Regional Simulation Model (RSM)
Hydrologic Simulation Engine (HSE) User's Manual

RSM Version 2.2.9
Document Last Updated on June 20, 2005
Document Produced on June 28, 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) Hydrologic Simulation Engine (HSE) User's Manual

Revision History Last updated on June 28, 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 HPM 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.
4.1	Tim,Bob, Ken	3/28/05	Additional editorial updates chap 5-end, color linking activated.
5.0	Ken Black	3/30/05	Renamed manual, removed chapters 1-3, 11-12.
5.01	Ken Black	4/5/05	Introduced transmissivity options in subsection 3.4.2.
5.10	Ken, Bob	4/14/05	Introduced the term HPM for pseudocell, removed HPM content as directed by E. Flaig on 4/7/05.
5.11	Ken	4/25/05	Revised Ch. 2 XML element figures.
5.20	Tim, Bob	5/17/05	Revised all tables to add defaults and dimensions, and QA'd using 2.2.9 DTD.
5.20	Tim, Bob	5/17/05	Note that new XML elements since 2.2.2 may have not yet been added to manual.
6.00	Bob	6/20/05	Updated to Version 2.2.9 with new water movers. Expanded output options to include those available in code and/or DTD.

Contents

List of Figures ix

List of Tables xi

1 Introduction 6

1.1 Programming Information 6

1.1.1 Programming Details 7

1.2 How Is A Model Solution Achieved? 7

2 RSM Input Using XML 9

2.1 Introduction 9

2.1.1 Naming Conventions 11

2.1.2 Software Setup Needed To Run The Model 12

2.1.3 Steps And Data Needed To Run The Model 12

2.1.4 HSE Specification Using XML 13

2.1.5 XML Elements Under The Root 15

2.2 Suggested Development Procedure for Applications 15

2.2.1 Sequence Of Object Creation 15

2.3 RSM Directory Structure 16

2.3.1 RSM Benchmarks 17

3 HSE Model Components and XML Input **28**

3.1 Basic Model Set-Up Parameters - The XML `<control>` Element 29

3.1.1 Model Units 33

3.2 Data For The Two-Dimensional Model `<mesh>` 35

3.2.1 Attributes of the Data File Formats Used In The `<mesh>` Environment 38

3.2.1.1 Examples Of 2-D Data Defined Within `<mesh>` 47

3.3 Two-Dimensional Grid Data `<geometry>` 48

3.4 Alternative Forms Of 2-D Flow Equations 51

3.4.1 Overland Flow Options 51

3.4.1.1 Conveyance Type `<mannings>` 52

3.4.1.2 Conveyance Type `<cadlec>` 57

3.4.1.3 Conveyance Type `<lookup>` 58

3.4.1.4 Mixing Overland Flow Types 58

3.4.2 Groundwater Flow `<transmissivity>` 59

3.5 Water Movers 68

3.5.1 Introduction to Water Movers 68

3.5.2 Default Water Movers 69

3.5.3 Concept Water Movers 69

3.5.3.1 Simple Power Law Based Water Mover `<standardweir>` 70

3.5.3.2 General Power Law Based Water Mover `<genweir>` 72

3.5.3.3 General Structure Water Mover `<genstruc>` 74

3.5.3.4 Rated Structure Water Mover `<ratedStruc>` 76

3.5.3.5 Coupled Source Sink Water Mover `<doublet>` 78

3.5.3.6 Controllable User-Defined Flow `<setflow>` 80

3.5.3.7 Lookup Table Based Water Movers 82

3.5.3.8 Single Control Water Movers <single_control> 82
3.5.3.9 Dual Control Water Movers <dual_control> 84
3.5.3.10 Delta Control Water Movers <delta_control> 86
3.5.3.11 Comments On The Use Of Lookup Tables 88
3.5.3.12 Shunt Watermover <shunt> 88
3.5.4 New And Borrowed Physical Structure Types 91
3.5.5 Culvert Water Mover <culvert> 91
3.5.5.1 MBR Pipe Flow <pipe> 97
3.5.5.2 MBR Broad Weir <mbrbroadweir> 99
3.5.5.3 MBR Sharp Weir <mbrsharpweir> 102
3.5.5.4 MBR Drop Weir <mbrdropweir> 105
3.5.5.5 NWS Uncontrolled Spill <spill> 108
3.5.5.6 NWS Gated Weir <gateweir> 111
3.5.6 Bleeders 114
3.5.6.1 V-Notch Bleeder <vnotchbleeder> 114
3.5.6.2 Circular Bleeder <circularbleeder> 117
3.5.6.3 Rectangular Bleeder <rectbleeder> 119
3.5.7 Bridges 121
3.5.8 Hydropower<hydropower> 124
3.6 Canal Network - The <network> Element 126
3.6.1 Canal Data Input Under The <network> Element 126
3.6.2 Canal Network Geometry File <geometry> 129
3.6.2.1 Description Of Nodes 131
3.6.2.2 Canal Cross Sectional Geometry 134
3.6.3 Stream-Aquifer Interaction 134
3.6.4 Stream-Overland Flow Interaction 136

3.6.5 Levee seepage 137
3.6.6 Initial Condition File <initial> 139
3.6.7 Overriding Canal Properties Using XML 139
3.7 Lakes and Ponds <lakes> 142
3.7.1 Rainfall and Evapotranspiration 146
3.7.2 Lake Seepage <lake_seepage> 148
3.8 Storage and Stage-Volume Converters - The XML <svconverter> Element 152
3.8.1 Representation Of A Flat Ground Surface 152
3.8.2 Representation Using A Lookup Table 153
3.8.3 Use Of More Than One Type of SV converter 156

4 Boundary Conditions 158

4.1 Boundary Conditions For Two-Dimensional Flow <mesh_bc> 159
4.1.1 Available Boundary Condition Types 159
4.1.2 Defining Attributes Of 2-D BC's 164
4.1.2.1 Definition Of BC Location <nodelist> and <walllist> 164
4.1.2.2 Interpolation Used For Wall Boundary Conditions 168
4.1.2.3 Time Series Data Format Used For Data Entry At Boundaries And Other Locations 169
4.1.3 Boundary Condition Types Available For Walls 169
4.1.3.1 No Flow BC For Walls <noflow> 170
4.1.3.2 Head BC For Walls <wallhead> 170
4.1.3.3 General Head BC For Walls <wallghb> 171
4.1.3.4 Uniform Flow BC For Walls <walluf> 172
4.1.4 Boundary Condition Types For Cells 172
4.1.4.1 Inflow BC <well> 172

4.1.4.2 Head Boundary Conditions For Cells <cellhead> 173
4.1.4.3 General Head Boundary Conditions For Cells <cellghb> . 174
4.2 Boundary Conditions for the Canal Network <network_bc> 175
4.2.1 Flow Boundary Condition <segmentsource> 178
4.2.1.1 Head Boundary Condition <segmenthead> 178
4.2.1.2 Installing A No-Flow Boundary Condition At Canal Junctions <junctionblock> 180
4.2.1.3 Uniform Flow In A Segment <uniformflow> 180
4.2.1.4 General Head Boundary Condition In A Segment <segmentghb> 181
4.2.1.5 Junction Head Boundary Condition <junctionhead> . . 181
4.3 Boundary Conditions For General Water Bodies 184
4.3.1 Sources And Sinks <source> 186
4.3.2 Boundary Conditions Based On Stage-Discharge Relationships <hq_relation> 186
4.4 Boundary Conditions For Lakes <lake_bc> 188
4.4.1 Sources And Sinks <lakesource> 190
4.4.2 Open Water Evaporation Boundary Condition <owet> 190

5 Hydrologic Process Module Approach and Models 192

5.1 HPM Types 192
5.2 Natural System HPMs 194
5.2.1 Natural Wetland System <layer1nsm> 195
5.2.2 Three Dimensional Groundwater Cell HPM <layerpc> 197
5.2.3 Multi-Basin Routing HPM <mbrcell> 198
5.2.4 Five Soil Layer HPM <layer5> 201
5.2.5 Unsaturated Soil HPM <unsat> 203
5.3 Urban HPMs 207

5.3.1 Impervious Area <imperv> 207
5.3.2 Precipitation-Runoff Routing HPM<prr> 209
5.3.2.1 Input Data 209
5.3.2.2 Initial Conditions 212
5.4 Complex HPMs 213
5.4.1 Water Management Systems <hub> 213
5.4.2 Large Agricultural Developments 215
5.4.2.1 Agricultural Irrigation Requirement HPM <afsirs> . . . 215
5.4.2.2 Drainage Collector Ditch HPM <pumpedditch> 223
5.4.2.3 Agricultural Impoundment HPM <agimp> 227
5.4.3 Urban Hubs <hub> 229
5.4.3.1 Consumptive Use <cu> 230
5.4.3.2 Urban Stormwater Retention/Detention HPM <urbandet> 232
5.4.4 Assignment Of HPMs To Various Land Use Types <indexed> . . . 236
5.4.5 Time Variation of HPM Parameters <ampmod> 237

6 Input and Output File Specifications 238

6.1 Time Series and Other Data Formats Used For Single Location Model Input 239
6.1.1 Constant Value 239
6.1.2 Rule Curve 241
6.1.3 DSS Time Series 243

7 RSM Post-Processing 250

7.1 Water Balance And Budgets 251
7.1.1 Water Budgets Of Water Bodies 251
7.1.2 Water Budgets Of Water Movers 251
7.1.3 Local and Global Mass Balance 253

7.2 RSM Output Options <output> 254
7.2.1 Saving Model Output <globalmonitor> 264
7.2.2 Water Budget Post-Processing 273
7.2.3 Monitoring Individual Points 279
7.3 RSM Uncertainty Analysis 280
7.3.1 Existing Capabilities For Evaluating RSM Uncertainty 281
7.3.2 Methods Available For Evaluating Model Results 282
7.3.3 Evaluation Based On The Significance Of Differences 282
7.4 RSM Graphical User Interface 283
7.4.1 Overview of The Current RSM GUI 283
7.4.2 Overview of The Early 2005 RSM GUI Development Activities 284

Bibliography **286**

A RSM Development History **288**

B Primer on Using XML **291**

B.1 What Is XML? 292
B.2 The RSM DTD File 293
B.3 The RSM XML Schema 294
B.3.1 How To Convert A DTD-Based RSM Input File To An XML Schema-Based Input File 295
B.3.2 How To Validate RSM Input Files Against The XML Schema 295
B.3.2.1 Case 1: Successful Validation Of Benchmark Problem 1 Using W3C Validating Routine 296
B.3.2.2 Case 2: Unsuccessful Validation Of Benchmark Problem 1 Using W3C Validating Routine 297

B.3.2.3 Case 3: Validation Of Models Having More Than One XML Input File . . . 298

B.3.2.4 Additional XML Details . . . 299

C Extending A 2D Model Into 3D - The XML `<multilayer>` Element 302

C.1 Overview of Building a 3D Model in RSM . . . 302

C.1.1 2D to 3D Grid Program . . . 302

C.1.1.1 Two-Dimensional Mesh File . . . 303

C.1.1.2 Added Layer File . . . 303

C.1.1.3 Output 2-D Mesh File . . . 305

C.1.1.4 Output Water Mover File . . . 305

C.1.2 Other Input Files And Modifications Needed For 3-D Groundwater Flow Modeling . . . 305

C.1.2.1 Starting Head File `<shead>` . . . 305

C.1.2.2 HPM Definition File `<HPM>` . . . 306

C.1.2.3 Horizontal Conductance Definition File `<transmissivity>` 306

C.1.2.4 SV Converter Definition File `<svconverter>` . . . 306

C.1.3 Putting It All Together . . . 306

C.2 Boundary Conditions For Three-Dimensional Flow`<multilayer>` . . . 307

List of Figures

1.1 RSM model execution flowchart. 8

2.1 The HSE root node and first-order children elements. 18

2.2 The `control` subelements. 19

2.3 The `mesh` subelements. 20

2.4 The `network` subelements. 21

2.5 The `watermovers` subelements. 22

2.6 The `lakes` subelements. 23

2.7 The `multilayer` subelements. 24

2.8 The `controller` subelements. 25

2.9 The `management` attributes. 26

2.10 The `output` subelements. 27

3.1 `<gridio>` file geometry 41

3.2 Discretization of a square area into 18 cells with 16 nodes.(See Table 6.10). . 50

3.3 Definition sketch for using lookup tables for transmissivity and conveyance. . 53

3.4 Definition sketch of a pipe. 97

3.5 Definition sketch of broad crested weir. 100

3.6 Definition sketch of a sharp crested weir. 102

3.7 Definition sketch of a drop weir. 106

3.8 Definition sketch of an uncontrolled spillway. 108

3.9 Definition sketch of a gated weir. 112

3.10 Definition sketch of bleeders. 115

3.11 Definition of cross sections used with the bridge routine. 122

3.12 A schematic of the canal network. 127

3.13 A schematic of the canal network with a levee. 131

3.14 Trapezoidal canal cross section. 134

3.15 A definition sketch showing flow interaction with the canal. 135

3.16 Definition sketch showing levee seepage. 137

3.17 Plan view showing the placement of a levee. 138

3.18 Schematic diagram of a reservoir formed in a river. 143

3.19 Discretization around a lake and a pond. 143

3.20 Describing stage-storage characteristics in micro-topography. 153

4.1 Illustration of the application of 2D Mesh Boundary Conditions. 164

4.2 Illustration of the application of Canal Network Boundary Conditions. 183

5.1 HPM Components of Water Budget for the `<layer1nsm>` HPM. The variation of the Reference ET Crop Coefficient as a function of input parameters is also shown. 195

5.2 Soil layers modeled in the `<layer5>` HPM and the variation of the ET coefficient, Kc with water table. 201

5.3 Schematic water budget for `<unsat>` HPM. 204

5.4 Conceptual diagram of the `<prr>` HPM. 210

5.5 Structure dimensions of the `<urbandet>` HPM discharge weir and bleeder. 233

C.1 Sketch of the multi-layered grid used to solve 3-D groundwater flow. 303

List of Tables

1 Revision history for the RSM User's Manual. 3

2 Acronyms used in RSM 3

2.1 Basic data types used in HSE. 10

2.2 Definition of elements defined in the `<hse>` root element. 14

3.1 Attributes defined with the XML `<control>` element. 30

3.2 Default units used by HSE. 34

3.3 Definition of data entered under subelements of the `<mesh>` element. 36

3.4 Specification of the geometry file under `<mesh>`. 37

3.5 Additional mesh elements. 39

3.6 Elements for specifying input formats for additional mesh properties under `<mesh>`. 40

3.7 Elements and attributes used with the `<indexed>` element. 42

3.8 Attributes used with `<gridio>`. 43

3.9 Attributes used with `<gms>`. 44

3.10 Attributes used with `<netcdf>`. 45

3.11 two-dimensional GMS mesh data file `"mesh3x3.2dm"`. 49

3.12 Elements and attributes under `<conveyance>`. 54

3.13 Overland flow options for the "compute" attribute under `<conveyance>`. . . 59

3.14 Allowable options under `<transmissivity>`. 60

3.15 Definition of variables for <confined> under <transmissivity>. 63

3.16 Definition of variables for <unconfined> under <transmissivity>. . . 63

3.17 Definition of variables for <confined_gms> under <transmissivity>. . 64

3.18 Definition of variables for <unconfined_gms> under <transmissivity>. 64

3.19 Definition of variables for <confined_gms_layer> under <transmissivity>. For 3D flow analysis (see Appendix C) 65

3.20 Definition of variables for <unconfined_gms_layer> under <transmissivity>. For 3D flow analysis (see Appendix C) 66

3.21 Definition of variables for <layered> under <transmissivity>. For 3D flow analysis (see Appendix C) 66

3.22 Definition of variables for <layered_gms_layer> under <transmissivity>. For 3D flow analysis (see Appendix C) 67

3.23 Definition of variables for <lookuptr> under <transmissivity>. 67

3.24 List of concept water movers. 70

3.25 Attribute definitions for <standardweir>. 71

3.26 Attribute definitions for <genweir>. 73

3.27 Attribute definitions for the <genstruc> water mover. 75

3.28 Attribute definitions for the <ratedStruc> water mover. 77

3.29 Element and attribute definitions for <doublet>. 79

3.30 Attribute definitions for <setflow>. 81

3.31 Attribute definitions for the <single_control> Water Mover. 83

3.32 Attribute definitions for the <dual_control> Water Mover. 85

3.33 Attribute definitions for the <delta_control> Water Mover. 87

3.34 Attribute definitions for a shunt Water Mover. 90

3.35 List of physical water movers. 92

3.36 Attributes of the <culvert> water mover. 96

3.37 Attributes used to define a <pipe> water mover. 98

3.38 Attributes of a broad crested weir, <mbrbroadweir>. 101

3.39 Attributes of a sharp crested weir, <mbrsharpweir>. 104

3.40 Attributes of a <mbrdropweir>. 107

3.41 Attributes of an uncontrolled spillway <spill>. 110

3.42 Attributes of <gateweir>. 113

3.43 Attributes of <vnotchbleeder>. 116

3.44 Attribute definitions for <circularbleeder>. 118

3.45 Attribute definitions for <rectbleeder>. 120

3.46 Attribute definitions for <yarnell>. 123

3.47 Attributes of <hydropower>. 125

3.48 Sub-elements and attributes under <network>. 128

3.49 Definition of attributes specified in the canal geometry and boundary condition files in GMS format. 130

3.50 Sample canal geometry file, part 1 of 2. 132

3.51 Sample canal geometry file, part 2 of 2. 133

3.52 Sample index file. 140

3.53 Skeleton xsentry that modifies no values. 141

3.54 Sub-elements and attributes used to define lake properties under the <lake> element. 144

3.55 Elements and attributes used to define lake area and volume under the <lakes> element. 145

3.56 Elements and attributes used to define <EvapRainStressors>. 147

3.57 Sample XML input for lakes and ponds. 149

3.58 Elements and attributes used to define the lake seepage water mover. 150

3.59 Sample XML input for lake seepage. 151

3.60 Elements and attributes used to define a <lookupsv> SV converter in the <mesh> environment. 155

4.1 Elements and attributes used to describe two-dimensional boundary conditions applied to cells in the `<mesh_bc>` environment. Element names are highlighted. 161

4.2 Elements used to define the `<wallhead>` and `<wallghb>` boundary conditions applied to walls in the `<mesh_bc>` environment. The elements are in shaded cells. 162

4.3 Elements and attributes used to define the `<noflow>` and `<walluf>` boundary conditions applied to walls. The elements are in shaded cells. 163

4.4 Example XML input for 2-D boundary conditions. 165

4.5 Elements and attributes used to assign interpolation weighting to the `<wallhead>` and `<wallghb>` boundary conditions. Element cells are shaded. 167

4.6 Elements and Attributes for Specifying Boundary Conditions for Canal Networks Part 1. Element cells are shaded. 176

4.7 Elements and Attributes for Specifying Boundary Conditions for Canal Networks Part 2. Element cells are shaded. 177

4.8 Example XML input canal network boundary conditions. 179

4.9 Elements and Attributes for Specifying Boundary Conditions for General Water Bodies in the `<watermover>` environment. Element cells are shaded. . 185

4.10 Example XML input for general water body boundary conditions. 186

4.11 Elements and Attributes for Specifying Boundary Conditions for Lakes. Element cells are shaded. 189

4.12 Example XML input for lake body boundary conditions. 190

5.1 Elements and Attributes for the `<layer1nsm>` HPM. Element cells are shaded. 196

5.2 Example XML input for a `<layer1nsm>` HPM. 197

5.3 Example XML input for a `<layerpc>` HPM. 198

5.4 Elements, attributes, and typical values used for the `<mbrcell>` HPM. . . 199

5.5 Example XML for an `<mbr>` HPM. 200

5.6 Elements and Attributes for the `<layer5>` HPM. 202

5.7 Example XML for `<layer5>` implementation. 203

5.8 Water table location, available water content and crop coefficient values for <unsat> HPMs. 204

5.9 Elements and attributes for the <unsat> HPM. 205

5.10 Example XML for an <unsat< HPM. 206

5.11 Elements and attributes for the <imperv> HPM. 208

5.12 Example xml for the <imperv> HPM. 209

5.13 Definition of attributes of the <prr> HPM 211

5.14 Example XML code from benchmark 56 for a <prr> HPM. The parameters are described in Table 5.13 212

5.15 Elements and attributes for the <hub> HPM. 214

5.16 Example XML for typical complex <hub> containing native, agricultural and urban HPM types. 216

5.17 Example XML for typical complex <hub> containing native, agricultural and urban HPM types (continued). 217

5.18 Elements and attributes used for the <afsirs> HPM. 219

5.19 Example XML for an <afsirs> HPM. 224

5.20 Example XML for an <afsirs> HPM (continued). 225

5.21 Elements and attributes and typical values used for the <pumpedditch> HPM as a component of a <hub>. 226

5.22 Example xml for <pumpedditch> HPM. 227

5.23 Elements, attributes and typical values used for the <AgImp> HPM as a component of a <hub>. 228

5.24 Typical example xml for an <agimp> HPM within a hub. 229

5.25 Elements and attributes for the <cu> object. 231

5.26 Example xml for consumptive use in HPM. 232

5.27 Elements and attributes for the <urbandet> HPM as a component of a <hub>. 234

5.28 Example xml for <urbandet> HPM in a hub. 235

5.29 Example index file for assigning HPMs to mesh cells. 236

5.30 Example XML for implementation of kveg parameter modification. 237

6.1 Elements and attributes used to define a `<const>` input value. 240

6.2 Sample XML for specifying a constant `<refet>`. 241

6.3 Elements and attributes used to define a rule curve and to use it. 242

6.4 Sample XML for specifying a rule curve `<rc>` and using it in the specification of mesh boundary conditions.. 244

6.5 Elements and attributes used for specifying time series data in a `<dss>` file. 245

6.6 Elements and attributes used to define a TSnode and to use it. 246

6.7 Path name definition for time series data in DSS format. 248

6.8 Default units used by the RSM model. 249

7.1 Reported water budget components of a water body. 252

7.2 Reported water budget components of a water mover. 252

7.3 Model output options available using `<output>`. 255

7.5 Attributes available with `<globalmonitor>`. 264

7.4 Time series formats available within the output options in Table 7.3. 265

7.6 Variables that can be monitored using `<cellmonitor>`. The usage is: <cellmonitor id="*cellid#*" attr="*attribute*"> <filetype in Table 7.4 > < /cellmonitor> 269

7.7 Variables that can be monitored using `<segmentmonitor>`. 270

7.8 Variables that can be monitored using `<junctionmonitor>`. 271

7.9 Variables that can be monitored using `<wmmonitor>`. 271

7.10 Variables that can be monitored using `<bcmonitor>`. 271

7.11 Variables that can be monitored using `<lakemonitor>`. 271

7.12 Variables that can be monitored using `<lkstressmonitor>`. 272

7.13 Variables that can be monitored using `<assessormonitor>`. 272

7.14 Variables that can be monitored using `<ctrlmonitor>`. 272

7.15 Variables that can be monitored using `<flowgage>`. 273

7.16 Variables that can be monitored using `<pseudomonitor>`. 274

7.17 Variables that can be monitored using `<reachmonitor>`. 275

7.18 Variables that can be monitored using `<wcumonitor>`. 276

7.19 Variables that can be monitored using `<spvrmonitor>`. 277

7.20 Variables that can be monitored using `<statmonitor> or <statemonitor>`. 277

7.21 Variables that can be monitored using `<trmonitor>`. 277

7.22 Variables that can be monitored using `<basinmonitor>`. 278

7.23 Variables that can be monitored using `<trigzonemonitor>`. 278

7.24 Variables that can be monitored using `<tkprmonitor>`. 278

C.1 Variables defined in the layer data input file. 304

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 hydrologic process modules (HPMs), which calculate the local water balance on a cell by cell basis. HPMs provide a method to simulate the local surface hydrology in a mesh cell or a collection of mesh cells.

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 technically reviewed, assembled, edited, and formatted by Ken Black, Robert Yager, and Timothy Newton of 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
HPM	Hydrologic Process Module
HSE	Hydrologic Simulation Engine
LEC	Lower East Coast
LPM	Local Process Module (formerly called HPMs)
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

Table 2 continued on next page

Acronym	**Description**
WASH123	WAterSHed Systems of 1-D stream-river networks, 2-D overland regimes and 3-D subsurface media (USACE Regional Model under development)
WCA	Water Conservation Area
WCU	Water Control Unit
WS	Water Supply
XML	Extensible Markup Language

Chapter 1

Introduction

This RSM Version 2.2.9 user's guide has been written as a companion to the RSM theory manual[1]. The purpose of this manual is to provide information necessary to build an RSM model or to modify the input and output of an existing model. New RSM users should consult the theory manual to gain an understanding of the conceptual framework upon which RSM is built. The theory manual discusses topics such as governing equations, numerical solution techniques, Hydrologic Process Modules (HPM's) and other topics.

1.1 Programming Information

The RSM model is a continuously evolving, object-oriented model coded in C++. In May 2004, RSM version 2.2.2 contained 216 source files (.cc and .h files) with nearly 46,000 lines of computer code (comments, blank lines, and active code) with 29,000 lines of active code. There were 263 classes, 3,085 functions, 10,976 declarative statements, and 14,764 executable statements. About one year later in April 2005, RSM version 2.2.9 contained 245 source files, nearly 57,000 lines of computer code, and 32,000 lines of active code. There were 302 classes, 3,587 functions, 13,368 declarative statements, and 18,441 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. It is possible that future releases of RSM will be made available for the Windows operating system.

