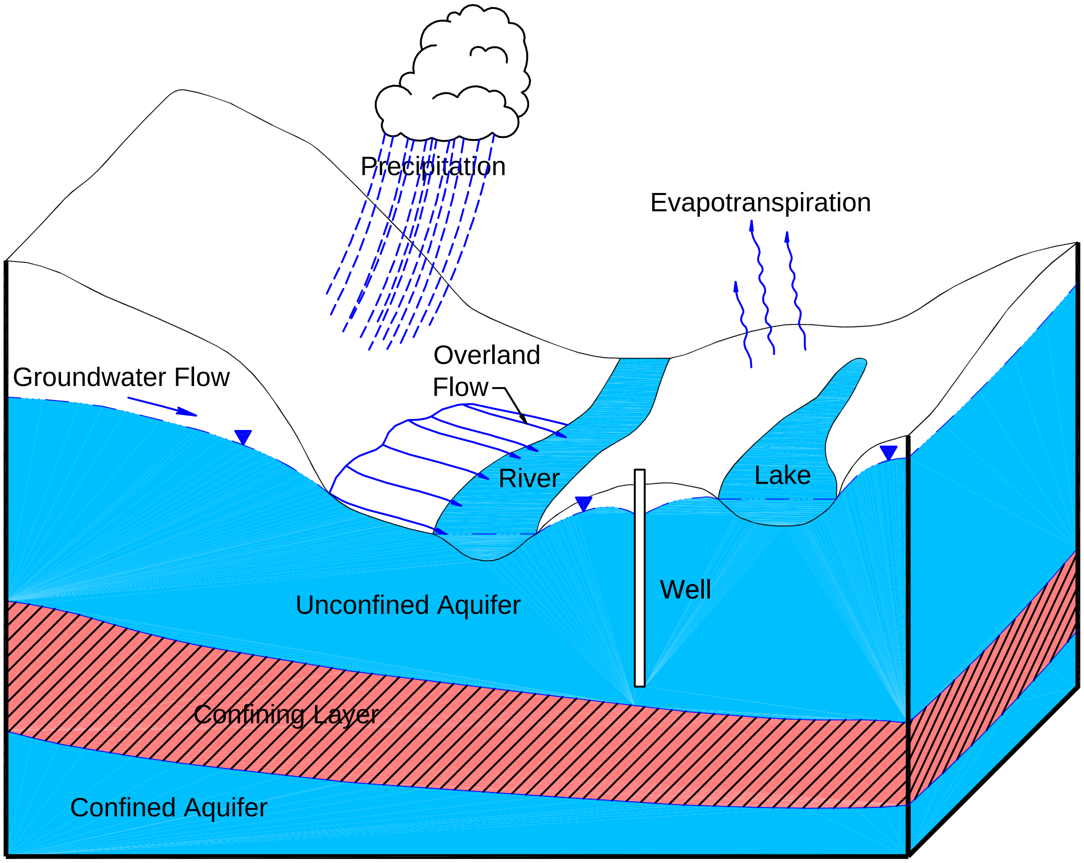


Figure 1.1: *A definition sketch depicting some components of the system.*

---

[4] http://www-unix.mcs.anl.gov/petsc/petsc-2/

In the model, an abstraction ”water body” is used to represent objects that carry conservative variables. An abstract object ”water mover” is used to represent fluxes between water bodies. Mass is conserved by using a conservative numerical method, and using data encapsulation methods in C++. Only ”water movers” can simulate fluxes in and out of water bodies and other water movements. The abstraction ”stage-volume converter” is used to distribute mass within water bodies. The abstraction ”pseudocell” is capable of supporting various ET functions, unsaturated flow distributions, and simulating local hydrologic conditions. Figure 1.2 shows an example of the conceptualization used in the model.

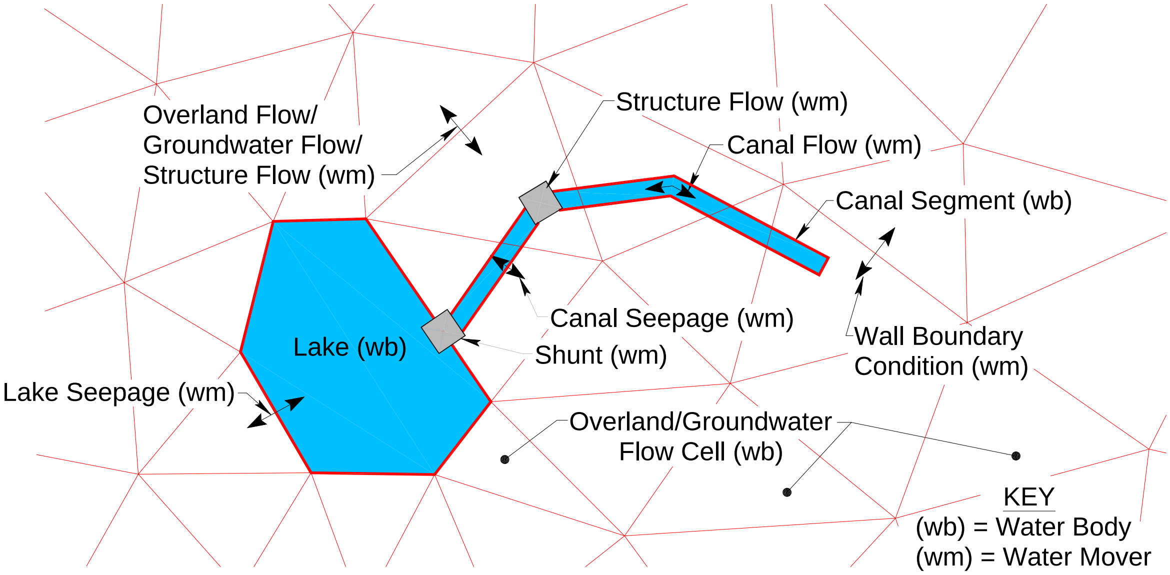


Figure 1.2: *An example of a model conceptualization.*

1.3 Model Capabilities

The Hydrologic Simulation Engine has the capability to simulate:

1. Two-dimensional overland flow over arbitrary water bodies.
2. Two-dimensional, or three-dimensional groundwater flow coupled to surface water bodies.
3. One-dimensional diffusion flow in canal networks.

4. Independent layouts of 2-D meshes and 1-D flow networks overlapping fully or partially over each other. The model can be used to simulate overland flow, canal flow, lake flow or any combination of them. The model is fully integrated, and all regional flow equations are solved simultaneously.

5. Constant or variable storage coefficients that can describe soil storage capacity that varies with depth. The variation can be described using lookup tables.

6. Various overland flow conveyance behaviors based on Manning's equation, wetland flow equations and look-up table type functions with values varying with depth.

7. Various transmissivity functions for confined and unconfined aquifers including lookup table type functions with values changing with depth.

8. Reservoirs, or large water bodies, in full interaction with aquifers.

9. Ponds or small water bodies residing within meshes but in full interaction with aquifers.

10. Many common types of structures are available and include weirs, pipes, bridges etc. with more than one flow regime. All the structure types used in National Weather Service (NWS) models, and the CASCADE model are available for use. Some of the USACE models are available as well.

11. Virtual water movers based on 1-D, 2-D, or water level difference use lookup table functions. These water movers can move water from any water body to any other water body controlled by state variables in a third water body. A lookup table is used as a mapping function. A number of pumping and flood control conditions can be simulated using these lookup tables.

12. Full three-dimensional simulation of groundwater flow, with any number of layers. Different numbers of layers can cover different parts of the horizontal domain.

13. Water budget features that can track the movement of water throughout the model.

14. A feature known as pseudo cells that can capture a wide variety of micro-hydrologic functions associated with urban and natural land use, agricultural management practices, irrigation practices, and routing.

15. Capable of simulating detention storage, and unsaturated moisture within pseudocells.

16. A management simulation engine (MSE) that is separated from the HSE. There are clear, well defined lines of communication decision variables between HSE and MSE. The MSE is capable of supporting the simulation of water management both in the past and the future under a variety of conditions. MSE has several layers of management to allow for the simulation of both local and regional level management options. The top of the management engine hierarchy is the capability to assign an overall mission

in water management. At the bottom layers are capabilities that will make it happen. MSE has a number of operational control features using open-loop control, closed-loop control, fuzzy control and number of other control algorithms in mostly local level water management. MSE can influence the HSE only through decision variables such as structure gate openings and pump discharges. Linear programming, goal programming and other options are also available to be used in the determination of appropriate water management options. The MSE is described in two separate manuals including Part II of III, RSM Controller User's Guide [5] and Part III of III, RSM Supervisor User's Guide [6].

1.4 Model Limitations

The HSE is intended to provide a comprehensive foundation for the solution of conservation laws of general hydraulics and hydrology. In its design, it uses a number of base classes such as water bodies, water movers, micro-process function objects such as pseudocells and mapping functions such as stage-volume converters to express the governing equations. The design allows for consistent tracking of water as part of the conserved property. The design allows for systematic growth of the model to cover many aspects of hydraulics and hydrology. However, the model is limited in some respects, including:

- HSE uses a number of assumptions and approximations in its formulas. One example is that HSE uses diffusion flow in the simulation of overland and canal flow. The user should be aware of the limitations and conditions of applicability of diffusion flow. MSE allows for a variety of management alternatives making the solution under each option unique. The user should also be aware of the limitations of the MSE.

- The user has to use methods of numerical error analysis to determine the cell size and the time step in the model. This is guaranteed to limit the numerical error in a model to a known quantity. The grid design must take this information into account to avoid cells that are not properly sized for the area being simulated. Guidelines and equations have been develop to help model users formulate stable grids and to select proper time stepping. References to this information are provided in Section 3.2.

- All the HSE components such as 2-D overland flow, groundwater flow and canal flow components are fully integrated through an implicit solution. Computation of local micro-hydrologic conditions within the pseudocells are explicit. The explicit approach can require small time-stepping for certain local problems. The user should be aware of the limitation of the explicit approach within pseudocells, even if this is a minor concern.

---

[5] http://gwmftp.jacobs.com/manuals/mse_controller.pdf

[6] http://gwmftp.jacobs.com/manuals/mse_supervisor.pdf

1.5 Model Documentation

RSM is documented in a number of ways. The historical development of RSM is presented in Appendix A. Selection of the most appropriate documentation for a given purpose depends on the type of need. The technical papers are the best sources for the theory and mathematical approaches used in the model. The RSM user's manuals are useful in understanding how to use the HSE and MSE. A number of other RSM documents are needed to fully understanding the model, the model input and output. Available RSM documents are listed below.

1. MSE Controllers User's Manual (South Florida Water Management District, 2005a). This document is needed when the MSE controllers are implemented in a model. Controllers can locally control structures so that water levels and other state variables can reach specified target values. Controllers provide lower level management in MSE. The Controller User's Guide [7] is available on-line.
2. MSE Supervisors User's Manual(South Florida Water Management District, 2005b). This document describes the higher level management capabilities of the MSE, and the use of LP in setting up rules and objectives or goals of a model scenario. The Supervisor User's Guide [8] is available on-line.
3. The calibration toolbox. This document provides a description of the calibration methods, and sensitivity and uncertainty analysis tools available with the model. The calibration toolbox is not yet available on-line.
4. The discretization toolbox. This document is useful in the proper selection of discretization parameters for a model run, numerical error and run time consequences of such a selection. The complete discretization toolbox is not yet available on-line.
5. The on-line HSE Data Input Guide. This resource allows user's to navigate through the XML input data structure. It has variable definitions and examples for all variables in the model. It is a companion resource to this printed document. The RSM XML Data Input Guide [9] can be viewed on-line to access the most up to date input guidelines.

The RSM model design is object oriented and understanding its structure is challenging. This differentiates RSM from other existing models that are coded serially. Reading the technical documents and the user manuals is considered mandatory prior to application of the model. Understanding the model results is dependent upon the understanding of conceptual framework of the model. The only guarantees provided in RSM is that it implements the

---

[7] http://gwmftp.jacobs.com/manuals/mse_controller.pdf

[8] http://gwmftp.jacobs.com/manuals/mse_supervisor.pdf

[9] http://gwmftp.jacobs.com/xml_schema/hse_222.html

most appropriate and efficient numerical methods available to solve the stated governing equations for regional hydrology. The selection of the appropriate input variables is up to the user. There can be consequences when a model user builds grids and selects time steps without a thorough working knowledge of the RSM stability guidelines. RSM error analysis must be considered a part of the model implementation.

1.5.1 For More Information

This section contains a list of technical papers that describe RSM development, testing and applications. Many of the papers contain information that will not likely change. Most of the projected changes will be due to future enhancements made to the model. Almost all of the papers can be found at the SFWMD website [10], but individual links are provided below for convenience.

1.5.1.1 RSM Architecture and Algorithms

These papers include information on how RSM was formulated. Computational algorithms are described as well as the conceptual platform of RSM.

1. Case Study: Model to simulate regional flow in South Florida [11](Lal et al., 2005).

   This paper describes the architecture of the RSM, and the full details of the HSE. It describes how model components are used to simulate the hydrology of South Florida.

---

2. Weighted implicit finite-volume model for overland flow [12] (Lal, 1998d).

   This manuscript describes overland and groundwater flow algorithms of the RSM model, including test problems used to verify the solution.

---

3. Performance comparison of overland flow algorithms [13] (Lal, 1998a)

   This publication describes some of the experiments carried out prior to the HSE to find out the most efficient algorithm for overland flow and groundwater flow. Some ideas of the efficiency, error and stability behavior were discovered from these experiments.

---

[10] http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/wlal.html

[11] http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/wlal.html

[12] http://gwmftp.jacobs.com/Peer_Review/poly.pdf

[13] http://gwmftp.jacobs.com/Peer_Review/alg_pap2.pdf

1.5.1.2 Error And Stability Analysis

1. Error analysis of the finite volume based regional simulation model RSM [14] (Lal and Van Zee, 2003).

   This paper describes how the accuracy of the HSE has been determined using analytical estimates of the solution and the error. This also describes how HSE errors can be predicted using analytical equations and how HSE time steps and cell sizes can be designed to achieve an error target.

---

2. Selection of time step and grid size in modeling integrated stream-aquifer interaction [15](Lal, 2001).

   When RSM or any other regional model is used to simulate canal-aquifer interaction or canal-overland flow interaction, certain rules of discretization have to be followed if the result is to be accurate. This paper described how to select the discretization in a general manner to make sure that the solution is accurate.

---

3. Numerical errors in groundwater and overland flow models [16] (Lal, 2000c)

   This paper describes numerical error in HSE, MODFLOW and other finite difference, finite-volume, and finite-element models when simulating a variety of transient and steady state-conditions. Some of the conditions include solutions close to the boundaries, pumping wells, rain driven stresses, etc.

---

4. Selection of spatial and temporal discretization in wetland modeling [17] (Lal, 1998b).

   Guidelines are developed to help model user's make appropriate selections of grid size and time stepping when performing wetland modeling using RSM.

1.5.1.3 Analytical Solutions Used In Developing and Verifying RSM

1. Modification of canal flow due to stream-aquifer interaction [18] (Lal, 2000b).

   This paper describes one of the analytical solutions used to verify the stream-aquifer interaction component of HSE. The paper describes the canal-aquifer interaction and what parameters describe the behavior. This is the only analytical solution available for dynamic solutions.

---

[14] http://gwmftp.jacobs.com/Peer_Review/hse_err.pdf

[15] http://gwmftp.jacobs.com/Peer_Review/stream-aquifer.pdf

[16] http://gwmftp.jacobs.com/Peer_Review/gw_err.pdf

[17] http://gwmftp.jacobs.com/Peer_Review/resolution.pdf

[18] http://gwmftp.jacobs.com/Peer_Review/ovlcan.pdf

2. An analytical solution for the stream-aquifer interaction problem [19] (Lal, 2000a).

   This paper describes one of the analytical solutions used to verify the stream-aquifer interaction component of HSE. This is the only analytical solution available for dynamic solutions.

1.5.1.4 Site-Specific Applications

1. Simulation of overland and groundwater flow in the Everglades National Park [20](Lal, 1998c).

   This is the first large scale application of RSM in the Everglades. The model contains both overland and groundwater flow capabilities.

---

2. Application of the South Florida Regional Simulation Model in the Southern Everglades [21] (Brion et al., 2001)

In this application, HSE is used to simulate the flow-dynamics in the Everglades National Park (ENP) area. Results are compared to the SFWMM.

1.5.1.5 Calibration Methods

1. Calibration of Riverbed Roughness [22] (Lal, 1995).

   This describes the theory behind the use of Singular Value Decomposition (SVD) that is to be used in the calibration of the RSM model. It only gives an example from the Niagara River in Buffalo, NY. An actual application to the Everglades is now being created.

---

2. ***Abstract only on-line***: Use of singular value decomposition (SVD) in calibrating regional models for South Florida [23] (Welter et al., 2005).

   Discusses the use of SVD in calibrating a new regional model of South Florida.

---

3. ***Abstract only***: Calibration of bulk aquifer parameters of regional models using hydraulic disturbances [24] (Lal and Van Zee, 2005).

---

[19] http://gwmftp.jacobs.com/Peer_Review/ovlcan_abs.pdf

[20] http://gwmftp.jacobs.com/Peer_Review/abs_tenn1.pdf

[21] http://gwmftp.jacobs.com/Peer_Review/southernglades.pdf

[22] http://gwmftp.jacobs.com/Peer_Review/calib_jo.pdf

[23] http://gwmftp.jacobs.com/Peer_Review/SVD_abstract.pdf

[24] http://gwmftp.jacobs.com/Peer_Review/Bulk_K_abstract.pdf

Describes calibration methods based on analytical solutions.

Chapter 2

RSM Governing Equations

The governing equations for the integrated overland - groundwater - canal - lake flow system consist of mass balance equations and equations of motion. In many overland flow systems, the equation of motion can be reduced to a friction flow type equation by neglecting inertia terms. Governing equations in conservative form are used in the current implicit implementation of the finite volume method. The steps involved with this process, the assumptions made, and the limitations of the resulting product are described in the references. New algorithms will have to be developed if the inertia effects of the Saint Venant equations are to be simulated.

2.1 Overland and Groundwater Flow

A complete description of the reduction of the complete Saint Venant equation to diffusion flow equations can be found in the references (Lal, 1998a), (Lal, 1998d) and (Lal et al., 2005). A brief summary is presented here for completeness. After neglecting the inertia terms, the 2-D Saint Venant equations, which consist of a continuity equation and an equation of motion, reduce to a simpler form. The continuity equation for both overland flow and saturated groundwater flow in the regional system becomes

$$S_c \frac{\partial H}{\partial t} + \frac{\partial (Hu)}{\partial x} + \frac{\partial (Hv)}{\partial y} - R_{rchg} + W = 0 \tag{2.1}$$

in which, u and v are the velocities in the x and y directions; H = Water head; R_{rchg} = net contribution of recharge from local hydrology into the regional system; W = source or sink terms due to pumping wells, etc.; and S_c = storage coefficient. $S_c = 1$ for overland

flow. The term R_{rchg} is computed for each of the discretized cells of the distributed system, assuming that a pseudo cell exists for each cell. Each pseudo cell is designed to capture all of the complexities related to local hydrology. A number of pseudo cell models are described later.
For overland flow under diffusion flow conditions (Akan and Yen, 1981) and for groundwater flow, average flow velocities in x and y directions are defined as

$$u = -\frac{T}{d}\frac{\partial H}{\partial x}, v = -\frac{T}{d}\frac{\partial H}{\partial y} \tag{2.2}$$

For groundwater flow, T = T(H)= transmissivity of the aquifer as a function of the water level;d = water depth. For overland flow,

$$T = C\left(H\right)S^{\lambda-1} \tag{2.3}$$

in which C(H)is defined as the conveyance; S = water surface slope and λ an empirical constant that is described later. Both T(H)and C(H) are useful in describing a whole range of overland and groundwater flow behaviors. In the model, object oriented design methods allow for the implementation of a variety of options for the functions T(H) and C(H)by making them as base class objects with metamorphic behavior. These objects can express the behavior of constants, analytic functions or lookup tables based on field experiments. This type of abstract representation of T(H) and C(H)is useful in the object-oriented design in describing specialized flows as in the case of ridge and slough flow of wetlands.

Mannings equation is commonly used to describe overland and canal flow. When using a general form of the Mannings equation described as

$$V = (1/n_b)d^{\gamma}S^{\lambda} \tag{2.4}$$

the corresponding expression for T can be expressed as

$$T\left(H\right) = \frac{d^{\gamma+1}S^{\lambda-1}}{n_b} \tag{2.5}$$

For the commonly used form of Mannings equation,
$\gamma = 2/3$;
$\lambda = 1/2$;
V = flow velocity;
S = water surface slope;
n_b = Manning's coefficient.

The Mannings equation can be used to describe the discharge per unit width as q(H) = T(H)S = C(H)S$^{\lambda}$. Functions T(H) and C(H) can be described using a variety of methods such as functions and lookup tables.

Because of the nonlinear nature of the Manning's equation and its singularity at S = 0, linearization can be a problem at small slopes. To avoid the resulting floating point overflow, S = Max(S, δ_n) is used when $\gamma < 1$. The variable δ_n was originally used as a numerical adjustment parameter, but it can also be used to separate turbulent flow from laminar flow. A large value of δ_n would result in a small value of T and make the matrix computations more accurate, but it would also force larger areas of relatively flat landscapes to have laminar-like flow. A value of $\delta_n = 10^{-13} - 10^{-7}$ is used in the flat terrain of South Florida. Equation 2.3 can also be used in wetlands by selecting the parameters suggested by (Kadlec and Knight, 1996).

2.2 Canal Flow

Canal flow
The 1-D St Venant equations are used to describe canal flow. A complete description of the derivation and the assumptions involved are shown in Lal(2005) or Lal(1998) The continuity equation in conservative form is

$$\frac{\partial \mathrm{A_c}}{\partial \mathrm{t}} + \frac{\partial \mathrm{Q}}{\partial \mathrm{n}} - R_{canal} + W = 0 \tag{2.6}$$

in which, A$_c$ = cross sectional area of the canal; Q = discharge rate; n = distance along the canal; R$_{canal}$ = rate (m^2/sec) at which water is entering the canal due to seepage and other sources per unit length;W = source and sink terms due to pumps. For canals under diffusion flow conditions,

$$Q = C(R)\sqrt{S} = \frac{AR^{2/3}}{n}\sqrt{S} \tag{2.7}$$

This can be linearized as

$$Q = T_c \frac{\partial h}{\partial n} \tag{2.8}$$

in which,
h = canal water level;

$$T_c = \frac{AR^{2/3}}{n\sqrt{S}} \tag{2.9}$$

using the Manning's equation, where
C(R) = canal conveyance;
R = hydraulic radius.
S = Energy slope and is defined as:
Max(S, δ_n) with
$\delta_n = 10^{-13} - 10^{-7}$ to avoid the singularity at S_0.

2.3 Lake Flow

The equation governing mass balance in a lake is

$$A_l \frac{\partial H_l}{\partial t} - R_{lake} + W = 0 \tag{2.10}$$

in which, A_l = lake area; H_l = water level; R_{lake} = net volume rate at which water is entering the lake water body due to leakage. The governing equations written in conservative form are used in the implicit implementation of the finite volume method.

2.4 Recharge From The Local Hydrologic System

The local hydrology in a regional system can depend on the local land use type. Different land use types generate different recharges and therefore different hydrologic responses. The recharge R_{rchg} described in Equation 2.1 therefore has to be computed separately for each cell covering the model. The computations take into account ET, rainfall, soil moisture effects, urban detention, local drainage effects and other factors depending on the land use type. The equation of mass balance for the local hydrology in a cell is

$$R_{rchg} = P - E + I - \frac{dU_s}{dt} - \frac{dD}{dt} \tag{2.11}$$

in which, R_{rchg} = recharge rate (m/s) computed as a volume rate per unit cell area entering into the cell; P = precipitation rate; E = evapotranspiration rate; I = water entering the cell for irrigation and other similar functions; U_s = unsaturated moisture depth; D = detention volume converted to depth. The rates $\frac{dU_s}{dt}$ and $\frac{dD}{dt}$ depend on infiltration and percolation rates of the local cell. In the model, these complex computations are performed within the pseudo cell of each respective cell. Pseudeo cell types are developed for various land use types and permitting conditions. More information about pseudo cells is provided under the object design.

Chapter 3

RSM Numerical Methods and Stability Guidelines

3.1 The Implicit Finite Volume Method

The finite-volume numerical approach has been selected to solve the governing equations for overland flow, groundwater flow, canal flow, lake flow and other types of flow included in RSM. These equations are based on conservation laws and are well suited for solution by the finite volume method. A complete description of the finite-volume method with the assumptions and the limitations can be found in (Hirsch, 1989), (Lal, 1998d) and (Lal et al., 2005). The equations can be written in conservative form as:

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+S=0 \tag{3.1}$$

in which, U is a conservative variable representing H, h, or H_l; variables F and G are the x and y components of flux in 2-D flow and S = -R_{rchg} + W = summation of source/sink terms. The finite volume formulation is applied to all 2-D, 1-D and lake related regional flows. The numerical model is developed for generic control volumes and is applied to all water bodies on an equal basis. The finite volume formulation governing equation is derived by integrating it over an arbitrary control volume or water body Ω

$$\frac{\partial}{\partial t}\int\limits_{\Omega} U d\Omega + \int\limits_{S} (\mathbf{E.n})\, dA + \int\limits_{\Omega} S d\Omega = 0 \tag{3.2}$$

in which, $\mathbf{E}$ = $[F,G]^T$ = flux rate across the wall and $\mathbf{n}$ = unit normal to the wall. The first term represents the rate of change of water mass in the water body. The second term is obtained using the Gauss theorem and contains the sum of fluxes crossing the control surfaces of the water bodies. This term contains all flow exchanges among the water bodies. Any mechanism that is capable of moving water between any two water bodies is defined as a water mover. Water bodies and water movers are two of the basic building blocks of the model. They eventually become abstractions in the object-oriented design. These abstractions are capable of growing and evolving into model objects as the model evolves. In the model, the equation is solved for average water heads H, h or H_l of all the water bodies simultaneously. The finite volume formulations represent a system of differential equations.

$$\frac{d\mathbf{V}}{dt} = \Delta\mathbf{A}\left(\mathbf{H}\right)\frac{d\mathbf{H}}{dt} = \mathbf{Q}\left(\mathbf{H}\right) + \mathbf{S} \tag{3.3}$$

in which, $\mathbf{H}$ = a vector containing the water heads of all 2-D cells, canal segments, and lakes; $\mathbf{V}$ = volumes of water contained in the water bodies;
Δ $\mathbf{A}$ is a diagonal matrix whose elements (i,i) are defined as the slopes of the stage-volume (SV) relationships of water bodies i = 1,2... The stage-volume relationship is defined as

$$V_i = f_{sv}(H_i) \tag{3.4}$$

which in each function $f_{sv}(H_i)$ is a single valued, stage-volume relationship that is defined for the water body i. The inverse relationship is defined as $H_i = f_{vs}(V_i)$. Element i of Δ $\mathbf{A}(\mathbf{H})$ is defined as

$$\Delta A_i\left(H_i\right) = \frac{\partial f_{SV}\left(H_i\right)}{\partial H_i} \tag{3.5}$$

in which the function $\Delta A_i(H_i)$ is the effective area of water body i. For 2-D open water flow, this is the cell area. For groundwater, this is s_c times the cell area. The third term $\mathbf{Q}(\mathbf{H})$gives the summation of all the inflows into each water body due to the action of all the water movers. In order to solve the coupled system, the water mover equations are first linearized to obtain

$$Q_r\left(\mathbf{H}\right) = k_0 + k_i H_i + k_j H_j \tag{3.6}$$

in which $Q_r(\mathbf{H})$ = discharge rate in the water mover r. The water mover r moves water from water body i to water body j. Linearization is performed using partial differentiation or approximate methods. Values k_0, k_i and k_j are applied to build the resistance matrix that is used to linearize $\mathbf{Q}(\mathbf{H})$ as $\mathbf{Q}(\mathbf{H}) = \mathbf{M}.\mathbf{H}$. The ordinary differential equations with linearized $\mathbf{Q}(\mathbf{H})$are solved using a weighted implicit method. (Lal, 1998d) used the following system of equations to solve Equation 3.3.

$$\left[\Delta\mathbf{A} - \alpha\Delta t\mathbf{M}^{n+1}\right].\Delta\mathbf{H} = \Delta t\left[\mathbf{M}^{n}\right].\mathbf{H}^{n} + \Delta t\left[\alpha\mathbf{S}^{n+1} + (1-\alpha)\,\mathbf{S}''\right] \tag{3.7}$$

in which, α = time weighting factor, assumed to be in the range 0.6-0.8 for small models and close to 1.0 for large models that may otherwise show signs of instability. This equation takes into account water balance of all the water bodies between times t^n and t^{n+1}. Knowing the volumes of water $\mathbf{V}(\mathbf{H}^n) = \mathbf{f}_{sv}\ (\mathbf{H}^n)$ at time step t^n and Δ H, it is possible to compute $\mathbf{V}^{n+1}$ using

$$\mathbf{V}^{n+1} = \mathbf{V}^{n} + \Delta\mathbf{A}.\Delta\mathbf{H} \tag{3.8}$$

The new heads $\mathbf{H}^{n+1}$ at time step (n+1) can now be computed using the storage-volume relationship $\mathbf{H}^{n+1} = \mathbf{f}_{VS}(\mathbf{V}^{n+1})$. The volume to stage conversion is necessary only to compute the heads used to compute the hydraulic driving forces in the water movers. The water balance computation itself does not involve any water heads. Equations 3.3 through 3.8 apply to all the water bodies equally, and the set of equations 3.7 solves the entire system simultaneously. Water heads $\overline{\mathbf{H}}$ at time $t^n + \alpha\Delta t$ act as the forcing functions for flow during the time step. Values of $\overline{\mathbf{H}}$ are computed as $\overline{\mathbf{H}} = \mathbf{H}^n + \alpha\Delta\mathbf{H}$ and are used to compute the volumes of water that pass between the water bodies in a model. These values add up to $\mathbf{Q}(\overline{\mathbf{H}})\Delta t$ for the water bodies. The water mover discharges $Q_r(\overline{\mathbf{H}})$ that contribute to equation 3.7 are computed using equation 3.6. The water balance in any water body i can be verified by comparing the change in water volume $\Delta A_i(H_i^{n+1} - H_i^n)$ with the summation of water mover discharges $Q_r(\overline{H})$.

3.2 Model Stability Guidelines

Even if the model is unconditionally stable under linear analysis, such stability does not imply a complete lack of oscillations in the solution. When conditions of linearity are violated due to the extreme nonlinearity of certain functions or stiffness, the capacity of the diffusive system to absorb spurious oscillations comes under stress, and the model may fail or show residual oscillations. Under these conditions, unscientific damping can be short sighted, and effect parts of the solution that are otherwise accurate. The solution to this problem is to understand the physical and numerical characteristics of the governing equations and the numerical methods. Methods exist to handle some of the inherent difficulties in simulating the non-linear behavior of hydrologic systems.

RSM is like many other computer models in some ways because it is built around numerical methods that solve ordinary and partial differential equations. It is also similar to other models because of the way in which empirical and semi-empirical equations are incorporated among regional equations to represent a wide variety of local and regional conditions. RSM simulates both natural and anthropogenic conditions like many of the recent models. However, it is different from many other models because of the use of object oriented methods in the design, and error analysis methods in the implementation. Model components of RSM have to be thought of as objects. A certain level of understanding of object oriented methods, and basic object types such as water bodies and water movers is required before making an application of the model. RSM is different from many other models because of the use of numerous MSE components such as optimal controllers and LP algorithms. Considering many of these factors, an RSM application can be much more complex than the application of other models.

Being able to run the model with a data set does not say anything about the validity or the accuracy of the data set or the output. The following checks can be helpful in assuring that the model results are relevant.

1. The input data collected has to be for a physically meaningful problem.
2. The problem should be one that can be solved with the governing equations used in the model.
3. The hydraulic problem has to be mathematically well posed. Only well posed problems can be solved using computational procedures for partial differential equations. Proper use of initial and boundary conditions are extremely important in setting up a well posed problem (Abbott, 1998), (Abbott and Cunge, 1982) or any other similar text for well-posedness.
4. Consider the fact that the model is built upon governing equations that are initial-boundary value problems. This means that the solution at any time and space depends

on both initial and boundary conditions. On one hand if the dependency does not exist, the problem is not a well posed problem, and the model results are meaningless. Similarly, if the model is based on faulty initial and boundary conditions, the results have to be considered meaningless depending on the severity of the dependence.

5. Select the time and cell size for the discretization based on (Lal, 2000b), (Lal, 2001) or any other text. The accuracy of the solution is limited by the conditions stipulated in these documents.

6. Follow an acceptable calibration procedure, and check if the correlation coefficient between the observed and simulated time series data is at least greater than 0.6. Tools based on generalized inverse solution using Singular Value Decomposition, and conjugate gradient methods (Lal, 1995) will be made available to calculate uncertainty and covariance matrices for models. These estimates can be used to evaluate model parameter uncertainty in a limited way.

7. If the model discharges are to be accurate, ensure that boundary condition discharges used as input to the model are accurate as well.

8. Check if the overall mass balance conditions in the model are within reasonable (<10%) limits. This still does not guarantee that there are no local numerical errors.

If *at least one* of the above conditions is violated, the model is not usable under general conditions. The model may ,however, be usable for certain regions on a limited basis if it can be scientifically justified.

During the interpretation of model results, one should consider that the regional results from the PDE's are valid only on areas larger than the size of the cells or the segments. If an imaginary Fourier sine component is used to describe this spatial extent, results are accurate only over areas covering multiple cells (2 to 5) as described by (Lal, 2000c).

Chapter 4

Model Structure

4.1 Programming Information

The RSM model is fully object-oriented and is coded in C++. There are 216 source files (.cc and .h files) for RSM version 2.2.2 with nearly 46,000 lines of computer code (comments, blank lines, and active code) with 29,000 lines of active code. There are 263 classes, 3,085 functions, 10,976 declarative statements, and 14,764 executable statements. The GNU C++ compiler is used on Red Hat Linux 9.0 to create the RSM executable used in simulations. The code currently operates only on Red Hat Linux 9.0 and uses at least seven external libraries for such items as XML technologies, netcdf files, solver technologies, etc.

4.1.1 Programming Details

A variety of code analysis methods have been used to gain an understanding of the RSM program structure. The nature of large object-oriented codes can make them difficult to comprehend and to visualize. A comprehensive code analysis of RSM 2.2.2 has been completed and includes a variety of ways of examining the code. The report can be viewed here.[1]. The source code has also been formatted and placed into a pdf file which can be retrieved by selecting the code review item in the source publisher report.[2].

---

[1] http://gwmftp.jacobs.com/sp_hse_print/undreports/index.html

[2] http://gwmftp.jacobs.com/sp_hse_print/index.html

4.2 How Is A Model Solution Achieved?

A flowchart has been created for RSM to show the sequence of events that occurs during the execution of RSM. The flowchart does not attempt to go into full detail of what the program accomplishes, but rather it traces the primary steps taken by RSM during the course of running a simulation. Much of the information has been extracted from the finite-volume class. The intent of the flow chart is to show a serial pathway taken by the code during execution. Of course, not all items in the flowchart are used for every simulation and the model input can vary from one application to another. The flowchart is displayed as a series of six figures because of its length. Part 1 is shown in Figure 4.1, Part 2 in Figure 4.2, Part 3 in Figure 4.3, Part 4 in Figure 4.4, Part 5 in Figure 4.5 and Part 6 in Figure 4.6.

Figure 4.1: *Part 1 of 6 of the RSM model execution flowchart.*

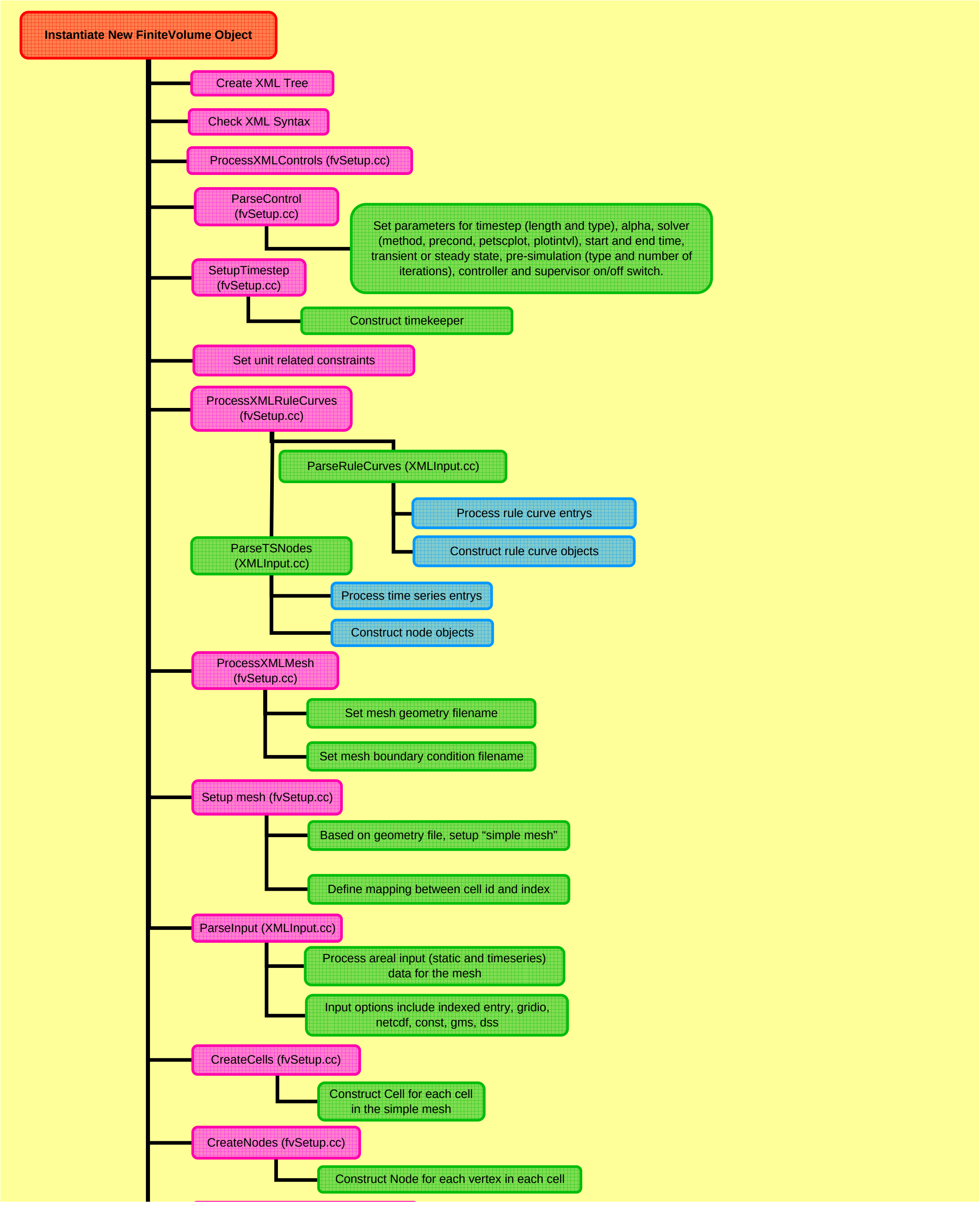


Figure 4.2: *Part 2 of 6 of the RSM model execution flowchart.*

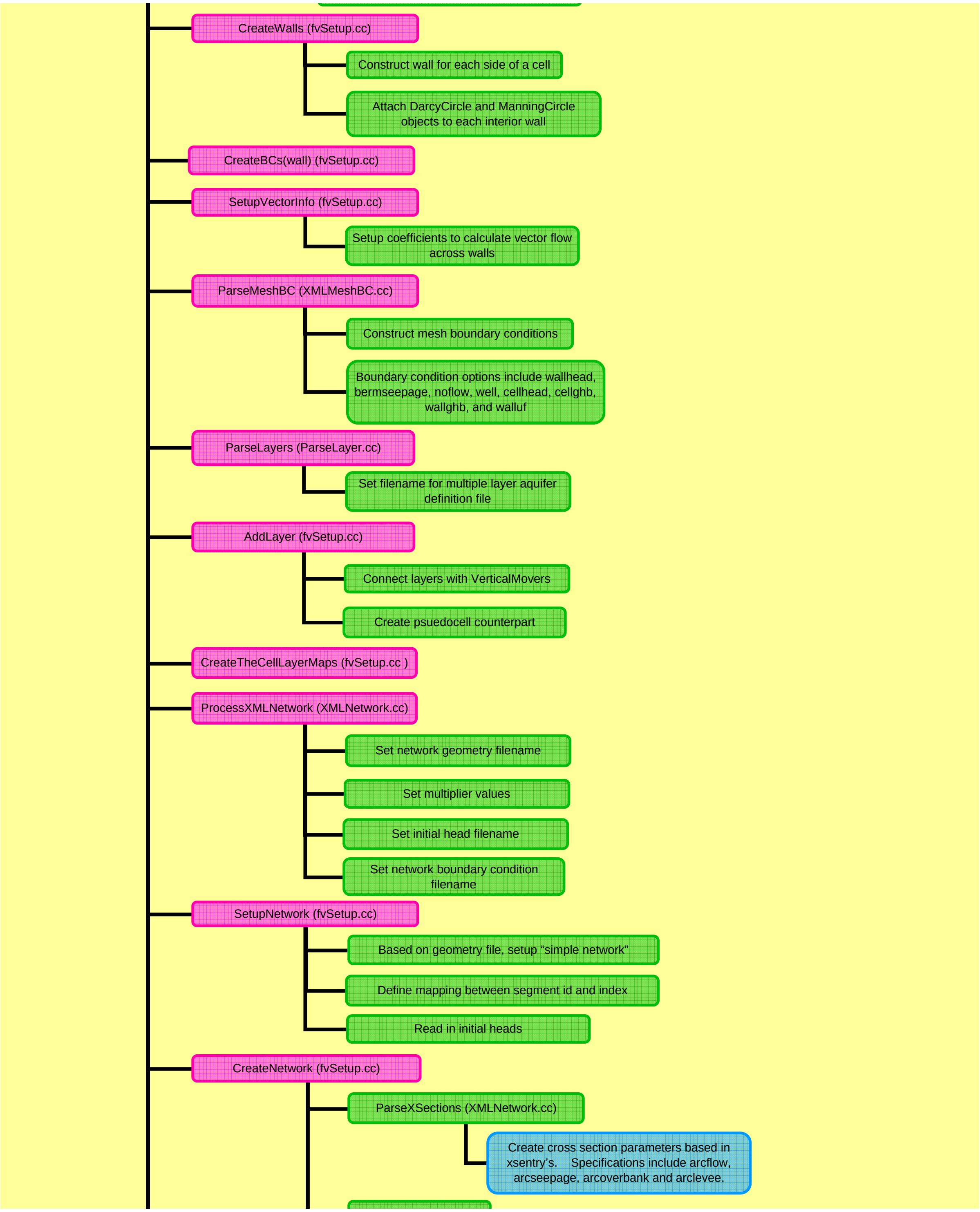


Figure 4.3: *Part 3 of 6 of the RSM model execution flowchart.*

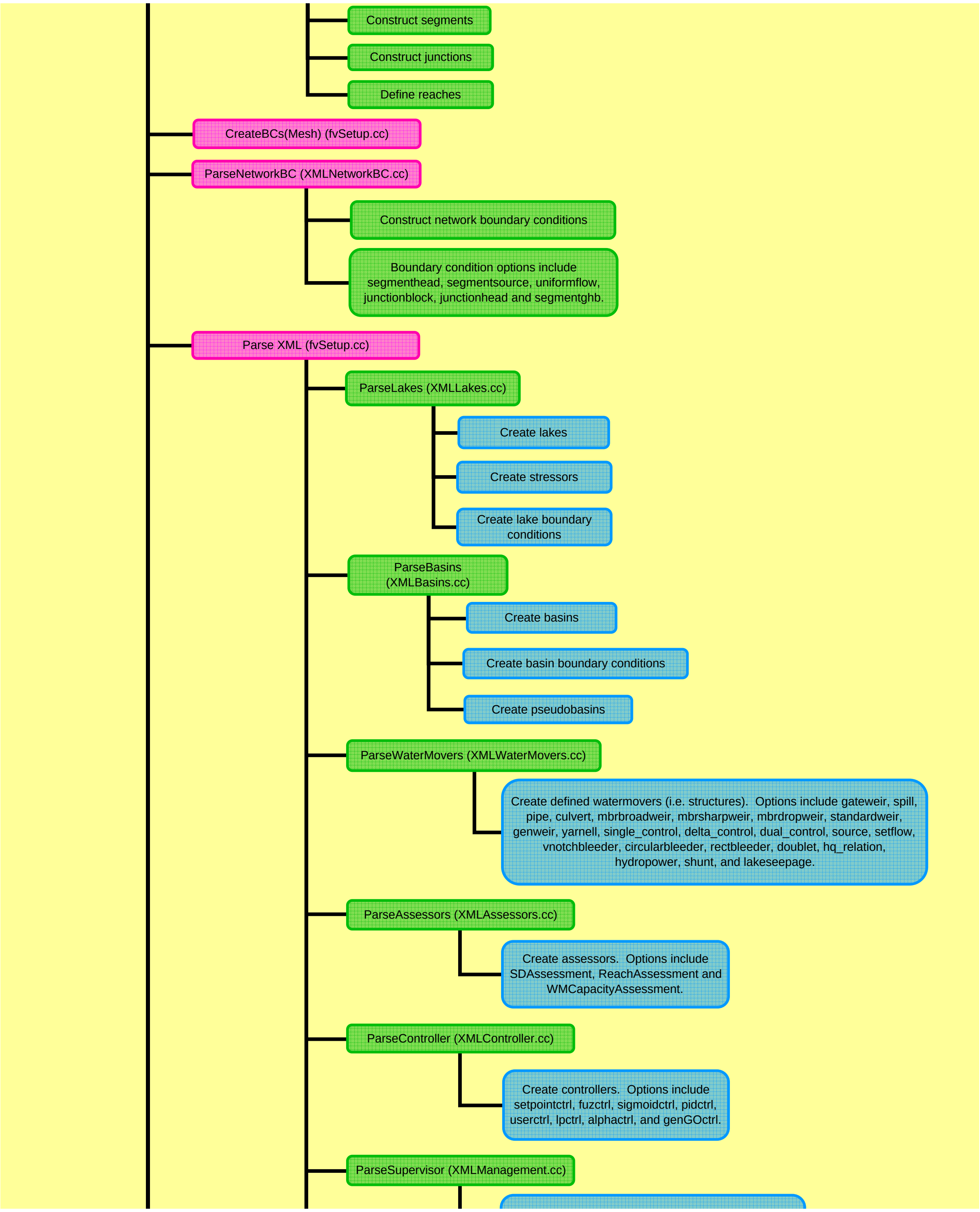


Figure 4.4: *Part 4 of 6 of the RSM model execution flowchart.*

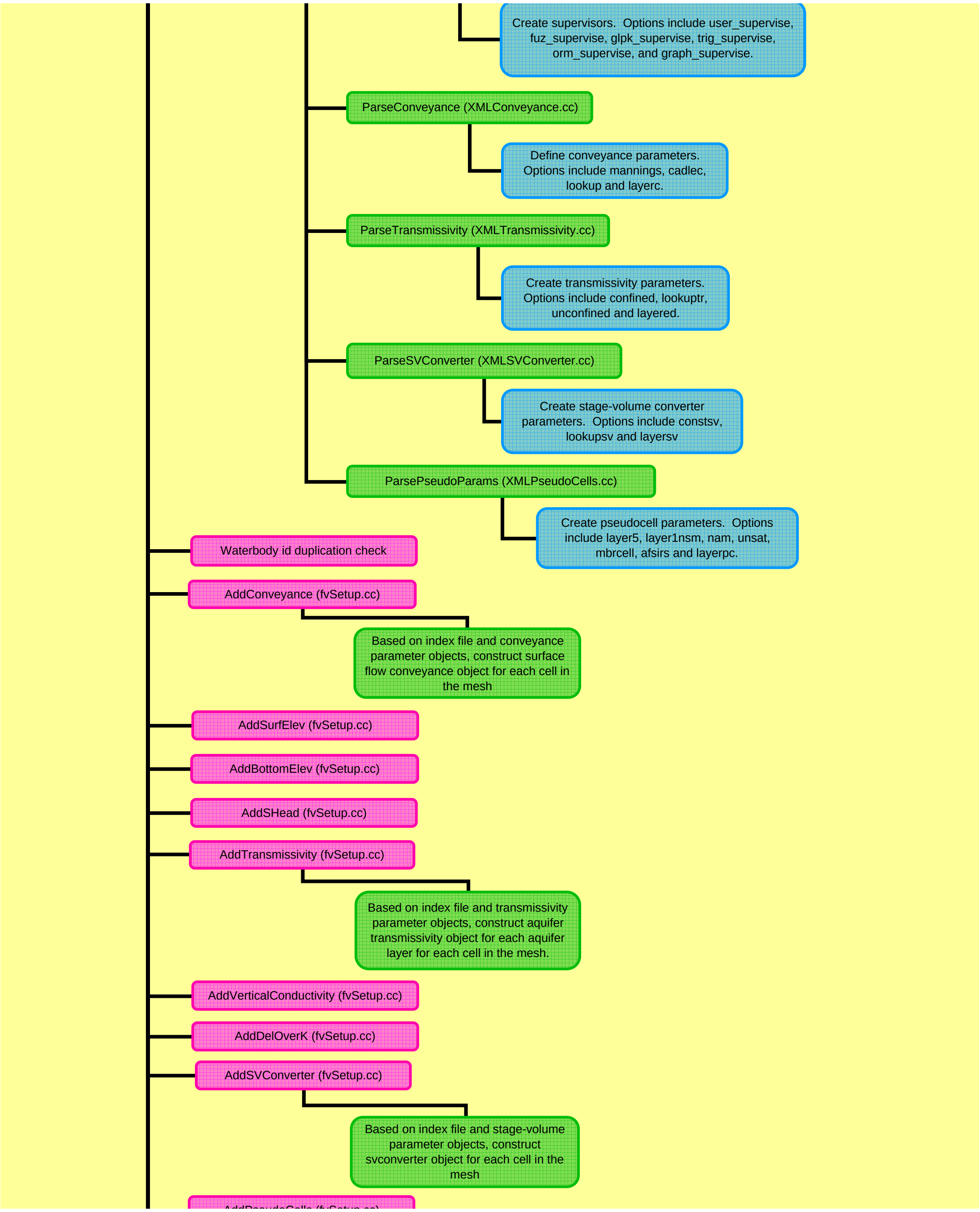


Figure 4.5: *Part 5 of 6 of the RSM model execution flowchart.*

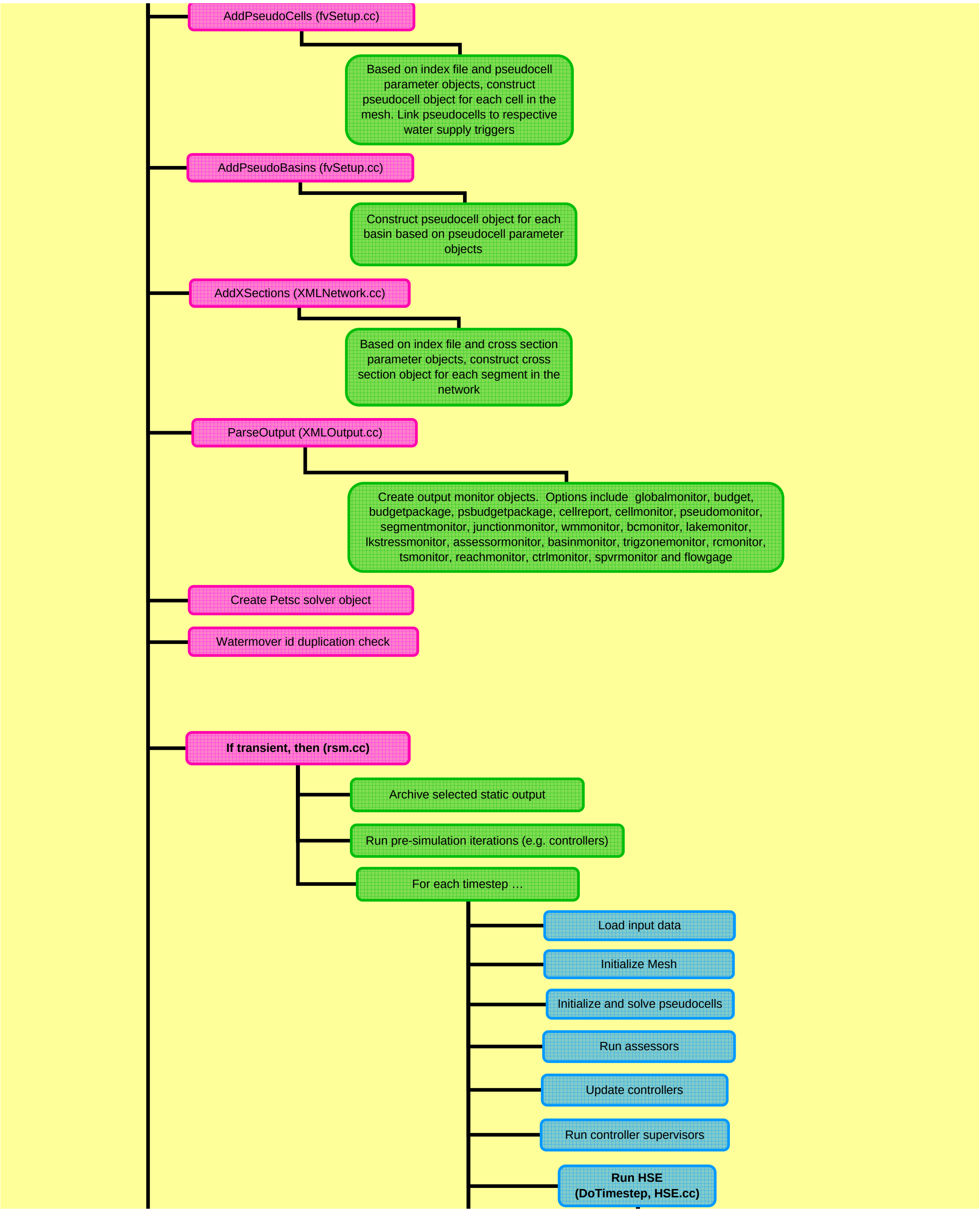
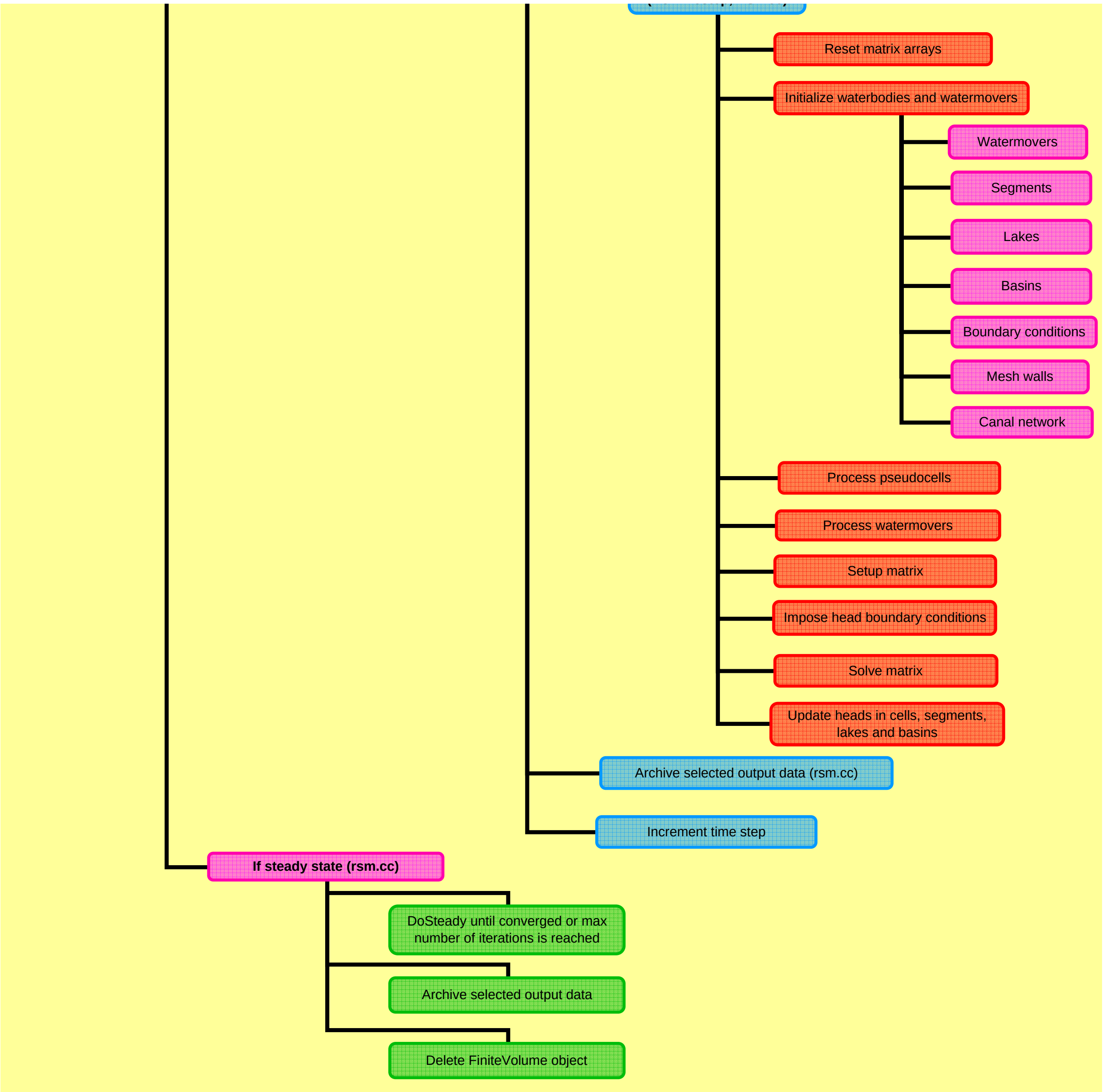


Figure 4.6: *Part 6 of 6 of the RSM model execution flowchart.*



Chapter 5

RSM Input Using XML

RSM data is generated in the self-descriptive file format of Extensible Markup Language (XML). This chapter provides an overview of RSM XML input, and is intended for users familiar with RSM input files. If you are new to XML input data files, it would be beneficial for you to study the XML overview information presented in Appendix B.

5.1 Introduction

The hydrologic simulation engine (HSE) is a fully integrated model that can simulate hydrologic components such as overland flow, canal flow, lake storage, seepage, etc. A large amount of carefully organized data is needed for the model to simulate a system. Depending on the type of water bodies and water movers available, 2-D overland flow, 2-D groundwater flow, canal flow, lake flow or any combination of these flow types can be simulated using the model. The types of information needed for a model run can be classified into the basic categories as described in Table 5.1.

The model is completely input data driven. Model objects are created as data specified in an XML input file are parsed. These objects accumulate and reside until the model run is complete. Once the model is finished, the objects are released from computer memory. The input data controls the creation of these objects and every aspect of the model run. Once the XML input files are built and a simulation is ready to executed, the following command is issued on the Linux command line.

```
hse filename [-v -l logfile]
```

in which, `filename` is the name of the XML input data file. The optional parameter `-v` may be specified to perform XML file syntax validation check without execution of HSE

Table 5.1: *Basic data types used in HSE.*

Data	**Description**
Setup	Global parameters, such as start and ending time, time step, and solver parameters.
Main objects	The basic building blocks for the model, which include: 2-D water bodies (e.g., cells, canal segments and lakes); water movers (e.g., pumps, weirs, etc.); pseudocells for land use types (e.g., agricultural and urban types).
Boundary conditions	Boundary conditions of the model domain.
Time series	Time series data needed.
Object characterizations	Local micro hydrological model characterizations using pseudocells, conveyances, transmissivities, etc.
Controllers	Low-level operational features using controllers are described here.
Management	High-level control of overall management direction of the model using LP or other coordination is defined here.
Output	Specifies what is to be output and in what form. In general, any variable in the model can be output for further processing.

hydrologic simulation time steps. This check currently is set to compare the input data against the Data Type Definition (DTD), not the schema. If user's prefer a higher degree of data validation, they should execute the schema validation steps outlined in Section B.3.2. If the option `-l` is specified with a log file argument, then all console information and error messages will be written to the log file.

HSE can be used as an overland flow model, ground water model, canal network model, or a lake model with any number of connecting water movers. Any combination of the water bodies is possible, regardless of whether they are connected or not. By default, water in the water bodies doesn't move if there are no water movers. There are five types of default horizontal water movers. They are generated automatically once proper geometry files are provided. This process takes into account geometrical overlaps and other conditions. Any additional water mover has to be defined using input data. The five default water movers are:

- Overland flow water mover moving water between all ponded cells
- Groundwater flow water mover moving water between ground water cells
- Canal flow water mover moving water between connecting canal segments
- Overland flow and canal-flow interaction moves water between cells having overland flow and canals
- Groundwater flow and canal-flow interaction moves water between the groundwater cells and canals

Unless blocked using no-flow boundary conditions, water can flow through all of the five default water movers. It is important to note that overland and groundwater flow interaction does not happen automatically. The user must specify overbank and seepage parameters. If these are *not* specified, the HSE will not create the associated water movers.

5.1.1 Naming Conventions

A number of conventions were used in writing this user's manual.

`filename`	name of a file used
`<xml_element_name>`	XML elements
`"nnn"`	xml attribute data values=nnn
`[abc]`	optional values

5.1.2 Software Setup Needed To Run The Model

The model is currently supported in Red Hat Linux 9.0. The following software environment is needed prior to running the model.

- A statically linked executable or a dynamically linked executable with proper libraries and `LD_LIBRARY_PATH` environment variables set
- The XML DTD file `hse.dtd`, with location specified in the main XML file
- The input XML data file, with the location of the DTD file specified in the `<!DOCTYPE hse SYSTEM>` XML tag
- Any required model geometry files, canal network files, boundary condition or initial condition input files

5.1.3 Steps And Data Needed To Run The Model

Significant effort is underway to create an RSM pre-processor that is graphically-based and will automate many of the steps required to create RSM applications, as discussed in Section 10.4.2. This pre-processor will directly create XML input data from GIS coverages, for example. Current functionality for the pre-processor is described in Section 10.4.1. The general steps required to build a model application are listed below.

1. *Create a triangular cell topographical mesh.* Import GIS coverages into GMS or any other mesh generator, and create a mesh file in GMS format covering the model domain. Certain rules are enforced regarding the cell enumeration and formatting, these rules are described in Section 3.2.
2. *Create mesh physical properties indexed files.* This step is currently under development. For more information, see Section 10.4.
3. *Assemble input/boundary condition files.* Obtain all required time-series input files required for boundary condition flows, heads, etc. in DSS or NETCDF formats.
4. *Assemble input XML model definition files.* This manual details the XML file syntax used as input to RSM.
5. *Execute the model.* XML syntax and content validation errors will be written to the file `xml.errors`. Time step execution information will optionally be echoed to the command terminal, linear system solver monitors will optionally be displayed in X-windows.

6. *Post-process model results.* After running RSM, the output data created can be viewed using HECDSSVUE, TECPLOT, IBM Data Explorer or even ARCVIEW. Python-based post-processing utilities are described in Section 10.4.1 and Section 10.4.2

5.1.4 HSE Specification Using XML

All XML style specifications are contained in the XML simulation file. All RSM XML documents begin with processing instructions given by the first three lines in the example below. If the RSM schema is used to validate the input as discussed in Section B.3.2, you will have to change the file back to the following format because the RSM source code and XML libraries have not yet been updated to utilize a XML schema document type.

```
<?xml version="1.0"?> <!DOCTYPE hse SYSTEM "../hse.dtd" [
<!ENTITY pseudocells SYSTEM "pseudo.xml"> <!ENTITY landscape
SYSTEM "landscape.xml"> ]> <hse version="0.1">
  <control
    tslen="24"
    .....
  </control>

 &pseudocells;
 .....

]>
```

These lines explicitly identify an XML document and indicate which version of XML was used. The name and the location of the DTD file is also indicated here. The content of the XML file creates a tree-like hierarchy of information. The uppermost `<hse>` element is termed the root element. All other elements are children of `<hse>`. Elements specific to the HSE, e.g. water movers, lakes and ponds, lake seepage, model output and pseudocells are nested within the root element named `<hse>`. A list of first-order children elements possible under the `<hse>` root element is given in Table 5.2 and depicted in Figure 5.1.

A “well-formed” XML for the HSE would look like:

```
<?xml version="1.0"?> <!DOCTYPE hse SYSTEM "../hse.dtd" [ ]>
<hse version=‘‘0.1’’>
  <control> ...    </control>
  <mesh> ...       </mesh>
  <network> ...    </network>
  <watermovers> .. </watermovers>
  <controller> ... </controller>
  <management> ... </management>
</hse>
```

where the ”hse” version attribute is used to ensure consistency between input specifications and version of HSE (not yet implemented). Space covering the dotted lines has to be

Table 5.2: *Definition of elements defined in the* `<hse>` *root element.*

Tag	Definition
`<control>`	All the program control parameters such as time step size, beginning time, ending time, etc. are defined using this XML element. See Table 6.1 for specific information.
`<mesh>`	Information regarding the 2-D mesh are defined within this XML element. See Table 6.3 for specific information.
`<network>`	Information regarding the canal network are defined within this XML element. See Table 6.3 for specific information.
`<watermovers>`	Water movers such as structures are defined here.
`<lakes>`	Lakes and ponds water bodies are defined here
`<multilayer>`	Information about 3-D or multi-layered groundwater is defined here.
`<controller>`	Information about controllers are provided here.
`<management>`	Information about management supervisors is defined here.
`<output>`	Specifying model output.
`<rulecurves>`	Information on watermover control rule curves.
`<basins>`	Not fully implemented in this version.

filled with other elements or attributes to be described later. Details and examples are given in the later chapters, the HSE benchmarks also provide numerous examples of HSE XML file usage. If any components are not present in the model, they can be skipped.