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

1.1.1 Programming Details

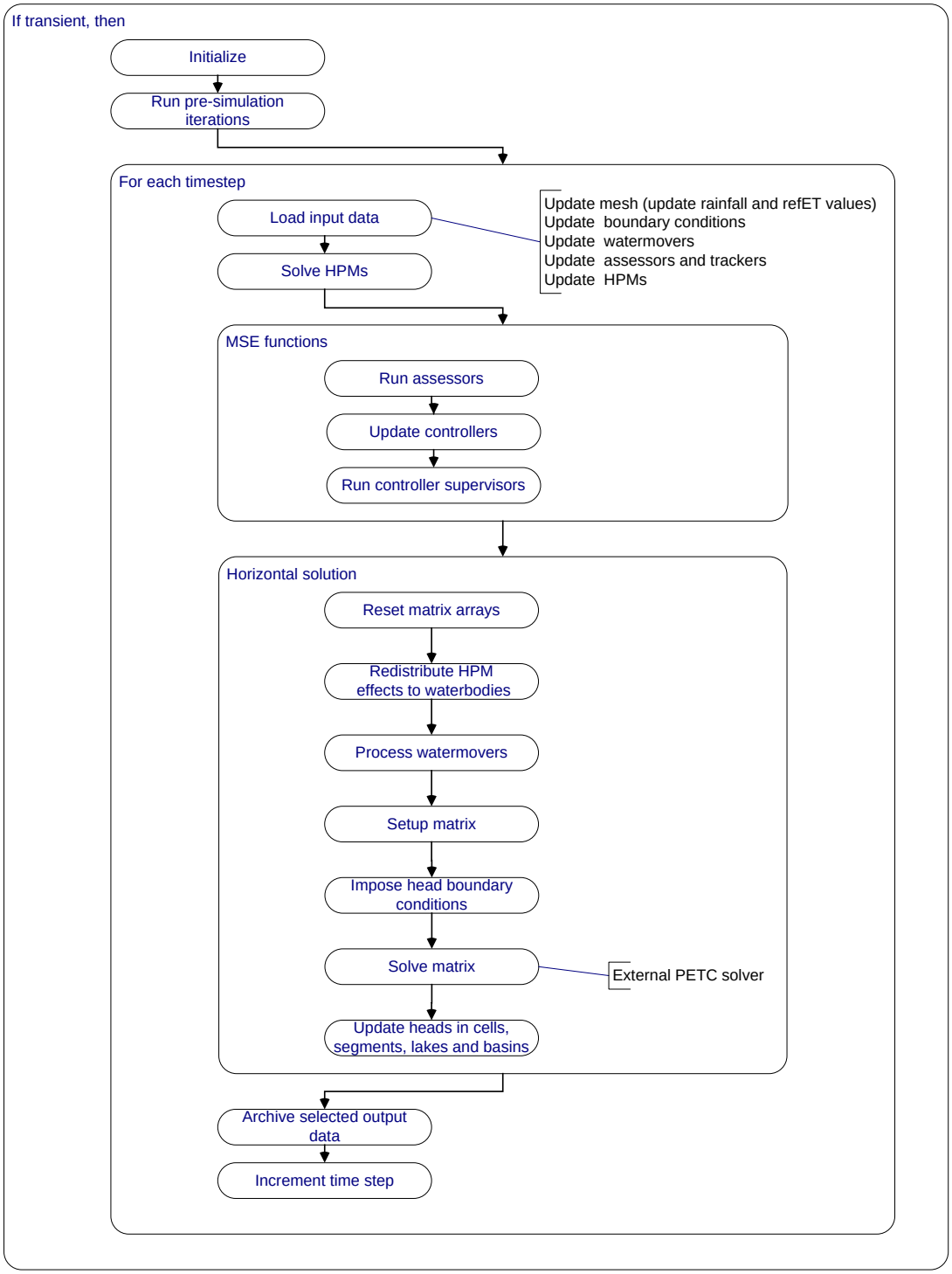
A variety of code analysis methods have been used to gain an understanding of the RSM program structure. The nature of large object-oriented codes can make them difficult to comprehend and to visualize. A comprehensive code analysis of RSM 2.2.9 has been completed and includes a variety of ways of examining the code. The RSM 2.2.9 reports can be viewed here.[2]. The RSM 2.2.9 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.[3].

1.2 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 a typical transient RSM simulation 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 in Figure 1.1.

[2]http://gwmftp.jacobs.com/models_2_2_9/hse_2_2_9_html/index.html
[3]http://gwmftp.jacobs.com/models_2_2_9/hse_2_2_9_sp/index.html

Figure 1.1: *RSM model execution flowchart.*



Chapter 2

RSM Input Using XML

RSM data are 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.

2.1 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 2.1.

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]
```

where `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 hy-

Table 2.1: *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.); HPMs 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 HPMs, 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.

drologic simulation time steps. This check currently is set to compare the input data against the Data Type Definition (DTD), not the schema. If users prefer a higher degree of data validation, they should execute the schema validation steps outlined in subsection 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 with the exception of the 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.

2.1.1 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

2.1.2 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

2.1.3 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 subsection 7.4.2. This pre-processor will directly create XML input data from GIS coverages, for example. Current functionality for the pre-processor is described in subsection 7.4.1. 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 the RSM theory manual (see the Model Stability Guideline Section).
2. *Create mesh physical properties indexed files.* This step is currently under development. For more information, see section 7.4.
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 subsection 7.4.1 and subsection 7.4.2

2.1.4 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 subsection 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 HPMs SYSTEM "pseudo.xml"> <!ENTITY landscape
SYSTEM "landscape.xml"> ]> <hse version="0.1">
  <control
    tslen="24"
    .....
  </control>

 &HPMs;
 .....

]>
```

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 HPMs 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 2.2 and depicted in Figure 2.1.

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 2.2: *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 3.1 for specific information.
`<mesh>`	Information regarding the 2-D mesh are defined within this XML element. See Table 3.4 for specific information.
`<network>`	Information regarding the canal network are defined within this XML element. See Table 3.4 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 cyclic input data.
`<basins>`	Not fully implemented in this version.
`<tsNodes>`	Specifying an object that contains time series data.
`<assessors>`	Specifying assessor information.
`<mse_networks>`	To be defined.

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.

2.1.5 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 3.2.

2.2 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 are pre-processed to accurately reflect the physical parameters of the modeled domain.

2.2.1 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 2.1. 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 **`output`** sub-elements are shown in Figure 2.10.
- The **`control`** sub-elements are shown in Figure 2.2.
- The **`mesh`** sub-elements are shown in Figure 2.3.
- The **`network`** sub-elements are shown in Figure 2.4.
- The **`watermovers`** sub-elements are shown in Figure 2.5.
- The **`controller`** sub-elements are shown in Figure 2.8.
- The **`management`** attributes are shown in Figure 2.9.
- The **`rulecurves`** KCB add.
- The **`tsNodes`** KCB add.
- The **`lakes`** sub-elements are shown in Figure 2.6.
- The **`basins`** KCB add.
- The **`assessors`** KCB add.
- The **`mse_network`** KCB add.
- The **`streambanks`** KCB add.
- The **`multilayer`** sub-elements are shown in Figure 2.7.

2.3 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

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

- **doc** - documentation
- **fcl_lib** - fuzzy control library
- **glop** - GNU linear programming kit library
- **psbud** - HPM water budget package
- **src** - source code and executable

2.3.1 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 (see the Benchmark and Verification Manual for more details). A complete listing of all benchmarks described in detail is available here[2]. 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.

[2] http://gwmftp.jacobs.com/benchmarks/bm_des.pdf

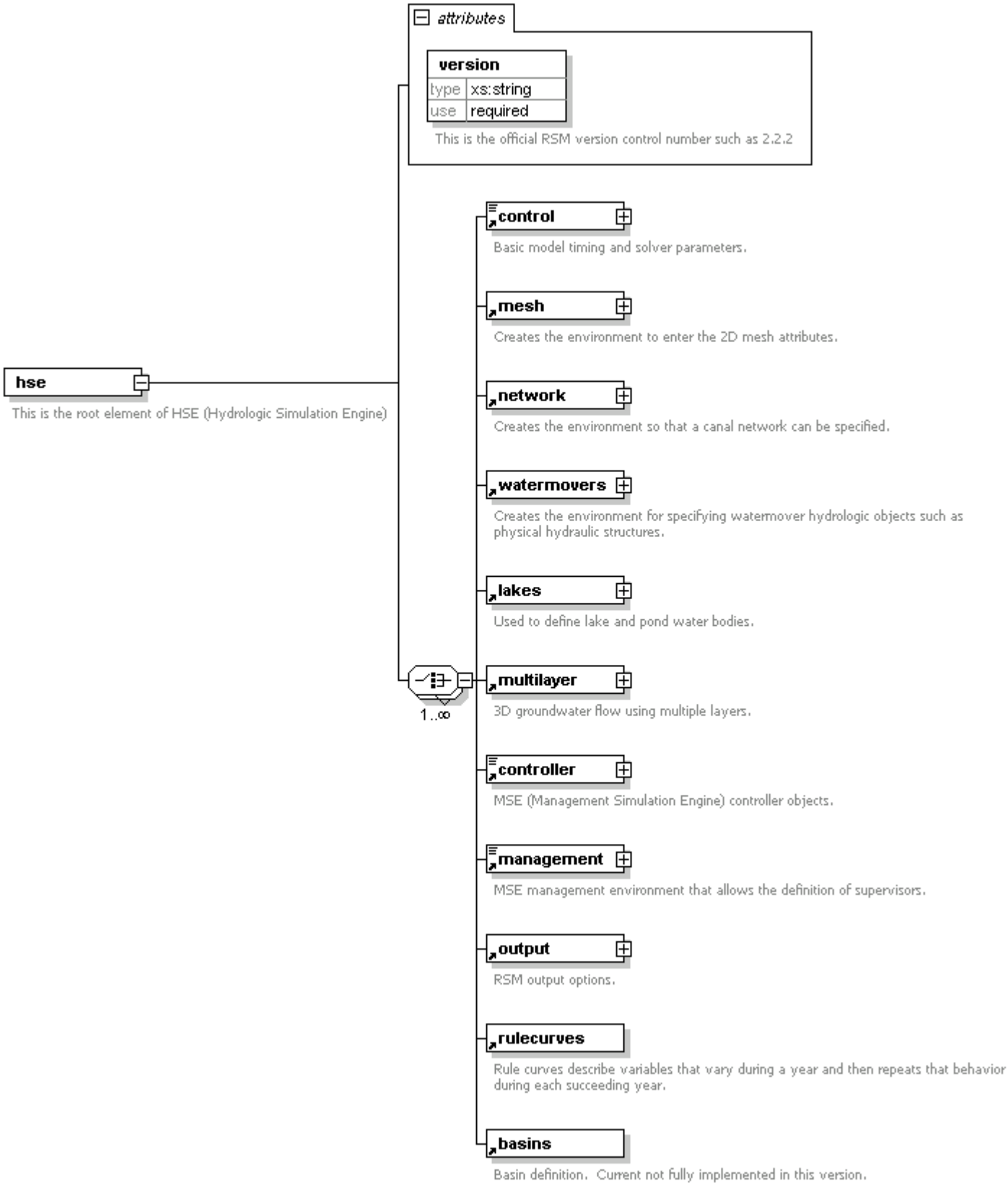


Figure 2.1: *The HSE root node and first-order children elements.*

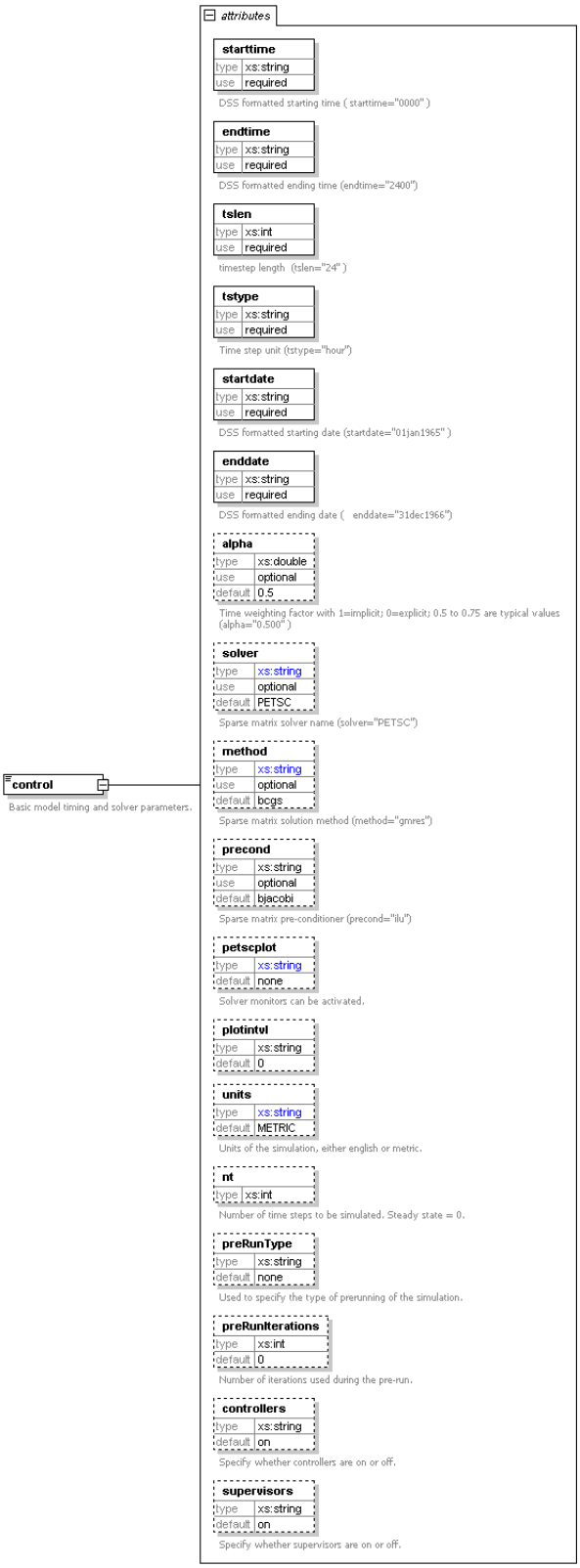


Figure 2.2: *The* `control` *subelements.*

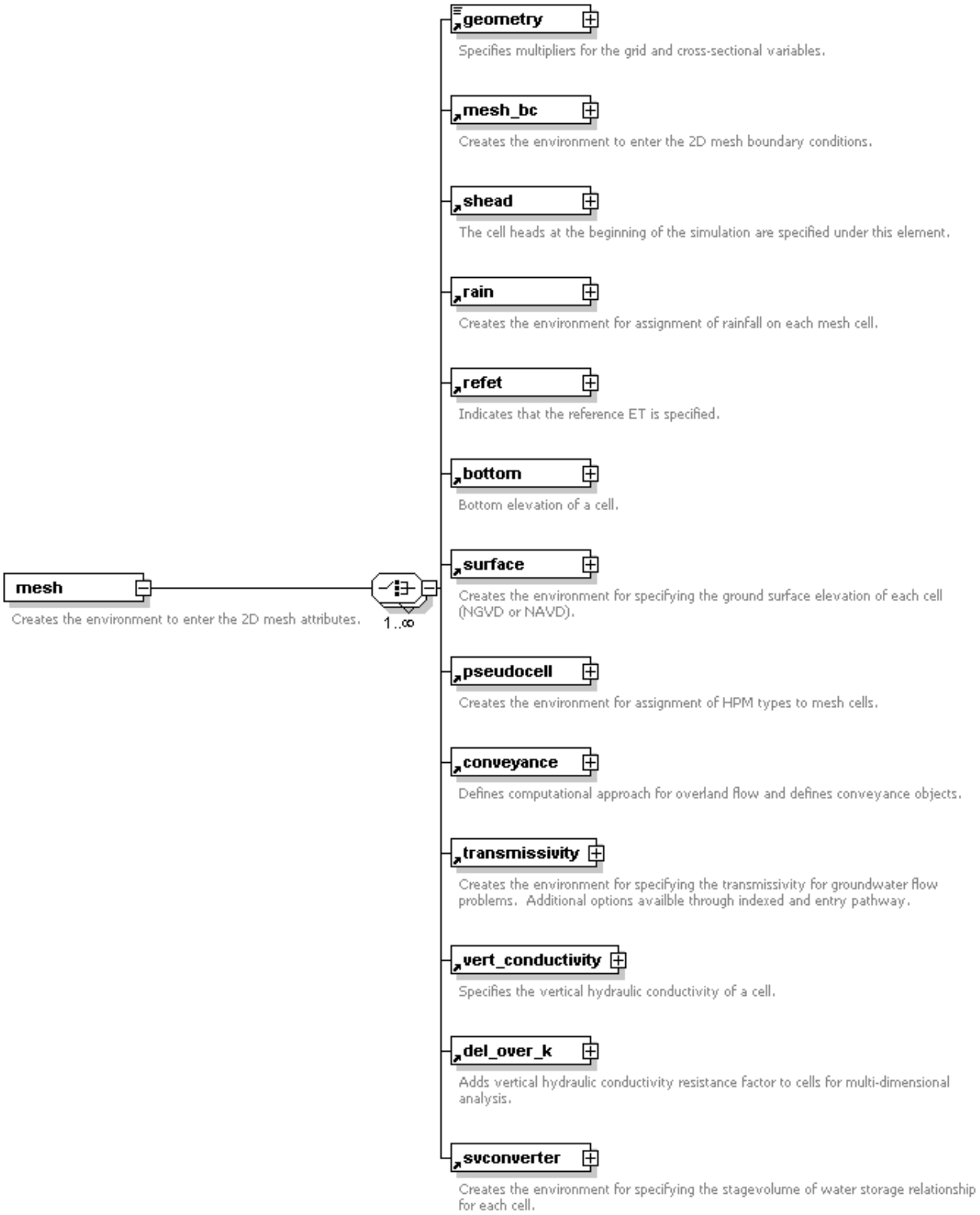


Figure 2.3: *The* `mesh` *subelements.*

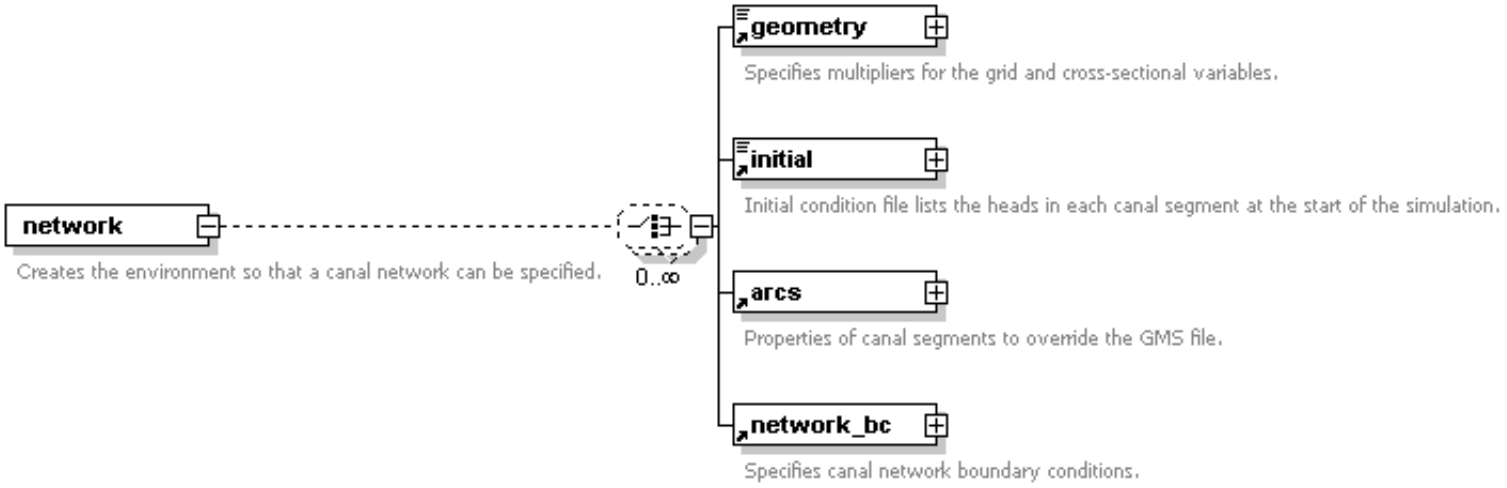


Figure 2.4: *The* `network` *subelements.*

Figure 2.5: *The* `watermovers` *subelements.*

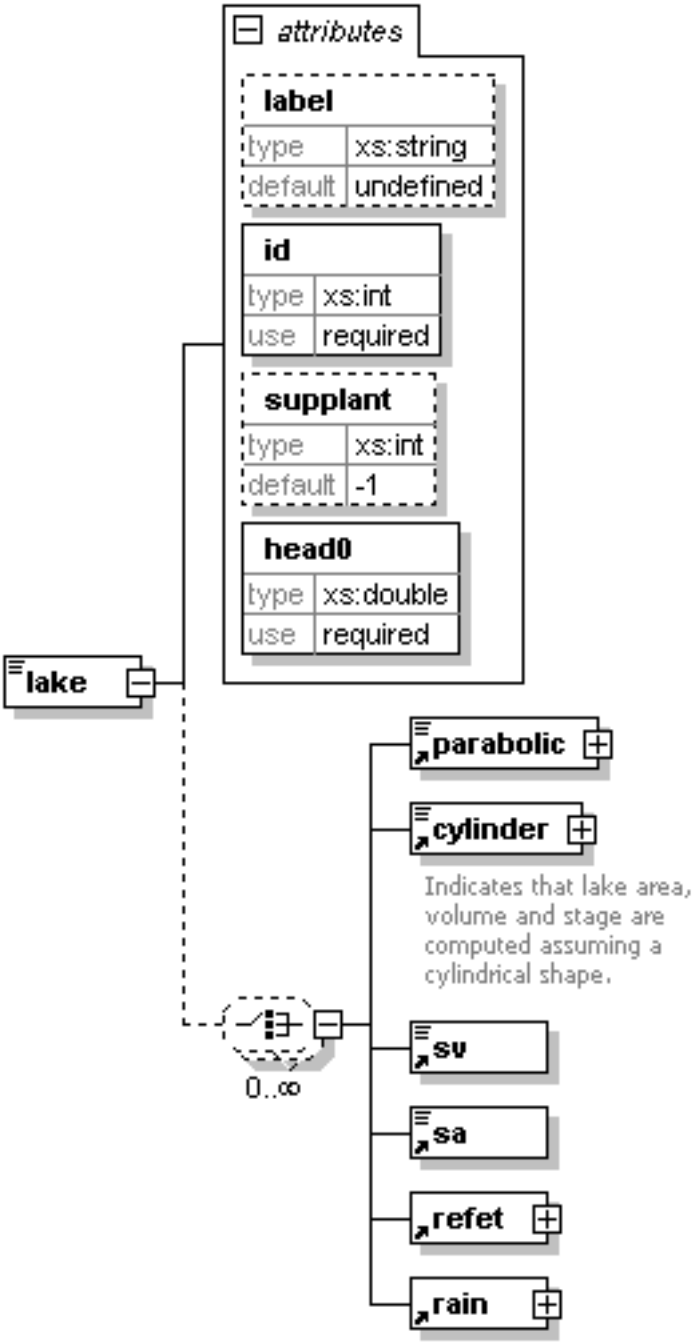


Figure 2.6: *The* `lakes` *subelements.*

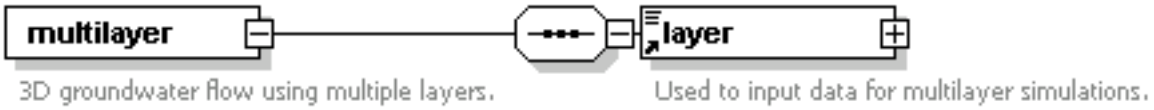


Figure 2.7: *The* `multilayer` *subelements.*

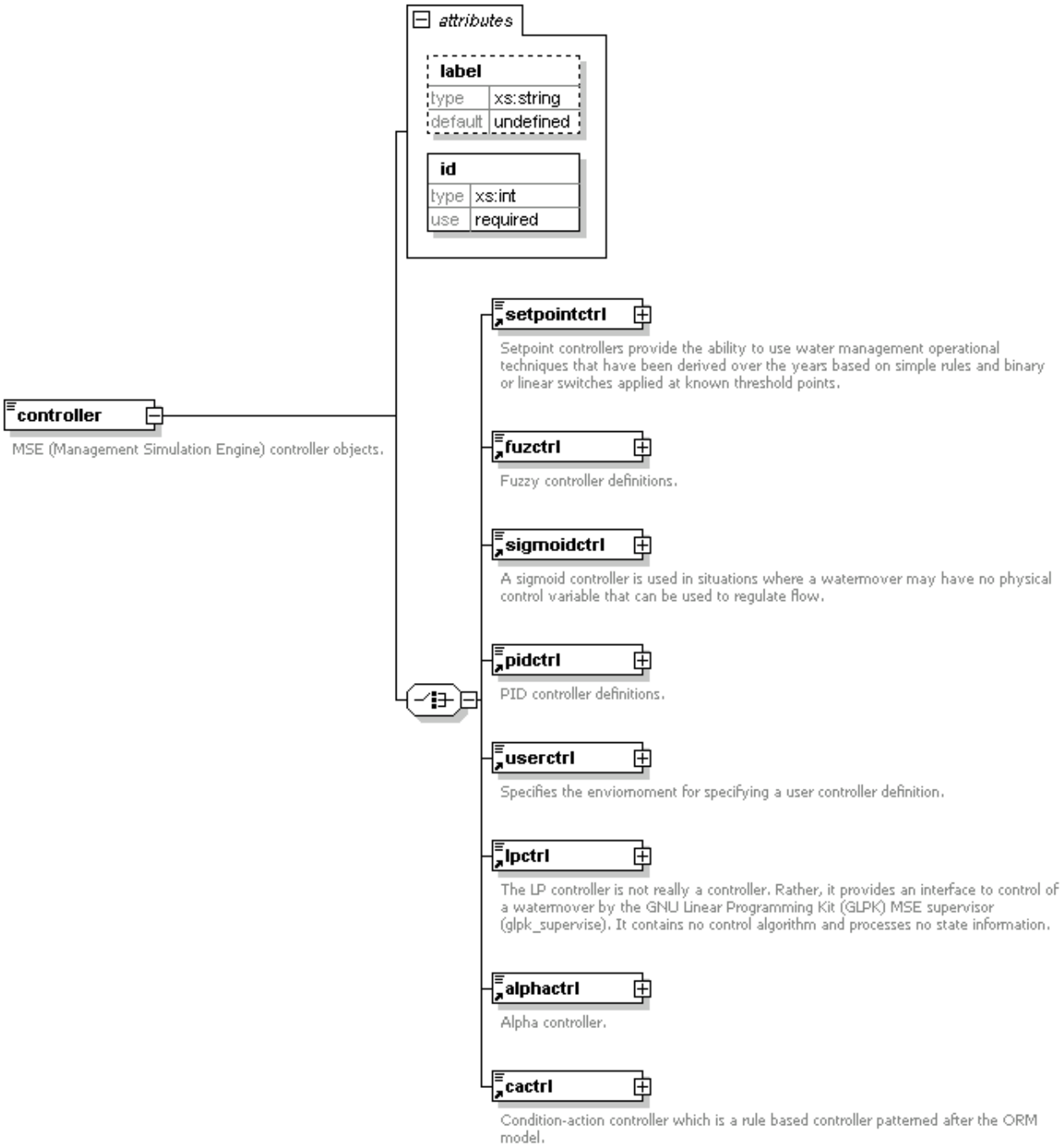


Figure 2.8: *The* `controller` *subelements.*

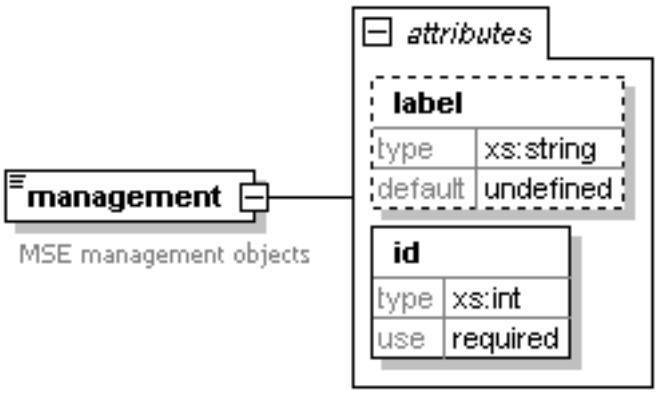


Figure 2.9: *The* `management` *attributes.*

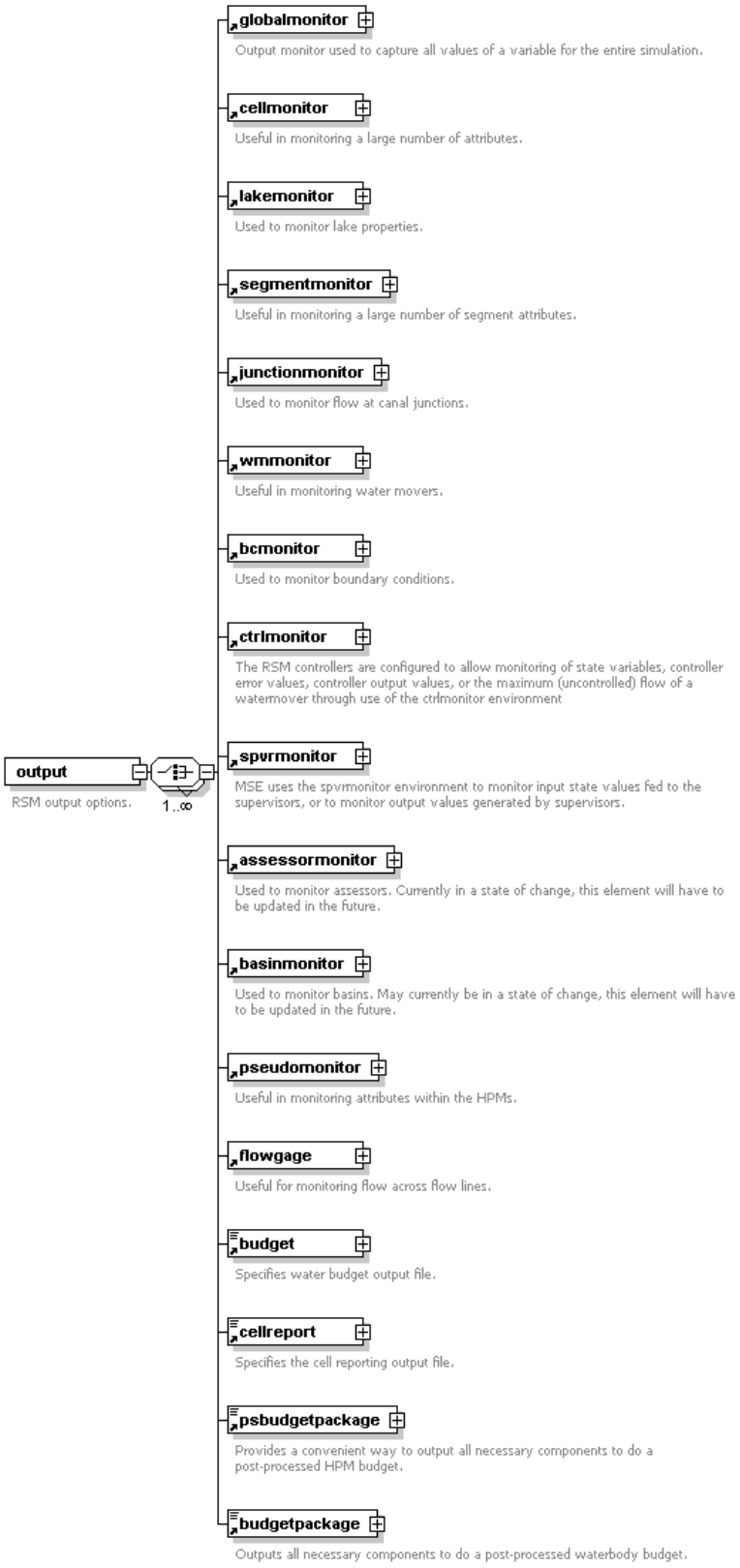


Figure 2.10: *The* `output` *subelements.*

Chapter 3

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.

3.1 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 3.1 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 3.1: *Attributes defined with the XML* `<control>` *element.*

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`starttime`	Starting time of the simulation	NA	DSS format hhmm	0000 to 2359	Req.	`starttime="2315"`
`endtime`	Ending time of the simulation	NA	DSS format hhmm	0000 to 2359	Req.	`endtime="0120"`
`tslen`	Timestep length	T	Integer	>0	Req.	`tslen="24"`
`tstype`	Timestep unit	NA	String	`Second, minute, hour, day, week`	Req.	`tstype="hour"`
`startdate`	Starting date of the simulation	NA	DSS format ddmmmyyyy	Any valid date	Req.	`startdate ="01jan1994"`
`enddate`	Ending date of the simulation	NA	DSS format ddmmmyyyy	Any valid date	Req.	`enddate ="30jan1994"`
`alpha`	Time weighting factor	NA	Real	1.0 = implicit, 0.5 = mid, 0.0 = explicit	Req.	`alpha="0.75"`
`solver`	Sparse solver name	NA	String	`PETSC` is the only option	Req.	`solver="PETSC"`
`method`	Sparse solver method	NA	String	See `PETSC` manual	Req.	`method="gmres"`
`precond`	Pre-conditioner	NA	String	See `PETSC` manual	Req.	`precond="bjacobi"`
`nt`	Number of time steps to be simulated. Steady state = 0	NA	Integer	Any integer	10	`nt="10"`
`units`	Model units	NA	String	"METRIC", "ENGLISH" is optional	METRIC	`units="english"`
`petscplot`	Solver monitors	NA	String	Options: "none", "yes", "text", "ksp", "all"	none	`petscplot="all"`

Table 3.1 continued on next page

Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
`preRunType`	Used to specify the type of pre-running of the simulation.	NA	String	Options include "controller", "filter", "all", "none"	none	`preRunType ="none"`
`preRun-Iterations`	Number of iterations used during the pre-run.	NA	Integer	Any integer, defaults to 0.	0	`preRunIterations ="0"`
`controllers`	Activate controllers	NA	String	"on" or "off"	on	`controllers="on"`
`supervisors`	Activate supervisors	NA	String	"on" or "off"	on	`supervisors="on"`
`runDescriptor`	Brief description of simulation	NA	String	Any String	none	`runDescriptor="NSRSM V2.1"`
`defDBintl`	Default database interval in DSS files	T	Integer	Time interval in minutes	0	`defDBintl="1440"`
`plotintvl`	Number of time steps between plots	NA	Integer	0 to 100	0	3
NA = Not Applicable; Req. =Required						

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>
```

3.1.1 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 3.1. 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 3.2.

Table 3.2: *Default units used by HSE.*

Quantity	**Unit**	**Type**
Head	METERS	INST-VAL
Flow	CU_METER/SEC	INST-VAL
Rain	METERS	PER-CUM
ET	METERS/time step	PER-CUM
Depth	METERS	INST-VAL
Water level	METER	INST-VAL
Transmissivity	$METER^2/SECOND$	PER-AVER
Definition of Available Unit Types		
Type	**Definition**	**Example**
PER-AVER	Period Average	Daily flow
PER-CUM	Period Cumulative	Monthly flow (volume)
INST-VAL	Instantaneous	Breakpoint Stage
INST-CUM	Instantaneous Cumulative	Rain mass curve

3.2 Data For The Two-Dimensional Model <mesh>

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 HPM to each mesh cell. The remaining data under the <mesh> element are the forcing functions that drive the regional flow. These are boundary conditions for the cells and the walls that divide the cells, and rainfall and evapotranspiration used by HPMs, which in turn produce a forcing function on the mesh.

Data for the 2-D model are entered in the <mesh> environment. The types of data that can be input under <mesh> are listed in Table 3.3 and are described in detail in subsequent sections. The elements under which these data are input are listed in Table 3.4, Table 3.5, and Table 3.6. Table 3.4 describes how to specify the 2-D geometry file that specifies the node locations. The elements listed in Table 3.5 describe data that are explained in greater detail in later sections, and Table 3.6 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 3.6 are listed in Table 6.1 and Table 6.5.

Table 3.3: *Definition of data entered under subelements of the* `<mesh>` *element.*

Element	Definition
`<geometry>`	The geometry of the mesh for the model.
`<mesh_bc>`	Mesh boundary conditions such as for cells and walls.
`<shead>`	The starting heads in each of the mesh cells.
`<rain>`	The precipitation on each cell.
`<refet>`	The reference evapotranspiration for each cell is assigned.
`<bottom>`	The bottom elevation of each mesh cell (the lower limit of the model).
`<surface>`	The land surface elevation for each cell.
`<pseudocell>`	Specifies which pseudocell type is associated with each mesh cell.
`<conveyance>`	Specifies the conveyance formulation for each cell .
`<transmsissivty>`	The formulation of transmissivity for each mesh cell.
`<vert_conductivity>`	Still to be defined.
`<del_over_k>`	Still to be defined.
`<svconverter>`	The relationship between stage and volume in each mesh cell.

Table 3.4: *Specification of the geometry file under* `<mesh>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<geometry>`	Creates the environment to specify a geometry file in GMS format.					
file	The name of the file	NA	String	Any valid GMS file name	Req.	enp.gms
mult	Multiplier for mesh node coordinates, often for unit conversion	NA	Real	Any valid real	1.0	0.3048
Details of the GMS file are explained in section 3.3. NA = Not Applicable; Req. = Required						

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

Under the <indexed> element entries can be defined as described in Table 3.7. Each entry specifies an input, in this case a time series, rule curve, or constant. These inputs may be applied to cells using an index file in which each cell is assigned an index number to specify which input is associated with each cell. The indexing concept has more general application as described throughout this manual. For example the indexing method may be used to assign HPMs to mesh cells or hydraulic properties to canal segments. Other formats for assigning input data to the cells in the mesh are <gridio>, <gms> and <netcdf>. Formats for <indexed>, <gridio>, <gms> and <netcdf> and the attributes are shown in Table 3.7, Table 3.8, Table 3.9, and Table 3.10. Examples are shown later in this section. Format and attributes for <const> and <dss> are explained in chapter 6.

Data in a <gridio> file are referenced to the xorig and yorig attributes in Table 3.8. The height and width of each grid cell are specified in the file along with data values associated with a row number and column number. By specifying xorig and yorig as shown in Figure 3.1 the data are assigned to locations at the centers of the appropriate cells. The model then interpolates these data to assign values to the triangular mesh cells.

Table 3.5: *Additional mesh elements.*

Element	Definition	Available subelements
`<transmissivity>`	The element under which transmissivity for groundwater flow is specified. The confined_gms, confined_gms_layer and layered_gms_layer elements are available only under the indexed and entry pathway. Details are given in subsection 3.4.2	`<indexed>` `<confined>` `<confined_gms>` `<confined_gms_layer>` `<layered>` `<layered_gms_layer>` `<lookuptr>` `<unconfined>` `<unconfined_gms>` `<unconfined_gms_layer>`
`<conveyance>`	Creates the environment for specifying the calculation of conveyance for overland flow. Details are given in subsection 3.4.1	`<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 3.8	`<indexed>` `<constsv>` `<lookupsv>` `<layersv>`
`<HPM>`	Assignment of HPM types to mesh cells are made under this element. Details are given in chapter 5	`<indexed>` `<layer5>` `<layer1nsm>` `<nam>` `<unsat>` `<mbrcell>` `<layerpc>`
`<mesh_bc>`	Creates the environment to specify the mesh boundary conditions for cells and walls. Details are given in chapter 4	`<well>` `<cellhead>` `<cellghb>` `<walluf>` `<wallhead>` `<wallghb>` `<noflow>` `<bermseepage>`

Table 3.6: *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 (NGVD or NAVD).	`<indexed>`
		`<const>`
		`<gms>`

`<surface>`	Creates the environment for specifying the ground surface elevation of each cell (NGVD or NAVD).	`<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>`
		`<tsin>`

`<refet>`	The reference crop potential evapotranspiration for each cell during the simulation is specified in this environment.	`<indexed>`
		`<const>`
		`<dss>`
		`<gridio>`
		`<netcdf>`
		`<tsin>`

Figure 3.1: `<gridio>` *file geometry*

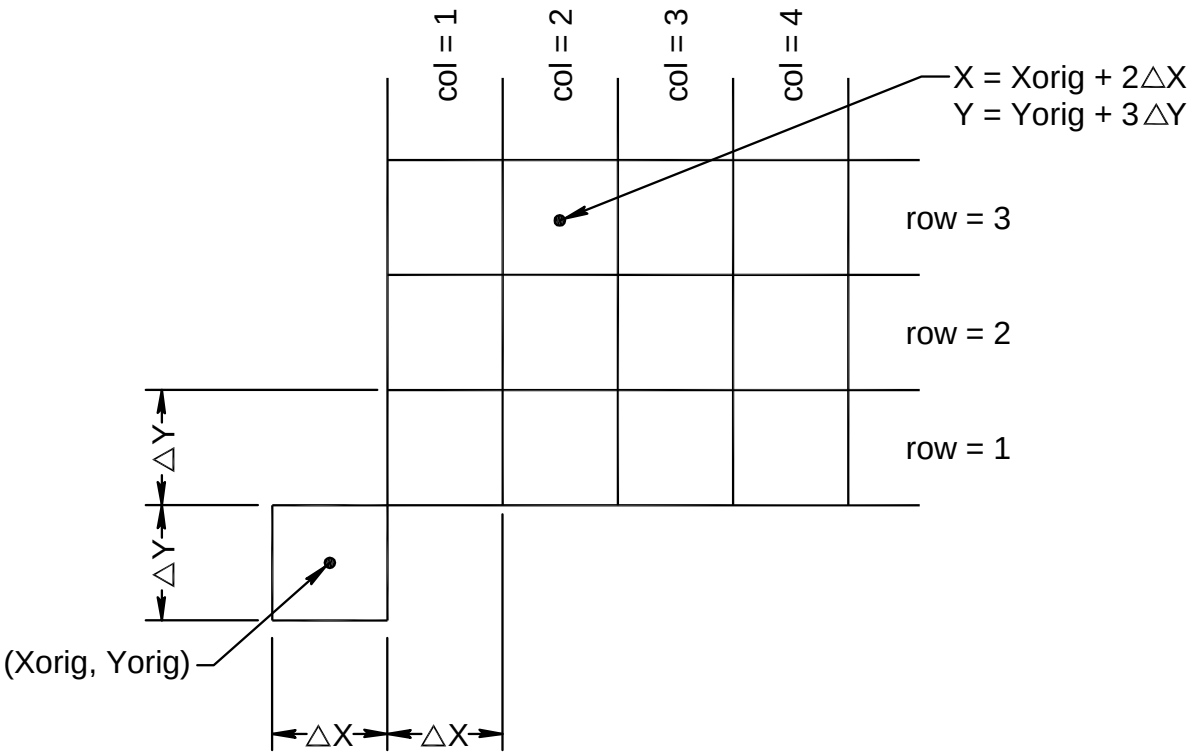


Table 3.7: *Elements and attributes used with the* `<indexed>` *element.*

<Element> or Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
file	name of the index file	NA	String	A valid file name	Req.	refet _index.dat
`<entry>`	Creates environment for entry data.					
id	Entry ID number	NA	Integer	Any integer	Req.	0
`<const>`	A constant value option					
`<dss>`	A DSS option					
Details of the `<dss>` and `<const>` formats are described in chapter 6.						
NA = Not Applicable; Req. = Required.						

Table 3.8: *Attributes used with* `<gridio>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`file`	Name of the file	NA	String	A valid gridio file name	Req.	rain.dat
`dbintl`	Database interval used in the gridio file (minutes)	T	Integer	-1 for steady state or > 0	1440	1440.
`xorig`	X co-ordinate of the origin	L	Real	Any real	Req.	543329
`yorig`	Y co-ordinate of the origin	L	Real	Any real	Req.	286761
`mult`	Multiplier for the data.	NA	Real	Any real	1.0	0.0254
NA = Not Applicable; Req. = Required.						

Table 3.9: *Attributes used with* `<gms>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`file`	Name of the GMS file	NA	String	A valid GMS file name	Req.	enp.gms
`mult`	Multiplier for the data values, often used for conversion between metric and English units	NA	Real	Any real	1.0	0.3048
NA = Not Applicable; Req. = Required						

Table 3.10: *Attributes used with* `<netcdf>`.

Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
`file`	Name of the netcdf file	NA	String	A valid netcdf file name	Req.	coast.dat
`variable`	Name of the variable	NA	String	A valid variable name	No default	rain
dbintl	Database interval used in the netcdf file (minutes)	T	Integer	> 0	-1	1440
`mult`	Multiplier for the data values	NA	Real	Any real	1.0	0.3048
`units`	Units of the variable	NA	String	Any string	Imp.	meters
NA = Not Applicable; Req. = Required; Imp. = Implied						

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 Table 3.4, Table 3.5 and Table 3.6 are needed to describe all cells fully. User specified water movers such as physical structures 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.

3.2.1.1 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 50 meters, the transmissivity is of the confined type with a value of 0.04 m^2/s and `<svconverter>` uses a constant storage coefficient of 0.2. The `<svconverter>` is explained in section 3.8.

```
<geometry file="mesh3x3.2dm"> </geometry>
 <shead><gms file="hin3x3.dat"></gms></shead>
 <bottom>  <const value="0.0">   </const> </bottom>
 <surface> <const value="50.0"> </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 are to be presented in the `<dss>` format, the following example shows a segment of the input XML file.

```
<refet>
  <dss file="rainET.dss"
    pn = "/S8/Areal/REFET/01JAN1994/1DAY/Estimated/"
      mult="0.0254" dbintl="1440">
  </dss>
</refet>
```

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>
```

3.3 Two-Dimensional Grid Data <geometry>

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 geometry file for HSE can be created graphically using the GMS package. The resulting GMS file does not need much modification. 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 L8mesh.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="L8mesh.2dm" mult="0.3048">
  </geometry>
...
...
</hse>
```

A small sample mesh (mesh3x3.2dm) is shown in Figure 3.2 with cells and nodes numbered. A GMS file that describes the mesh is shown in Table 3.11. 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 3.11: *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.
```

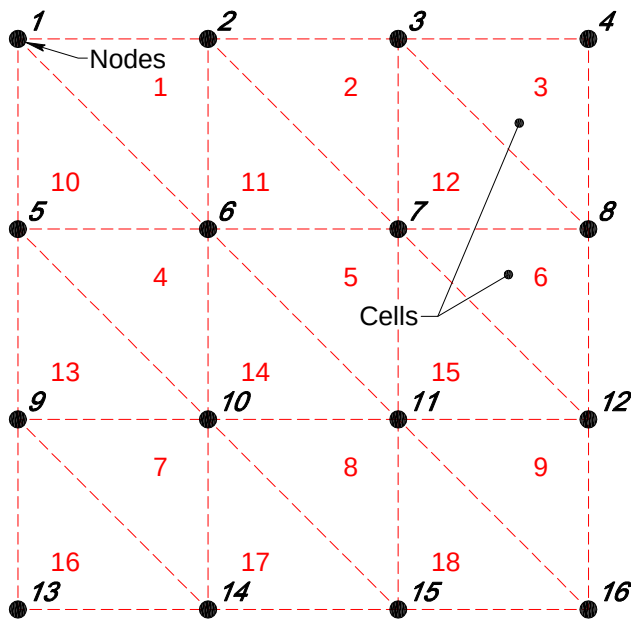


Figure 3.2: *Discretization of a square area into 18 cells with 16 nodes.(See Table 6.10).*

3.4 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.

3.4.1 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

$$Q = \frac{L}{n} d^{\frac{5}{3}} \sqrt{S} \tag{3.1}$$

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

It is well known that the appropriate value of Manning's n decreases with the depth of water. When a thick layer of vegetation is present Manning's n can be quite large for shallow flow but decrease significantly with flow depth. One way to improve the accuracy of Manning's equation is to represent n as

$$n = Ad^B \tag{3.2}$$

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

$$Q = Lad^\beta S^\alpha \tag{3.3}$$

where
Q = volume flow rate,
L = width of flow,
d = flow depth, and
a, α and β = 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. Flow volume is then computed as

$$Q = LC(d)S^{\alpha} \tag{3.4}$$

where, $C(d)$ is a lookup table that describes the variation of conveyance with depth. This is a useful approach for simulating the effects of microtopography found in the ridge and slough formations (Figure 3.3). In these areas the effect of the significant relief of the ground surface and the nature of the vegetation makes Manning's equation (Equation 3.1) or Kadlec's equation (Equation 3.3) (Kadlec and Knight, 1996) inappropriate. To construct the lookup table, the average land surface elevation of a cell is specified as well as two other elevations. The elevation $H = z_{lo}$ in Figure 3.3 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 $H = z_{lo}$, there is no overland flow. In the range $z_{lo} < H < z_{hi}$, Equation 3.4 is used.

3.4.1.1 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 3.12. 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.

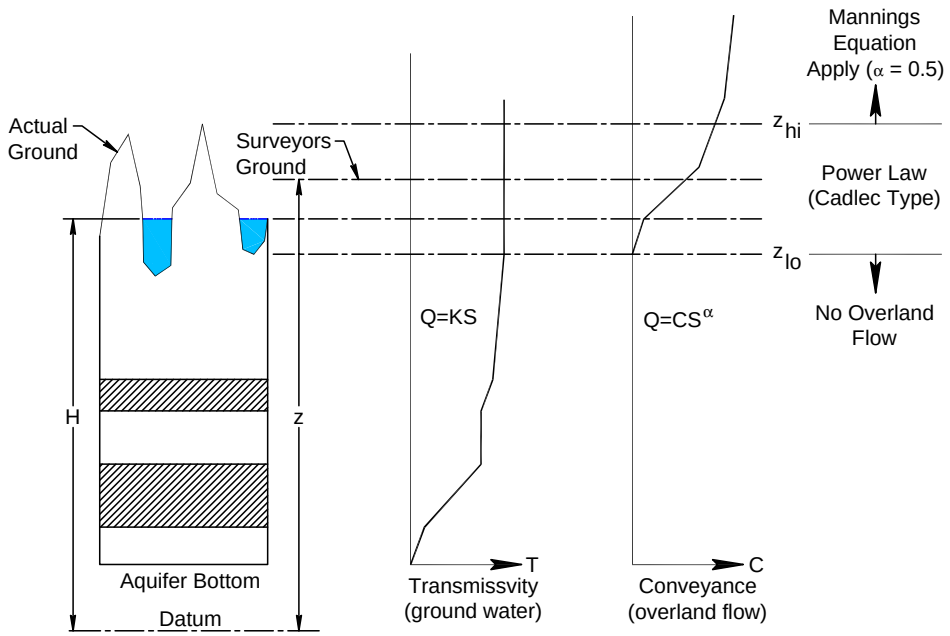


Figure 3.3: *Definition sketch for using lookup tables for transmissivity and conveyance.*

Table 3.12: *Elements and attributes under* `<conveyance>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
compute	Describes methods of computing average conveyance between cells. Details are described in Table 3.13.				mixed	
`<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 3.2	NA	Real	> 0.0	Req.	0.5
b	The exponent B in Equation 3.2	NA	Real	> 0.0	0	0.1
detent	Minimum value of d in Equation 3.2. The value of d is MAX(d,detent). (meters)	L	Real	> 0.0	Req.	0.1
`<cadlec>`	The conveyance is computed from Kadlec's equation 3.3.					
a	The coefficient a in Equation 3.3	NA	Real	> 0.0	Req.	0.5
b	The exponent α in Equation 3.3	NA	Real	> 0.0	Req.	0.1
beta	The exponent β in Equation 3.3	NA	Real	> 0.0	Req.	0.1
detent	Water depth below which there is no flow (meters)	L	Real	> 0.0	Req.	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 (meters)	L	Real	> 0.0	Req.	0.1
above	The depth above the "surveyor's" land surface above which the Manning equation applies (meters)	L	Real	> 0.0	Req.	0.1

Table 3.1 continued on next page

<Element> or Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
base	The parameter A in the equation for the Manning's roughness coefficient	NA	Real	> 0.0	Req.	0.1
expon	The parameter B in the equation for the Manning's roughness coefficient	NA	Real	> 0.0	Req.	0.1
`<convey>`	A lookup table of depth and conveyance for the range between (land surface - below) and (land surface + above)	NA	Real	> 0.0	No default	0.1
NA = Not Applicable; Req. = Required.						

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

$$n = 0.5d^{-0.77} \tag{3.5}$$

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

$$n = 0.4d^{-0.63} \tag{3.6}$$

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>
```

3.4.1.2 Conveyance Type `<cadlec>`

The method can be assigned to the entire model domain with the XML example below.

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

In this case, discharge is computed as

$$Q = L(2.0(10^6))d^3S^{0.6} for \quad d > 0.01 \tag{3.7}$$

$Q = 0$ if $d \leq 0.01$

where L = the length of the wall through which flow occurs, and d = depth of flow. The `<cadlec>` conveyance option is also used in benchmark 38.

3.4.1.3 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

$$Q = LC(d)S^{\alpha} \tag{3.8}$$

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 and in benchmark 36, 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>
```

3.4.1.4 Mixing Overland Flow Types

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 3.13. An example demonstrating this option is shown below.

Table 3.13: *Overland flow options for the "compute" attribute under* `<conveyance>`.

Value of "compute"	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.

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

3.4.2 Groundwater Flow `<transmissivity>`

Nine different methods are available to enter the transmissivity (or hydraulic conductivity) of confined and unconfined aquifers, as shown in Table 3.14. Seven methods are available directly under the `<transmissivity>` element including `<indexed>`, `<confined>`, `<layered>`, `<lookuptr>`, `<unconfined>`, `<unconfined_gms>`, and `<unconfined_gms_layer>`. Under the `<indexed>` and `<entry>` pathway, three more options are available including `<confined_gms>`, `<confined_gms_layer>`, and `<layered_gms_layer>`. Some of these options have been developed to facilitate the 3D groundwater flow analysis, which is described in Appendix C. Input options for all available methods of specifying transmissivity are defined in this section.

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

Table 3.14: *Allowable options under* `<transmissivity>`.

Element	Definition	Reference Table
`<confined>`	Specifies the transmissivity of a confined aquifer	Table 3.15
`<unconfined>`	Specifies the hydraulic conductivity of an unconfined aquifer	Table 3.16
`<confined_gms>`	Specifies the transmissivity of a confined aquifer using a gms style file. (Requires modification to the DTD for implementation)	Table 3.17
`<unconfined_gms>`	Specifies the hydraulic conductivity of an unconfined aquifer using a gms style file. (Requires modification to the DTD for implementation)	Table 3.18
`<confined_gms_layer>`	Specifies the transmissivity of a 3D confined aquifer using a gms style file	Table 3.19
`<unconfined_gms_layer>`	Specifies the hydraulic conductivity of a 3D unconfined aquifer using a gms style file	Table 3.20
`<layered>`	Specifies the hydraulic conductivity of a 3D unconfined aquifer	Table 3.21
`<layered_gms_layer>`	Specifies the transmissivity of a 3D confined aquifer using a gms style file	Table 3.22
`<lookuptr>`	Specifies the transmissivity as a function of head	Table 3.23

conductivity

$$Q = LkdS \tag{3.9}$$

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 (see Table 3.23) 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

$$Q = LK(H)S \tag{3.10}$$

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

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 $k = 0.04$ m/s.