5.1.5 XML Elements Under The Root

All the model details are provided under the above-mentioned first-order children elements listed in the XML file. The dots in the XML simulation file shown above represent real information. A brief description of some of the children elements is given in this section. Remaining elements such as the `<mesh>` element that require more information will be described in Section 6.2.

5.2 Suggested Development Procedure for Applications

To reduce errors, it is convenient to build models up from a very basic overland flow model or a canal flow model, and gradually add features. It is easy to detect errors this way because the last component to be added is most likely to have caused the problem. When errors occur at any time, one of the methods of diagnosis involves taking out the RSM objects one at a time until the model starts to respond accurately. Commenting out lines in the XML file is an easy method to search for errors.

The majority of errors associated with running RSM are due to missing, incorrect, or mal-formed input data. If the problem is one of missing or extra data elements in the XML, the DTD syntax validation will create an error report named `xml.errors`. The schema validation methods will uncover more difficult to find data errors. If the XML data set passes the schema validation, it will run in RSM. However if the data problems are due to data falling outside a valid range of input, the schema validation routine is not yet completed at this level to catch this type of problem. The user has to be careful to make sure that the data is pre-processed to accurately reflect the physical parameters of the modeled domain.

5.2.1 Sequence Of Object Creation

RSM objects are created in a certain sequence partly for historical reasons, and partly because of the C++ object inheritance present in RSM. Therefore, the data set sequence is important because certain objects can be created only after other objects are created. It is safe to follow the XML data input ordering contained in the benchmarks to eliminate errors related to the ordering of the XML data. The first-order children elements should be arranged in the sequence that is shown in Figure 5.1. The sequencing of children elements that are nested beneath these are also diagrammed and can be viewed by navigating through the on-line data input guide.[1]

- The **`control`** sub-elements are shown in Figure 5.2.
- The **`mesh`** sub-elements are shown in Figure 5.3.
- The **`network`** sub-elements are shown in Figure 5.4.
- The **`watermovers`** sub-elements are shown in Figure 5.5.
- The **`lakes`** sub-elements are shown in Figure 5.6.
- The **`multilayer`** sub-elements are shown in Figure 5.7.
- The **`controller`** sub-elements are shown in Figure 5.8.
- The **`management`** sub-elements are shown in Figure 5.9.
- The **`output`** sub-elements are shown in Figure 5.10.
- The **`rulecurves`** and `basins` sub-elements are not shown because they are not fully implemented at this time.

5.3 RSM Directory Structure

The directory structure of RSM is organized as follows:

- **`hse`** - top level directory containing RSM
- **`benchmarks`** - test cases and hse.dtd file
- **`budtool`** - water budget tool
- **`doc`** - documentation
- **`fcl_lib`** - fuzzy control library
- **`glop`** - GNU linear programming kit library
- **`psbud`** - pseudocell water budget package
- **`src`** - source code and executable

---

[1] http://gwmftp.jacobs.com/xml_schema_corrected/graphics/hse_222.html

5.3.1 RSM Benchmarks

HSE incorporates a suite of standard benchmark tests to provide quality assurance and validation of new features added to the model. There are over 60 benchmarks. The individual benchmarks reside in subdirectories of the `hse/benchmarks/` such as `BM1`, `BM2`, etc. To run the benchmark suite, the user can perform the following commands from the hse root directory:

```
cd benchmarks
./test.script
```

The benchmarks can serve as a valuable training resource for the novice modeler. Most of the main features of the RSM are exercised in the benchmarks. Section 11.1 contains descriptions of all existing benchmarks. The most up-to-date description for your RSM release can be obtained by processing the benchmarks/descriptions.tex file with the tex or LaTeX program.

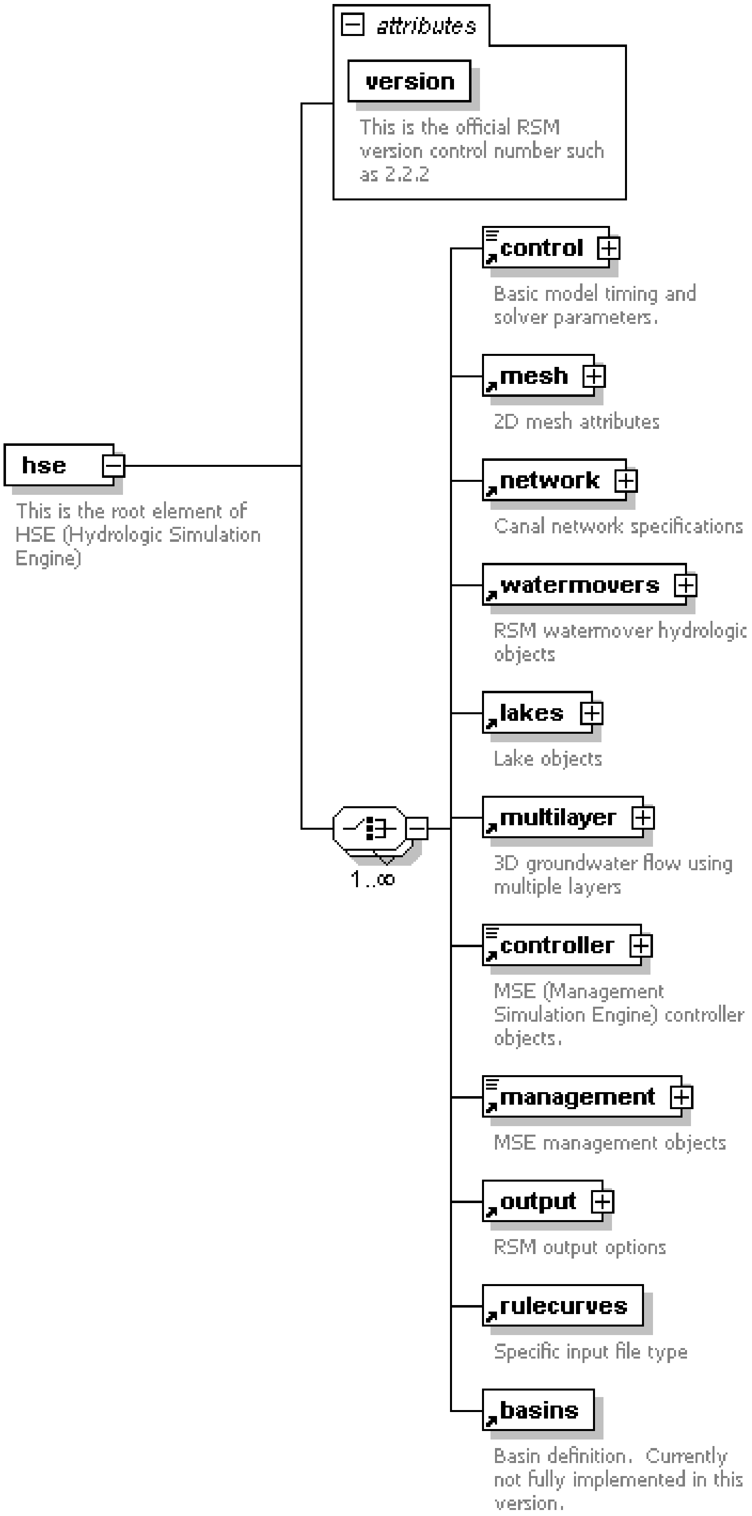


Figure 5.1: *The HSE root node and first-order children elements.*

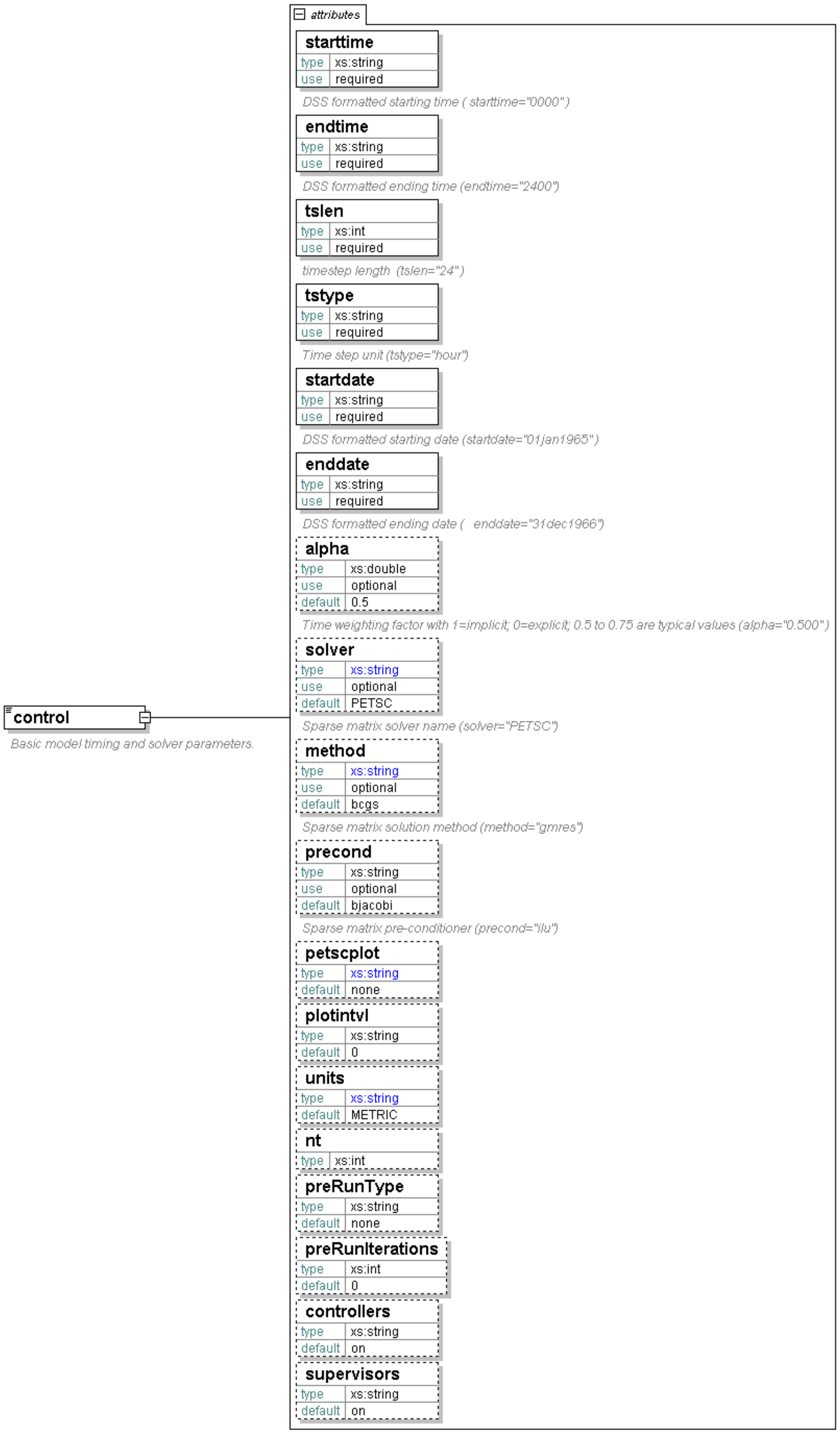


Figure 5.2: *The* `control` *subelements.*

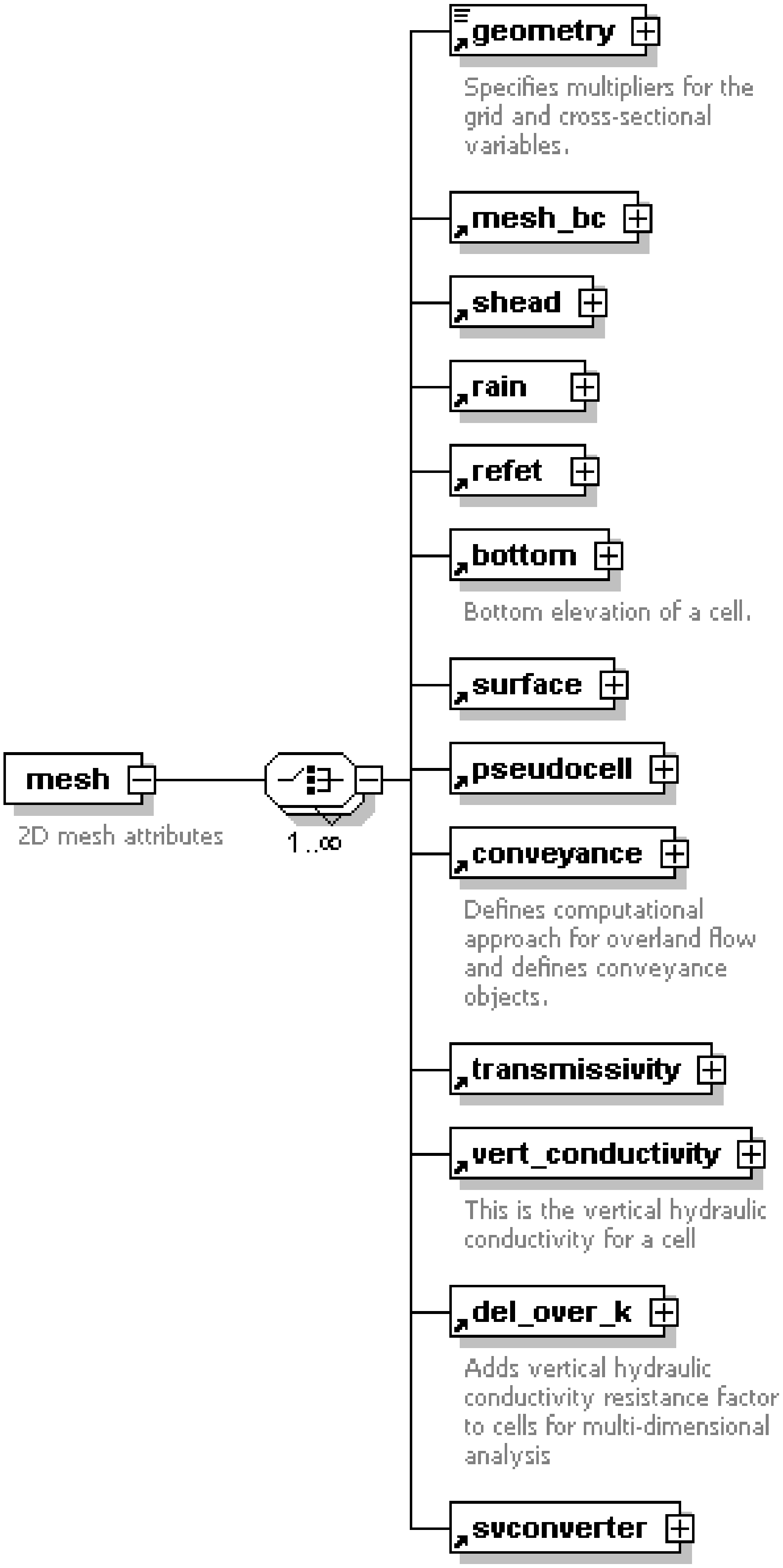


Figure 5.3: *The* `mesh` *subelements.*

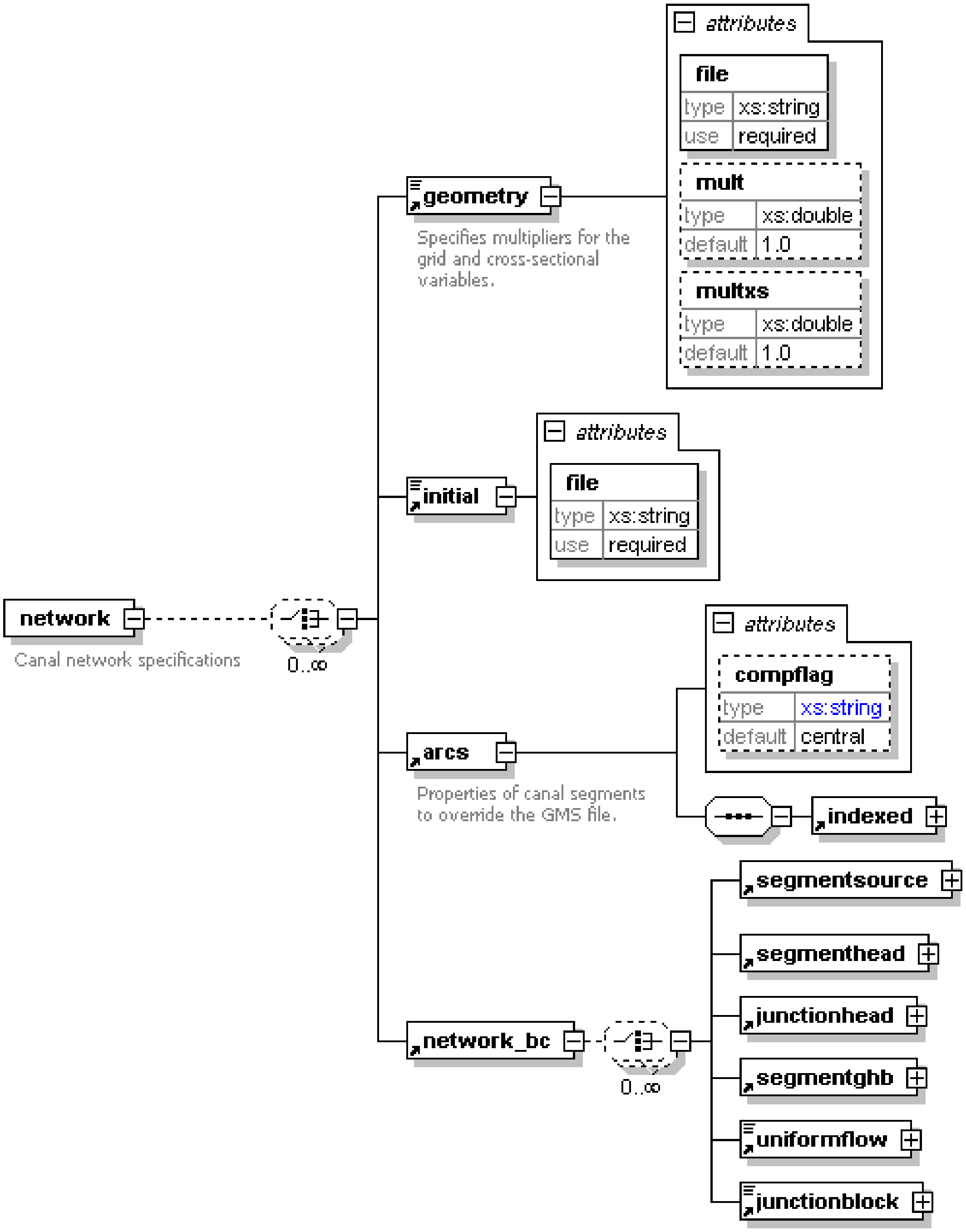


Figure 5.4: *The* `network` *subelements.*

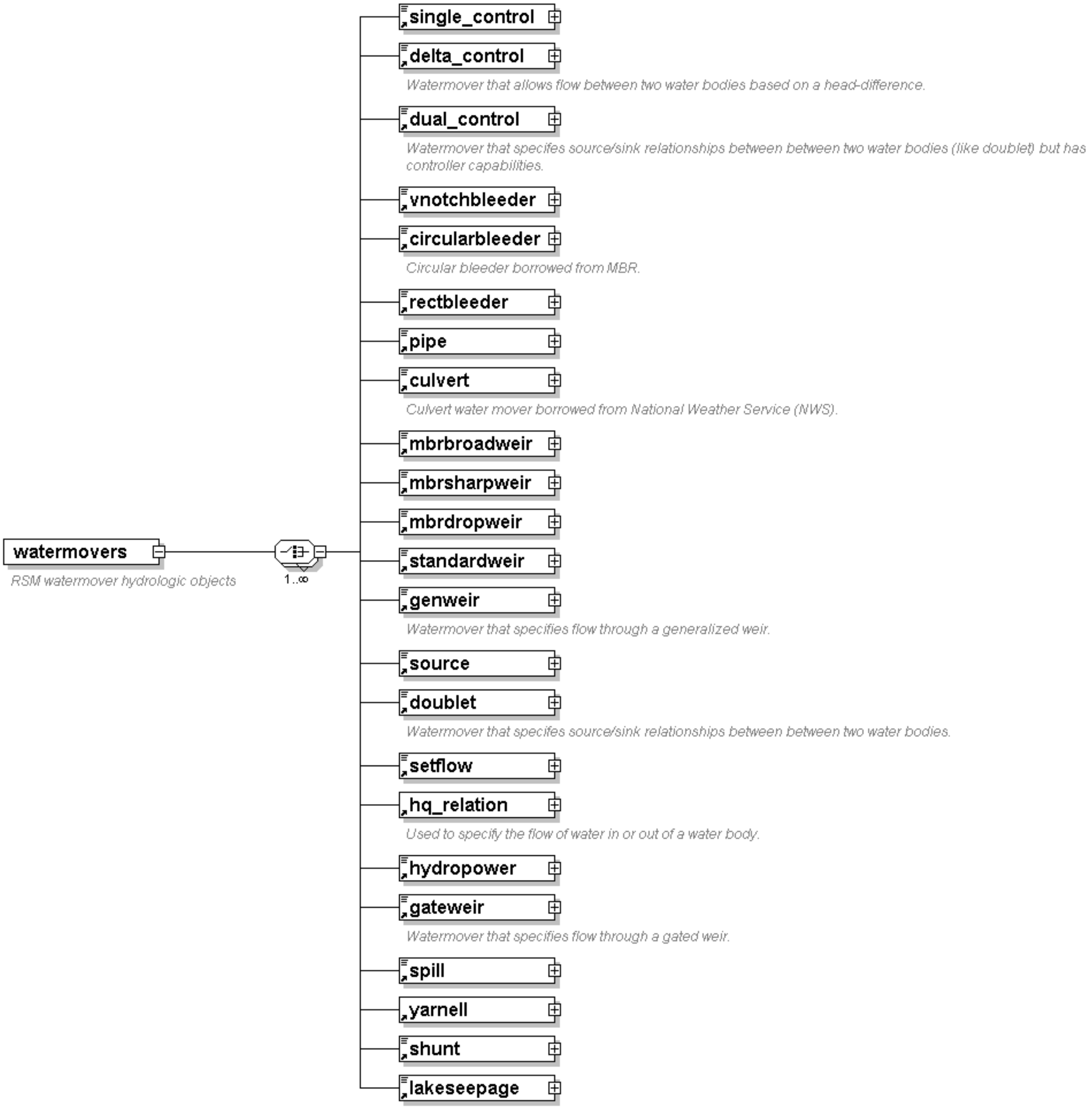


Figure 5.5: *The* `watermovers` *subelements.*

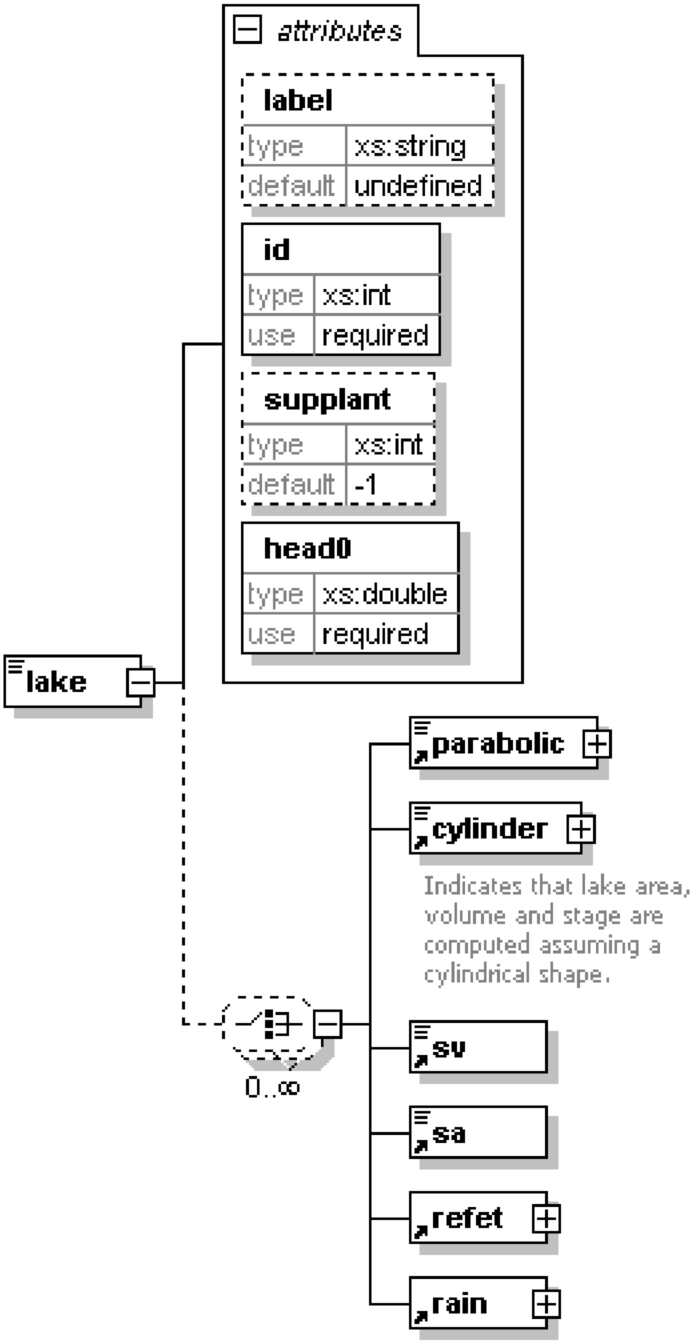


Figure 5.6: *The* `lakes` *subelements.*

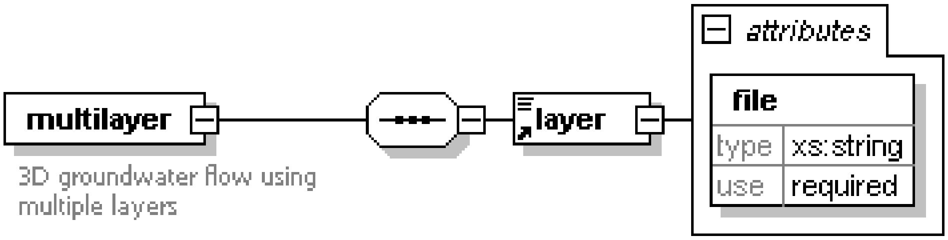


Figure 5.7: *The* `multilayer` *subelements.*

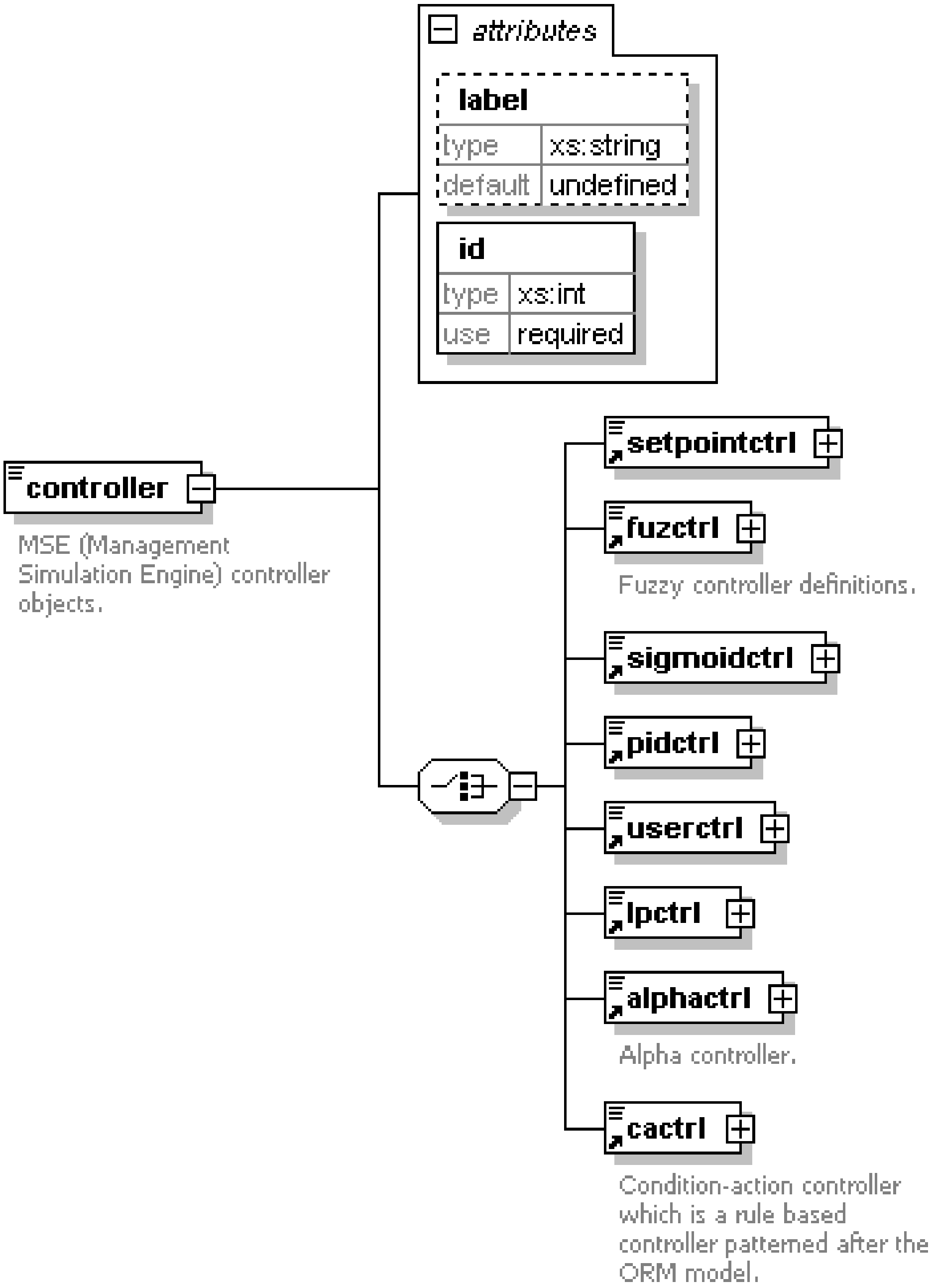


Figure 5.8: *The* `controller` *subelements.*

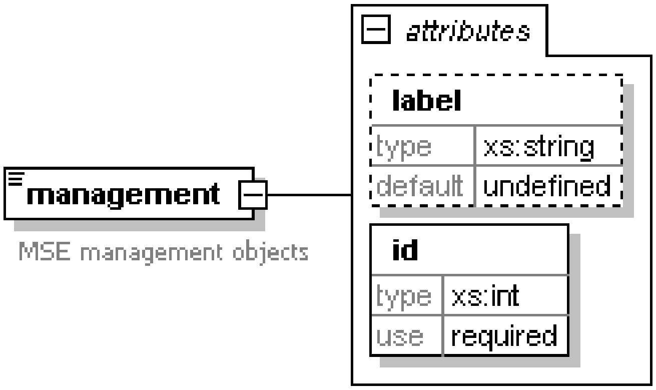


Figure 5.9: *The* `management` *subelements.*

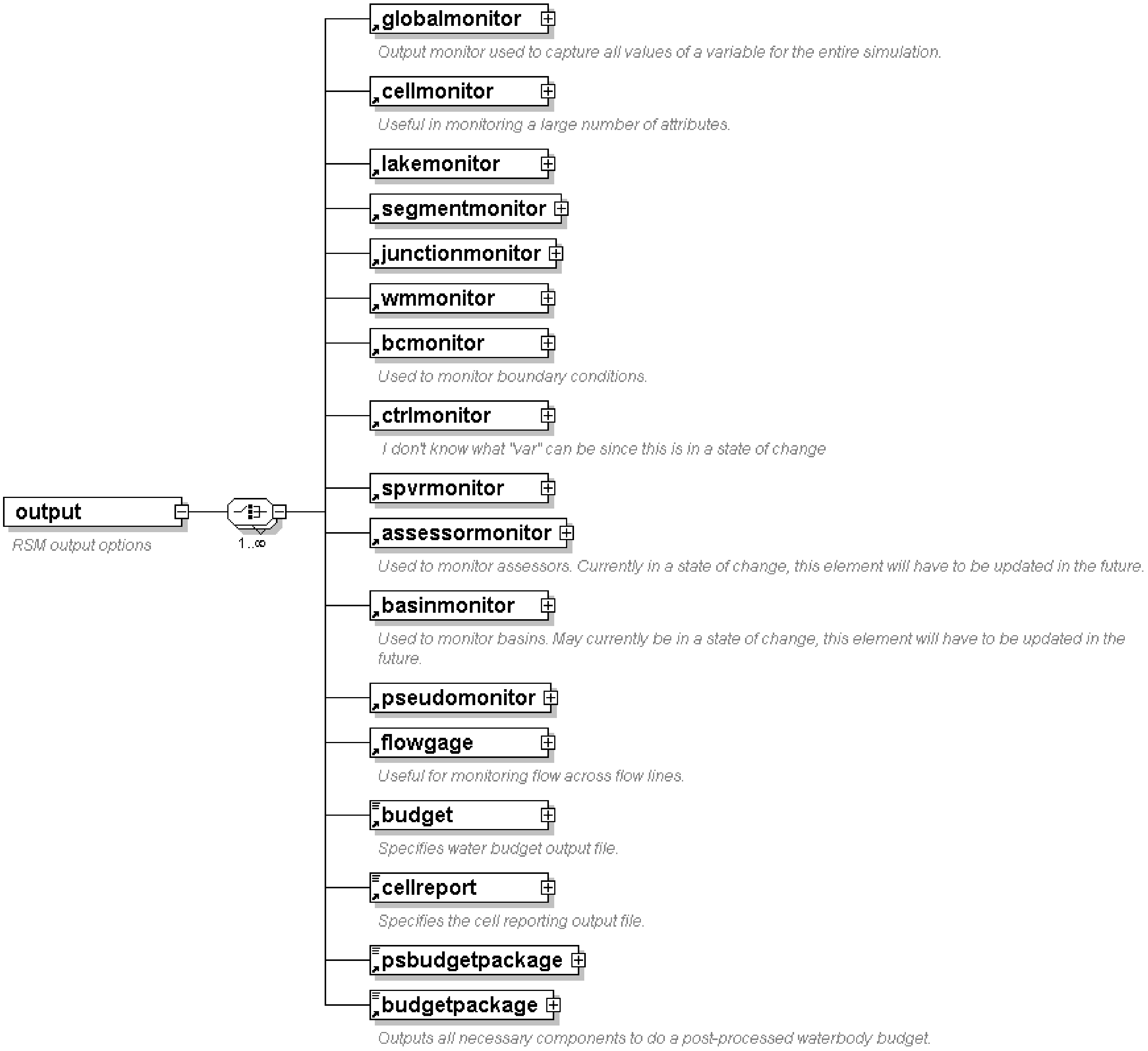


Figure 5.10: *The* `output` *subelements.*

Chapter 6

HSE Model Components and XML Input

This chapter contains details of the model components (i.e., hydrologic objects) and the XML instructions that represent these objects. There are over 200 XML elements and over 700 attributes corresponding to these elements that are supported in Version 2.2.2 of the model. Details are provided for the XML input of each object. The functional, but not fully implemented 3D groundwater flow input requirements, are presented in Appendix C.

6.1 Basic Model Set-Up Parameters - The XML `<control>` Element

The XML `<control>` element is used to define the basic model set-up parameters, which includes a variety of terms related to time stepping, solver choices, and other topics. Table 6.1 contains the XML attributes used within the `<control>` element. The order of placement of the tags inside the `<control>` element is not important. If not specified, some defaults will be assigned as shown in the on-line data input guide.[1] The default values are shown for attributes such as alpha and solver.

---

[1] http://gwmftp.jacobs.com/xml_schema_corrected/graphics/hse_222.html#element_control_Link03048EC8

Table 6.1: *Attributes defined with the XML* `<control>` *element.*

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`starttime`	Starting time of the simulation	DSS format hhmm	0000 to 2359	`starttime="2315"`
`endtime`	Ending time of the simulation	DSS format hhmm	0000 to 2359	`endtime="0120"`
`tslen`	Timestep length	int	Greater than 0	`tslen="24"`
`tstype`	Timestep unit	string	`second, minute, hour, day, week`	`tstype="hour"`
`startdate`	Starting date of the simulation	DSS format ddmmmyyyy	any valid date	`startdate ="01jan1994"`
`enddate`	Ending date of the simulation	DSS format ddmmmyyyy	any valid date	`enddate ="30jan1994"`
`alpha`	Time weighting factor	double	1.0 = implicit, 0.5 = mid, 0.0 = explicit	`alpha="0.75"`
`solver`	Sparse solver name	string	`PETSC` is the only option	`solver="PETSC"`
`method`	Sparse solver method	string	see `PETSC` manual	`method="gmres"`
`precond`	Pre-conditioner	string	see `PETSC` manual	`precond="bjacobi"`
`nt`	Number of time steps to be simulated. Steady state = 0	integer	any integer	`nt="10"`
`units`	Model units	string	”METRIC” is default,”ENGLISH” is optional	`units="english"`

Table 6.1 continued on next page

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`petscplot`	Solver monitors	string	options: ”none”, ”yes”, ”text”, ”ksp”, ”all”	`petscplot="all"`
`PreRunType`	Used to specify the type of pre-running of the simulation.	string	Options include ”controller”, ”filter”, ”all”, ”none”=default	`preRunType ="none"`
`PreRunIterations`	Number of iterations used during the pre-run.	integer	Any integer, defaults to 0.	`preRunIterations ="0"`
`controllers`	Activate controllers	string	”on” (default) or ”off”	`controllers="on"`
`supervisors`	Activate supervisors	string	”on” (default) or ”off”	`supervisors="on"`

An example of a data definition used with a `<control>` keyword is shown below.

```
<control
  tslen="15"
  tstype="minute"
  startdate="01jan1994"
  starttime="0000"
  enddate="01jan1994"
  endtime="0230"
  alpha="0.500"
  solver="PETSC"
  method="gmres"
  precond="ilu">
</control>
```

6.1.1 Model Units

By default the model uses SI units. However, input data in English units can be used by entering the attribute `units="english"` as described in Table 6.1. Any other system of units can be implemented by using optional multipliers as described later. Some of the units and data types used with the DSS file format are described in Table 6.2.

Table 6.2: *Default units used by HSE.*

Quantity	Unit	Type
Head	METERS	INST-VAL
Flow	CU_METER/SEC	INST-VAL
Rain	METERS	PER-CUM
ET	METERS	PER-CUM
Depth	METERS	PER-CUM
Water level	METER	INST-VAL
Transmissivity	METER^2/SECOND	PER-AVER
Volume	CU_METER	PER-CUM
Storage	CU_METER	INST-VAL

6.2 Data Input For The Two-Dimensional Model - The XML `<mesh>` Element

Setting up the basic two-dimensional regional model requires geometric input data to describe sizes, shapes, locations of cells, and elevations of the bottom and the ground surface of each cell. Additional information to characterize the hydrologic properties of each cell includes the relationship between head and volume of water stored, the description of groundwater and surface water flow properties and mechanisms, and the description of the local hydrologic processes for each cell from the assignment of a pseudocell to each mesh cell. The remaining data under the `<mesh>` element are the forcing functions that drive the regional flow. These are rainfall, evapotranspiration, and boundary conditions for the cells and the walls that divide the cells.

Data for the 2-D model is entered in the `<mesh>` environment. These data are described in detail in subsequent sections. The elements under which these data are input are listed in Tables 6.3, 6.4, and 6.5. Table 6.3 describes how to specify the 2-D geometry file that specifies the node locations. The elements listed in Table 6.4 describe data that are explained in greater detail in later sections, and Table 6.5 lists additional data that are read from data files and the file formats available. The attributes needed to describe the data input for the elements in Table 6.5 are listed in Table 9.1 and Table 9.9.

Table 6.3: *Specification of the geometry file under* `<mesh>`*.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<geometry>`	Creates the environment to specify a geometry file in GMS format.			
file	The name of the file	String	Any valid GMS file name	enp.gms
mult	Multiplier for mesh node coordinates, often for unit conversion	Real	Any valid real	0.3048
Details of the GMS file are explained in Section 6.3				

Table 6.4: *Specification of additional mesh properties.*

Element	**Definition**	**Available subelements**

`<transmissivity>`	The element under which transmissivity for groundwater flow is specified. Details are given in Section 6.4.2	`<indexed>`
		`<confined>`
		`<lookuptr>`
		`<unconfined>`
		`<layered>`

`<conveyance>`	Creates the environment for specifying the calculation of conveyance for overland flow. Details are given in Section 6.4.1	`<indexed>`
		`<manning>`
		`<cadlec>`
		`<layerc>`
		`<lookup>`

`<svconverter>`	Creates the environment for specifying the stage-volume of water storage relationship for each cell. Details are given in Section 6.8	`<indexed>`
		`<constsv>`
		`<lookupsv>`
		`<layersv>`

`<pseudocell>`	Assignment of pseudocell types to mesh cells are made under this element. Details are given in Chapter 8	`<indexed>`
		`<layer5>`
		`<layer1nsm>`
		`<nam>`
		`<unsat>`
		`<mbrcell>`
		`<afsirs>`
		`<layerpc>`

`<mesh_bc>`	Creates the environment to specify the mesh boundary conditions for cells and walls. Details are given in Chapter 7	`<well>`
		`<cellhead>`
		`<cellghb>`
		`<wallhead>`
		`<wallghb>`
		`<noflow>`
		`<walluf>`

Table 6.5: *Elements for specifying input formats for additional mesh properties under* `<mesh>`.

Element	**Definition**	**Available input formats**

`<bottom>`	Creates the environment for specifying the bottom elevation of each cell.	`<indexed>`
		`<const>`
		`<gms>`

`<surface>`	Creates the environment for specifying the ground surface elevation of each cell.	`<indexed>`
		`<const>`
		`<gms>`

`<shead>`	The cell heads at the beginning of the simulation are specified under this element.	`<indexed>`
		`<const>`
		`<gms>`

`<rain>`	The rainfall on each cell during the simulation is specified in this environment.	`<indexed>`
		`<const>`
		`<dss>`
		`<gridio>`
		`<netcdf>`
		`<gms>`

`<refet>`	The reference ET for each cell during the simulation is specified in this environment.	`<indexed>`
		`<const>`
		`<dss>`
		`<gridio>`
		`<netcdf>`
		`<gms>`

Table 6.6: *Elements and attributes used with the* `<indexed>` *element.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
file	name of the index file	String	A valid file name	refet_index.dat
`<entry>`	Creates environment for entry data.			
id	Entry ID number	Integer	Any integer	3
`<const>`	A constant value option			
`<dss>`	A DSS option			
`<rc>`	A rule curve option			
`<csv>`	A csv ASCII file option			
`<asciiform>`	An asciiform ASCII file option			
Details of the `<dss>`, `<const>`, `<rc>`, `<csv>` and `<asciiform>` formats are described in Chapter 9.				

6.2.1 Attributes of the Data File Formats Used In The `<mesh>` Environment

Among all the types, `<indexed>` is special because it can be used to assign a mixture of sub-types to different cells, by using integers in an index file, and defining the types under the element `<entry>`. Formats `<indexed>`, `<gridio>`, `<gms>` and `<netcdf>` and the attributes are shown in Tables 6.6, 6.7, 6.8, and 6.9. Examples are shown later in this section. Format and attributes for `<const>` and `<dss>` are explained in Chapter 9.

Certain basic types of data are required for a model run. In the case of 2-D flow, all the information about 2-D cells listed in Tables 6.3, 6.4 and 6.5 are needed to describe all cells fully. Water movers, in addition to the default water movers that are created after the mesh is defined, may be needed depending on the system being simulated. When a model is constructed, it is better to assemble the components a few at a time, so that the evolution can be followed, and bugs detected as early as possible and fixed in a systematic way before they become too numerous.

Table 6.7: *Attributes used with* `<gridio>`*.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`file`	name of the file	String	A valid gridio file name	rain.dat
`dbintl`	database interval used in the gridio file (minutes)	Integer	-1 for steady state or > 0	1440.
`xorig`	X co-ordinate of the origin	Real	Any real	543329
`yorig`	Y co-ordinate of the origin	Real	Any real	286761
`mult`	multiplier for the data.	Real	Any real	0.0254

Table 6.8: *Attributes used with* `<gms>`*.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`file`	name of the GMS file	String	A valid GMS file name	enp.gms
dbintl	database interval used in the GMS file (minutes)	Integer	> 0	1440.
`mult`	multiplier for the data values	Real	Any real	0.3048

Table 6.9: *Attributes used with* `<netcdf>`.

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`file`	name of the netcdf file	String	A valid netcdf file name	coast.dat
`variable`	Name of the variable (optional)	String	A valid variable name	rain
dbintl	database interval used in the netcdf file (minutes)	Integer	> 0	1440.
`mult`	multiplier for the data values	Real	Any real	0.3048
`units`	units of the variable (optional)	string	Any string	meters

6.2.1.1 Examples Of 2-D Data Defined Within `<mesh>`

The following examples demonstrate the use of modifiers to describe a variety of data sources. In the following example, the geometry file is specified with the multiplier omitted with a default of 1.0, the initial heads are read from a GMS file with a default time interval of 1440 minutes and a multiplier of 1.0. The ground surface elevation is input as a constant value of 500 meters, the transmissivity is of the confined type with a value of 0.04 and `<svconverter>` uses a constant storage coefficient of 0.2.

```
<geometry file="mesh3x3.2dm"> </geometry>
 <shead><gms file="hin3x3.dat"></gms></shead>
 <bottom>  <const value="0.0">   </const> </bottom>
 <surface> <const value="500.0"> </const> </surface>
 <conveyance>
   <mannings a="1.000" detent="0.00001"></mannings>
 </conveyance>
 <transmissivity>
   <confined trans ="0.04"></confined>
 </transmissivity>
 <svconverter>
   <constsv sc="0.2"></constsv>
 </svconverter>
```

In the example below the rainfall data file `sfrain_v1.2.bin` is in binary `gridio` format, with an x and y origin (543329, 286761) of the gridio input file. The time step is one day (1440 minutes) and the multiplier converts inches to meters.

```
<refet>
  <gridio file="/vol/hsm/data/db/grid_io/rain/sfrain_v1.2.bin"
    xorig="543329" yorig="286761" mult="0.0254" dbintl="1440">
  </gridio>
</refet>
```

If the ET data is to be presented in the `<dss>` format, the following example shows a segment of the input XML file.

```
<rain>
  <dss file="rain.dss"
    pn = "/S8/S8 canal/FLOW/01JAN1994/1DAY/normal/"
      mult="0.0254" dbintl="1440">
  </dss>
</rain>
```

If unconfined flow is to be specified, with hydraulic conductivity, $k = 0.02\ m/sec$ the following XML input may be used.

```
<transmissivity>
  <unconfined k = "0.02"> </unconfined>
</transmissivity>
```

6.3 Two-Dimensional Grid Generation - The XML `<geometry>` Element

Geometry data for the two-dimensional overland flow / groundwater flow model are described in this chapter. These data consist of the (x,y) coordinates of each node and the node connectivity. Each cell is numbered and the nodes that the define each cell are associated with the cell. The name of the geometry file is entered in the `<mesh>` environment using the `<geometry>` element. An example XML input for specifying the GMS file is displayed below. The file name is 18.2dm and a multiplier of 0.3048 is specified to convert feet to meters.

```
<?xmll version="1.0"?> <!DOCTYPE hse SYSTEM "../hse.dtd"[ ]>
  <hse version="0.1">
  <geometry file="l8.2dm" mult="0.3048">
  </geometry>
... ... </hse>
```

The geometry file for HSE can be created graphically using the GMS package. The resulting GMS file does not need much modification. A small sample mesh is shown in Figure 6.1 with cells and nodes numbered. A GMS file that describes the mesh is shown in Table 6.10. The mesh consists of 16 nodes and 18 cells with the equilateral leg of each mesh cell 5000 m long. (This is the typical mesh used in the benchmarks and model testing chapter)

The first line of the GMS file *must* be MESH2D. Each line that describes the node connectivity begins with E3T and each line describing the node locations begins with ND. The format of the node connectivity information is

```
E3T IC N1 N2 N3 NN
```

where, IC is the cell number and N1, N2, N3 are the nodes defining cell IC in a *counter-clockwise* direction. NN is not currently used in the model. After the connectivity is defined, nodal coordinates are defined using the format

```
ND IN X Y Z
```

where, ND designates that these are nodal coordinates, IN is the node number, and X and Y are the coordinates of the node. The last entry, Z, is not used in the 2-D model and is set equal to 0.0.

As an example, in this mesh, cell 12 is defined by nodes 3, 7, and 8 and the coordinates of node 7 are x = 10000.000, y = 10000.00.

Table 6.10: *two-dimensional GMS mesh data file* `"mesh3x3.2dm"`*.*

```
MESH2D
 E3T     1    1     6     2     1
 E3T     2    2     7     3     1
 E3T     3    3     8     4     1
 E3T     4    5    10     6     1
 E3T     5    6    11     7     1
 E3T     6    7    12     8     1
 E3T     7    9    14    10     1
 E3T     8   10    15    11     1
 E3T     9   11    16    12     1
 E3T    10    1     5     6     1
 E3T    11    2     6     7     1
 E3T    12    3     7     8     1
 E3T    13    5     9    10     1
 E3T    14    6    10    11     1
 E3T    15    7    11    12     1
 E3T    16    9    13    14     1
 E3T    17   10    14    15     1
 E3T    18   11    15    16     1
 ND  1        0.000       15000.000 0.
 ND  2     5000.000       15000.000 0.
 ND  3    10000.000       15000.000 0.
 ND  4    15000.000       15000.000 0.
 ND  5        0.000       10000.000 0.
 ND  6     5000.000       10000.000 0.
 ND  7    10000.000       10000.000 0.
 ND  8    15000.000       10000.000 0.
 ND  9        0.000        5000.000 0.
 ND 10     5000.000        5000.000 0.
 ND 11    10000.000        5000.000 0.
 ND 12    15000.000        5000.000 0.
 ND 13        0.000           0.000 0.
 ND 14     5000.000           0.000 0.
 ND 15    10000.000           0.000 0.
 ND 16    15000.000           0.000 0.
```

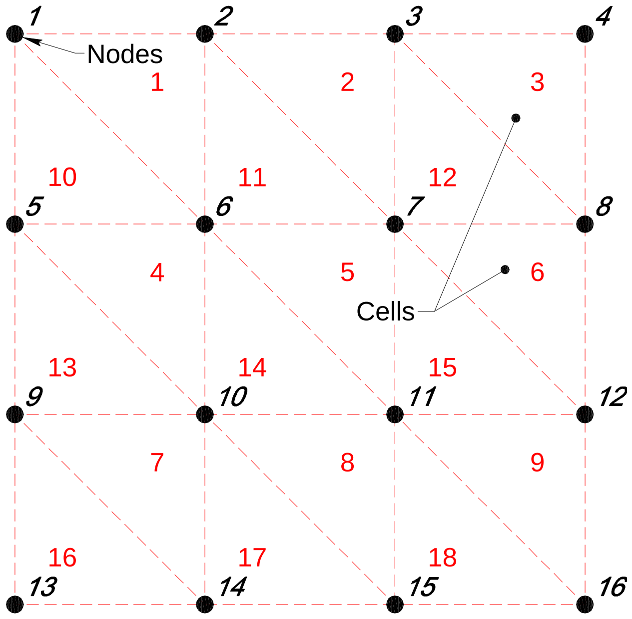


Figure 6.1: *Discretization of a square area into 18 cells with 16 nodes.(See Table 6.10).*

6.4 Alternative Forms Of 2-D Flow Equations

Different equations of overland flow may be appropriate for different regions of the model domain. Overland flow is described in the model under the `<conveyance>` element and, and groundwater flow is described under `<transmissivity>`. For example, flow through wetlands is different from flow across a golf course or a ridge and slough area. While Manning's equation is used most often, other formulations of overland flow are available in the HSE.

6.4.1 Overland Flow Options

In an integrated model, overland and groundwater flows are modeled together. In the formulation used in HSE, these flows can be separated. Overland flow is commonly characterized as conveyance. Manning's equation, which is valid mostly under turbulent flow conditions, is often utilized. For overland flow

$$Q = \frac{L}{n} d^{\frac{5}{3}} \sqrt{S} \tag{6.1}$$

where
$L =$ length of the flow face perpendicular to the flow direction,
$n =$ Manning's coefficient,
$d =$ water depth, and
$S =$ water surface slope.

When a thick layer of vegetation is present, the equation can be modified with a constant n value since the relative roughness decreases with the depth of water. One way to address this problem is to use the following relationship to represent n

$$n = Ad^B \tag{6.2}$$

where
$d =$ water depth, and
A and $B =$ empirical constants.

This modification to Manning's equation is still not sufficient under many wetland conditions with thick vegetation. In this case, another power law equation can be used to replace Manning's equation. Under laminar or transition flow conditions found in wetlands, the $\sqrt{S}$ term needs to be replaced by S^α where α is a user-defined exponent. The form used by (Kadlec and Knight, 1996) gives

$$Q = Lad^\beta S^\alpha \tag{6.3}$$

where
$Q =$ volume flow rate,
$L =$ width of flow,
$d =$ flow depth, and
a, α and $\beta =$ empirical constants.
A general form of this equation uses a lookup table to describe the behavior of conveyance with depth, assuming α can be considered as constant.

$$Q = LC(d)S^{\alpha} \tag{6.4}$$

in which, $C(d)$ is a lookup table that describes the variation of conveyance with depth. Such modifications can also be used to simulate effects of microtopography found in the ridge and slough formations. This is useful when the ground surface has significant relief or the nature of the vegetation makes Manning's equation (equation 2.4) or Kadlec's equation (equation 6.8) (Kadlec and Knight, 1996) inappropriate. To use the lookup table, the average land surface elevation of a cell is specified as well as two other elevations. The elevation z_{lo} in Figure 6.2 is defined as the elevation at which overland flow will begin. Elevation $H > z_{hi}$ is the elevation at which the flow can be assumed to be turbulent, and Manning's equation with $S^{1/2}$ becomes valid. When the water level is below z_{lo}, there is no overland flow. In the range $z_{lo} < H < z_{hi}$, the following equation is used.

$$Q = LC(d)S^{\alpha} \tag{6.5}$$

where
$C(d) =$ lookup table of conveyance as a function of depth.

6.4.1.1 Conveyance Type `<mannings>`

Data for conveyance are entered within the `<mesh>` environment under the `<conveyance>` element. The environments available under `<conveyance>` are described in Table 6.11. They can be used to assign conveyance values by

- Assigning any of the conveyance formulations to the entire model domain.
- Assigning different formulations `<mannings>, <lookup>, <cadlec>` to different parts of the model domain using an index file to specify the distribution.

An example of the XML input required to assign a `<mannings>` type conveyance to different zones of the model domain is shown below. In this example, Manning's equation is used to compute conveyance with the value of the roughness coefficient computed as

$$n = 0.5d^{-0.77} \tag{6.6}$$

Table 6.11: *Elements and attributes under* `<conveyance>`.

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<compute>`	Describes methods of computing average conveyance between cells. Details are described in Table 6.12.			
`<indexed>`	Indicates that conveyance methods are specified under `<entry>` and assigned to cells with an index file.			
`<mannings>`	The Manning equation is used and the roughness coefficient is defined as $n = A(depth)^b$.			
a	The coefficient A in equation 6.2	Real	> 0.0	0.5
b	The exponent b in equation 6.2	Real	> 0.0	0.1
detent	Minimum value of d in Equation 6.2	Real	> 0.0	0.1
`<cadlec>`	The conveyance is computed from Kadlec’s equation 6.3.			
a	The coefficient a in equation 6.3	Real	> 0.0	0.5
b	The exponent α in equation 6.3	Real	> 0.0	0.1
beta	The exponent β in equation 6.3	Real	> 0.0	0.1
detent	Water depth below which there is no flow	Real	> 0.0	0.1
`<lookup>`	The conveyance is read from a lookup table.			
below	The depth below the ”surveyor’s” land surface below which overland flow ceases	Real	> 0.0	0.1
above	The depth above the ”surveyor’s” land surface above which the Manning equation applies	Real	> 0.0	0.1
base	The parameter A in the equation for the Manning’s roughness coefficient	Real	> 0.0	0.1
exponent	The parameter B in the equation for the Manning’s roughness coefficient	Real	> 0.0	0.1
convey	A lookup table of depth and conveyance for the range between (land surface - below) and (land surface + above)	Real	> 0.0	0.1

with the conveyance = 0 for water depths less than 0.01 m, or by

$$n = 0.4d^{-0.63} \tag{6.7}$$

with conveyance = 0 for depth less than 0.02 m. The zones where each of the two values for n are applied is defined by the index file, lu.index. Using an index file, the conveyance over a large region of the model domain may be changed by changing one entry under `<indexed>`.

```
<conveyance>
  <indexed file="lu.index">
    <entry id="1" label="type1">
      <mannings a="0.5" b="-0.77" detent="0.01">
      </mannings>
    </entry>
    <entry id="2" label="type2">
      <mannings a="0.4" b="-0.63" detent="0.02">
      </mannings>
    </entry>
  </indexed>
</conveyance>
```

6.4.1.2 Conveyance Type `<cadlec>`

The `<cadlec>` conveyance method can be assigned to the entire model domain with the XML example below. In this case, discharge is computed as

$$Q = L(2.0(10^6))d^3 S^{0.6} for \quad d > 0.01 \tag{6.8}$$

```
 <conveyance>
    <cadlec a="2.0E6" b="3.0" beta = "0.6" detent="0.01">
    </cadlec>
</conveyance>
```

6.4.1.3 Conveyance Type `<lookup>`

This method allows the conveyance to be read from lookup tables based on field measurements or other sources. The equation for overland flow is

$$Q = LC(d)S^{\alpha} \tag{6.9}$$

where $C(d)$ = a lookup table function of conveyance versus depth and S = slope. This equation applies between the elevations (Land Surface - below) and (Land Surface + above). Below this range there is no overland flow and above this range the Manning equation is used. In the following example, Manning's equation is used over the portion of the model domain where the index file specifies index id = 1. Where index id = 2 is specified, overland flow is computed from the conveyance in a lookup table specified within the environment `<convey>` for the range of heads between (Land Surface - 10) and (Land Surface + 30). The value of α is 0.7. Above this range the Manning equation is applied with n = 1.0 and below this range there is no overland flow.

```
<conveyance>
  <indexed file="man.index">
    <entry id="1" label="type1">
      <mannings a="1.000" detent="0.00001">
      </mannings>
    </entry>
    <entry id="2" label="type2">
      <lookup below ="10.0" above="30.0" base="1.0" expon="0.7">
        <convey>
             0.0   0.0
            10.0 100.0
            20.0 400.0
            30.0 500.0
        </convey>
      </lookup>
    </entry>
  </indexed>
</conveyance>
```

6.4.1.4 Mixing Overland Flow Types `<compute>`

Overland flow computations may involve different land use types in adjacent cells requiring different conveyance functions. When this happens the overland flow water mover needs the average conveyance between the cells, or something even more complex than simple averaging. Simple averaging or central differencing has its limitations under certain flow conditions, and can create oscillations at large gradients. Pure upwinding may be too diffusive. Considering this, a number of options are available for conveyance computations based on upwind or central methods. These options are described in Table 6.12. An example demonstrating this option is shown below.

```
<conveyance compute = "sep-upwind-par">
  <indexed file="lu.index">
    <entry id="1">
      <mannings a="1.000" detent="0.00001"></mannings>
    </entry>
  </indexed>
</conveyance>
```

Table 6.12: *Overland flow options for the* `<compute>` *environment under* `<conveyance>`.

Attribute	Definition
mixed	When using the `compute = "mixed"` option, properties of overland flow depths and parameters are averaged to compute average properties which are used in computing the conveyance. This option can be used only when Manning's type conveyance is used. An average can be determined only if both cells have the same type.
sep-central	Under this option, the two cells can have two different conveyance functions which will be linearly averaged to compute the water mover conveyance.
sep-upwind-par	This partial upwinding assumes upwind method when the upwind conveyance is less that the downwind, and average otherwise. This helps to make sure that upwinding is used when really necessary.
sep-upwind	This is the complete upwind method applied whether the conveyance types are the same or different.

6.4.2 Groundwater Flow `<transmissivity>`

For groundwater flow, transmissivity may be simply computed from a single hydraulic conductivity

$$Q = LkdS \tag{6.10}$$

where
L = width of the aquifer,
d = the aquifer thickness,
k = average hydraulic conductivity, and
S = head gradient (i.e., hydraulic gradient) in the direction of flow.

When the aquifer is layered, a lookup table can be used to obtain the transmissivity of the aquifer at different heads. This has the advantage that all the information about the horizontal stratigraphy can be captured into one equation. The groundwater discharge is computed as

$$Q = LK(H)S \tag{6.11}$$

where
Q = discharge,

L = length of the wall,
$K(H)$ = lookup table function describing transmissivity as a function of head.

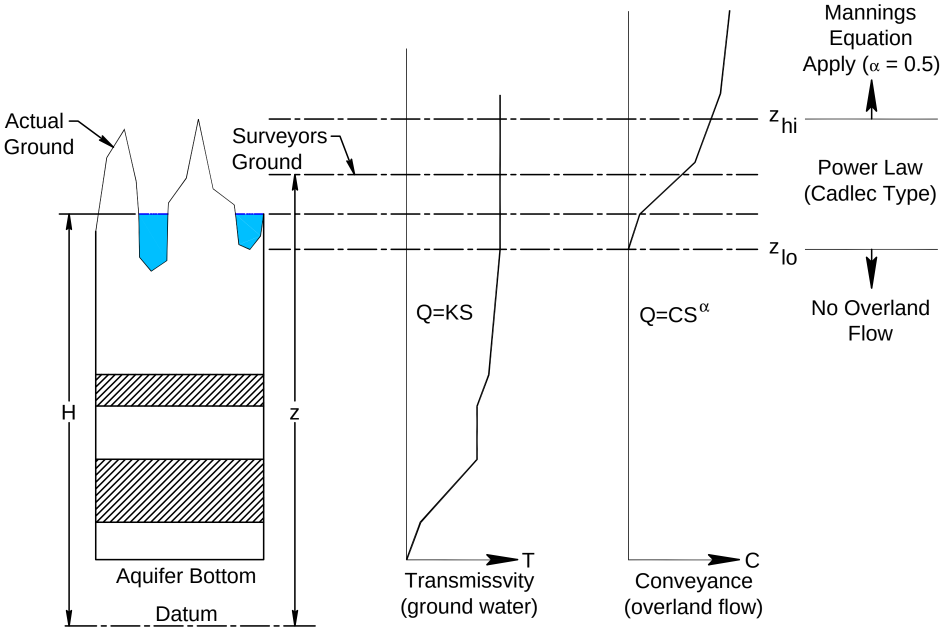


Figure 6.2: *Definition sketch for using lookup tables for transmissivity and conveyance.*

Following is an example where mixed transmissivity types are used in a model with various regions specified with indices. Index 1 in the example is a transmissivity lookup table where the two columns show the elevation and the transmissivity. The conveyance read from the lookup table is used between elevations 490.0 and 510.0. For index 2, the transmissivity is the product of the depth of water in the aquifer and the hydraulic conductivity $k = 0.04$.

Table 6.13: *Definition of variables for* `<lookuptr>` *under* `<transmissivity>`.

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<lookuptr>`	Denotes a lookup table of transmissivity as a function of head.			
below	The elevation of the lower end of the lookup table	Real	> 0.0	490.0
above	The elevation of the upper end of the lookup table	Real	> 0.0	520.0
flstat	*Not available in current version*	string	y or n	y
`<transm>`	A 1D lookup table of depth and transmissivity			

```
<transmissivity>
    <indexed file="tran.index">
      <entry id="1" label="type1">
        <lookuptr below ="490.0" above="510.0">
          <transm>
               0.0   0.0
             100.0  10.0
             400.0  15.0
             600.0  20.0
          </transm>
        </lookuptr>
      </entry>
      <entry id="2" label="type2">
        <unconfined k=".04">
        </unconfined>
      </entry>
    </indexed>
</transmissivity>
```

6.5 Water Movers

6.5.1 Introduction to Water Movers

Movement of water between water bodies in the HSE can take place only through water movers. Water mover objects contain functions to compute the flow of water from one water body to another.

Water movers fall into three general categories.

1. **Default water movers** are automatically created when the mesh and canal network are set up. Overland flow and groundwater flow water movers between adjacent cells in the mesh, and canal flow between adjacent canal segments are examples of default water movers that are created automatically based on the 2-D mesh or canal network geometry files.
2. **Concept water movers** in which water flow is computed using generic equations that can be used to represent actual structures in a limited way. Lookup tables, time series, and power functions are examples of concept water movers. These are intended to provide flexibility for the user to represent movement of water with methods that are not included in the other categories.
3. **Physical structure water movers** are designed to represent man-made structures such as weirs, culverts, and orifices.

Water movers are cumulative by design. Addition of a water mover will not replace an existing water mover including the default overland flow, groundwater flow, and canal flow water movers. An unlimited number of water movers can be implemented between any two water bodies. As a result, when a concept or structure water mover is added between two water bodies, it does not replace any of the existing water movers, but merely adds to them. To prevent unintended flow from a default water mover between water bodies when a structure is added, a no-flow boundary condition for default water movers must be placed at the location of the structure. Examples where this is appropriate are a weir that controls flow in a canal so that all the water passes through the weir or a road that blocks flow so that all the flow between cells on opposite sides of the road occurs through culverts under the road. The water mover that replaces the default water mover is created by the user. Any number of water movers may be used to simulate flow between water bodies. If the default water mover is not removed, both the water mover and the user created water mover(s) are implemented. The XML input for a junction block that eliminates the default water mover between canal segments 23 and 34 is shown below. This section primarily describes the user-defined water movers. Most of the water movers described here such as culverts and weirs are designed to move water between adjacent water bodies, but several may reasonably be used to move water from any water body to any other water body whether they are adjacent or not. The user must decide the most appropriate water mover for the physical situation being simulated. Some water movers can be user-defined to prevent reverse flow or flow against gravity.

> The following XML input will block flow by the default water mover between canal segments 23 and 34.
>
> ```
> <network_bc>
> <junctionblock id1="23" id2="34"> </junctionblock>
> </network_bc>
> ....
> ```
>
> In the case of a road separating two cells, overland flow will be prevented by the following instructions. Water movers for walls defined by nodes 2-4 and 4-67 will be removed.
>
> ```
>  <mesh_bc>
> .....
>    <noflow section="ol">
>      <nodelist> 2 4 67 </nodelist>
>    </noflow>
> ....
>  </mesh_bc>
>  ....
> ```

Groundwater flow can be prevented through the same walls by replacing ”ol” with ”gw”. Water movers may also be assigned water mover identification numbers that can be used later in specifying water budget output.

6.5.2 Default Water Movers

No XML input is needed to define the default water movers. These are automatically generated by HSE after the 2D grid is input to the model and after the 1D canal network is built.

6.5.3 Concept Water Movers

Concept water movers are versatile, but they must be used carefully because they are very general and are intended to give the user flexibility when the flow characteristics are known, although they are not designed to represent any predefined structure. For example, the general power law water mover may represent a rectangular weir, but may not consider the effects of downstream submergence or end contractions. Table 6.14 lists some available concepts for water movers.

Table 6.14: *List of concept water movers.*

Element	Description
`<delta_control>`	A 1-D lookup table type function of discharge against water head difference
`<doublet>`	A source and a sink couple
`<setflow>`	Controllable user-defined flow
`<dual_control>`	A 2-D lookup table type function with discharge read against upstream and downstream head values.
`<genweir>`	A general weir equation applicable for many weir types
`<hq_relation>`	A simple head - discharge relationship
`<shunt>`	A short circuit between water bodies
`<single_control>`	A lookup table between head and discharge
`<source>`	A pure source or a sink
`<standardweir>`	A common weir type

6.5.3.1 Simple Power Law Based Water Mover `<standardweir>`

The standard weir water mover is created by the `<standardweir>` element. Discharge from water body 1 to water body 2 through a standard weir is expressed as

$$Q_{12} = CL(H_1 - z)^b \tag{6.12}$$

where
C = weir coefficient,
H_1= water level in water body 1,
z = weir crest elevation, and
b = user specified exponent.

A complete specification of the attributes of a standard weir are defined in Table 6.15.

Table 6.15: *Attribute definitions for* `<standardweir>`.

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	Long Integer	200000-300000	256987
`id2`	ID of the downstream water body	Long Integer	200000-300000	268546
`wmID`	ID of the water mover	Long Integer	500000-600000	568546
`fcoeff`	Weir coefficient for flow in the forward direction NOTE: `<bcoeff> is used for both forward and reverse flow`	Real	> 0.0	0.53
`bcoeff`	Weir coefficient for flow in the reverse direction NOTE: `<bcoeff> is used for both forward and reverse flow`	Real	> 0.0	0.53
`length`	Crest length (L)	Real	> 0.0	85.6
`crestelev`	Crest elevation (z) of the weir	Real	> 0.0	26.3
`power`	The exponent (b) in the weir equation	Real	> 0.0	0.6

The following example describes a `<standardweir>` water mover connecting water body 2 to water body 7.

```
<watermovers>
....
<standardweir
 id1 = "2"  id2 = "7"  wmID = "201"
 fcoeff = "0.2" bcoeff = "0.1" length = "20.0" crestelev = "501.0"
 power = "2.3">
</standardweir>
...
<watermovers>
```

6.5.3.2 General Power Law Based Water Mover `<genweir>`

The `<genweir>` water mover uses a power equation of the form

$$Q_{12} = CL(H_1 - z)^a (H_1 - H_2)^b \tag{6.13}$$

where C = user specified weir coefficient,
H_1 and H_2= water levels in water bodies 1 and 2,
z = weir crest elevation,
a = user specified coefficient, and
b = user specified exponent.

The user may decide to use this equation for certain types of weirs or bridges. A V-notch weir flow is calculated as

$$Q = \frac{8}{15} K \sqrt{2g} \tan(\theta/2)(H - Z)^{2.5} \tag{6.14}$$

may be simulated by assigning appropriate values of the variables. Complete specification of the attributes of a standard weir are defined in Table 6.16.
The following example describes a `<genweir>` water mover connecting water body 5 to water body 11.

```
<watermovers>
....
   <genweir id1 = "5" id2 = "11" fcoeff = "1.0" bcoeff="0.0"
      crestelev = "500.5" crestlen = "100" dpower = "1.5" spower = "0.5" >
   </genweir>
....
</watermovers>
```

Table 6.16: *Attribute definitions for* `<genweir>`*.*

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	Long Integer	200000-300000	256987
`id2`	ID of the upstream water body	Long Integer	200000-300000	268546
`wmID`	ID of the water mover	Long Integer	500000-600000	568546
`fcoeff`	Weir coefficient used for forward (1→2) flow.	Real	> 0.0	0.38
`bcoeff`	Weir coefficient used for reverse (2→1) flow	Real	> 0.0	0.46
`crestelev`	Crest elevation of the weir	Real	> 0.0	23.8
`crestlen`	Weir crest length	Real	> 0.0	58.3
`dpower`	Exponent a for upstream depth over the weir	Real	$>= 0.0$	1.4
`spower`	Exponent b for water level difference	Real	$>= 0.0$	0.56

6.5.3.3 Coupled Source Sink Water Mover `<doublet>`

The `<doublet>` water mover is a coupled source and a sink of equal magnitudes. Water may be removed from any water body and discharged into any other. Internally, this is treated the same as a boundary condition. The water is removed from one water body and put into a boundary condition reservoir. The same amount of water is removed from the boundary condition reservoir and discharged into the second water body. This is applied if a constant or time series discharge from one water body to another is to be implemented. A complete specification of the attributes of a doublet water mover are defined in Table 6.17

The following sample XML input implements a constant flow from water body 5 to water body 11 and a time series of flows in DSS format from water body 6 to waterbody 31.

```
 <watermovers>
 ....
    <doublet id1 = "5" id2 = "11"  label = "whatever">
      <const value = "-100.5" dbintl = "300"> </const>
    </doublet>
    <doublet id1 = "6" id2 = "31"  label = "S-205">
      <dss file ="S-201/dss" pn="/hse/t3x3 P02/FLOW //300min/CALC/">
      </dss>
    </doublet>
......
 </watermovers>
```

6.5.3.4 Controllable User-Defined Flow `<setflow>`

The setflow watermover `<setflow>` is identical to the doublet watermover except that a controller may be applied. It applies a user-defined constant flow or time series flow from one waterbody to another. A controller may be applied to modulate the flow of the watermover.

Complete specifications of the attributes of a `<setflow>` water mover are defined in Table 6.18 The following XML implements a controllable constant flow from water body 16 to water body 32 through water mover and a controllable time series of flow from water body 64 to water body 128 through water mover 124.

```
    <watermovers>
    ....
       <setflow wmID="123" id1 = "16" id2 = "32"  label = "transfer">
         <const value = "64" dbintl = "15"> </const>
       </setflow>
       <setflow wmID=’’124’’ id1 = "64" id2 = "128"  label = "transfer">
         <dss file ="S-201/dss" pn="/hse/t3x3 P02/FLOW //300min/CALC/"> </dss>
       </setflow>
   ......
    </watermovers>
```

Table 6.17: *Attribute definitions for* `<doublet>`.

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	Long Integer	200000-300000	256987
`id2`	ID of the upstream water body	Long Integer	200000-300000	268546
`wmID`	ID of the water mover	Long Integer	500000-600000	568546
`label`	An optional label for the doublet	string	any string	Pump 416
`<const>`	A constant flow rate will be specified			
`<rc>`	A flow rate defined by a rule curve is provided.			
`<dss>`	A time series in DSS format is provided.			
`<asciiform>`	A time series in asciiform format is provided.			
`<csv>`	A time series in csv format is provided.			
Details on specifying input data in `<const>`, `<rc>`, `<dss>`, `<asciiform>`, and `<csv>` format are provided in Chapter 9				

Table 6.18: *Attribute definitions for* `<setflow>`.

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<id1>`	ID of the upstream water body	Long Integer	200000-300000	256987
`<id2>`	ID of the upstream water body	Long Integer	200000-300000	268546
`<wmID>`	ID of the water mover	Long Integer	500000-600000	568546
`label`	An optional label for the water mover	string	any string	Drain 345
`<const>`	A constant flow rate will be specified			
`<rc>`	A flow rate defined by a rule curve is provided.			
`<dss>`	A time series in DSS format is provided.			
`<asciiform>`	A time series in asciiform format is provided.			
`<csv>`	A time series in csv format is provided.			
Details on specifying input data in `<const>`, `<rc>`, `<dss>`, `<asciiform>`, and `<csv>` format are provided in Chapter 9				

6.5.3.5 Lookup Table Based Water Movers

Three water movers are lookup table types that are useful if the discharge from one water to another depends on the value of the head or heads in one or more water bodies. The lookup table types that can be used are `<single_control>`, `<dual_control>`and `<delta_control>` as described below.

6.5.3.6 Single Control Water Movers `<single_control>`

The `<single_control>` water mover uses a 1-D lookup table to determine the discharge from any water body to any other water body. The flow, Q, is determined from a user specified one dimensional rating curve or lookup table with the head in any specified water body being the independent variable. The first column of the table is the value of the independent variable and the second column is the corresponding discharge. The upstream head is the most commonly used control point although the head in any water body may be used. Complete specifications of the attributes of a `<singlecontrol>` water mover are defined in Table 6.19

An example of <`single_control`> is shown below. The example shows how a single control water mover can be used to trigger the operation of a pump as the water level at water body 10034 reaches 5.8-5.9 m. The flow is from a pump that has the pumping characteristics shown in the lookup table. The pump starts pumping when the head in water body 10034 reaches 5.0 m and reaches a maximum pumping rate of 31.545 m^3/sec at a head of 6.1 m.

```
<watermovers>
   <single_control id1="10034" id2="354" wmID = "35" control="10034" cutoff="3.0"
       gravflow = "no" revflow = "yes"  label="Pump at the impoundment" >
               3.0   0.0
               5.0   0.0
               5.8   6.309
               5.9  25.236
               6.1  31.545
               30.0  31.545
   </single_control >
</watermovers>
```

6.5.3.7 Dual Control Water Movers `<dual_control>`

The `<dual_control>` water mover uses a 2D lookup table to determine the discharge from any water body to any other water body. The flow, Q, is determined from a user specified two-dimensional rating curve or lookup table with the heads in the specified water bodies being the independent variables. The first column of the table is the value of the head in the first water body, the first row is the head in the second water body and the remaining

Table 6.19: *Attribute definitions for the* `<single_control>` *Water Mover.*

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<id1>`	ID of the upstream water body	Long Integer	200000-300000	256987
`<id2>`	ID of the upstream water body	Long Integer	200000-300000	268546
`<wmID>`	Water Mover ID	Long Integer	200000-300000	268546
`<control>`	ID of the control water body.	Long Integer	200000-300000	568546
`<cutoff>`	Head in the control water body below which no flow occurs.	Real	>= 0.0	15.36
`<gravflow>`	Specifies whether only gravity flow can occur (flow can occur only from upstream to downstream).	String	”yes” or ”no”	”yes”
`<revflow>`	”yes” allows reverse flow	String	”yes” or ”no”	”yes”
`<label>`	Optional label for the water mover.	String	Any String	”Irrigation Pump”

rows and columns are the corresponding discharges. The water bodies need not be adjacent although they often will be.

The attributes of a `<dual_control>` water mover are explained in Table 6.20

An example of the input for a `<dual_control>` water mover is shown below. In the example, cell 5 is the upstream water body, and segment 21 is the downstream water body. Columns 2-4 represent various watermover flow values corresponding to the downstream head specified in row 1 (segment 21), and the upstream head of column 1 (cell 5). Only gravity flow can occur and reverse flow is not allowed.

```
<watermovers>
  <dual_control id1="5" id2="21" cutoff = "1.0" gravflow = "yes"
    revflow = "no" label="Flooding Overflow">
            495  500  505  510
      495.0   0 1000 2000 3000
      500.0   0    0 1500 2500
      505.0   0    0    0 2000
      510.0   0    0    0    0
  </dual_control>
</watermovers>
```

6.5.3.8 Delta Control Water Movers `<delta_control>`

The `<delta_control>` water mover uses a 1-D lookup table to determine the discharge from any water body to any other water body. The flow, Q, is determined from a user specified one dimensional rating curve or lookup table with the independent variable being the difference between the upstream head and the downstream head. The first column of the table is the value of the independent variable and the second column is the corresponding discharge.

The attributes of a `<delta_control>` water mover are explained in Table 6.21

Flow through a delta control structure is from water body id1 to id2 if the head difference H_1 - H_2 is positive and from id2 to id1 if it is negative. An example of input for a delta control lookup table is shown below. Flow is in either direction between water body 5 and water body 15. The flow increases rapidly as the head difference increases from 4 to 6 meters.

Table 6.20: *Attribute definitions for the* `<dual_control>` *Water Mover.*

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<id1>`	ID of the upstream water body	Long Integer	200000-300000	256987
`<id2>`	ID of the upstream water body	Long Integer	200000-300000	268546
`<wmID>`	Water Mover ID	Long Integer	200000-300000	268546
`<control>`	ID of the control water body (Not Implemented).	Long Integer	200000-300000	568546
`<cutoff>`	Head in the upstream water body below which no flow occurs.	Real	≥ 0.0	15.36
`<gravflow>`	Specifies whether only gravity flow can occur (flow can occur only from upstream to downstream).	String	”yes” or ”no”	”yes”
`<revflow>`	”yes” allows reverse flow	String	”yes” or ”no”	”yes”
`<label>`	Optional label for the water mover.	String	Any String	”Weir 1A”

Table 6.21: *Attribute definitions for the* `<delta_control>` *Water Mover.*

Attribute	Definition	Variable type	Suggested range	Example
`<id1>`	ID of the upstream water body	Long Integer	200000-300000	256987
`<id2>`	ID of the upstream water body	Long Integer	200000-300000	268546
`<wmID>`	Water Mover ID	Long Integer	200000-300000	268546
`<label>`	Optional label for the water mover.	String	Any String	"Flood Control Pump"

```
<watermovers>
    <delta_control id1="5" id2 ="15" label="t3x3-2">
        4.0 0
        5.0 10
        6.0 10000
    </delta_control>
</watermovers>
```

6.5.3.9 Comments On The Use Of Lookup Tables

Since a large variety of flow discharge relationships can be described using lookup tables, it is important to follow a number of simple rules to prevent the model from generating erroneous results.

1. It is not necessary to provide smooth stage-discharge relationships. However, smooth relationships give more accurate results because the slope estimates of the curve become more accurate when the flow relationship is linearized.
2. When water levels encountered are outside the bounds described in the lookup table, linear extrapolation is used to compute discharge. However, to prevent the discharges becoming excessively large or small during extrapolation, it is recommended that the user define a rating curve over a broad range of heads to prevent unrealistic flows being imposed.
3. A 1-D lookup table needs a minimum of two points and a two-dimensional lookup table needs a minimum of four points.

Table 6.22: *Attribute definitions for a shunt Water Mover.*

Attribute	Definition	Variable type	Suggested range	Example
`id1`	ID of the upstream water body	Long Integer	200000-300000	256987
`id2`	ID of the downstream water body	Long Integer	200000-300000	268546
`sconst`	Conveyance of the shunt (K in the flow equation)	Real	≥ 0.0	4.7
`bottom`	Head in water body 1 below which no flow occurs.	Real	≥ 0.0	11.64

6.5.3.10 Shunt Watermover `<shunt>`

The shunt water mover, `<shunt>`, is used when two water bodies are connected, and there is no obvious choice except for a shunt. A shunt can be used if a canal ends up in a reservoir with no structure to separate them. The flow, Q, is

$$Q_{12} = K(H_1 - H_2) \tag{6.15}$$

where K is a user specified constant and H_1 and H_2 are the heads in the two water bodies. Complete specifications of the attributes of a `<shunt>` water mover are defined in Table 6.22.

The XML input below creates a water mover with ID 9 that moves water between water bodies 10134 and 20001 according to the shunt equation. Flow may occur in either direction.

```
<watermovers> ...
 <shunt wmID = "9" id1 = "10134" id2 = "20001"
 sconst = "0.1" bottom = "128.2"> </shunt>
</watermovers>
```

6.5.4 New And Borrowed Physical Structure Types

A number of structure types have been added from other models such as the Multi Basin Routing Model (MBR) that has evolved into the CASCADE model. Others structure types

Table 6.23: *List of physical water movers.*

Element	Description
`<culvert>`	Culvert water mover borrowed from National Weather Service (NWS). The code was written for the FLDWAV model, but was never thoroughly tested and incorporated into FLDWAV
`<gateweir>`	Gated weir borrowed from the NWS FLDWAV manual
`<spill>`	Spillway water mover borrowed from the NWM FLDWAV Manual
`<pipe>`	Pipe water mover borrowed from MBR
`<mbrbroadweir>`	Broad crested weir borrowed from MBR
`<mbrsharpweir>`	Sharp crested weir borrowed from MBR
`<mbrdropweir>`	Drop weir borrowed from MBR
`<yarnell>`	Bridge routine based on Yarnell equations derived from the HECRAS Technical Reference Manual
`<vnotchbleeder>`	V-notch bleeder borrowed from MBR
`<circularbleeder>`	Circular bleeder borrowed from MBR
`<rectbleeder>`	Rectangular bleeder borrowed from MBR
`<hydropower>`	Hydropower plants - from general properties of hydropower.

were added to the HSE model by coding algorithms for equations that were published in model users manuals and technical guidance documents. These structures can be applied between any two water bodies. Downstream submergence conditions are considered for some of these structures. When a structure is added, a no-flow condition has to be applied for the specific wall if the water bodies are adjacent (the normal case). These water movers and the sources of the algorithms used to compute flow are listed in Table 6.23.

”Borrowed” water movers have been included in the code to provide flexibility for modelers although the approaches and algorithms have not been extensively tested and verified by OoM staff. Water movers that fall into this category will be identified in the description of each and the modeler will be advised to use them only with a clear understanding of the underlying algorithms.

6.5.5 Culvert Water Mover `<culvert>`

This water mover has not been thoroughly tested or verified by the SFWMD Office of Modeling (OoM) and is not used in the SFRSM.

The culvert water mover was borrowed from the NWS. While it was written for the FLDWAV model it was never thoroughly tested and was not incorporated into the FLDWAV manual (Danny Fread, Ming Jin, Personal Communication). Figure 6.3 shows a definition sketch of a circular culvert. It uses six different flow types as is commonly done for culvert analysis. These are listed below along with the equations to compute flow. Table 6.24 shows the attributes of a culvert.

1. If the outlet is submerged, discharge is computed in English units as

$$Q = CA\sqrt{\frac{2g(Y1 - Y2)}{1 + \frac{29.1C^2n^2L}{R^{4/3}}}} \tag{6.16}$$

   where
   C = a user specified discharge coefficient,
   A = cross sectional area of the culvert,
   R = hydraulic radius,
   Y1 = upstream head,
   Y2 = downstream head,
   n = Manning's roughness coefficient,
   L = culvert length,
   g = acceleration due to gravity.

2. Upstream depth ≥ 1.3 culvert diameter (or height for a rectangular culvert) and length <= 20∗diameter, the culvert is hydraulically short and discharge is inlet controlled.

$$Q = 0.7CA\sqrt{2g(H1 - 0.6D)} \tag{6.17}$$

   A = cross sectional area of the culvert
   H1 = upstream depth
   D = culvert diameter (height for a rectangular culvert).

3. Upstream depth ≥ 1.3 culvert diameter (or height for a rectangular culvert) and length > 20∗diameter, the culvert is hydraulically long and discharge is outlet controlled.

$$Q = 0.7CA\sqrt{\frac{2g(Y1 - max(Y2, CH2 + 0.5D))}{1 + \frac{29.1C^2n^2L}{R^{4/3}}}} \tag{6.18}$$

where CH2 = culvert invert at the downstream end.

4. If upstream depth < 1.3D and the culvert has a mild slope

$$Q = CA\sqrt{2g * (Y1 - CH2)} \tag{6.19}$$

where

$$A = W\frac{Y1 - CH2}{1.25 + \frac{0.5}{C^2}} \tag{6.20}$$

for a rectangular culvert and

$$A = [0.785Y - 0.045\sin(2\pi Y)]D^2 \tag{6.21}$$

for a circular culvert.
W = width of a rectangular culvert

$$Y = \frac{DC}{D} \tag{6.22}$$

$$DC = \frac{Y1 - CH2}{1.25 + \frac{0.425}{C^2}} \tag{6.23}$$

with DC = D if DC > D and D = diameter of a circular culvert.

5. If upstream depth < 1.3D and the culvert has a steep slope with the downstream water depth less than critical depth.

$$Q = CA\sqrt{2g * (Y1 - CH1)} \tag{6.24}$$

where

$$A = W\frac{Y1 - CH1}{1.05 + \frac{0.5}{C^2}} \tag{6.25}$$

for a rectangular culvert and

$$A = [0.785Y - 0.045\sin(2\pi Y)]D^2 \tag{6.26}$$

for a circular culvert.
In these equations, W = width of a rectangular culvert,

$$Y = \frac{DC}{D} \tag{6.27}$$

$$DC = \frac{Y1 - CH1}{1.05 + \frac{0.425}{C^2}} \tag{6.28}$$

with $DC = D$ if $DC > D$.

6. If upstream depth < 1.3D and the culvert has a steep slope with the downstream water depth greater than or equal to critical depth.

$$Q = CA\sqrt{2g * (Y1 - Y2)} \tag{6.29}$$

where

$$A = WH2 \tag{6.30}$$

for a rectangular culvert

$$A = [0.785Y - 0.045\sin(2\pi Y)]D^2 \tag{6.31}$$

for a circular culvert

where W = width of a rectangular culvert and

$$Y = \frac{DC}{D}, \tag{6.32}$$

$$DC = Y2 - Y1 \tag{6.33}$$

and DC = D if DC > D.

An example XML input for a circular culvert is shown below. A culvert labeled water mover 231 connects water body 5 with water body 21. Since "width" = 0 this is a circular culvert with diameter 3.2 meters and a length of 40 meters. It is probably concrete (Manning's n = 0.012) and the discharge coefficient is 0.4. The culvert is horizontal with the same invert at both ends. An overland flow block should be created between water bodies 5 and 21 if the culvert carries all the flow.

```
 <watermovers>
 ....
 <culvert id1 = "5" id2="21" wmID = "231" width = "0" height = "3.2" length = "40"
     mann = "0.012" coeff = "0.4" hw_inv = "0.91" tw_inv = "0.91" rev = "Y" >
 </culvert>
..
 </watermovers>
```

6.5.5.1 MBR Pipe Flow `<pipe>`

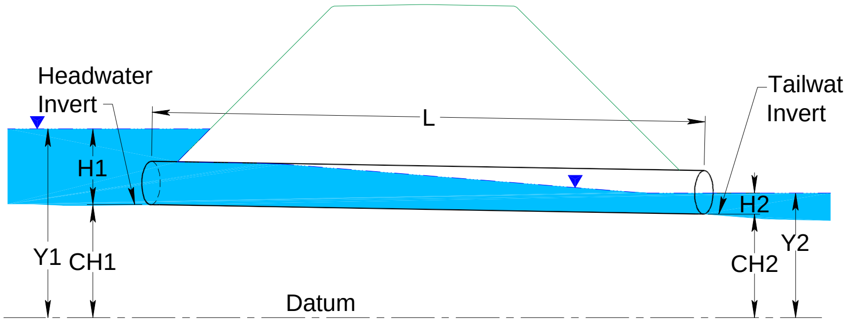
The `<pipe>` water mover has not been extensively tested or verified by the OoM staff. It should be used by experienced users of the MBR or CASCADE model with a full understanding of the algorithms used to compute flow.

Table 6.24: *Attributes of the* `<culvert>` *water mover.*

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmid`	ID of the water mover being created	long integer	200000-300000	256987
`width`	Width of a box culvert. Set=0 for a circular culvert	real	≥ 0.0	2.5
`height`	Height of a box culvert or diameter of a circular culvert	real	≥ 0.0	4.25
`length`	Length of the culvert barrel	real	≥ 0.0	110.6
`mann`	Manning's roughness coefficient	real	$0.0 - 0.2$	0.024
`coeff`	Discharge coefficient for the culvert	real	$0.39 - 0.98$	0.42
`hw_inv`	Elevation of the upstream culvert invert	real	≥ 0.0	13.86
`tw_inv`	Elevation of the downstream culvert invert	real	≥ 0.0	10.48
`rev`	Specifies whether reverse flow is allowed. N prevents reverse flow. Y is default.	String	N or Y	Y

This feature borrowed from MBR is not as robust as `<culvert>`. Depending on various upstream and downstream conditions, this too uses six flow regimes. Figure 6.3 shows a definition sketch for pipe flow. Table 6.25 shows the XML attributes used to define the pipe water mover.

Figure 6.3: *Definition sketch of a pipe.*



Following is an example of a data set for a pipe.

```
 <watermovers>
 ....
 <pipe id1 = "5" id2="21" diameter = "3.71"
       length = "212.0" mann = "0.012"
       hw_inv = "0.91" tw_inv = "0.91" rev = "Y">
 </pipe>
..
 </watermovers>
```

The pipe in this example connects the water body 5 to water body 21 with a pipe of diameter 3.71 m with reverse flow allowed.

6.5.5.2 MBR Broad Weir `<mbrbroadweir>`

The broad crested weir water mover equations have not been thoroughly tested and verified by OoM staff. The broad crested weir shown in Figure 6.4 is borrowed from the MBR model with discharge equation

$$Q = C_d L (H1 - z)^{1.5} \tag{6.34}$$

The discharge is modified with a tailwater correction CS so that $Q = Q * CS$

If $TW \geq 0.99H$ then $Q = 0$

Table 6.25: *Attributes used to define a* `<pipe>` *water mover.*

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmid`	ID of the water mover being created	long integer	200000-300000	256987
`diameter`	Diameter of a circular pipe	real	≥ 0.0	4.25
`length`	Length of the pipe	real	≥ 0.0	110.6
`mann`	Manning's roughness coefficient	real	$0.0 - 0.2$	0.024
`hw_inv`	Elevation of the upstream culvert invert	real	≥ 0.0	13.86
`tw_inv`	Elevation of the downstream culvert invert	real	≥ 0.0	10.48
`rev`	Specifies whether reverse flow is allowed. N prevents reverse flow. Y is default.	String	N or Y	Y

If $0.95H \leq TW < 0.99H$

$$CS = 0.965 - \exp(\frac{100(TW/H - 1)}{4} - 0.43) \tag{6.35}$$

If $0.76H \leq TW < 0.95H$

$$CS = 1.0 - \exp(\frac{100(TW/H - 1)}{4} - 0.2) \tag{6.36}$$

If $TW < 0.76H$, $CS = 1.0$

in which, L = weir length; z = crest elevation, TW = H2 - z, and H = H1 - z. The value of C_d should be in the range 3.05-3.10. The attributes needed to define a broad crested weir water mover are explained in Table 6.26

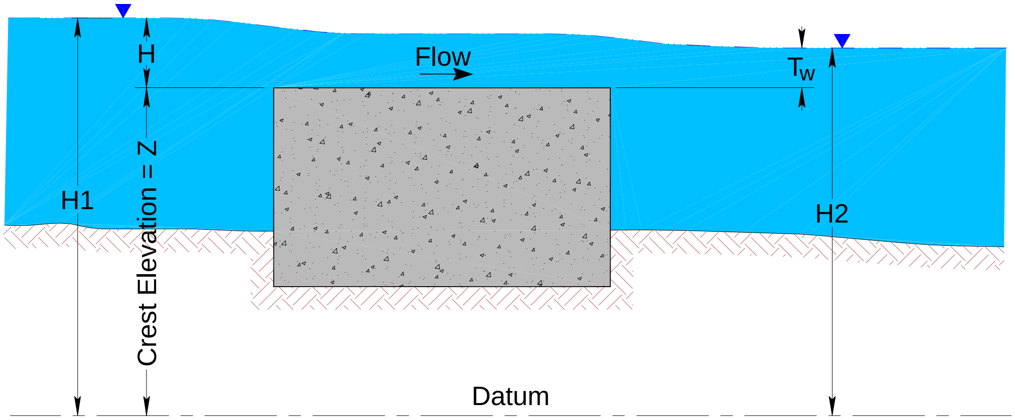


Figure 6.4: *Definition sketch of broad crested weir.*

The following example describes a broadcrested weir that moves water from water body 5 to water body 21 with a discharge coefficient of 2.9. The weir length is 100 meters and the crest elevation = 5.2 meters.

```
 <watermovers>
 ....
 <mbrbroadweir id1   = "5" id2="21"
    crestelev = "5.2" length = "100.0" coeff = "2.9">
 </mbrbroadweir>
..........
 </watermovers>
```

Table 6.26: *Attributes of a broad crested weir,* `<mbrbroadweir>`.

Attribute	Definition	Variable type	Suggested range	Example
`<id1>`	ID of the upstream water body	long integer	100000-200000	153675
`<id2>`	ID of the downstream water body	long integer	100000-200000	148322
`<wmid>`	ID of the water mover being created	long integer	200000-300000	256987
`<crestelev>`	Elevation of the weir crest	real	≥ 0.0	14.78
`<length>`	Length of the weir across the channel	real	≥ 0.0	116.8
`<coeff>`	Discharge coefficient	real	≥ 0.0	0.48

6.5.5.3 MBR Sharp Weir `<mbrsharpweir>`

The sharp crested weir is also borrowed from the MBR model. Figure 6.5 shows a definition sketch of a sharp crested weir.

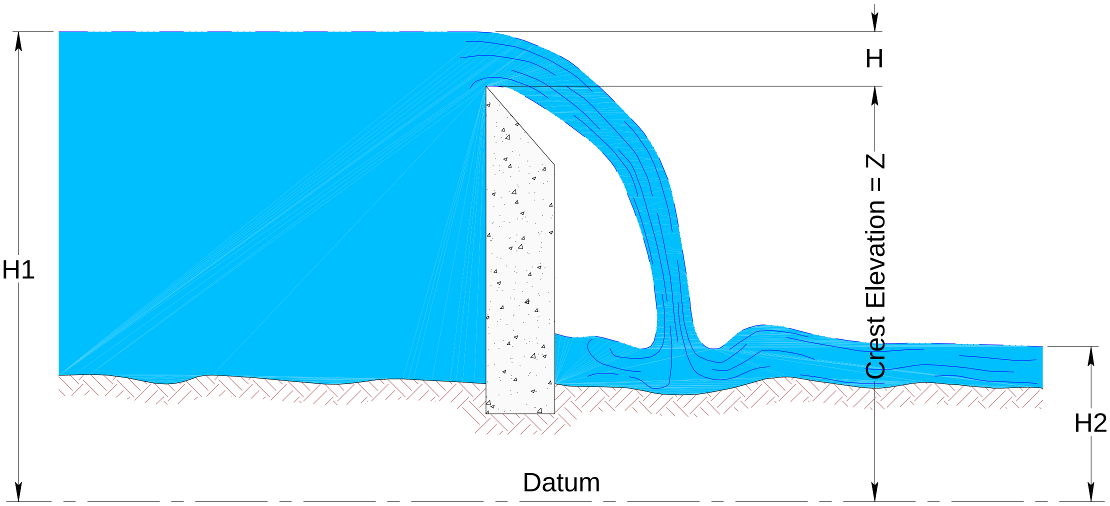
Discharge is computed as

$$Q = 3.13(CS)L(H - z)^{1.5} \tag{6.37}$$

where
L = weir length,
CS = Tailwater correction,
H = Head over weir crest = H1 - z,
z = Elevation of the weir crest, and
CS = 1.0 unless TW = H2 - z > 0 and then,

$$CS = (1 - (\frac{TW}{H})^{1.5})^{0.385} \tag{6.38}$$

Figure 6.5: *Definition sketch of a sharp crested weir.*



according to (Brater et al., 1996).

The following example creates a sharp crested weir that carries water from water body 7 to water body 20.

```
 <watermovers>
 ....
 <mbrsharp id1 = "7" id2="20"
   crestelev = "2.3" length = "80.0">
 </mbrsharp>
........
 </watermovers>
```

6.5.5.4 MBR Drop Weir `<mbrdropweir>`

The drop weir (Figure 6.6)is taken from the United States Bureau of Reclamation (Bureau of Reclamation, 1977). The physical structure is a "morning glory" spillway. A vertical pipe with water flowing over the lip. In this model, it is assumed that the control is at the entrance to the pipe and that free flow exists below the entrance. Discharge is computed as

$$Q = CLH^{1.5} \tag{6.39}$$

Table 6.27: *Attributes of a sharp crested weir,* `<mbrsharpweir>`.

Attribute	Definition	Variable type	Suggested range	Example
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmid`	ID of the water mover being created	long integer	200000-300000	256987
`crestelev`	Elevation of the weir crest	real	≥ 0.0	12.93
`length`	Length of the weir across the channel	real	≥ 0.0	63.7

where C depends on the ratio of Head to pipe diameter.

$$R = \frac{H}{D} \tag{6.40}$$

If $R > 2$, $C = 1$
If $R < 0.3$, $C = 4$.
Otherwise

$$C = 4.01 + 0.72R - 6.12R^2 + 4.37R^3 - 0.93R^4 \tag{6.41}$$

The equation for computing C was obtained by fitting data from a nomograph in the USBR publication "Design of Small Dams" with a fourth order polynomial.

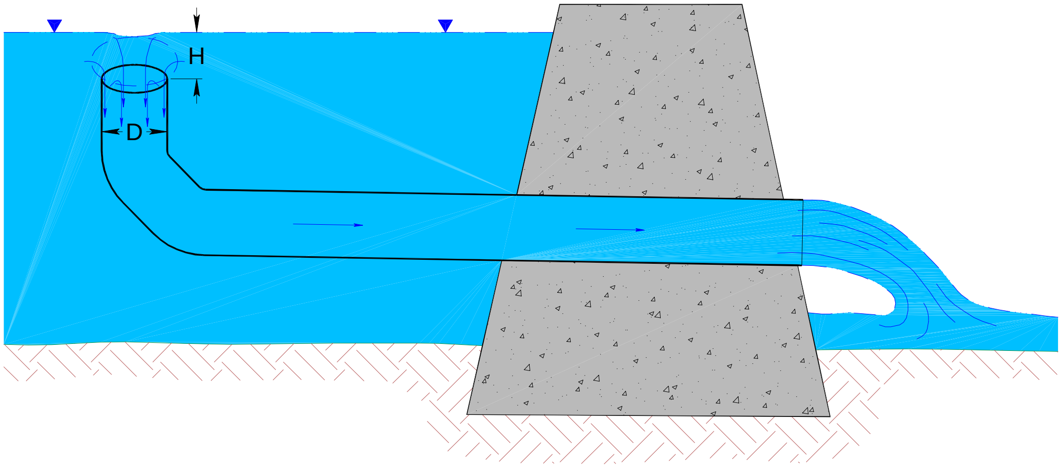


Figure 6.6: *Definition sketch of a drop weir.*

In the following example water spills from water body 7 into a vertical circular pipe of diameter 4.0 meters and top at elevation 2.3 meters, and is discharged to water body 20.

```
 <watermovers>
 ....
 <mbrdropweir id1   = "7" id2="20"
   crestelev = "2.3" length = "4.0">
 </mbrdropweir>
........
 </watermovers>
```

6.5.5.5 NWS Uncontrolled Spill `<spill>`

This is a spillway routine borrowed from the NWS FLDWAV model (Figure 6.7). Only downstream flow is allowed. Discharge is computed as

$$Q = CS * CTW * L * \sqrt{2g} * H^{1.5} \tag{6.42}$$

where
CS is a user specified coefficient, CTW is a tailwater correction,
H = H1 - Crest Elevation,
TW = H2 - Crest Elevation,
$CTW = 1.0$ unless $TW/H > 0.67$. Then

Table 6.28: *Attributes of a drop weir.*

Attribute	Definition	Variable type	Suggested range	Example
`<id1>`	ID of the upstream water body	long integer	100000-200000	153675
`<id2>`	ID of the downstream water body	long integer	100000-200000	148322
`<wmid>`	ID of the water mover being created	long integer	200000-300000	256987
`<crestelev>`	Elevation of the weir crest	real	≥ 0.0	23.6
`<length>`	Length of the weir (Circumference of the pipe)	real	≥ 0.0	76.3

$$CTW = 27.8(\frac{TW}{H} - 0.67)^3 \tag{6.43}$$

Attributes of the uncontrolled spillway are listed in Table 6.29

In the following example water flows from water body 2381 over a spillway to water body 4216. The crest elevation is 23.6 meters and the length of the spillway is 76.3 meters.

```
 <watermovers>
 ....
 <spill id1   = "2381" id2="4216"
   crest = "23.6" width = "76.3" c15="0.42">
 </spill>
........
 </watermovers>
```

6.5.5.6 NWS Gated Weir `<gateweir>`

This water mover coding was based on an earlier version of the FLDWAV manual and has not been thoroughly tested and verified by OoM staff. Modelers should study the equations and use it if they simulate the structure being modeled.

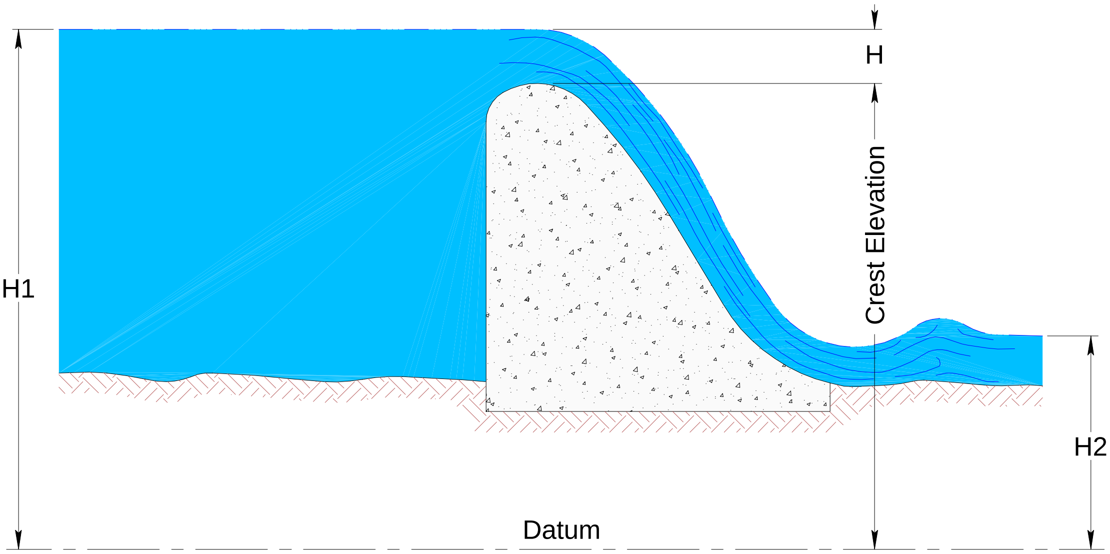


Figure 6.7: *Definition sketch of an uncontrolled spillway.*

The gated weir routine was borrowed from the NWS FLDWAV model. The gated weir is a weir with a gate that can be lowered to control the flow (Figure 6.8). As the gate is lowered, weir flow becomes orifice flow when the gate impinges on the water surface. The flow is divided into two flow regimes, rectangular weir flow when the upstream water surface elevation is below or at the bottom of the gate and orifice flow when the water surface is above the bottom of the gate. The flow equations for weir flow are

$$Q = C(CS)WH_u^{\frac{3}{2}} \tag{6.44}$$

where

C = user specified coefficient,
CS = tailwater correction,
W = width of the weir,
z = elevation of the weir crest,
$H_u = H1 - z$ = upstream head on the weir and

$$Q = C * CS * W * H_G\sqrt{2gH_u} \tag{6.45}$$

where for orifice flow
H_G = Gate opening above the weir crest

Table 6.29: *Attributes of an uncontrolled spillway* `<spill>`.

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`label`	Label for the spillway	string	Any string	Fontana Dam
`c15`	Discharge coefficient	real	0.35 to 0.45	0.42
`crest`	Elevation of the weir crest	real	≥ 0.0	16.7
`width`	Length of the weir (Circumference of the pipe)	real	≥ 0.0	18.7

The tailwater correction for both weir flow and orifice flow is

$$CS = 1.0 - 27.83 \left[\frac{H_u}{H_d} - 0.67 \right]^3 \tag{6.46}$$

for $H_d/H_u \geq 0.67$ $CS = 1.0$ otherwise.

and H_d = downstream head on the weir, $H2 - z$.

The attributes of the gated weir watermover are detailed in Table 6.30.

In the following example water flows from water body 3156 through a gated weir to water body 1358. The crest elevation is 14.5 meters and the gate opening is 4.6 meters.

```
<watermovers>
 ....
 <gateweir id1   = "3156" id2="1358"
   crest = "14.5" width = "4.6" c05="0.4" c15="0.62" gopen="3.2">
 </spill>
........
 </watermovers>
```

Table 6.30: *Attributes of* `<gateweir>`.

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmid`	ID of the water mover being created	long integer	200000-300000	256987
`co5`	Discharge coefficient for weir flow	Real	0.35 -0.45	0.42
`c15`	Discharge coefficient for orifice flow	Real	0.65 - 0.75	0.67
`crest`	Elevation of the weir crest.	real	≥ 0.0	13.2
`width`	Length of the weir.	real	≥ 0.0	32.6
`gopen`	Gate opening (height of the opening in orifice flow)	real	≥ 0.0	5.3

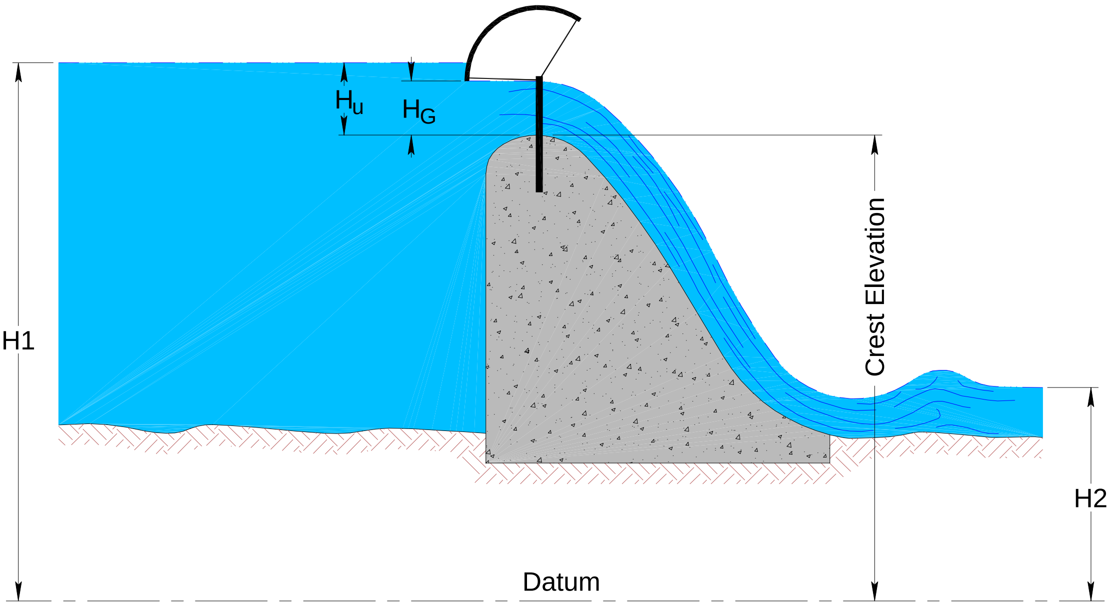


Figure 6.8: *Definition sketch of a gated weir.*

6.5.6 Bleeders

Bleeders are designed to allow small discharges to pass through structures, and may provide control over the rate at which the water level recedes from small basins. Three bleeder configurations have been borrowed from the MBR model; v-notch, circular, and rectangular. All the bleeders consider some downstream submergence effects. The three types of bleeders are shown in Figure 6.9.

6.5.6.1 V-Notch Bleeder `<vnotchbleeder>`

Discharge is computed as weir or orifice flow depending on whether the upstream head is above the top of the bleeder. For orifice flow (Upstream head above the top of the bleeder)

$$Q = 0.6A\sqrt{2g(H_u - H_{centroid})} \tag{6.47}$$

for $H_d \leq H_{centroid}$ and

$$Q = 0.6A\sqrt{2g(H_u - H_d)} \tag{6.48}$$

for $H_d > H_{centroid}$

where
H_u = upstream head,
H_d = downstream head,
$H_{centroid}$ = elevation of the bleeder centroid, and
A = the area of the bleeder.

For weir flow

$$Q = 2.5 * CS * \tan(\theta/2) H_u^{2.5} \tag{6.49}$$

with the tailwater correction CS = 1.0 for H_d below the bottom of the bleeder and

$$CS = \left[1 - \left(\frac{H_d - H_{inv}}{H_u - H_{inv}}\right)^{2.5}\right]^{0.385} \tag{6.50}$$

for $H_d > H_{inv}$ where
H_{inv} = the elevation of the lowest point of the bleeder.

Attributes of `<vnotchbleeder>` are listed in Table 6.31.

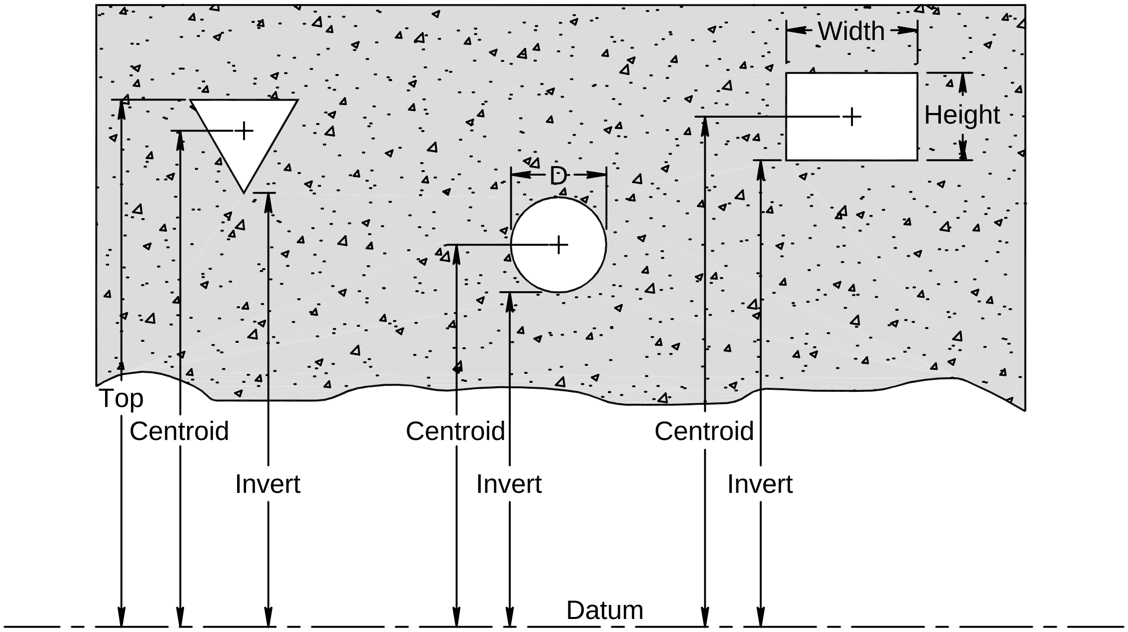


Figure 6.9: *Definition sketch of bleeders.*

Table 6.31: *Attributes of* `<vnotchbleeder>`.

Attribute	Definition	Variable type	Suggested range	Example
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmid`	ID of the water mover being created	long integer	200000-300000	256987
`invert`	Elevation of the lowest point of the bleeder opening	Real	≥ 0.0	13.24
`top`	Elevation of the highest point of the bleeder opening	Real	≥ 0.0	15.23
`angle`	Angle of the V at the invert in degrees	real	$0 - 180$	60.0

```
The following is the XML code for a v-notch bleeder
that can move water between water bodies 3 and 5.

 <watermovers>
 ....
  <vnotchbleeder id1="3" id2 = "5" invert = "5.6" top = "6.0" angle = "30">
  </vnotchbleeder>
..........
 </watermovers>
```

6.5.6.2 Circular Bleeder `<circularbleeder>`

This water mover has not been thoroughly tested and verified by the OoM.

A circular bleeder is a circular opening in a wall or weir. Flow can be weir flow or orifice flow depending on whether the upstream head is above the top of the bleeder. For orifice flow discharge is

$$Q = 0.6A\sqrt{2g(H_u - H_{centroid})} \tag{6.51}$$

for $H_d \leq H_{centroid}$ and

$$Q = 0.6A\sqrt{2g(H_u - H_d)} \tag{6.52}$$

for $H_d > H_{centroid}$

Weir flow is

$$Q = 0.6A\sqrt{\frac{2g(H_u - H_{invert})}{2}} \tag{6.53}$$

where H_{invert} = elevation of the bottom of the bleeder and A is the flow area based on the upstream head. If $H_d > H_{invert}$ then

$$Q = 0.6A\sqrt{2g(H_u - H_d)} \tag{6.54}$$

The attributes of `<circularbleeder>` are defined in Table 6.32.

The following XML input will create a circular bleeder with diameter = 0.2 m. and centroid at an elevation of 7.6 meters to move water between water bodies 3 and 5.

```
 <watermovers>
 ....
  <circularbleeder id1="3" id2 = "5" wmID="210"
      centroid = "7.6" diameter="0.2">
  </circularbleeder>
..........
 </watermovers>
```

6.5.6.3 Rectangular Bleeder `<rectbleeder>`

A rectangular bleeder is a rectangular opening in a wall or weir. Discharge may be either orifice or weir flow. The attributes of `<rectbleeder>` are defined in Table 6.33. Discharge is computed as

$$Q = 0.6A\sqrt{2g(H_u - H_{centroid})} \tag{6.55}$$

for orifice flow (the upstream head, H_u > top of the bleeder) and

Table 6.32: *Attribute definitions for* `<circularbleeder>`.

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmid`	ID of the water mover being created	long integer	200000-300000	256987
`centroid`	Elevation of the centroid of the bleeder opening	Real	≥ 0.0	11.56
`diameter`	Diameter of the bleeder	Real	≥ 0.0	3.24

$$Q = 0.6A\sqrt{2g(H_u - H_d)} \tag{6.56}$$

for orifice flow with $H_d > H_{centroid}$ where H_u and H_d are the upstream and downstream heads.

For weir flow ($H_u \leq$ top of the bleeder) discharge is

$$Q = 3.13L(H_u - H_{invert})^{1.5} \tag{6.57}$$

If $H_d > Hinvert$, the discharge is multiplied by a tailwater correction factor (Brater and King, 1996).

$$CS = \left[1 - \left(\frac{H_d - H_{invert}}{H_u - H_{invert}}\right)^{1.5}\right]^{0.385} \tag{6.58}$$

Table 6.33: *Attribute definitions for* `<rectbleeder>`.

Attribute	Definition	Variable type	Suggested range	Example
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmid`	ID of the water mover being created	long integer	200000-300000	256987
`centroid`	Elevation of the centroid of the bleeder opening.	Real	≥ 0.0	13.24
`height`	Height of the bleeder opening	Real	≥ 0.0	2.58
`width`	Width of the bleeder opening	Real	≥ 0.0	2.65

The following example creates a rectangular bleeder with height = 0.2 m, width = 1.3 m, and elevation of the centroid = 7.6 meters to move water between water bodies 3 and 5.

```
 <watermovers>
 ....
  <rectbleeder id1="3" id2 = "5" wmID="210"
     centroid = "7.6" height="0.2" width="1.3">
  </rectbleeder>
..........
 </watermovers>
```

6.5.7 Bridges

The Yarnell equation borrowed from the HECRAS technical Reference Manual (Brunner, 2002) is used in the model to obtain a relationship between the discharge and the upstream and downstream heads. An example of its use is two cells separated by a road with a bridge spanning an opening in the road. A definition sketch is shown in Figure 6.10 and the attributes of `<yarnell>` are presented in Table 6.34

The Yarnell equation (Yarnell, 1934) is given in the HECRAS Technical Reference Manual (Brunner, 2002) as

$$H_1 - H_2 = 2K\left(K + 10\omega - 0.6\right)\left(\alpha + 15\alpha^4\right)\frac{V_2^2}{2g} \tag{6.59}$$

where
H_1 = upstream head,
H_2 = downstream head,
ω = ratio of velocity head to depth at the downstream cross section = $(V_2)^2/(2gD_2)$ where
D_2 = downstream depth, and
V_2 =downstream velocity,
α = Ratio of the area obstructed by the piers to the unobstructed area at the downstream cross section.

A lookup table is used to define the cross section as a function of head.

The pier coefficient, K, for various shape piers is listed below

Pier Shape	Yarnell K Coefficient
Semi-circular nose and tail	0.90
Twin-cylinder piers with connecting diaphragm	0.95
Twin-cylinder piers without diaphragm	1.05
90-degree triangular nose and tail	1.05
Square nose and tail	1.25
Ten pile trestle bent	2.50

Solving the Yarnell equation for discharge as a function of $H_1 and H_2$ yields

$$Q = A\left[2g\left\langle\sqrt{\left(K-0.6\right)^2 + \frac{20\left(H_1 - H_2\right)}{D_2K\left(\alpha + 15\alpha^4\right)}} - \left(K-0.6\right)\frac{D_2}{20}\right\rangle\right]^{\frac{1}{2}}\left[H_1 - H_2\right] \tag{6.60}$$

Table 6.34: *Attribute definitions for* `<yarnell>`*.*

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmid`	ID of the water mover being created	long integer	200000-300000	256987
`pshape`	Yarnell's pier shape coefficient	Real	0.5 - 3.0	1.05
`pwidth`	Total width of al the piers	Real	≥ 0.0	8.62
`botelev`	Bottom elevation below which there is no flow	Real	≥ 0.0	7.52
`da`	Depth-Area lookup table. The first column is the depth and the second column is the area of the canal cross-section	Real	≥ 0.0	see example input

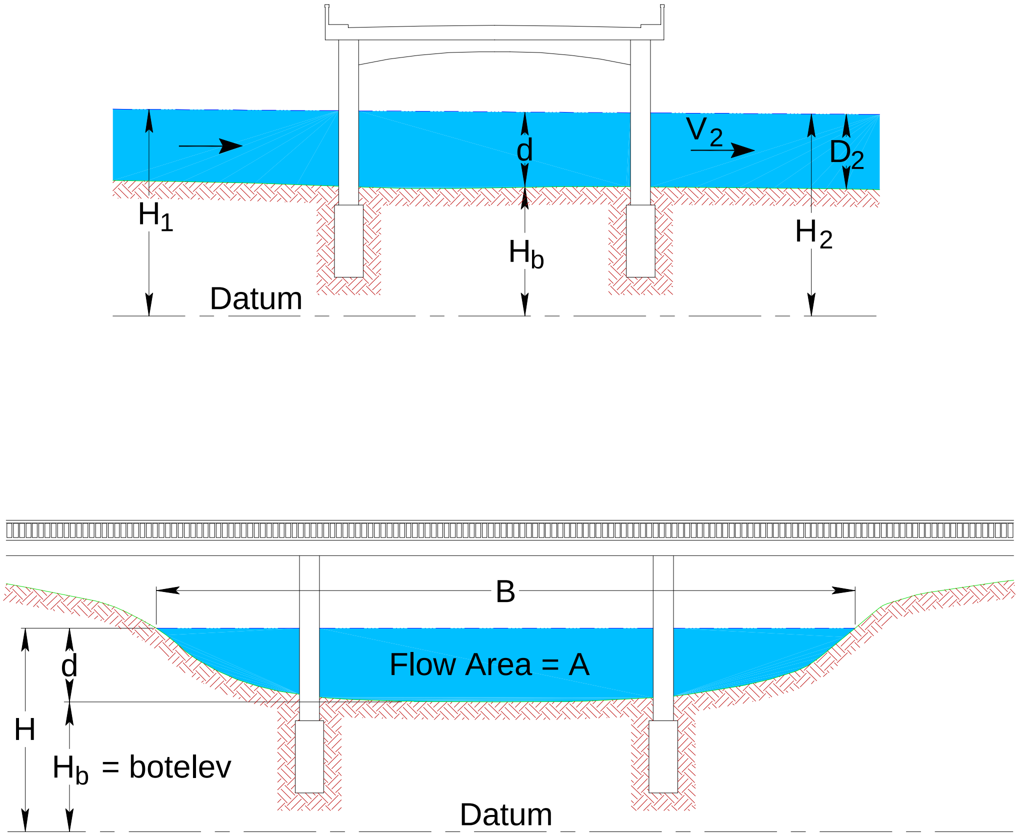


Figure 6.10: *Definition of cross sections used with the bridge routine.*

An example of a data set used for the Yarnell bridge routine is shown below. The bridge spans an opening between water bodies 5 and 11. The pier shape is a square nose and tail and the total pier width is 2 *m*. The cross sectional area increases from 0 to 1600 *m* as the depth increases from 0 to 8 *m*.

```
 <watermovers>
 ....
    <yarnell idY1 = "5" id2 = "11" pshape = "1.25" pwidth = "2.0" botelev = "499">
     <da>
     0.  0.
     1.  200.
     2.  400.
     8.  1600.
     </da>
    </yarnell>
....
  </watermovers>
```

6.5.8 Hydropower`<hydropower>`

Hydropower generation is part of water resources development. With the power demand in megawatts (MW), the flow is computed as

$$Q = \frac{Power * 1000}{\rho g * efficiency} \tag{6.61}$$

where the power is determined as the demand, or the capacity, whichever is higher. The attributes of a `<hydropower>` object are listed in Table 6.35

The following input will create an object to simulate a hydropower plant with a capacity of 10 MW between water bodies 12 and 45. The demand is specified in a DSS file ”demand.dss”.

```
<watermovers>
  <hydropower wiID = "212" id1 = "12" id2 = "45"
       capacity = "10" highhead = "12.3"
       lowhead = "5.6" efficiency = "0.9"
    <dss file= "demand.dss" pn="/L8/ ST 1/FLOW/01JAN1994/1DAY/ NON/"
       mult="0.2">
    </dss>
  </hydropower>
....
<watermovers>
```

Table 6.35: *Attributes of* `<hydropower>`.

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`id1`	ID of the upstream water body	long integer	100000-200000	153675
`id2`	ID of the downstream water body	long integer	100000-200000	148322
`wmID`	ID of the water mover	long integer	100000-200000	25456
`label`	Label for the water mover object	string	any string	Hoover Dam
`capacity`	Plant Capacity, MW	Real	≥ 0.0	1.5
`lowelev`	Lowest headwater elevation at which plant will operate	Real	≥ 0.0	8.6
`highhead`	Design head drop. Used to check the efficient operating range	Real	≥ 0.0	17.6
`lowhead`	Minimum head drop for plant operation	Real	≥ 0.0	3.25
`efficiency`	Overall plant efficiency as a fraction	Real	≥ 0.0	0.9
`mult`	Multiplier for demand	Real	≥ 0.0	0.78
The power demand is specified as `<const>`, a `<rc>` or in a file in `<dss>`, `<asciiform>`, or `<csv>` format as in the example in this section. Details are provided in Chapter 9				

6.6 One-Dimensional Canal Network Data - The XML `<network>` Element

HSE is capable of simulating diffusion flow in a canal network. The network can be a single network with loops, trees, and joints with up to four limbs. Completely disconnected pieces of canal networks can be simulated using the model, with proper boundary conditions. The primary advantage of diffusion flow models over full equation models using Preissmann's (Preissman, 1961) method is error control in the inertia terms. In areas such as the Everglades, the enhancement of the solution by the addition of the inertia term is negated by the errors in the same term under erratic topography. Any computational method used to simulate diffusion flow should be subjected to error control using methods suggested by (Lal, 2000c). Even when the model is unconditionally stable, use of short segments with large depths, low bed friction values and low slopes may create erroneous solutions.

6.6.1 Canal Data Input Under The `<network>` Element

The first step in setting up the canal network model is discretization of the canal network. GMS or ARCVIEW using the HSE-GUI can be used to carry out the discretization. After the discretization, the nodal connectivity, nodal coordinates, segment properties and segment connectivity are defined under the <`network`> element in the XML input. The subelements that are available to describe a canal network model are shown in Table 6.36. A canal simulation also requires a boundary condition file that is described in Section **??**.

As a general rule, geometry, cross section, and parameter data can all be described using the GMS map file. When the parameters of individual segments are not known, and only regional values are known, XML input can be used to assign values to the regions using an index file. A number of sets of parameters are specified in the XML input with each set assigned an id number. The index file lists the id of the parameter set to be assigned to each segment in the same order that the segments are defined in the GMS map file. In the model, values assigned using XML have precedence over values assigned in the GMS map file. Values in the map file may then, be modified using XML input without changing the map file.

Table 6.36: *Sub-elements and attributes under* `<network>`.

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<geometry>`	Network geometry file is to be specified.			
`file`	GMS file name	String	Any valid GMS file name	`enp_can.map`
`mult`	Multiplier for nodal coordinates	real	> 0	0.3048
`multxs`	Multiplier for segment cross section values. Often for unit conversion	real	any real	0.3048
`initial`	Name of the file containing segment initial conditions			
`file`	Initial condition file name	string	Any valid file name	L8ini.ini
`<network_bc>`	Network boundary conditions are specified in XML. Details in Section **??**			
`<arcs>`	Properties of canal segments to override the GMS file			
`<indexed>`	Indicates there is a file for indexing of canal segment properties			
`file`	Index file name	String	any valid file name	arcs.index
`<xsentry>`	Indicates that following information is to override GMS data			
`id`	Entry id for indexed values	integer	any integer	3
`label`	label for the entry	string	any string	Big Swamp
`<arcflow>`	Manning's n for canal flow is specified			
`n`	Manning's n for canal flow	Real	0.0 - 1.0	0.04
`<arcseepage>`	parameters for leakage between canal and cell are specified			
`leakage_coeff`	Leakage coefficient in equation 6.62	Real	0.0 - 1.0	0.00045
`<arcoverbank>`	parameters for overland flow between canal and cell are specified			
`bank_height`	Height of the canal lip	Real	0.0 - 5.0	0.6
`bank_coeff`	Coefficient for flow over the canal lip	Real	0.0 - 1.0	0.3
`<arclevee>`	Parameter for seepage through a levee is specified			
`coeff`	Coefficient for seepage between canal and cell through a levee. Overrides levseep1	Real	0.0 - 1.0	0.003

An example of XML input under the elements listed in Table 6.36 is shown below. Each of these inputs is described in detail in the following sections.

```
<network>
  <geometry file="mod_can.map" mult = "0.3048" multxs = "0.3048" </geometry>
  <initial file="mod_can.ini" </initial>
  <network_bc file="mod_can.bc" </network_bc>
  <arcs>
     <indexed file="arcs.index">
       <xsentry id="1">
         <arcflow n="0.2"></arcflow>
         <arcseepage leakage_coeff="0.000405">
       </xsentry>
       <xsentry id="2">
         <arcflow n="0.1"></arcflow>
         <arcoverbank bank_height="0.03" bank_coeff="0.2">
       </xsentry>
       <xsentry id="3">
         <arclevee  coeff = "0.0005"/>
         <arcflow n = "0.04"/>
       </xsentry>
     </indexed>
  </arcs>
</network>
```

6.6.2 Canal Network Geometry File `<geometry>`

The name of the canal network geometry file is specified using the `<geometry>` element. It is created in the GMS map format. Data in the canal network file is in either the node environment or the arc environment. Data in the node environment gives the two-dimensional layout of the canal. It specifies the locations of the ends of each canal segment and the id number for the segment. The arc environment includes data on each canal segment cross section and the values of parameters required to compute flow in the canal and the interaction between the canal and the surrounding cells.