```
<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>
```

The `<lookuptr>` example is also demonstrated in benchmark 36.

Table 3.15: *Definition of variables for* `<confined>` *under* `<transmissivity>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<confined>`	Specifies the transmissivity of a confined aquifer.					
trans	The transmissivity of a confined aquifer	L^2T^{-1}	Real	> 0.0	Req.	0.04
NA = Not Applicable; Req. = Required.						

Table 3.16: *Definition of variables for* `<unconfined>` *under* `<transmissivity>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<unconfined>`	Specifies the hydraulic conductivity of a unconfined aquifer.					
k	The hydraulic conductivity of the unconfined aquifer	L^2T^{-1}	Real	> 0.0	Req.	0.004
NA = Not Applicable; Req. = Required.						

Table 3.17: *Definition of variables for* `<confined_gms>` *under* `<transmissivity>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<confined_gms>`	Specifies the transmissivity of a confined aquifer using a gms style.					
file	The name of the gms file containing confined transmissivities	NA	String	Any valid GMS file name		tr.gms
mult	Multiplier converting transmissivity units	NA	Real	Any valid real	1.0	0.3048
NA = Not Applicable; Req. = Required.						

Table 3.18: *Definition of variables for* `<unconfined_gms>` *under* `<transmissivity>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<unconfined_gms>`	Specifies the hydraulic conductivity of an unconfined aquifer using a gms style file.					
file	The name of the gms file containing unconfined K's	NA	String	Any valid GMS file name		tr.gms
mult	Multiplier converting hydraulic conductivity units	NA	Real	Any valid real	1.0	0.3048
NA = Not Applicable; Req. = Required.						

Table 3.19: *Definition of variables for* `<confined_gms_layer>` *under* `<transmissivity>`. *For 3D flow analysis (see Appendix C)*

<Element> or Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
`<confined_gms_layer>`	Specifies the transmissivity of a confined aquifer using a gms style file.					
file	The name of the gms file containing confined transmissivities	NA	String	Any valid GMS file name	Req.	tr.gms
mult	Multiplier converting transmissivity units	NA	Real	Any valid real	1.0	0.3048
layer	Layer number	NA	Integer	Any valid layer number	Req.	2
NA = Not Applicable; Req. = Required.						

Table 3.20: *Definition of variables for* `<unconfined_gms_layer>` *under* `<transmissivity>`. *For 3D flow analysis (see Appendix C)*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<unconfined_gms_layer>`	Specifies the hydraulic conductivity of an unconfined aquifer using a gms style file.					
file	The name of the gms file containing unconfined K's	NA	String	Any valid GMS file name	Req.	tr.gms
mult	Multiplier converting hydraulic conductivity units	NA	Real	Any valid real	1.0	0.3048
NA = Not Applicable; Req. = Required.						

Table 3.21: *Definition of variables for* `<layered>` *under* `<transmissivity>`. *For 3D flow analysis (see Appendix C)*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<layered>`	Specifies the hydraulic conductivity of a layered aquifer.					
cond	The hydraulic conductivity of a layer	LT^{-1}	Real	> 0.0	Req.	0.004

Table 3.22: *Definition of variables for* `<layered_gms_layer>` *under* `<transmissivity>`. *For 3D flow analysis (see Appendix C)*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<layered_gms_layer>`	Specifies the transmissivity of a layered aquifer using a gms style file.					
file	The name of the gms file containing transmissivities per layer	NA	String	Any valid GMS file name	Req.	tr.gms
mult	Multiplier converting transmissivities units	NA	Real	Any valid real	1.0	0.3048
layer	Layer number	NA	Integer	Any valid layer number	Req.	2

Table 3.23: *Definition of variables for* `<lookuptr>` *under* `<transmissivity>`.

`<Element>` or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<lookuptr>`	Denotes a lookup table of transmissivity as a function of head.					
below	The elevation of the lower end of the lookup table	L	Real	> 0.0	0	490.0
above	The elevation of the upper end of the lookup table	L	Real	> 0.0	0	520.0
flstat	*Not available in current version*	NA	String	y or n	n	y
`<transm>`	A 1D lookup table of depth and transmissivity					
NA = Not Applicable; Req. = Required						

3.5 Water Movers

3.5.1 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. An example of 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.

3.5.2 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.

3.5.3 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 3.24 lists some available concepts for water movers.

Table 3.24: *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

3.5.3.1 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

$$Q_{12} = CL(H_1 - z)^b \tag{3.11}$$

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 3.25.

Table 3.25: *Attribute definitions for* `<standardweir>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	200000-300000	Req.	256987
`id2`	ID of the downstream water body	NA	Integer	200000-300000	Req.	268546
`wmID`	ID of the water mover	NA	Integer	500000-600000	-1	568546
`fcoeff`	Weir coefficient for flow in the forward direction NOTE: `<bcoeff> is used for both forward and reverse flow`	LT^{-1}	Real	> 0.0	Req.	0.53
`bcoeff`	Weir coefficient for flow in the reverse direction NOTE: `<bcoeff> is used for both forward and reverse flow`	LT^{-1}	Real	> 0.0	Req.	0.53
`length`	Crest length (L)	L	Real	> 0.0	Req.	85.6
`crestelev`	Crest elevation (z) of the weir	L	Real	> 0.0	Req.	26.3
`power`	The exponent (b) in the weir equation	NA	Real	> 0.0	Req.	0.6
`label`	An optional label for the weir	NA	String	Any string	wm+wmID	Miami weir
NA = Not Applicable; Req = Required.						

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>
```

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

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

$$Q_{12} = CL(H_1 - z)^a (H_1 - H_2)^b \tag{3.12}$$

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.

Because of the flexibility of the `<genweir>` water mover, the user may decide to use this equation for certain types of weirs or bridges. For example, flow through a V-notch weir is calculated as

$$Q = \frac{8}{15} K \sqrt{2g} \tan(\theta/2)(H - Z)^{2.5} \tag{3.13}$$

and may be simulated by assigning appropriate values of the variables. Complete specification of the attributes of a general power law based water mover are defined in Table 3.26.

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 3.26: *Attribute definitions for* `<genweir>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	200000-300000	Req.	256987
`id2`	ID of the upstream water body	NA	Integer	200000-300000	Req.	268546
`wmID`	ID of the water mover	NA	Integer	500000-600000	-1	568546
`fcoeff`	Weir coefficient used for forward (1→2) flow.	NA	Real	> 0.0	Req.	0.38
`bcoeff`	Weir coefficient used for reverse (2→1) flow	NA	Real	> 0.0	Req.	0.46
`crestelev`	Crest elevation of the weir	L	Real	> 0.0	Req.	23.8
`crestlen`	Weir crest length	L	Real	> 0.0	Req.	58.3
`dpower`	Exponent a for upstream depth over the weir	NA	Real	≥ 0.0	Req.	1.4
`spower`	Exponent b for water level difference	NA	Real	≥ 0.0	Req.	0.56
`label`	Label to describe the weir.	NA	String	Any String	wm+wmID	Farmer Jones irrigation weir
NA = Not Applicable; Req. = Required.						

3.5.3.3 General Structure Water Mover `<genstruc>`

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

$$Q_{12} = C(H_1 - H_2)^a \tag{3.14}$$

where C = user specified coefficient,
H_1 and H_2= water levels in water bodies 1 and 2,
a = user specified coefficient, and
The minimum flow is a user specified value (design). The attributes of a general structure water mover are defined in Table 3.27.

The following example describes a `<genstruc>` water mover connecting water body 5 to water body 11.

```
<watermovers>
....
   <genstruc label="Structure 1" id1 = "5" id2 = "11" wmID = "13"
      design = "65.0" dischar = "2.8" a = "1.3" >
   </genstruc>
....
</watermovers>
```

Table 3.27: *Attribute definitions for the* `<genstruc>` *water mover.*

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`label`	Description of the structure	NA	String	Any String	wm+wmID	Structure 3
`id1`	ID of the upstream water body	NA	Integer	200000-300000	Req.	23657
`id2`	ID of the upstream water body	NA	Integer	200000-300000	Req.	27824
`wmID`	ID of the water mover	NA	Integer	500000-600000	-1	57896
`design`	The user specified minimum flow.	L^3T^{-1}	Real	> 0.0	Req.	2.3
`dischar`	Exponent a in the discharge equation	NA	Real	> 0.0	Req.	1.7
NA = Not Applicable; Req. = Required.						

3.5.3.4 Rated Structure Water Mover `<ratedStruc>`

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

$$Q = Coeff \left[\frac{H_1 - Close}{Open - Close} \right]^b [H_1 - H_2]^a \tag{3.15}$$

where Coeff = user specified coefficient,
H_1 and H_2= water levels in water bodies 1 and 2,
b = user specified exponent a = user specified exponent
Open and Close are user specified stages.
If $H_1 <$ Close , $\frac{H_1 - Close}{Open - Close}$ is replaced by 1.0. The minimum flow is 0. Complete specifications of the attributes of a rated structure water mover are defined in Table 3.27.

The following example describes a `<ratedStruc>` water mover connecting water body 7 to water body 14.

```
<watermovers>
....
   <ratedStruc label="Structure 14" id1 = "7" id2 = "4" wmID = "16"
      design = "58.0" dischar = "1.35" open="13.8" close="8.6" a = "1.3" b="0.9 >
   </ratedStruc>
....
</watermovers>
```

Table 3.28: *Attribute definitions for the* `<ratedStruc>` *water mover.*

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`label`	Description of the structure	NA	String	Any String	wm+wmID	Structure 3
`id1`	ID of the upstream water body	NA	Integer	200000-300000	Req.	23657
`id2`	ID of the upstream water body	NA	Integer	200000-300000	Req.	27824
`wmID`	ID of the water mover	NA	Integer	500000-600000	-1	57896
`design`	The user specified minimum flow.	L^3T^{-1}	Real	> 0.0	Req.	2.3
`dischar`	User specified coefficient.	NA	Real	> 0.0	Req.	1.7
`open`	User specified elevation	L	Real	> 0.0	Req.	13.6
`close`	User specified elevation	L	Real	> 0.0	Req.	8.8
`a`	User specified exponent	NA	Real	> 0.0	Req.	0.85
`b`	User specified exponent	NA	Real	> 0.0	Req.	1.54
NA = Not Applicable; Req. = Required.						

3.5.3.5 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 as could be done by pumping. 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 3.29

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>
```

Table 3.29: *Element and attribute definitions for* `<doublet>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	200000-300000	Req.	256987
`id2`	ID of the upstream water body	NA	Integer	200000-300000	Req.	268546
`wmID`	ID of the water mover	NA	Integer	500000-600000	Req.	568546
`label`	An optional label for the doublet	NA	String	Any string	wm+wmID	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.					
Details on specifying input data in `<const>`, `<rc>`, and `<dss>` formats are provided in chapter 6						
NA = Not Applicable; Req. = Required.						

3.5.3.6 Controllable User-Defined Flow `<setflow>`

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

Complete specifications of the attributes of a `<setflow>` water mover are defined in Table 3.30 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 3.30: *Attribute definitions for* `<setflow>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<id1>`	ID of the upstream water body	NA	Integer	200000-300000	Req.	256987
`<id2>`	ID of the upstream water body	NA	Integer	200000-300000	Req.	268546
`<wmID>`	ID of the water mover	NA	Integer	500000-600000	Req.	568546
`label`	An optional label for the water mover	NA	String	Any string	wm+wmID	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.					
Details on specifying input data in `<const>`, `<rc>`, and `<dss>` formats are provided in chapter 6						
NA = Not Applicable; Req. = Required.						

3.5.3.7 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.

3.5.3.8 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 3.31

Table 3.31: *Attribute definitions for the* `<single_control>` *Water Mover.*

Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
id1	ID of the upstream water body	NA	Integer	200000-300000	Req.	256987
id2	ID of the upstream water body	NA	Integer	200000-300000	Req.	268546
wmID	Water Mover ID	NA	Integer	200000-300000	-1	268546
control	ID of the control water body.	NA	Integer	200000-300000	Req.	568546
cutoff	Head in the control water body below which no flow occurs.	L	Real	≥ 0.0	Req.	15.36
gravflow	Specifies whether only gravity flow can occur (flow can occur only from upstream to downstream).	NA	String	"yes" or "no"	Req.	"yes"
revflow	"yes" allows reverse flow	NA	String	"yes" or "no"	Req.	"yes"
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	"Irrigation Pump"
NA = Not Applicable; Req. = Required.						

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>
```

3.5.3.9 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 rows and columns are the corresponding discharges. The discharges for intermediate heads is determined by 4-point linear interpolation. The water bodies need not be adjacent although they often will be.

The attributes of a <dual_control> water mover are explained in Table 3.32

Table 3.32: *Attribute definitions for the* `<dual_control>` *Water Mover.*

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
id1	ID of the upstream water body	NA	Integer	200000-300000	Req.	256987
id2	ID of the upstream water body	NA	Integer	200000-300000	Req.	268546
wmID	Water Mover ID	NA	Integer	200000-300000	-1	268546
control	ID of the control water body (Not Implemented)	NA	Integer	200000-300000	Req.	568546
cutoff	Head in the upstream water body below which no flow occurs.	L	Real	≥ 0.0	Req.	15.36
gravflow	Specifies whether only gravity flow can occur (flow can occur only from upstream to downstream).	NA	String	"yes" or "no"	Req.	"yes"
revflow	"yes" allows reverse flow	NA	String	"yes" or "no"	Req.	"yes"
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	"Weir 1A"
NA = Not Applicable; Req. = Required.						

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>
```

3.5.3.10 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 3.33

Table 3.33: *Attribute definitions for the* `<delta_control>` *Water Mover.*

Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
id1	ID of the upstream water body	NA	Integer	200000-300000	Req.	256987
id2	ID of the upstream water body	NA	Integer	200000-300000	Req.	268546
wmID	Water Mover ID	NA	Integer	200000-300000	-1	268546
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	"Flood Control Pump"
NA = Not Applicable; Req = Required.						

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.

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

3.5.3.11 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.

3.5.3.12 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

$$Q_{12} = K(H_1 - H_2) \tag{3.16}$$

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 3.34.

Table 3.34: *Attribute definitions for a shunt Water Mover.*

Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
`id1`	ID of the upstream water body	NA	Integer	200000-300000	Req.	256987
`id2`	ID of the downstream water body	NA	Integer	200000-300000	Req.	268546
wmID	Water Mover ID	NA	Integer	200000-300000	-1	24956
`sconst`	Conveyance of the shunt (K in the flow equation)	L^2T^{-1}	Real	$\geq$0.0	Req.	4.7
`bottom`	Head in water body 1 below which no flow occurs.	NA	Real	$\geq$ 0.0	Req.	11.64
NA = Not Applicable; Req = Required.						

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>
```

3.5.4 New And Borrowed Physical Structure Types

A number of water movers have been included in the code to provide flexibility for modelers although the approaches and algorithms for some have not been extensively tested and verified by OoM staff. Only some of these water movers will be used in the SFWMD implementation of the RSM to South Florida (SFRSM), although they are supported for other implementations. Water movers that will not be used in the SFRSM and have not been thoroughly tested by the OoM will be identified in the description of each and the modeler is advised to use them only with a clear understanding of the underlying algorithms. It is the modelers responsibility to ensure that an algorithm is suitable for use in a particular application.

Several 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 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 3.35.

3.5.5 Culvert Water Mover `<culvert>`

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

The culvert water mover was borrowed from the National Weather Service. 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 3.4 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 3.36 shows the attributes of a culvert.

Table 3.35: *List of physical water movers.*

Element	Description
`<culvert>`	Culvert water mover borrowed from the National Weather Service (NWS).
`<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.

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

$$Q = CA\sqrt{\frac{2g(Y1 - Y2)}{1 + \frac{29.1C^2n^2L}{R^{4/3}}}} \tag{3.17}$$

   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 $\geq$ 1.3 culvert diameter (or height for a rectangular culvert) and length <= 20∗diameter, the culvert is hydraulically short and discharge is inlet controlled.

$$Q = 0.7CA\sqrt{2g(H1 - 0.6D)} \tag{3.18}$$

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

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

$$Q = 0.7CA\sqrt{\frac{2g(Y1 - max(Y2, CH2 + 0.5D))}{1 + \frac{29.1C^2n^2L}{R^{4/3}}}} \tag{3.19}$$

   where CH2 = culvert invert at the downstream end.

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

$$Q = CA\sqrt{2g * (Y1 - CH2)} \tag{3.20}$$

   where

$$A = W\frac{Y1 - CH2}{1.25 + \frac{0.5}{C^2}} \tag{3.21}$$

   for a rectangular culvert and

$$A = [0.785Y - 0.045\sin(2\pi Y)]D^2 \tag{3.22}$$

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

$$Y = \frac{DC}{D} \tag{3.23}$$

$$DC = \frac{Y1 - CH2}{1.25 + \frac{0.425}{C^2}} \tag{3.24}$$

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.

$$Q = CA\sqrt{2g * (Y1 - CH1)} \tag{3.25}$$

where

$$A = W\frac{Y1 - CH1}{1.05 + \frac{0.5}{C^2}} \tag{3.26}$$

for a rectangular culvert and

$$A = [0.785Y - 0.045\sin(2\pi Y)]D^2 \tag{3.27}$$

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

$$Y = \frac{DC}{D} \tag{3.28}$$

$$DC = \frac{Y1 - CH1}{1.05 + \frac{0.425}{C^2}} \tag{3.29}$$

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.

$$Q = CA\sqrt{2g * (Y1 - Y2)} \tag{3.30}$$

where

$$A = WH2 \tag{3.31}$$

for a rectangular culvert

$$A = [0.785Y - 0.045\sin(2\pi Y)]D^2 \tag{3.32}$$

for a circular culvert

where W = width of a rectangular culvert and

$$Y = \frac{DC}{D}, \tag{3.33}$$

$$DC = Y2 - Y1 \tag{3.34}$$

and DC = D if DC > D.

Table 3.36: *Attributes of the* `<culvert>` *water mover.*

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
`wmid`	ID of the water mover being created	NA	Integer	200000-300000	-1	256987
`width`	Width of a box culvert. Set=0 for a circular culvert	L	Real	≥ 0.0	Req.	2.5
`height`	Height of a box culvert or diameter of a circular culvert	L	Real	≥ 0.0	Req.	4.25
`length`	Length of the culvert barrel	L	Real	≥ 0.0	Req.	110.6
`mann`	Manning's roughness coefficient	$TL^{-1/3}$	Real	$0.0 - 0.2$	Req.	0.024
`coeff`	Discharge coefficient for the culvert	NA	Real	$0.39 - 0.98$	Req.	0.42
`hw_inv`	Elevation of the upstream culvert invert	L	Real	≥ 0.0	Req.	13.86
`tw_inv`	Elevation of the downstream culvert invert	L	Real	≥ 0.0	Req.	10.48
`rev`	Specifies whether reverse flow is allowed. N prevents reverse flow. Y is default.	NA	String	N or Y	Req.	Y
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	Route 75 culvert
NA = Not Applicable; Req. = Required.						

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>
```

3.5.5.1 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.

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 3.4 shows a definition sketch for pipe flow. Table 3.37 shows the XML attributes used to define the pipe water mover.

Figure 3.4: *Definition sketch of a pipe.*

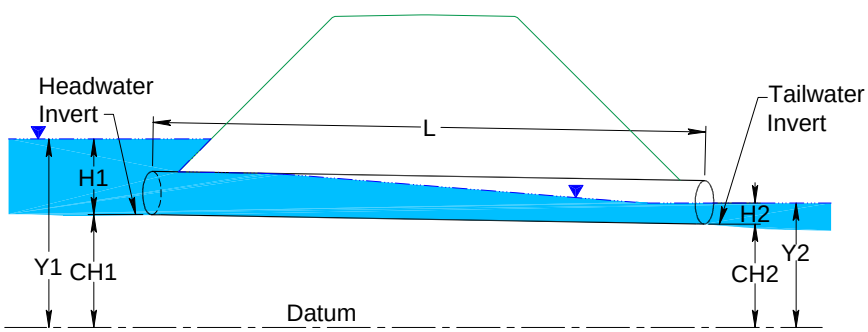


Table 3.37: *Attributes used to define a* `<pipe>` *water mover.*

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
`wmid`	ID of the water mover being created	NA	Integer	200000-300000	-1	256987
`diameter`	Diameter of a circular pipe	L	Real	≥ 0.0	Req.	4.25
`length`	Length of the pipe	L	Real	≥ 0.0	Req.	110.6
`mann`	Manning's roughness coefficient	$TL^{-1/3}$	Real	$0.0 - 0.2$	Req.	0.024
`hw_inv`	Elevation of the upstream culvert invert	L	Real	≥ 0.0	Req.	13.86
`tw_inv`	Elevation of the downstream culvert invert	L	Real	≥ 0.0	Req.	10.48
`rev`	Specifies whether reverse flow is allowed. N prevents reverse flow.	NA	String	N or Y	Req.	Y
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	Highway 66 culvert
NA = Not Applicable; Req. = Required.						

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.

3.5.5.2 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 3.5 is borrowed from the MBR model with discharge equation

$$Q = C_d L(H1 - z)^{1.5} \tag{3.35}$$

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

If $TW \geq 0.99H$ then $Q = 0$
If $0.95H \leq TW < 0.99H$

$$CS = 0.965 - \exp(\frac{100(TW/H - 1)}{4} - 0.43) \tag{3.36}$$

If $0.76H \leq TW < 0.95H$

$$CS = 1.0 - \exp(\frac{100(TW/H - 1)}{4} - 0.2) \tag{3.37}$$

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

where, 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 3.38

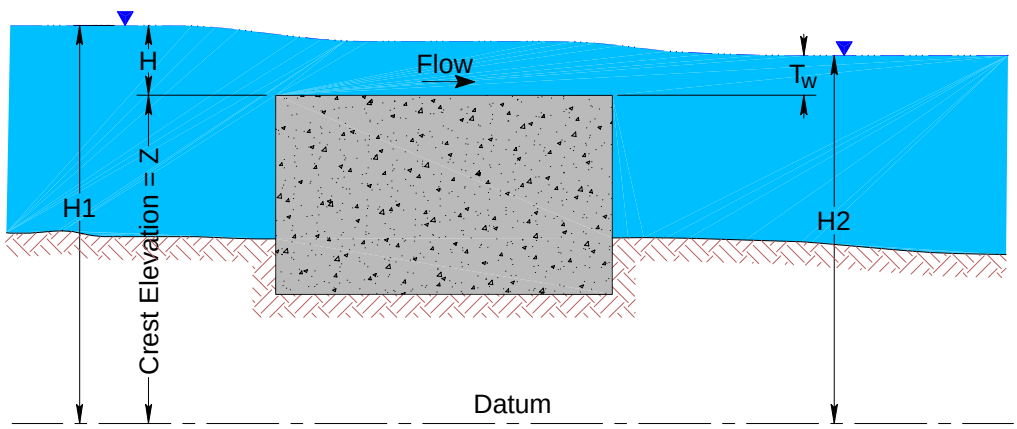


Figure 3.5: *Definition sketch of broad crested weir.*

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>
```

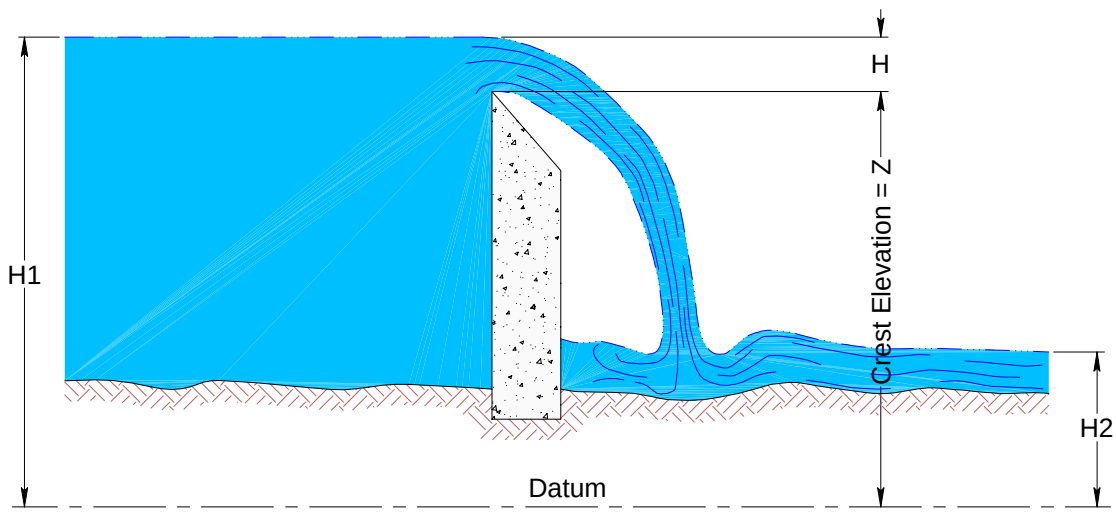
Table 3.38: *Attributes of a broad crested weir,* `<mbrbroadweir>`.

Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
id1	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
id2	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
wmid	ID of the water mover being created	NA	Integer	200000-300000	-1	256987
crestelev	Elevation of the weir crest	L	Real	≥ 0.0	Req.	14.78
length	Length of the weir across the channel	L	Real	≥ 0.0	Req.	116.8
coeff	Discharge coefficient	$L^{1/2}T^{-1}$	Real	≥ 0.0	Req.	0.48
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	Domino Sugar weir
NA = Not Applicable; Req. = Required.						

3.5.5.3 MBR Sharp Weir `<mbrsharpweir>`

The sharp crested weir is also borrowed from the MBR model. Figure 3.6 shows a definition sketch of a sharp crested weir.

Figure 3.6: *Definition sketch of a sharp crested weir.*



Discharge is computed as

$$Q = 3.13(CS)L(H - z)^{1.5} \tag{3.38}$$

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,

$$CS = (1 - (\frac{TW}{H})^{1.5})^{0.385} \tag{3.39}$$

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>
```

Table 3.39: *Attributes of a sharp crested weir,* `<mbrsharpweir>`.

Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
id1	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
id2	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
wmID	ID of the water mover being created	NA	Integer	200000-300000	-1	256987
crestelev	Elevation of the weir crest	L	Real	≥ 0.0	Req.	12.93
length	Length of the weir across the channel	L	Real	≥ 0.0	Req.	63.7
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	Jones Farm weir
NA = Not Applicable; Req. = Required.						

3.5.5.4 MBR Drop Weir `<mbrdropweir>`

The drop weir (Figure 3.7)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

$$Q = CLH^{1.5} \tag{3.40}$$

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

$$R = \frac{H}{D} \tag{3.41}$$

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

$$C = 4.01 + 0.72R - 6.12R^2 + 4.37R^3 - 0.93R^4 \tag{3.42}$$

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.

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>
```

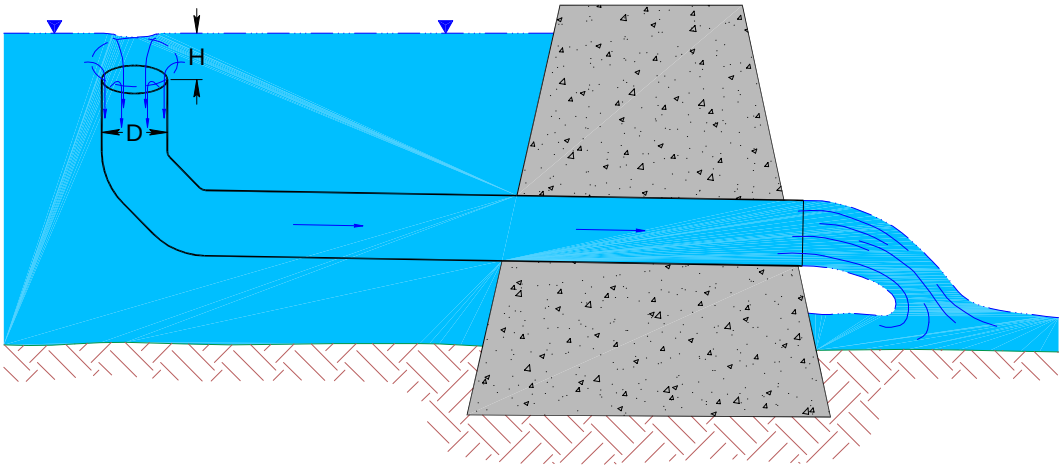


Figure 3.7: *Definition sketch of a drop weir.*

Table 3.40: *Attributes of a* `<mbrdropweir>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
id1	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
id2	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
wmid	ID of the water mover being created	NA	Integer	200000-300000	-1	256987
crestelev	Elevation of the weir crest	L	Real	≥ 0.0	Req.	23.6
length	Length of the weir (Circumference of the pipe)	L	Real	≥ 0.0	Req.	76.3
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	Detent Pond Outlet
NA = Not Applicable; Req. = Required.						