The contents of the geometry file are described in Table 6.37. An example of a canal network within a 2-D mesh is shown in Figure 6.11. Table 6.38 and Table 6.39 show a sample geometry file for the canal. The canal consists of three segments configured to demonstrate the inputs available to describe all the properties of a canal. The subsequent sections describe canal flow and an interpretation of the information in the geometry file in Table 6.38 and Table 6.39

Table 6.37: *Definition of tokens used in the canal geometry and boundary condition files in GMS format.*

Token	Definition
Canal geometry	
`map`	This token indicates that this is the geometry file
Node environment	
`node`	Indicates the beginning of a node environment. This node environment ends with the `end` token. `id` and `xy` tokens are defined within this environment
`xy`	The x and y coordinate values of the node are provided on this line, with an optional third value not used
`id`	The ID of the node with the coordinates xy
Segment or arc environment	
`arc`	Indicates the beginning of the `arc` environment within which the canal segment properties are defined. The `arc` environment ends with the `end` token
Tokens within the `arc` environment	
`id`	The id of the segment between the specified nodes
`nodes`	Specifies the two nodes that define the end points of the segment
`type`	The type of the canal segment. `trapezoid` is the only option available. The trapezoid properties *bottom width*, *bottom elevation*, *side slope*, and *Manning's constant* are listed on the same line in order
`flowtype`	The options available are; 0 for normal flow when the flow length between segments is considered as the distance between the center points of the segments; 1 when flow head is assigned at the end of a canal, the segment length is doubled to account for the distance from the midpoint of the segment to the end. 2 when the segment length is considered to be very small. This is an optional parameter, and the default is 0
`length`	Length of the segment. This is an optional parameter, because the length is automatically calculated. If the canal is meandering, this parameter can be used to give a more accurate length
`leakage_coeff`	Defines the existence of stream-aquifer interaction, and provides the value of the k/δ coefficient. The cell id and the length of the segment in the cell are optional parameters
`bank_height`	Defines the existence of stream-overland flow interaction, and provides the ”lip height” defined later. The cell id and the length of the segment in the cell are optional parameters
`bank_coeff`	Value of the coefficient for flow from the cell over the lip to the segment. The cell id and the length of the segment in the cell are optional parameters
`levseep1`	Levee seepage coefficient 1. The cell id and the length of the segment in the cell are required parameters
`levseep2`	Levee seepage coefficient 2. The cell id and the length of the segment in the cell are required parameters

Table 6.38: *Sample canal geometry file, part 1 of 2.*

```
MAP
BEGCOV
ACTCOV
COVNAME "default coverage"
COVELEV 0.0
COVATTS GENERAL
NODE
XY 2.501E+3 2.501e3 0.0
ID 1
END
NODE
XY 5.001e+3 5.001E3 0.0
ID 2
END
NODE
XY 1.101e+4 5.001e3 0.0
ID 3
END
NODE
XY 1.2501e4 1.2501e4 0.0
ID 4
END
ARC
type trapezoid 100.0 498.0 0.5 0.05
ID 1
```

Table 6.39: *Sample canal geometry file, part 2 of 2.*

```
NODES        1        2
flowtype 1
length 4250.0
leakage_coeff 0.0005  7  4250.0
bank_height 0.5  7  4250.0
bank_coeff 3.2  7  4250.0
END
ARC
type trapezoid 100.0 496.0 0.5 0.05
ID 2
NODES        2        3
flowtype 1
length 6200.0
leakage_coeff 0.0005  8  5100.0  9  1100.0  3  1100.0
bank_height 0.5  8  5100.0  9  1100.0  3  1100.0
bank_coeff 3.2  8  5100.0  9  1100.0  3  1100.0
levseep1  0.004  14  5100.0  15  1100.0
levseep2  0.006  14  5100.0  15  1100.0
END
ARC
type trapezoid 100.0 495.0 0.5 0.05
ID 3
NODES        3        4
flowtype 1
length 13200.0
leakage_coeff 0.0005  15  6000.0  6  3000.0  12  4200.0
bank_height 0.5  15  6000.0  6  3000.0  12  4200.0
bank_coeff 3.2  15  6000.0  6  3000.0  12  4200.0
END
ENDCOV
```

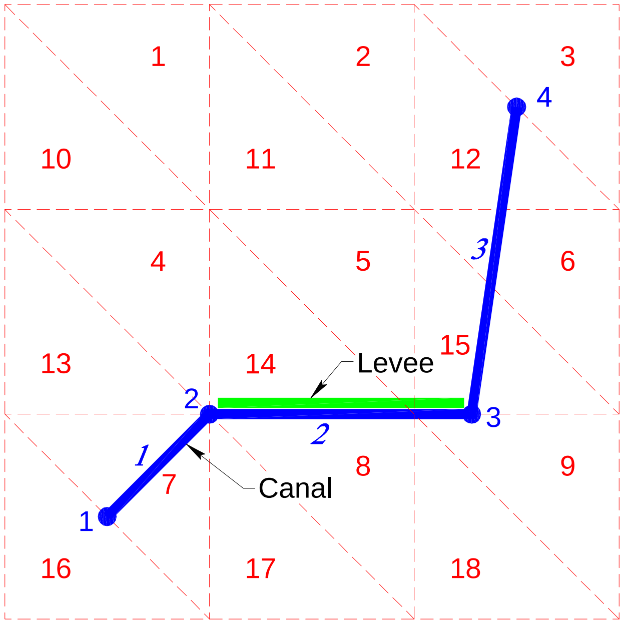


Figure 6.11: *A sketch of the canal network.*

6.6.2.1 Description Of Nodes

The four node environments in the geometry file specify the locations of nodes 1, 2, 3 and 4 in Figure 6.11. Each node is assigned an id number by which it can be referenced. The 0.0 after each set of coordinates is reserved for a vertical position that is not used in a 2D model. The format is shown below. In the sample geometry file node 3 is located at $x = 1.101e + 4$ and $y = 5.001e + 3$.

```
NODE
XY x_coordinate y_coordinate 0.0
ID node_ID
END
```

6.6.2.2 Canal Cross Sectional Geometry

Canal segments are defined within the `ARC` environment. The id's of the nodes defining the ends of a segment are specified after the token `NODES`. Figure 6.12 shows a trapezoidal canal cross section used to simplify a typical canal cross section. In addition to the geometry of the canal cross section a number of segment parameters can be specified. These are Manning's n, and the parameters required to describe seepage between a canal and adjacent cell through groundwater, levee, or overland flow between the segment and one or more cells. These parameters are described in detail in the following sections.

The straight line distance between nodes is computed internally as the length of a segment. If the segment is not straight the length is greater than this computed value and the actual length may be specified in the `ARC` environment after the token `length` in the map file. The flow type for segment 1 is 0, indicating that normal flow calculations are used. For segment 2, the flow type is 1 meaning that a flow head is assigned at the end of the canal.

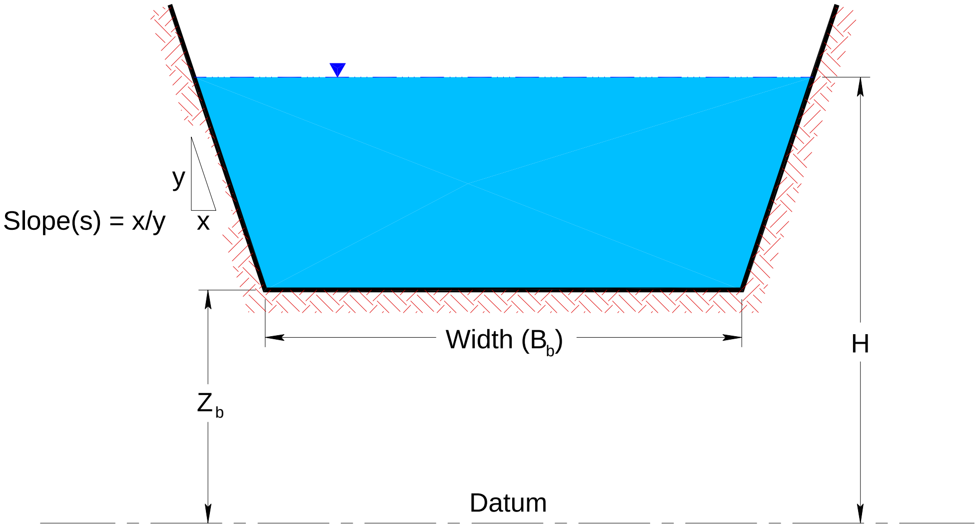


Figure 6.12: *Trapezoidal canal cross section.*

6.6.3 Stream-Aquifer Interaction

The values of parameters for calculating the seepage and overland flow between a canal segment and the neighboring cell(s) can be specified in the `ARC` environment. The format is shown below.

```
ARC
ID segment_ID
NODES  defining_node_1 defining_node_2
type trapezoid [canal_width] [bot_elev] [side_slope_x/y] [Manning]
leakage_coeff value_of_coeff [cell_no.] [overlap_len] ... [cell_no.] [overlap_len]
bank_height value_of_coeff [cell_no.] [overlap_len] ... [cell_no.] [overlap_len]
bank_coeff value_of_coeff [cell_no.] [overlap_len] ... [cell_no.] [overlap_len]
levseep1 value_of_coeff [cell_no.] [overlap_len] ... [cell_no.] [overlap_len]
levseep2 value_of_coeff [cell_no.] [overlap_len] ... [cell_no.] [overlap_len]
END
```

The ... represent any number of pairs of `[cell_no.]  [overlap_len]` in the same line. They need to be on the same line for the interaction to be based on the coefficient at the beginning of the line.

Figure 6.13 shows a definition sketch used in the conceptualization of stream-aquifer and stream-overland flow interaction.

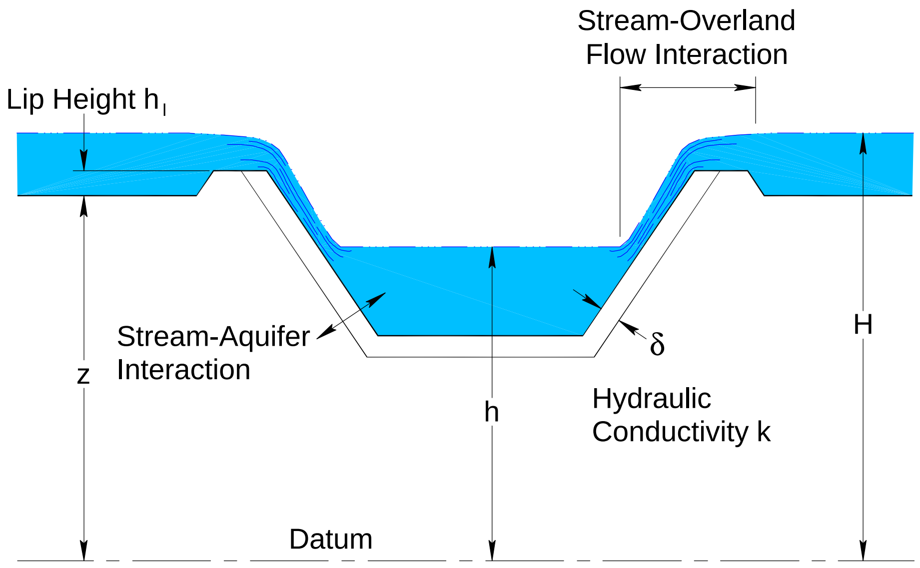


Figure 6.13: *A definition sketch showing flow interaction with the canal.*

The token `leakage_coeff` is used to represent k/δ from which flow between the aquifer and the canal is computed as

$$q = \frac{k}{\delta} p(H - h) \tag{6.62}$$

where q = seepage flow per unit length of the canal,
k = hydraulic conductivity of bottom sediment,
δ = thickness of the sediment layer,
p = wetted perimeter of the canal,
h = water level in the canal segment,
H = water level in the cell.
Water may flow in either direction.

Individual segment stream-aquifer interaction parameter values are specified with the token `leakage_coeff` defined within the `ARC` environment in the canal geometry file. When `leakage_coeff` is non-zero, canal ground water interaction becomes active. (Lal, 2001) described critical values of k/δ below which the interaction is insignificant, and above which the interaction is full. In segment 3 in the sample geometry file, the coefficient used for k/δ is 0.0005, and the length of the segment in cells 15, 6 , and 12 is 6000.0, 3000.0, and 4200.0 meters, respectively. The sum of these three overlaps is 13200.0 meters, the total length of the segment.

The dotted lines represent any number of pairs of `[cell_no.]` `[overlap_len]` in the same line. They need to be on the same line for the interaction to be based on the coefficient at the beginning of the line.

6.6.4 Stream-Overland Flow Interaction

Overland flow between a canal segment and a cell is modeled as weir flow over a "lip" along the edge of the canal segment. The flow is shown schematically in Figure 6.13. The lip height is specified after the `bank_height` token and the weir coefficient, C, after the `bank_coeff` token in the canal geometry file. Flow is computed as

$$Q = CL\sqrt{g}h^{1.5}$$

where
C = weir coefficient,
L = length of overlap between the segment and the cell, and
h = H - (Z + h_l), defined in Figure 6.13
A tailwater correction of

$$Q = Q * [1 - (\frac{h_{tw}}{h})^{1.5}]^{0.385}$$

is applied, where h_{tw} = height of downstream head above the "lip." When the head in the canal is greater than the head in the cell, flow from the canal to the cell is computed using the same equation with the heads in the canal and in the cell reversed. This streambank type water mover is created *only* if bank height ≥ 0.

6.6.5 Levee seepage

Levee seepage is an important flow component in South Florida modeling. In the SFWMM, a separate function had to be developed for levee seepage because of the difficulty of capturing it easily using a single transmissivity based flow function. Figure 6.14 shows a cross section of the levee illustrating components of seepage.

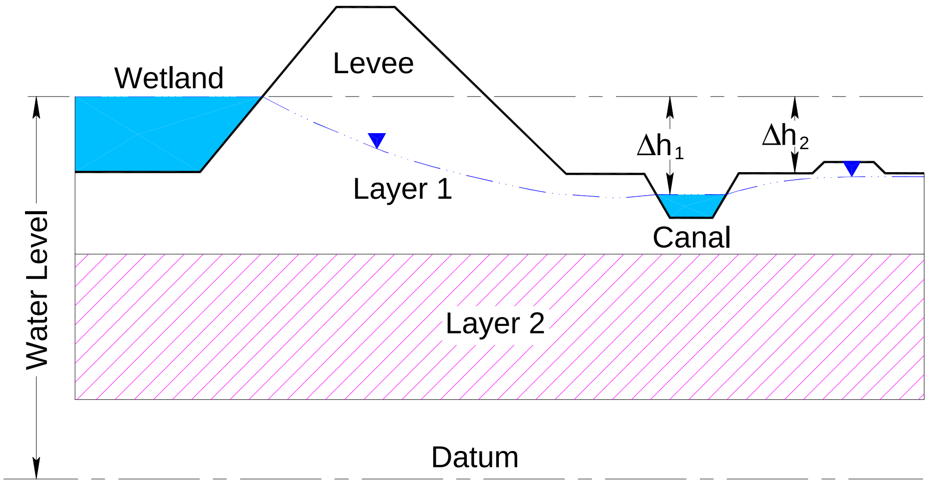


Figure 6.14: *Definition sketch showing levee seepage.*

Figure 6.15 shows the plan view of the same levee along a cell wall. In HSE, cell walls that are configured as no-flow walls are often placed along levees. Often a levee is placed on only one side of a canal, so that it is necessary to specify which cell interacts with the canal by seepage through the levee.

In the water management model, levee seepage is defined as the total discharge into the canal, and is computed using the equation

$$q_l = \beta_0 + \beta_1 \Delta h_1 + \beta_2 \Delta h_2 \tag{6.63}$$

where
β_0, β_1 and β_2 are constants derived from experimental data, and
$\Delta h_1 = h_{wetland} - h_{canal}$, and
$\Delta h_2 = h_{wetland} - h_{cell}$,
h_{canal} is the head in the canal,
$h_{wetland}$ is the head in the cell across the levee from the canal, and
h_{cell} is the head in cell that the canal crosses,
While the coefficients of equation 6.63 can be derived from experimental data, they are typically derived from analytical or numerical models.

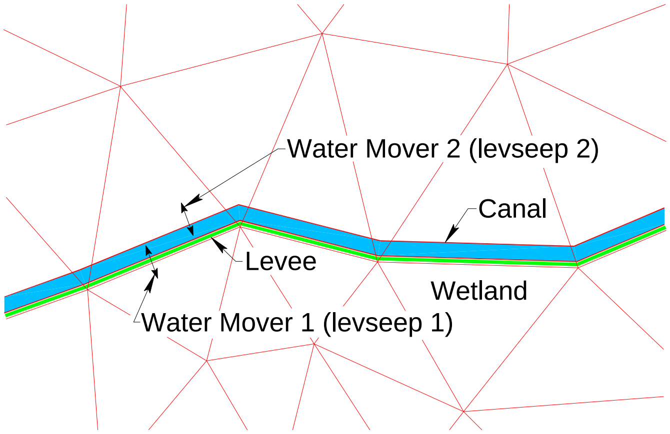


Figure 6.15: *Plan view showing the placement of a levee.*

Since HSE is based on water movers that consider only two water bodies at a time, equation 6.63 can be written as

$$q_l = -\beta_2(\Delta h_1 - \Delta h_2) + (\beta_1 + \beta_2)\Delta h_1 \tag{6.64}$$

in which, $\beta_1 + \beta_2$ is the coefficient for moving water between the wetland and the canal; β_2 is the coefficient for moving water from the right bank of the canal to the canal, assuming the constant β_0 to be negligible in equation 6.63. The constants β_2 and $\beta_1 + \beta_2$ are the coefficients of the new levee seepage water movers. Levee seepage is computed as the sum of two water movers;

$$Q1 = levseep1\,(h_{wetland} - h_{canal})$$

$$Q2 = levseep2\,(h_{cell} - h_{canal})$$

Coefficients `levseep1` $= \beta_1 + \beta_2$ and `levseep2` $= -\beta_2$ are defined in the map file using the following format.

```
ARC
type trapezoid canal_width bott_elev side_slope_x/y Manning
leakage_coeff value_of_coeff
bank_coeff value_of_coeff [cell_no.] [overlap_len] ... [cell_no.] [overlap_len]
levseep1 value_of_coeff [cell_no.] [overlap_len] ... [cell_no.] [overlap_len]
levseep2 value_of_coeff [cell_no.] [overlap_len] ... [cell_no.] [overlap_len]
.....
ID segment_ID
NODES  defining_node_1 defining_node_2
END
```

In the sample canal geometry file displayed in Table 6.38 and Table 6.39 as applied to Figure 6.11, there is a levee between canal segment 2 and cells 14 and 15. The coefficients β_1 and β_2 are 0.004 and 0.006 and the length of the canal segment along the boundary between cells 8 and 2 is 5100.0 m and between 9 and 15 is 1100.0 m.

6.6.6 Initial Condition File `<initial>`

The initial condition file lists the heads in each canal segment at the start of the simulation. An example for the canal network shown in Figure 6.11 is shown below. The heads are specified so that the depth in each canal segment is 5.5 meters.

```
netinit
503.5
501.5
500.5
```

6.6.7 Overriding Canal Properties Using XML

When a few sets of canal parameters are to be applied to the individual canal segments or zones of segments, the element `<arcs>` with an index file can be used. Each set of canal parameters in the XML file is assigned an id number under the `<xsentry>` subelement of `<indexed>`; see Table 6.36. The entries in the index file specify which set of parameters is used for each canal segment. Parameter values specified under `<xsentry>` and assigned

Table 6.40: *Sample index file.*

```
DATASET
OBJTYPE "network"
BEGSCL
ND   3
NAME "segment index"
TS 0 0.0
1.0
3.0
2.0
```

to segments with the index file will override parameter values specified in the map file. The index file in Table 6.40 when used with the sample XML input file in Section 6.6.1 will change `leakage_coeff` in segment 1 from 0.0005 to 0.000405, the `bank_height` from 0.5 to 0.03 and the `bank_coeff` from 3.2 to 0.2 in segment 2. The levee seepage parameter `<coeff>` will be set to 0.0005 and Manning's n to 0.04 in segment 3. The parameters in a segment may be left unchanged from the values specified in the map file by assigning an index value ”j” to the segment along with the following XML input where j is an integer. No new values will be assigned.

```
<network>
   <arcs>
      <indexed file="arcs.index">
       <xsentry id="j">
       </xsentry>
      </indexed>
  </arcs>
</network>
```

6.7 Lakes and Ponds `<lakes>`

Lakes and ponds are simulated as independent water bodies in the model. They do not act as cells in the regional solution and their only interaction with cells in the mesh is through seepage in either direction or through other user created water movers. There are no default water movers for lakes. The amount of water in a reservoir is calculated using the equation of mass balance

$$A_s \frac{dH}{dt} = \sum Q_{in} - \sum Q_{out} \tag{6.65}$$

where
A_s = the surface area of the lake,
H = the head in the lake, and
$\sum Q_{in}$ and $\sum Q_{out}$ = rainfall, evaporation, seepage into and out of the lake/pond and the flow in any user created water movers.

Once the storage is calculated, the water level is estimated using a 1-D lookup table or from a calculation assuming a cylindrical or parabolic shape for the lake as selected by the user.

Neither lakes nor ponds are discretized in the model. Lakes are larger water bodies, and the mesh cell discretization can surround the lake with cell walls in contact with the lake boundary. Ponds are smaller water bodies, and occupy a small space inside a triangular model mesh cell. Ponds situated within a single cell are considered to be sufficiently small such that they do not disrupt the 2-D flow although they do decrease the area of the cell by the area of the pond. Whether a water body is treated as a lake or a pond is specified by the user. Figure 6.16 shows a definition sketch of a reservoir, to which water is fed from an upstream river. Figure 6.17 shows the discretization around a lake and the placement of a pond entirely within a cell.

Both lakes and ponds are defined in the model as water bodies. However, there are no default water movers for lakes, unlike the case of cells or segments. Lakes and ponds are defined under the `<lakes>` element in the XML input. Valid sub-elements and attributes within `<lakes>` are shown in Table 6.41 with additional details in Tables 6.42 and 6.43.

A sample XML input for a lake and a pond is shown in Table 6.44. In this input Lake Kalawewa is a lake since "supplant" < 0. The relationships between stage, area and volume are defined by 1-D lookup tables; rainfall is a time series in a DSS file; RefET is constant, and ET is greater over shallow water than deep water with 4.3 meters (specified by the user) being the depth dividing shallow and deep water. Frog pond (supplant = +2) has daily rainfall in a csv file and a constant RefET. There is less ET as a fraction of refET than in the lake because of the smaller coefficients under `<litZoneET>` and the dividing depth between deep and shallow water (specified by the user) is only 2.8 meters.

Table 6.41: *Sub-elements and attributes used to define lake properties under the* `<lakes>` *element.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<lake>`	Indicates a new lake/pond water body is to be created			
`id`	Lake or Pond ID	Long	Any valid long integer	234625
`label`	Name of the lake	string	Any valid string	Ocala
`head0`	Initial head in the lake	real	any real	13.64
`supplant`	Indicates whether the wb is a lake or a pond	integer	$\leq$ 0 for lake, $\geq$ 0 for pond	-2
`<parabolic>`	Indicates that lake area, volume, and stage are computed assuming a parabolic shape. Details in Table 6.42			
`<cylinder>`	Indicates that lake area, volume and stage are computed assuming a cylindrical shape. Details in Table 6.42			
`<sv>`	Indicates that the stage-volume relationship follows in a 1-D lookup table			
`<sa>`	Indicates that the stage-area relationship follows in a 1-D lookup table			
`<refet>`	Indicates that the reference ET is specified			
`<rain>`	Indicates that rain is specified			
Details on specifying input data for `<refet>` and `<rain>` in `<const>`, `<rc>`, `<dss>`, `<asciiform>`, and `<csv>` formats is explained in Chapter 9				
`<EvapRainStressors>`	Parameters for calculation of ET are specified. Details in Table 6.43			
`<lake_bc>`	Lake boundary conditions are defined. Details in Section 7.4			

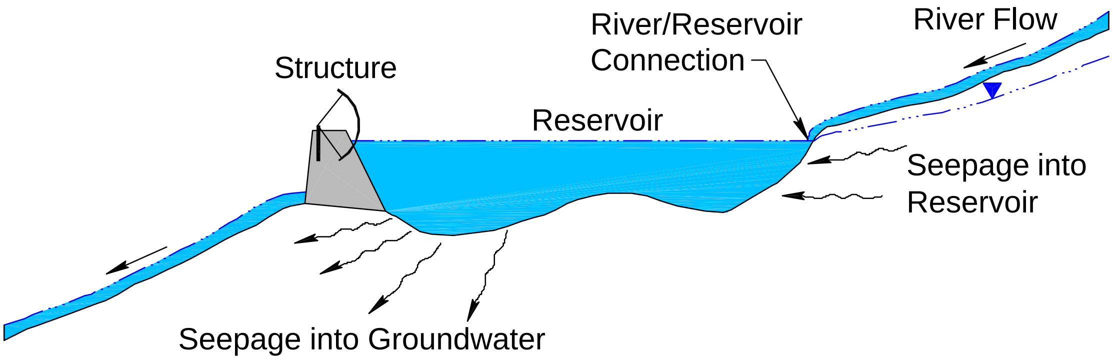


Figure 6.16: *Schematic diagram of a reservoir formed in a river.*

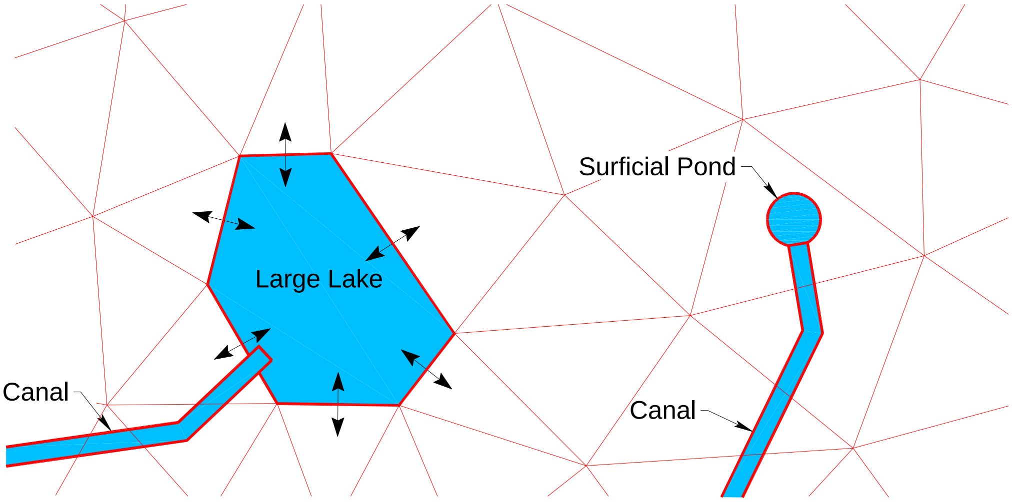


Figure 6.17: *Discretization around a lake and a pond.*

Table 6.42: *Elements and attributes used to define lake area and volume under the* `<lakes>` *element.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<parabolic>`	Indicates that lake area, volume, and stage are computed assuming a parabolic shape			
`toparea`	Area of the lake surface when full	real	Any valid real	6.3E+5
`top`	Head of the lake when full	real	Any valid real	23.56
`bot`	Elevation of the lake bottom	real	Any valid real	4.87
`<cylinder>`	Indicates that lake area, volume and stage are computing assuming a cylindrical shape			
`toparea`	Area of the lake surface when full	real	Any valid real	6.3E+5
`bot`	Elevation of the lake bottom	real	Any valid real	4.87

Table 6.43: *Elements and attributes used to define* `<EvapRainStressors>`.

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<litZoneET>`	Lake ET parameters are specified			
`lakeID`	ID of the lake	Integer	Any valid wb ID	138245
`owcoef`	Open Water coefficient for RefET	Real	Any real	1.00
`swcoef`	Shallow water coefficient for RefET	Real	Any real	0.85
`swdepth`	Depth that divides shallow and deep water	Real	Any real	5.5

6.7.1 Rainfall and Evapotranspiration

Two major components of the water budget of a lake or pond are precipitation and evapotranspiration. While the contribution of precipitation is straightforward, evapotranspiration depends on the surface area of the lake and the depth of the water in addition to the RefET values assigned to the water body. In order to account for the different rates of evapotranspiration over shallow and deep water, the total ET over the lake is calculated as

$$ET\_Volume = [swcoef * (DryArea + ShallowArea) + owcoef * DeepArea]\, RefET \tag{6.66}$$

where
DryArea = the area of the lake that is dry,
ShallowArea = the area of the lake that is shallow,
DeepArea = the area of the lake that is deep,
and the coefficients and the dividing depth between deep and shallow water are specified under `<EvapRainStressors>` as described in Table 6.43.

Table 6.44: *Sample XML input for lakes and ponds.*

```
<?xml version="1.0">
<sfrsm version="1.0">
 <lakes>
   <lake id="235823" label="Kalawewa" head0="504.0" supplant="-2"
     <sv>
          0.0     0.0
        400.0     3.0e8
        600.0    10.0e8
     </sv>
     <sa>
          0.0     0.0
        400.0     1.5e7
        600.0     4.2e7
     </sa>
     <rain>
       <dss file="SFPrec.dss" pn="/ENP/Reg1//EVAPI/1DAY/SFCalc/"
       mult="0.3048" units="feet" </dss>
     </rain>
     <refet>
       <const dbintl="1440" value="0.14" mult="0.3048" </const>
     </refet>
     <EvapRainStressors>
       <litZoneET lakeID="324572" owcoef="0.8" swcoef="0.9"
       swdepth="4.3" </litZoneET>
     </EvapRainStressors>
   </lake>

   <lake id="248795" name="Frog Pond" head0="504.0" supplant="2"
     <parabolic toparea="6.3e+5" top="510.0" bot="490.5"> </parabolic>
     <rain>
       <csv file="FrogPrec.dat" dbintl="1440" label="gauge1" </csv>
     </rain>
     <refet>
       <const dbintl="1440" value="0.28" mult="1.0" </const>
     </refet>
     <EvapRainStressors>
       <litZoneET lakeID="31568" owcoef="0.6" swcoef="0.85"
       swdepth="2.8" </litZoneET>
     </EvapRainStressors>
   </lake>

  </lakes>
```

Table 6.45: *Elements and attributes used to define the lake seepage water mover.*

Element or Attribute	Definition	Variable type	Suggested range	Example
`<lakeseepage>`	Indicates the creation of a `<lakeseepage>`watermover.			
`lakeID`	Lake ID number	integer	Any valid lake ID	123723
`wbID`	Cell ID number that interacts with lake	integer	Any valid cell ID	469852
`wmID`	Water Mover id	integer	Any valid wm ID	35469
`label`	Label for the Water Mover	string	Any string	Lake Wales seepage
`length`	Length of the shoreline in contact with the cell	real	≥ 0.0	689.4
`conveyance`	Transmissivity of aquifer as defined in Eq. 6.67	real	≥ 0.0	0.0054

6.7.2 Lake Seepage `<lake_seepage>`

Seepage into and out of lakes is computed using `<lake_seepage>` water movers that are defined by the user in the main `<hse>` XML environment under the `<watermovers>` element. If `<lake_seepage>` is not defined, there is no mechanism for leakage. Seepage can occur from the lake to one or more cells and/or from cells to the lake. The rate of seepage is computed as

$$Seepage = LCD(H_u - H_d) \tag{6.67}$$

where L and C are the length and conveyance as described in Table 6.45,
H_u and H_d are the higher and lower heads in the lake and the cell, and
D = the depth of water in the lake if the head in the lake is higher or (H_{cell}-$H_{lakebottom}$) if the head in the cell is higher.
The XML input to create a `<lake_seepage>` water mover is defined in Table 6.45. Table 6.46 shows an XML file that defines seepage between Lake Kalawewa (LakeID 235823) and wb 597351 and between Frog Pond (LakeID 248795) and wb 746533.

Table 6.46: *Sample XML input for lake seepage.*

```
<watermovers>
  <lakeseepage>
    <lakeseepage lakeID ="235823" wbID="597351"
          length="1000.0" conveyance="0.05">
    </seepage >
    <seepage lakeID ="248795" wbID="746533"
          length="2000.0" conveyance="0.08">
    </seepage >
  </lakeseepage>
</watermovers>
```

6.8 Storage and Stage-Volume Converters - The XML `<svconverter>` Element

The South Florida landscape is relatively flat when compared with the rest of the country. But within the range of elevations close to the average ground elevation, the hydrological characteristics may change significantly. Some of the characteristics that change rapidly are the water storage volume per unit change in head, the ET rate, and the overland flow roughness. Stage-volume converters `<svconverter>` have been developed to allow a more accurate representation of the volume of water stored at different water levels. Depending on the area under water, wetlands can store variable amounts of water at various depths. A flat ground with a designated storage coefficient below ground level and the assumption of open water above ground level is generally a poor representation of wetland storage conditions. However, this has been the standard method used to conceptualize water storage above and below ground. This section describes ways of representing elevation-storage relationships that better represent the micro-topography in the cell of a regional model. Figure 6.18 shows a section of a cell with an undulating ground surface. In the XML representation, the stage-storage conversion behavior is defined in the `<mesh>` environment using the element `<svconverter>`. A single `<scvonverter>` can be defined for the entire model, or the cells can be indexed to use different converters in different areas. Examples of both are given in the following sections.

The same idea can be extended to capture the cross-sectional area versus stage characteristics of canal segments. Canal data are specified using a number of methods. Many times, cross sections are approximated using rectangular and trapezoidal shapes. At other times, lookup tables are used. The SV converter is used to translate any type of cross section data to give the required area and the width properties. The reverse translation is carried out by the same converter without any loss of mass. The SV converter simplifies many of the complexities associated with model geometry.

6.8.1 Representation Of A Flat Ground Surface

In most early representations of a flat ground surfaces, the volume of water in a cell below the ground level is computed as

$$(H - z_B)s_c, \tag{6.68}$$

and the volume above ground is represented as

$$(H - z) + (z - z_B)s_c, \tag{6.69}$$

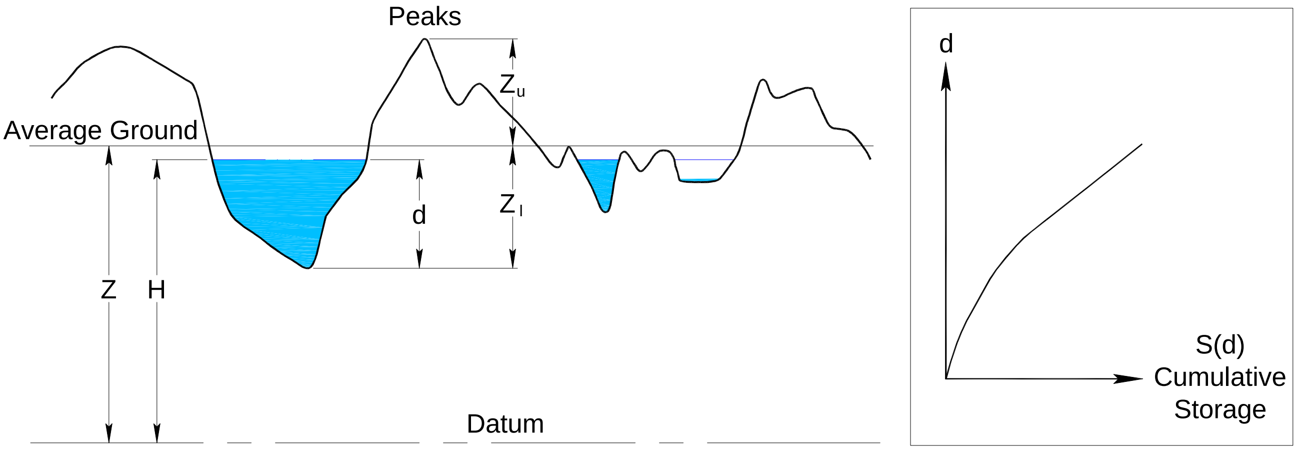


Figure 6.18: *Describing stage-storage characteristics in micro-topography.*

where
H = the head,
z = the ground surface elevation,
z_B = the elevation of the bottom of the aquifer, and
S_c = the storage coefficient.

If this behavior is to be imposed for the entire model, the following could be defined within the `<svconverter>` environment.

```
<svconverter>
      <constsv sc="0.2"> </constsv>
</svconverter>
```

6.8.2 Representation Using A Lookup Table

The actual ground surface elevation varies between the highest and lowest points in a cell. Until the water starts to collect as puddles, at the lowest elevations, the water storage is computed as $(z - z_b)s_c$. As the water level rises above the lowest point (Z-Z_l in Figure 6.18), the volume of water stored above the lowest point can be measured as a function of depth above this elevation. The water stored in this layer per unit thickness is

$$S(d) = \sum \alpha_A(H) + (1 - \alpha_A(H))s_c(H) \tag{6.70}$$

where
$\alpha_A(H)$ = fraction of open water area at a water level of H and

$s_c(H)$ = storage coefficient of solid ground.

When the water level is above the highest peak ($Z+Z_u$ in Figure 6.18), the added volume has a storage coefficient of 1.0. The total volume of water can be computed by integrating (6.70) over depth. This integral, which represents the total volume of water stored above the lowest ground depression of a cell per unit area of a cell, is represented in the model with a stage-volume converter using the elements and attributes in Table 6.47.

Below the range of the `<lookupsv>` stage-volume converter, the available storage in the soil is $(H - datum) * s_c$. Beginning at the elevation (*ground surface - below*) the storage in the lookup table is added to the storage below that elevation to compute the *total* storage. The highest elevation at which the lookup table applies is (*ground surface + above*). Assuming a ground elevation of 10 m above the datum, the total storage at elevation 10.6 m in the example below is

$$storage = (10 - 1.5)(0.2) + 0.24 = 1.94$$

where 0.24 m is the storage interpolated at a depth of 0.6 m in the lookup table `<sv>.`

The curve in Figure 6.18 describes the storage properties of a cell as a function of the head and the average ground elevation for the cell. If the entire model domain is to use the same curve, but each cell adjusted to its own ground elevation, the following example can be followed.

```
<lookupsv sc="0.2" below="1.5" above="1.0">
         <sv>
           0.0   0.0
           0.2   0.04
           0.4   0.12
           0.8   0.36
           1.0   0.56
           2.0   1.56
           3.0   2.56
         </sv>
 </lookupsv>
```

The attributes `<below>` and `<above>` can be used to assign the same SV behavior to different terrain when the vertical undulations are greater or less. These values specify the vertical extent of the lookup table about the ”surveyor’s” ground surface (the elevation of the ground surface if it were leveled).

Table 6.47: *Elements and attributes used to define a* `<lookupsv>` *SV converter in the* `<mesh>` *environment.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<lookupsv>`	Designates an `<lookupsv>` stage-volume converter.			
sc	Storage coefficient below the SV converter, s_c	Real	0 - 1	0.23
below	Depth of the lowest point below the ground surface at which the lookup table applies	Real	0 - 5	1.3
above	Height of the highest point above the ground surface at which the lookup table applies	Real	0 - 5	1.3
sv	Table of depth(d) and cumulative storage values. d is defined as $H - (Z - below)$. The SV converter is applied between $Z - below$ and $Z + above$ where Z is the "surveyor's" ground elevation			

6.8.3 Use Of More Than One Type of SV converter

If a mixture of two or more types of SV converters is to be used, indexing is available as shown in the following example.

```
<svconverter>
  <indexed file="sv.index">
    <entry id="1" label="const">
      <constsv sc="0.2"> </constsv>
    </entry>
    <entry id="2" label="lookup">
      <lookupsv sc="0.2" below="1.5" above="1.0">
        <sv>
          0.0   0.0
          0.2   0.04
          0.4   0.12
          0.8   0.36
          1.0   0.56
          2.0   1.56
          3.0   2.56
        </sv>
      </lookupsv>
    </entry>
  </indexed>
</svconverter>
```

In the example, the index file ”sv.index” is an ASCII file in GMS format describing the assignment of the two types in this example to the cells in the mesh.

Chapter 7

Boundary Conditions

As with any model, RSM simulations depend heavily on the forcing functions that drive the model. The major forcing functions are rainfall, evapotranspiration, inflow from rivers and streams, known outflows from the model domain, and the specification of known heads during the simulation period. These functions generate the hydraulic conditions that are used to set initial head conditions as well as boundary conditions. The boundary conditions specify the inflows and outflows at specified locations during the period of simulation. Much of the effort in building a model is the collection and preparation of data to accurately represent these processes.

The input data needed to specify initial and boundary conditions can be a limiting factor in creating accurate RSM models. Good quality input for initial and boundary conditions are essential for successful system simulation. Proper conceptualization and quantification of these conditions can be accomplished if sufficient good quality data exists and enough knowledge exists about the hydraulic behavior of the system being simulated. In this Chapter, the boundary conditions for regional 2-D flow, network, and lake are presented.

7.1 Boundary Conditions For Two-Dimensional Flow `<mesh_bc>`

Boundary condition (BC) for 2-D overland flow and 2-D groundwater flow are described in this chapter. Both 2-D overland flow and groundwater flow boundary conditions can be specified using cells or cell walls. The most commonly used boundary conditions are flow boundary conditions and head boundary conditions. All 2-D BC's for mesh cells are defined within the `<mesh_bc>` environment as described in detail and illustrated below. Time dependent boundary conditions for general water bodies such as canal segments, lakes and ponds as well as 2-D flow can be specified as a time series in file formats described later in Section 9.1 . All boundary conditions are implemented by creating water mover objects designed to move water in such a way as to achieve the prescribed flow or head.

7.1.1 Available Boundary Condition Types

Both cells and cell walls can be used to impose 2-D boundary conditions. Walls are ideal to assign head and general head type BC's. Cells are ideal for discharge type BC's. When assigning values to a series of cells or walls, methods are available to specify the locations of the cells or the walls, and assign values obtained by interpolating between given time series or constant values. These methods can save time by not requiring the user to enter a long list of values for individual cells or walls. The available cell based boundary conditions are listed in Table 7.1 and wall based BC's are shown in Tables 7.2 and 7.3. Section 7.1.2.1 describes the methods available to define the locations of the walls and cells to which the boundary conditions apply, Section 7.1.2.2 describes the methods available to define the available interpolation methods and the interpolation weightings. There are numerous options for specifying cell and wall boundary conditions. These options will be explained along with example XML input to demonstrate each. These descriptions and examples should clarify the information in Tables 7.1 through 7.3.

7.1.2 Defining Attributes Of 2-D BC's

The example XML input in Table 7.4 demonstrates many of the elements and attributes available to define 2-D boundary conditions. The application of these specifications on a mesh are indicated in Figure 7.1. The following sections refer to this example XML input to demonstrate the application of the boundary condition options described. The relevant XML input from Table 7.4 is also included in the description of each option.

Table 7.1: *Elements and attributes used to describe two-dimensional boundary conditions applied to cells in the* `<mesh_bc>` *environment. Element names are highlighted.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<well>`	Indicates a constant or time series of flow into or out of a cell			
`cellid`	ID of the cell that receives the flow	Long	Any valid cell ID	234625
`wellid`	Optional user-defined ID for the cell to be used later in water budget output	Long	Any valid cell ID	234625
`label`	Optional label used to describe the flow	String	Any string	WS well or creek inflow
`<cellhead>`	Indicates a constant or time series head value assigned to a cell			
`id`	ID of the cell whose head is specified	Long	Any valid cell ID	546987
`bcid`	ID assigned to the BC	Long	Any valid cell ID	234625
`label`	Optional label to describe the boundary condition	String	Any string	STA−3
`<cellghb>`	Indicates a constant or time series head value assigned to a general head boundary condition for a cell			
`id`	ID of the cell that the BC is applied to	Long	Any valid cell ID	546987
`value`	The coefficient in equation 7.5	Real	Any valid real	0.35
`label`	Optional label to describe the boundary condition	String	Any string	Lake Calloway
Each of the elements `<well>, <cellhead>` and `<cellghb>` has the following sub-elements available for specifying the flow or head: `<const>`, `<dss>`, `<asciiform>`, `<csv>`, and `<rc>`. These elements and their attributes are described in detail in Section 9.1				

Table 7.2: *Elements used to define the* `<wallhead>` *and* `<wallghb>` *boundary conditions applied to walls in the* `<mesh_bc>` *environment. The elements are in shaded cells.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<wallhead>`	Indicates a constant or time series head value assigned to a wall or walls.			
`section`	Specifies whether overland flow, groundwater flow or both are affected by the BC	String	ol, gw, `ol_gw`	`ol_gw`
`label`	An optional label identifying the boundary condition	String	Any String	Tampa Tide
`<wallghb>`	Indicates a constant or time series general head boundary condition assigned to a wall or walls			
`value`	The value of the coefficient for `<wallghb>` in eq. 7.2	Real	any real	0.046
`label`	An optional label identifying the boundary condition	String	Any String	Lake Level
The following sub-elements and attributes are available under both `<wallhead>` and `<wallghb>`.				
`<walllist>`	Indicates that the walls subject to the boundary condition are specified by a block of text containing pairs of nodes, each pair representing a wall			
A block of text within `<walllist>`.	Pairs of node ids	Long	Pairs of adjacent nodes	234625 456987 256354 458657
`<nodelist>`	Indicates that the walls subject to the boundary condition are specified in a block of text containing a list of contiguous nodes specifying a continuous wall			
A block of text within `<nodelist>`.	A list of node ids	Long	A series of adjacent nodes	235467 546575 345764 867456 654763
`label`	An optional label identifying the boundary condition	String	Any String	STA-3
Sub-elements available for specifying the method of interpolating constant or time series heads to apply to wall boundary conditions `<wallhead>` and `<wallghb>` are defined by the elements and attributes in Table 7.5				
Sub-elements available for specifying the head for the wall boundary conditions `<wallhead>` and `<wallghb>` are `<const>`, `<dss>`, `<asciiform>`, `<csv>`, and `<rc>`. These are described in detail in Chapter 9				

Table 7.3: *Elements and attributes used to define the* `<noflow>` *and* `<walluf>` *boundary conditions applied to walls. The elements are in shaded cells.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<noflow>`	Indicates that flow through a wall or walls will be blocked			
`section`	Specifies whether overland flow, groundwater flow or both are affected by the BC	String	ol, gw, `ol_gw`	`ol_gw`
`<walluf>`	Indicates a uniform flow boundary condition applied to a wall or walls			
`value`	The value of the coefficient for `slope` in eq. 7.3	Real	any real	0.014
The following sub-elements and attributes are available under both `<noflow>` and `<walluf>`				
`<walllist>`	Indicates that the walls subject to the boundary condition are specified by a block of text containing pairs of nodes, each pair representing a wall			
A block of text within `<walllist>`.	Pairs of node ids	Long	Pairs of adjacent nodes	234625 456987 256354 458657
`<nodelist>`	Indicates that the walls subject to the boundary condition are specified in a block of text containing a list of contiguous nodes specifying a continuous wall			
A block of text within `<nodelist>`	A list of node ids	Long	A series of adjacent nodes.	235467 546575 345764 867456 654763

Table 7.4: *Example XML input for 2-D boundary conditions.*

```
<mesh_bc>
   <wallhead section="ol_gw" label="STA-3">
      <walllist> 1 2 2 3 3 4 8 12 </walllist>
         <endpts>
           <entry id="1"
              <dss file="headbc.dss"
               pn="/HSE/T C03/HEAD/01JAN1994/15MIN/CALC/"> </dss>
           </entry>
           <entry id="2">
              <const value="13.6"> </const>
           </entry>
         </endpts>
   </wallhead>
   <cellghb id="4" value="0.15" label="Lake Isabel">
      <csv file="PondHead.csv" dbintl="1440"
      label="Head maintained in irrigation pond 3A" >
   </cellghb>
   <well  cellid="3" wellid="7658" label="Pump Station 8">
      <asciiform file="Pump8.dat" format="%10.2lf">
   </well>
   <wallghb value="0.064">
      <nodelist> 9 13 14 15 16 </nodelist>
      <wts2pts>
         <entry id="3">
            <dss file="Loc1GHB.dss"
            pn="/Loc1//Head/02FEB1994/15MIN//"> </dss>
         </entry>
         <entry id="4">
            <csv file="Loc2.dat dbintl="15" label="Loc2"> </csv>
         </entry>
         <wts> 1.0 0.2 0.8 0.0 </wts>
      </wts2pts>
   </wallghb>
   <cellhead id="7" bcid="42" label="Long Pond">
      <rc id="12> </rc>
   </cellhead>
   <noflow section="ol">
      <nodelist> 2 6 11 </nodelist>
   </noflow>
   <wallghb value="0.47">
      <nodelist> 1 5 9 </nodelist>
      <uniform>
         <const value="17.43"> </const>
      </uniform>
   </wallghb>
   <walluf value="-0.12">
     <nodelist> 12 16 </nodelist>
   </walluf>
<\mesh_bc>
```

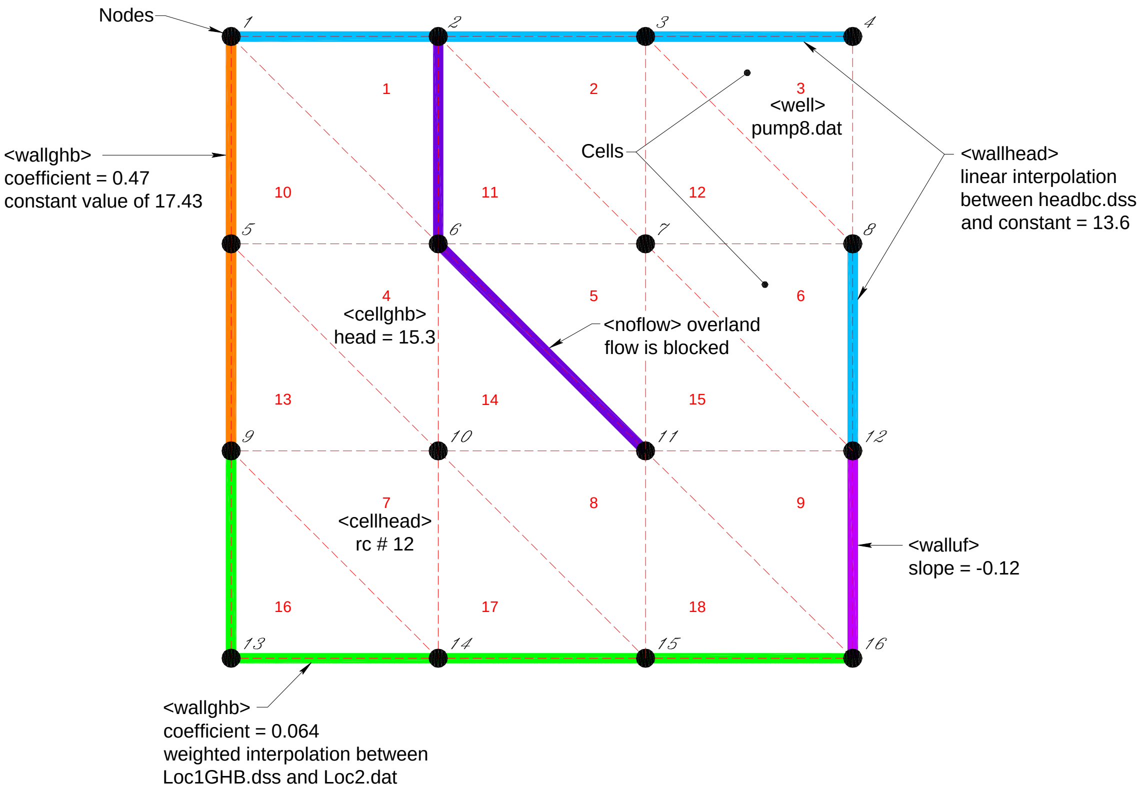


Figure 7.1: *Illustration of the application of 2D Mesh Boundary Conditions.*

7.1.2.1 Definition Of BC Location `<nodelist>` and `<walllist>`

Cell boundary conditions are applied to a single cell specified by the cell ID as demonstrated by the `<well>` discharge into cell 3 defined in Table 7.4 and shown in Figure 7.1. Wall boundary conditions often need to be applied to a large number of walls such as when a road blocks overland flow or one boundary of a model domain is subject to a tidal head. The location and application of a wall boundary condition to one or more walls is specified by pairs of nodes under the element `<walllist>`. Each pair defines a wall between cells or by a continuous series of nodes `<nodelist>` defining one or more adjacent walls, to which the boundary condition is applied. In the example XML input in Table 7.4 and shown below, the `<wallhead>` boundary condition applies to the walls designated by pairs of nodes $1-2$, $2-3$, $3-4$, and $8-12$ as shown in Figure 7.1. Note that the walls need not be contiguous. The `<noflow>` boundary condition for overland flow in Table 7.4 and below applies to the continuous wall defined by contiguous nodes 2, 6 and 11. The `<noflow>` and `<wallhead>` boundary conditions may be applied to overland flow, groundwater flow, or both as specified by the value assigned to `<section>` as described in Tables 7.2 and 7.3.

```
<mesh_bc>
   <wallhead section="ol_gw" label="STA-3">
      <walllist> 1 2 2 3 3 4 8 12 </walllist>
         <endpts>
           <entry id="1"
              <dss file="headbc.dss"
               pn="/HSE/T C03/HEAD/01JAN1994/15MIN/CALC/"> </dss>
           </entry>
           <entry id="2">
              <const value="13.6"> </const>
           </entry>
         </endpts>
   </wallhead>
   <noflow section="ol">
      <nodelist> 2 6 11 </nodelist>
   </noflow>
<\mesh_bc>
```

7.1.2.2 Defining The Type Of Interpolation Used For Wall Boundary Conditions

The `<wallghb>`, and `<wallhead>` boundary conditions may be assigned to walls based on a designated weighting pattern. The head value is assigned along the entire length of each wall, so the interpolation is from wall to wall and not within one wall. The available options for assigning weighting values of these boundary conditions to walls are specified in Table 7.5. `<uniform>` indicates that all walls have the boundary condition equally applied as shown by the application of a wall general head boundary condition `<wallghb>` to the wall defined by nodes 1, 5, and 9 as shown below and in Table 7.4 and Figure 7.1.

```
<mesh_bc>
   <wallghb value="0.47">
      <nodelist> 1 5 9 </nodelist>
      <uniform>
         <const value="17.43"> </const>
      </uniform>
   </wallghb>
 <\mesh_bc>
```

With the `<wts2pt>` option, a list of weights and a corresponding list of walls are specified. The list of walls may be specified under either `<nodelist>` or `<walllist>`. The values of the boundary condition at the first and the last wall in the list are specified under `<entry>` as a constant, time series, or a rating curve. The values at the other walls are determined by non-linear interpolation from the end wall values using the weighting specified under `<wts>`. The first and last entries under `<wts>` must be 1.0 and 0.0. In the example XML input below, the `<wallghb>` boundary condition at the wall defined by nodes 14 and

Table 7.5: *Elements and attributes used to assign interpolation weighting to the* `<wallhead>` *and* `<wallghb>` *boundary conditions. Element cells are shaded.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<wallhead>` or `<wallghb>`	Indicates a constant or time series wall general head or wall head boundary condition value assigned to a wall or walls			
The following interpolation specifications apply to both `<wallhead>` and `<wallghb>`				
`<uniform>`	Each wall is assigned the same value specified by `<const>` or by a time series			
`<wts2pt>`	Entries for interpolation will be specified			
`<wts>`	A sub-element under `<wts2pt>` that indicates the interpolation values will be specified in a block of text within the `<wts>` environment. The first weight must be 1.0 and the last 0.0			
A block of real numbers are inserted.	The interpolation weights assigned to the walls specified by `<nodelist>` or `<walllist>`. One entry for each wall	Real	$\geq$ 0.0 and $\leq$ 1.0	1.0 0.65 0.5 0.4 0.0
`<entry>`	Indicates that a constant or time series of head values will be specified. Two entries are required; one to specify the head for the first wall and one for the last			
`<id>`	The ID of the entry	Long	Any valid long	234
`<endpts>`	Entries for linear interpolation will be specified			
`<entry>`	Indicates that a constant or time series of head values will be specified. Two entries are required; one to specify the head for the first wall and one for the last			
`<id>`	The ID of the entry	Long	Any valid long	234

15 is 0.8*(the value in Loc1GHB.dss) + 0.2*(the value in Loc2.dat). An example of the use of `<wts2pt>` is shown below and in Table 7.4 for a `<wallghb>` boundary condition.

```
<mesh_bc>
   <wallghb value="0.064">
      <nodelist> 9 13 14 15 16 </nodelist>
      <wts2pts>
         <entry id="1">
            <dss file="Loc1GHB.dss"
            pn="/Loc1//Head/02FEB1994/15MIN//"> </dss>
         </entry>
         <entry id="2">
            <csv file="Loc2.dat dbintl="15" label="Loc2"> </csv>
         </entry>
         <wts> 1.0 0.2 0.8 0.0 </wts>
      </wts2pts>
   </wallghb>
 <\mesh_bc>
```

With the `<endpts>` option, the values of the boundary condition at the first and last wall are specified under `<entry>` as a constant, time series, or a rating curve. The values at the other walls are determined by linear interpolation from the end wall values. An example of the use of `<endpts>` is shown below and in Table 7.4 for the `<wallhead>` boundary condition. In the example below, the wall defined by nodes 2-3 has a head boundary condition applied that is `0.667*(the value in headbc.dss) + 0.333*(13.6)`.

```
<mesh_bc>
   <wallhead section="ol_gw" label="STA-3">
      <walllist> 1 2 2 3 3 4 8 12 </walllist>
         <endpts>
           <entry id="1"
              <dss file="headbc.dss"
               pn="/HSE/T C03/HEAD/01JAN1994/15MIN/CALC/"> </dss>
           </entry>
           <entry id="2">
              <const value="13.6"> </const>
           </entry>
         </endpts>
   </wallhead>
<\mesh_bc>
```

7.1.2.3 Time Series Data Format Used For Data Entry At Boundaries And Other Locations

A number of data formats are used to enter time series data into the model. These data formats may be used to specify data for boundary conditions and forcing functions applied to cells, segments, lakes, walls, water movers or similar objects. The formats available are `<dss>`, `<asciiform>`, `<csv>`, `<const>`, and `<rc>`. These formats and their use are described in detail in Chapter 9.

7.1.3 Boundary Condition Types Available For Walls

Various wall boundary conditions are described in this section. These include head, flow, general head, and no flow boundary conditions.

7.1.3.1 No Flow BC For Walls `<noflow>`

No flow boundary conditions prevent default 2-D overland and/or groundwater flow water movers from becoming effective. In other words, they remove the default water movers so there is no flow through a wall unless the user creates a replacement water mover. The no flow boundary conditions must be placed after the creation of overland flow and groundwater flow objects, but before adding user created flow objects since the addition of a `<noflow>` water mover removes existing water movers for the wall. The `<noflow>` element has an attribute `section` with a value `section="ol"` to block overland flow, `section="gw"` to block groundwater flow, or `section ="ol_gw"` to block both overland and groundwater flow. This XML structure is specified in Table 7.3 and demonstrated below and in Table 7.4 and Figure 7.1 where the overland flow through the walls defined by nodes 2, 6 and 11 is blocked.

```
<mesh_bc>
   <noflow section="ol">
      <nodelist> 2 6 11 </nodelist>
   </noflow>
<\mesh_bc>
```

7.1.3.2 Head BC For Walls `<wallhead>`

The head on a model domain boundary can be specified using a wall head `<wallhead>` boundary condition. The head at the wall is specified as a constant or a time variable value as indicated in Table 7.4 and equation 7.1. The head may be applied to groundwater flow, overland flow, or both.

This type of BC makes the most sense when only one wall of a cell is subjected to the BC. In the example in Table 7.4 and below, the heads for both overland and groundwater flow are specified at walls defined by pairs of nodes $1-2$, $2-3$, $3-4$, and $8-12$ with the head at each wall determined by linear interpolation between a time series at wall 1-2 and a constant value at wall $8 - 12$.

$$H_i = H_B(t) \tag{7.1}$$

```
<mesh_bc>
   <wallhead section="ol_gw" label="STA-3">
      <walllist> 1 2 2 3 3 4 8 12 </walllist>
         <endpts>
           <entry id="1"
              <dss file="headbc.dss"
               pn="/HSE/T C03/HEAD/01JAN1994/15MIN/CALC/"> </dss>
           </entry>
           <entry id="2">
              <const value="13.6"> </const>
           </entry>
         </endpts>
   </wallhead>
 <\mesh_bc>
```

7.1.3.3 General Head BC For Walls `<wallghb>`

The wall general head boundary `<wallghb>` is similar to the wall head boundary. This boundary condition specifies a constant or time series head. If the specified head is not equal to the head in the cell adjacent to the wall, water flows through the wall to or from the cell as computed in Equation 7.2

$$Q_i(t) = K_B(H_B(t) - H_i) \tag{7.2}$$

where
H_i = head in the cell,
H_B = specified boundary condition head, and
K_B = a coefficient that controls the rate of flow through the wall.

Only one general head BC should be applied per cell, because it essentially modifies the source term. It can be applied only to walls on the boundary of the model domain.

In the XML example in Table 7.4 and the example below, a wall general head boundary condition is applied to walls defined by the `<nodelist>` 9 13 14 15 16. The flow coefficient in Equation 7.2 is 0.064. The specified head is interpolated according to the specified weights between two time series, one in a DSS file and one in a csv file.

```
<mesh_bc>
   <wallghb value="0.064">
      <nodelist> 9 13 14 15 16 </nodelist>
      <wts2pts>
         <entry id="3">
```

```
            <dss file="Loc1GHB.dss"
            pn="/Loc1//Head/02FEB1994/15MIN//"> </dss>
         </entry>
         <entry id="4">
            <csv file="Loc2.dat dbintl="15" label="Loc2"> </csv>
         </entry>
         <wts> 1.0 0.2 0.8 0.0 </wts>
      </wts2pts>
   </wallghb>
 <\mesh_bc>
```

7.1.3.4 Uniform Flow BC For Walls `<walluf>`

A uniform flow boundary is defined by assuming that there is uniform overland flow that discharges water through the boundary wall. It can be defined by specifying a slope of S_B at the boundary cell, with a resulting flow of Q_B as in Equation 7.3.

$$Q_B = K_i S_B \tag{7.3}$$

in which, K_i is the conveyance of the cell. A positive slope yields a uniform flow into the cell from beyond the boundary and a negative slope yields flow from the cell across the boundary out of the model domain. If the cell becomes dry, there will be no flow. An example of XML input for uniform flow is shown below and included as the last boundary condition in Table 7.4 which specifies a uniform flow out of the cell bordered by wall $12-16$ with a slope of -0.12.

```
<mesh_bc>
   <walluf value="-0.12">
     <nodelist> 12 16 </nodelist>
   </walluf>
<\mesh_bc>
```

7.1.4 Boundary Condition Types For Cells

Various cell boundary conditions are described in this section. These include, flow, head and general head options.

7.1.4.1 Inflow BC `<well>`

A commonly used upstream boundary condition is an inflow boundary condition defined as

$$Q_i = Q_B(t) \tag{7.4}$$

where
i represents the cell ID, and
$Q_B(t)$ = constant value or a time series of flow.
Inflow into the model is generally described using the inflow boundary condition. More than one of these boundary conditions can be applied at any water body. When this happens, the flow is considered additive. This boundary condition may also be used to define a constant or time series of flow out of a cell, although this application must be used with care and a clear understanding of the physical process being represented.
The attributes and subelements of `<well>` are defined in Table 7.1. An example of a time series of pumping rates into cell 3 is demonstrated below, in Table 7.4, and Figure 7.1. The flow is defined by a time series in an asciiform format.

```
<mesh_bc>
   <well  cellid="3" wellid="7658" label="Pump Station 8">
      <asciiform file="Pump8.dat" format="%10.2lf">
   </well>
<\mesh_bc>
```

The user has the option to specify a volume using the following:

```
<const value = "1000" type="PER-CUM" dbintl="43200"/>
```

This specifies 1000 units of volume is distributed uniformly within each month. The model determines the volume to be applied in each time step and converts this to a rate.

7.1.4.2 Head Boundary Conditions For Cells `<cellhead>`

This boundary condition forces the cell to the specified head at each model time step where the head may be a constant or specified by a time series. The elements and attributes required to specify this boundary condition are listed in Table 7.1. This boundary condition should be used only if there is no other choice, and then only with a clear understanding of the implications. Physically there may be a net flow of water to or from the cell without the water level in the cell changing. Computationally, if the head at cell n is specified, the corresponding row in the solution matrix is set to zero except for the diagonal element which is set equal to 1.0. In addition the corresponding entry in the source vector is set equal to the difference between the specified head and the present head in the cell. Both physically and computationally, this has the effect of adding or subtracting water from the model domain without accounting for it. The water budget, therefore, will be incorrectly computed. The XML input below and in Table 7.4 specifies that the head in cell 7 is varied according to the values generated by rating curve 12.

```
<mesh_bc>
   <cellhead id="7" bcid="42" label="Long Pond">
      <rc id="12> </rc>
   </cellhead>
<\mesh_bc>
```

7.1.4.3 General Head Boundary Conditions For Cells `<cellghb>`

The general head BC for a cell is similar to the `<cellhead>` boundary condition but it preserves the accuracy of the water budget. A constant or time series head is specified along with a constant that controls the rate at which water flows into or out of the cell to approach the specified head. The flow is calculated as

$$Q_i = K_B(H_B(t) - H_i) \tag{7.5}$$

where
Q_i = the discharge into the cell,
H_i = head in the cell i,
$H_B(t)$ = head specified by the boundary condition, and
K_B = a constant that controls the flow into or out of the cell as defined by the `<cellghb>` attribute `<value>`.

The flow computed by Equation 7.5 is added to or subtracted from an imaginary water body that holds all boundary condition flows except `<cellhead>`. This water body is included in the water budget calculations. An example of the XML input for a `<cellghb>` boundary condition is shown below and included in Table 7.4. In this example water flows into or out of cell 4 at a rate determined by the current head in the cell, the coefficient ”value”=0.15, and the head specified by the time series in the file PondHead.csv.

```
<mesh_bc>
   <cellghb id="4" value="0.15" label="Lake Isabel">
      <csv file="PondHead.csv" dbintl="1440"
      label="Head maintained in irrigation pond 3A" >
   </cellghb>
<\mesh_bc>
```

7.2 Canal Network `<network_bc>`

As in the case of overland flow, solution of the canal flow equations depends on the initial and boundary conditions, in addition to the governing equation itself. In contrast to the flow solution described by the full St. Venant equations, only one boundary condition can be applied at each boundary in the case of 1-D diffusion flow. Both head and flow boundary conditions are commonly used. They can be applied to canal segments or junctions, however, segment head boundary conditions should be avoided to maintain the integrity of water budget formulations.

Flow BC's are generally applied to segments. The no-flow boundary condition applied between two segments is a special case intended to prevent flow between adjacent segments when a structure is modeled. Tables 7.6 and 7.7 below show the elements and attributes available for specifying the boundary conditions for canals. Boundary conditions for the 1-D canal networks are defined under the `<network_bc>` element within the `<network>` environment. The boundary conditions specified in the environments `<segmentsource>`, `<segmenthead>`, `<junctionhead>`, and `<segmentghb>` have flows or heads specified by a constant, a time series, or a rating curve under the sub-elements `<const>`, `<dss>`, `<asciiform>`, `<csv>`, and `<rc>`. Details on the use of these options are explained in Chapter 9.

Sample XML input for all six network boundary conditions under the `<network_bc>` element is shown in Table 7.8 and demonstrated in Figure 7.2. Each available boundary condition will be explained in detail in the following sections.

7.2.1 Flow Boundary Condition `<segmentsource>`

The `<segmentsource>` boundary condition is often used at the upstream end of a canal. It is similar to the `<well>` boundary condition for a cell. The user may specify an inflow or outflow from a canal segment according to the following equation.

$$Q_i = Q_B(t) \tag{7.6}$$

Where i represents the segment ID and $Q_B(t)$ = a constant, rating curve, or time series flow. Methods specifying $Q_B(t)$ are explained in detail in Chapter 9. The XML input below excerpted from Table 7.8 specifies a flow into segment 1 defined by a time series in the file "DevilFlow.dss". The application of this BC is shown graphically in Figure 7.2. The multiplier 0.0283 converts the input from cfs to m^3/sec. This boundary condition may also be used to specify an outflow from a segment by a time series of negative numbers in the DSS file or by specifying a negative multiplier.

Table 7.6: *Elements and Attributes for Specifying Boundary Conditions for Canal Networks Part 1. Element cells are shaded.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<segmentsource>`	Used to specify the inflow to a segment as a constant or a time series			
`id`	ID of the segment	Long	10000-20000	156324
`bcid`	Optional BC ID	Long	> 0	23
`label`	Optional label to identify the BC	String	Any String	Lake Kahuna
`<segmenthead>`	Used to specify the water level at a segment as a constant or a time series. This BC can corrupt the water budgets.			
`id`	ID of the segment	Long	10000-20000	164824
`bcid`	Optional BC ID	Long	> 0	23
`label`	Optional label to identify the BC	String	Any String	Pond 3B
`<segmentghb>`	Used to specify the water level at a segment as a general head BC. The head can be a constant, a rating curve, or a time series.			
`id`	ID of the segment	Long	10000-20000	164824
`kcoeff`	Coefficient in equation 7.9	Real	≥ 0.0	0.12
`bcid`	Optional BC ID	Long	> 0	23
`label`	Optional label to identify the BC	String	Any String	Pond 3B
`<junctionhead>`	The head at a junction is specified as a constant, rating curve, or a time series.			
`id`	ID of the segment to or from which flow occurs	Long	10000-20000	164824
`bcid`	Optional BC ID	Long	> 0	23
`label`	Optional label to identify the BC	String	Any String	Pond 3B
Sub-elements available for specifying the head for the canal boundary conditions `<segmentsource>`, `<segmenthead>`, `<segmentghb>`, and `<junctionhead>` are `<const>`, `<dss>`, `<asciiform>`, `<csv>`, and `<rc>`. These are described in detail in Chapter 9				

Table 7.7: *Elements and Attributes for Specifying Boundary Conditions for Canal Networks Part 2. Element cells are shaded.*

Element or Attribute	Definition	Variable type	Suggested range	Example
`<uniformflow>`	Used to specify the canal segment at the end of a network as having uniform flow. Not implemented in the current version of the model.			
`<junctionblock>`	The default water mover between two segments is removed. This is needed if a structure is to be placed at the junction.			
`id1`	ID of the segment on one side of the junction	Long	10000-20000	164824
`id2`	ID of the segment on the other side of the junction	Long	10000-20000	164824

```
<network>
   <network_bc>
     <segmentsource id="1" bcid="35" label="Inflow from Dirty Devil River">
        <dss file="DevilFlow.dss" pn="/HSE/T C03/HEAD/01JAN1994/15MIN/CALC/"
                  mult="0.0283" dbintl="15" units="cfs" >
        </dss>
     <segmentsource>
  <network_bc>
</network>
```

7.2.1.1 Head Boundary Condition `<segmenthead>`

The `<segmenthead>` boundary condition is similar to the `<cellhead>` boundary condition for the 2D mesh. This boundary condition type can be used to specify the water level in a canal segment at the model domain boundary. An example would be as an upstream boundary condition for a canal that drains water from a large lake. The head in the canal segment is specified as shown in equation 7.7.

$$H_i = H_B(t) \tag{7.7}$$

where $H_B(t)$ can be a time series, a constant, or rating curve. As with the `<cellhead>` boundary condition, `<segmenthead>` can introduce errors in the water budget calculations. This boundary condition type modifies the solution matrix by setting all entries in the row corresponding to the segment number equal to 0.0 except for the diagonal term which is

Table 7.8: *Example XML input canal network boundary conditions.*

```
<network>
   <network_bc>
      <segmentsource id="1" bcid="35" label="Inflow from Dirty Devil River">
         <dss file="DevilFlow.dss"
                  pn="/HSE/T C03/HEAD/01JAN1994/15MIN/CALC/"
                  mult="0.0283" units="cfs" >
         </dss>
      <segmentsource>
      </segmenthead id="11" bcid="22" label="Big Lake">
         <asciiform file="BigLake.dat" format="%10.2lf">
         </asciiform>
      </segmenthead>
      <junctionhead id="5" bcid="7" label="Downstream Tide">
         <rc id="17">
         </rc>
      </junction>
      <segmentghb id="4" kcoef="0.045" bcid="11" label="Lake Lola">
         <csv file="Lola.csv" dbintl="15" label="Lola">
         </csv>
      </segmentghb>
      <uniformflow id1="8" id2="??" bcid="19" slope="0.006">
      </uniformflow>
      <junctionblock id1="9" id2="10" label="Highway 66 culvert">
      </junctionblock>
   </network_bc>
</network>
```

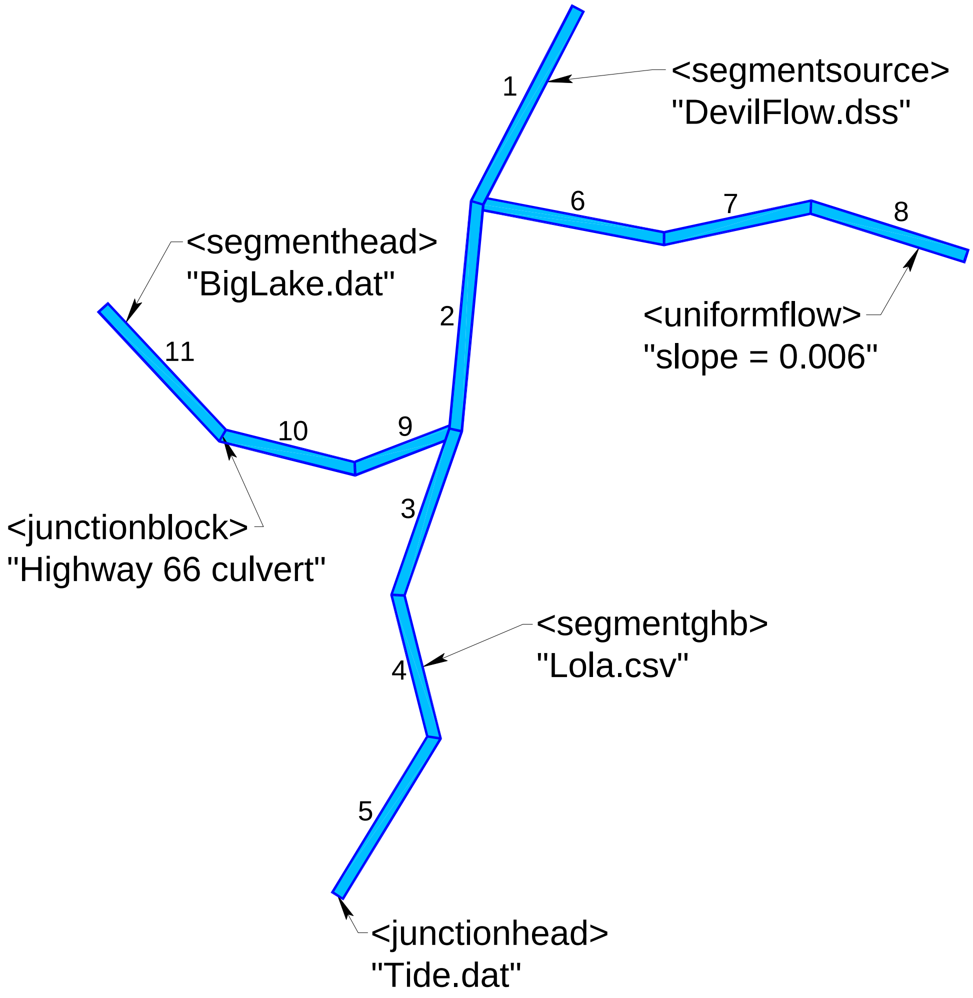


Figure 7.2: *Illustration of the application of Canal Network Boundary Conditions.*

set equal to 1.0. The corresponding entry in the source vector is set equal to the difference between specified and existing head in the segment. This allows water to flow into or out of the segment subject to the head boundary condition without changing the volume of water in the segment. If possible, use the `<segmentghb>` or `<junctionhead>` boundary condition instead, as they preserve the water budget. In the example below, the head in a segment is set equal to a time series of water levels in a large lake that drains into the canal.

```
....
   <network_bc>
     </segmenthead id="11" bcid="22" label="Big Lake">
        <asciiform file="BigLake.dat" format="%10.2lf">
        </asciiform>
     </segmenthead>
```

```
    </network_bc>
....
```

7.2.1.2 Installing A No-Flow Boundary Condition At Canal Junctions `<junctionblock>`

If two canal segments are adjacent such as segments 10 and 11 in Figure 7.2 a default water mover is constructed during the network set up to move water between the segments. In the case that this flow is physically replaced by a structure such as a culvert under a road that intersects the canal, it is necessary to remove the default water mover before replacing it with a user created water mover as described in Section 6.5.3. The `<junctionblock>` boundary condition removes any existing water movers between the segments and prevents flow unless another water mover is created. The example below shows the use of the `<junctionblock>` boundary condition to remove the default water mover where a highway crosses the canal.

```
 ....
    <network_bc>
      <junctionblock id1="10" id2="11" label="Highway 66 culvert">
      </junctionblock>
...
    <network_bc>
```

7.2.1.3 Uniform Flow In A Segment `<uniformflow>`

This boundary condition is not implemented in the current version of the model.

A common boundary condition for simulating flow in a river or canal is that of uniform flow at the downstream end. This is common practice in modeling backwater profiles with models such as HEC-RAS. Uniform flow is computed as a function of the canal geometry, roughness, and slope. The general expression for uniform flow is

$$\frac{\partial H}{\partial x} = S_0 \tag{7.8}$$

in which S_0 is the uniform flow slope assigned to the segment.

7.2.1.4 General Head Boundary Condition In A Segment `<segmentghb>`

This boundary condition is similar to the mesh boundary condition `<cellghb>`. It specifies a head as a constant, time series or a rating curve. Water flows into or out of the segment in a way as to tend toward the specified head according to the equation;

$$Q_i(t) = K_B(H_B(t) - H_i) \tag{7.9}$$

where
K_B = a user specified coefficient,
$H_B(t)$ = the specified head, and
H_i = the head in the target segment.

The following example specifies a general head condition in segment 4 with the specified head as a time series in the file Lola.csv.

```
 ....
<network>
   <network_bc>
       <segmentghb id="4" kcoef="0.045" bcid="11" label="Lake Lola">
         <csv file="Lola.csv" dbintl="15" label="Lola">
         </csv>
      </segmentghb>
   </network_bc>
</network>
....
```

7.2.1.5 Junction Head Boundary Condition `<junctionhead>`

The `<junctionhead>` boundary condition is a method for specifying the head at a junction adjacent to a canal segment. The flow into or out of the segment from or to an imaginary waterbody outside the model domain is given by Equation 7.10.

$$Q = \frac{A_i}{n} {R_i}^{2/3} (H_B - H_i) \tag{7.10}$$

where
A_i = the cross-sectional area of the segment,
n = the Manning's roughness coefficient,
R_i = the hydraulic radius, and
H_B and H_i = the specified boundary head and the head in the designated segment.

This boundary condition is appropriate for specifying the head at the end of a canal. In the example below the flow to or from segment 5 is determined by the canal geometry and roughness, and the head difference between the head specified in the file Tide.dat and the head in canal segment 5.

```
 ....
<network>
```

```
   <network_bc>
      </segmenthead id="5" bcid="7" label="Downstream Tide">
         <csv file="Tide.dat" dbintl="15" label"Tide">
         </csv>
      </segmenthead>
   </network_bc>
</network>
....
```

7.3 Boundary Conditions For General Water Bodies

Some of the same boundary conditions defined for the 2-D mesh or network boundary conditions can be defined using generic water body boundary conditions. They are more powerful because they can be applied equally to cells, segments and lakes. The basic categories are listed in Table 7.9 and described in detail in the following sections. Sample XML input for these boundary conditions is displayed in Table 7.10. Note that these boundary conditions are specified within the `<watermover>` environment.

7.3.1 Sources And Sinks `<source>`

The `<source>` boundary condition performs the same operation as the `<segmentsource>` and `<well>` boundary conditions, but it can be applied to any water body. The inflow to or outflow from a water body is defined as

$$Q_i = Q_B(t) \tag{7.11}$$

where $i =$ the water body ID,
$Q_B(t) =$ constant, rating curve, or time series flow.

Methods specifying $Q_B(t)$ are explained in detail in Chapter 9. In the example in Table 7.10 a constant flow of 78.0 cfs flows into Walden Pond (water body 34) with a database interval of 15 minutes. The multiplier of 0.0283 will convert the flow from cfs to $m^3/sec.$

7.3.2 Boundary Conditions Based On Stage-Discharge Relationships `<hq_relation>`

A boundary condition described using a stage-discharge relationship for a water body i can be expressed mathematically as

$$Q_i = f(H_i) \tag{7.12}$$

where
$Q_i =$ discharge into water body i,
$H_i =$ stage of water body i,
$f(H_i)$ represents a function of H_i.

The effect would be an inflow into water body i as described by the function. The example in Table 7.10 describes inflow into Pond 3D as a function of head in the pond. The flow is regulated so as to keep the water level in the pond at about 520 with the flow rapidly reduced as the water level increases from 500 to 520.

Table 7.9: *Elements and Attributes for Specifying Boundary Conditions for General Water Bodies in the* `<watermover>` *environment. Element cells are shaded.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<source>`	Used to specify the inflow to or outflow from a water body as a constant, a time series, or a rating curve.			
`id`	ID of the water body	Long	any valid water body ID	187324
`label`	Optional label to identify the BC	String	Any String	L3 irrigation pond
`<hq_relation>`	The flow into or out of a water body is determined by a 1D lookup table.			
`id`	ID of the water body	Long	10000-20000	164824
`wmid`	Water mover ID	Long	> 0	23
`mult`	Multiplier for the 1D lookup values, often for unit conversion	Real	Any real	0.0328
`<hq>`	The lookup table is included as text in the `hq` environment. The data are entered as two columns with stage in the first column and discharge in the second.			
The `<source>` element has the following sub-elements available for specifying the flow or head: `<const>`, `<dss>`, `<asciiform>`, `<csv>`, and `<rc>`. These elements and their attributes are described in detail in Chapter 9				

Table 7.10: *Example XML input for general water body boundary conditions.*

```
 <watermovers>
  <source id="34" label="Walden Pond">
      <const value="78.0" dbintl="15" mult="0.0283"> </const>
  </source>
  <hq_relation id="2" mult="0.5" label="Pond 3D">
    <hq>
      400.0 50000.0
      500.0 40000.0
      520.0   10.0
      530.0    0.0
    </hq>
  </hq_relation>
</watermovers>
```

7.4 Boundary Conditions For Lakes `<lake_bc>`

Boundary conditions for lakes are defined under the `<lake_bc>` element in the `<lakes>` environment. There are two boundary conditions available, `<lakesource>` and `<owet>`. The elements and attributes are listed in Table 7.11 and described in detail in the following sections. Sample XML input for these boundary conditions is displayed in Table 7.12.

7.4.1 Sources And Sinks `<lakesource>`

The `<lakesource>` boundary condition performs the same operation as the `<well>`, `<source>`, and `<segmentsource>` boundary conditions, but it can be applied only to lakes. The inflow to or outflow from a lake is defined as

$$Q_i = Q_B(t) \tag{7.13}$$

where
i = represents the lake ID, and
$Q_B(t)$ = constant, rating curve, or time series flow.

Methods specifying $Q_B(t)$ are explained in detail in Chapter 9. In the example in Table 7.12 a time series flow specified in STA17Pump.dss flows into STA17 (lake id = 237345) with a database interval of 1 day.

7.4.2 Open Water Evaporation Boundary Condition `<owet>`

This boundary condition removes water from a lake according to the equation

$$Q_i = Area(H_i)RefET(t) \tag{7.14}$$

where
Q_i = evaporation rate from the lake i,
H_i = water level in the lake i,
$Area(H_i)$ = the lake surface area interpolated from a 1D lookup table, and
$RefET(t)$ is the potential evaporation defined as a constant, a rating curve, or a time series.

The example in Table 7.12 describes evaporation from lake STA17 as a function of head in the lake and the specified RefET. The surface area of the lake is read from the 1D lookup table `<sa>` with the area increasing from 0 to 500 as the head increases from 5.0 to 25.0 meters. RefEt is specified as a time series in the file STA17Evap.csv.

Table 7.11: *Elements and Attributes for Specifying Boundary Conditions for Lakes. Element cells are shaded.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<lakesource>`	Used to specify the inflow to or outflow from a lake as a constant, a time series, or a rating curve.			
`lakeID`	ID of the lake	Long	any valid lake ID	237345
`id`	Boundary condition ID	Long	Any integer	17
`label`	Optional label to identify the BC	String	Any String	FPL pump 4
`<owet>`	The ET from the surface of a lake is specified as a function of the surface area of the lake.			
`<sa>`	A 1D lookup table is included as text in the `<sa>` environment. The data are entered as two columns with stage in the first column and surface area in the second.			
`lakeID`	ID of the lake	Long	10000-20000	254824
`id`	Boundary Condition ID	Long	> 0	31
`label`	Optional label to identify the BC	String	Any String	Measured ET
The `<lakesource>` and `<owet>` elements have the following sub-elements available for specifying the flow or RefET: `<const>`, `<dss>`, `<asciiform>`, `<csv>`, and `<rc>`. These elements and their attributes are described in detail in Chapter 9				

Table 7.12: *Example XML input for lake body boundary conditions.*

```
<lakes>
  <lakesource lakeID="237345" id="17" label="STA17">
      <dss file="STA17Pump.dss pn="/STA/17/FLOW/08MAR2003/1DAY// </dss>
  </lakesource>
  <owet lakeID="254824" id="31" label="STA17ET">
    <sa>
       5.0    0.0
      10.0   50.0
      15.0  100.0
      25.0  500.0
    </sa>
    <csv file="STA17Evap.csv" dbintl="1440">
    </csv>
  </owet>
</lakes>
```

Chapter 8

Pseudocell Approach and Models

Pseudocells provide a method to simulate the local surface hydrology in a mesh cell or a collection of mesh cells. The mesh cells are used in the implicit finite volume solution for the regional flow while the pseudocells explicitly simulate the local hydrology before the next time step of the implicit regional solution. Pseudocell types available are designed to simulate

1. Unsaturated flow in soil
2. Interception and detention of flow
3. Interflow, field drainage
4. Urban hydrology and related management practices
5. Rainfall-Runoff simulation
6. Agricultural irrigation and drainage practices
7. Everglades ridge and slough hydrology
8. Small creek and tributary flow
9. Discharge, seepage and aquifer recharge from detention and retention ponds

Two different pseudocell types are illustrated in Figure 8.1. The land within a cell can be represented as a simple wetland with primary processes of rainfall, evapotranspiration and change in water table depth, or the hydrology may be more complex as represented by the `<hub>` pseudocell that simulates irrigation and consumptive use as well as natural hydrologic processes.

8.1 PseudoCell Overview

Pseudocells provide the method for simulating the local surface hydrology for each mesh cell. Where the mesh cells are used in the implicit finite volume solution for the regional flow, the pseudocells are used to simulate the hydrology within a mesh cell or between a few mesh cells explicitly before the implicit solution is executed.

The primary function of pseudocells is to provide recharge to the mesh cells as a function of rainfall, evapotranspiration and runoff. Without pseudocells, there is no way for rainfall and evapotranspiration to supply water to the mesh cells of the SFRSM. Pseudocells can simulate vertical flow in the soil and complex flow in the landscape including surface runoff, interflow, baseflow, and water management systems including stormwater routing and detention/retention ponds. It is possible to combine simple pseudocell types to model complex landscapes within a single mesh cell or a collection of mesh cells.

The unconditional stability of the implicit Finite Volume solution allows the implementation of a variety of pseudocells throughout the mesh as necessary to best reflect the characteristics of the landscape. It is possible to select agricultural pseudocells adjacent to urban.

In this chapter we will present pseudocell concepts and the types of pseudocells that can be uses in the SFRSM. The pseudocell types include simple types and complex pseudocells that are a combination of simpler pseudocells. Benchmarks, which are used to document the functionality of the pseudocells will be discussed as well as input XMLs. Although the algorithms are presented for each pseudocell, the computations are not included in This User's Manual.

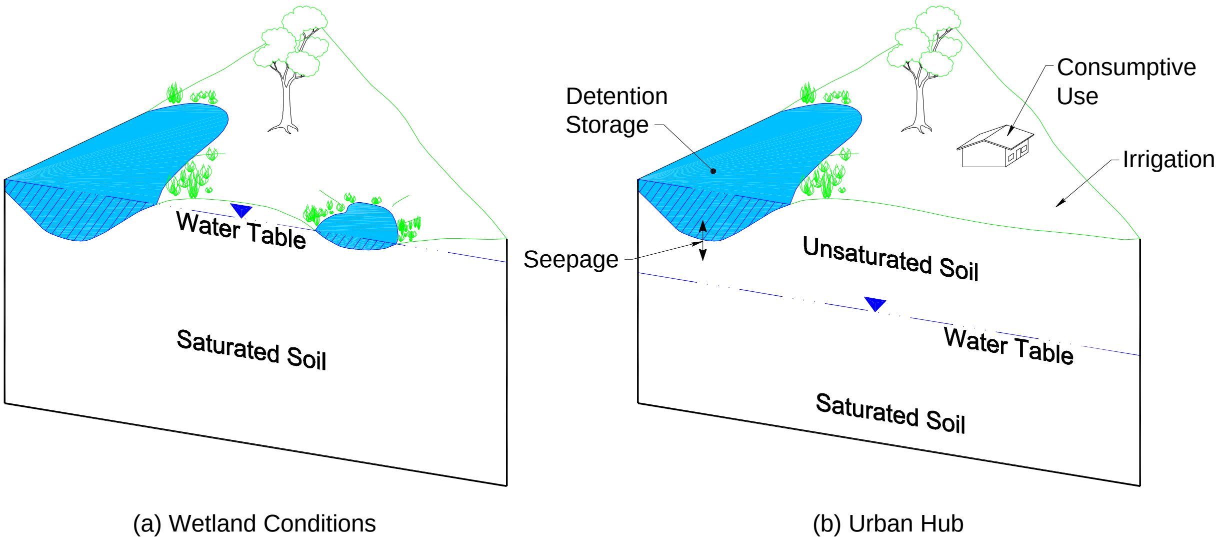
Benchmarks

The RSM benchmarks are an important component of the RSM documentation and are described in Section 11.1. The benchmarks provide an example of the functionality of each aspect of the SFRSM in a standard framework. The framework is a standard domain (see Figure 8.2) with consistent features for which single features are changed in each benchmark. Each benchmark consists of a 3x3 square domain 1500 meters on a side divided into 18 cells. Each benchmark provides the characteristic XML input and an example of the output. The typical output is a DSS file although a few benchmarks produce comma delimited files. The DSS files are viewed using the dssviewer software. There are also more complex benchmarks that demonstrate accepted application of RSM, such as BM18 which an implementation of RSM to the L8 basin. As additional functionality or implementation of pseudocells is developed, additional benchmarks will be added to the collection.

Each pseudocell is demonstrated in a benchmark and these are provided with each pseudocell description. These benchmarks are identified with each pseudocell type. The benchmark provide typical values for the pseudocell type. These benchmarks provide a test bed for the evaluation of the functionality of each pseudocell, conduct user defined sensitivity analysis and compare different pseudocells types.

Limitations of Pseudocells

Pseudocells are the means by which local hydrology is simulated in the RSM. They contribute to the governing equations primarily through the source term. Furthermore, the contributions are assumed as explicit for convenience and stability. Since pseudocells are open for user development, the explicit option is the only one available, as this protects the user from making the equations stiff, or unstable. Pseudocells are also assumed to be non-interactive, or interactive in only a linear way. This preserves the stability of the regional solution and and the reliability of the water budgets.



(a) Wetland Conditions

(b) Urban Hub

Figure 8.1: *Schematic representation of two pseudocell types (a) Wetland and (b) Hub.*

8.1.1 Pseudocell Water Budgets

Water budgets within pseudocells must be maintained carefully to maintain mass balance within the RSM model. Considering that pseudocells have a number of standard interfaces such as rainfall and ET, and a number of water storages that are independent of the mesh such as the unsaturated zone and impoundments, a basic expression can be written to summarize interface flows and storage mechanisms. Equation 8.1 is a mass balance expression that can be applied to any pseudocell.

$$R_{rech} = P - ET + I - \frac{dUs}{dt} - \frac{dD}{dt} - RO \tag{8.1}$$

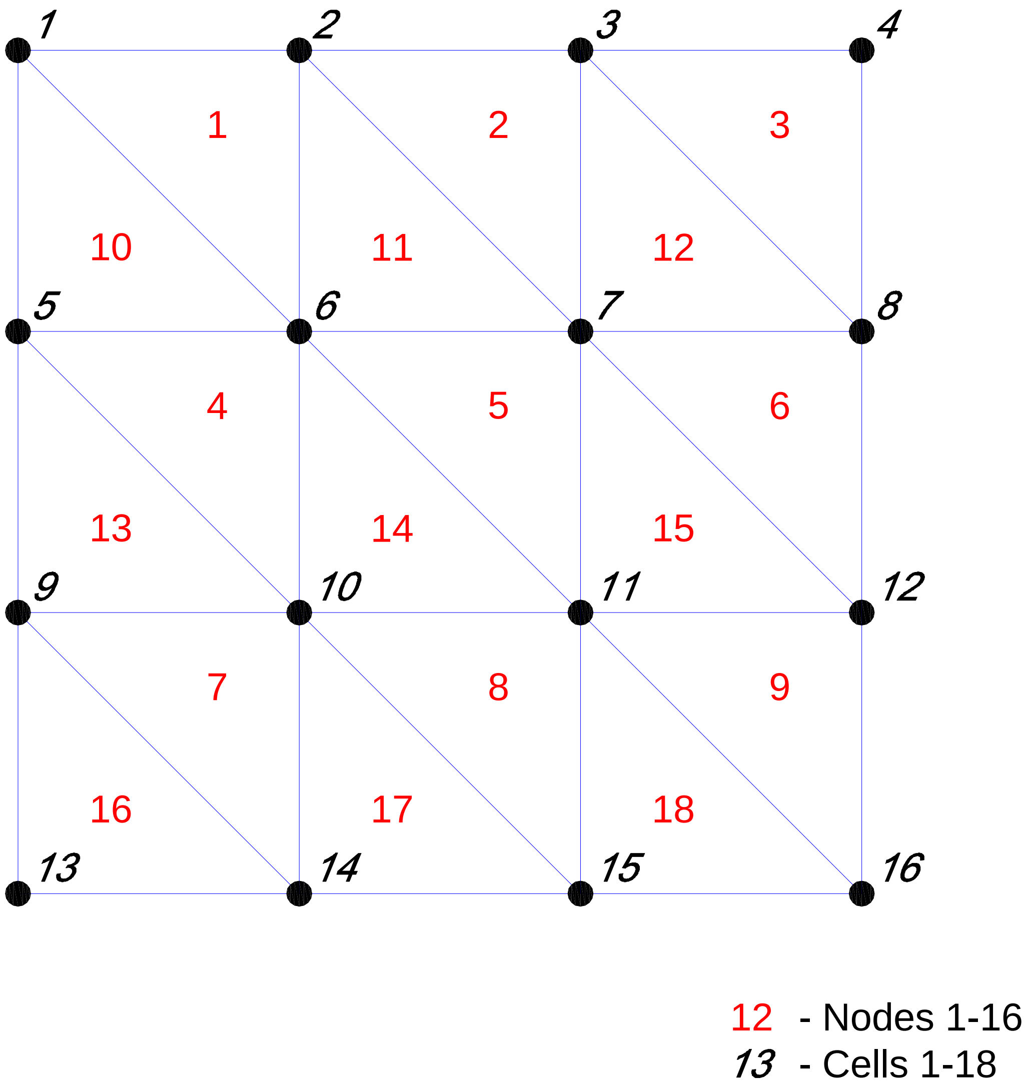


Figure 8.2: *Standard Grid for Benchmarks.*

where,

R_{rech} = recharge computed as a depth/unit time entering into the cell as a source term;
P = precipitation depth/unit time;
ET = evapotranspiration depth/unit time;
I = irrigation depth/unit time;
Us = summation of unsaturated zone moisture as a depth;
D = summation of detention water as a depth.
RO = surface runoff leaving the pseudocell as a depth/unit time

8.1.2 Partitioning Of Pseudocell Water

An actual pseudocell however can carry out a number of additional functions to compute local hydrologic flows beyond the simple hydrology represented by equation 8.1. One of the functions is the computation of routed water into other specified pseudocells. Pseudocells also need additional methods to exchange data with the regional solution in the form of a standard interface so that third party vendors can write pseudocell models that can be readily included in the RSM. A pseudocell socket provides these methods as access functions. An arbitrary pseudocell and the support socket are illustrated in Figure 8.3. The following equation shows how mass balance in the pseudocell is maintained when using some of these access functions

$$S_{new} = S_{init} + P - ET + DS + Q_{watersupply} - Q_{runoff} - Q_{seepage} + Q_{irrigation} - Q_{septic} - Q_{recharge} \tag{8.2}$$

where S_{init} and S_{new} are the volumes of water in the pseudocells at the beginning and the end of the time step respectively. All the values are computed as depths, or volumes per cell area.

Definitions of the terms in Equation 8.2 and the access functions that could be used in the mass balance accounting for a pseudocell following this equation are listed in Table 8.1. There are additional access functions available for pseudocells to exchange data with the regional solution.

A new pseudocell would be incorporated by defining a new class as a derived class from the pseudocell base class. The new pseudocell would consist of a model to simulate local hydrology with data obtained from the regional model, with the results of the simulation of the local hydrology included in the regional solution by use of the access functions. This description is not intended to be detailed enough to allow a new user to add a new pseudocell type to the model, but to describe the flexibility allowed by the object oriented programming approach. Adding a new pseudocell would require access to the RSM code and consultation with one or more of the RSM development team.

8.1.3 Pseudocell Types

The area simulated by the RSM may include native lands as well as developed land, both urban and agricultural. The natural areas are represented by hydrologic processes only slightly impacted by human activities. These include much of the everglades and other swamps and protected upland areas. Large parts of the model domain such as the EAA, South Dade County, and the Caloosahatchie basin are predominantly agricultural with intensive cultivation and water management including irrigation. Agricultural areas include constructed

Table 8.1: *Some of the standard access functions provided by a pseudocell socket.*

Variable	Function	Description
P	Rain()	Rainfall during the time step
ET	Et()	Evapotranspiration during the time step
DS	CellDelta()	Water added from the regional system during the time step
$Q_{watersupply}$	WSupply()	Water supply to the pseudocell that can originate from the home cell or from another location
Q_{runoff}	Runoff()	Overland flow runoff, that can be directed into a specified location
$Q_{seepage}$	Seepage()	Seepage of water into the home cell. This seepage can come from small detention and retention ponds or other flow mechanisms
Q_{irrig}	Irr()	Water added to the pseudocell for irrigation
Q_{rech}	Recharge()	Water added to the underlying mesh cell as recharge

drainage ditches to capture field drainage and impoundments to hold water and release it at a controlled rate. Other regions, particularly along Florida's east coast, are highly urbanized with large areas of impervious surfaces, constructed drainage systems, and retention/detention ponds. The pseudocells currently in the RSM have been designed to simulate each of these types of areas. There is also a pseudocell that does nothing that is useful for turning off hydrologic processes in selected areas for model testing and development.

In the following discussion, pseudocell descriptions will be broadly classified into these three general types. The term in brackets such as `<layer1nsm>` is the XML element that creates the environment for specifying the attributes of the pseudocell.

1. Natural System Pseudocells
    - Natural Wetland System `<layer1nsm>`
    - No Action `<layerpc>`
    - Precipitation Runoff Routing `<prr>`
    - Five Unsaturated Soil Layer `<layer5>`
    - Unsaturated Soil `<unsat>`
2. Agricultural Pseudocells
    - Agricultural Irrigation Requirements `<afsirs>`
    - Drainage Collector Ditch `<pumpedditch>`

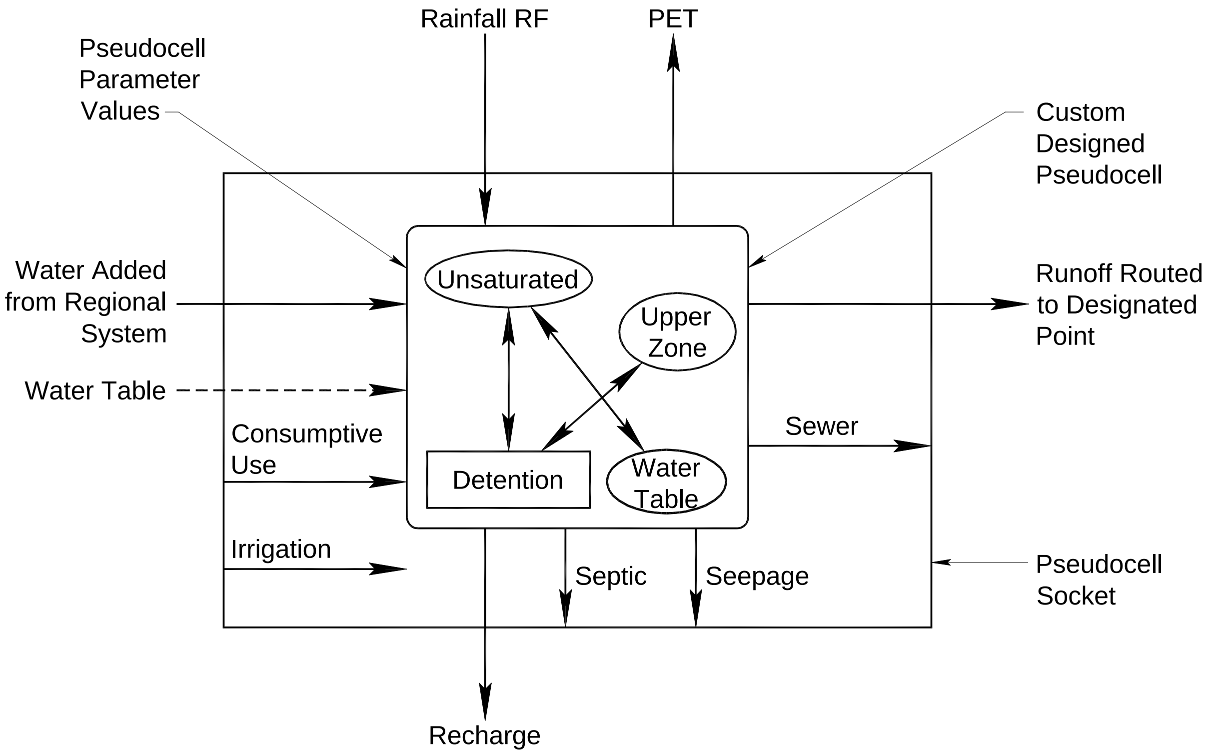


Figure 8.3: *Pseudocell support socket for addition of custom designed pseudocells. Five components link pseudocells to the mesh: Rain, PET, Water Added, Runoff, and Recharge.*

Agricultural Impoundment `<agimp>`

3. Urban Pseudocells

    Multi-Basin Routing `<mbrcell>`

    Impervious Land `<imperv>`

    Urban Detention `<urbandet>`

    Consumptive Use `<cu>`

8.1.4 Aggregating Pseudocells `<hub>`

The original concept of pseudocells envisioned one pseudocell associated with each mesh cell to simulate the detailed local hydrology for that area. It is possible that a small number of generic pseudocells could provide adequate local hydrology for the RSM. In most water management applications, however, water managers need to simulate areas more complex than application of single pseudocells allows. Although simple pseudocells may be used in large areas of the model domain, in regions where the landscape is complex one mesh cell may include a combination of natural, agricultural, and urban areas; or several mesh cells may cover a partially urbanized area, with agricultural areas interspersed. To simulate such areas without unduly complicated arrangements of mesh cells, the "hub" was introduced. This provides a means to combine the hydrologic aspects of the area covered by several mesh cells into a single object. The hydrology of the hub may be represented by several types of pseudocells with varying hydrologic parameters. The relative effect of each pseudocell type is controlled by specifying the percentage of the area of the hub that each pseudocell represents. The implementation of the hub is described in section 8.4.1. Four pseudocell types; `<pumpedditch>`, `<agimp>`, `<imperv>`, and `<urbandet>`; were developed specifically to be used as part of a hub while the others may be applied as stand-alone pseudocells or as part of a hub.

8.2 Natural System Pseudocells

The natural system pseudocells that are designed to simulate local hydrology in relatively undisturbed areas can be grouped by hydrologic processes into two distinct groups of land uses, wetlands and uplands. The principal distinction is the interaction with the surficial aquifer. In wetlands and other areas where the water table is in the root zone for most of the year, the local hydrology is largely controlled by the depth to the water table. In upland areas there is substantial water storage in the unsaturated zone above the water table but below the root zone. This water will drain from the soil over extended periods contributing to surface water and regional groundwater. These natural areas differ from developed areas in that the hydrology is controlled by the native landscape features and water moves slowly through the landscape. In developed areas, the hydrology is controlled by man-made features that are designed to move water quickly to maintain water levels. The natural systems pseudocells are briefly described below:

1. The `<layer1nsm>` pseudocell type is used to represent the local hydrology for wetlands and high water table soils. This pseudocell works well where the water table is in the root zone for extended periods of the year. The available soil water for evapotranspiration is determined by the location of the water table. When the water table is below the root zone the simple algorithm used in this model does not accurately describe evapotranspiration and the water budget is not accurately simulated.

2. The `<mbrcell>` pseudocell was developed as a simple runoff model that combines the NRCS curve number runoff algorithm with simple linear reservoir routing.

3. The `<unsat>` pseudocell is an extension of the layer1nsm pseudocell type. Whereas the layer1nsm assumes that there is no unsaturated soil and all of the water for evapotranspiration is extracted from the water table, unsat maintains moisture accounting in the unsaturated zone as well as tracking the water table. The available moisture in the unsaturated zone is extracted for evapotranspiration demand before water is removed from the water table.

4. The `<layer5>` pseudocell is an extension of the unsat pseudocell. The layer5 pseudocell is composed of 2 water layers above the ground surface, the shallow root zone, the deep root zone, and the deep soil layer. `<layer5>` tracks the unsaturated zone soil moisture and water table. In addition, the `<layer5>` pseudocell has the capability of modeling the soil moisture of multiple soil horizons as would occur in a typical soil profile.

5. The `<layerpc>` pseudocell performs no calculations and uses none of the pseudocell access functions. It is used as a place-holder when the simulation of local hydrology is not desired. This may occur during the calibration process to test limited sections of the model domain; either to conduct testing of pseudocells in a limited area or testing

of other model components without pseudocells. It may also be used with lower layer cells in a three dimensional groundwater simulation to maintain mass balance.

8.2.1 Natural Wetland System `<layer1nsm>`

This pseudocell was introduced to satisfy the need to simulate natural hydrology in natural system models. This pseudocell calculates a simple water budget for the soil with a water table that is defined by the water level in the mesh cell. In the sequence of computations the following processes occur as shown in Figure 8.4. Rain is applied to the surface and initially, interception storage (Interception) is satisfied. The remaining water infiltrates to the water table. Evaporation (Evap) occurs from the Interception at the rate of the potential evapotranspiration rate (PET) until the water is gone or the PET is satisfied. The amount of infiltration and inflows from other mesh cells (CellDelta) is added to the water table. The change in the position of the water table is determined by the amount of water added, Equation 8.3, where Sy is the specific yield of the mesh cell if the water table is below the ground surface and is 1.0 if the water table is above the ground surface.

$$H_{temp} = H_{i-1} + \frac{Rain + CellDelta - Interception - Evap}{Sy} \tag{8.3}$$

where
H_{i-1} is the head at the beginning of the time step and
H_{temp} is a temporary head from which additional ET occurs to produce the head at the end of the time step.

Evapotranspiration depth is removed from the re-computed water table according to Equation 8.4.

$$H_i = H_{temp} - \frac{AET}{Sy} \tag{8.4}$$

Where actual evapotranspiration (AET) is calculated as the remaining PET after evaporation from interception storage times a crop correction coefficient (Kc). The value of Kc shown in Figure 8.5 and Table 8.2 depends on the location of the water table in Figure 8.5. Z is the elevation of the ground surface. The linear decrease of Kc with depth below the root zone reflects the influence of capillary upflux from the water table into the root zone. The values for Xd, the deep root zone or extinction depth, were quantified as calibration coefficients for producing reasonable dry season ET budgets.

It is assumed that the Actual Evapotranspiration (AET) from a pond is greater than the ET from the vegetation. When the water table is below the root zone AET decreases linearly with depth, which mimics the effect of capillary upflux from the water table. ET in these cells is completely cut off when the water level falls below the extinction depth, Xd.

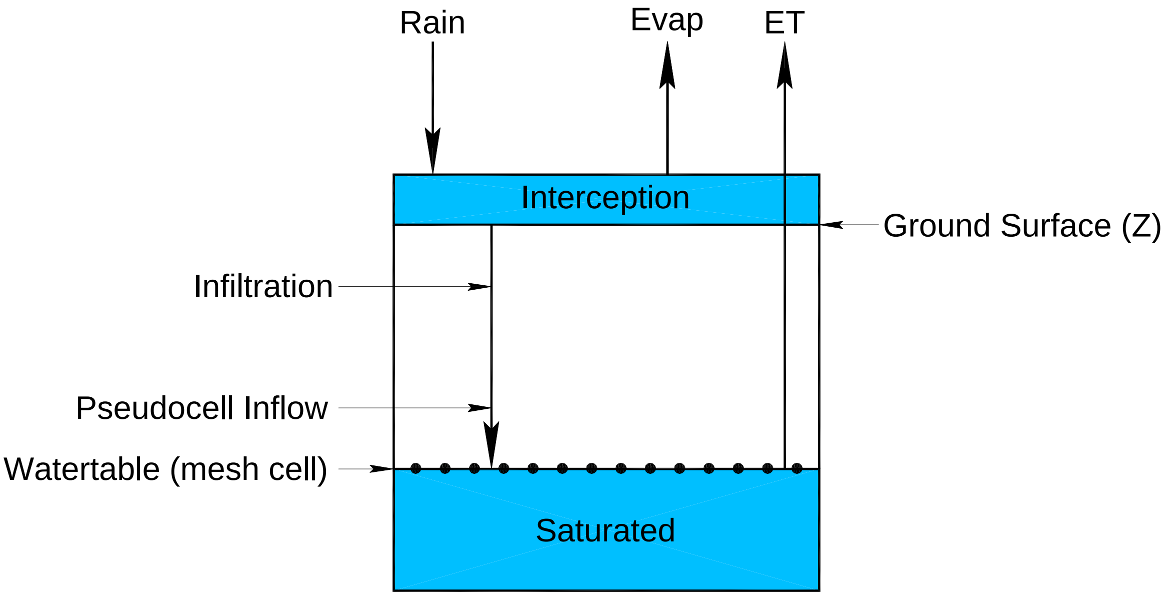


Figure 8.4: *Pseudocell Components Water Budget for the layer1nsm pseudocell.*

As a result, when the water level is low as in upland areas, this pseudocell can underestimate the ET and overestimate the runoff. This pseudocell may be better configured to simulate upland areas if Xd is set to a relatively large value.

The XML elements and attributes used to describe a `<layer1nsm>` pseudocell are described in Table 8.3. An example of the XML input for a `<layer1nsm>` pseudocell is shown in Table 8.4 and used in BM16.

lu.index is an index file that assigns a pseudocell identified by its `<entry>` id to each mesh cell.

8.2.2 Three Dimensional Groundwater Cell Pseudocell `<layerpc>`

The `<layerpc>` pseudocell simulates no hydrologic processes and requires no attributes in the XML input. When simulating three dimensional flow under both confined and unconfined conditions there is no need to carry out hydrologic functions, but there is a need to maintain mass balance. This pseudocell acts as a placeholder for this function in a three dimensional groundwater simulation. A sample XML input for a `<layerpc>` pseudocell is shown Table 8.5.

Table 8.2: *Water table location and value of the crop coefficient correction for* `<layer1nsm>` *pseudocells.*

Watertable	Crop Coefficient (Kc)
WT - Z $\geq$ Pd	Kw
0 $\leq$ WT - Z $<$ Pd	$(Kw - Kveg)\frac{WT-Z}{Pd} + Kveg$
Rd $\geq$ Z - WT $>$ 0	Kveg
Xd $>$ Z - WT $>$ Rd	$Kveg\frac{(Xd-(Z-WT))}{Xd-Rd}$
Z - WT $\geq$ Xd	0
Z = mesh cell ground elevation	
WT = water table elevation	
Kw = open water crop coefficient	
Kveg = vegetation area crop coefficient	
Rd = shallow root zone depth	
Xd = extinction depth below which no ET occurs	

Table 8.3: *Elements and Attributes for the* `<layer1nsm>` *pseudocell. Element cells are shaded.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<layer1nsm>`	Designates the NSM type pseudocell			
kw	Maximum crop coefficient for open water	Real	0 - 1	1.0
rd	Shallow root zone depth, (m)	Real	0 - 10	2.6
xd	Extinction depth below which no ET occurs (m)	Real	0 - 10	5.6
pd	Open Water ponding depth (m)	Real	0 - 2	1.3
kveg	Vegetation crop coefficient	Real	0 - 1	0.7
imax	Maximum Interception (m)	Real	0 - 1	0.15

Table 8.4: *Example XML input for a* `<layer1nsm>` *pseudocell.*

```
...
   <pseudocell>
      <indexed file="lu.index">
        <walllist> 1 2 2 3 3 4 8 12 </walllist>
          ...
            <entry id="2">
              <layer1nsm kw="1.0" rd="0.5" xd="2.0" pd="3.0" kveg="0.75" imax="0.0">
              </layer1nsm>
            </entry>
          ...
      </indexed>
   <\pseudocell>
```

Table 8.5: *Example XML input for a* `<layerpc>` *pseudocell.*

```
...
   <pseudocell>
      <indexed file="lu.index">
         ...
         <entry id="2">
           <layerpc> </layerpc>
         </entry>
         ...
      </indexed>
   <\pseudocell>
...
```

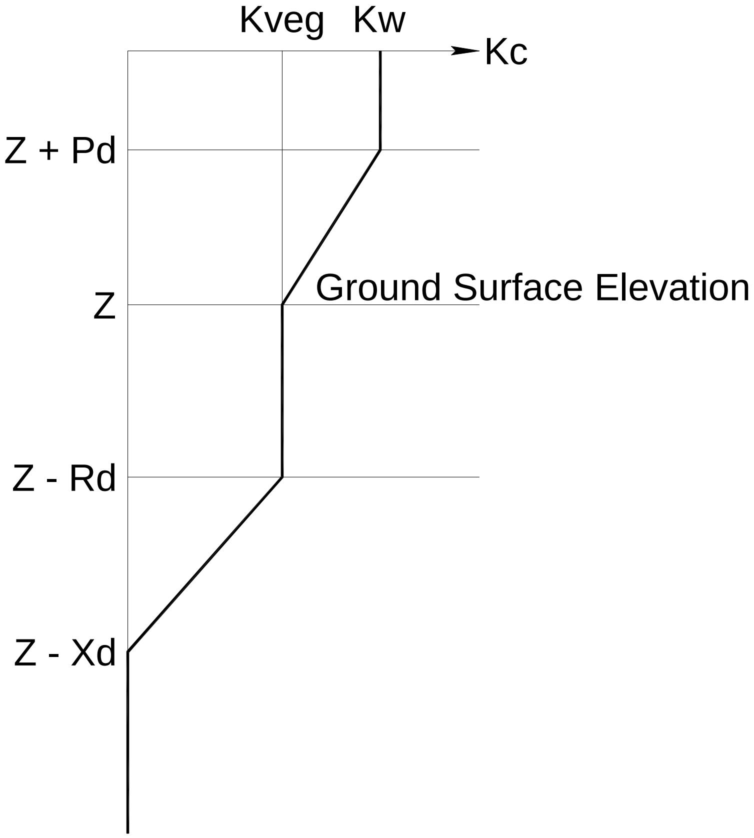


Figure 8.5: *Variation of the Reference ET Crop Correction Coefficient, K_c, in* `<layer1nsm>` *with water table. Pd, Rd, and Xd are positive and measured from the ground surface elevation (Z).*

8.2.3 Multi-Basin Routing Pseudocell `<mbrcell>`

The MBRcell was created to provide a simple pseudocell for modeling runoff and routing. Urban areas are different from natural areas mainly because of the more rapid runoff in urban areas with less infiltration and recharge. Much of this runoff occurs before the soil is fully saturated. For urban runoff, the time of concentration is short because of a high percentage of impervious land and numerous drainage canals and storm sewers. The runoff approach in the `<mbrcell>` is similar to the NRCS curve number method for calculating the volume of runoff. The runoff volume is routed through a linear reservoir to control the hydrograph timing. The NRCS method is used because of its simplicity and because the method accounts for unsaturated soil moisture and infiltration. The linear routing is based on the Multi-Basin Routing model (MBR) used in the SFWMD. The MBR module used in the HSE is different from the MBR model in the SFWMD because of the accounting of infiltration and treatment of flooding conditions.

The MBRcell pseudocell is an uncoupled pseudocell as the water table is not obtained from the home cell head. Although the water table depth is an input variable in MBRcell, the water table head is not explicitly tracked. In the MBRcell pseudocell there are three water storages (Figure 8.6). The unsaturated layer takes a fraction of the rainfall according to the NRCS method. The excess rainfall (runoff) is directed to urban detention (linear reservoir). Any excess water from the unsaturated storage (greater than field capacity) enters the saturated layer. ET is withdrawn from the unsaturated layer first, and the saturated layer next. Urban detention storage retains the water that is not routed in the current time step.

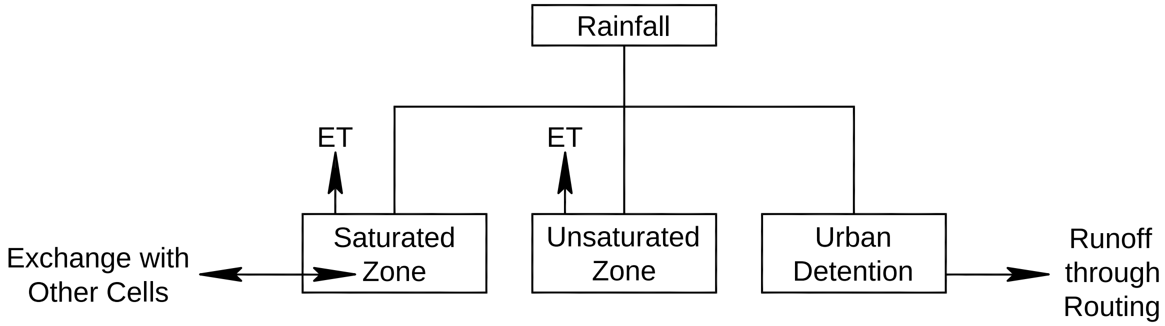


Figure 8.6: *Schematic of hydrology in MBRcell pseudocell.*

The Natural Resources Conservation Service (NRCS) curve number (CN) method was developed to determine the volume of runoff from a given amount of precipitation (USDA-NRCS, 1985). The method is based on the following relationship. All the quantities are measured in units of depth.

$$\frac{F}{S} = \frac{Q}{P_e} \tag{8.5}$$

where P =accumulated total storm rainfall
Ia =initial abstraction
$Pe = P - Ia$ = effective storm runoff
$F = Pe - Q$, infiltration occurring after the beginning of runoff
S =potential abstraction
Q =runoff
Since it normally assumed that Ia =0.2S, runoff can be computed as

$$Q = \frac{(P - 0.2S)^2}{P + 0.8S} \tag{8.6}$$

In the MBRcell the CN approach is modified by evaluating the potential abstraction at every time step and controlling the timing of direct runoff by using a linear reservoir routing approach. The use of a curve number is avoided by using soil moisture values from the model itself. The potential abstraction is computed as:

$$S = (z - h)(Sy + Fld\_cap) - uns \tag{8.7}$$

where
Sy = specific yield,
Fld_cap = field capacity,
uns = water storage in the unsaturated zone,
z = land surface elevation, and
h = head

Runoff is then computed as

$$Q = \frac{(P - 0.2S)^2}{P + 0.8S + uns} \tag{8.8}$$

which differs from the standard NRCS curve number method by the addition of the variable "uns" in the denominator.

The curve number approach for estimating runoff is commonly used in the temperate U.S. However, it does not work well in subtropical South Florida which has high rainfall rates, high infiltration rates and low soil water storage.

The second function of the MBR vertical solution is to route the water using a linear reservoir model. If the urban catchment has a time of concentration of *tc*, and if the time step used in the simulation is t, then the amount of discharge leaving the catchment during one time step is

$$Q_{(i)} = RO(1.0 - K_s) + K_s Q_{(i-1)} \tag{8.9}$$

in which,

$$K_s = \frac{(t_c - 0.5t)}{(t_c + 0.5t)} \tag{8.10}$$

$Q_{(i)}$ is the discharge at time step i and RO is the runoff during the current time step.

Evapotranspiration (ET) is computed using the following equations with z being the elevation of the ground surface.

If the head is above the shallow root zone; ($H \geq z - D_{shallow}$) then
ET = PET * Kveg;
If the head is in the deep root zone; ($H \geq z - D_{deep}$ and $H < z - D_{shallow}$) then

$$ET = \frac{[H - (z - D_{deep})]}{[(z - D_{shallow}) - (z - D_{deep})]} PET * Kveg \tag{8.11}$$

and if ($H < z - D_{deep}$) ET = 0.0;

where
$D_{shallow}$ and D_{deep} are depths of the shallow and deep root zones respectively,
PET = reference ET,
Kveg is a crop coefficient for ET.

Once runoff and ET are subtracted from rainfall, the remaining water is added to the unsaturated zone. When the unsaturated zone is filled to field capacity the remaining water is added to the water table increasing the head. If there is enough water, ponding occurs. In the current implementation, the amount of runoff in excess of the routed runoff, Q - RO, remaining in the linear reservoir, is lost from detention as ET. The intent is to calibrate the runoff algorithm so there is no excess runoff.

The elements and attributes used in defining an `<mbrcell>` pseudocell are shown in Table 8.6. A simple example of the MBRcell pseudocell is presented in Benchmark 25 and in Table 8.7.

Table 8.6: *Elements, attributes, and typical values used for the MBRcell pseudocell.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<mbrcell>`	Designates the mbrcell type pseudocell			
route	wb ID for destination of discharge	Long	Any valid wb ID	7
tc	Time of concentration (s)	Integer	900 - 50,000	3600
kveg	Evapotranspiration crop coefficient	Real	0.0 1.0	0.85
`d_shal`	Shallow root zone depth (m)	Real	0.5 - 3.0	2.0
`d_deep`	Deep root zone depth (m)	Real	1.0 - 5.0	3.5
`fld_cap`	Water content at field capacity m)	Real	0.0 - 2.0	0.2

Table 8.7: *Example XML for an mbr pseudocell.*

```
<pseudocell>
   <indexed file="lu.index">
       <entry id="5">
           <mbrcell route="7" tc="3600.0" kveg="1.0"  d_deep="2.0"
                          d_shal="0.5"  fld_cap="0.2"
           </mbrcell>
       </entry>
   </indexed>
</pseudocell>
```

8.2.4 Five Soil Layer Pseudocell `<layer5>`

The `<layer5>` pseudocell shown in Figure 8.7 simulates the water budget in a layered unsaturated soil with rainfall and potential evapotranspiration as meteorological input. In the current version, the soil is divided into three layers. The other two layers are ponded layers above ground level with the lower layer being in the vegetated zone and the top layer above the elevation at which open water occurs. Figure 8.8 shows the layers along with the ET coefficient Kc. Kc decreases from Kw in the open water layer to Kveg in the vegetated zone and the shallow root zone, and then decreases linearly to zero at the bottom of the deep root zone. The linear decrease of Kc with depth below the root zone reflects the influence of capillary upflux from the water table into the root zone. The values for Xd, the deep root zone or extinction depth, were quantified as calibration coefficients for producing reasonable dry season ET budgets. The pseudocell is designed to model several soil layers that have different soil water characteristic values, extractable water and gravitational water. Gravitational water is the amount of water that will drain freely from the soil (specific yield) and extractable water is the difference between field capacity and wilting point. The amount of water in each soil layer is tracked during the simulation.

The `<layer5>` pseudocell is coupled to the mesh cell in that the head is obtained from the mesh cell at the beginning of the simulation. The amount of extractable water and gravitational water in each soil layer is adjusted by the amount of lateral inflow or outflow from adjacent mesh cells (Celldelta) and inflows from other pseudocells (PsInflow). Inflow water is added to the soil from the water table upward and outflow water is removed from the water table downward. Rainfall fills the extractable water and then the gravitational water from the surface to the bottom of the cell and then from the surface upward as required, (Equation 8.12).

$$WaterContent = InitialWaterContent + CellDelta + PsInflow + Rain \tag{8.12}$$

where CellDelta is the inflow from the regional solution and
PsInflow is inflow from other pseudocells.

Evapotranspiration occurs from the top layer downward through the soil. The amount of potential ET that can be removed from each soil layer is calculated using the crop adjustment factors as indicated in Figure 8.7. As can be seen evapotranspiration does not occur from the bottom layer below the deep root zone. The actual amount of ET removed from each layer is limited by the amount of gravitational water and extractable water in each layer. The fraction of potential ET demand satisfied by each layer is calculated progressively from the ponded layer downward. For each layer the available water is removed from the gravitational water and then from the extractable water, Equation 8.13. After ET is complete, the water is redistributed by removing all gravitational water and then filling the soil column from the bottom to the top, first filling the capillary storage in all the layers and then the gravitational storage.

$$WaterContent = InitialWaterContent - \sum_{i=0}^{3} KC_i PET_i \tag{8.13}$$

where
i = the index for each layer,
KC_i is the PET coefficient, and
PET_i is the PET remaining for layer i.

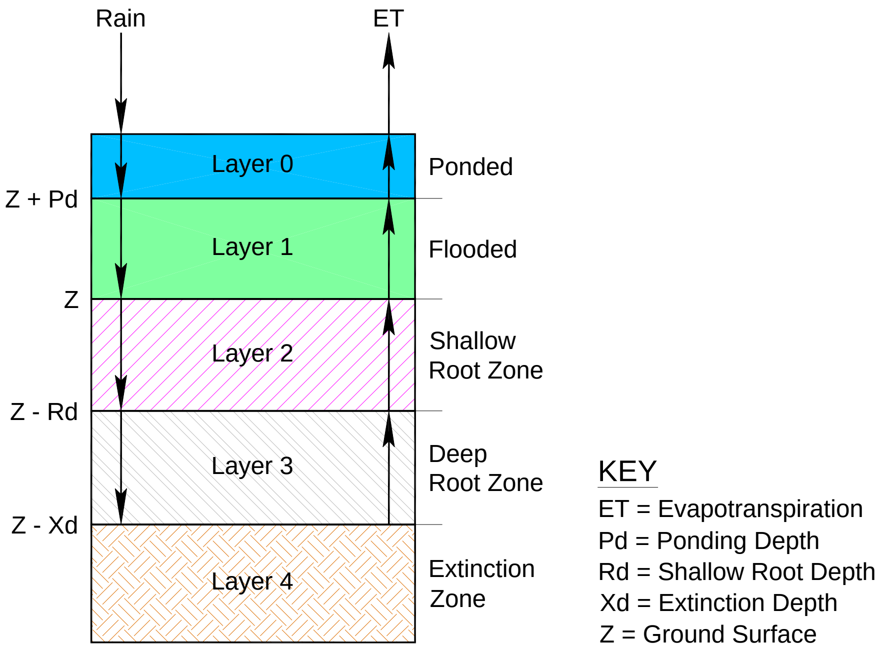


Figure 8.7: *Soil layers modeled in the Layer5 pseudocell.*

In the current implementation, the input parameter values for the `<layer5>` pseudocell are limited to the description of the five layers in the soil (Table 8.8). The elements and attributes used to define a `<layer5>` pseudocell are described in Table 8.8. The extractable water is a depth of water equivalent to field capacity minus wilting point. The gravitational water which is equal to saturation minus field capacity is equivalent to the specific yield. The specific yield is obtained from the properties of the mesh cell to which the pseudocell is attached.

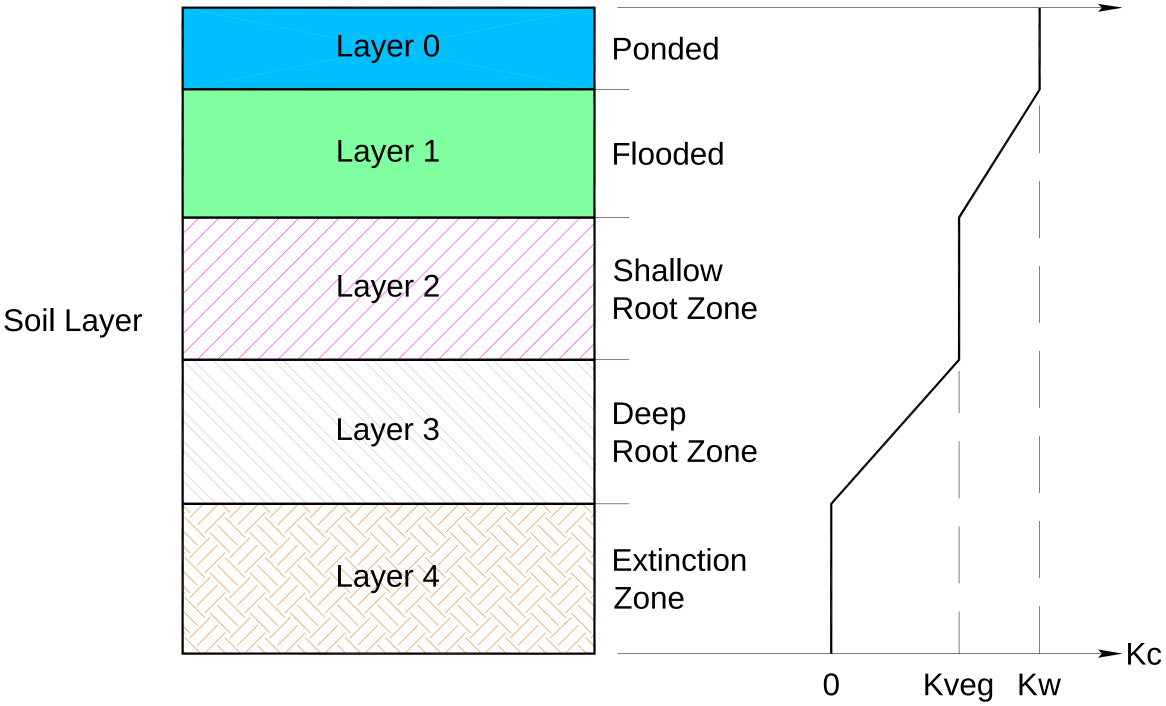


Figure 8.8: *ET coefficient, Kc with water level in the Layer5 pseudocell.*

An example of the XML input for the layer5 pseudocell is provided in Table 8.9 and in Benchmark 18.