3.5.5.5 NWS Uncontrolled Spill `<spill>`

This is a spillway routine borrowed from the NWS FLDWAV model (Figure 3.8). Only downstream flow is allowed. Discharge is computed as

$$Q = CS * CTW * L * \sqrt{2g} * H^{1.5} \tag{3.43}$$

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

$$CTW = 27.8(\frac{TW}{H} - 0.67)^3 \tag{3.44}$$

Attributes of the uncontrolled spillway are listed in Table 3.41

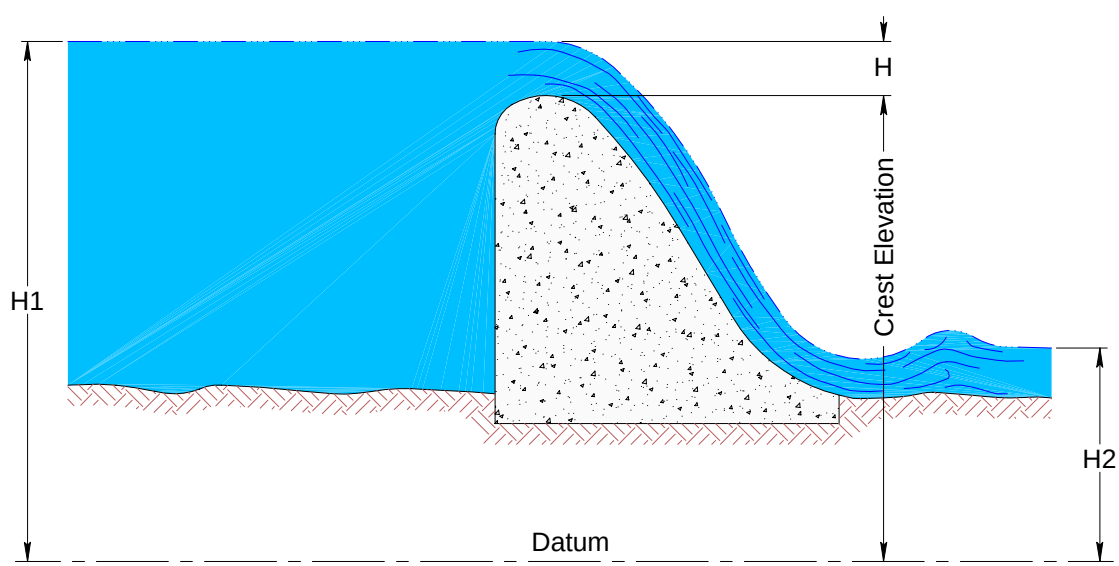


Figure 3.8: *Definition sketch of an uncontrolled spillway.*

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>
```

Table 3.41: *Attributes of an uncontrolled spillway* `<spill>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
wmid	ID of the water mover being created	NA	Integer	200000-300000	Req.	256987
`c15`	Discharge coefficient	NA	Real	0.35 to 0.45	Req.	0.42
`crest`	Elevation of the weir crest	L	Real	≥ 0.0	Req.	16.7
`width`	Length of the weir (Circumference of the pipe)	L	Real	≥ 0.0	Req.	18.7
NA = Not Applicable; Req. = Required.						

3.5.5.6 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.

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 3.9). 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

$$Q = C(CS)WH_u^{\frac{3}{2}} \tag{3.45}$$

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

$$Q = C * CS * W * H_G\sqrt{2gH_u} \tag{3.46}$$

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

The tailwater correction for both weir flow and orifice flow is

$$CS = 1.0 - 27.83\left[\frac{H_u}{H_d} - 0.67\right]^3 \tag{3.47}$$

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 3.42.

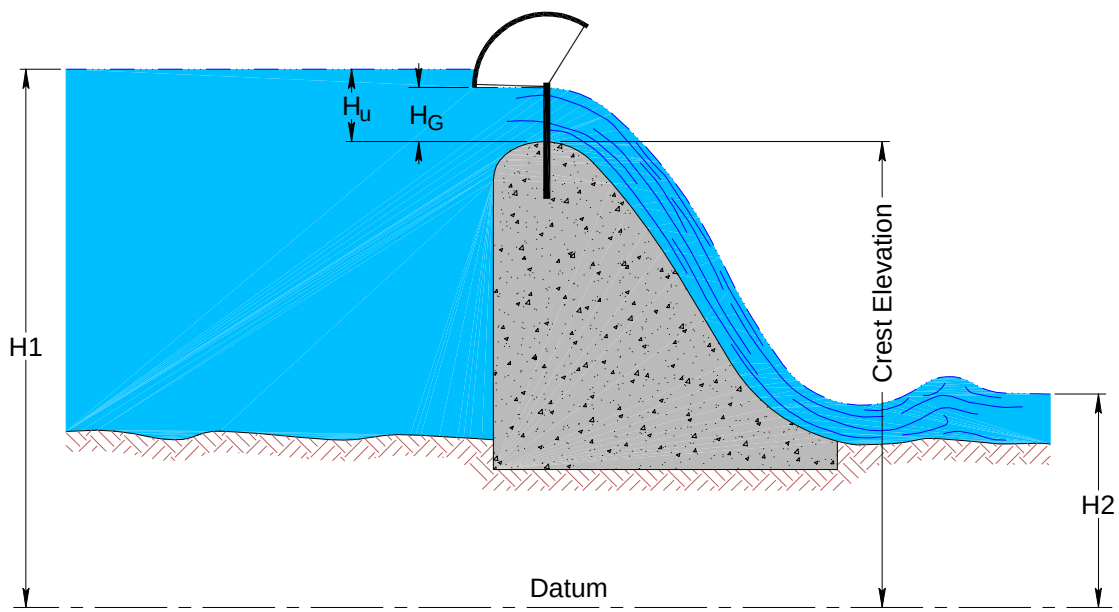


Figure 3.9: *Definition sketch of a gated weir.*

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 3.42: *Attributes of* `<gateweir>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
`wmid`	ID of the water mover being created	NA	Integer	200000-300000	Req.	256987
`co5`	Discharge coefficient for weir flow	NA	Real	0.35 -0.45	Req.	0.42
`c15`	Discharge coefficient for orifice flow	NA	Real	0.65 - 0.75	Req.	0.67
`control`	Specifies what is controlled by controller	NA	String	gate or flow	flow	flow
`crest`	Elevation of the weir crest.	L	Real	≥ 0.0	Req.	13.2
`width`	Length of the weir.	L	Real	≥ 0.0	Req.	32.6
`gopen`	Gate opening (height of the opening in orifice flow)	L	Real	≥ 0.0	1.0m	5.3
`label`	Label for the weir	NA	String	Any string	wm+wmID	Weir 34
NA = Not Applicable; Req = Required.						

3.5.6 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 3.10.

3.5.6.1 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)

$$Q = 0.6A\sqrt{2g(H_u - H_{centroid})} \tag{3.48}$$

for $H_d \leq H_{centroid}$ and

$$Q = 0.6A\sqrt{2g(H_u - H_d)} \tag{3.49}$$

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

$$Q = 2.5 * CS * \tan(\theta/2)H_u^{2.5} \tag{3.50}$$

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

$$CS = \left[1 - \left(\frac{H_d - H_{inv}}{H_u - H_{inv}}\right)^{2.5}\right]^{0.385} \tag{3.51}$$

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

Attributes of `<vnotchbleeder>` are listed in Table 3.43.

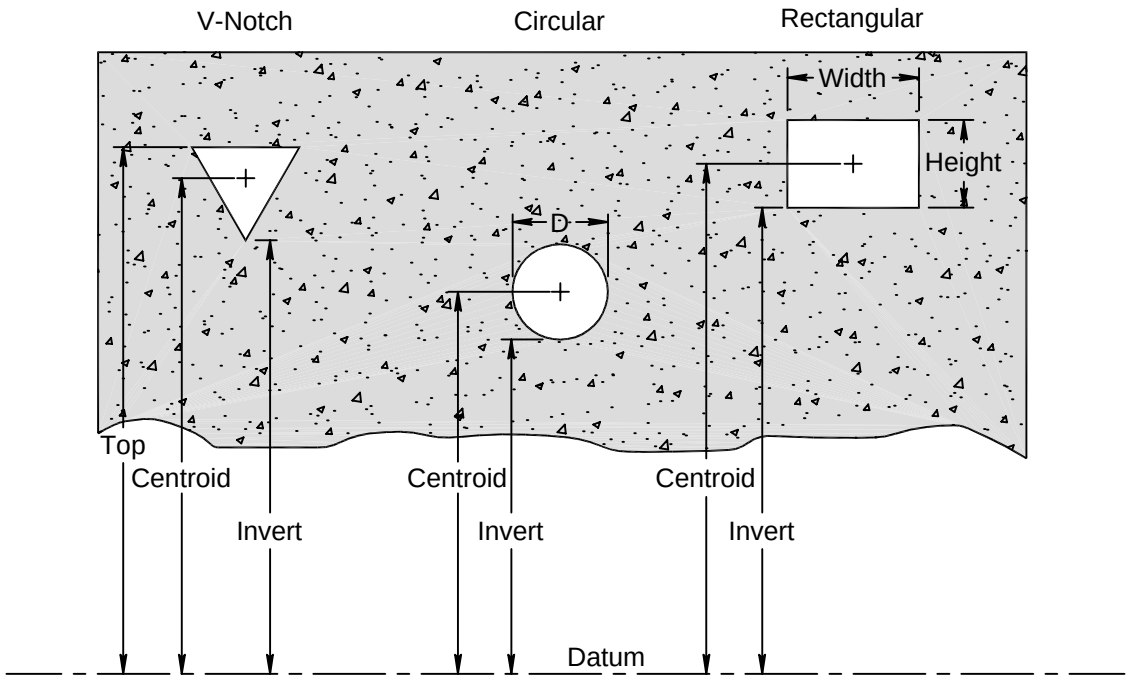


Figure 3.10: *Definition sketch of bleeders.*

```
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>
```

Table 3.43: *Attributes of* `<vnotchbleeder>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
`wmid`	ID of the water mover being created	NA	Integer	200000-300000	-1	256987
`invert`	Elevation of the lowest point of the bleeder opening	L	Real	≥ 0.0	Req.	13.24
`top`	Elevation of the highest point of the bleeder opening	L	Real	≥ 0.0	Req.	15.23
`angle`	Angle of the V at the invert in degrees	NA	Real	$0 - 180$	Req.	60.0
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	vbleeder 1
NA = Not Applicable; Req. = Required.						

3.5.6.2 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

$$Q = 0.6A\sqrt{2g(H_u - H_{centroid})} \tag{3.52}$$

for $H_d \leq H_{centroid}$ and

$$Q = 0.6A\sqrt{2g(H_u - H_d)} \tag{3.53}$$

for $H_d > H_{centroid}$

Weir flow is

$$Q = 0.6A\sqrt{\frac{2g(H_u - H_{invert})}{2}} \tag{3.54}$$

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

$$Q = 0.6A\sqrt{2g(H_u - H_d)} \tag{3.55}$$

The attributes of `<circularbleeder>` are defined in Table 3.44.

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>
```

Table 3.44: *Attribute definitions for* `<circularbleeder>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
`wmid`	ID of the water mover being created	NA	lInteger	200000-300000	-1	256987
`centroid`	Elevation of the centroid of the bleeder opening	L	Real	≥ 0.0	Req.	11.56
`diameter`	Diameter of the bleeder	L	Real	≥ 0.0	Req.	3.24
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	bleeder 2
NA = Not Applicable; Req. = Required.						

3.5.6.3 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 3.45. Discharge is computed as

$$Q = 0.6A\sqrt{2g(H_u - H_{centroid})} \tag{3.56}$$

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

$$Q = 0.6A\sqrt{2g(H_u - H_d)} \tag{3.57}$$

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

$$Q = 3.13L(H_u - H_{invert})^{1.5} \tag{3.58}$$

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

$$CS = \left[1 - \left(\frac{H_d - H_{invert}}{H_u - H_{invert}}\right)^{1.5}\right]^{0.385} \tag{3.59}$$

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>
```

Table 3.45: *Attribute definitions for* `<rectbleeder>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
`wmid`	ID of the water mover being created	NA	Integer	200000-300000	-1	256987
`centroid`	Elevation of the centroid of the bleeder opening.	L	Real	≥ 0.0	Req.	13.24
`height`	Height of the bleeder opening	L	Real	≥ 0.0	Req.	2.58
`width`	Width of the bleeder opening	L	Real	≥ 0.0	Req.	2.65
label	Optional label for the water mover.	NA	String	Any String	wm+wmID	rectbleeder 41
NA = Not Applicable; Req = Required.						

3.5.7 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 3.11 and the attributes of `<yarnell>` are presented in Table 3.46

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

$$H_1 - H_2 = 2K\left(K + 10\omega - 0.6\right)\left(\alpha + 15\alpha^4\right)\frac{V_2^2}{2g} \tag{3.60}$$

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.

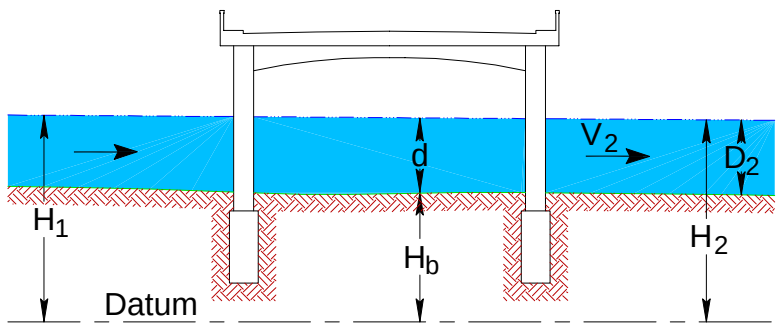
A lookup table is used to define the cross section as a function of head.

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

$$Q = A\left[2g\left\langle\sqrt{(K-0.6)^2 + \frac{20\,(H_1 - H_2)}{D_2 K\,(\alpha + 15\alpha^4)}} - (K-0.6)\,\frac{D_2}{20}\right\rangle\right]^{\frac{1}{2}}[H_1 - H_2] \tag{3.61}$$



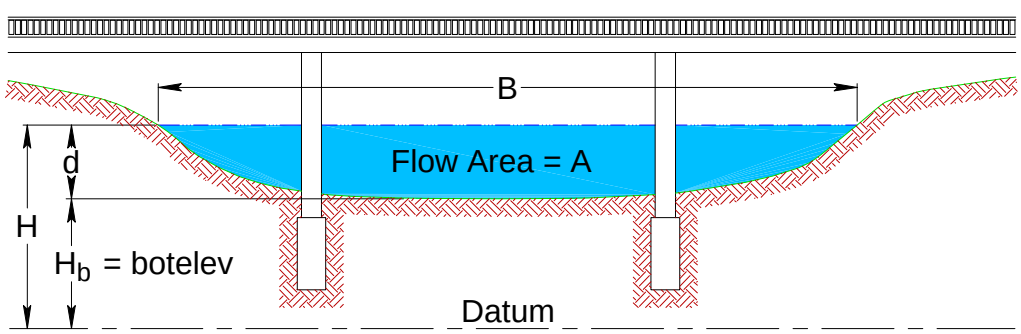


Figure 3.11: *Definition of cross sections used with the bridge routine.*

Table 3.46: *Attribute definitions for* `<yarnell>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
`wmid`	ID of the water mover being created	NA	Integer	200000-300000	Req.	256987
`pshape`	Yarnell's pier shape coefficient	NA	Real	0.5 - 3.0	Req.	1.05
`pwidth`	Total width of al the piers	L	Real	$\geq$ 0.0	Req.	8.62
`botelev`	Bottom elevation below which there is no flow	L	Real	$\geq$ 0.0	Req.	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	NA	Real	$\geq$ 0.0	Req.	see example input
NA = Not Applicable; Req. = Required.						

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>
```

3.5.8 Hydropower<hydropower>

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

$$Q = \frac{Power * 1000}{\rho g * efficiency} \tag{3.62}$$

where the power is determined as the demand, or the capacity, whichever is higher. The attributes of a `<hydropower>` object are listed in Table 3.47

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 3.47: *Attributes of* `<hydropower>`.

Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`id1`	ID of the upstream water body	NA	Integer	100000-200000	Req.	153675
`id2`	ID of the downstream water body	NA	Integer	100000-200000	Req.	148322
`wmID`	ID of the water mover	NA	Integer	100000-200000	Req.	25456
`label`	Label for the water mover object	NA	String	Any string	wm+wmID	Hoover Dam
`capacity`	Plant Capacity, MW	ML^2T^{-3}	Real	≥ 0.0	Req.	1.5
`lowelev`	Lowest headwater elevation at which plant will operate	L	Real	≥ 0.0	Req.	8.6
`highhead`	Design head drop. Used to check the efficient operating range	L	Real	≥ 0.0	Req.	17.6
`lowhead`	Minimum head drop for plant operation	L	Real	≥ 0.0	Req.	3.25
`efficiency`	Overall plant efficiency as a fraction	NA	Real	≥ 0.0	Req.	0.9
`mult`	Multiplier for demand	NA	Real	≥ 0.0	1.0	0.78
The power demand is specified as `<const>`, as `<rc>` or in a file in `<dss>` format as in the example in this section. Details are provided in chapter 6						
NA = Not Applicable; Req = Required.						

3.6 Canal Network - The <network> Element

HSE is capable of simulating diffusion flow in a canal network, Figure 3.12. The network is defined by the input of data that describe

1. The geometry that defines the location and cross sectional shapes of the canal segments.
2. Flow in the canal and interactions of the canal with the mesh. These include Manning's n and coefficients for overland flow between the canal and the mesh, seepage into and out of the bottom of the canal, and seepage through levees adjacent to canals.
3. The initial conditions (water levels) in the canal segments.

Construction of the XML data set to describe and set up the canal network is described in the following sections.

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, 2000b). 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.

3.6.1 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 3.48. A canal simulation also requires a boundary condition file that is described in section 4.2.

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 groups of segments using an index file. A number of sets of parameters are specified in the XML input with each

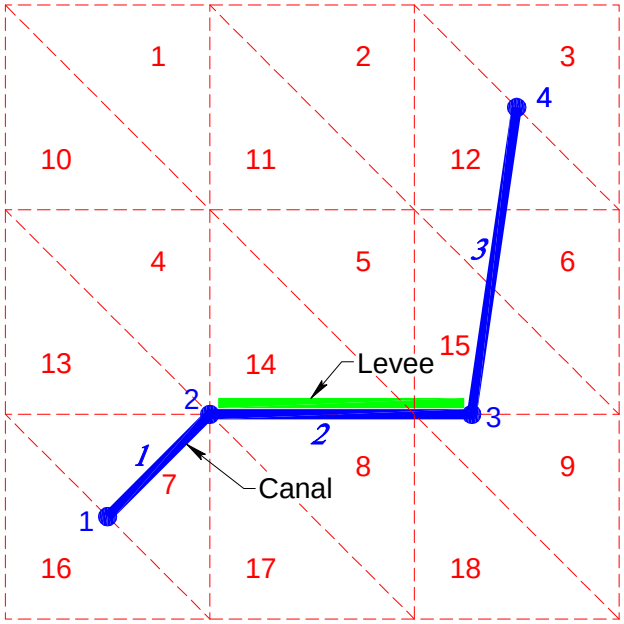


Figure 3.12: *A schematic of the canal network.*

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

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<geometry>`	Network geometry file is to be specified.					
`file`	GMS file name	NA	String	Any valid GMS file name	Req.	`enp_can.map`
`mult`	Multiplier for nodal coordinates	NA	Real	> 0	1.0	0.3048
`multxs`	Multiplier for segment cross section values. Often for unit conversion	NA	Real	Any real	1.0	0.3048
`<initial>`	Segment initial condition file to be specified					
`file`	Initial condition file name	NA	String	Any valid file name	Req.	L8ini.ini
`<reaches>`	Still to be described					
`file`	File to be described	NA	String	Any valid file name	Req.	File.Fil
`<network_bc>`	Network boundary conditions are specified in XML. Details in section 4.2					
`<arcs>`	Properties of canal segments to override the GMS file. The properties are specified under the following subelements of `<indexed>` and assigned to canal segments using an index file.					
`compflag`	Specifies numerical computation approach for canal flow	NA	String	central or upwind	central	upwind
`<indexed>`	Indicates there is a file for indexing of canal segment properties					
`file`	Index file name	NA	String	Any valid file name	Req.	arcs.index
`<xsentry>`	Indicates that following information is to override GMS data					
`id`	Entry id for indexed values	NA	Integer	Any integer	Req.	3
`label`	label for the entry	NA	String	Any string	Unspecified	Big Swamp
`<arcflow>`	Manning's n for canal flow is specified					
`n`	Manning's n for canal flow	$TL^{-1/3}$	Real	0.0 - 1.0	Req.	0.04
`<arcseepage>`	parameters for leakage between canal and cell are specified					
`leakage_coeff`	Leakage coefficient in Equation 3.63	T^{-1}	Real	0.0 - 1.0	Req.	0.00045
`<arcoverbank>`	parameters for overland flow between canal and cell are specified					
`bank_height`	Height of the canal lip	L	Real	0.0 - 5.0	Req.	0.6
`bank_coeff`	Coefficient for flow over the canal lip	NA	Real	0.0 - 1.0	Req.	0.3
`<arclevee>`	Parameter for seepage through a levee is specified					
`coeff`	Coefficient for seepage between canal and cell through a levee. Overrides levseep1	NA	Real	0.0 - 1.0	Req.	0.003

NA = Not Applicable; Req. = Required.

An example of XML input under the elements listed in Table 3.48 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>
```

3.6.2 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 (Figure 3.12). 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 (Figure 3.14) 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 3.49. An example of a canal network within a 2-D mesh is shown in Figure 3.13. Table 3.50 and Table 3.51 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 3.50 and Table 3.51

Table 3.49: *Definition of attributes specified 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

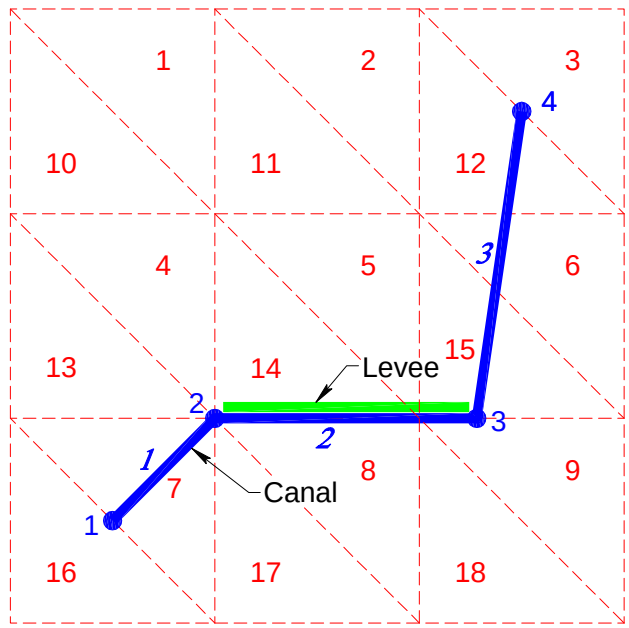


Figure 3.13: *A schematic of the canal network with a levee.*

3.6.2.1 Description Of Nodes

The four node environments in the geometry file specify the locations of nodes 1, 2, 3 and 4 in Figure 3.13. 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 (Table 3.50) 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
```

Table 3.50: *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
```

Table 3.51: *Sample canal geometry file, part 2 of 2.*