Table 8.8: *Elements and Attributes for the* `<layer5>` *pseudocell.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<layer5>`	Designates the layer5 type pseudocell			
ew	Extractable water	Real	0.0 - 0.5	0.2
kw	Maximum crop coefficient for open water	Real	0.5 - 1.2	1.0
rd	Shallow root zone depth, (m)	Real	0.2 - 2.0	1.0
xd	Extinction depth below which there is no ET, (m)	Real	1.0 - 10.0	3.0
pd	Ponding depth, (m)	Real	0.0 - 2.0	1.0
kveg	Vegetation crop coefficient	Real	0.0 - 1.0	0.85

Table 8.9: *Example XML for* `<layer5>` *implementation.*

```
...
    <pseudocell>
      <indexed file="lu.index">
        <entry id="1">
          <unsat ew="0.2" kw="1.0" rd="0.5" xthresh="0.02"
            pthresh="0.10" pd="3.0" kveg="0.75">
          </unsat>
        </entry>
        <entry id="2">
          <layer5 ew="0.2" kw="1.0" rd="2.0" xd="5.0" pd="3.0" kveg="0.5">
          </layer5>
        </entry>
        <entry id="3">
          <layer5 ew="0.2" kw="1.0" rd="0.0" xd="0.5" pd="3.0" kveg="0.65">
          </layer5>
        </entry>
      </indexed>
    </pseudocell>
...
```

8.2.5 Unsaturated Soil Pseudocell `<unsat>`

The Unsat pseudocell computes a simple water budget for a single-layer soil, Figure 8.9. This pseudocell is similar to the `<layer1nsm>` pseudocell except it considers water in the unsaturated soil above the water table in the water balance accounting whereas `<layer1nsm>` does not. Since the water budget accounts for the water content of the unsaturated zone, this can be a useful option to use in areas where the water table may be well below ground for a significant portion of the year.

The initial water table at the beginning of the simulation is mapped from the underlying mesh cell. Water is added to or subtracted from the pseudocell by the horizontal inflows from adjacent mesh cells (CellDelta), from other pseudocells (PsInflow) and rainfall. The total water in the pseudocell is then redistributed to fill the soil from bottom to top and to determine the water content (Θ) in the unsaturated zone. Evapotranspiration is determined by PET and the elevation of the water table which determines the water availability to the roots, the ET crop coefficient, Kc (Table 8.10) and the soil water content of the soil in the unsaturated root zone. The linear decrease of Kc with depth below the root zone reflects the influence of capillary upflux from the water table into the root zone. The values for Xd, the deep root zone or extinction depth, were quantified as calibration coefficients for producing reasonable dry season ET budgets. After ET occurs, the water is again redistributed in the vertical.

$$Head_i = Head_{i-1} \pm CellDelta \pm PsInflow + Rain - Kc * PET \tag{8.14}$$

where Head$_i$ = head at the end of the time step
Head$_{i-1}$ = head at the beginning of the time step
CellDelta = water added from the regional solution
PsInflow = water added from other pseudocells
Rain = rainfall during the time step
Kc = the ET coefficient, and PET = the potential evapotranspiration.

The elements and attributes used to describe the Unsat pseudocell are presented in Table 8.11. The specific yield is determined by the soil properties of the underlying mesh cell.

An example of XML input for an Unsat pseudocell is presented in benchmark BM18 and in Table 8.12.

Table 8.10: *Water table location, available water content and crop coefficient values for* `<unsat>` *pseudocells.*

Watertable	Available water content	Reference Crop ET Correction Coefficient (Kc)
> Pd	Ponded water	Kw
> Land surface and < Pd	Flooding	Pwd/pd x (Kw - Kveg)+ Kveg
< Land surface	> Pthres	Kveg
< Land surface	Xthres < Θ < Pthres	(Θ − Xthres)/ (Pthres − Xthres) * Kveg
< Land surface	< Xthres	0
The parameters above are defined in Table 8.11		

Table 8.11: *Elements and attributes for the* `<unsat>` *pseudocell.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<unsat>`	Designates the unsaturated type pseudocell			
ew	Extractable water (m)	Real	0.0 - 1.0	0.2
kw	Maximum crop coefficient for water	Real	0.5 - 1.0	1.0
rd	Root zone depth, (m)	Real	0.0 - 2.0	0.5
wilt	Soil water content at wilting point, (m)	Real	0.0 - 0.1	0.02
pd	Ponding depth (m)above which open water occurs	Real	0.0 - 2.0	1.0
xthresh	Soil water content when ET ceases (m)	Real	0.0 - 0.3	0.1
xthresh	Soil water content when Kc begins to decrease from Kveg	Real	0.0 - 0.5	0.22
kveg	Vegetation crop coefficient	Real	0.5 - 1.0	0.75

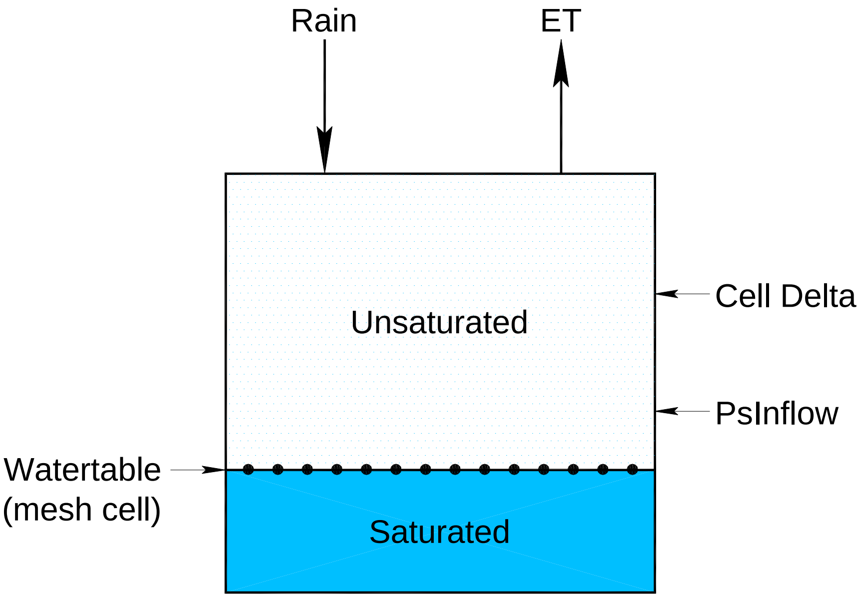


Figure 8.9: *Schematic water budget for Unsat pseudocell.*

Table 8.12: *Example XML for an unsat pseudocell.*

```
...
    <pseudocell>
        <indexed file="lu.index">
           <entry id="1">
                <unsat ew="0.2" wilt="0.03" kw="1.0" rd="0.5" xthresh="0.02"
                             pthresh="0.10" pd="3.0" kveg="0.75">
                </unsat>
           </entry>
        ...
        </indexed>
    </pseudocell>
...
```

8.3 Urban PseudoCells

The key characteristic of simple urban pseudocells is the amount of impervious land that results in greater runoff and reduces the amount of evapotranspiration. The simplest urban pseudocell is for impervious land. It is possible to represent a fraction of of the urban land as turf grass representing lawns and landscaping, and model that land using afsirs. Urban land can also be modeled using the precipitation-runoff routing pseudocell which can be calibrated for stormwater detention and routing. The two simple pseudocells designed for simulation of urban areas are

1. The `<imperv>` pseudocell simulates impervious areas with rainfall, ET, surface storage and runoff. There is no recharge to the underlying soil.
2. The `<prr>` pseudocell is a deterministic lumped parameter conceptual model with moderate input data requirements. Water is stored in various compartments; interception, upper zone and lower zone from where is is apportioned to runoff, groundwater recharge, evapotranspiration, and interflow.

8.3.1 Impervious Area `<imperv>`

Specific hydrologic processes occurring on impervious areas include rainfall, evaporation, interception, surface detention, runoff, and seepage from storm sewers and ditches carrying water from the impervious areas to detention ponds or canals. These processes are simulated by the pseudocell `<imperv>`, Equations 8.15 through Equation 8.18. Rain falls directly on the roofs and sidewalks where some is intercepted by plants, buildings, and other structures as well as stored as a thin layer over the surface that is a function of surface roughness. Inflow from other pseudocells in the hub is added to interception storage at this time. Evaporation removes water from the interception storage at the rate of RefET.

$$IntSto_i = IntSto_{i-1} + Rain + PsInflow - RefET \tag{8.15}$$

If the remaining interception storage is greater than a user input maximum interception storage, the excess (IntRunoff) enters depression storage; depressions in the impervious surface such as holes and other low areas.

$$IntRunoff = IntSto - MaxIntSto \tag{8.16}$$

Water is then evaporated from depression storage up to any remaining RefET and the depression storage in excess of the user input maximum becomes runoff that is directed to a user designated water body such as a canal or a detention pond `<urbandet>`, or to a pervious area of another pseudocell, or a pervious area within the hub.

$$DetSto_i = DetSto_{(i-1)} + IntRunoff - DetET \tag{8.17}$$

$$Runoff = DetSto - MaxDetSto \tag{8.18}$$

If the impervious area is is directly connected, 5 percent of the runoff is lost as seepage from storm sewers and drainage ditches. The `<imperv>` pseudocell is implemented with the elements and attributes shown in Table 8.13 and an example of xml input to implement an `<imperv>` pseudocell is presented in Table 8.14

Table 8.13: *Elements and attributes for the* `<imperv>` *pseudocell.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<imperv>`	Designates the `<imperv>` pseudocell			
dirconn	Specifies directly or indirectly connected	Integer	0=DCIA, 1=UCIA	1
isto	Maximum interception storage depth, (ft)	Real	0.001 0.1	0.03
isto1	Initial interception storage depth, (ft)	Real	0.001 0.1	0.02
sdet	Maximum depressional surface storage depth, (ft)	Real	0.01 0.2	0.08
sdet1	Initial depressional surface storage depth, (ft)	Real	0.01 0.2	0.05

8.3.2 Precipitation-Runoff Routing Pseudocell`<prr>`

The PRR is a modified version of a model originally developed by the Technical University of Denmark and which is now used by the Danish Hydraulic Institute (DHI) in the MIKE 11 model (DHI,1997). It is characterized as a "deterministic lumped parameter conceptual model with moderate input data requirements".

In the model (Figure 8.10), water is stored as interception storage and in an upper storage zone denoted by U and a lower storage zone denoted by L. The meteorological input data are precipitation, and potential evapotranspiration. On this basis, it produces catchment runoff, and groundwater recharge. The resulting catchment runoff is split conceptually into overland flow, interflow and baseflow components.

Upper zone storage

Moisture intercepted on vegetation is stored as interception storage I with a maximum value of IMAX. The upper zone storage, U, is water trapped in depressions in the soil surface and in the uppermost, cultivated part of the ground. Umax denotes the upper limit to the amount of water in the upper zone storage. Water in interception storage and in the upper zone storage is continuously depleted by evaporation. The upper zone storage is also diminished by horizontal leakage (interflow) and by infiltration into the lower zone. When the upper zone storage exceeds the maximum value, a portion of the excess water,

Table 8.14: *Example xml for the* `<imperv>` *pseudocell.*

```
<pseudocell>
   <indexed file="lu.index">

        <entry id="2" label="connected impervious">
          <imperv sdet="0.15" isto="0.01" dirconn="1"></imperv>
        </entry>
        <entry id="3" label="unconnected impervious">
          <imperv sdet="0.05" isto="0.01"></imperv>
        </entry>

  </indexed>
</pseudocell>
```

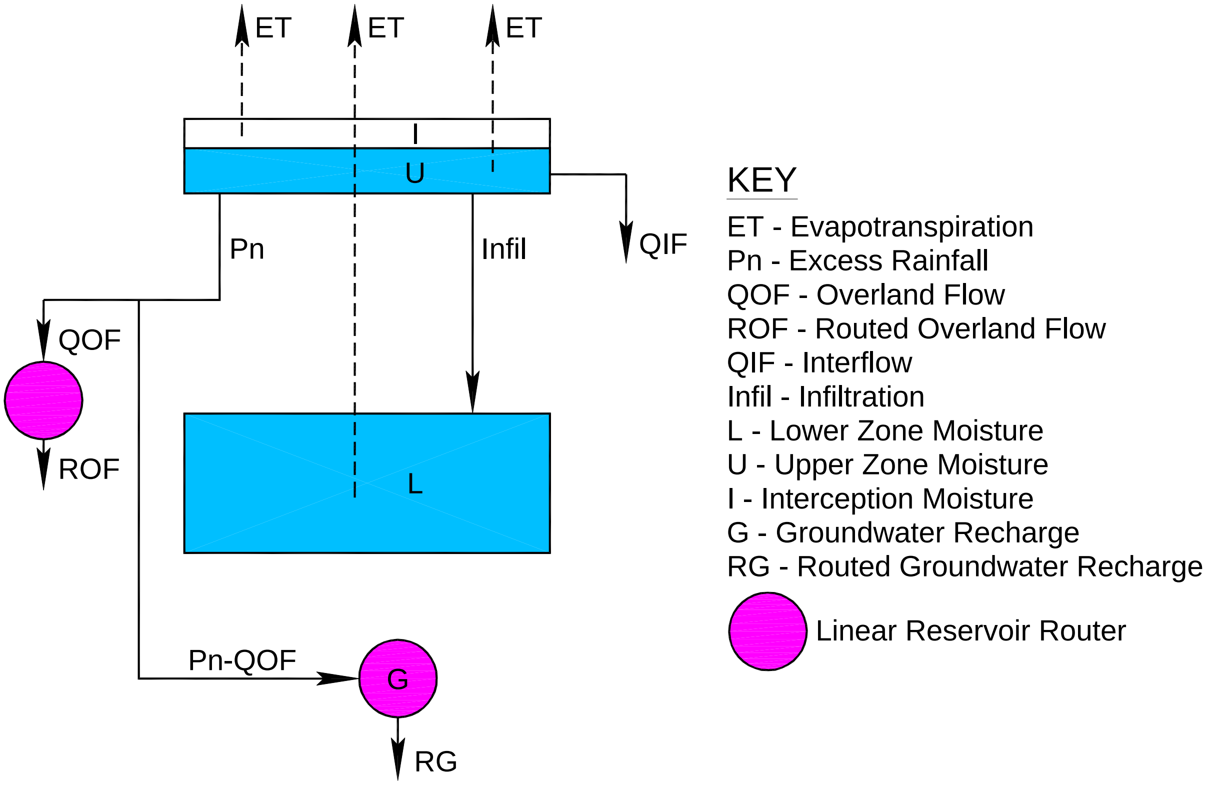


Figure 8.10: *Conceptual diagram of the PRR model.*

$$PN = U - Umax \tag{8.19}$$

will enter the streams as overland flow, whereas the remainder is diverted as infiltration into lower zone and groundwater storage.

Infiltration and Overland flow

When the surface storage spills, i.e. when $U > Umax$, the excess water, PN = U - Umax gives rise to overland flow. QOF denotes the portion of PN that contributes to overland flow while the remainder goes to the lower layer. Overland flow is assumed to be proportional to PN and to vary linearly with the relative soil moisture content, L/Lmax, of the lower zone storage.

$$QOF = PN * CQOF \frac{{}^{L}\!/_{L\max} - TOF}{1 - TOF} \tag{8.20}$$

for $\frac{L}{Lmax} > TOF$

QOF = 0 otherwise

where
CQOF is the overland flow runoff coefficient, $0 \leq CQOF \leq 1$, and TOF is the threshold value of L/Lmax for overland flow where $0 \leq TOF \leq 1$.

A portion of the remaining upper zone storage infiltrates to the lower zone storage. Infiltration is computed as

$$Infil = \left(1.0 - \frac{L}{L\max}\right)(k0inf)dt \tag{8.21}$$

where $k0inf$ is the maximum infiltration rate and dt is the model time step. Infiltration is added to lower zone storage after the remaining hydrologic processes have occurred.

Interflow

The interflow contribution, QIF, is assumed to be proportional to the remaining upper zone storage, U and to vary linearly with the relative moisture content of the lower zone storage.

$$QIF = \frac{dt}{CKIF} \frac{\left({}^{L}\!/_{L\max} - TIF\right)}{(1 - TIF)} U \tag{8.22}$$

for

$$\frac{L}{L\max} > TIF \tag{8.23}$$

QIF = 0 otherwise, where
CKIF is the time constant for interflow, and
TIF is the root zone threshold value for interflow ($0 \leq TIF \leq 1$).

Lower zone or root zone storage
The soil moisture in the root zone is stored in the soil lower zone. The maximum possible value of lower zone moisture content L is Lmax. L is subject to consumptive loss from transpiration and recharge to the groundwater.

Evapotranspiration.
Evapotranspiration demands are first met at the potential rate from interception storage, I, and then from the upper zone storage, U. If the total moisture content in I and U is less than these requirements, the remaining fraction is assumed to be withdrawn by root activity from the lower zone storage at the rate Ea.

$$E_a = E_p \frac{L}{L_{\max}} \tag{8.24}$$

where Ep = remaining potential evapotranspiration after ET from I and U.

Groundwater recharge
The amount of water G recharging the groundwater storage depends on the soil moisture content in the root zone and the remaining excess upper zone storage after surface runoff is subtracted.

$$G = (PN - QOF)\,\frac{\left({}^{L}\!/_{L\max} - TG\right)}{(1 - TG)}$$

for

$$\frac{L}{L\max} > TG$$

G = 0 otherwise

where TG is the root zone threshold value for groundwater recharge
($0 \leq TG \leq 1$).

Soil moisture content

The lower zone storage represents the water content within the root zone. After apportioning the net rainfall (PN) between overland flow and groundwater recharge, the remainder of the net rainfall and the infiltration increases the moisture content L within the lower zone storage.

$$L = L + PN + Infil - QOF - G \tag{8.25}$$

If $L > Lmax$ the excess lower zone storage $L - Lmax$ is added to the routed surface runoff. If the resulting value of L is < 0, L is set $= 0$. The depth of water remaining in the upper and lower zone storages are the beginning values for the next model time step.

Overland flow and baseflow routing

The overland flow is routed through a linear reservoir with time constant KOF and the baseflow is routed through a linear reservoir with time constant KBF. Routing can be suppressed by assigning very small values such as 1.0×10^{-13} to these parameters so that the water partitioned to overland flow or base flow is instantaneously routed from the linear reservoir storage. The equations for routing through a linear reservoir are

$$Sout = KS\frac{(Sin + 2.0Sto)}{(1 + KS)} \tag{8.26}$$

$$Sto_i = Sto_{(i-1)} + Sin_{(i-1)} - Sout_{(i-1)} \tag{8.27}$$

where

$KS = \frac{0.5dt}{K}$,

K = the linear reservoir time constant,

Sin = water added to the reservoir in the time step,

Sout = water leaving the reservoir during the time step,

Sto = water stored in the reservoir at the beginning of the time step, and

dt = model time step.

8.3.2.1 Input Data

The data needed for the PRR pseudocell are defined in Table 8.15. Typical values obtained for a calibration are shown in Table 8.16. The simplified version of PRR contains nine parameters to be determined by calibration. Some of the less important parameters can be set to default values. In particular, TOF, TIF, and TG can often be set to zero. An example of XML input for a `<prr>` pseudocell is shown in Table 8.17 from benchmark 56.

8.3.2.2 Initial Conditions

The initial conditions required by the PRR model consist of the initial water contents in the surface and root zone storages, together with initial values of storages in the two routing reservoirs for overland flow and baseflow. In the current implementation these initial values are set equal to zero. In the calibration, it is recommended to disregard the first half year or so of the PRR simulation to eliminate the influence of erroneous initial conditions.

8.3.2.3 Numerical Experiments

An experiment was carried out to understand the behavior of the PRR model to changing variables and parameters. The purpose of the experiment is to understand the use of pseudocells in upland areas of completely unknown regions. A rainfall data set from the Kalawewa Basin of Sri Lanka was used for the purpose because this is a unique example of a basin where only the annual water budgets and the rainfall are known in terms of parameters. The basin has a tropical forest type landuse. About 26% of the rainfall is known to run off into the ocean. Starting with the parameters shown in Table 8.16, the influence of a number of parameters on the 1999 water budget was studied in this experiment.

A 3x3 RSM mesh was used in the experiment to test the PRR pseudocells. Figure 8.11 shows the mesh and the boundary conditions. In the mesh, the row of cells 2, 11, 5, 14, 8 and 17 is isolated by assigning no-flow boundary conditions along the node lists 2-6-10-14 and 3-7-11-15. Rain is applied on middle cells 5 and 14 while keeping cells 2, 8, 11 and 17 dry. The purpose is to monitor the behavior of cell 5 for the immediate rain impact and the downstream dry cell 11 for the movement of the flood peak. The output is analyzed for water budget components of rain, ET, runoff and seepage when various parameters and variables are changed. The purpose is also to understand the behavior of the level and the lag in the flood peak when various parameters are altered.

Table 8.15: *Definition of attributes of the* `<PRR>` *pseudocell*

Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<prr>`	Designates the PRR type pseudocell			
etcoef	ET crop coefficient	Real	0.0-1.0	0.8
k0inf	Maximum infiltration rate, m/s	Real	0.0-1.0	0.4
imax	Maximum interception storage, m	Real	0.0-1.0	0.05
umax	Maximum upper zone moisture content, m	Real	0.0-1.0	0.5
lmax	Maximum lower zone moisture content, m	Real	0.0-2.0	1.3
tof	Threshold value of lower zone storage L for overland flow	Real	0.0-1.0	0.3
tif	Threshold value of lower zone storage L for interflow	Real	0.0-1.0	0.3
tg	Threshold value of lower zone storage L for groundwater recharge	Real	0.0-1.0	0.3
cqof	Overland flow runoff coefficient	Real	0.0-1.0	0.3
ckol	Time constant for overland flow routing, hrs	Real	0.0-200	47.0
ckif	Time constant for interflow in equation 8.22, hrs	Real	0.0-2000	48.3
ckbf	Time constant for groundwater routing, hrs	Real	0.0-5000	567.8

Table 8.16: *Values of PRR parameters obtained from a calibration.*

Parameter	Value
`etcoef`	0.7
`k0inf`	3.5 10-6 m/s
`umax`	0.025 m
`lmax`	0.27 m
`tof`	0.01 m
`tif`	0.01 m
`tg`	0.01 m
`cqof`	0.5
`ckif`	460 hrs
`ckol`	52.8 hrs
`ckbf`	2784.0 hrs

Table 8.17: *Example XML code from benchmark 56 for a* `<prr>` *pseudocell. The parameters are described in Table 8.15*

```
...
    <pseudocell>
      <nam etcoef="1.0" k0inf="3.5E-6" umax="0.025" lmax="0.27" imax="0.01"
            tof="0.01" cqof="0.50" ckif="480.0" ckol="528." ckbf="2784.0">
      </nam>
    </pseudocell>
...
```

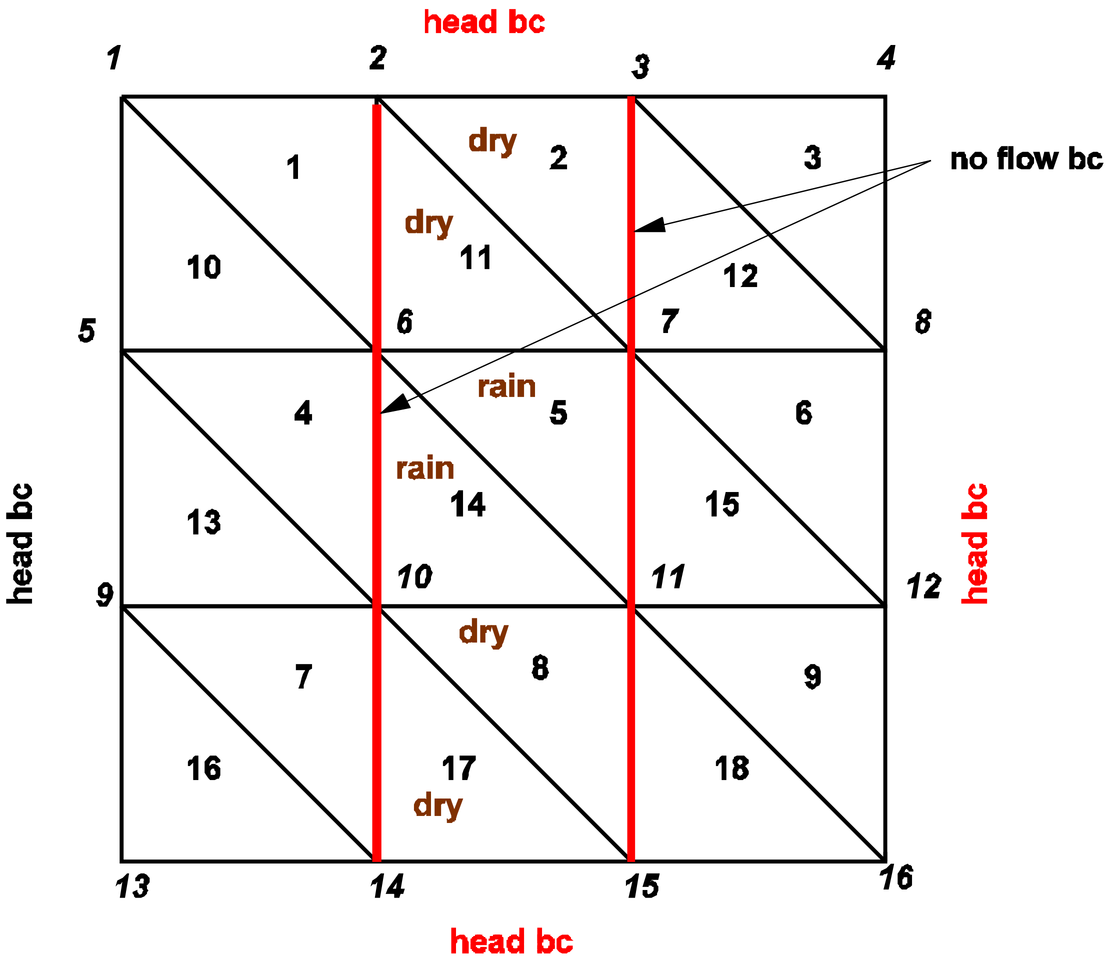


Figure 8.11: *The mesh used to test the pseudocells.*

8.3.2.4 Behavior Of The PRR Model with Variation OF Selected Parameters

Sensitivity of the water budget to PET, water level, Lmax, CQOF, CKOL, CKBF and routing is examined in this section. The first experiment is to understand the influence of PET on the system. Figure 8.12 shows that when PET increases, the annual runoff decreases and the actual ET increases. The results are based on 1999 data for the Kalawewa basin. In the plot, small errors are possible because the data set available is only for one year, and the initialization period used was only for five days.

Figure 8.13 shows how the water budgets change with the water table elevation. The

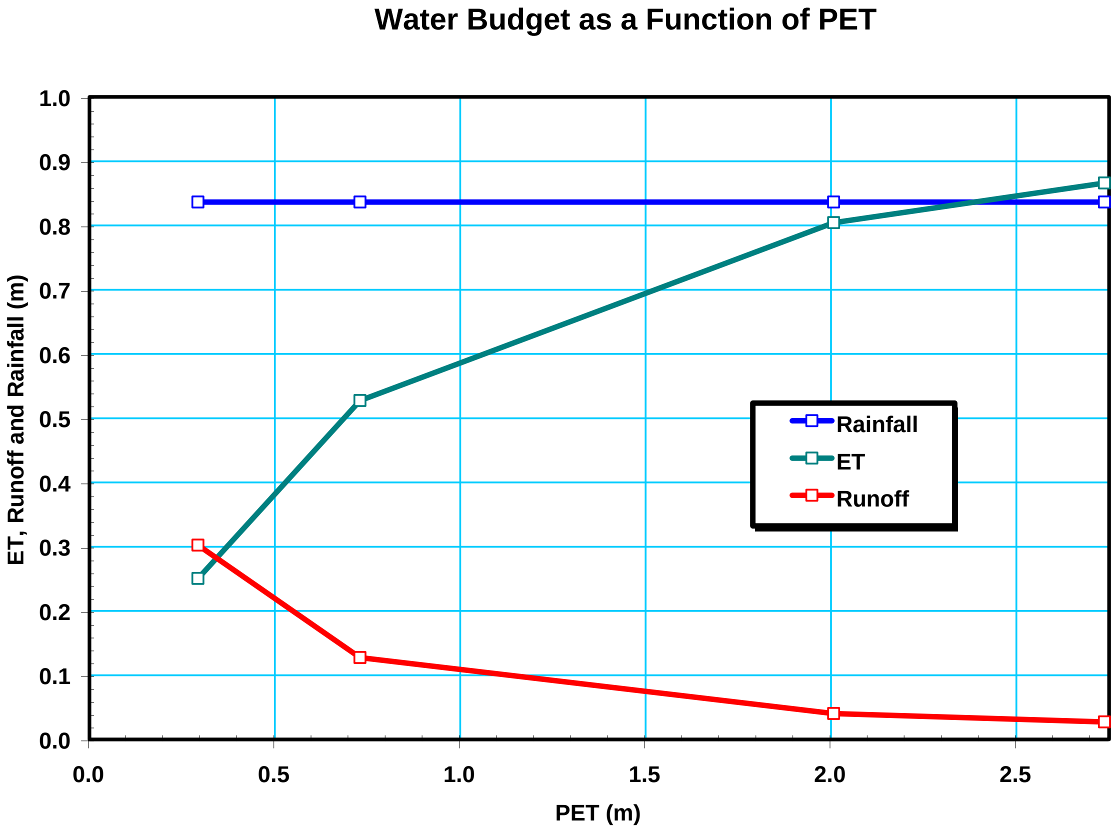


Figure 8.12: *Variation of water budget quantities in (m) with potential ET.*

ground elevation for the problem is assumed as 500.0 m, and the bottom of the aquifer is assumed to be at 0.0 m. When the water level is much below the ground elevation, the ET is capped at about 0.8 m. As the water level gets closer to ground, the ET becomes larger until it reaches PET under flooded conditions. The runoff also increases as the water level gets closer to the ground surface. When it gets flooded (elevation $\geq$ 500.0 meters), runoff becomes zero since runoff routing completely stops, and free overland flow begins. There is no runoff from the pseudocell since there is ponded water. When there is ponded water PET and the deficits in interception and upper and lower zone storages ($\mathrm{I}_{max} - I$, $\mathrm{U}_{max} - U$, and $\mathrm{L}_{max} - L$) are subtracted from rainfall and the remainder becomes seepage that recharges the mesh cell.

Figure 8.14 shows how ET varies with the maximum lower zone moisture. As Lmax increases, overland flow runoff decreases because more water seeps into groundwater. ET also increases with increasing Lmax because more water is available for evaporation from lower zone moisture.

The next experiment is used to understand the influence of the runoff coefficient. Fig-

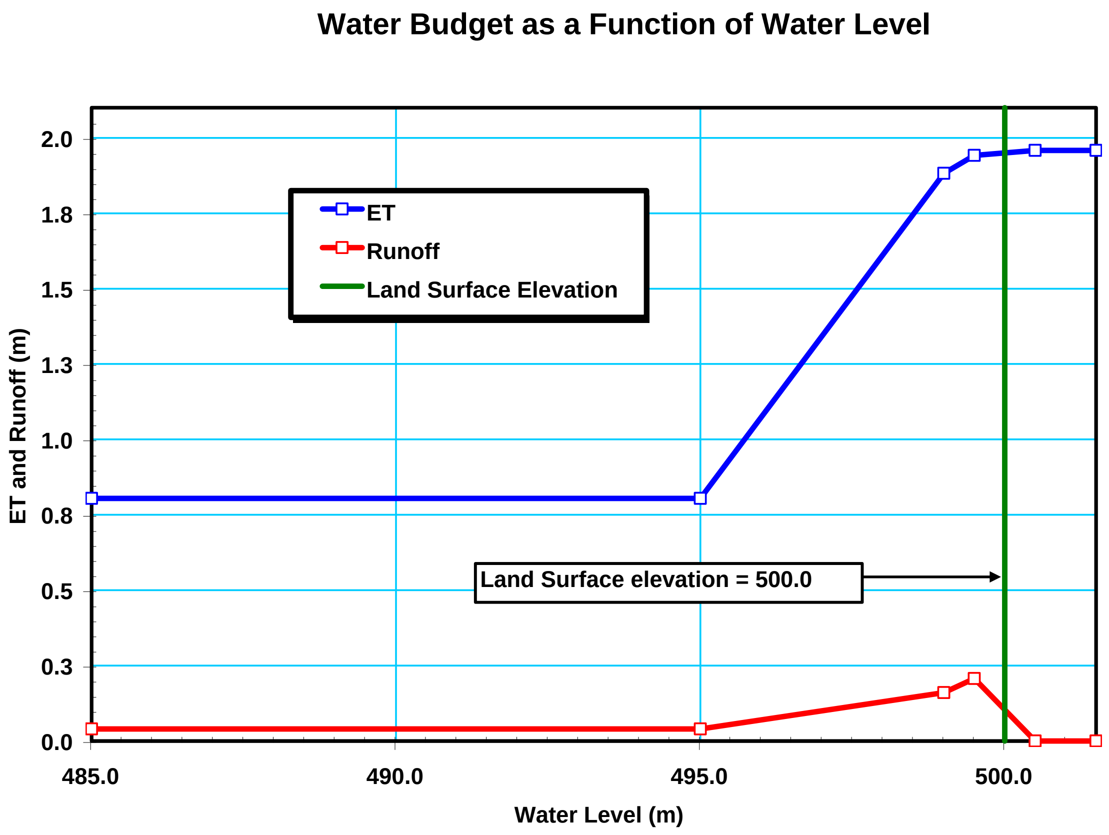


Figure 8.13: *Variation of water budget quantities in (m) with water levels.*

ure 8.15 shows that the volume of runoff is linearly proportional to the runoff coefficient CQOF as expected. This coefficient can therefore be used as a calibration constant for water budgets. If hydrologic data are not available for a basin, a better approach to calibrate a model is to use the NRCS curve number to obtain a value for Lmax. Work in this area is continuing.

Figure 8.16 shows the ability of the PRR pseudocell to route overland flow. The two plots were obtained by changing the time constant from 52.8 hrs to 528 hrs. With longer time constants, the flood peaks are low as expected, and the recession tails are also long. The routing can be eliminated by using extremely small values for the time constant such as 1.0×10^{-13}.

The variation of water levels in cells 5 and 11 changes when the time constant for groundwater routing, CBKF, is changed from 2784 hrs to 1784 hrs. The peak of the hydrograph remains relatively intact while the base flow gets somewhat delayed when the time constant is increased, Figure 8.17

Figure 8.18 shows a portion of the water level hydrograph as a result of no routing

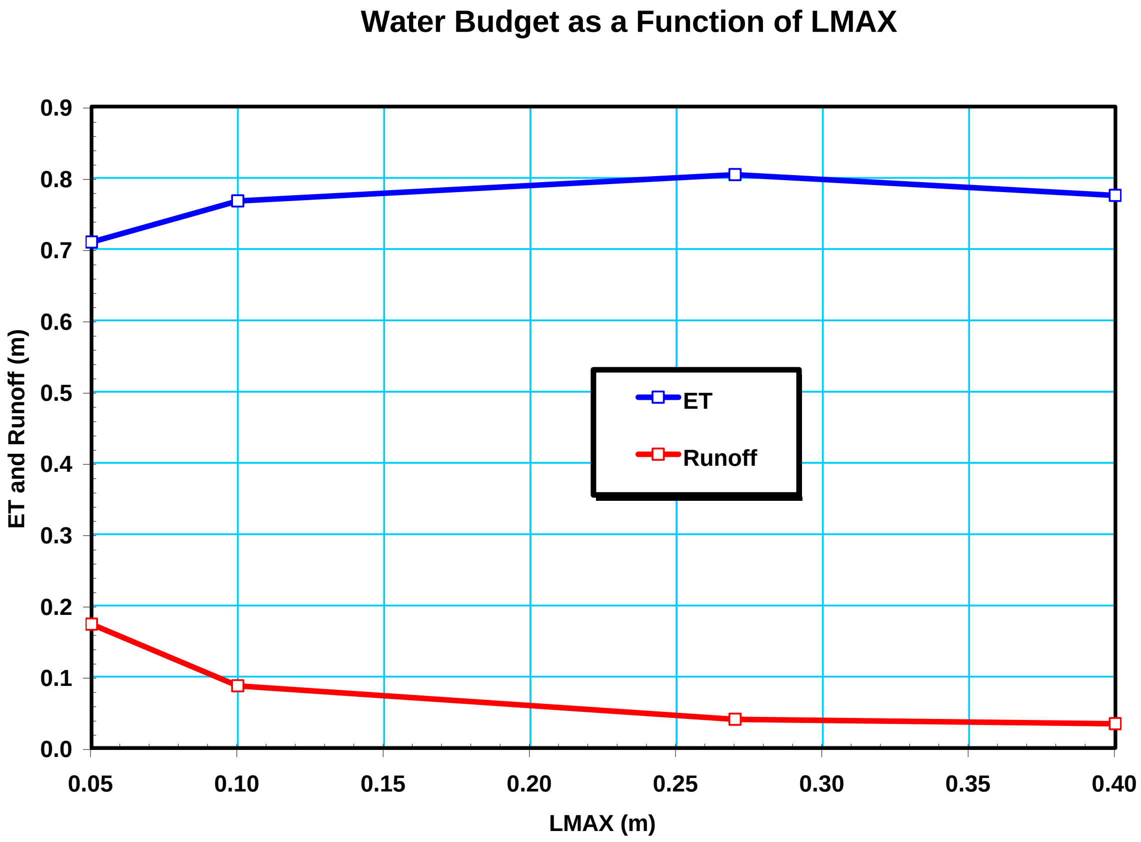


Figure 8.14: *Variation of water budget quantities in (m) with LMAX.*

(extremely small routing time constant) when compared to routing with ckol = 52.8 hrs and ckbf = 2784 hrs. The figure shows that there is some delay of the peak in the downstream cell 11, compared with the peak in cell 5, with no routing and that routing makes the delay longer.

8.3.2.5 Suggested Calibration Steps

RSM and its pseudocells utilize a number of parameters that can be used to explain many different hydrologic behaviors of the system. Some of these behaviors can be attributed to the regional hydrologic parameters, and some may be attributed to the pseudocell parameters. To prevent conflicts and obtain consistent values for parameters, the general calibration process must be based on simple rules.

For best results the parameters should be adjusted in a pre-determined order. Parameter Lmax which strongly affects the annual volume of runoff is generally calibrated in an isolated pseudocell first. The NRCS curve number method is used to determine the maximum

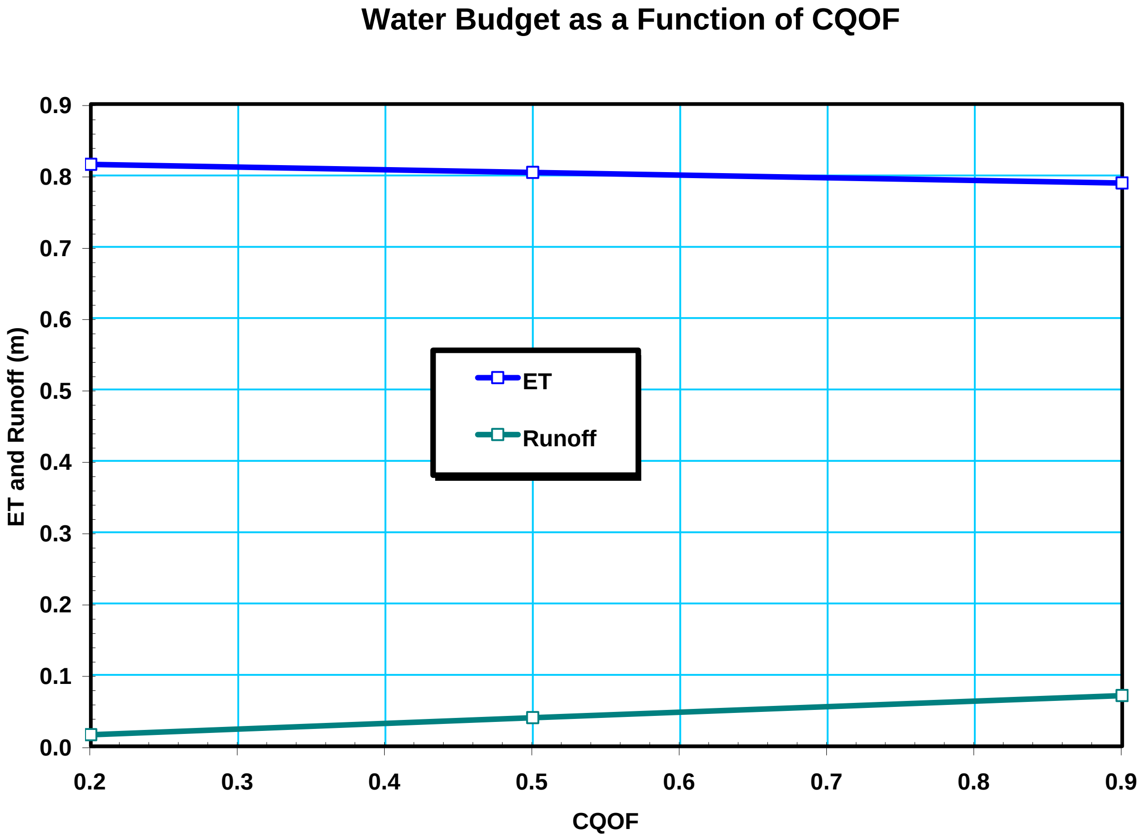


Figure 8.15: *The behavior of annual water budgets with CQOF.*

soil moisture capacity. This is done by choosing an appropriate curve number (CN) and computing Lmax as

$$Lmax = \frac{1000}{CN} - 10 \tag{8.28}$$

The overland flow runoff coefficient, CQOF, can also be used to fine tune the runoff volume, which essentially determines the water budget for the region. ET crop coefficient also has an effect on the runoff, but this value should be determined using independent experimental observations as opposed to model calibration to prevent the system from becoming over determined. Calibration of transmissivity and Manning's roughness parameters in the regional solution can take place next, in order to achieve proper redistribution of water. The time constants for routing overland and groundwater flow through linear reservoirs should be carried out last to match the peaks and timing of the flow hydrographs.

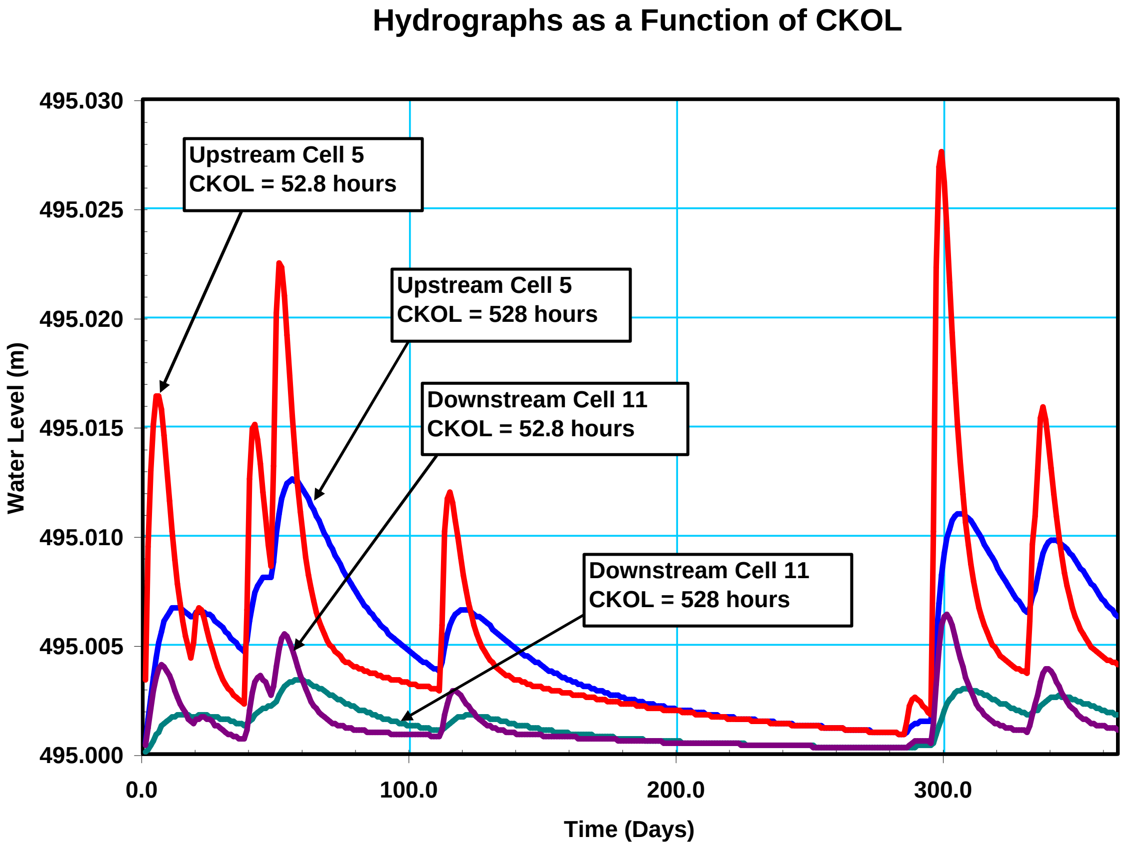


Figure 8.16: *The behavior of flow hydrographs with CKOL.*

8.4 Complex Pseudocells

The simple pseudocells described above are useful for modeling simple landscapes and land use types with simple hydrology. In complex landscapes or land use types with highly managed local hydrology, it is necessary to use complex pseudocells to obtain an appropriate description of the hydrology. The principal tool for modeling the complex location hydrology is the hub. The hub allows the combination of several pseudocell types based on areal distribution. The hub also allows for the implementation of additional pseudocell types that handle local routing.

8.4.1 Water Management Systems <hub>

Although simple urban and agricultural pseudocells can be used to simulate simple, relevant features of the landscape, irrigation demand and accelerated runoff, the landscape of South Florida is more complex and the local hydrology can be better represented by developing

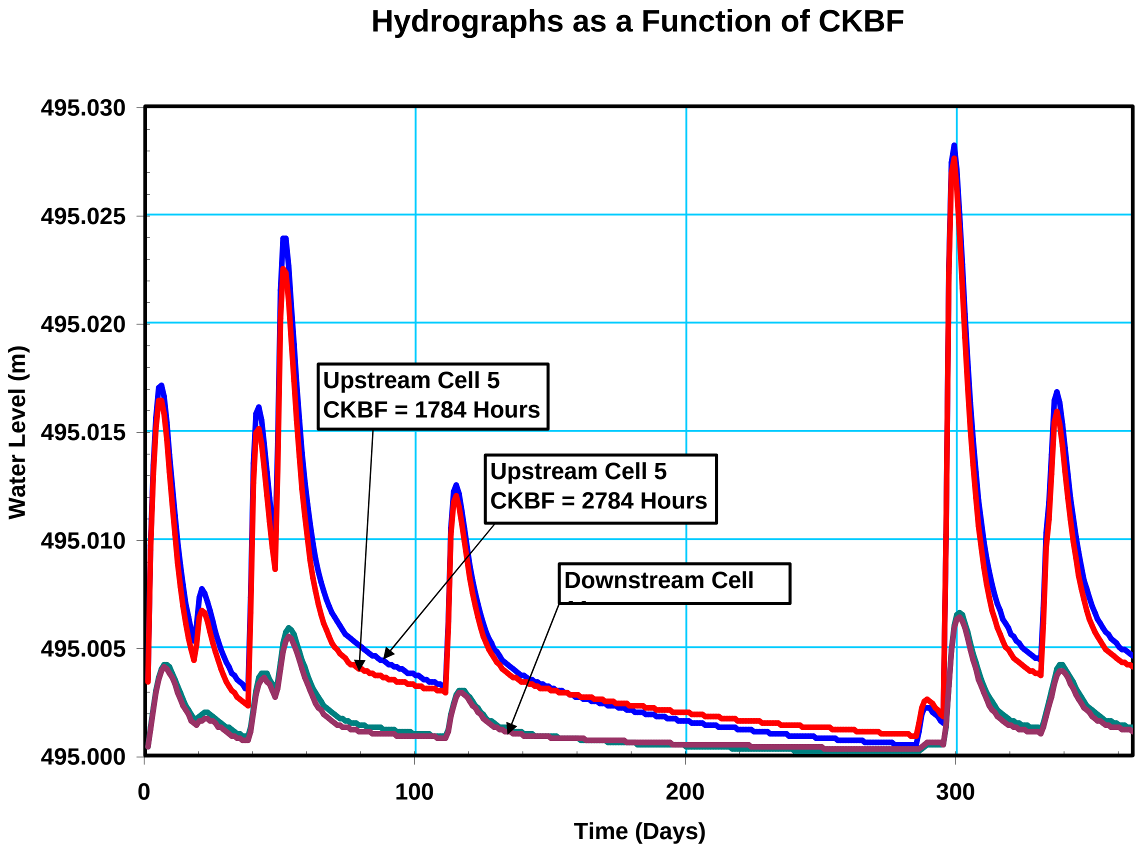


Figure 8.17: *The behavior of flow hydrographs with CKBF.*

water management systems. Water management systems can be represented by the use of hubs. Hubs have the capability of linking several pseudocell types together to model the hydrology of urban and agricultural developments.

The Hub is normally used to model an urban area or a large agricultural operation that has a mixture of land uses and a single water source and runoff destination. This is appropriate for citrus groves or vegetable farms that withdraw irrigation water from a single well or canal and discharge it through a pump or weir to an off-site canal. A single composite pseudocell is created for all of the mesh cells within the farm. The composite pseudocell identifies which pseudocell types are in the Hub, the percent of the total land in each type, water source for each pseudocell type and the destination for runoff from each pseudocell type. For example, in Figure 8.19, a citrus grove may have 60% afsirs-citrus, 15% wetlands, 10% pumped ditches and 15% agricultural impoundment. The water source is a single well for afsirs. The afsirs area drains to the pumped ditch and the pumped ditch drains to the AgImp impoundment which drains to a canal. The composite pseudocell is applied to each mesh cell in the grove. An alternative configuration could have the runoff from 40% of the afsirs drain to the pumped ditch and 20% of the afsirs drain to the isolated wetland. The

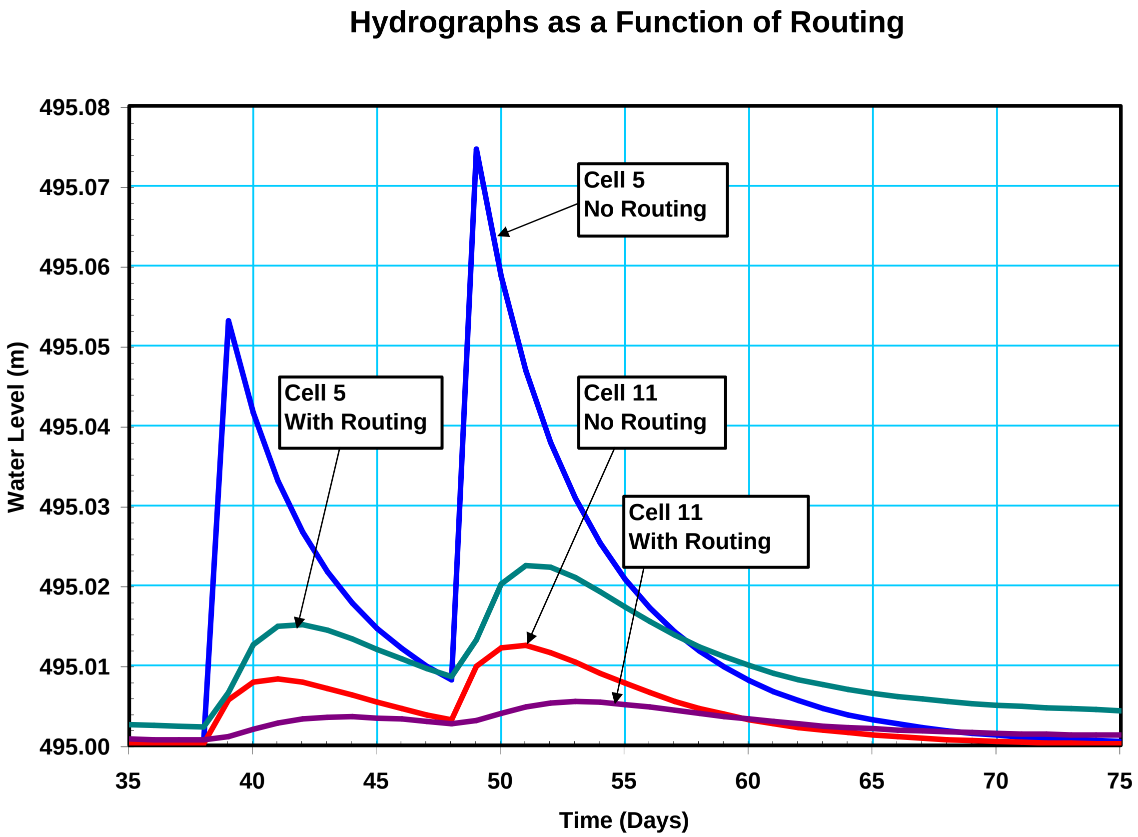


Figure 8.18: *The influence of routing on the behavior of pseudocells.*

wetland would drain to the home cell (immediately below the pseudocell). Use of a Hub simplifies the development of pseudocells as well as creates a better representation of the farm operations.

A Hub is appropriate where the composite pseudocell can be applied uniformly for all cells in the hub and the resulting seepage and water use can be applied uniformly. However, where the location of the impoundment or pumped ditches is important an agricultural hub and an AgImp hub can be linked to obtain the appropriate functionality.

The elements and attributes required to create a hub are listed in Table 8.18. An example xml input showing the creation of a hub with a complex mix of land use types is shown in Tables 8.19 and 8.20.

The preferred implementation of pseudocells will be the utilization of a small number of simple pseudocell types applied throughout the domain, on a one pseudocell per mesh cell basis. For example, all cells with a greatest percent of any land use type within the cell become citrus cells. This is appropriate where the source of irrigation comes from the cell and runoff is directed to the cell. If each citrus cell requires a specific well or discharges to a

Table 8.18: *Elements and attributes for the* `<hub>` *pseudocell.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<hub>`	Designates the `<hub>` pseudocell			
wsupply	Source of the water supply for the hub	String	"homecell" "wb-nnn" "well-nnn" where nnn is a water body or well ID	well-342
sewer	Destination of sewer flow	String	"homecell" "wb-nnn" where nnn is a waterbody number	wb-345654
runoff	Destination of hub runoff	String	"homecell" "wb-nnn" where nnn is a waterbody number	homecell
`<psentry>`	Environment for specification of pseudocells in the hub			
percentarea	Percent of the hub covered by this pseudocell	Real	0.0 100.0	20.0
runoff	Destination of runoff from this pseudocell	String	"homecell" "hub" "ps-nnn" where nnn is a pseudocell ID	hub
wsupply	Source of water supply for this pseudocell	String	"homecell" "hub" "ps-nnn" where nnn is a pseudocell ID	homecell
The pseudocell for this entry `<afsirs>`, `<imperv>` etc., is specified in this environment.				

Table 8.19: *Example XML for typical complex Hub containing native, agricultural and urban pseudocell types.*

```
<pseudocell>
  <indexed file="lu.index">
     ...
    <entry id="5">
// define HUB
      <hub runoff="homecell"    wsupply="wb-206"   sewer="homecell">
// Nature systems
        <psentry psID="2" percentarea=" 10.0" runoff="homecell">
           <layer1nsm kw="1.1" rd="2.0000" xd="4.0000" pd="1.8400" veg="0.85">
           </layer1nsm>
        </psentry>
        <psentry psID="6" percentarea="  5.0" runoff="homecell">
          <layer1nsm kw="1.1" rd="2.0000" xd="4.0000" pd="1.8400" kveg="0.85">
          </layer1nsm>
        </psentry>
// Agricultural Land
// annual crop - tomato
        <psentry psID="11" label="fall tomato - micro irrigation">
           <afsirs coupled="no">
             <afcrop label="tomato" id="60" j1="09-01" jn="12-31" depth1="9"
             depth2="12">
                 <kctbl>
                      1.05 0.75 0.22 0.30 0.30 0.18
                </kctbl>
                <awdtbl>
                      0.40 0.40 0.40 0.65
                </awdtbl>
             </afcrop>
             <afirr label="MICRO, SPRAY" wtd="3.0">
                 <irrmeth id="3" eff="0.8" arzi="0.5" exir="0.4"></irrmeth>
                 <irrmgmt trigcode="0"></irrmgmt>
             </afirr>
             <afsoil label="0.8  SOILS" depth="96" minwc=".07" maxwc=".07" cond="1">
             </afsoil>
           </afsirs>
        </psentry>
// Urban land
// Consumptive use
        <psentry>
           <cu label="HI" percentarea=" 100.0" wsupply="hub">
              <const value=" 0.00221"></const>
              <sewer fracloss="0.1"></sewer>
           </cu>
        </psentry>
```

Table 8.20: *Example XML for typical complex Hub containing native, agricultural and urban pseudocell types (continued).*

```
// UrbanDet
       <psentry psID="16" percentarea="  7.0" runoff="homecell">
           <urbandet rks="10.0000">
              <vnotchweir wlen="0.71" angle="20.0" top="6.42" apx="  5.69">
              </vnotchweir>
           </urbandet>
       </psentry>
// Unconnected Impervious land
     <psentry psID="18" percentarea="  25.0" runoff="ps-17">
         <imperv sdet="0.0328" isto="0.0984"></imperv>
     </psentry>
// Directly connected Impervious land
     <psentry psID="19" percentarea=" 27.0" runoff="ps-16">
         <imperv sdet="0.0328" isto="0.1312" dirconn="1"></imperv>
     </psentry>
// Pervious land
     <psentry psID="17" percentarea=" 10.0" runoff="ps-16" wsupply="hub">
         <afsirs coupled="no">
            <afcrop label="TURF,LNDSCP." id="16" j1="1-1" jn="12-31" depth1=" 6"
                    depth2="24">
               <kctbl>
                   0.40 0.40 0.40 0.90 0.99 0.99
                   0.99 0.99 0.99 0.90 0.50 0.40
               </kctbl>
               <awdtbl>
                   0.50 0.50 0.50 0.50 0.50 0.50
                   0.50 0.50 0.50 0.50 0.50 0.50
               </awdtbl>
            </afcrop>
            <afirr label="SPRINKLER, LARGE GUNS       " wtd="2.5">
                <irrmeth id ="6" eff="0.85" arzi="0.9" exir="0.8"></irrmeth>
                <irrmgmt label = "DROUGHT" trigcode="1 value=0.10"></irrmgmt>
            </afirr>
            <afsoil label=" dirt" depth="80" minwc="0.09" maxwc="0.15" cond="1">
            </afsoil>
         </afsirs>
       </psentry>
     </hub>
   </entry
   ...
</pseudocell>
```

specific canal segment, it is necessary to have a unique pseudocell for each mesh cell. Where there are large blocks of citrus that use the same well, these can be consolidated into a Hub and that unique pseudocell can be applied to several cells. There are several pseudocells that have been developed to work within the hub construct. These pseudocells interact with other pseudocells to provide a better representation of the urban and agricultural landscape. These pseudocells are listed here and described in detail in the following sections.

1. The `<afsirs>` is an agricultural irrigation requirements pseudocell that compute the water budget of agricultural fields. It is customizable for specific crops and irrigation schedules and computes drainage from the soil as well as irrigation demands.
2. The `<pumpedditch>` pseudocell simulates a ditch or system of ditches that is maintained at a nearly constant water level by pumping. The water budget includes inflow, evaporation, pumping and seepage between the ditch and the aquifer.
3. The `<agimp>` pseudocell simulates an agricultural impoundment constructed to meet local environmental requirements. It first computes the size of the impoundment and the outlet structure and then routes inflow through the impoundment to a designated water body.
4. Consumptive use`<cu>` allows the extraction of water from wells for domestic or other use and the return of used water through sewers or septic systems.
5. Retention/detention storage and discharge is modeled using the `<urbandet>` pseudocell. Water is input from other pseudocells and rainfall and leaves through ET, seepage, and discharge through the outlet structure.

8.4.2 Large Agricultural Developments

Typically, the local hydrology for agriculture is conceptualized as a series of containers that represent the agricultural field including root zone soil, runoff collector ditch systems and stormwater detention impoundments. Water moves among the storage containers within the timestep of the model such that the hydrodynamics of water movement are not simulated. Three pseudocells are classified as agricultural pseudocells:

Water in the agricultural pseudocells does not interact directly with the mesh cell water bodies except for seepage to/from groundwater. When the SFRSM is run on a daily timestep this is adequate. When the timestep is substantially shorter, the processes such as evapotranspiration, rain, runoff and irrigation can be adjusted to better reflect the appropriate time of occurrence of each event. For example, rain and runoff occur primarily during the afternoon while irrigation occurs primarily during the morning. Figure 8.19 shows an agricultural area. This area might be modeled by `<afsirs>` for the agricultural land and fields, `<pumpedditch>` for the drainage system, and `<agimp>` for the impoundment.

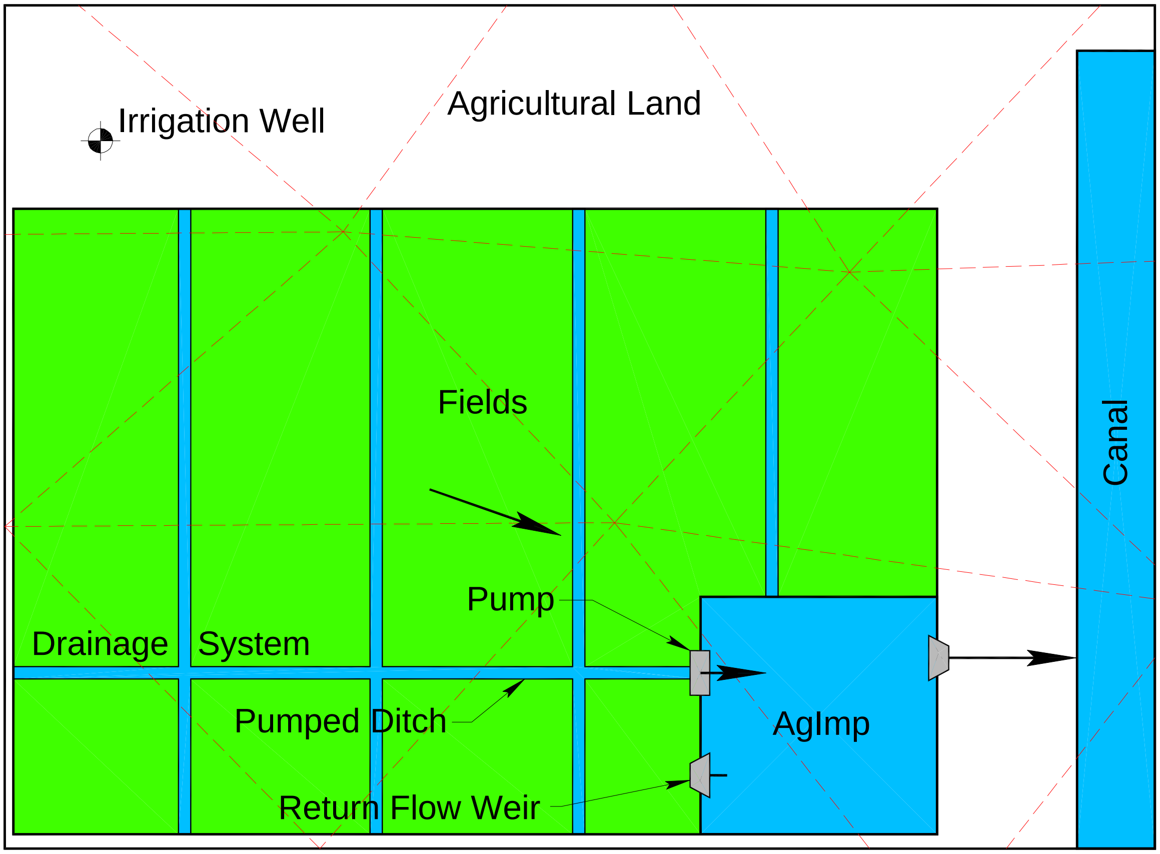


Figure 8.19: *Schematic of hydrology for agricultural land.*

8.4.2.1 Agricultural Irrigation Requirement Pseudocell `<afsirs>`

The Afsirs pseudocell is the primary pseudocell used to estimate irrigation demand and drainage from agricultural land. The afsirs pseudocell was an adaptation of the Agricultural Field-Scale Irrigation Requirement System (AFSIRS) model (Smajstrla, 1990). With minor exceptions discussed below, the Afsirs pseudocell exactly matches the AFSIRS model and the reader is referred to that model documentation for the details on the model structure, components and parameters. The Afsirs pseudocell is a direct implementation of the AFSIRS Fortran model. The model has been decomposed into several Fortran subroutines and a C++ wrapper has been created to pass information to the Fortran routines. The model has been modified to provide a more accurate accounting of crop ET. AFSIRS estimates gross and net irrigation requirements for selected crop type, soil, irrigation method and irrigation management type for a given daily reference crop potential evapotranspiration and rain time series. The model accounts for conveyance and for irrigation application efficiencies. AFSIRS maintains a soil water budget in the root zone. Rainfall is assumed to infiltrate the soil and be available for crop use. The AFSIRS model assumes that the agricultural land is well drained. This works well for South Florida conditions. When the soil water content falls below a set threshold irrigation water is requested from a water source. The water source can be designated as the home cell, a specific well or a canal segment. The AFSIRS model calculates drainage as water that drains from the soil profile below the root zone but the model does not specify whether the drainage becomes runoff or aquifer recharge. In the Afsirs pseudocell drainage is directed to surface runoff and then the remaining water is directed to aquifer recharge. The surface runoff can be directed to the home cell, a well, a canal segment, a specific cell or a specific pseudocell. Typically the runoff is directed to an agricultural impoundment or pumped ditch (discussed below). Where there is no impoundment, runoff may be directed to an adjacent canal segment or adjacent cell. The selection of the destination can be determined based on the drainage characteristics of the watershed and the timestep for the RSM implementation. Where the watershed has a large volume of drainage and runoff is expected to drain to the canal network within a daily time step, the drainage destination for the runoff is set to the nearest canal segment.