```
ARC
type trapezoid 100.0 498.0 0.5 0.05
ID 1
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
```

3.6.2.2 Canal Cross Sectional Geometry

The properties of 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 3.14 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.

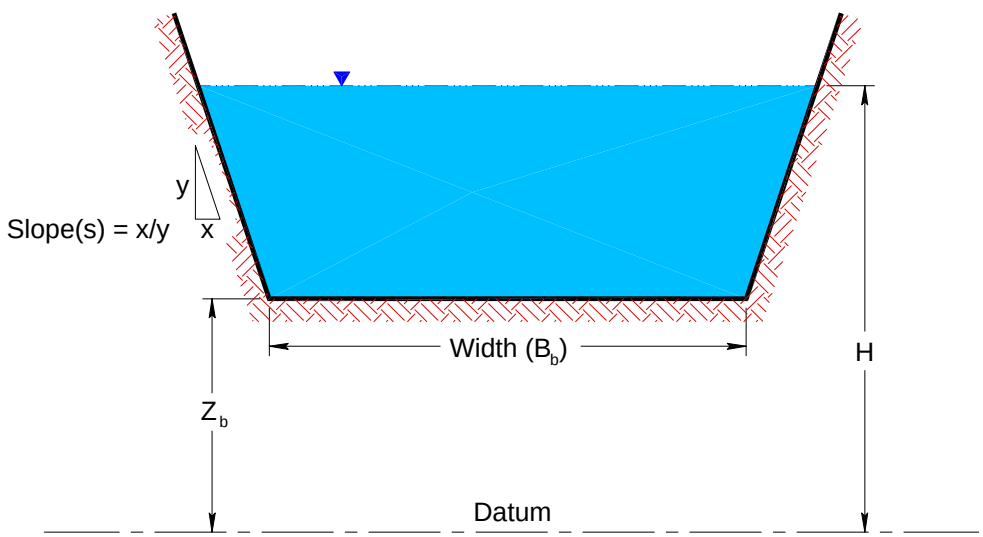


Figure 3.14: *Trapezoidal canal cross section.*

3.6.3 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 (Table 3.51). 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 3.15 shows a definition sketch used in the conceptualization of stream-aquifer and stream-overland flow interaction.

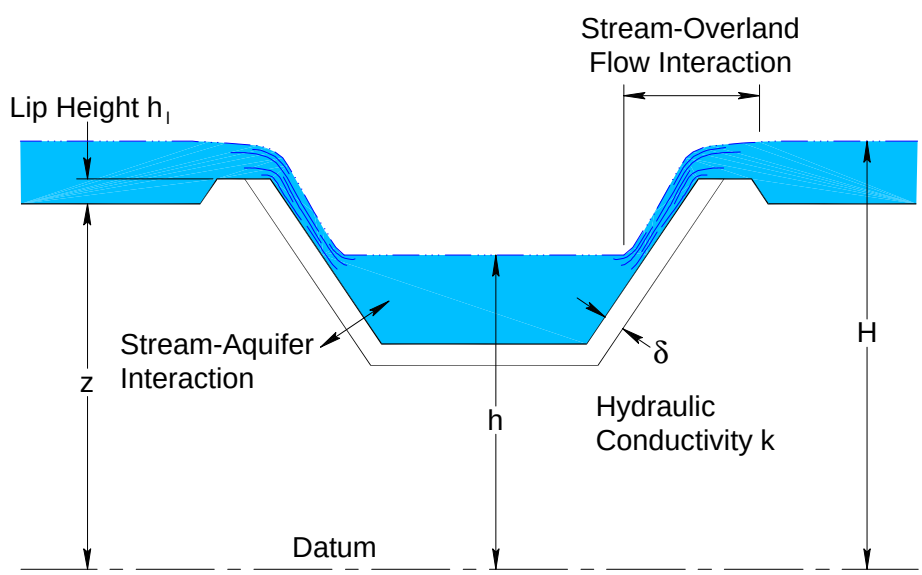


Figure 3.15: *A definition sketch showing flow interaction with the canal.*

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

$$q = \frac{k}{\delta} p(H - h) \tag{3.63}$$

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. There is not a single value of k/δ that separates the two regions, rather k/δ is a function of dimensionless parameters and depends on the details of the aquifer and the canal segment. 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.

3.6.4 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 3.15. 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 3.15
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 the bank height (h_l in Figure 3.15) ≥ 0.

3.6.5 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 3.16 shows a cross section of the levee illustrating components of seepage.

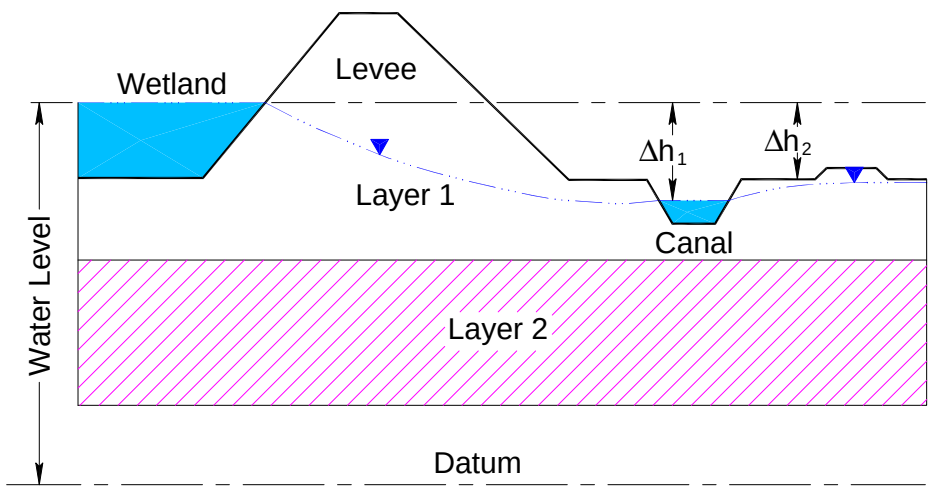


Figure 3.16: *Definition sketch showing levee seepage.*

Figure 3.17 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

$$q_l = \beta_0 + \beta_1 \Delta h_1 + \beta_2 \Delta h_2 \tag{3.64}$$

where
β_0, β_1 and β_2 are constants derived from experimental data, and
$\Delta h_1 = h_{wetland} - h_{canal}$, and
$\Delta h_2 = h_{wetland} - h_{cell}$,

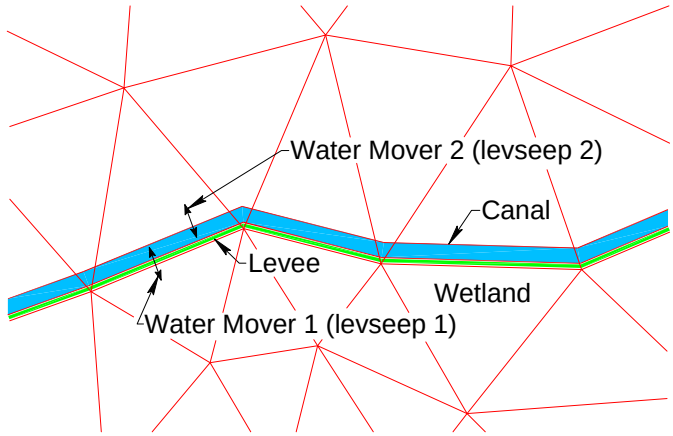


Figure 3.17: *Plan view showing the placement of a levee.*

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 3.64 can be derived from experimental data, they are typically derived from analytical or numerical models.
Since HSE is based on water movers that consider only two water bodies at a time, Equation 3.64 can be written as

$$q_l = -\beta_2(\Delta h_1 - \Delta h_2) + (\beta_1 + \beta_2)\Delta h_1 \tag{3.65}$$

where $\beta_1 + \beta_2$ 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 3.64. 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 3.50 and Table 3.51 as applied to Figure 3.13, 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.

3.6.6 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 3.13 is shown below. The heads are specified so that the depth in each canal segment specified in Table 3.51 is 5.5 meters.

```
netinit
503.5
501.5
500.5
```

3.6.7 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 3.48. 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 3.52: *Sample index file.*

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

to segments with the index file will override parameter values specified in the map file. The index file in Table 3.52 when used with the sample XML input file in subsection 3.6.1 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. If no values are to be overwritten, include a skeleton xsentry "j" that is assigned to the segments where no values are to be modified. This structure is shown in Table 3.53 where xsentry 2 is a "skeleton" entry. Using this xml file with the index file in Table 3.52 will modify no parameters in segment 2.

Table 3.53: *Skeleton xsentry that modifies no values.*

```
...
    <arcs>
      <indexed file="arcs.index">
        <xsentry id="1">
          <arcflow n="0.2"></arcflow>
          <arcseepage leakage_coeff="0.000405">
        </xsentry>
        <xsentry id="2">
        </xsentry>
       </indexed>
        <xsentry id="3">
          <arcflow n="0.2"></arcflow>
          <arcseepage leakage_coeff="0.000405">
        </xsentry>
    </arcs>
...
```

3.7 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

$$A_s \frac{dH}{dt} = \sum Q_{in} - \sum Q_{out} \tag{3.66}$$

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 3.18 shows a definition sketch of a reservoir, to which water is fed from an upstream river. Figure 3.19 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 3.54 with additional details in Table 3.55 and Table 3.56.

A sample XML input for a lake and a pond is shown in Table 3.57. 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 dss 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.

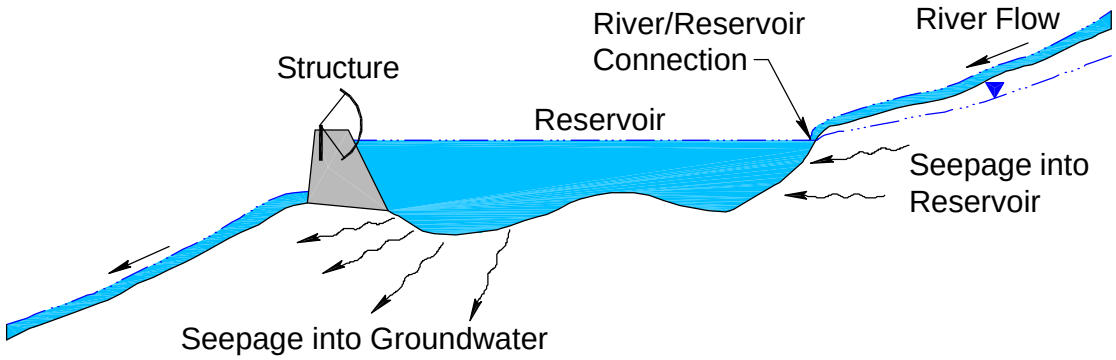


Figure 3.18: *Schematic diagram of a reservoir formed in a river.*

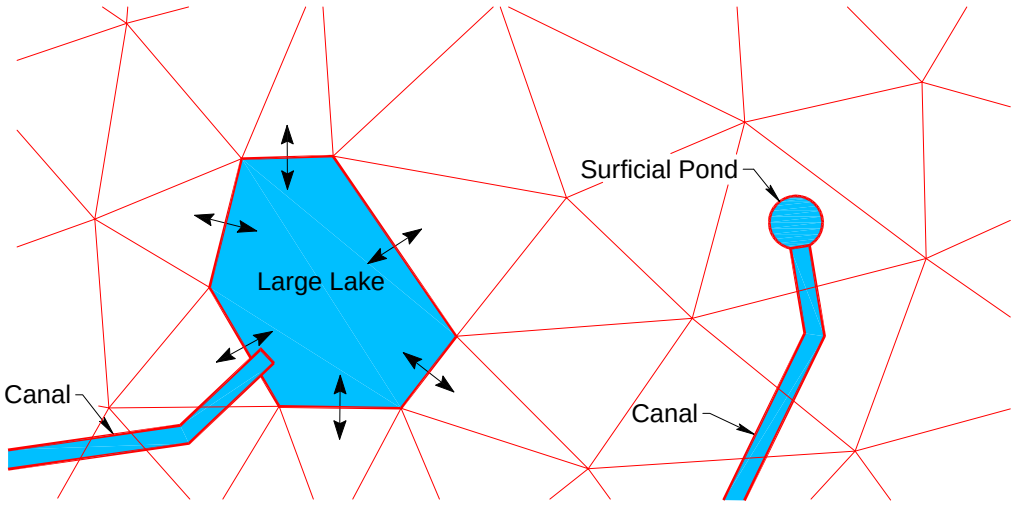


Figure 3.19: *Discretization around a lake and a pond.*

Table 3.54: *Sub-elements and attributes used to define lake properties under the* `<lake>` *element.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<lake>`	Indicates a new lake/pond water body is to be created					
id	Lake or Pond ID	NA	Integer	Any valid long integer	Req.	234625
label	Name of the lake	NA	String	Any valid string	undefined	Ocala
head0	Initial head in the lake	L	Real	Any real	Req.	13.64
top	Full supply level - not necessarily the same as "top" in Table 3.55	L	Real	Any real	1000	17.32
supplant	Indicates whether the wb is a lake or a pond	NA	Integer	$\leq$ 0 for lake, $\geq$ 0 for pond	-1	-2
`<parabolic>`	Indicates that lake area, volume, and stage are computed assuming a parabolic shape. Details in Table 3.55					
`<cylinder>`	Indicates that lake area, volume and stage are computed assuming a cylindrical shape. Details in Table 3.55					
`<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>`, and `<dss>` formats is explained in chapter 6						
`<EvapRainStressors>`	Parameters for calculation of ET are specified. Details in Table 3.56					
`<lake_bc>`	Lake boundary conditions are defined. Details in section 4.4					
NA = Not Applicable; Req = Required.						

Table 3.55: *Elements and attributes used to define lake area and volume under the* `<lakes>` *element.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<parabolic>`	Indicates that lake area, volume, and stage are computed assuming a parabolic shape					
`toparea`	Area of the lake surface when full	L^2	Real	Any valid real	Req.	6.3E+5
`top`	Head of the lake when full	L	Real	Any valid real	Req.	23.56
`bot`	Elevation of the lake bottom	L	Real	Any valid real	Req.	4.87
`<cylinder>`	Indicates that lake area, volume and stage are computing assuming a cylindrical shape					
`toparea`	Area of the lake surface when full	L^2	Real	Any valid real	Req.	6.3E+5
`bot`	Elevation of the lake bottom	L	Real	Any valid real	Req.	4.87
NA = Not Applicable; Req. = Required.						

3.7.1 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

$$ET_Volume = [swcoef * (DryArea + ShallowArea) + owcoef * DeepArea]\, RefET \tag{3.67}$$

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 3.56.

Table 3.56: *Elements and attributes used to define* `<EvapRainStressors>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<litZoneET>`	Lake ET parameters are specified					
`lakeID`	ID of the lake	NA	Integer	Any valid wb ID	Req.	138245
`owcoef`	Open Water coefficient for RefET	NA	Real	Any real	Req.	1.00
`swcoef`	Shallow water coefficient for RefET	NA	Real	Any real	Req.	0.85
`swdepth`	Depth that divides shallow and deep water	L	Real	Any real	Req.	5.5
NA = Not Applicable; Req = Required.						

3.7.2 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

$$Seepage = LCD(H_u - H_d) \tag{3.68}$$

where L and C are the length and conveyance as described in Table 3.58,
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 3.58. Table 3.59 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 3.57: *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>
       <dss file="FrogPrec.dss" pn="/Area1/Frog//RAIN/1DAY/TippBucket/" dbintl="1440"
       type="PER-CUM"label="gauge1" </dss>
     </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 3.58: *Elements and attributes used to define the lake seepage water mover.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<lakeseepage>`	Indicates the creation of a `<lakeseepage>`watermover.					
`lakeID`	Lake ID number	NA	Integer	Any valid lake ID	Req.	123723
`wbID`	Cell ID number that interacts with lake	NA	Integer	Any valid cell ID	Req.	469852
`wmID`	Water Mover id	NA	Integer	Any valid wm ID	-1	35469
`label`	Label for the Water Mover	NA	String	Any string	seepage-lakewm+wmID	Lake Wales seepage
`length`	Length of the shoreline in contact with the cell	L	Real	≥ 0.0	Req.	689.4
`conveyance`	Transmissivity of aquifer as defined in Equation 3.68	T^{-1}	Real	≥ 0.0	Req.	0.0054
NA = Not Applicable; Req. = Required.						

Table 3.59: *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>
```

3.8 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 3.20 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.

3.8.1 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

$$(H - z_B)s_c, \tag{3.69}$$

and the volume above ground is represented as

$$(H - z) + (z - z_B)s_c, \tag{3.70}$$

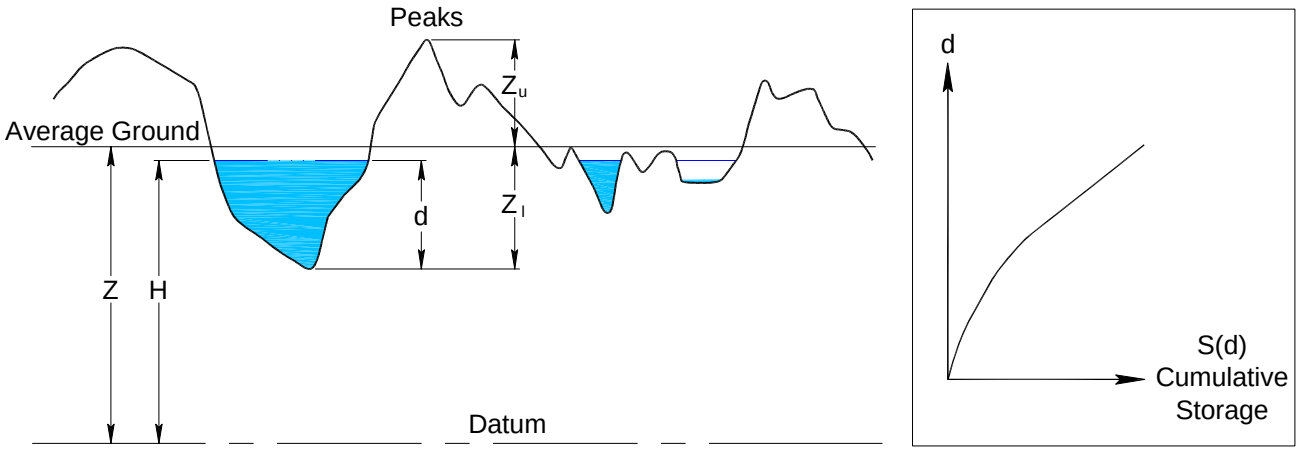


Figure 3.20: *Describing stage-storage characteristics in micro-topography.*

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>
```

3.8.2 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 3.20), 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

$$S(d) = \sum \alpha_A(H) + (1 - \alpha_A(H))s_c(H) \tag{3.71}$$

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 3.20), the added volume has a storage coefficient of 1.0. The total volume of water can be computed by integrating (3.71) 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 3.60.

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>`.

Table 3.60: *Elements and attributes used to define a* `<lookupsv>` *SV converter in the* `<mesh>` *environment.*

<Element> or Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
`<lookupsv>`	Designates an `<lookupsv>` stage-volume converter.					
sc	Storage coefficient below the SV converter, s_c	NA	Real	0 - 1	Req.	0.23
below	Depth of the lowest point below the ground surface at which the lookup table applies	L	Real	0 - 5	Req.	1.3
above	Height of the highest point above the ground surface at which the lookup table applies	L	Real	0 - 5	Req.	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					
NA = Not Applicable; Req. = Required.						

The curve in Figure 3.20 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).

3.8.3 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 4

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, networks, and lakes are presented.

4.1 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 and walls separating cells are defined within the <mesh_bc> environment as described in detail and illustrated below.

Figure 4.1 shows a 3x3 mesh with the boundary conditions available for cells and walls indicated. The constant <wallghb> boundary condition is applied to the walls defined by nodes 1, 5, 9. Walls defined by nodes 9, 13, 14, 15 , and 16 also has a <wallghb> boundary condition but the head value is interpolated according to specified weighting between two time series. A <wallhead. boundary condition is applied to the walls defined by pairs of nodes 1-2, 2-3, 3-4, and 8-12 with a head value linearly interpolated between a constant and a time series. A <walluf> boundary condition is indicated for wall 12-16 although this boundary codition is not yet fully implemented. Overland flow is blocked by a <noflow> boundary condition along the walls defined by nodes 2, 6, and 11. Cell boundary conditions are a <well> pumping a time series of flows from cell 3, a <cellheasd> specified by a rating curve in cell 7, and a <cellghb> with a constant head in cell 4. These boundary conditions and the the XML input required to apply them are described in detail in the following sections.

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 6.1 . All boundary conditions except <cellhead> and <segmenthead> are implemented by creating water mover objects designed to move water in such a way as to achieve the prescribed flow or head.

4.1.1 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 4.1 and wall based BC's are shown in Table 4.2 and Table 4.3. The methods available to define the locations of the walls and cells to which the boundary conditions apply

are described in subsubsection 4.1.2.1, while subsubsection 4.1.2.2 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 Table 4.1 through Table 4.3.

Table 4.1: *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	Dimensions	Variable type	Suggested range	Default	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	NA	Integer	Any valid cell ID	Req.	234625
`wellid`	Optional user-defined ID for the cell to be used later in water budget output	NA	Integer	Any valid cell ID	-1	234625
`label`	Optional label used to describe the flow	NA	String	Any string	well-cell +cellID	Water Supply Well
`<cellhead>`	Indicates a constant or time series head value assigned to a cell					
`id`	ID of the cell whose head is specified	NA	Integer	Any valid cell ID	Req.	546987
`bcid`	ID assigned to the BC	NA	Integer	Any valid cell ID	-1	234625
`label`	Optional label to describe the boundary condition	NA	String	Any string	headbc-cell +ID	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	NA	Integer	Any valid cell ID	Req.	546987
`bcid`	ID assigned to the BC	NA	Integer	Any valid ID	-1	35896
`value`	The coefficient in Equation 4.5	L^2T^{-1}	Real	Any valid real	Req.	0.35
`label`	Optional label to describe the boundary condition	NA	String	Any string	ghb-cell+ID	Lake Calloway
Each of the elements `<well>`, `<cellhead>` and `<cellghb>` has the following sub-elements available for specifying the flow or head: `<const>` and `<dss>`. `<well>` can also have data input under `<rc>` and `<tsin>`. The current version of the DTD limits the input options under `<cellhead>` and `<cellghb>`. These elements and their attributes are described in detail in section 6.1; Note: NA = Not Applicable; Req. = Required.						
Note: NA = Not Applicable; Req. = Required.						

Table 4.2: *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	Dimensions	Variable type	Suggested range	Default	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	NA	String	ol, gw, `ol_gw`	`ol_gw`	`ol_gw`
`label`	An optional label identifying the boundary condition	NA	String	Any String	wallhead-cell+idCell+-sw	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 Equation 4.2	LT^{-1}	Real	Any real	Req.	0.046
`label`	An optional label identifying the boundary condition	NA	String	Any String	wallghb-cell +idcell	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	NA	Integer	Pairs of adjacent nodes	No default	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	NA	Integer	A series of adjacent nodes	No default	235467 546575 345764 867456 654763
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 4.5. `<endpts>` is not available for the `<wallghb>` boundary condition unless the DTD is modified.						
Sub-elements available for specifying the input data for the wall boundary conditions `<wallhead>` and `<wallghb>` are `<const>`, `<dss>`, `<rc>` and `<tsin>` The particular subelements available as defined by the DTD depend on the type of interpolation as detailed in Table 4.5. The input formats are described in detail in chapter 6						
NA = Not Applicable; Req. = Required.						

Table 4.3: *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	Dimensions	Variable type	Suggested range	Default	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	NA	String	ol, gw, `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 Equation 4.3	NA	Real	Any real	Req.	0.014
The following 2 sub-elements and attributes are available under `<walluf>`. Only `<nodelist>` is available under `<noflow>` as specified in the DTD.						
`<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	NA	Integer	Pairs of adjacent nodes	No default	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	NA	Integer	A series of adjacent nodes.	No default	235467 546575 345764 867456 654763
NA = Not Applicable; Req. = Required.						

4.1.2 Defining Attributes Of 2-D BC's

The example XML input in Table 4.4 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 4.1. 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 4.4 is also included in the description of each option.

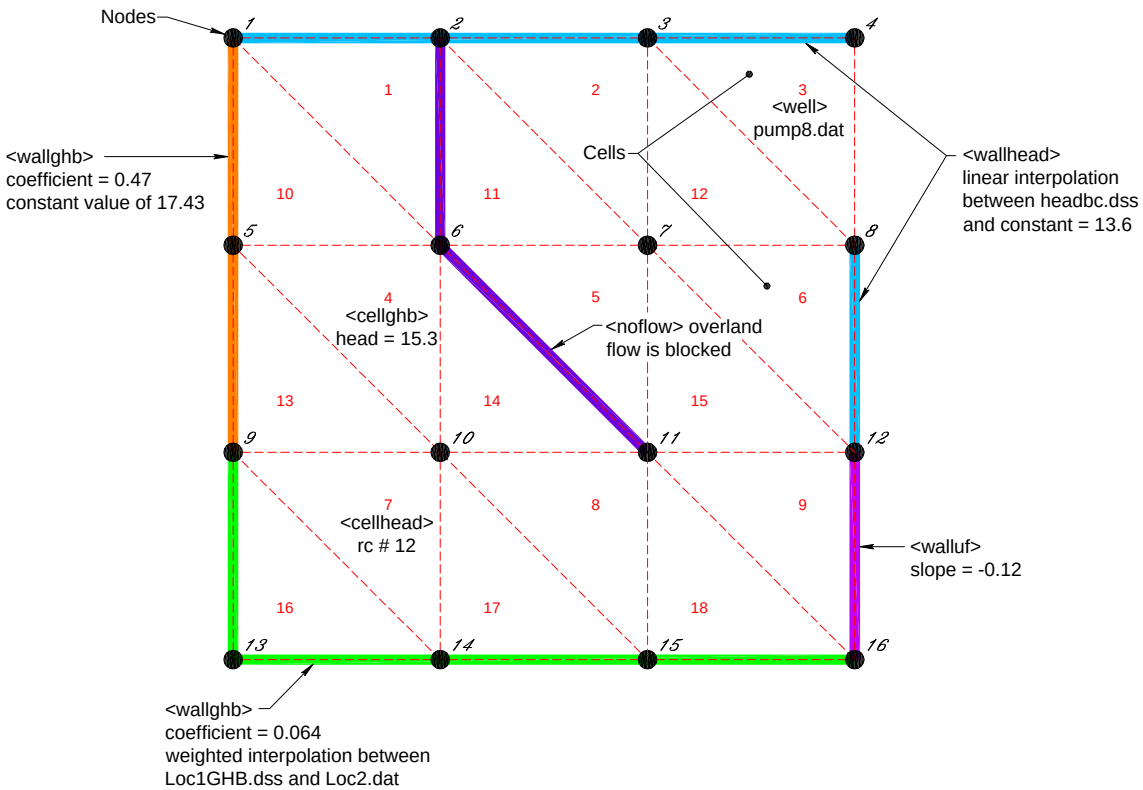


Figure 4.1: *Illustration of the application of 2D Mesh Boundary Conditions.*

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

A cell boundary condition is applied to a single cell specified by the cell ID as demonstrated by the `<well>` discharge into cell 3 defined in Table 4.4 and shown in Figure 4.1. 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 under one of two sub-elements. Under `<walllist>` pairs of nodes are listed with each

Table 4.4: *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">
      <dss file="PondHead.dss" pn = "/AvgHead/Pond/Stage/05AMR1993/1DAY/normal/ "dbintl="1440" >
      </dss>
   </cellghb>
   <well  cellid="3" wellid="7658" label="Pump Station 8">
      <dss file="Pump8.dss" pn = "/NormalRules/Pump8/Flow/05AMR1993/1DAY/normal/"
         mult="0.02831685" dbintl="1440">">
      </dss>">
   </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">
            <dss file="Loc2.dss" pn = "/AvgHead/Loc2/Stage/05AMR1993/1DAY/normal/ "dbintl="1440" >
            </dss>
         </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>
```

pair specifying a wall. These walls need not be adjacent. Under `<nodelist>` a continuous series of adjacent nodes is listed defining one or more adjacent walls to which the boundary condition is applied. Only a continuous series of walls may be specified under `<nodelist>`.

In the example XML input in Table 4.4 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 4.1. Note that the walls need not be contiguous. The `<noflow>` boundary condition for overland flow in Table 4.4 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 Table 4.2 and Table 4.3.

```
<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>
```

Table 4.5: *Elements and attributes used to assign interpolation weighting to the* `<wallhead>` *and* `<wallghb>` *boundary conditions. Element cells are shaded.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	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>` except that `<endpoints>` does not apply to `<wallghb>` as specified in the DTD.						
`<uniform>`	Each wall is assigned the same value specified by `<const>` or by a time series under the subelements `<dss>`, `<rc>`, or `<tsin>` as described in chapter 6					
`<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 is inserted.	The interpolation weights assigned to the walls specified by `<nodelist>` or `<walllist>`. One entry for each wall	NA	Real	≥ 0.0 and ≤ 1.0	No default	1.0 0.65 0.5 0.4 0.0
`<entry>`	Indicates that a constant or time series of head values will be specified under the subelements `<const>` or `<dss>`. Two entries are required; one to specify the head for the first wall and one for the last.					
`id`	The ID of the entry	NA	Integer	Any valid integer	Req.	234
`<endpts>`	Entries for linear interpolation will be specified					
`<entry>`	Indicates that a constant or time series of head values will be specified under the subelements `<const>` or `<dss>`. Two entries are required; one to specify the head for the first wall and one for the last.					
`id`	The ID of the entry	NA	Integer	Any valid integer	Req.	234
NA = Not Applicable; Req. = Required.						