Afsirs calculates a daily water budget for crop root zone to determine irrigation requirements.

$$STO_i = STO_{i-1} + RAIN + NIR - DRAIN - RUNOFF - ET \tag{8.29}$$

Where STO is the soil water storage, NIR is net irrigation, DRAIN is drainage, RUNOFF is surface runoff and ET is crop evapotranspiration. All of the units are in inches. For the typical solution of optimal irrigation, NIR is the required water necessary to bring the available soil water (ASW) storage to field capacity after the ASW has fallen below a minimum value set for each crop. For drought conditions or fixed frequency irrigation, NIR may be reduced. The gross irrigation requirement is the net irrigation divided by the efficiency. This efficiency, which ranges from 0.5 to 0.9 for various irrigation methods, accounts for atmospheric losses and field-scale conveyance losses to drainage. The amount of ET is determined by the crop coefficient, rooting depth and ASW. Drainage is determined by rainfall in excess of maximum soil water storage. The Afsirs model requires parameters to describe the crop, soil, irrigation system and irrigation management plan (Table 8.21).

Table 8.21:*Elements and attributes used for the Afsirs pseudocell.*

Element or Attribute	**Description**	**Variable type**	**Suggested Range**	**Example**
`<afsirs>`	Designates the `<afsirs>` type pseudocell			
label	Description of pseudocell	String	Any String	Eric's Farm
coupled	Is water table coupled with water table in mesh cell(Yes) of does afsirs maintain separate soil water accounting(No)	String	Yes or No	Yes

Table 8.21 continued on next page

Element or Attribute	**Description**	**Variable type**	**Suggested Range**	**Example**
`<afcrop>`	Designates attributes for the `<afcrop>` subelement			
label	Description of crop	String	Any String	Strawberries
id	crop id	integer	(1-16, perennial crops, 17-44 annual crops)	23
j1	Crop season start date	string (month - date)	1-1 to 12-31	10-1
jn	Crop season end date	string (month - date)	1-1 to 12-31	2-15
depth1	Irrigated soil depth (in)(perennial crops)	Real	0.0 - 48.0	6.0
	Early season irrigated soil depth (in) (annual crops)	Real	0.0 - 48.0	12.0
depth2	Root depth (in) (perennial crops)	Real	0.0 - 48.0	9.0
	Late season irrigated soil depth (in) (annual crops)	Real	0.0 - 48.0	27.0
kctbl	Table of 12 monthly crop ET coefficients (perennial crops)	Real	0.0-1.0	0.90 0.90 0.90 0.90 0.99 0.99 0.99 0.99 0.99 0.90 0.90 0.90
	or			
	Table of 6 values: Peak crop ET, crop ET coefficient at harvest, and fraction of growing season in each of four crop growth stages (annual crops) - Total of 12 values	Real	0.0 1.0	1.05 0.75 0.22 0.30 0.30 0.18
awdtbl	Table of 12 monthly values for allowable soil water depletion before irrigation (perennial crops)	Real	0.0-1.0	0.67 0.67 0.33 0.33 0.33 0.33 0.67 0.67 0.67 0.67 0.67 0.67
	or			
	Table of 4 values of allowable soil water depletion by crop growth stage (annual crops)	Real	0.0-1.0	0.40 0.40 0.40 0.65

Table 8.21 continued on next page

Element or Attribute	**Description**	**Variable type**	**Suggested Range**	**Example**
`<afirr>`	Designates the `<afirr>` type subelement			
label	Description of irrigation method	String	Any String	Sprinkler, Large Guns
wtd	Managed water table depth	Real	0.0-5.0	2.5
`<irrmeth>`	Designates the `<irrmeth>` type subelement			
label	Description of the irrigation method	String	Any string	Drip Irr
Id	Number for irrigation method type	Integer	See AFSIRS documentation	6
eff	Irrigation application efficiency	Real	0.0-1.0	0.7
arzi	Fraction of area of parcel irrigated	Real	0.0-1.0	0.9
exir	Fraction of crop water use extracted from irrigated soil	Real	0.0-1.0	0.95
drinc	Flood storage depth for rice (in)	Real	0.0-60.0	3.0
crown	Citrus bed height (in)	Real	0.0-36.0	1.5
`<irrmgmt>`	Designates the `<irrmgmt>` type subelement			
label	Irrigation management scenario	String	Drought or Normal	Normal
trigcode	Designates irrigation trigger	Integer	0 - optimum irrigation 1 - fixed irrigation rate (in/day) 2 - irrigate at fixed soil water deficit	
value	Fixed irrigation rate (in/day) or fraction of ASW at which irrigation is applied	Real	0.0-2.0	0.2
`<afsoil>`	Designates the `<afsoil>` type subelement			
label	Soil type description	String	Any soil Description	Myakka fine sand
depth	Thickness of soil (in)	Real	0.0-200.0	80.0
minwc	minimum soil water holding capacity (in)	Real	0.0-0.5	0.09
maxwc	maximum soil water holding capacity (in)	Real	$\geq minwc - 0.8$	0.15

Table 8.21 continued on next page

Element or Attribute	Description	Variable type	Suggested Range	Example
cond	condition code for selecting available soil water capacity	Integer	1 – average 2 – minwc 3 – maxwc	1

A simple example of the afsirs pseudocell is presented in Benchmark 33. This benchmark shows the implementation of several afsirs pseudocells including pasture, sugar cane and citrus as perennial crops and tomato as a typical annual crop. An example of the XML that implements afsirs pseudocells is presented in Tables 8.22 and 8.23. The crops simulated are

1. **Irrigated pasture**

   The irrigated pasture is a perennial crop that is irrigated with large overhead irrigation guns. All of the land irrigated (arzi=1) and the roots obtain all of the water for ET from the irrigated land (exir=1). The irrigation is applied daily (trigcode=1) at a rate of 0.10 in/d.

2. **Crown flood citrus**

   This is a special case of a perennial crop that is irrigated by flooding water furrows along side the citrus trees planted on beds. The efficiency of the irrigation system is low (eff=0.5); fifty percent of the applied water is lost to groundwater recharge. The water table is maintained 1.5 feet below ground surface (crown=1.5) for this crop.

3. **Tomato**

   This is an example of a short season annual crop. The crop is irrigated with micro-jet/low volume irrigation system which is highly efficient and covers a small portion of the field (arzi=0.40). The tomatoes are optimally irrigated (trigcode=0); when the soil water moisture falls below 40 or 65 percent of field capacity (awdtbl=0.4, 0.4, 0.4, 0.65).

4. **Rice**

   The rice crop is a special case of a seasonal crop. Common to the Everglades Agricultural Area, seepage irrigation is used to irrigate the rice crop. The crop is kept flooded during the short season and the depth of water stored on the field in addition to soil water storage is one inch (drinc=1.0).

8.4.2.2 Drainage Collector Ditch Pseudocell `<pumpedditch>`

The pumped ditch pseudocell simulates canal storage that is controlled by a pump. The canal storage can be a series of collector ditches or a detention storage area internal to a farm or a canal in a water control district/drainage district. The prototype is a large citrus grove or vegetable farm where runoff from the field flows to a large collector ditch system. The collector ditches are typically large enough to have a large storage volume. The collector ditches may occupy as much as 10 percent of the area of the agricultural parcel and are sufficiently deep to contain a substantial volume of runoff and provide a sump for the "throwout" pump. Because the collector ditch occupies a substantial area of a farm, a water budget is calculated for the ditch (Equation 8.30).

$$Stage_{(i)} = Stage_{(i-1)} + Rain + PsInflow - ET \pm Seepage - Pumpage \tag{8.30}$$

where $Stage_i$ and $Stage_{(i-1)}$ are the stages at the end and beginning of a time step. Rain is precipitation depth. PsInflow is the inflow of water from other pseudocells. ET is evapotranspiration at the PET rate. Seepage is seepage from the ditch to the mesh cell or into the ditch from the mesh cell. Pumpage is the depth of water pumped from the ditch.

Seepage is computed as

$$Seepage = K\left(Stage - MeshCellHead\right) \tag{8.31}$$

where K is a user specified coefficient.

The pseudocell tracks the ditch water level which is the storage divided by the area of the ditch. The ditch receives rainfall and runoff from the grove (PsInflow). Water is lost from the ditch by evapotranspiration (ET), seepage and pumped discharge (Pumpage). This pseudocell interacts with the local water table only by seepage between the mesh cell and the ditch. Water is removed from the collector ditch by a throwout pump that can remove the water from the farm at a rate as high as six inches per day to a detention impoundment or an offsite drainage ditch. The pump is set to begin pumping at a specified high water level and turn off when the collector ditch level drops to a level one meter below the trigger stage. The maximum pumping during a time step is specified by the user. The elements and attributes used in the `<pumpedditch>` pseudocell are provided in Table 8.24.

A simple example of the pumped-ditch pseudocell is presented in Benchmark 57, and another in Table 8.25. The benchmark shows the implementation and a simple comparison between a citrus grove with a pumped ditch for discharge and a citrus grove with drainage to an impoundment.

Table 8.22: *Example XML for an afsirs pseudocell.*

```
<pseudocell>
  <indexed file="lu.index">
//  perennial crop - irrigated pasture
      <entry id="211" label="Improved pasture">
        <afsirs coupled="yes">
           <afcrop label="TURF,LNDSCP." id="16" j1="1- 1" jn="12-31" depth1=" 6" depth2="24">
              <kctbl>
                    0.90 0.90 0.90 0.90 0.99 0.99
                    0.99 0.99 0.99 0.90 0.90 0.90
               </kctbl>
               <awdtbl>
                    0.50 0.50 0.50 0.50 0.50 0.50
                    0.50 0.50 0.50 0.50 0.50 0.50
               </awdtbl>
           </afcrop>
           <afirr label="SPRINKLER, LARGE GUNS       " wtd="2.5">
                <irrmeth id ="6" eff="0.70" arzi="1.00" exir="1.00"> </irrmeth>
                <irrmgmt label = "NORMAL" trigcode="1" value="0.10"> </irrmgmt>
           </afirr>
               <afsoil label="ave dirt" depth="80" minwc="0.09" maxwc="0.15" cond="1"> </afsoil>
          </afsirs>
      </entry>
// Citrus - crown flood
    <entry id="2211" label="citrus - crown flood">
        <afsirs>
          <afcrop label="citrus" id="4" j1="01-01" jn="12-31" depth1="30" depth2="60">
             <kctbl>
                   0.90 0.90 0.90 0.90 0.95 1.00
                   1.00 1.00 1.00 1.00 1.00 1.00
             </kctbl>
             <awdtbl>
                   0.67 0.67 0.33 0.33 0.33 0.33
                   0.67 0.67 0.67 0.67 0.67 0.67
             </awdtbl>
          </afcrop>
          <afirr label="CROWN FLOOD" wtd="2.5">
              <irrmeth id="8" eff="0.5" arzi="1.0" exir="0.7" crown="1.5"></irrmeth>
              <irrmgmt trigcode="0"></irrmgmt>
          </afirr>
              <afsoil label="0.8  SOILS" depth="96" minwc=".07" maxwc=".07" cond="1"> </afsoil>
        </afsirs>
    </entry>
```

Table 8.23: *Example XML for an afsirs pseudocell (continued).*

```
// annual crop - tomato
    <entry id="2561" label="fall tomato - micro irrigation">
     <afsirs>
       <afcrop label="tomato" id="60" j1="09-01" jn="12-31" depth1="9" depth2="12">
            <kctbl>
                1.05 0.75 0.22 0.30 0.30 0.18
            </kctbl>
            <awdtbl>
                0.40 0.40 0.40 0.65
            </awdtbl>
       </afcrop>
       <afirr label="MICRO, SPRAY" wtd="3.0">
            <irrmeth id="3" eff="0.8" arzi="0.4" exir="0.4"></irrmeth>
            <irrmgmt trigcode="0"></irrmgmt>
       </afirr>
       <afsoil label="0.8  SOILS" depth="96" minwc=".07" maxwc=".07" cond="1"> </afsoil>
     </afsirs>
    </entry>
// Rice - seepage irrigation
    <entry id="9" label="rice - seepage irrigation">
     <afsirs>
        <afcrop label="rice" id="49" j1="01-01" jn="04-30" depth1="12" depth2="18">
            <kctbl>
                1.20 1.05 0.25 0.25 0.25 0.25
            </kctbl>
            <awdtbl>
                0.00 0.00 0.00 0.00
            </awdtbl>
        </afcrop>
        <afirr label="SEEPAGE IRRIGATION" wtd="0.5">
            <irrmeth id="9" eff="0.5" arzi="1.0" exir="1.0" drinc="1.0"></irrmeth>
            <irrmgmt trigcode="0"></irrmgmt>
        </afirr>
        <afsoil label="0.8  SOILS" depth="96" minwc=".20" maxwc=".50" cond="1"> </afsoil>
     </afsirs>
    </entry>
   </indexed>
</pseudocell>
```

Table 8.24: *Elements and attributes and typical values used for the* `<pumpedditch>` *pseudocell as a component of a* `<hub>`*.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<psentry>`	Designates the indexed `<psentry>` environment			
psID	Pseudocell ID	long	100000-200000	134768
percentarea	Percent area of farm occupied by collector ditches (%)	Real	0.0-15.0	8.5
runoff	destination of runoff	String	Homecell or hub	Homecell
`<pumpedditch>`	Designates the `<pumpedditch>` type pseudocell			
rks	Seepage Coefficient (1/day)	Real	0.001 – 0.1	0.025
psize	Maximum Pump discharge rate (in/d)	Real	1.0 – 6.0	3.5
ptrig	Depth above the land surface at which the pump is turned on (m)	Real	≤ 0	-1.0

Table 8.25: *Example xml for PumpedDitch pseudocell.*

```
<pseudocell>
  <indexed>
      ...
    <entry id="1">
      <hub runoff="homecell" wsupply="homecell" sewer="homecell">
      ...
            <psentry psID="2" percentarea="10" runoff="hub">
             <pumpedditch rks="0.001" psize="0.5" ptrig="-2.0"></pumpedditch>
         </psentry>
         ...
      </hub>
      ...
    </entry>
    ...
   </indexed>
</pseudocell>
```

Although developed for large farms and groves, the pumped ditch can be used for golf courses, urban developments and internal canals of drainage districts where the discharge is controlled by a pump.

8.4.2.3 Agricultural Impoundment Pseudocell `<agimp>`

The agricultural impoundment pseudocell was created to simulate the impoundments required by the Surface Water Regulation permitting process developed by SFWMD. All agricultural operations constructed since the mid-1980s have been required to construct an impoundment to capture runoff such that post-development runoff does not exceed pre-development runoff. The Environmental Resource Permit Information Manual, Volume IV (SFWMD, 2000) provides the design specifications for agricultural impoundments. The design specifications provide criteria for the impoundment size and discharge structure (weir and bleeder) characteristics. Review of the agricultural permits from Caloosahatchee watershed indicates that the farms in that region have constructed the agricultural impoundments closely according to the specified criteria and thus the design criteria can reasonably be used to simulate agricultural impoundments in the model. The AgImp pseudocell uses input parameters to compute the size of the impoundment and the design of the outlet structure, and then uses the resulting structure design to compute discharge from the impoundment as a function of stage.

The dimensions of the AgImp pseudocell are determined by the design criteria presented in Vol IV. The crest elevation of the discharge weir is determined by the wet season water table elevation. The crest elevation of the rectangular discharge weir is set above the wet season water table to store the first inch of runoff. The invert of the bleeder is set at the wet season water table. If the impoundment is designed to protect enclosed wetlands, the crest may be raised an additional one or two feet. The storage volume of the impoundment is determined by the runoff created by the 25-year, 3-day design storm rainfall, r25y3d, (Trimble, 1990). The amount of runoff is determined by the NRCS curve number method:

$$Q = \frac{(r25y3d - 0.2S)^2}{(r25y3d + 0.8S)} \tag{8.32}$$

where S is the available soil storage. The soil storage depends on the soil type and landscape and the depth to the wet season water table. The value for S is determined from the soil series GIS coverage. The design volume of the impoundment is determined by the design runoff times the parcel area. The maximum design head is set by dividing the required storage volume by the impoundment area. The length of the discharge weir is determined by considering the allowable basin discharge, Figure 8.21

$$wlen = \frac{AllowableDischarge}{3.13 DesignHead^{1.5}} \tag{8.33}$$

There is a bleeder device below the crest that allows for the discharge of the first inch of runoff at a rate 0.5 inch/d or less for water quality protection. Typically, a v-notch bleeder is used to control this discharge. The discharge rate from the bleeder is calculated by the following equation:

$$Q = 2.5 \tan\left(\frac{\Theta}{2}\right) H^{2.5} \tag{8.34}$$

Since the discharge decreases as the water level decreases, the time average of the discharge as the first 0.5 inch of runoff is discharged in 24 hours is

$$Q_{avg} = 0.41[2.5 \tan\left(\frac{\Theta}{2}\right) H_{max}^{2.5}] \tag{8.35}$$

where the factor 0.41 is computed from the time average of Q, and H_{max} is the head in the impoundment storing the first inch of runoff. Since Q_{avg} and H_{max} are known, this equation is used to calculate the angle of the v-notch weir. When the head is above the crest of the rectangular weir, the v-notch weir is treated as an orifice and the discharge from the impoundment is calculated by the following equation.

$$Q = 4.8 H_2 \tan\left(\frac{\Theta}{2}\right)\left(H + \frac{H_2}{3}\right)^{0.5} + 3.13 L H^{1.5} \tag{8.36}$$

Where
H is the head above the weir crest,
H_2 is the height of the v-notch bleeder,
and L is the length of the rectangular weir.

The first term describes discharge through the v-notch bleeder and the second term describes discharge through the weir. When the head is at or below the crest of the rectangular weir Equation 8.34 is used for flow through the v-notch weir where H is the head above the bottom of the v-notch weir.

The AgImp pseudocell estimates the volume of the surface water runoff detention impoundment and the dimensions of the discharge structure. The impoundment is assumed to be constructed above grade so that the detention drains down to land surface. The storage in the impoundment includes water in the unsaturated zone between the land surface and the root depth. There is an option to provide a storage-volume converter that accounts for borrow ditches or other excavations within the impoundment. The design criteria include setting the discharge weir invert high enough to protect included wetlands. Unlike urban detention ponds, the AgImp is expected to dry out unless it is designed to protect wetlands.

Water loss from the AgImp occurs through evapotranspiration and seepage computed as a function of the head difference between the impoundment stage and the mesh cell head, and through flow from the discharge structure. Discharge from the impoundment is integrated over the time step of the SFRSM. It was originally assumed that SFRSM would use a daily time step. Unlike the Afsirs pseudocell and the Pumped Ditch pseudocell that assume complete discharge for each day, the discharge from the AgImp pseudocell occurs over several days. To accomplish this the discharge for a given day is calculated by a loop that integrates the discharge at 30 minute intervals during each day. For time steps less than a day the discharge is calculated as a fraction of the expected daily discharge. The water volume of the impoundment is calculated for each time step so the mass balance is maintained.

The AgImp pseudocell is useful for any water storage area that dries out and for which discharge is controlled by a weir/bleeder. It may be appropriate to treat the drainage canals and effective runoff detention of selected WCDs that connect to the primary system as large AgImp pseudocells.

The elements and attributes used in the definition of an `<agimp>` pseudocell are listed in

Table 8.26: *Elements, attributes and typical values used for the* `<AgImp>` *pseudocell as a component of a* `<hub>`.

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<psentry>`	Designates the indexed `<psentry>` environment			
psID	Pseudocell ID	long	100000-200000	157843
percentarea	Percent area of parcel in impoundment (%)	Real	5.0-25.0	8.5
runoff	destination of runoff	String	Homecell or hub	hub
`<agimp>`	Designates the `<agimp>` pseudocell			
rks	Seepage coefficient (1/d)	Real	0.0010.1	0.03
height	Wetseason water table elevation (ft -NVGD)	Real	0.045.0	23.6
rd	root depth(bottom of storage)	Real	0.0-5.0	2.6
`<stdtriorifice>`	Creates the environment for the input of discharge parameters			
r25y3d	Rainfall depth for 25-year, 3-day return period storm (ft)	Real	3.0 8.0	5.2
allow	Allowable basin discharge (cfs)	Real	5.0 70.0	35.0
s	Abstraction in the NRCS runoff method (ft)	Real	0.7 1.0	0.85

Table 8.27: *Typical example xml for an agricultural impoundment pseudocell within a hub.*

```
<pseudocell>
    <indexed>

        <entry id="12" label="citrus with agimp" mode="one2many">
           <hub wsupply="wb-21" runoff="wb-21">
              <psentry psID="1" percentarea="90"
                    runoff="ps-2" wsupply="hub">
                   &citrus_drip;
              </psentry>
              <psentry psID="2" percentarea="10" runoff="hub">
                 <agimp rks="0.0005" height="6.0" rd="4.0">
                    <stdtriorif r25y3d="0.792" allow="0.0625" s="0.85">
              </stdtriorif>
                 </agimp>
              </psentry>
        ...
          </hub>
        </entry>
    ...
   </indexed>
</pseudocell>
```

Table 8.26. A simple example of the `<agimp>` pseudocell is in Table 8.27. In this example, drainage from an `<afsirs>` pseudocell simulating a citrus grove with low-volume drip irrigatin drains into an agricultural impoundment. A more complete example is presented in Benchmark 57. This benchmark shows the implementation and a simple comparison between a citrus grove with a pumped ditch for discharge to a citrus grove with drainage to an impoundment. The pseudocell contains a citrus grove (90 percent of the land) and an agricultural impoundment. The runoff from the citrus grove is directed to the impoundment and the runoff from the impoundment is directed to mesh cell 21 (wb-21).

8.4.3 Urban Developments

Urban developments have complex hydrology that can best be modeled using the hub. This provides the mechanism to combine the hydrology of impervious land, pervious land, stormwater detention, urban consumptive use and disposal of septic water.

8.4.3.1 Introduction

Urban hydrology differs from natural and agricultural hydrology in that significant areas are impervious and runoff is often routed through storm sewers and ditches to detention basins, canals, or other water bodies. Lawns, parks and other open areas are often irrigated, and water is withdrawn from the saturated zone for irrigation and domestic use and returned to the soil through septic drain fields, leaking ditches and sewers, and infiltration from irrigation. Urban areas can be simulated by associating several types of pseudocells with the mesh cells in an urban area through the use of a `<hub>`.

An urban area is shown schematically in Figure 8.20. The areas in the figure represent directly connected impervious areas (DCIA) such as parking lots, roads and storm sewers, unconnected impervious areas (UCIA) such as roofs and sidewalks, pervious areas (PA) such as lawns and landscaped areas, a detention pond (Det) and a canal. The UCIA and DCIA areas could be modeled with the `<imperv>` pseudocell, the PA area by `<afsirs>` and the detention area by `<urbandet>`. This is the urban landscape used in the Surface Water Management Model (Huber and Dickinson, 1988). In the typical urban development a large portion of the landscape is impervious area. The runoff from these areas either drains to the pervious area, to onsite detention or offsite to a secondary canal. Urban areas that have surface water permits will have onsite detention ponds. The urban developments receive water from offsite public water supply wells (PWS), are self-served or have both where landscape irrigation comes from a local source. The hydrologic characteristics of each of these areas is described below as well as consumptive use, `<cu>`, and the impacts of drought on the hydrology of an urban area.

In the urban area the pervious areas are typically covered with grass or landscaping shrubbery. The cover is assumed to be uniform turf or landscaping that is irrigated frequently. The pervious area is treated as agricultural land and simulated using the Agricultural Field-Scale Irrigation Requirement Simulation (AFSIRS) Model. This model simulates the water budget in the soil based on water storage capacity, rainfall, evapotranspiration, drainage and irrigation. Evapotranspiration is estimated from potential ET modified by a crop coefficient. The infiltration rate is ignored in this application, since the SFRSM has a daily time step and infiltration is assumed to be complete within a day. The PA infiltration rate is generally similar to the rate of an open field but may be substantially lower depending on the amount of traffic on the site. Drainage is a simple excess inflow over soil water storage. The drainage from AFSIRS is divided into seepage or recharge to the cell and runoff contribution to a detention pond where appropriate, with off-site discharge to a canal.

8.4.3.2 Consumptive Use `<cu>`

The distinction between consumptive (CU) and non-consumptive use of water is a critical aspect of effective water management. Consumptive use of water means that water is not directly returned to the water source from which it was withdrawn. Non-consumptive water use means that, after use, the water is directly returned to the source for use by others. Urban consumptive use is an important component of surface and shallow water table aquifer hydrology, particularly during the dry season. During the dry season, water use for lawns and landscaping is high due to the lack of rainfall, and the higher population of seasonal residents. Although a portion of CU can be modeled as a stress on the aquifer at the locations of public water supply wells and a discharge at the sewage treatment plants, a substantial portion of CU does not return to treatment plants and is either used for landscaping or is discharged through septic fields. There are also several small water supply systems that provide water for individual mobile home parks and other small developments. These systems have small package plants (POTWs) to handle the wastewater.

The volume of CU for each parcel is determined by the land use type. A lookup table provides the volume of water used by each of the primary Florida Land Unified Classification System (FLUCS level II) urban classes. This information is based on data obtained from the USACOE to determine overall urban water use. The volume of CU in a hub is determined by the sum of the CU of each of the urban land uses.

The fate of CU water is determined by the areal extent of the public water supply coverage area and the areal extent of the sewage connections. Areas outside that boundary are assumed to have local disposal. Local disposal is through a seepage field, either individual or community. Although the design of most seepage fields is intended to have discharged water consumed by evapotranspiration, it is expected that the water will percolate to the water table aquifer.

Drought Management

Drought management is modeled by the user inputting a reduced volume of consumptive use assigned to each urban land use and by a reduction in the volume of water used to irrigate the urban turf and landscaping. The CU reductions are based on the expected reduction identified by implementing the selected water conservation practices. The application of irrigation water is reduced by a known fraction according to the phase of water restrictions the basin is experiencing.

Phase I = 15 percent irrigation reduction
Phase II = 30 percent irrigation reduction
Phase III = 45 percent irrigation reduction
Phase IV = 60 percent irrigation reduction

The volume of estimated irrigation requirement is calculated for each crop and reduced

Table 8.28: *Elements and attributes for the* `<cu>` *object.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<cu>`	Designates the `<cu>` environment			
label	High density or low low urban land use	String	HI or LI	HI
percentarea	Percent area of the hub with the specified consumptive use (%)	Real	0-100	50.0
wsupply	Water supply source	String	"homecell" or "hub"	hub
Sub-elements available for flow for `<wsupply>` are `<const>`, `<dss>`, `<asciiform>`, `<csv>`, and `<rc>`. These are described in detail in Section 9.1				
`<sewer>`	Indicates sewer environment			
fracloss	Fraction of water lost from sewer system to groundwater	Real	0.0-1.0	0.1
`<septic>`	Specifies that the consumptive use goes to recharge, otherwise it goes to sewer flow			

by the appropriate factor.

Consumptive use is simulated by the `<cu>` object that is defined within the `<hub>` environment. The elements and attributes used to specify consumptive use are detailed in Table 8.28. A simple example of the consumptive use pseudocell is presented in Table 8.29. See benchmark 55 for a more detailed example.

8.4.3.3 Urban Stormwater Retention/Detention Pseudocell `<urbandet>`

Urban stormwater runoff may be collected and routed through a stormwater detention facility. This facility may include detention for water quality treatment, a retention pond or a stormwater detention pond. Typically, ditches, swales or storm sewers capture runoff that is discharged to a pond designed to meet the SFWMD stormwater permit requirements (SFWMD 2000). The ponds are typically deep and not likely to dry out even during droughts. The area of the pond is typically determined by the type of urban land. The detention pond is simulated by the pseudocell `<urbandet>` in a hub.

Table 8.29: *Example xml for consumptive use in pseudocell.*

```
<pseudocell>
    <indexed>
     ...
        <entry id="1">
          <hub runoff="homecell" wsupply="homecell" sewer="homecell">
        ...
            <cu label="HI" percentarea=" 50.00" wsupply="hub">
              <const value=" 0.00361"></const>
              <sewer fracloss="0.1"></sewer>
            </cu>
            <cu label="LI" percentarea=" 50.00" wsupply="hub">
              <const value=" 0.00361"></const>
              <sewer fracloss="0.1"></sewer>
            <septic></septic>
            </cu>
        ...
          </hub>
        </entry>
    ...
   </indexed>
</pseudocell>
```

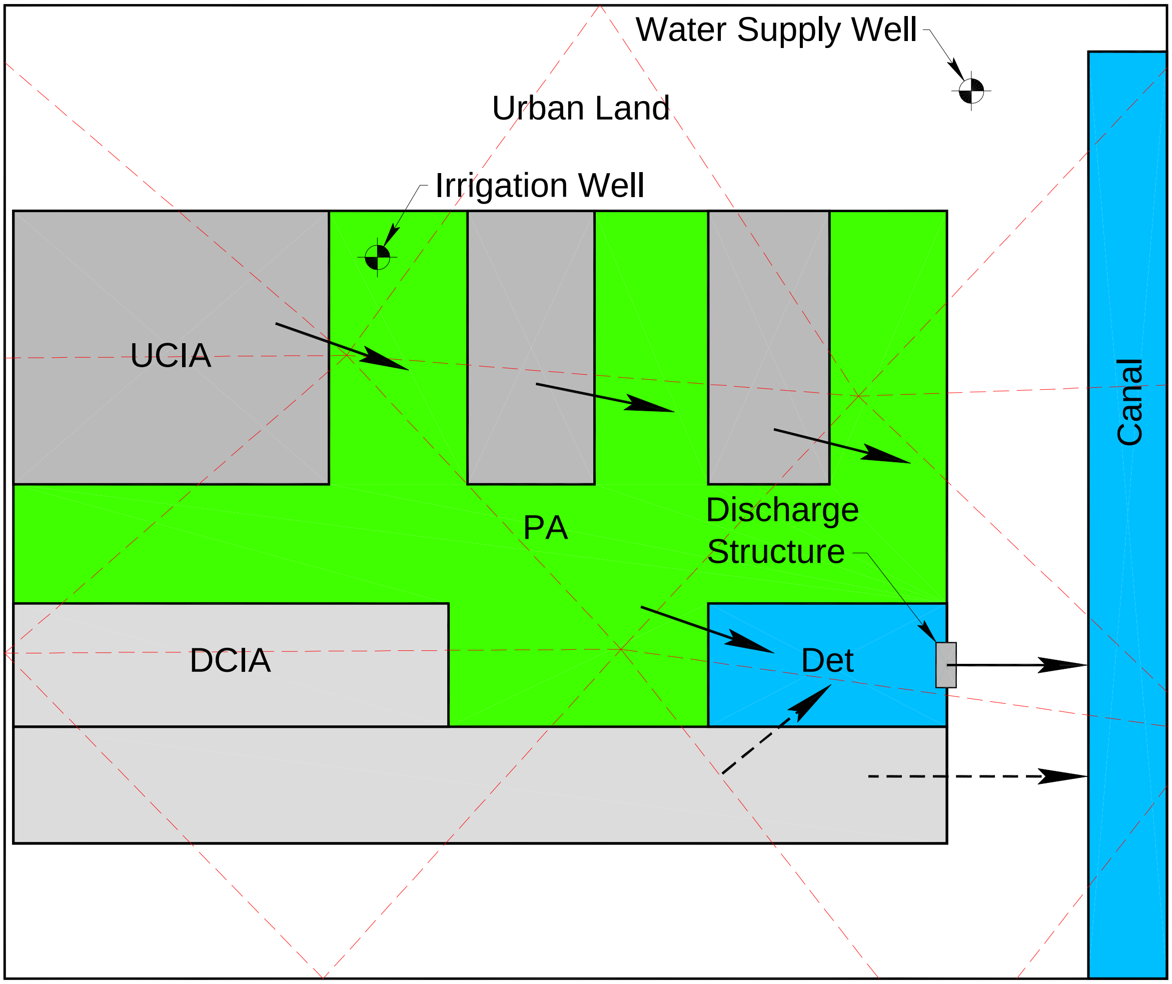


Figure 8.20: *Schematic hydrology for urban land.*

The storm water detention area is treated as a pond with a discharge weir. The storage in the detention pond changes in a time step according to Equation 8.37.

$$S_i = S_{i-1} + Rain + PsInflow - ET - Seepage - Discharge \tag{8.37}$$

where
PsInflow is inflow from other pseudocells in the hub,
ET occurs at the RefET rate of the homecell,and
Discharge is the flow from the outlet structure on the pond.
Seepage (Equation 8.38) recharges the groundwater.

$$Seepage = rks\left(H_{pond} - H_{cell}\right) \tag{8.38}$$

where rks is a user specified coefficient.

Since the `<urbandet>` pseudocell assumes that the RSM operates on a daily time step, a simple hydrologic routing is conducted during each day to calculate the daily water loss from the each impoundment. The water budget is calculated for 48 intervals during each time-step, or at 30-minute intervals for a daily time step. Rainfall is added to the impoundment in equal values during the afternoon hours and ET is removed from the impoundment in equal values during daylight hours. Recharge to the shallow aquifer occurs as seepage based on the difference in head between the pond and the surficial aquifer.

The criteria for detention pond weirs are as follows. For water quality protection, the urban land is required to retain the first one-inch of runoff from the total parcel area or 2.5 times the area of impervious land. The pond is required to retain the runoff from the 25-year, 3-day design storm in the pond. The invert elevation for the bleeder on the discharge structure can be set no lower than the elevation of the average wet season water table. The discharge from the pond into waters of the state can not exceed one-half inch of runoff per day. The maximum allowable discharge for the pond can not exceed the allowable discharge rate for the subbasin in which the project is located. The allowable discharge for each sub-basin is determined based on the conveyance capacity of the ditches and canals. This information is all preprocessed and input by the modeler.

The discharge structure characteristics (Figure 8.21) are determined using the methods in ERP Volume IV using the parcel size, impervious area, pond area, allowable basin discharge, and the 25-year, 3-day return period rainfall volume. The characteristics of the discharge structure include the dimensions of the v-notch bleeder (angle and top elevation) and the length of the sharp-crested rectangular weir (wlen). Discharge from the detention pond through the bleeder is defined by the following equation when the head is below the crest of the rectangular weir:

$$Q = 2.5 \tan\left(\frac{\Theta}{2}\right) H^{2.5} \tag{8.39}$$

Q is discharge in cfs, Θ is the angle of the v-notch in degrees and H is the head on the vertex of the notch (ft). Discharge from the detention pond through the rectangular weir is calculated by the standard weir equation:

$$Q = 3.13 L H^{2.5} \tag{8.40}$$

where L is the length of the sharp-crested weir and H is the head of water on the weir. When the stage exceeds the crest of the weir, discharge through the bleeder should be calculated using the orifice equation:

$$Q = 4.8AH_c^{0.5} \tag{8.41}$$

where A is the area of the notch orifice and H_c is the head above the notch centroid. For triangular orifices the equation converts to:

$$Q = 4.8\left[H_2^2 \tan\left(\frac{\Theta}{2}\right)\right]\left[H + \frac{H_2}{3}\right] \tag{8.42}$$

where
H is the head above the top of the v-notch orifice, and
H_2 is the height of the v-notch orifice (Figure 8.21).

The total flow for the sharp-crested weir and the v-notch bleeder is approximated by the following equation:

$$Q = 3.1LH^{1.5} + 4.8\left[H_2^2 \tan\left(\frac{\Theta}{2}\right)\right]\left[H + \frac{H_2}{3}\right]^{0.5} \tag{8.43}$$

where H = the head on the rectangular weir.

The flow is computed in a loop of 48 iterations for each time step. For a daily time step, the instantaneous discharge is calculated at 30-minute intervals and summed to provide daily discharge.

The elements and attributes used to specify an `<urbandet>` pseudocell in a hub are presented in Table 8.30.

A simple example of the urbandet pseudocell is presented in Table 8.31 and another in Benchmark 52.

Table 8.30: *Elements and attributes for the* `<urbandet>` *pseudocell as a component of a* `<hub>`*.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<psentry>`	Designates the indexed `<psentry>` environment			
psID	Pseudocell ID	long	100000-200000	157843
percentarea	Percent area of parcel in impoundment (%)	Real	5.0-25.0	8.5
runoff	destination of runoff	String	Homecell or hub	hub
`<urbandet>`	Designates the `<urbandet>` environment			
rks	Seepage coefficient in eq. 8.38	Real	0.001-0.1	.007
`<vnotchweir>`	Designates the `<vnotch weir>` outlet structure			
wlen	Length of rectangular weir (m)	Real	0.0 10.0	7.5
angle	Angle of V-notch weir (deg)	real	0 120	75.0
top	Elevation of top of v-notch bleeder and invert of the rectangular weir (m, NGVD)	Real	0 20	13.6
apx	Elevation of invert of v-notch bleeder (m, NGVD)	Real	0 20	11.4

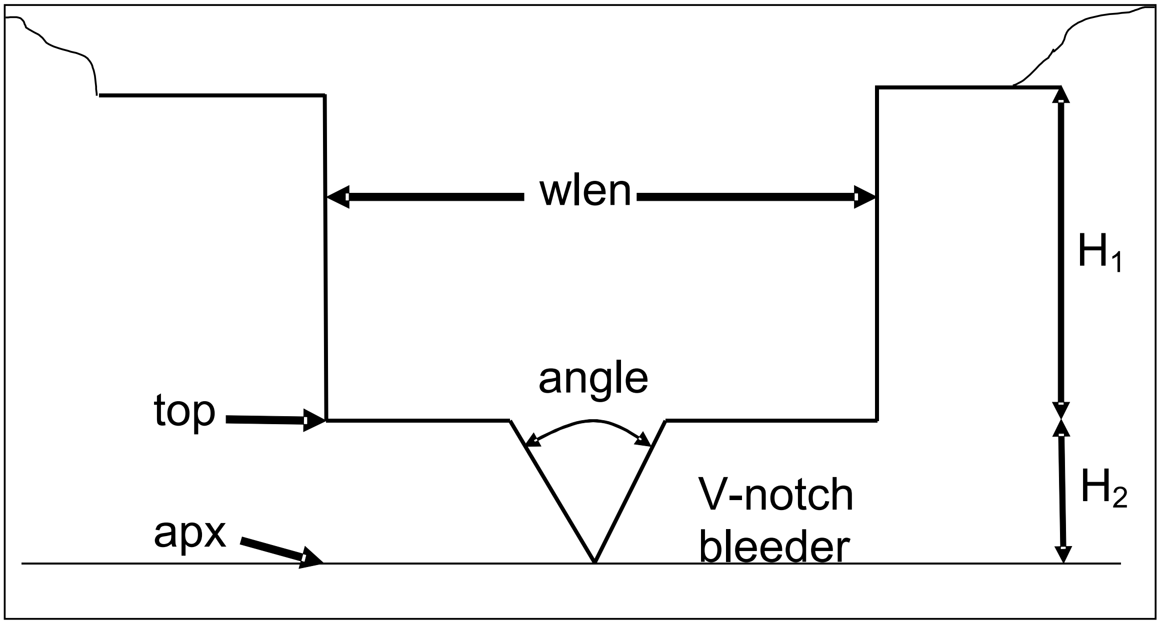


Figure 8.21: *Structure dimensions of urban detention pond discharge weir and bleeder.*

Table 8.31: *Example xml for* `<urbandet>` *pseudocell in a hub.*

```
<pseudocell>
   <indexed file="lu.index">

      <entry id="1">
        <hub runoff="homecell" wsupply="homecell" sewer="homecell">
         ...
          <psentry psID="23" percentarea="10.0" runoff="homecell">
             <urbandet rks="0.001">
               <vnotchweir wlen="10.0" angle="21.0" top="10.58" apx="9.75" />
             </urbandet>
          </psentry>
        ...
        </hub>
      </entry>
      ...
   </indexed>
</pseudocell>
```

Table 8.32: *Example index file for assigning pseudocells to mesh cells.*

```
DATASET
OBJTYPE "mesh2d"
BEGSCL
ND 18
NAME "landuse type"
TS 0 0.0
1
1
1
2
2
2
3
3
...
...
...
```

8.5 Additional Pseudocell Simulation Options

8.5.1 Assignment Of Pseudocells To Various Land Use Types `<indexed>`

Each cell in the model is assigned a particular type of pseudocell depending on the land use type. Cell ID's are used to assign cells or fractions of cells to the pseudocells as needed. An index is assigned for each pseudocell type, and an index file is used to associate the cell ID's with the pseudocell types. An indexed entry element `<indexed>` can be used where multiple pseudocells must be defined. To assign various pseudocell types to various cells, areal index maps are used. In using this method, a GMS type data file is used to assign various cell ID's to pseudocell type indices. The example in Table 8.32 below shows one of these index files used in Benchmark 14.

Table 8.33: *Example XML for implementation of kveg parameter modification.*

```
<pseudocell>
  <indexed file="lu.index">
     ...
    <entry id="3">
       <layer1nsm kw="1.0" rd="0.5" xd="2.0" pd="3.0" kveg="0.00">
              <ampmod para="kveg">
                   1   0.75
                  15   0.75
                  16   1.0
                365   1.0
              </ampmod>
       </layer1nsm>
    </entry>
    ...
  </indexed>
</pseudocell>
```

8.5.2 Time Variation of Pseudocell Parameters `<ampmod>`

Particularly for agricultural pseudocells it is appropriate to vary parameters with the season of the year. As an example, the `<afsirs>` pseudocell internally computes irrigation requirements as a function of the phase of the growing season. A parameter that might reasonably be varied seasonally is potential evapotranspiration. There is a mechanism available to allow for modifying selected parameters to take this effect into consideration. This option is currently implemented in the `<layer1nsm>`, `<layer5>`, and `<unsat>` pseudocells. The most commonly used parameter with seasonal variability is the vegetation crop ET coefficient, kveg. The seasonal variability is implemented using a keyword `<ampmod>` meaning ”amplitude modulation”. This allows the use of a 1-D lookup table to describe the variation of the parameter during the year, kveg as an example. In the following example in Table 8.33, kveg is 0.75 for the first 15 days of the year, and 1.0 for the rest of the year. The XML element is `<ampmod>` and the only attribute is ”para” that can be any parameter in the pseudocell. Following the ”para” attribute designation RSM will read a 1-D lookup table entered as text as in the example below. The text consists of pairs of numbers specifying ”serial day of the year” and ”multiplier for attribute”. The user must determine which parameters are appropriate candidates for annual variation.

Chapter 9

Input and Output File Specifications

Model input and output data can be in several formats. In this chapter the methods for specifying input data at a single location are described. Constant, repeating, and time series data are included.

Table 9.1: *Elements and attributes used to define a* `<const>` *input value.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<const>`	Designates a `<const>` input.			
dbintl	the time interval of the simulation in minutes	Real	60-2880	1440
value	Value if the input variable.	Real	Any real	1.34
mult	Multiplier for the value. may be used to change units.	Real	Any real	0.3048

9.1 Time Series and Other Data Formats Used For Single Location Model Input

Data at a single location in the model which is required for a boundary condition, rainfall, or evapotranspiration, may be input as a constant, a rating curve or as a time series. Examples where each of these could be appropriate include a constant water level in a water body where the level is maintained by pumping into or out of the water body, a rating curve that describes evapotranspiration that varies with the season but remains unchanged from year to year, and time series of daily rainfall for the duration of a multi-year model run. Constant data are input in the `<const>` environment, repeating time dependent input as a rule curve under `<rc>`, and time series data in the `<dss>`, `<csv>`, or `<asciiform>` formats.

9.1.1 Constant Value

A constant value can be specified with a very simple XML construction. The elements and attributes available are shown in Table 9.1

An example of XML input that specifies a constant `<refet>` of 0.14 with a multiplier of 0.3048 to convert from feet to meters is shown in Table 9.2

9.1.2 Rule Curve

A rule curve describes a variable that varies during a year and then repeats that behavior during each succeeding year. Common examples are reservoir operations for which there is a target headwater elevation for each season. For most large reservoirs in temperate climates,

Table 9.2: *Sample XML for specifying a constant* `<refet>`.

```
...
  <refet>
     <const dbintl="1440" value="0.14" mult="0.3048" </const>
  </refet>
...
```

for example, the water is maintained at a high level during the late spring and summer, and then lowered in the fall to furnish flood storage volume for the winter months when more runoff is expected. In SFRSM, rule curves may be used for seasonal evapotranspiration, irrigation requirements, and other variables. When used to input a variable, a rule curve is referenced by number. Rule curves are created in the `<rulecurves>` environment. The elements and attributes for defining and using a rule curve are explained in Table 9.3. An example showing the creation and use of a rule curve shown in Table 9.4 is used in Benchmark 52.

9.1.3 CSV and ASCIIFORM Time Series

Time series data can also be read from a comma delimited file under the element `<csv>`. This is simply an ascii file that contains pairs of (time,value) data such as shown in Table 9.5. An alternative ascii format is `<asciiform>` as shown in Table 9.6. The elements and attributes used for both `<csv>` and `<asciiform>` time series are given in Table 9.7. A sample XML input that species data in both `<csv>` and `<asciiform>` formats is shown in Table 9.8.

9.1.4 DSS Time Series

The most common time series format for single station data input has been the US Army Corps of Engineers (USACE) Hydrologic Engineering Center (HEC) DSS format. DSS stands for Data Storage System adopted by HEC. A file may contain many sets of time series data with each time series referenced by a path names. Each path name has six parts; A, B, C, D, E and F as described in Table 9.10. DSS files are described in detail in (Hydrologic Engineering Center, 1994). The elements and attributes used for specifying the data in a DSS file are explained in Table 9.9. An example of a DSS path name is shown below.

```
     A           B       C        D     E   F
/RED RIVER/BEND MARINA/FLOW/01JAN1975/1DAY/OBS/
```

Table 9.3: *Elements and attributes used to define a rule curve and to use it.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
Element and attribute for defining rule curves.				
`<rulecurves>`	Designates rule curves will be defined.			
`<rcentry>`	Designates that a particular rule curve will be specified.			
id	The ID number of the rule curve	Integer	any integer	3
label	A label to describe the rule curve.	String	Any string	seasonal water level
xunits	Units for the first column (time) in a 1D lookup table.	String	A valid DSS time interval	1DAY
cycle	The time length of the rule curve.	String	A valid DSS time interval	1YEAR
yunits	Units of the variable.	String	Any string	1YEAR
type	The type of data.	String	INST-VAL PER-AVER PER-CUM	INST-VAL
A table of x and y values in two columns to define the rule curve. See the example in Table 9.4				
Element and attribute for applying a rule curve.				
`<rc>`	Designates the data specified by a rule curve.			
id	The id number of the rule curve.	Integer	Any rule curve id	3

Table 9.4: *Sample XML for specifying a rule curve* `<rc>` *and using it in the specification of mesh boundary conditions..*

```
 ...
<rulecurves>
    <rcentry id="1" label="seasonal water level" xunits="1day"
             yunits="m" type="INST-VAL" cycle="1YEAR">
      1 498
     90 498
    120 500
    300 500
    330 498
    366 498
    </rcentry>
  </rulecurves>

  <mesh>
    <geometry file="mesh3x3.2dm"> </geometry>
    <mesh_bc>
      <wallhead section="gw">
        <nodelist> 1 2 3 4 </nodelist>
        <uniform> <rc id="1"></rc> </uniform>
      </wallhead>
      <wallhead section="gw">
        <nodelist> 13 14 15 16 </nodelist>
        <uniform><rc id="1"></rc></uniform>
      </wallhead>
    </mesh_bc>
 ...
 ...
```

Table 9.5: *Sample* `<csv>` *time series file.*

```
 ...
0.0, 3.5
1.0, 4.6
2.0, 5.3
3.0, 4.2
4.0, 3.8
 ...
 ...
 ...
```

Table 9.6: *Sample* `<asciiform>` *time series file.*

```
 ...
0.0, 3.5
1.0, 4.6
2.0, 5.3
3.0, 4.2
4.0, 3.8
 ...
 ...
 ...
```

Table 9.7: *Elements and attributes used for specifying* `<csv>` *and* `<asciiform>` *time series.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
Element and attributes for specifying `<csv>` time series file.				
`<csv>`	Designates a `<csv>` file.			
file	The name of the file	String	Any valid file name	Refet.csv
dbintl	Data time interval in minutes	Integer	any valid integer	1440
label	A label to describe the file.	String	Any string	Refet at Miami
Element and attributes for specifying `<asciiform>` time series file.				
`<asciiform>`	Designates an `<asciiform>` file.			
file	The name of the file	String	Any valid file name	Rain.dat
format	The format of the data.	String	A valid C-style format string	%10.2lf

Table 9.8: *Sample* `<csv>` *and* `<asciiform>` *XML input.*

```
...
   <refet>
     <csv file="refet.csv" label="Miami station"dbintl="1440">
   </refet>

   <rain>
     <asciiform file="rain.dat" format="%10.2lf">
   </rain>
...
...
...
```

The same path name with the optional parts omitted still requires the slashes (/) to be a valid path name.

```
        B        C     E
 //BEND MARINA/FLOW//1DAY//
```

In SFRSM a number of variable types are assigned default units. If other units are to be used, the ”multiplier” option needs to be used to convert the units as appropriate. Head measurements for example use METERS as the default unit with type INST-VAL. The remaining unit sets are shown in Table 9.11.

Table 9.9: *Elements and attributes used for specifying time series data in a* `<dss>` *file.*

Element or Attribute	**Definition**	**Variable type**	**Suggested range**	**Example**
`<dss>`	Designates a `<csv>` file.			
file	The name of the DSS file	String	Any valid DSS file name	RRFlow.dss
pn	The DSS path name of the times series in the DSS file	string	a valid DSS path name	/REDRIVER/BEND MARINA/FLOW/ 01JAN1975/1DAY/OBS/
mult	A multiplier for the data	Real	Any Real	0.02831685
units	Units of the variable	string	Any string	cfs
type	??	string	??	??

Table 9.10: *Path name definition for time series data in DSS format.*

Part	Description	Format	Acceptable values
A	Basin or project name. (optional)	String	Any string
B	Location or gage identifier (required).	String	Any String
C	Data variable or parameter.	String	FLOW, STG, FLOW-CUM, ELEV, STAGE, PH, PRECIP, etc (required).
D	Starting date for block data	ddmmmyyyy	01JAN1981 (optional)
E	Time interval.	String	1MIN, 2MIN, 3MIN, 4MIN, 5MIN, 10MIN, 15MIN, 20MIN, 30MIN, 1HOUR, 2HOUR, 3HOUR, 4HOUR, 6HOUR, 8HOUR, 12HOUR, 1DAY, 1WEEK, 1MON, 1YEAR (required)
F	Additional user-defined data	String	Any String (optional)

Table 9.11: *Default units used by the RSM model.*

Quantity	Unit	Type
Head	METERS	INST-VAL
Flow	CU_METER	
Rain	METERS	PER-CUM
ET	METERS/time step	PER-CUM
Depth	METERS	INST-VAL
Water level	METER	INST-VAL
Transmissivity	$METER^2/SECOND$	PER-AVER

Chapter 10

RSM Post-Processing

This chapter contains four topics related to post-processing RSM, including:

1. A review of RSM water budgets for water bodies and water movers (Section 10.1). A local and global water balance discussion is also provided in this chapter.
2. RSM output options (Section 10.2) specified in the XML input files
3. RSM uncertainty analysis (Section 10.3)
4. RSM graphical user interface (Section 10.4 currently under development)

10.1 Water Balance And Budgets

Keeping track of water is a basic responsibility of the water bodies and the water movers, which are the basic building blocks of the HSE. Water bodies, regardless of their sizes or shapes, track how much water is contained in them at the end of every time step. Similarly, water movers regardless of their size or shape should know the volume of water that pass through them. All the water budgets are tied to the governing equations and the finite volume method.

$$\nabla \mathbf{A} . \frac{\partial \mathbf{H}}{\partial t} = \mathbf{Q}(\mathbf{H}) + \mathbf{S} \tag{10.1}$$

in which, $\mathbf{Q}(\mathbf{H})$ = flows into the water bodies in vector form; $\mathbf{S}$ = water entering water bodies through recharge. Recharge occurs after ET, rain, unsaturated flow storage, etc. and all are taken into account.

Water in a water body can be stored in a saturated compartment and in a pseudocell compartment. The saturated water is used with the stage-volume (SV) relationship to compute the water level. Saturated ground water, canal water, lake water and overland flow water all fall into this category. Pseudocells represent the local hydrology which may account for water above the water table. Water in the pseudocell takes into account the unsaturated water, urban detention, agricultural residue, etc. This water does not relate to the water level in the regional system.

Within a time step in the computations, the following water balance equation can be written for the total water content.

$$V_s^{(n+1)} + V_p^{(n+1)} - V_s^{(n)} - V_p^{(n)} = Q_r + Q_d + Q_s + Q_i + Q_b \tag{10.2}$$

The components are defined in Table 10.1 and Table 10.2.

10.1.1 Water Budgets Of Water Bodies

Water bodies have heads H associated with them, which drive the water movers. The volume of water in a cell has two water budget components due to their contribution from the saturated cell and the pseudocell. Only the saturated water is related to the head in the water body. The reported components of water are listed in Table 10.1.

10.1.2 Water Budgets Of Water Movers

Water movers provide the only way to move water in and out of a water body. Any water moving through a water mover is accounted for. The reporting categories of moving water are divided into the following categories.

Table 10.1: *Reported water budget components of a water body.*

Component	Variable	Definition
`<saturated>`	V_s	This is the total volume of water in the water body below the free surface. This water includes saturated ground water and overland flow water. The SV converter can be used to compute the relationship between this volume and the head.
`<pseudo>`	V_p	This is the volume of water in the water body that is not in the saturated head dependent water body. This volume is made up of unsaturated water, detention ponds, and water in the process of being routed in urban cells.

Table 10.2: *Reported water budget components of a water mover.*

Component	Variable	Definition
`<recharge>`	Q_r	Volume of water entering the saturated compartment of the water body from the pseudocell compartment as a result of the internal processes of the pseudocell. Rainfall, ET, and unsaturated flow storage and other local hydrological functions enter into the computation.
`<drainage>`	Q_d	Volume entering the saturated compartment of the water body as a result of non-recharge type or planned releases from its own or other pseudocells. Such releases take place in urban and agricultural areas. This water adds to the source term in the computations.
`<srcbnd>`	Q_s	Volume of water entering into the saturated water body as a result of pumping, and similar source types that are entered as source/sink boundary conditions.
`<inflow>`	Q_i	Volume of water that enters the saturated part of the water body through the water movers. This is known at the end of the time step.
`<borrow>`	Q_b	Volume of water entering into the saturated water body after the entire horizontal flow computations are complete, when the water left in a water body is negative. This is common when the canals are dry and pumping continues.

10.1.3 Local and Global Mass Balance

The governing equations solved by RSM are conservative and mass balance should be achieved both locally and globally in well-posed problems. However, the use of constant head boundary condition for a water body (cell, segment or a lake) can cause a mass imbalance in the model because it cause the entire row of the solution matrix to be replaced by a row of zeros and a diagonal term during the assignment of the boundary condition. The process of elimination of terms destroys the integrity of the water movers, which guarantee the mass balance of RSM. This affects the mass balance of surrounding cells, which rely on water movers for the flow information. This may, however, not affect the mass balance of far away cells. The replacements for head boundary conditions for cells or segments can be wall or node based, respectively.

10.2 RSM Output Options `<output>`

A number of model output options are available for both state variables and parameters using the `<output>` option. These options can be classified into three categories.

1. Under the first category of options, a comprehensive export of a selected set of variables is possible. This output is in GMS format and can be used for GMS or TECPLOT animations. This type of export will include all the cell, segment, or lake values at all times.
2. Under the second category, a netCDF export is possible. This file can be post-processed using a budget package/budget tool. The budget tool gives a balance of budget components as well as other debugging capabilities indicating whether some of the model objects were ever created.
3. Under the third category, it is possible to export various monitors for a selected set cells at a selected intervals. This option can be used to focus on small areas.

The list of all available options under these three categories is shown in Table 10.3. The output file formats available under most of the output options is listed in Table 10.4. Details on some of the most important options are described below, with examples.

Table 10.3:*Model output options available using* `<output>`*.*

Elements	Description
`<globalmonitor>`	Used to dump all the values of a variable for the entire duration of the model run. The list of attributes available under this option are in Table 10.5. The format used can be `<gms>` with file name `file = "myfile.dat"` or any other format in Table 10.4.
`<budget>`	Options available are as in `budget="Lm43" dbintl="10080"`.
`<budgetpackage>`	The entire water budget is dumped into a designated netCDF file for post processing. A program ”psbud” is used to for this purpose. The only options under `<budgetpackage>` are file name and data base interval as in `budgetpackage="C4.nc" dbintl="1440"` showing output data interval.
`<psbudgetpackage>`	The entire water budget is dumped into a designated netCDF file to be used later in post processing. A program ”psbud” is used to for this purpose. The options available under `<budgetpackage>` are as in `budgetpackage="C4.nc" dbintl="1220"`.
`<cellreport>`	The option available is `file="mon.dat"`
`<cellmonitor>`	Useful in monitoring a large number of attributes. The options available are cell ID `<id>` and `<attr>`. The list of attributes is shown in Table 10.6.
`<segmentmonitor>`	Useful in monitoring a large number of segment attributes. The options available are segment ID `<id>` and `<attr>`. The list of attributes is shown in Table 10.7.
`<junctionmonitor>`	Useful in monitoring flow in junctions. The available options are `<id1>`, `<id2>` and `<attr>`. The list of attributes is shown in Table 10.8.
`<wmmonitor>`	Useful in monitoring water movers. The available options are `<wmID>` and `<attr>`. The list of attributes is shown in Table 10.9.
`<bcmonitor>`	Useful in monitoring boundary conditions. The available options are `<bcID>` and `<attr>`. The list of attributes is shown in Table 10.10.

Table 10.3 continued on next page

Elements	**Description**
`<lakemonitor>`	Useful in monitoring lakes. The available options are `<idD>` and `<attr>`. The list of attributes is shown in Table 10.11.
`<assessormonitor>`	Used to monitor assessors. The attributes include `<ormid>`, `<aid>` and `<attr>`. The list of attributes is shown in Table 10.12.
`<ctrlmonitor>`	Used to monitor controllers. The attributes include `<wmID>` and `<attr>`. The options available under `<attr>` are described in Table 10.13.
`<flowgage>`	Useful in monitoring flow across flow lines. The available options are `<section>` and `<nodelist>`. The list of `<section>` options included are shown in Table 10.14. The keyword `<nodelist>` is a list of nodes defining the flow line.
`<pseudomonitor>`	Useful in monitoring attributes within the pseudocells. The available options are `<id>` and `<attr>`. The list of attributes is shown in Table 10.15.

Table 10.4: *Time series formats available within the output options in Table 10.3.*

Element	Description
`<gms>`	GMS format, to be used as in `<gms file="c51.gms" />`
`<netcdf>`	netCDF format, to be used as in `<netcdf file="c51.nc" dbintl="18000" />`
`<dss>`	DSS format, to be used as in `<dss file="c51.dss" dbintl="18000" />`
`<csv>`	Comma separated ASCII format, as in `<cvs file="c51.cv" dbintl="18000" label="My name" />`
`<ascii>`	ASCII format, as in `<asciiform file="c51.txt" format="%5d %5d %5d" />`, which outputs the date (year month day) and the value in simple ASCII format. Any c-style formats are allowable, but floating point output formats are normally used such as (%lg %10.2lf %f).

10.2.1 Saving Model Output `<globalmonitor>`

The head, velocity and a number of other variables (Table 10.5) in the entire domain for the entire duration can be output into an ASCII file in GMS format using `<globalmonitor>` option. Table 10.3 shows the file formats available for saving the file. The output information can be used to create animations using GMS software or TECPLOT software after post processing using the hse2tec program. The following is an example of a data input included in the XML file to obtain both head and velocity data sets in GMS format.

```
..
  <output>
    <globalmonitor attr="totalvector">
       <gms file="outvect.dat"> </gms>
    </globalmonitor>
    <globalmonitor attr="head">
       <gms file="outheads.dat"> </gms>
    </globalmonitor>
  </output>
```

10.2.2 Water Budget Post-Processing

Considering that a large number of variables are involved in water budget calculations, this task is often most suitable outside of the model as a post-processing exercise. The first step in budget post processing is to create a netCDF file from the model run. This is accomplished using `<budget>`, `<budgetpackage>` or `psbudgetpackage>` as shown in Table 10.3. These tags will create the netCDF files at the requested time interval in minutes specified

Table 10.5: *Attributes available with* `<globalmonitor>`. *The usage is: <globalmonitor attr="attribute"> <filetype in Table 10.3 > < /globalmonitor>*

Global Monitor

Element	Description
`ponding`	Ponding.
`wtdepth`	Ponding.
`head`	Water head in cells
`recharge`	Recharge into cells
`runoff`	Runoff from cells
`wsupply`	Water supply in cells
`rain`	Rain in cells
`refet`	Reference ET in cells
`rainvol`	Rainfall volume in cells
`rchgvol`	Recharge volume in cells
`watercontent`	Water content in cells
`wcvol`	Water content in cells
`inflow`	Inflow into cells
`waterlevel`	Water level in cells
`sy`	Storage coefficient
`transmissivity`	Transmissivity

Table 10.6: *Variables that can be monitored using* `<cellmonitor>`*. The usage is: <cellmonitor id="cellid#" attr="attribute"> <filetype in Table 10.3 > < /cellmonitor>*

Cell Monitor

Attribute	Name	Description	Assessor
`ponding`	PondDepth	depth of water above land surface	Cell::Ponding()
`wtdepth`	WTDepth	depth to water table	Cell::Wtdepth()
`head`	ComputedHead	elevation of water table	Cell::Head()
`recharge`	Recharge	volume of recharge received from pseudocell	Cell::RechargeVolume()
`runoff`	RunoffVolume	volume of runoff received from pseudocell	Cell::RunoffVolume()
`wsupply`	Westinghouse	volume of water supply withdrawals by pseudocell	Cell::Westinghouse()
`rain`	Rainfall	depth of rainfall	Cell::Rainfall()
`rainvol`	RainfallVolume	volume of rainfall	Cell::RainfallVolume()
`rchgvol`	RechargeVolume	volume of recharge received from pseudocell	Cell::RechargeVolume()
`watercontent`	WaterContent	depth of saturated water content	Cell::WaterContent()
`initvol`	InitVol	volume of initial water content	Cell::InitVol()
`wcvol`	WCVolume	volume of saturated water content	Cell::SatVol()
`topo`	Topography	land surface elevation	Cell::LandSurface()
`inflow`	Inflow	volume of inflow from adjacent water bodies in previous time step	Cell::Delta()
`waterlevel`	WaterLevel	head - land surface	Cell::WaterLevel()
`sy`	SpecificYield	specific yield (depth fraction)	Cell::Sy()
`transmissivity`	Transmissivity	aquifer transmissivity	Cell::TransValue()

Table 10.7: *Variables that can be monitored using* `<segmentmonitor>`. *The usage is: <segmentmonitor id="segmentid#" attr="attribute"> <filetype in Table 10.3 > < /segmentmonitor>*

Segment Monitor

Attribute	Name	Description	Assessor
head	segmenthead	elevation of water level	Segment::Head()
segmenthead	segmenthead	elevation of water level	Segment::Head()
depth	segmentdepth	depth of water	Segment::Depth()
segmentdepth	segmentdepth	depth of water	Segment::Depth()
sbflow	sbflow	sum of all streambank flow in segment	Segment::SBFlow()
seepageflow	seepageflow	streambank flow – aquifer seepage	Segment::SeepageFlow()
overbankflow	overbankflow	streambank flow – overbank flow	Segment::OverbankFlow()
lev1flow	lev1flow	streambank flow – type 1 levee seepage	Segment::Lev1Flow()
lev2flow	lev2flow	streambank flow – type 2 levee seepage	Segment::Lev2Flow()
sbvolume	sbvolume	sum of all streambank flow volumes	Segment::SBVolume()
seepagevolume	sbvolume	streambank flow volume – aquifer seepage	Segment::SeepageVolume()
overbankvolume	sbvolume	streambank flow volume – overbank flow	Segment::OverbankVolume()
lev1volume	sbvolume	streambank flow volume – type 1 levee seepage	Segment::Lev1Volume()
lev2volume	sbvolume	streambank flow volume – type 2 levee seepage	Segment::Lev2Volume()
initsegsto	InitSegSto	initial volume of storage	Segment::InitVol()
segsto	SegmentStorage	volume of storage	Segment::SatVol()

Table 10.8: *Variables that can be monitored using* `<junctionmonitor>`. *The usage is: <junctionmonitor id1="segment1 id#" id2="segment2 id#" attr= "attribute"> <filetype in Table 10.3 > < /junctionmonitor>*

Network Junction Monitor

Attribute	Name	Description	Assessor
flow	JunctionFlow	volume of flow through junction	WaterMover::ReportFlow()

Table 10.9: *Variables that can be monitored using* `<wmmonitor>`.*The usage is: <wmmonitor id1="segment1 id#" attr= "attribute"><filetype in Table 10.3 > < /wmmonitor>*

Watermover Monitor

Attribute	Name	Description	Assessor
`flow`	WaterMoverFlow	volume of flow through water mover	WaterMover::ReportFlow()

Table 10.10: *Variables that can be monitored using* `<bcmonitor>`. *The usage is: <bcmonitor bcID="bcid#" attr="attribute"> <filetype in Table 10.3 > < /bcmonitor>*

Boundary Condition Monitor

Attribute	Name	Description	Assessor
`flow`	BCVolume	volume of flow through boundary	ExternalBC::ReportFlow()
`head`	BCHead	head applied at boundary	ExternalBC::Value()

Table 10.11: *Variables that can be monitored using* `<lakemonitor>`. *The usage is: <lakemonitor id="lakeid#" attr="attribute"> <filetype in Table 10.3 > < /lakemonitor>*

Lake Monitor

Attribute	Name	Description	Assessor
lakesto	LakeStorage	volume of lake storage	Lake::SatVol()
initlakesto	InitLakeSto	initial volume of storage	Lake::InitVol()
head	LakeHead	elevation of water level	Lake::Head()

Table 10.12: *Variables that can be monitored using* `<assessormonitor>`. *The usage is: <assessormonitor ormid="ormid#" aid="aid#"attr="attribute"> <filetype in Table 10.3 > < /assessormonitor>*

Assessor Monitor

Attribute	Name	Description	Assessor
`depth`	AssessorDepth	depth of water in waterbody	Assessor::Depth()
`volume`	AssessorVolume	volume stored in waterbody	Assessor::Volume()

Table 10.13: *Variables that can be monitored using* `<ctrlmonitor>`. *The usage is: <ctrlmonitor wmID="wmid#" attr="attribute"> <filetype in Table 10.3 > < /ctrlmonitor>*

Control Monitor

Attribute	Name	Description	Assessor
error	ControlError	error	Controller::ReportError()
control	ControlOutput	output	Controller::ReportControlOut()
state	ControlState	state variable	Controller::ReportStateIn()
maxflow	ControlMaxFlow	flow	Controller::ReportMaxFlow()

Table 10.14: *Variables that can be monitored using* `<flowgage>`. *The usage is: <flowgage section= "attribute"> <filetype in Table 10.3 > < /flowgage>*

Flowline Monitor

Attribute	Name	Description	Assessor
ol	OverlandFlow	volume of overland flow across defined flowline	FlowGage::OverLand()
gw	GroundwaterFlow	volume of groundwater flow across defined flowline	FlowGage::GroundWater()
ol_gw	TotalFlow	volume of total flow across defined flowline	FlowGage::Total()

Table 10.15: *Variables that can be monitored using* `<psmonitor>`*. The usage is: <pseudomonitor id="pseudocell id#" attr= "attribute"> <filetype in Table 10.3 > < /pseudomonitor>*

Pseudocell Monitor

Attribute	Name	Description	Assessor
ps_celldeltavol	PS_CellDeltaVolume	volume of inflow	PseudoCell::CellDeltaVolume()
ps_rechargevol	PS_RechargeVolume	volume of recharge to homecell	PseudoCell::RechargeVolume()
ps_et	PS_Et	depth of evapotranspiration	PseudoCell::Et()
ps_rain	PS_Rain	depth of rainfall	PseudoCell::Rain()
ps_etvol	PS_EtVolume	volume of evapotranspiration	PseudoCell::EtVol()
ps_rainvol	PS_RainVolume	volume of rainfall	PseudoCell::RainVol()
ps_watercontent	PS_WaterContent	depth of storage	PseudoCell::WaterContent()
ps_wcvol	PS_WCVolume	volume of storage	PseudoCell::WCVolume()
ps_wsupply	PS_WaterSupply	depth of water imported for water supply	PseudoCell::WSupply()
ps_wslocal	PS_WSLocal	depth of water withdrawn from homecell for water supply	PseudoCell::WSLocal()
ps_runoff	PS_Runoff	depth of runoff	PseudoCell::Runoff()
ps_runoffvol	PS_RunoffVolume	volume of runoff	PseudoCell::RunoffVolume()
ps_seepagevol	PS_SeepageVolume	volume of seepage	PseudoCell::SeepageVolume()
ps_wsupplyvol	PS_WSupplyVolume	volume of water imported for water supply	PseudoCell::Westinghouse()
ps_cuvol	PS_CUVolume	volume of import consumptive use	PseudoCell::CUVolume()
ps_sewervol	PS_SewerVolume	volume of consumptive use return flow	PseudoCell::SewerVolume()
ps_septicvo	PS_SepticVolume	volume of consumptive use return flow – septic	PseudoCell::SepticVolume()
ps_wslocalvol	PS_WSLocalVolume	volume of water withdrawn from homecell for water supply	PseudoCell::WSLocalVolume()
ps_initvol	PS_InitVol	initial volume of storage	PseudoCell::InitVol()

using `<dbintl>`.

```
 <output>
..
     <psbudgetpackage file="pseudo.nc" dbintl="10080"></psbudgetpackage>
     <psbudgetpackage file="pseudo_yr.nc" dbintl="525600"></psbudgetpackage>
     <budgetpackage file="budget.nc" dbintl="10080"></budgetpackage>
 </output>
```

in which, the file `"budget.nc"` is set to record time series data at 10080 minutes or 7 day intervals.