4.1.2.2 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 defined by two nodes. 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 4.5. `<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 4.4 and Figure 4.1.

```
<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>`. For an intermediate weighting value of x for the jth wall, the boundary condition value assigned to the jth wall is x*(value at the first wall)+(1-x)*(value at the last wall). 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 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 4.4 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">
            <dss file="Loc2.dss"
            pn = "/AvgHead/Loc2/Stage/05AMR1993/1DAY/normal/ "dbintl="1440" >
```

```
            </dss>
         </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 4.4 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>
```

4.1.2.3 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>`, `<const>`, and `<rc>`. These formats and their use are described in detail in chapter 6.

4.1.3 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.

4.1.3.1 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 4.3 and demonstrated below and in Table 4.4 and Figure 4.1 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>
```

4.1.3.2 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 4.4 and Equation 4.1. 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 4.4 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 $8-12$.

$$H_i = H_B(t) \tag{4.1}$$

```
<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>
```

4.1.3.3 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 4.2

$$Q_i(t) = \left[\frac{(K_B L_w)^2}{K_w L_w + \frac{K_g L_w}{L_f} + K_B L_w} \right] (H_B - H_i) \tag{4.2}$$

where
H_i = head in the cell,
H_B = specified boundary condition head.
K_B = a user input coefficient that controls the rate of flow through the wall.
K_w = conveyance in the cell.
K_g = transmissivity in the cell.
L_w = length of the wall forming the boundary of the cell.
L_f = characteristic flow length.

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 4.4 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 4.2 is 0.064. The specified head is interpolated according to the specified weights between two time series in DSS files.

```
<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">
            <dss file="Loc2.dss"
```

```
                pn = "/AvgHead/Loc2/Stage/05AMR1993/1DAY/normal/ dbintl="1440" >
              </dss>
            </entry>
            <wts> 1.0 0.2 0.8 0.0 </wts>
          </wts2pts>
        </wallghb>
      </mesh_bc>
```

4.1.3.4 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 4.3.

$$Q_B = K_i S_B \tag{4.3}$$

where 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 4.4 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>
```

4.1.4 Boundary Condition Types For Cells

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

4.1.4.1 Inflow BC `<well>`

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

$$Q_i = Q_B(t) \tag{4.4}$$

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 4.1. An example of a time series of pumping rates into cell 3 is demonstrated below, in Table 4.4, and Figure 4.1. The flow is defined by a time series in a dss format.

```
<mesh_bc>
   <well  cellid="3" wellid="7658" label="Pump Station 8">
      <dss file="Pump8.dss" pn = "/NormalRules/Pump8/Flow/05AMR1993/1DAY/normal/"
         mult="0.02831685" dbintl="1440">
      </dss>
   </well>
</mesh_bc>
```

The default data type from the file Pump8.dat is INST-VAL or flow. The user has the option to specify a volume by specifying the data type PER-CUM as in the following example. In this example 1000 units of volume is distributed uniformly over each month. The entry dbintl="43200" indicates a monthly value and the model adjusts for the number of days in each month to achieve a uniform distribution. The model determines the volume to be applied in each time step and converts this to a rate.

```
<mesh_bc>
   <well  cellid="3" wellid="7658" label="Pump Station 8">
      <const value="1000 type="PER-CUM" dbintl="43200"\>
   </well>
</mesh_bc>
```

4.1.4.2 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 4.1. 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 4.4 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>
```

4.1.4.3 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

$$Q_i = K_B(H_B(t) - H_i) \tag{4.5}$$

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 4.5 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 4.4. 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.dss.

```
<mesh_bc>
   <cellghb id="4" value="0.15" label="Lake Isabel">
      <dss file="PondHead.dss" pn = "/AvgHead/Pond2/Stage/05AMR1993/1DAY/normal/
      "dbintl="1440" label="Head maintained in irrigation pond 3A" >
      </dss>
   </cellghb>
</mesh_bc>
```

4.2 Boundary Conditions for the 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. Table 4.6 and Table 4.7 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> and <rc>. Details on the use of these options are explained in chapter 6.

Table 4.6: *Elements and Attributes for Specifying Boundary Conditions for Canal Networks Part 1. Element cells are shaded.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<segmentsource>`	Used to specify the inflow to a segment as a constant or a time series					
`id`	ID of the segment	NA	Integer	10000-20000	Req.	156324
`bcid`	Optional BC ID	NA	Integer	> 0	-1	23
`label`	Optional label to identify the BC	NA	String	Any String	source-seg+ID	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	NA	Integer	10000-20000	Req.	164824
`bcid`	Optional BC ID	NA	Integer	> 0	-1	23
`label`	Optional label to identify the BC	NA	String	Any String	headbc-seg+ID	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	NA	Integer	10000-20000	Req.	164824
`kcoeff`	Coefficient in Equation 4.9	L^2T^{-1}	Real	≥ 0.0	Req.	0.12
`bcid`	Optional BC ID	NA	Integer	> 0	-1	23
`label`	Optional label to identify the BC	NA	String	Any String	ghb-seg+ID	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	NA	Integer	10000-20000	Req.	164824
`bcid`	Optional BC ID	NA	Integer	> 0	-1	23
`label`	Optional label to identify the BC	NA	String	Any String	jctheadbc-seg+ID	Pond 3B
Sub-elements available for specifying the head for the canal boundary conditions `<segmentsource>`, `<segmenthead>`, `<segmentghb>`, and `<junctionhead>` are `<const>`, `<dss>`, `<rc>`, and `<tsin>`. These are described in detail in chapter 6.						
NOTE: NA = Not Applicable; Req. = Required;						

Table 4.7: *Elements and Attributes for Specifying Boundary Conditions for Canal Networks Part 2. Element cells are shaded.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	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.					
`id1`	ID of the canal segment	NA	Integer	10000-20000	Req.	15639
hline `id2`	Needs definition	NA	Integer	10000-20000	Req.	13568
`slope`	The slope in equation Equation 4.8	NA	Real	0.0-0.2	Req.	0.015
`bcID`	Boundary condition ID	NA	Integer	10000-20000	Req.	14793
`label`	Description of the boundary condition	NA	String	Any String	uniform-seg+bcID	End of Miami canal
`<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	NA	Integer	10000-20000	Req.	164824
`id2`	ID of the segment on the other side of the junction	NA	Integer	10000-20000	Req.	164824
NA = Not Applicable; Req. = Required.						

Sample XML input for all six network boundary conditions under the `<network_bc>` element is shown in Table 4.8 and demonstrated in Figure 4.2. Each available boundary condition will be explained in detail in the following sections.

4.2.1 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.

$$Q_i = Q_B(t) \tag{4.6}$$

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 6. The XML input below excerpted from Table 4.8 specifies a flow into segment 1 defined by a time series in the file "DevilFlow.dss". The boundary condition may also be specified as a volume using the data type PER-CUM as described for the `<well>` boundary condition for cells in subsubsection 4.1.4.1. The application of this BC is shown graphically in Figure 4.2. 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.

```
<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>
```

4.2.1.1 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 4.7.

$$H_i = H_B(t) \tag{4.7}$$

Table 4.8: *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">
         <dss file="BigLake.dss"
         pn = "/Southshore/BigLake/STAGE/01JAN1996/1DAY/normal/"
            mult="1.0" dbintl="1440">">
         </dss>
      </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">
         <dss file="Lola.dss"
         pn = "/AvgHead/Lola/Stage/05AMR1993/1DAY/normal/ "dbintl="1440" >
         </dss>
      </segmentghb>
      <junctionblock id1="9" id2="10" label="Highway 66 culvert">
      </junctionblock>
   </network_bc>
</network>
```

where $H_B(t)$ can be a time series, a constant, or a 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 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">
         <dss file="BigLake.dss" pn = "/Southshore/BigLake/STAGE/01JAN1996/1DAY/normal/"
            mult="1.0" dbintl="1440">">
         </dss>
      </segmenthead>
    </network_bc>
....
```

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

If two canal segments are adjacent such as segments 10 and 11 in Figure 4.2 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 subsection 3.5.3. 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>
```

4.2.1.3 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

$$\frac{\partial H}{\partial x} = S_0 \tag{4.8}$$

where S_0 is the uniform flow slope assigned to the segment.

4.2.1.4 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;

$$Q_i(t) = K_B(H_B(t) - H_i) \tag{4.9}$$

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.dss.

```
 ....
<network>
   <network_bc>
       <segmentghb id="4" kcoef="0.045" bcid="11" label="Lake Lola">
         <dss file="Lola.dss" pn = "/AvgHead/Lola/Stage/05AMR1993/1DAY/normal/ "dbintl="14
         </dss>
      </segmentghb>
   </network_bc>
</network>
....
```

4.2.1.5 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 4.10.

$$Q = \frac{A_i}{n} {R_i}^{2/3} \left(H_B - H_i\right) \tag{4.10}$$

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">
         <dss file="Tide.dss" pn = "/Tide/Predicted/Stage/05AMR1993/15MIN/normal/ "dbintl=
         </dss>
      </segmenthead>
   </network_bc>
</network>
....
```

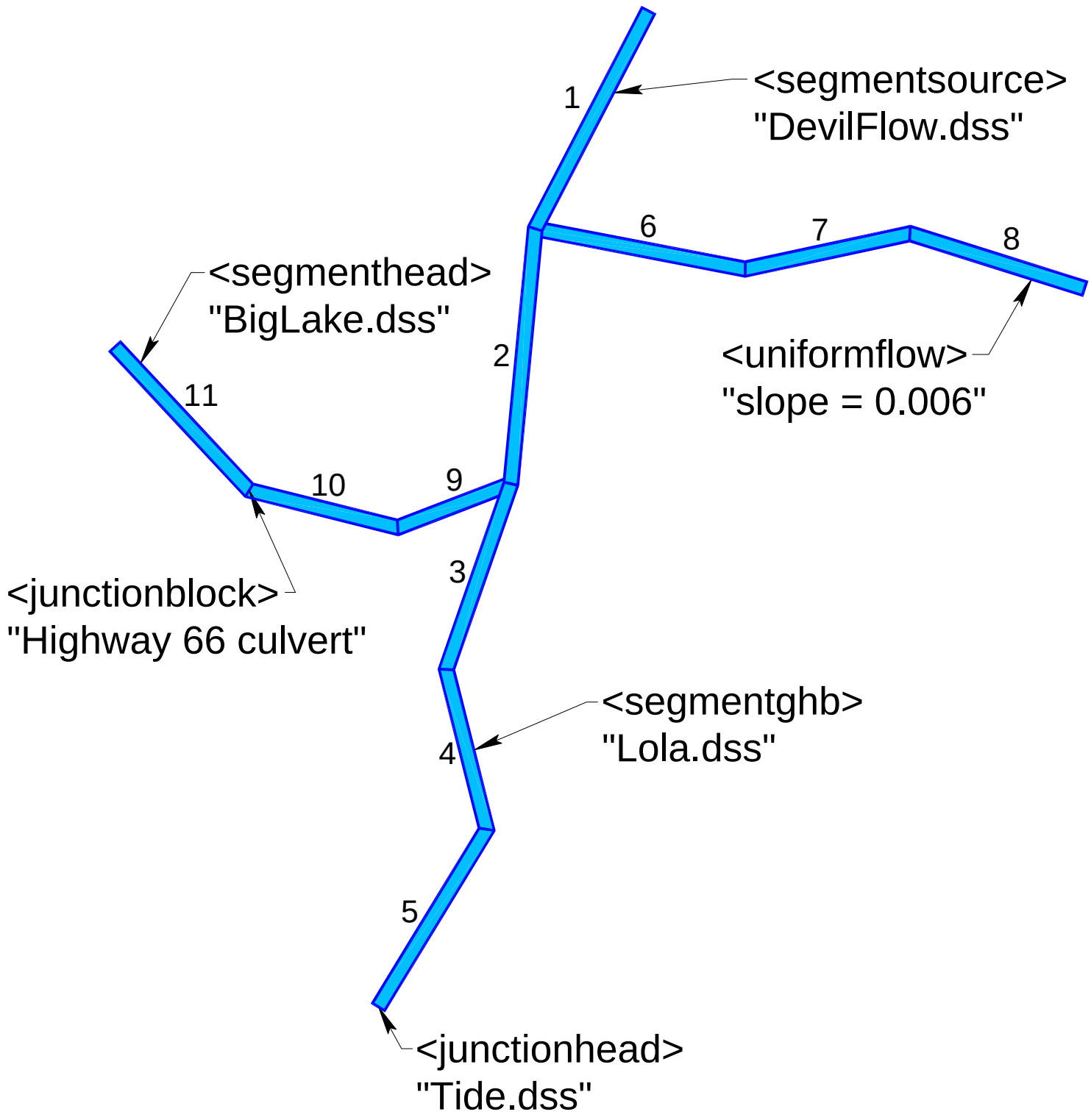


Figure 4.2: *Illustration of the application of Canal Network Boundary Conditions.*

4.3 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 4.9 and described in detail in the following sections. Sample XML input for these boundary conditions is displayed in Table 4.10. Note that these boundary conditions are specified within the `<watermover>` environment.

Table 4.9: *Elements and Attributes for Specifying Boundary Conditions for General Water Bodies in the* `<watermover>` *environment. Element cells are shaded.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	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	NA	Integer	Any valid water body ID	Req.	187324
`label`	Optional label to identify the BC	NA	String	Any string	wm+ID	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	NA	Integer	10000-20000	Req.	164824
`label`	Optional label to identify the BC	NA	String	Any string	wm+ID	Long Pond weir
`wmid`	Water mover ID	NA	Integer	> 0	Req.	23
`mult`	Multiplier for the 1D lookup values, often for unit conversion	NA	Real	Any real	1.0	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>`, and `<rc>`. These elements and their attributes are described in detail in chapter 6						
NA = Not Applicable; Req. = Required.						

Table 4.10: *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>
     40.0 5000.0
     50.0 4000.0
     52.0   10.0
     52.2    0.0
   </hq>
 </hq_relation>
</watermovers>
```

4.3.1 Sources And Sinks `<source>`

The `<source>` boundary condition performs nearly the same operation as the `<segmentsource>` and `<well>` boundary conditions, but it can be applied to any water body. Unlike the `<segmentsource>` and `<well>` boundary conditions, only flow and not volume may be specified. The inflow to or outflow from a water body is defined as

$$Q_i = Q_B(t) \tag{4.11}$$

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 6. In the example in Table 4.10 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.

4.3.2 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

$$Q_i = f(H_i) \tag{4.12}$$

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 4.10 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 52 meters with the flow rapidly reduced as the water level increases from 50 to 52 meters.

4.4 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 4.11 and described in detail in the following sections. Sample XML input for these boundary conditions is displayed in Table 4.12.

Table 4.11: *Elements and Attributes for Specifying Boundary Conditions for Lakes. Element cells are shaded.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	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	NA	Integer	Any valid lake ID	Req.	237345
`id`	Boundary condition ID	NA	Integer	Any integer	-1	17
`label`	Optional label to identify the BC	NA	String	Any String	No default	FPL pump 4
`<owet>`	The ET from the surface of a lake is specified as a function of the surface area of the lake.					
`lakeID`	ID of the lake	NA	Integer	10000-20000	Req.	254824
`id`	Boundary Condition ID	NA	Integer	> 0	-1	31
`label`	Optional label to identify the BC	NA	String	Any String	No default	Measured ET
`<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.					
The `<lakesource>` and `<owet>` elements have the following sub-elements available for specifying the flow or RefET: `<const>`, `<dss>`, `<rc>` and `<tsin>`. These elements and their attributes are described in detail in chapter 6						
NA = Not Applicable; Req. = Required.						

Table 4.12: *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>
    <dss file="STA17Evap.dss pn="/STA/17/FLOW/08MAR2003/1DAY// </dss>
  </owet>
</lakes>
```

4.4.1 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

$$Q_i = Q_B(t) \tag{4.13}$$

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 6. In the example in Table 4.12 a time series flow specified in STA17Pump.dss flows into STA17 (lake id = 237345) with a database interval of 1 day.

4.4.2 Open Water Evaporation Boundary Condition `<owet>`

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

$$Q_i = Area(H_i)RefET(t) \tag{4.14}$$

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 4.12 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.dss.

Chapter 5

Hydrologic Process Module Approach and Models

HPMs 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 HPMs explicitly simulate the local hydrology before the next time step of the implicit regional solution. HPM 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

5.1 HPM 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 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 HPMs currently in the RSM have been designed to simulate each of these types of areas. There is also a HPM that does nothing that is useful for turning off hydrologic processes in selected areas for model testing and development.

In the following discussion, HPM 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 HPM.

1. Natural System HPMs

   Natural Wetland System `<layer1nsm>`

   No Action `<layerpc>`

   Precipitation Runoff Routing `<prr>`

   Five Unsaturated Soil Layer `<layer5>`

   Unsaturated Soil `<unsat>`

2. Agricultural HPMs

   Agricultural Irrigation Requirements `<afsirs>`

   Drainage Collector Ditch `<pumpedditch>`

   Agricultural Impoundment `<agimp>`

3. Urban HPMs

   Multi-Basin Routing `<mbrcell>`

   Impervious Land `<imperv>`

   Urban Detention `<urbandet>`

   Consumptive Use `<cu>`

5.2 Natural System HPMs

The natural system HPMs 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. The natural systems HPMs are briefly described below:

1. The `<layer1nsm>` HPM type is used to represent the local hydrology for wetlands and high water table soils. This HPM 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>` HPM was developed as a simple runoff model that combines the NRCS curve number runoff algorithm with simple linear reservoir routing.

3. The `<unsat>` HPM is an extension of the layer1nsm HPM 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>` HPM is an extension of the unsat HPM. The layer5 HPM 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>` HPM has the capability of modeling the soil moisture of multiple soil horizons as would occur in a typical soil profile.

5. The `<layerpc>` HPM performs no calculations and uses none of the HPM 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 HPMs in a limited area or testing of other model components without HPMs. It may also be used with lower layer cells in a three dimensional groundwater simulation to maintain mass balance.

5.2.1 Natural Wetland System `<layer1nsm>`

This HPM was introduced to satisfy the need to simulate natural hydrology in natural system models. This HPM calculates a simple water budget for the soil with a water table that is defined by the water level in the mesh cell. Hydrologic processes that occur in this HPM are shown in Figure 5.1.

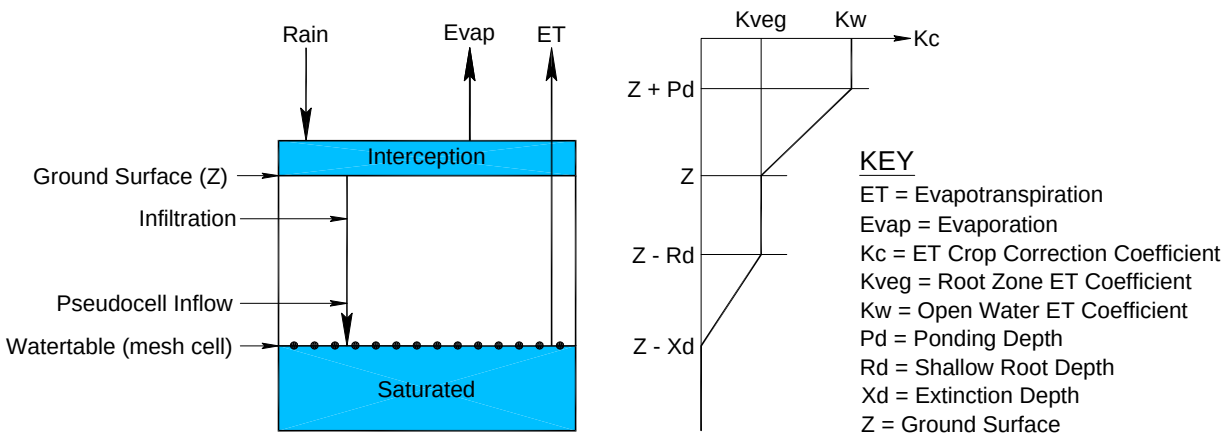


Figure 5.1: *HPM Components of Water Budget for the* `<layer1nsm>` *HPM. The variation of the Reference ET Crop Coefficient as a function of input parameters is also shown.*

Table 5.1: *Elements and Attributes for the* `<layer1nsm>` *HPM. Element cells are shaded.*

<Element> or Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
`<layer1nsm>`	Designates the NSM type HPM					
kw	Maximum crop coefficient for open water	NA	Real	0 - 1	Req.	1.0
rd	Shallow root zone depth (m)	L	Real	0 - 10	Req.	2.6
xd	Extinction depth below which no ET occurs (m)	L	Real	0 - 10	Req.	5.6
pd	Open Water ponding depth (m)	L	Real	0 - 2	Req.	1.3
kveg	Vegetation crop coefficient	NA	Real	0 - 1	Req.	0.7
`fld_cap`	Water content at field capacity (m)	L	Real	0.0 - 0.5	0.0	0.32
imax	Maximum Interception (m)	L	Real	0 - 1	0.0	0.15
NA = Not Applicable; Req. = Required.						

Table 5.2: *Example XML input for a* `<layer1nsm>` *HPM.*

```
...
   <pseudocell>
      <indexed file="lu.index">
         ...
            <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>
```

The XML elements and attributes used to describe a `<layer1nsm>` HPM are described in Table 5.1. An example of the XML input for a `<layer1nsm>` HPM is shown in Table 5.2. lu.index is an index file that assigns a HPM identified by its `<entry>` id to each mesh cell.

5.2.2 Three Dimensional Groundwater Cell HPM `<layerpc>`

The `<layerpc>` HPM 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 HPM acts as a placeholder for this function in a three dimensional groundwater simulation. A sample XML input for a `<layerpc>` HPM is shown Table 5.3.

Table 5.3: *Example XML input for a* `<layerpc>` *HPM.*

```
...
   <pseudocell>
      <indexed file="lu.index">
         ...
         <entry id="2">
           <layerpc> </layerpc>
         </entry>
         ...
      </indexed>
   <\pseudocell>
...
```

5.2.3 Multi-Basin Routing HPM `<mbrcell>`

The MBRcell was created to provide a simple HPM for modeling runoff and routing. Much of the runoff occurs before the soil is fully saturated. To simulate this behavior 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.

Table 5.4: *Elements, attributes, and typical values used for the* `<mbrcell>` *HPM.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<mbrcell>`	Designates the mbrcell type HPM					
route	wb ID for destination of discharge	NA	Integer	Any valid wb ID	Req.	7
tc	Time of concentration (s)	T	Integer	900 - 50,000	Req.	3600
kveg	Evapotranspiration crop coefficient	NA	Real	0.0 1.0	Req.	0.85
`d_shal`	Shallow root zone depth (m)	L	Real	0.5 - 3.0	Req.	2.0
`d_deep`	Deep root zone depth (m)	L	Real	1.0 - 5.0	Req.	3.5
`fld_cap`	Water content at field capacity (m)	L	Real	0.0 - 2.0	Req.	0.2
NA = Not Applicable; Req. = Required.						

Table 5.5: *Example XML for an* `<mbr>` *HPM.*

```
<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>
```

The elements and attributes used in defining an `<mbrcell>` HPM are shown in Table 5.4. A simple example of the MBRcell HPM is presented in Table 5.5.

5.2.4 Five Soil Layer HPM `<layer5>`

The `<layer5>` HPM shown in Figure 5.2 simulates the water budget in a layered unsaturated soil with rainfall and potential evapotranspiration as meteorological input. 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. 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 elements and attributes used to define a `<layer5>` HPM are described in Table 5.6. 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 HPM is attached.

An example of the XML input for the layer5 HPM is provided in Table 5.7 and in Benchmark 18.

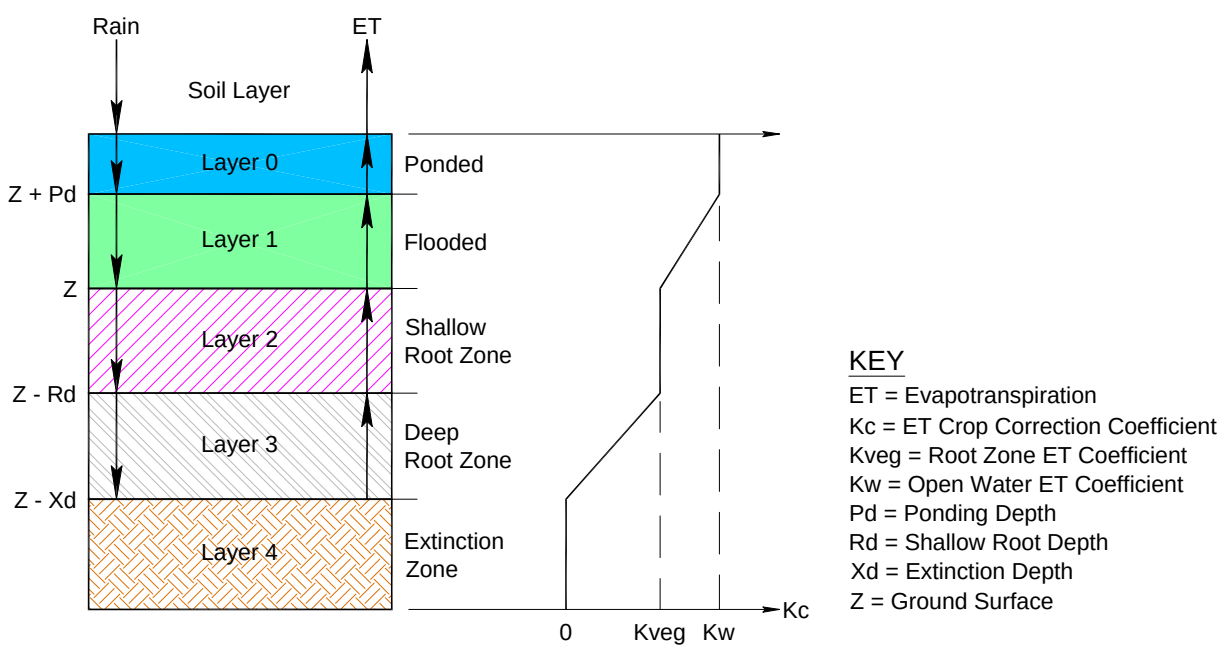


Figure 5.2: *Soil layers modeled in the* `<layer5>` *HPM and the variation of the ET coefficient, Kc with water table.*

Table 5.6: *Elements and Attributes for the* `<layer5>` *HPM.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<layer5>`	Designates the layer5 type HPM					
ew	Extractable water	L	Real	0.0 - 0.5	Req.	0.2
kw	Maximum crop coefficient for open water	NA	Real	0.5 - 1.2	Req.	1.0
rd	Shallow root zone depth, (m)	L	Real	0.2 - 2.0	Req.	1.0
xd	Extinction depth below which here is no ET, (m)	L	Real	1.0 - 10.0	Req.	3.0
pd	Ponding depth, (m)	L	Real	0.0 - 2.0	Req.	1.0
kveg	Vegetation crop coefficient	NA	Real	0.0 - 1.0	Req.	0.85
NA = Not Applicable; Req. = Required.						

Table 5.7: *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>
    </HPM>
...
```

5.2.5 Unsaturated Soil HPM `<unsat>`

The Unsat HPM computes a simple water budget for a single-layer soil, Figure 5.3. This HPM is similar to the `<layer1nsm>` HPM 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 elements and attributes used to describe the Unsat HPM are presented in Table 5.9. The specific yield is determined by the soil properties of the underlying mesh cell.