Once the netCDF file is created, the water budget post-processor program ”psbud” is used to create a water budget for a cluster of cells listed in a ASCII file. When the command is used as shown below, it is assumed that there is access to the code in ”../../psbud/”.

```
../../psbud/psbud -n pseudo.nc -s subset.unsat > unsat.out
```

The file `"subset.unsat"` is simply an ASCII file with the required cells listed as shown below.

```
1
10
32
```

The following shows part of the output file `"unsat.out"` from the budget tool. In the file, the first three columns show the dates Jan 2, Jan 7, etc. indicating that the output is at the data base interval of 10800 min or 7 days as given in the `<budgetpackage>` example shown above.

```
              Rainfall     Et   CellDelta WSupply   CU  Sewer   Septic  Runoff  Seepage  Residual
                   M^3    M^3     M^3   M^3        M^3    M^3      M^3     M^3      M^3       M^3
1965 1  2 24  0  57150  74136       0     0          0      0        0       0   1341.8        32
1965 1  9 24  0      0  1.4208e+05  0     0          0      0        0       0    19742         0
```

The columns of the table show that various water budget components add to create a very small residual. The quantities shown are water budget volumes balanced during the time period.

10.2.3 Monitoring Individual Points

When individual objects of the model are to be monitored for state variables, a number of monitoring options are available through `<cellmonitor>`, `<segmentmonitor>`, etc. as described in Table 10.3. The attributes available for monitoring using these tags are described in the tables listed in Table 10.3. The following example shows a cell head monitor, lake head monitor, segment head monitor and watermover flow monitor:

```
<output>
..
  <cellmonitor id="6" attr="head">
    <csv file="head.csv" label="head"></csv>
 </cellmonitor>

 <lakemonitor id="101" attr="head"><csv file="stages.csv"
    label="lo" dbintl="43200"></csv></lakemonitor>

    <segmentmonitor id="22" attr="head">
      <dss file="t3x3out.dss" pn="/hse/segment_4/head//1day/calc/">
      </dss>
    </segmentmonitor>

    <wmmonitor wmID="10" attr="flow">
      <dss file="t3x3out.dss" pn="/hse/wm10_P_78/flow//1day/calc/">
      </dss>
    </wmmonitor>

</output>
```

10.3 RSM Uncertainty Analysis

A limited number of tools are provided with RSM to carry out uncertainty analysis related to the model. A limited discussion is presented here to explain the scope of the tools associated with RSM. Uncertainty is a key link in the chain connecting the raw data in a hydrologic system and the decisions made regarding issues such as construction or operation of components. Factors influencing the decision include the following.

(A) *Input uncertainty*: One of the factors outside the model itself that affects a decision is the uncertainty of the data. Most hydrologic data has errors associated with them due to equipment failures, recording errors, interpretation errors or a number of other causes. Lack of spatial resolution of input data is also part of the same problem. An example is the poor resolution of the rainfall and ET data sets.

(B) *Parameter uncertainty*: Uncertainty associated with a model itself can be partly due to parameter uncertainty. Poor parameter data or lack of calibration can be the reason for such uncertainty. Lack of spatial resolution in the available parameter data set is also part of the same problem.

(C) *Algorithm uncertainty*: Uncertainty associated with numerical error resulting from truncation error, inappropriate discretizations, poorly selected algorithms etc. can result in this type of uncertainty.

(D) *Forecasting uncertainty*: This is the uncertainty of not knowing the future data set for which the decisions are to be made. Rainfall for the next 30 years can be different from the previous 30 years, and decisions may substantially change if long periods of data are available.

(E) *Operational uncertainty*: Even if decisions are made considering past operational rules, and assuming rules for the present, exact future or past operational rules can never be completely known or modeled. This places an additional uncertainty on the model output.

(F) *Performance measure uncertainty*: This uncertainty indicates that even if the hydrology of the system can be established using a model simulation, the ecological or environmental implications may not be known with certainty. As a result, the actual implications of a proposal may be substantially uncertain.

Among the six types of uncertainty discussed, parameter uncertainty (B) and the algorithm uncertainty (C) are the only types investigated at some depth within SFWMD. These and operational uncertainty (E) are the only types closely associated with the model. The remaining uncertainties are external to the model. Other activities associated with hydrologic model uncertainty in South Florida include: (a) conducting a model uncertainty workshop by (Loucks et al., 2002); (b) results from ***Lal (1996) and Trimble (1997)***(kcb note: don't have these ref's) on parameter uncertainties of the SFWMM and the NSM based on first order, Latin hypercube and Rosenbleuth methods; (c) workshop on model uncertainty by (Loucks et al., 2002). The method by ***Trimble (1997)*** and (Lal, 1998a) involved disturbing regional parameters in a systematic way, and determine the sensitivity matrix. There are some documents available at the SFWMD describing previous studies carried out on the topic.

Algorithm uncertainty was investigated by (Lal, 2000c) using methods related to stability and spectral analysis. The primary parameters used in the analysis are the dimensionless time step and cell size. The effects of these parameters on numerical error and run times were investigated during the study.

The remaining types of uncertainty, and how to deal with them in the decision making process remain unknown to a great extent. However, several decision making methods can be adopted to avoid having to evade this problem.

10.3.1 Existing Capabilities For Evaluating RSM Uncertainty

The current capabilities of the RSM toolbox includes the following as described by (Lal, 1995) and (Lal, 2000a).

1. Tools for parameter sensitivity analysis using the first order method, and creating a sensitivity matrix
2. Tools for determining singular values and vectors of the sensitivity matrix. This tool is capable of determining the most significant parameter groups.
3. Tools to determine the model output uncertainty using a known input parameter data set.
4. Tools to determine the parameter covariance matrix, which gives the uncertainty of the raw parameters in the model, for unit output uncertainty.
5. Tools to determine the parameter correlation matrix, parameter resolution matrix and data ignorance matrix
6. Tools to determine the numerical error associated with a known cell size and a time step.

10.3.2 Methods Available For Evaluating Model Results

Considering that full evaluation of a model run and making decisions on its merits is an immense task, a number of interim approaches have been used in the past. Some of these approaches are described below.

10.3.3 Evaluation Based On The Significance Of Differences

One of the approaches is to compare alternatives with the same base data set. The purpose of this method is to determine the significance of the differences between two proposed scenarios, and make a decision about the incremental benefits. Methods needed to statistically quantify the significance of the difference are not fully developed. The advantage of this method is that it does not rely heavily on the accuracy of the past data. This data is assumed as a standard data set against which all runs are compared.

10.4 RSM Graphical User Interface

Currently, the RSM GUI is undergoing significant development. At this time, no up-to-date documentation is available for this manual. RSM GUI information will be published by the SFWMD when code development is completed.

10.4.1 Overview of The Current RSM GUI

The existing RSM pre- and post-processing GUI has significant capabilities. These include:

1. Visualization of model mesh and canal segments
2. Color-flood display of cell-by-cell values and segment-based values for any value stored in NetCDF
3. Support for more than one NetCDF file (e.g. one of heads and one of fluxes) in the same GUI run
4. Postscript / PDF / PNG dumps of color-floods
5. Shapefile (vector) basemaps
6. Fast zoom/pan navigation
7. Navigation of time steps (forward / back / fast-forward / fast-back)
8. Conversion of time-step to time-stamp (and back) with ”jump to this time step” feature
9. Continuous feedback dialog – updates current color flood variable, cell / segment ids, spatial coordinates as mouse moves
10. Tool for computation and color flood of hydroperiod
11. Selection of data subsets for computations, based on start/end time
12. Selection of data subsets for computations based on digitized polygons
13. ”Movie” display – steps through model time steps (optionally drops PNG scenes on disk for AVI generation via e.g. ImageMagick)
14. Calibration tool – reads an XML file of an RSM <observations> container, then allows user to select a station for review. Computes calibration statistics with optional plotting via xmgrace (with VPython support planned for next version, see below)
15. Well-documented ”plug-in” interface for adding tool functionality

The features currently included in the RSM GUI are:

1. Visualize Mesh
2. Color Flood Display
3. Multiple NetCDF files
4. Postscript/PDF/PNG
5. Vector Basemaps
6. Fast Zoom/Pan
7. Nav. Timesteps
8. Convert. Timesteps
9. Feedback Dialog
10. Color flood Hydroperiod
11. Start/end selection
12. Select Data by Polygon
13. Movie Display
14. Calibration Tool
15. Plug-in Interface

10.4.2 Overview of The Early 2005 RSM GUI Development Activities

Other GUI features are currently undergoing development including:

1. Hydrograph generation for current colorflood variable arbitrary cells / segments
2. Dynamic display of canal stages along a canal reach (via VPython / OpenGL)
3. Support for DSS files in calibration datasets
4. Support for ”comparison” models – loads similar NetCDF files from several model runs, color-floods differences or comparative statistics for the various runs. Note that hydrograph, etc. are supported, based on the stats

5. Computation of summations and averages of NetCDF values based on spatial regions
6. Display of structures (including mouse-over feedback dialog)

Features to be added to RSM GUI are:

1. Hydrograph generation
2. Dynamic Display
3. DSS Support
4. Comparison Models
5. Summations and Averages
6. Mouseover Structures

Chapter 11

RSM Validation

A series of validation benchmarks exist for RSM. As development continues, new benchmarks are created. All existing benchmarks are described in Section 11.1.

11.1 Model Validation Benchmarks And Test Cases

All benchmarks can be accessed with hyperlinks [1]. A complete listing of all benchmarks described in detail is available here[2].

Table 11.1: *Benchmarks established for the HSE, hyperlinks yield full descriptions*

---

[1] http://gwmftp.jacobs.com/benchmarks/BM1/BM1.pdf, for example

[2] http://gwmftp.jacobs.com/benchmarks/bm_des.pdf

Hyperlink	**Mesh/Network**	**Feature Tested**
BM1	3x3	Overland flow
BM2	3x3	Overland and groundwater flow
BM3	canal	Canal flow
BM4	3x3/canal	Overland flow, gw flow, canal flow, and streambank.
BM5	3x3	Single_control watermovers
BM6	3x3	Steady state solution
BM7	3x3	Pumping wells
BM8	3x3	5-layer pseudocell
BM9	3x3/canal	Dual_control watermover
BM10	3x3/canal	Head boundary conditions
BM11	Pinder	GW flow, canal flow and mesh to canal interaction
BM12	3x3	General head boundary conditions
BM13	3x3	Lakes and ponds
BM14	3x3/canal	Culverts
BM15	3x3	Indexed entry of pseudocells
BM16	3x3	Nsm1layer pseudocell, w/ampmod feature
BM17	3x3	Svconverter lookup table
BM18	3x3	Unsat pseudocell
BM19	3x3	Output options, including netcdf
BM20	3x3/canal	Single_control watermover (segment h20)
BM21	3x3/canal	Single_control watermover (cell h20)
BM22	3x3/canal	MBR pipes
BM23	3x3/canal	Three MBR weirs
BM24	3x3	Indexed entry of rain and refet
BM25	3x3	Mbrcell pseudocell
BM26	3x3/canal	Three MBR bleeders
BM27	3x3/canal	Canal streambank implementation
BM30	ENP	Include external files, mesh and network bc's
BM31	ENP	Separate type conveyance , like BM30
BM33	3x3	Afsirs pseudocell
BM34	L8	Wts2pt wallhead, various conveyance formulations

Table 11.1 continued on next page

Hyperlink	**Mesh/Network**	**Feature Tested**
BM35	3x3	General head boundary imposed on walls
BM36	3x3	Lookup tables for conveyance and transmissivity
BM37	3x3	Lookup table for soil storage coefficient
BM38	3x3	Kadlec formulation for conveyance
BM40	canal	Pidctrl controllers
BM41	canal	Setpointctrl controllers
BM42	canal	GHB boundary conditions
BM43	canal	Fuzctrl controllers
BM44	canal	Upwind methods in overland and canal flows
BM45	canal	User Defined controller
BM47	3x3	GLPK optimization problem
BM48	3x3	MSE network rep. and HSE to MSE network mapping
BM49	3x3	MSE network rep. and HSE to MSE network mapping
BM50	3x3	Pseudocell hub
BM51	3x3	Impervious pseudocell
BM52	3x3	Urbandet pseudocell with transient wallhead bc's
BM53	3x3	Urbanhub pseudocell with urbanhub feature
BM54	3x3	Urbanhub runoff and wsuppy routing
BM55	3x3	Urbanhub consumptive use and return flow
BM56	3x3	Precipitation runoff model (nam or prr) pseudocell
BM57	3x3	One2many, pumpedditch and agimp
BM58	3x3	Lake boundary conditions

11.2 RSM Peer Review Findings and Suggestions

The SFWMD is planning a three part peer review of RSM. Part I of III is scheduled to occur between March 15, 2005 and July 31, 2005. The statement of work for the Part I peer review is available here [3]. This review will principally investigate HSE but will also consider the MSE. The part II review is anticipated to occur in late 2005 and will cover validation of the Natural System Regional Simulation Model(NSRSM). The part III review is anticipated to occur in 2006 and will cover calibration of the South Florida Regional Simulation Model(SFRSM). Findings of the peer review will be available as separate documents.

---

[3] http://gwmftp.jacobs.com/Peer_Review/final_sow_district/PeerReviewSOW-020805.pdf

Chapter 12

Overview Of The Management Simulation Engine

The MSE has been an active area of research during 2004 and 2005. For this reason, this manual only presents an overview of the MSE approach in the following section and the user is directed to find the MSE details in the following documents.

1. MSE Controllers User's Manual. This document is needed when the MSE controllers are implemented in a model. Controllers can locally control structures so that water levels and other state variables can reach specified target values. Controllers provide lower level management in MSE. The Controller User's Guide [1] is available on-line.
2. MSE Supervisors User's Manual. This document describes the higher level management capabilities of the MSE, and use of LP in setting up rules and objectives or goals of a model scenario. The Supervisor User's Guide [2] is available on-line.
3. The on-line HSE Data Input Guide. This resource allows user's to navigate through the XML input data structure. It has variable definitions and examples for all variables in the model. It is a companion resource to this printed document. The RSM XML Data Input Guide [3] can be viewed on-line to access the most up to date input guidelines.

---

[1] http://gwmftp.jacobs.com/manuals/mse_controller.pdf

[2] http://gwmftp.jacobs.com/manuals/mse_supervisor.pdf

[3] http://gwmftp.jacobs.com/xml_schema/hse_222.html

12.1 Flow Management Via The Use Of Controllers and Supervisors

The RSM consists of two interactive, primary components:

- Hydrologic Simulation Engine (HSE)
- Management Simulation Engine (MSE)

The Hydrologic Simulation Engine (HSE) is fully described in the complementary chapters of this manual. The MSE is designed to simulate the operational control characteristics encompassed by the wide spectrum of water flow control structures and algorithms currently in use by the SFWMD.

MSE is an integral component of the RSM, and is intended to provide two major modes of functionality in assisting the hydrological modeler involved in analysis and prediction of water control structure operational behaviors:

- Assessment of currently implemented management operational policies in response to hydrological forcing
- Optimization of operational policies

The first task is a critical capability for the assessment of water control operations in response to historic, real-time, or forecast forcing conditions. The latter facility forms an important analysis tool aimed at identification of alternative operational policies which must perform complex, multi-variate, resource allocation functions under the control of system boundary conditions and constraints. The MSE is formulated to address both of these needs by incorporating a variety of supervisory control algorithms including rule-based expert systems, finite state-machine processors, as well as a generic mathematical programming language interface, which provides access to a suite of state-of-the-art optimization algorithms.

The MSE consists of a multi-level hierarchical control scheme, incorporating a wide selection of control algorithms and decision making tools, each of which is integrated seamlessly with the hydrological computations of the HSE. From a hydroinformatics perspective, the RSM architecture emphasizes the decoupling of hydrological state information from the management information processing applied to the states. Given a well defined interface between the two, this approach enables multiple information processing algorithms to execute in parallel, with higher levels of the hierarchical management able to synthesize the individual results, which are best suited to the conjunctive managerial objectives. Essentially, the HSE provides hydrological and hydraulic state information (Σ), while external policies dictate managerial constraints and objectives (Λ).

In the MSE, state and process information can be functionally transformed by an independent set of filters, which can be viewed as information pre-processors. These processors are denoted as Assessors (A) and Filters. For example, an Assessor may perform statistical filtering such as spatio-temporal expectations, amplitude or time-delay modulation, or any other suitable data filtering operation. The MSE is then tasked with appropriately processing the assessed state information in order to produce water management control signals (χ, μ) which are applied to the hydraulic control structures in order to satisfy the desired constraints and objectives. Figure 12.1 illustrates this overall cyclic flow of state and management information in the RSM.

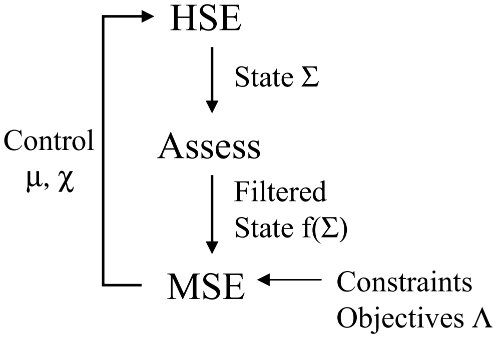


Figure 12.1: *RSM state and management information flow.*

More specifically, the MSE architecture is based on a multilayered hierarchy, with individual water control structures regulated by 'controllers' while the regional coordination and interoperation of controllers is imposed by 'supervisors'. Supervisors can change the functional behavior of controllers, completely switch control algorithms for a structure, or override the controller output based on integrated state information and/or rules. A schematic depiction of the HSE-MSE layered hierarchy is shown in Figure 12.2.

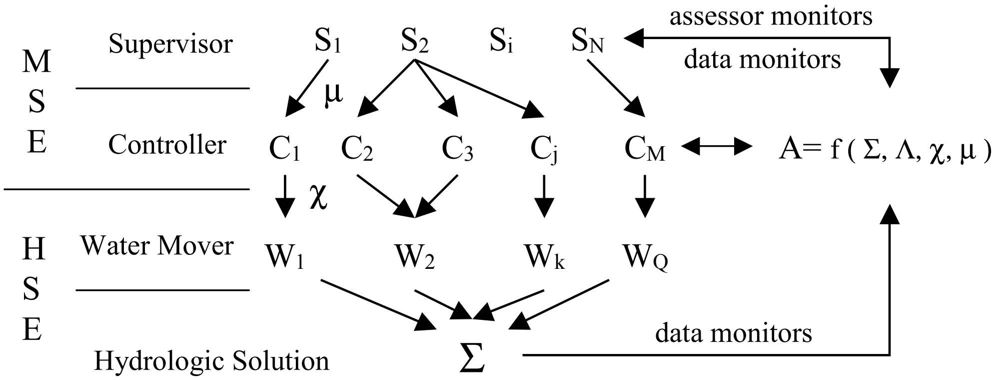


Figure 12.2: *HSE and MSE schematic.*

At the lowest layer is the hydrological state information (Σ) computed by the HSE. This information includes water stages, flow values, rainfall, ET, hydrologic boundary conditions, or any other state variable used as input or computed as output by the HSE. All such variables are made available to the MSE and Assessors through the implementation of a uniform data monitor interface. The data monitor interface extends naturally to the MSE input/output variables. Therefore, the input state information available to a controller or supervisor is not limited to water levels or flow values, but can include control information, decision variables, constraints or other management variables from other controllers or supervisors in the model. This transparency of state and process information throughout the model is central to the efficient synthesis and processing of heterogeneous information required to simplify and naturally express complex water management policies.

The top level of the MSE is the supervisory layer. There is no limit on the number of supervisory algorithms, or constraint on the number of controllers that a supervisor may influence. Based on state and process information, which optionally may have been filtered or assessed, the function of a supervisor is to produce the supervisory control signal (μ) for a single, or collection of hydraulic structure controllers. The supervisors are therefore able to comprehensively coordinate the global behavior of multiple independent, or coupled hydraulic structures.

The intermediate layer consists of the hydraulic structure watermover controllers. A controller is responsible for local regulation of structure flow. It is possible to attach multiple controllers to a structure watermover, although only one controller at a time is activated. This activation is controlled by a supervisor. For example, a fuzzy controller optimized for wet condition operations may be selected by a supervisor during significant rain events, while a standard rule curve could be enforced during normal operations. In this manner the MSE provides for dynamic switching of hydraulic structure control functions in response to state or process information.

Once the controllers have computed their respective control values (χ), these signals are applied as flow constraints to the structure watermovers in the HSE. Each watermover will compute a maximum flow capacity based on the hydrological state conditions and hydraulic transfer function of the structure. The resultant controlled flow will be some fraction of the currently available maximum flow capacity.

12.1.1 MSE Controllers `<controller>`

The MSE controller layer is the intermediary between the hydraulic structure watermovers and the regional-scale supervisory coordinators. The controllers can operate independently of the supervisors, in fact they are not required at all for uncontrolled operation of a hydraulic structure. The essential purpose of a controller is to regulate the maximum available flow through a structure to satisfy a local constraint. A controller may take as an input variable any state or process information, which can be monitored within the RSM. Since the interface between a structure watermover and any controller is uniform, it is possible to change controllers dynamically with a supervisory command, or manually with a simple XML input change. The unitary interface also allows for the modeler to mix and match controllers in a particular model application so that the local control schemes are a hybridization of any of the available control algorithms.

The currently available controller modules in the RSM include:

- One- and two-dimensional rulecurves
- Piecewise linear transfer function
- Proportional Integral Derivative (PID) feedback control
- Sigmoid PI feedback control
- Fuzzy control
- User-defined finite state machine

The controllers are specified in the `<controller>` section of the XML model file. Documentation, specifications and example usage of the controllers are specified in the MSE Controllers user manual.The MSE Controller User's Guide [4] is available on-line.

12.1.1.1 Microhydrological Control for Urban And Agricultural Zones

Agricultural and urban areas can be managed using control functions implicit in the pseudo cells. These controllers can decide when to pump water out of an urban cell or when to provide irrigation to an agricultural cell. Details of these options are given in the pseudocell sections on urban and agricultural modeling.

12.1.2 MSE Supervisors `<management>`

An MSE supervisor is effectively a meta-controller, a controller of controllers. The addition of this supervisory layer considerably simplifies the control expression of multiple, coordinated hydraulic structures.

In relation to the controllers, which are multi-input, single-output (MISO) processors, the supervisors are multi-input, multi-output (MIMO) processors. Supervisors have the

---

[4] http://gwmftp.jacobs.com/manuals/mse_controller.pdf

ability to change individual response characteristics of controllers, or, in the case of multiple controllers attached to a watermover, to dynamically select and activate a specific controller for a watermover. Specifically, the supervisory functions include

- Synoptic assessment of state and process information
- Controlling multiple parameters of multiple controllers
- Dynamic switching of multiple controllers
- Flow regulation override for controller(s)

This is done through a uniform interface to the controllers ensuring interoperability between different supervisory processors and any controller.

There is no practical limit on the number of supervisors allowed in a model, or on the number of controllers that a supervisor may affect. It is common to have a hybrid selection of different supervisors, each one regulating a specific sub-regional collection of hydraulic structures. The ability to selectively tailor management control algorithms, as well as the flexibility to easily reconfigure them in a plug-and-play fashion lends considerable power to the implementation of diverse and complex operational management scenarios.

The currently available supervisor modules in the MSE include:

- Fuzzy supervision
- User-defined finite state machine
- Linear Programming
- Graph flow
- Heuristic Object Routing Model

The supervisors are specified in the `<management>` section of the XML model file. Documentation, specifications and example usage of the supervisors are specified in the MSE Supervisors user manual.The MSE Supervisor User's Guide [5] is available on-line.

---

[5] http://gwmftp.jacobs.com/manuals/mse_supervisor.pdf

Bibliography

Abbott, M. B. (1998). *Computational Hydraulics*. Aldershot, Hampshire: Ashgate Publishing Company.

Abbott, M. B. and J. A. Cunge (1982). *Engineering Applications of Computational Hydraulics*. Pitman, London: ??

Akan, O. and B. C. Yen (1981, April). Diffusion-wave flood routing in channel networks. *ASCE Journal of Hydraulics Division 107*(6), 719–732.

Brater, E. F., H. W. King, J. E. Lindell, and C. Y. Wei (1996, March 01). *Handbook of Hydraulics*. McGraw-Hill Professional.

Brion, L., S. Senarath, A. M. W. Lal, and M. Belnap (2001, May 20-24). Application of the south florida regional simulation model in the southern everglades. In B. K. Panigrahi (Ed.), *Proceedings of the Specialty Symposium Held in Conjunction With the World Water and Environmental Resources Congress*, 1801 Alexander Bell Drive, Reston, Virginia 20191-4400, pp. p. 110. Environmental and Water Resources Institute of ASCE: American Society of Civil Engineers.

Brunner, G. W. (2002, November). *HEC-RAS River Analysis System Hydraulic Reference Manual* (Version 3.1 ed.). USACE-HEC 609 Second Stree Davis CA 95616-4687: United States Army Corps of Engineers. http://www.hec.usace.army.mil/software/hec-ras/documents/hydref/cvr_incvr_toc.pdf.

Bureau of Reclamation (1977). Design of small dams. Water resources technical publication, Bureau of Reclamation, Washington D.C.

Hirsch, C. (1989). *Numerical Computation of External and Internal Flow*. New York: Wiley.

Huber, W. and R. Dickinson (1988). *Storm Water Management Model User's Manual* (Version 4 ed.). Athens, Georgia: United States Environmental Protection Agency. EPA/600/3-88/001a (NTIS PB88-236641/AS).

Hydrologic Engineering Center (1994, October). *HEC-DSS User's Guide and Utility Manuals: User's Manual.* Davis CA: United States Army Corps of Engineers. http://modeling.water.ca.gov/delta/models/dss/.

Kadlec, R. H. and R. L. Knight (1996). *Treatment Wetlands.* Boca Raton, Florida: Lewis Publishers.

Lal, A. M. W. (1995). Calibration of riverbed roughness. *Journal of Hydraulic Engineering 121*(9), 664–671. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/calib_jo.pdf.

Lal, A. M. W. (1998a, April). Performance comparison of overland flow algorithms. *Journal of Hydraulic Engineering 124*(4), 342–349. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/alg_pap2.pdf.

Lal, A. M. W. (1998b, August 3-7). Selection of spatial and temporal discretization in wetland modeling. In *Proceedings of the International Water Resources Engineering Conference*, Memphis, Tennessee, pp. 604–609. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/resolution.pdf.

Lal, A. M. W. (1998c, August 3-7). Simulation of overland and groundwater flow in the everglades national park. In *Proceedings of the International Water Resources Engineering Conference*, Memphis, Tennessee, pp. 610–615. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/abs_tenn1.pdf.

Lal, A. M. W. (1998d, September). Weighted implicit finite-volume model for overland flow. *Journal of Hydraulic Engineering 124*(9), 941–950. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/poly.pdf.

Lal, A. M. W. (2000a, July). An analytical solution for the stream-aquifer interaction problem. In *Proceedings of the EWRI Watershed Management 2000 Conference*, Colorado State University, Fort Collins, Colorado. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/ovlcan_abs.pdf.

Lal, A. M. W. (2000b, July). Modification of canal flow due to stream-aquifer interaction. *Journal of Hydraulic Engineering 127*(7), 567–576. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/ovlcan.pdf.

Lal, A. M. W. (2000c). Numerical errors in groundwater and overland flow models. *Water Resources Research 36*(5), 1237–1247. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/gw_err.pdf.

Lal, A. M. W. (2001, May 20-24). Selection of time step and grid size in modeling integrated stream-aquifer interaction. In *Proceedings of the World Environmental Congress*, Orlando, Florida. http://gwmftp.jacobs.com/Peer_Review/stream-aquifer.pdf.

Lal, A. M. W. and R. Van Zee (2003, June 23-26). Error analysis of the finite volume based regional simulation model rsm. In *Proceedings of the World Water and Environmental Resources Congress*, Philadelphia, Pennsylvania. http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/hse_err.pdf.

Lal, A. M. W. and R. Van Zee (2005, May 15-19). Calibration of bulk aquifer parameters of regional models using hydraulic disturbances. In *Proceedings of ??*, Anchorage, AK. EWRI World Water and Environmental Resources Congress.

Lal, A. M. W., R. Van Zee, and M. Belnap (2005, April). Case study: Model to simulate regional flow in south florida. *Journal of Hydraulic Engineering ?*(?), ??–?? http://www.sfwmd.gov/org/pld/hsm/pubs/wlal/oodpaper.pdf.

Loucks, D. P., L. Upmanu, D. L. Phillips, and K. H. Reckhow (2002). Quantifying and communicating model uncertainty for decision making in the everglades. Model uncertainty workshop report, SFWMD and USASC.

Preissman, A. (1961). Propagation des intumescences dans les canaux et rivieres. In *First Congress of the French Association for Computation*, Grenoble, France.

South Florida Water Management District (2005a). *MSE Controllers, Documentation and User Manual*. 3301 Gun Club Road, West Palm Beach, FL 33406: South Florida Water Management District. http://gwmftp.jacobs.com/manuals/mse_controller.pdf.

South Florida Water Management District (2005b). *MSE Supervisors, Documentation and User Manual*. 3301 Gun Club Road, West Palm Beach, FL 33406: South Florida Water Management District. http://gwmftp.jacobs.com/manuals/mse_supervisor.pdf.

Welter, D., A. W. Lal, and R. Van Zee (2005). Abstract: Use of singular value decomposition (svd) in calibrating regional models for south florida. Where will this be submitted?

Yarnell, D. (1934). Bridge piers as channel obstructions. Technical Bulletin 442, U.S. Department of Agriculture, Washington D.C.

Appendix A

RSM Development History

In 1994, SFWMD engineers (lead by Jayantha Obeysekera), recognized the need for a robust, comprehensive integrated hydrologic model to simulate flow and water management within the District boundaries. The need was determined because of the deficiencies of existing models to simulate the natural hydrologic conditions coupled with the variety of man-made water control and distribution structures in South Florida. As a first action to develop a new model, Randy Van Zee of the SFWMD worked with a New York based company RPA and a Colorado based company WRMI to begin formulating a new model. The first deliverables of these groups was termed the South Florida Regional Simulation Model SFRSM. The SFRSM was a computer code written in C++ to simulate a small region of South Florida. In 1995, Wasantha Lal joined the District and began working with Randy Van Zee. Lal began work to defend SFWMM algorithms and develop computational methods for the new model. Discussions were initiated at the time in the District to systematically develop a new regional model.

Once the RPA product came up for review, Lal, Randy, Mark Belnap and Ken Tarboton wanted to take an ”in-house” approach to development. Considering the special needs of the district, Lal wrote a joint memo laying out a work plan, and proposed algorithms for the initial test code. Encouraged by the success of RPA's first version to write hydrologic models using object oriented methods, Randy Van Zee wrote the first 30 lines of RSM to follow a FORTRAN code written by Lal to simulate a small rectangular canal. These were the first 30 lines of the RSM used now.

Initially the FORTRAN version of RSM grew rapidly with overland, groundwater, canal and structure flow capabilities in an integrated fully implicit setting. Lal was the only FORTRAN author, and a multiple developer environment did not exist. Mark Belnap took over Randy's 30 line code and created the first object design of RSM, and expanded the code to give the same results given by the FORTRAN code written by Lal. This design allowed the growth of the code with time and allowed multiple users. Work flourished with gnu compilers, DTD debuggers and CVS version control. During this critical phase, Belnap coined terms such as water bodies and water movers to name abstract objects. This architecture formed the core of HSE, and allowed others to work simultaneously. Lal (Lal, 1998a) published the first paper on HSE algorithms for a 2-D implicit finite volume method, and added large sparse solver to solve the equations. David Welter added the first sparse solver PETSC to the model, to give the necessary flexibility of using a variety of solvers. David later added the first vertical solution, written by Ken Tarboton. This feature is called pseudocell now. The term was coined by Randy after abstracting it from the vertical solution. As the code became large, the FORTRAN version was abandoned, and C++ version was taken as the official version. Victor Kelson (currently WPA, MN) suggested to use XML to enter input data into the model, and Belnap had the first versions with XML working shortly.

During second rapid growth phase of development, Randy added various time series (e.g. DSS) capabilities, numerous data entry formats such as index entry, NETCDF, and Pseudocells. Lal added structures, overland, groundwater, canal and lake interactions, various test cases, and certain types of pseudocells. With inspiration from Ken Khonyha, Lal added SV converter, transmissivity and conveyance objects.

The earliest applications of the model were conducted by Lal using an old Kissimmee data set. Later, Belnap applied HSE on Everglade National Park, and Randy, Lal and Belnap applied on the L-8 basin. Subsequently, Senarat applied it on Everglades National Park, and Maged Hussein and David Welter used it for the South West Feasibility study. Ruben and Eric Flaig also started applying it on a South Dada site. David Welter turned into a first test pilot on many new features, and started to fix many of the bugs himself.

Two other significant contributions to RSM development effort have to do with the use of analytical solutions such as the one for stream-aquifer interaction so that the model can be verified independent of field data. Error analysis by Lal also contributed to understanding the proper spatial temporal resolutions to be used. Multi-layer capability is one more addition to the RSM by Lal during the 2001-2002 period, in preparation of the South West Feasibility study model.

In preparation of the management simulation engine (MSE), Ray Santee, Paul Trimble, Ron Mierrau, George Hwa and Cal Neidrauer of SFWMM were consulted in early 2002 to understand various management aspects within the system. As a result of the consultations, Lal was able to extract a simple adaptive algorithm as the first management component, which Randy implemented into the first feedback controller (alpha).

In September 2002 Joseph Park was hired by Randy to assist in development of the MSE. Joseph worked with Lal, Randy, and Dave to reformulate the existing controller into a water mover flow regulation design, and coined the term MSE. Based on this design, Joseph implemented a generic controller class, and several controllers including PID, PI-Sigmoid, piecewise linear transfer function, generic fuzzy controller, and a user-defined finite state machine were implemented as a dynamically loaded shared library. Another design decision by the developers was that the MSE and the HSE should be decoupled, and that coordination and dynamic modification of controllers would be needed.

Joseph designed a multi-layer control hierarchy consisting of an Event Manager running multiple Control Supervisory Algorithms (CSA's), the output of which could be synthesized in a Decision Manager-Arbiter based on constraint and objective function input from a Condition Manager. The Decision Manager output would then control the behavior of the water mover controllers. The Event Manager and CSA's were eventually condensed into supervisors, and the Condition Manager dropped.

In the summer of 2003 Pete Loucks of Cornell University worked with the developers and added a LP module based on a GLPK interface to the MSE, resulting in the LP supervisor. Work in 2004 centered on the development of supervisors. In late 2004, based on analogy with the intensively used SFWMM, Randy recognized that synoptic assessment capabilities were needed to simplify the supervisory algorithms. Randy developed an implementation of the Object Routing Model supervisory control, and eventually transformed this into a set of assessors. Another revelation in late 2004 was the recognition by Joseph, based on input from Raul Novoa, Michelle Irizarry and Ray Santee, that a managerial abstraction of the HSE canal network was needed to consolidate and simplify the MSE interface with the HSE and assessors. Joseph designed and implemented an MSE Network based on standard graph theory abstractions, and implemented the graph supervisor capable of maxflow and mincost flow routing solutions. It was also recognized that the MSE network would form a natural data store for managerial parameters, constraints, and assessed state information relevant to the water mover controllers and the Water Control Units.

Numerous applications of RSM became possible only because of some of the GUI products by Clay Brown, David Welter, and Vic Kelson (RMA). One of the first products used as the HSE-GUI was a product developed by RTI. Clay Brown revamped the earlier version the first deliverable of RTI's ARCVIEW GIS base interface, and started providing support to many added features of the RSM. This completely changed the way RSM was used because of the increased size of data sets possible. In place of the TECPLOT data conversion program, Dave Welter started using IBM Data Explorer and initiated a contract with Vic Kelson to build a GUI for RSM using Python.

Appendix B

Primer on Using XML

The World Wide Web Consortium (W3C) develops technologies (specifications, guidelines, software, and tools)to allow people to use the World Wide Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding of new technologies. The use of XML is one topic covered by W3C. Three very good XML guidance documents exist on the W3C web site. These documents include:

1. An XML primer that describes the XML Schema facilities, and is oriented towards quickly understanding how to create schemas using the XML Schema language: W3C XML Primer[1].

2. An overview of the XML Schema definition language, which offers facilities for describing the structure and constraining the contents of XML 1.0: W3C XML Structures[2].

3. An overview of the XML Schema definition language, which offers facilities for defining data types to be used in XML Schemas as well as other XML specifications W3C XML Datatypes[3].

---

[1] http://www.w3.org/TR/xmlschema-0/
[2] http://www.w3.org/TR/xmlschema-1/
[3] http://www.w3.org/TR/xmlschema-2/

B.1 What Is XML?

The e**X**tensible **M**arkup **L**anguage (XML) is a very flexible text format derived from the Standard Generalized Markup Language (SGML). XML is a meta-markup language that provides a format for describing structured data. XML is used to describe documents and data in a standardized, text-based format that can be easily transported via standard Internet protocols. XML was originally designed to meet the challenges of large-scale electronic publishing. Current usage of XML is now much more widespread. XML is used to manage data on major web sites such as Barnes and Noble, Amazon, etc, and is becoming the standard for multi-project, multi-user data interchange. Moreover, XML is now considered to be the universal language for data on the Web. XML gives developers the power to deliver structured data from a wide variety of applications to the desktop for local computation and presentation.

There are two types of XML data used in RSM: ¡Elements¿ and attributes. The <> nomenclature is used throughout this manual to denote XML elements, which may also be referred to as nodes (although we try to avoid this term in the XML context because the term ”nodes” is also used in the context of the finite-volume gridded domain). The attributes, which are properties of elements, are not placed in <>, but rather they are typed normally. The attribute values will be enclosed in quotes. A simple example is shown below of the <control> element with two of its attributes (tslen and tstype):

```
<control>
   tslen="15000"
   tstype="minute"
 </control>
```

XML data is validated through the use of a Data Type Definition (DTD) or an XML Schema, so that the XML data used as input to a program can be checked before being used. Both the DTD and the XML schema define the data structures and data types used by the program that receives the XML input file and they both can validate the data in the XML input files. The original XML data validation standard was the DTD, but now the DTD standard has been largely replaced with the newer XML Schema Standard. The XML schema can include a variety of data fields that allow for strict data prototyping, data ranges (max and min, both inclusive and exclusive values), default values, whether the data is required or not, among other things. Although an XML schema itself is usually complex and somewhat difficult to read, XML parsers do a great job at extracting the data from the XML file. Since XML is an extensible language, it is easy to extend a schema to handle whatever type of data the programmer wants to include. Compared to HTML, XML data typing is rigidly enforced, such that data type checking and data validation are native to XML. The beauty of using a schema is that the data stored in XML files can be validated before being sent to the program that will use the data. This relieves the programmer from having to check the user input. In the case of RSM, which has about 700 input variables, or

XML attributes (268 doubles, 174 integers, and 257 string variables), that can potentially be used in any model, data validation is a necessary step in developing defensible models.

Prior to RSM version 2.2.2, a DTD has been used to validate input to RSM. Due to inherent limitations of DTD's in validating input, an XML schema has been written for RSM version 2.2.2. The schema will be maintained into the future because of its data validation advantages over the DTD. More specific information on the RSM DTD (see Section B.2) and the XML Schema (see Section B.3) is available.

XML is a text-based mark up language that allows easy exchange of data. Since XML is self-describing, it is a natural choice for data input to RSM. The self-describing nature of XML means that information about the data is easily discernible. The XML schema employed for HSE also:

1. Indicates how the data is structured into a hierarchy of elements and associated attributes understood by the RSM objects
2. Indicates the usage and content of specific data items needed for RSM
3. Provides the syntax to allow hydrologic objects to be cast as XML elements and attributes and thus interpreted correctly by the RSM.

B.2 The RSM DTD File

Prior to RSM version 2.2.2, the only method of validation available for RSM XML input files used the DTD file residing in the directory `../hse/benchmarks`. This file defined the XML document structure with a list of legal RSM elements and attributes. When the XML document was processed by the XML Parser, the DTD file specified the elements and attributes which were valid in the XML file. Strict data typing did not occur as part of this validation, which means that the DTD cannot determine if the proper data types are being used for an attribute. Due to inherent limitations in DTD's, XML schemas have been developed to allow more accurate data validation to occur on XML data sets. Starting at RSM version 2.2.2, an XML schema has been written to improve the validation of RSM input data sets.

B.3 The RSM XML Schema

From RSM version 2.2.2 onward, an XML schema can be used to validate the model input files. This schema is located on the web and serves as the single point of reference for RSM input. The RSM XML data structures defined in the schema can be viewed here: Graphical portrayal of the RSM 2.2.2 XML Schema[4]. The actual RSM XML Schema can be downloaded here: Download the RSM 2.2.2 XML Schema [5]

As of February 2005, the RSM 2.2.2 schema accurately represents the available XML elements and attributes present in the model. The documentation of all elements and attributes is still not complete but continues to progress as time allows. The schema, since it is written in XML, is easier to read and understand than the RSM DTD.

The following example shows how the RSM schema (hse_222_corrected.xsd) is called in RSM XML input files and used to validate model input. The `<hse>` tag must include the reference to the hse schema as shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<hse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="http://gwmftp.jacobs.com/xml_schema_corrected/
            hse_222_corrected.xsd" version="2.2.2" >

  <control
    tslen="15000"
    tstype="minute"
    startdate="01jan1994"
    starttime="0000"
    enddate="01jan1994"
    endtime="0230"
    alpha="0.500"
    solver="PETSC"
    method="gmres"
    precond="ilu">
  </control>
```

---

[4] http://gwmftp.jacobs.com/xml_schema_corrected/graphics/hse_222.html

[5] http://gwmftp.jacobs.com/xml_schema_corrected/hse_222_corrected.zip

B.3.1 How To Convert A DTD-Based RSM Input File To An XML Schema-Based Input File

Only one change is needed to convert the DTD based XML input data file to a schema based XML input file that can be validated. The beginning of the DTD-based file needs to be changed into the proper XML format, with the proper hse version number listed. To update the XML data file, simply cut-out:

```
<?xml version="1.0"?>
<!DOCTYPE hse SYSTEM "../hse.dtd" [
]>
<hse version="0.1">
```

and replace it with:

```
<hse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation=
     "http://gwmftp.jacobs.com/xml_schema_corrected/
            hse_222_corrected.xsd" version="2.2.2" >
```

B.3.2 How To Validate RSM Input Files Against The XML Schema

A variety of programs and web-based utilities exist for XML validation using schemas. Examples of these include the W3C XML schema validating routine, Microsoft Visual Studio, XMLSPY Home, and many other programs.

Three validation examples are demonstrated here using the W3C validating routine. The first case demonstrates a successful validation, which means that all of the XML data is of the proper data type and all required data is present. The second case demonstrates unsuccessful validation of an XML data file. This file contains an invalid double precision number that was intentionally entered in the data set. Although this data set contains invalid data it will pass a DTD validation routine but it will not pass the schema-based validation because the schema validation is much more strict with respect to data typing. The third case ??

B.3.2.1 Case 1: Successful Validation Of Benchmark Problem 1 Using W3C Validating Routine

Begin by clicking here for the W3C XML schema validating routine[6]

---

[6] http://www.w3.org/2001/03/webdata/xsv

On this page, there are two forms for validating your data. The first form is for checking a schema which is accessible via the Web, and/or schema-validating an instance with a schema of your own. The second form is used if you are behind a fire wall or have a schema to check, which is not accessible via the Web. For validating local data files scroll down to use the second validation form and select your local RSM data file by browsing to it. In this example shown below, the Benchmark 1 case has been selected.

```
<?xml version="1.0" encoding="UTF-8"?>

<hse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation=
     "http://gwmftp.jacobs.com/xml_schema_corrected/
     hse_222_corrected.xsd" version="2.2.2" >

  <control
    tslen="15000"
    tstype="minute"
    startdate="01jan1994"
    starttime="0000"
    enddate="01jan1994"
    endtime="0230"
    alpha="0.500"
    solver="PETSC"
    method="gmres"
    precond="ilu">
  </control>

  <mesh>
    .. To save space in this document, content removed from benchmark 1
  </mesh>

  <output>
    .. To save space in this document, content removed from benchmark 1
  </output>

</hse>
```

The following results of the validation are returned from the W3C site.

```
Schema validating with XSV 2.7-1 of 2004/04/01 13:40:50
Target: file:///usr/local/XSV/xsvlog/@31924.1uploaded
   (Real name: M:\Data\models_2.2.2\hse\kcb_benchmarks\BM1\run3x3.xml)
docElt: {None}hse
Validation was strict, starting with type [Anonymous]
schemaLocs: None -> http://gwmftp.jacobs.com/
```

```
    xml_schema_corrected/hse_222_corrected.xsd
The schema(s) used for schema-validation had no errors
No schema-validity problems were found in the target

--------------------------------------------------------------------------------

Schema resources involved
Attempt to load a schema document from
http://gwmftp.jacobs.com/xml_schema_corrected/hse_222_corrected.xsd
(source: schemaLoc) for no namespace, succeeded

--------------------------------------------------------------------------------
```

B.3.2.2 Case 2: Unsuccessful Validation Of Benchmark Problem 1 Using W3C Validating Routine

To corrupt the input file, the letter ”l” (which looks a lot like a 1) has been placed at the end of the alpha input line as shown:

```
alpha="0.500l"
```

.

As shown below, this incorrect input triggers an error because alpha is defined as a double precision number in the schema and the value of ”0.500l” is not a valid double precision number. This case would be considered valid in a DTD-based validation check because this type of examination simply checks to see that a value exists for the alpha term.

```
Schema validating with XSV 2.7-1 of 2004/04/01 13:40:50
Target: file:///usr/local/XSV/xsvlog/@32070.1uploaded
   (Real name: M:\Data\models_2.2.2\hse\kcb_benchmarks\BM1\run3x3.xml)
docElt: {None}hse
Validation was strict, starting with type [Anonymous]
schemaLocs: None -> http://gwmftp.jacobs.com/
    xml_schema_corrected/hse_222_corrected.xsd
The schema(s) used for schema-validation had no errors
1 schema-validity problem was found in the target

--------------------------------------------------------------------------------

Schema resources involved
Attempt to load a schema document from
http://gwmftp.jacobs.com/xml_schema_corrected/hse_222_corrected.xsd
(source: schemaLoc) for no namespace, succeeded
```

```
--------------------------------------------------------------------------------

Problems with the schema-validity of the target
file:///usr/local/XSV/xsvlog/@32070.1uploaded:6:3: Invalid per cvc-attribute.1.2:

 attribute type check failed for {None}:alpha: 0.500l is not a valid double literal

--------------------------------------------------------------------------------
```

B.3.2.3 Case 3: Validation Of Models Having More Than One XML Input File

Currently, validation routines that check input data against a schema currently cannot automatically validate files. An emerging technology called XInclude[7] is now being promoted to include external files so that users can break long XML documents up into smaller pieces. Several RSM benchmarks do break the XML files into several pieces (i.e., Benchmark 55). This technology will eventually replace the current ”Entity” approach used with DTD-based document types as shown below.

```
<?xml version="1.0" ?>
<!DOCTYPE hse SYSTEM "../hse.dtd" [
<!ENTITY pseudo SYSTEM "pseudo.xml"> <- this is an external file
that will be included by using the & command
]>
<hse version="0.1">
  <control
    tslen="24"
 .. content removed to save space
    method="gmres"
    precond="ilu">
  </control>
... content removed to save space
    &pseudo;   <- this includes the content of the file pseudo.xml
... content removed to save space
</hse>
```

If XInclude is used in an XML data set, the validators do not automatically load the included files and check them against the schema. In this case, you *must* create a new XML data set that contains all of the XML data by loading all of the included files into one file. This file can then be validated. As the technology is standardized, more information will become available for XInclude.

---

[7] http://www.w3.org/TR/xinclude/

B.3.2.4 Additional XML Details

The RSM XML input files are composed of mark-up fields and content. The mark-up fields describe elements and attributes and the content is the assigned values for the attributes. The types of attributes needed for any element depend upon the structure of the element, which is found in the XML schema. The schema completely defines the structure and relationships between all elements and their attributes. The schema may also define the data types, whether the data is a required input, the allowable range for the data, the default values, the maximum and minimum number of occurrences of any element, among other things. The schema is intimately tied to the objects present in the RSM source code, and the degree of data control entered in the schema is up to the programmer to decide. Although it is tempting to think of XML elements as hydrological objects in RSM, there is not a one-to-one correspondence between these items.

- There are nearly 200 elements that loosely represent the primary building blocks for the hydrological objects in the RSM. XML elements are denoted with starting and ending tags. Elements can contain other elements or they can contain attributes. For elements containing other elements, an example using the element `<mesh>` would look like:

  ```
  <mesh>
     <geometry file="mesh3x3.2dm"> </geometry>
     <bottom>  <const value="0.0">   </const> </bottom>
     <surface> <const value="500.0"> </const> </surface>
     <conveyance>
       <mannings a="1.000" detent="0.00001"></mannings>
     </conveyance>
   </mesh>
  ```

  For elements containing only attributes, an example using `<control>` would look like:

  ```
  <control>
    tslen="15000"
    tstype="minute"
    startdate="01jan1994"
    starttime="0000"
    enddate="01jan1994"
    endtime="0230"
    alpha="0.500"
    solver="PETSC"
    method="gmres"
    precond="ilu">
  </control>
  ```

Note that the `">"` portion of the beginning tag for `<control>` occurs after the attributes are defined, and is immediately followed by the ending tag `</control>`.

- Attributes are the variables used to describe the properties of the elements. As shown in the second example above, attributes are placed within the opening tag of the element. Attribute definitions come in name="value" pairs. For example `<tslen="15000">` is the time step length attribute of the control element and it is assigned a value of 15000. In XML, all attribute values must be placed within quotes.

- There are nearly 200 elements and about 700 attributes supported in RSM. The 700 attributes come in the form of double precision numbers (268), long integers (174), and text entities (257). As model development continues, the number of elements and attributes may change. The validation of the XML input files checks the validity of the double precision number and integers. The schema can also check the validity of the string variables by comparing the string values to an enumeration list. Most attributes that are string variable type in the RSM schema have enumeration lists that contain the allowable text strings. If a text string, other than one that is allowed and used as input, the XML schema validator will report this to the user as incorrect input.

- Comments can be used in XML data sets and they begin with `‘‘<!--’’` and end with `‘‘-->’’`. Comments can contain any data except the literal string `‘‘--’’`. You can place comments between mark-up anywhere in your document.

- An XML file can be separated into several files for convenience. These files are defined using `<!ENTITY pseudocells SYSTEM "pseudo.xml">`, and referred to later when necessary using `&pseudocells;`. The file `pseudo.xml` should be placed within the directory. An example of this has been shown above in Section B.3.2.3.

- The XML schema contains information on Elements and Attributes. The attribute data types and relationships between elements are established in the schema. The RSM XML data structures can be viewed here[8].

- In the XML data files that are DTD-based, PCDATA means parsed character data. Character data is the text found between the start tag and the end tag of an XML element. PCDATA is text that will be parsed by a parser. Tags inside the text will be treated as mark-up and entities will be expanded.

- In the XML data files that are DTD-based, CDATA also means character data. CDATA is text that will *not* be parsed by a parser. Tags inside the text will *not* be treated as mark-up and entities will not be expanded.

---

[8] http://gwmftp.jacobs.com/xml_schema_corrected/hse_222_corrected.html

Appendix C

Extending A 2D Model Into 3D - The XML `<multilayer>` Element

Although the following component of HSE is functional, but is not optimally implemented. Additional capabilities need to be added to this portion of the code for greater flexibility in usage. User's may need additional guidance in building 3D flow models.

C.1 Overview of Building a 3D Model in RSM

The HSE model was originally designed to perform only two-dimensional overland and groundwater flow simulations. However, the model has been extended to simulate fully three-dimensional ground water flow. This was accomplished without applying undue pressure on the existing model architecture because Darcy's Law is easily extended from 2D to 3D.

Converting a 2D model to 3D is achieved by pre-processing the 3-D groundwater data and creating special HSE water bodies and water movers that replicate what occurs in an actual 3-D groundwater system. A preprocessor is used to convert the existing base cell data in a GMS mesh (2dm) format into a new layered data set. This consists of a new 2-D mesh file and new water mover file. The new water mover file is referred to in the XML file under the keyword `<multilayer>`. An example of a multilayered grid is shown in Figure C.1.

C.1.1 2D to 3D Grid Program

The C++ program `gw3d2hse` converts a base 2-D mesh data set and a layer data set into a second 2-D mesh data set and a water mover data set. HSE can process this new data as

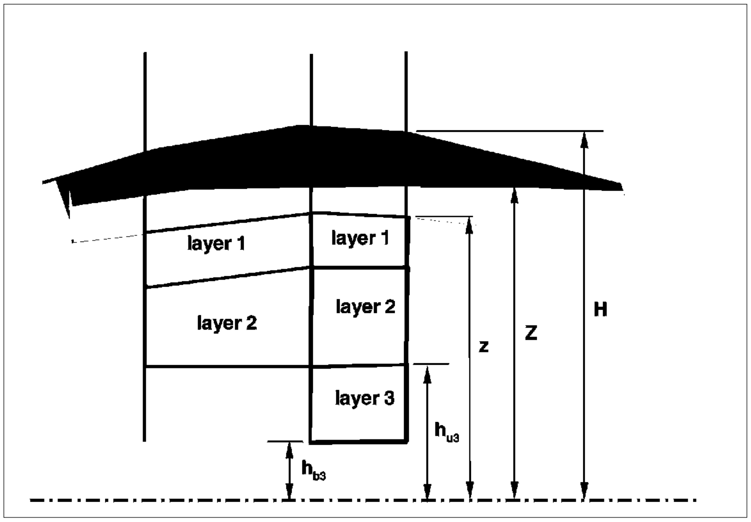


Figure C.1: *Sketch of the multi-layered grid used to solve 3-D groundwater flow.*

if it is a set of water movers to the 2-D problem. The results which are actually in 3-D have to be mapped to 2-D using RSM GUI tools or some other tool. The input and output files for the preprocessor are described below.

C.1.1.1 Two-Dimensional Mesh File

This is the first input file to `gw3d2hse` and is standard RSM 2-D mesh file described in Section 6.2 of this manual. This grid file will be extended from 2d to 3d based on the user-defined layering described in subsection C.1.1.2, so the number of nodes and elements will be increased in the resulting grid output file.

C.1.1.2 Added Layer File

This file contains all the layers added to the bottom of the base mesh layer described earlier. The typical data set consist of data blocks, which describe layers below each base cell, started from top to bottom. The format is shown below with actual numbers replacing contents in square brackets. An example is given directly below the definitions.

```
.....
 nb [base cell ID] [base cell vertical hydraulic conductivity]
 lay [top] [bot] [sc1] [sc2] [hor. hyd. cond.] [vert. hyd. cond.] [d/k]
 lay [top] [bot] [sc1] [sc2] [hor. hyd. cond.] [vert. hyd. cond.] [d/k]
```

Table C.1: *Variables defined in the layer data input file.*

Tag	Definition
`[base cell ID]`	Cell ID of the uppermost layer (i.e., the base cell)
`[base cell vertical hydraulic conductivity]`	Vertical hydraulic conductivity of the base cell (l/t)
`[top elevation]`	Elevation at the top of the layer (l)
`[bot elevation]`	Elevation at the bottom of the layer (l)
`[sc (unconfined)]`	Storage coefficient when acting as an unconfined layer
`[sc (confined)]`	Storage coefficient when fully saturated.
`[horizontal hydraulic conductivity]`	Hydraulic conductivity in the horizontal direction (l/t)
`[vertical hydraulic conductivity]`	Hydraulic conductivity in the vertical direction (l/t)
`[d/k]`	d/k Depth/conductivity value of thin layer assumed to be at the top of the layer (l/t)

```
 lay [top] [bot] [sc1] [sc2] [hor. hyd. cond.] [vert. hyd. cond.] [d/k]
....
where\\
top = top elevation,
bot = bottom elevation,
sc1 = storage coefficient for unconfined conditions,
sc2 = storage coefficient for confined conditions,
vert. = vertical,\\
hor. = horizontal,\\
hyd. cond. = hydraulic conductivity, and
elev. = elevation
d/k = a unit of thickness (d) and hyd. cond. (k) that impedes vertical flow
```

Definitions of the variables are given below. An example will follow.

```
This example places two additional layers beneath base cells 5 and 14.
Beneath cell 5, a 140 foot thick layer and a 70 foot layer are added.
Beneath cell 14, a 60 ft and 40 ft layer are added.
This is a highly simplified example.  In most situations,
it would be expected that laterally continuous layers
would be added over more than 1 cell.

nb  5 0.02
```

```
lay 440.0 300.0 0.2 0.001 0.15 0.015 0.0
lay 300.0 230.0 0.2 0.001 0.15 0.015 0.3
nb 14 0.04
lay 400.0 340.0 0.2 0.001 0.15 0.015 0.2
lay 340.0 300.0 0.2 0.001 0.15 0.015 0.1
```

C.1.1.3 Output 2-D Mesh File

This file is like any other 2-D mesh file with additional cells to represent the new layers. This becomes the new input mesh file for the HSE model.

C.1.1.4 Output Water Mover File

The output watermover file from the preprocessor is to be used as an input to the model. The name of this file is included in the XML file as shown below. If the file is `layered.dat`, the XML input would be

```
..............
  <multilayer>
    <layer file = "layered.dat">  </layer>
  </multilayer>
.............
```

C.1.2 Other Input Files And Modifications Needed For 3-D Groundwater Flow Modeling

A number of files have to be modified when 3-D or layered groundwater flow is modeled. This is mainly because of the additional water bodies introduced as a result of the 3-D layers need some extra information. The following sections describe how properties are described for the additional water bodies describing the layered or 3-D formulations.

C.1.2.1 Starting Head File `<shead>`

The number of cells that have to be initialized is different with the layered cells. This has to be modified to include the initial heads for these cells.

C.1.2.2 Pseudocell Definition File `<pseudocell>`

The pseudo cell types for the new layered cells has to be added before a model run. However the new pseudo cells of the layered formulation don't do anything internally. The pseudo cell behavior is introduced using the "index entry" option. The format for the new pseudo cell entry type is as follows.

```
..............
        <entry id ="2" label="lay">
          <layerpc> </layerpc>
        </entry>
.............
```

C.1.2.3 Horizontal Conductance Definition File `<transmissivity>`

The horizontal conductivity of the new layers or 3-D cells has to be added as `<layered>` type for a layered or 3D model run. The numerical values, however, will be overwritten with values read from the `<multilayer>` file. The `<layered>` tag defines the water bodies layered type so that they can be overwritten. Values of conductivity, lower layer elevation and upper layer elevation are all included in this file.

```
..............
     <entry id="2" label="type1">
     <layered cond = "10" lower = "300" higher = "450">
     </layered>
.............
```

C.1.2.4 SV Converter Definition File `<svconverter>`

SV converters also have to be defined for the new layer cells. The one to chose for layered cells is shown below. The index entry option is used for this purpose.

```
..............
        <entry id="2" label="type1">
          <layersv sccon= "0.2" scunc = "0.0002"> </layersv>
        </entry>
.............
```

C.1.3 Putting It All Together

The file `layered.dat` that comes as output from the preprocessor file is shown below. This file gets defined in the XML file under the tag `<multilayer>`. The first line is a plain text line that helps to keep track of the layer information for debugging. Layered of 3-D features of the model are activated when the existence of a file is detected.

```
layer cel b_cel l_no  z_l  z_u    sc   scc  Hcond  Vcon  d/k
base     5 0.02
base    14 0.02
cel  19  5  1      400      450      0.2    0.001     0.15    0.015     0
cel  20 14  1      400      450      0.2    0.001     0.15    0.015     0
cel  21  5  2      350      400      0.2    0.001     0.15    0.015     0
cel  22 14  2      350      400      0.2    0.001     0.15    0.015     0
cel  23  5  3      300      350      0.2    0.001     0.15    0.015     0
cel  24 14  3      300      350      0.2    0.001     0.15    0.015     0
cel  25  5  4      250      300      0.2    0.001     0.15    0.015     0
cel  26 14  4      250      300      0.2    0.001     0.15    0.015     0
cel  27  5  5      200      250      0.2    0.001     0.15    0.015     0
cel  28 14  5      200      250      0.2    0.001     0.15    0.015     0
link 19 5
link 21 19
link 23 21
link 25 23
link 27 25
link 20 14
link 22 20
link 24 22
link 26 24
link 28 26
```

C.2 Boundary Conditions For Three-Dimensional Flow`<multilayer>`

Needs to be written