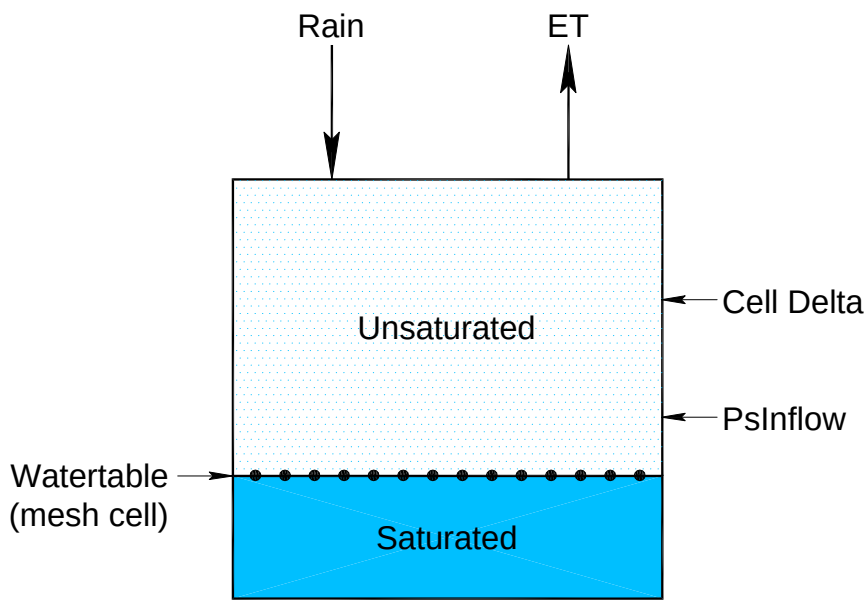


Figure 5.3: *Schematic water budget for* `<unsat>` *HPM.*

Table 5.8: *Water table location, available water content and crop coefficient values for* `<unsat>` *HPMs.*

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 5.9		

Table 5.9: *Elements and attributes for the* `<unsat>` *HPM.*

<Element> or Attribute	**Definition**	**Dimen-sions**	**Variable type**	**Suggested range**	**Default**	**Example**
`<unsat>`	Designates the unsaturated type HPM					
ew	Extractable water (m)	L	Real	0.0 - 1.0	Req.	0.2
kw	Maximum crop coefficient for water	NA	Real	0.5 - 1.0	Req.	1.0
rd	Root zone depth, (m)	L	Real	0.0 - 2.0	Req.	0.5
wilt	Soil water content at wilting point (m)	L	Real	0.0 - 0.1	0.0	0.02
pd	Ponding depth (m) above which open water occurs	L	Real	0.0 - 2.0	Req.	1.0
xthresh	Soil water content when ET ceases (m)	L	Real	0.0 - 0.3	Req.	0.1
pthresh	Soil water content (m) when Kc begins to decrease from Kveg to 0	L	Real	0.0 - 0.5	Req.	0.22
kveg	Vegetation crop coefficient	NA	Real	0.5 - 1.0	Req.	0.75
NA = Not Applicable; Req. = Required.						

Table 5.10: *Example XML for an* `<unsat<` *HPM.*

```
...
    <HPM>
        <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>
    </HPM>
...
```

An example of XML input for an Unsat HPM is presented in benchmark BM18 and in Table 5.10.

5.3 Urban HPMs

The key characteristic of simple urban HPMs is the amount of impervious land that results in greater runoff and reduces the amount of evapotranspiration. The simplest urban HPM 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 HPM which can be calibrated for stormwater detention and routing. The two simple HPMs designed for simulation of urban areas are

1. The `<imperv>` HPM simulates impervious areas with rainfall, ET, surface storage and runoff. There is no recharge to the underlying soil.
2. The `<prr>` HPM 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 it is apportioned to runoff, groundwater recharge, evapotranspiration, and interflow.

5.3.1 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.

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>` HPM is implemented with the elements and attributes shown in Table 5.11 and an example of xml input to implement an `<imperv>` HPM is presented in Table 5.12

Table 5.11: *Elements and attributes for the* `<imperv>` *HPM.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<imperv>`	Designates the `<imperv>` HPM					
dirconn	Specifies directly or indirectly connected	NA	Integer	0=DCIA, 1=UCIA	0	1
isto	Maximum interception storage depth, (m)	L	Real	`0.001 - 0.1`	Req.	0.03
isto1	Initial interception storage depth, (m)	L	Real	`0.001 - 0.1`	0.0	0.02
sdet	Maximum depressional surface storage depth, (m)	L	Real	`0.01 - 0.2`	Req.	0.08
sdet1	Initial depressional surface storage depth, (m)	L	Real	`0.01 - 0.2`	0.0	0.05
NA = Not Applicable; Req. = Required.						

Table 5.12: *Example xml for the* `<imperv>` *HPM.*

```
<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>
```

5.3.2 Precipitation-Runoff Routing HPM`<prr>`

In the model (Figure 5.4), 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.

5.3.2.1 Input Data

The data needed for the `<prr>` HPM are defined in Table 5.13. 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>` HPM is shown in Table 5.14 and in Benchmark 56.

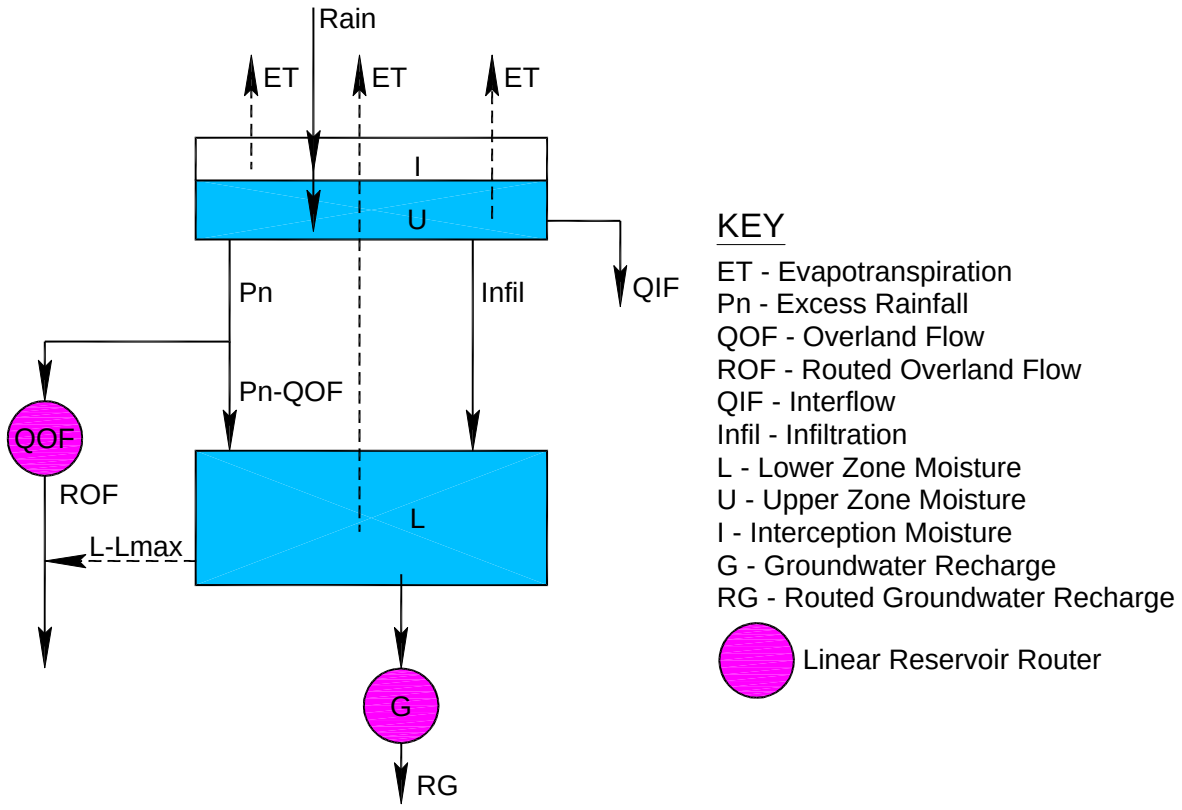


Figure 5.4: *Conceptual diagram of the* `<prr>` *HPM.*

Table 5.13: *Definition of attributes of the* `<prr>` *HPM*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<prr>`	Designates the PRR type HPM					
etcoef	ET crop coefficient	NA	Real	0.0-1.0	Req.	0.8
k0inf	Maximum infiltration rate, m/s	LT^{-1}	Real	0.0-1.0	Req.	0.4
imax	Maximum interception storage, m	L	Real	0.0-1.0	Req.	0.05
umax	Maximum upper zone moisture content, m	L	Real	0.0-1.0	Req.	0.5
lmax	Maximum lower zone moisture content, m	L	Real	0.0-2.0	Req.	1.3
tof	Threshold value of lower zone storage L for overland flow	L	Real	0.0-1.0	0	0.3
tif	Threshold value of lower zone storage L for interflow	L	Real	0.0-1.0	0	0.3
tg	Threshold value of lower zone storage L for groundwater recharge	L	Real	0.0-1.0	0	0.3
cqof	Overland flow runoff coefficient	NA	Real	0.0-1.0	0	0.3
ckol	Time constant for overland flow routing	T	Real	0.0-200	Req.	47.0
ckif	Time constant for interflow from the upper zone storage	T	Real	0.0-2000	Req.	48.3
ckbf	Time constant for groundwater routing	T	Real	0.0-5000	Req.	567.8
NA = Not Applicable; Req. = Required.						

Table 5.14: *Example XML code from benchmark 56 for a* `<prr>` *HPM. The parameters are described in Table 5.13*

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

5.3.2.2 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.

5.4 Complex HPMs

The simple HPMs 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 HPMs 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 HPM types based on areal distribution. The hub also allows for the implementation of additional HPM types that handle local routing.

5.4.1 Water Management Systems <hub>

Although simple urban and agricultural HPMs 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 water management systems. Water management systems can be represented by the use of hubs. Hubs have the capability of linking several HPM 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 Hub is appropriate where the composite HPM 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 5.15. An example xml input showing the creation of a hub with a complex mix of land use types is shown in Table 5.16 and Table 5.17.

Table 5.15: *Elements and attributes for the* `<hub>` *HPM.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<hub>`	Designates the `<hub>` HPM					
wsupply	Source of the water supply for the hub	NA	String	"homecell" "wb-nnn" "well-nnn" where nnn is a water body or well ID	homecell	well-342
sewer	Destination of sewer flow	NA	String	"homecell" "wb-nnn" where nnn is a water-body number	homecell	wb-345654
runoff	Destination of hub runoff	NA	String	"homecell" "wb-nnn" where nnn is a water-body number	homecell	homecell
`<psentry>`	Environment for specification of HPMs in the hub					
psID	HPM ID	NA	Integer	100000-200000	-1	134768
percentarea	Percent of the hub covered by this HPM	NA	Real	0.0 100.0	100	20.0
runoff	Destination of runoff from this HPM	NA	String	"homecell" "hub" "ps-nnn" where nnn is a HPM ID	homecell	hub
wsupply	Source of water supply for this HPM	NA	String	"homecell" "hub" "ps-nnn" where nnn is a HPM ID	homecell	homecell
The HPM for this entry `<afsirs>`, `<imperv>` etc., is specified in this environment.						
NA = Not Applicable; Req. = Required.						

The preferred implementation of HPMs will be the utilization of a small number of simple HPM types applied throughout the domain, on a one HPM 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 specific canal segment, it is necessary to have a unique HPM 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 HPM can be applied to several cells. There are several HPMs that have been developed to work within the hub construct. These HPMs interact with other HPMs to provide a better representation of the urban and agricultural landscape. These HPMs are listed here and described in the following sections.

1. The `<afsirs>` is an agricultural irrigation requirements HPM 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>` HPM 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>` HPM 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>` HPM. Water is input from other HPMs and rainfall and leaves through ET, seepage, and discharge through the outlet structure.

5.4.2 Large Agricultural Developments

5.4.2.1 Agricultural Irrigation Requirement HPM `<afsirs>`

The Afsirs HPM is the primary HPM used to estimate irrigation demand and drainage from agricultural land. The afsirs HPM was an adaptation of the Agricultural Field-Scale Irrigation Requirement System (AFSIRS) model (Smajstrla, 1990). 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.

Table 5.16: *Example XML for typical complex* <hub> *containing native, agricultural and urban HPM 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 5.17: *Example XML for typical complex* `<hub>` *containing native, agricultural and urban HPM 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
   ...
</HPM>
```

The Afsirs model requires parameters to describe the crop, soil, irrigation system and irrigation management plan (Table 5.18). An example of the XML that implements `<afsirs>` HPMs is presented in Table 5.19 and Table 5.20.

Table 5.18: *Elements and attributes used for the* `<afsirs>` *HPM.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<afsirs>`	Designates the `<afsirs>` type HPM					
label	Description of HPM	NA	String	Any String	unspecified	Eric's Farm
coupled	Is water table coupled with water table in mesh cell(Yes) or does afsirs maintain separate soil water accounting	NA	String	Yes or No	no	Yes
trigcat	Trigger Category-Controls the amount of water use reduction during water shortages. (Get exact definition from HPM white paper)	NA	String	To be defined	ag	ag
`<afcrop>`	Designates attributes for the `<afcrop>` subelement					
label	Description of crop	NA	String	Any String	unspecified	Strawberries
id	crop id	NA	Integer	(1-16, perennial crops, 17-44 annual crops)	Req.	23
j1	Crop season start date	NA	string (month-date)	1-1 to 12-31	Req.	10-1
jn	Crop season end date	NA	string (month-date)	1-1 to 12-31	Req.	2-15
depth1	Irrigated soil depth (in)(perennial crops)	L	Real	0.0 - 48.0	Req.	6.0
	Early season irrigated soil depth (in) (annual crops)	L	Real	0.0 - 48.0	Req.	12.0
depth2	Root depth (in) (perennial crops)	L	Real	0.0 - 48.0	Req.	9.0
	Late season irrigated soil depth (in) (annual crops)	L	Real	0.0 - 48.0	Req.	27.0

Table 5.18 continued on next page

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<kctbl>`	Table of 12 monthly crop ET coefficients (perennial crops)	NA	Real	0.0-1.0	No default	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	NA	Real	0.0-1.0	No default	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)	L	Real	0.0-1.0	No default	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)	L	Real	0.0-1.0	No default	0.40 0.40 0.40 0.65
`<afirr>`	Designates the `<afirr>` type subelement					
label	Description of irrigation method	NA	String	Any String	No default	Sprinkler, Large Guns
wtd	Managed water table depth	L	Real	0.0-5.0	Req.	2.5
`<irrmeth>`	Designates the `<irrmeth>` type subelement					
label	Description of the irrigation method	NA	String	Any string	No default	Drip Irr
id	Number for irrigation method type	NA	Integer	See AFSIRS documentation	Req.	6
eff	Irrigation application efficiency	NA	Real	0.0-1.0	Req.	0.7
arzi	Fraction of area of parcel irrigated	NA	Real	0.0-1.0	Req.	0.9
exir	Fraction of crop water use extracted from irrigated soil	NA	Real	0.0-1.0	Req.	0.95

Table 5.18 continued on next page

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
drinc	Flood storage depth for rice (in)	L	Real	0.0-60.0	0.0	3.0
crown	Citrus bed height (in)	L	Real	0.0-36.0	0.0	1.5
`<irrmgmt>`	Designates the `<irrmgmt>` type subelement					
label	Irrigation management scenario	NA	String	Drought or Normal	No default	Normal
trigcode	Designates irrigation trigger	NA	Integer	0 - optimum irrigation 1 - fixed irrigation rate (in/day) 2 - irrigate at fixed soil water deficit	Req.	
value	Fixed irrigation rate (in/day) or fraction of ASW at which irrigation is applied	LT^{-1}	Real	0.0-2.0	0.0	0.2
`<afsoil>`	Designates the `<afsoil>` type subelement					
label	Soil type description	NA	String	Any soil Description	No default	Myakka fine sand
depth	Thickness of soil (in)	L	Real	0.0-200.0	Req.	80.0
minwc	minimum soil water holding capacity (in)	L	Real	0.0-0.5	Req.	0.09
maxwc	maximum soil water holding capacity (in)	L	Real	$\geq minwc - 0.8$	Req.	0.15
cond	condition code for selecting available soil water capacity	NA	Integer	1 – average 2 – minwc 3 – maxwc	Req.	1
NA = Not Applicable; Req. = Required.						

5.4.2.2 Drainage Collector Ditch HPM `<pumpedditch>`

The pumped ditch HPM 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 elements and attributes used in the `<pumpedditch>` HPM are provided in Table 5.21.

Table 5.19: *Example XML for an* `<afsirs>` *HPM.*

```
<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 5.20: *Example XML for an* `<afsirs>` *HPM (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>
</HPM>
```

Table 5.21: *Elements and attributes and typical values used for the* `<pumpedditch>` *HPM as a component of a* `<hub>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<psentry>`	Designates the indexed `<psentry>` environment					
psID	HPM ID	NA	Integer	100000-200000	-1	134768
wsupply	Source of the water supply for the entry	NA	String	"homecell" "wb-nnn" "well-nnn" where nnn is a water body or well ID	homecell	well-342
percentarea	Percent area of farm occupied by collector ditches (%)	NA	Real	0.0-15.0	100	8.5
runoff	destination of runoff	NA	String	Homecell or hub	homecell	Homecell
`<pumpedditch>`	Designates the `<pumpedditch>` type HPM					
rks	Seepage Coefficient (1/day)	NA	Real	$0.001 - 0.1$	Req.	0.025
psize	Maximum Pump discharge rate (in/d)	LT^{-1}	Real	$1.0 - 6.0$	Req.	3.5
ptrig	Depth above the land surface at which the pump is turned on (m)	L	Real	≤ 0	Req.	-1.0
NA = Not Applicable; Req. = Required.						

Table 5.22: *Example xml for* `<pumpedditch>` *HPM.*

```
<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>
</HPM>
```

A simple example of the pumped-ditch HPM is presented in Table 5.22 and another in Benchmark 57.

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.

5.4.2.3 Agricultural Impoundment HPM `<agimp>`

The agricultural impoundment HPM 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. The AgImp HPM 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.

Table 5.23: *Elements, attributes and typical values used for the* `<AgImp>` *HPM as a component of a* `<hub>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<psentry>`	Designates the indexed `<psentry>`environment					
psID	HPM ID	NA	Integer	100000-200000	-1	157843
percentarea	Percent area of parcel in impoundment(%)	NA	Real	5.0-25.0	100	8.5
runoff	destination of runoff	NA	String	Homecell or hub	homecell	hub
`<agimp>`	Designates the `<agimp>`HPM					
rks	Seepage coefficient (1/d)	NA	Real	0.001-0.1	Req.	0.03
height	Wetseason water table elevation (ft -NVGD)	L	Real	0.0-45.0	Req.	23.6
rd	root depth(bottom of storage)	L	Real	0.0-5.0	Req.	2.6
`<stdtriorif>`	Creates the environment for the input of discharge parameters					
r25y3d	Rainfall depth for 25-year, 3-day return period storm (ft)	L	Real	3.0 - 8.0	Req.	5.2
allow	Allowable basin discharge (cfs)	L^3T^{-1}	Real	5.0 - 70.0	Req.	35.0
s	Abstraction in the NRCS runoff method (ft)	L	Real	0.7 - 1.0	Req.	0.85
NA = Not Applicable; Req. = Required.						

Table 5.24: *Typical example xml for an* `<agimp>` *HPM 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>
</HPM>
```

The elements and attributes used in the definition of an `<agimp>` HPM are listed in Table 5.23. A simple example of the `<agimp>` HPM is in Table 5.24.

5.4.3 Urban Hubs `<hub>`

Urban hubs include 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>` HPM, the PA area by `<afsirs>` and the detention area by `<urbandet>`. 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. This is modeled as consumptive use, `<cu>`. Return flow to septic drain fields and sewer systems is also modeled.

5.4.3.1 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.

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.

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 5.25. A simple example of the consumptive use HPM is presented in Table 5.26.

Table 5.25: *Elements and attributes for the* `<cu>` *object.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<cu>`	Designates the `<cu>` environment					
label	High density or low low urban land use	NA	String	HI or LI	undefined	HI
percentarea	Percent area of the hub with the specified consumptive use (%)	NA	Real	0-100	Req.	50.0
wsupply	Water supply source	NA	String	"homecell" or "hub"	homecell	hub
Sub-elements available for flow for `<wsupply>` are `<const>`, `<dss>`, `<rc>`, and `<tsin>`. These are described in detail in section 6.1						
`<sewer>`	Indicates sewer environment					
fracloss	Fraction of water lost from sewer system to groundwater	NA	Real	0.0-1.0	0.0	0.1
`<septic>`	Specifies that the consumptive use goes to recharge, otherwise it goes to sewer flow					

Table 5.26: *Example xml for consumptive use in HPM.*

```
<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>
</HPM>
```

5.4.3.2 Urban Stormwater Retention/Detention HPM `<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. The discharge structure is shown in Figure 5.5. The detention pond is simulated by the HPM `<urbandet>` in a hub. The elements and attributes used to specify an `<urbandet>` HPM in a hub are presented in Table 5.27 and in Benchmark 52.

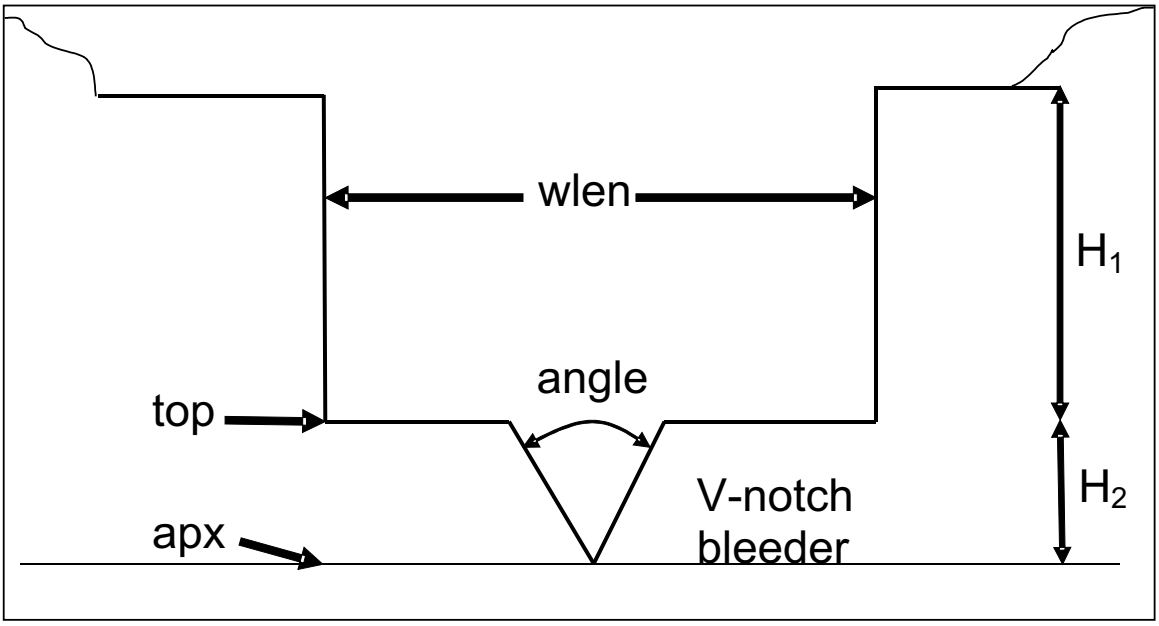


Figure 5.5: *Structure dimensions of the* `<urbandet>` *HPM discharge weir and bleeder.*

Table 5.27: *Elements and attributes for the* `<urbandet>` *HPM as a component of a* `<hub>`.

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<psentry>`	Designates the indexed `<psentry>`environment					
psID	HPM ID	NA	Integer	100000-200000	-1	157843
wsupply	Source of the water supply for the entry	NA	String	"homecell" "wb-nnn" "well-nnn" where nnn is a water body or well ID	homecell	well-342
percentarea	Percent area of parcel in impoundment(%)	NA	Real	5.0-25.0	100	8.5
runoff	Destination of runoff	NA	String	Homecell or hub	homecell	homecell
`<urbandet>`	Designates the `<urbandet>` environment					
rks	Seepage coefficient for flow between the detention basin and the aquifer	NA	Real	0.001-0.1	Req.	0.007
`<vnotchweir>`	Designates the `<vnotch weir>` outlet structure					
wlen	Length of rectangular weir (m)	L	Real	0.0 - 10.0	Req.	7.5
angle	Angle of V-notch weir (degrees)	NA	Real	0 - 120	Req.	75.0
top	Elevation of top of v-notch bleeder and invert of the rectangular weir (m, NGVD)	L	Real	0 - 20	Req.	13.6
apx	Elevation of invert of v-notch bleeder (m, NGVD)	L	Real	0 - 20	Req.	11.4
NA = Not Applicable; Req. = Required.						

Table 5.28: *Example xml for* `<urbandet>` *HPM 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>
</HPM>
```

A simple example of the urbandet HPM is presented in Table 5.28.

Table 5.29: *Example index file for assigning HPMs to mesh cells.*

```
DATASET
OBJTYPE "mesh2d"
BEGSCL
ND 18
NAME "landuse type"
TS 0 0.0
1
1
1
2
2
2
3
3
...
...
...
```

5.4.4 Assignment Of HPMs To Various Land Use Types `<indexed>`

Each cell in the model is assigned a particular type of HPM depending on the land use type. Cell ID's are used to assign cells or fractions of cells to the HPMs as needed. An index is assigned for each HPM type, and an index file is used to associate the cell ID's with the HPM types. An indexed entry element `<indexed>` can be used where multiple HPMs must be defined. To assign various HPM 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 HPM type indices. The example in Table 5.29 below shows one of these index files used in Benchmark 14.

Table 5.30: *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>
</HPM>
```

5.4.5 Time Variation of HPM Parameters <ampmod>

Particularly for agricultural HPMs it is appropriate to vary parameters with the season of the year. As an example, the `<afsirs>` HPM 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>` HPMs. 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 5.30, 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 HPM. 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 6

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.

6.1 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>` format.

6.1.1 Constant Value

A constant value can be specified with a very simple XML construction. The elements and attributes available are shown in Table 6.1

Table 6.1: *Elements and attributes used to define a* `<const>` *input value.*

<Element> or Attribute	**Definition**	**Dimensions**	**Variable type**	**Suggested range**	**Default**	**Example**
`<const>`	Designates a `<const>` input.					
dbintl	The time interval of the simulation in minutes.	T	Real	60-2880	1440	1440
value	Value of the input variable.	NA	Real	Any real	Req.	1.34
mult	Multiplier for the value. May be used to change units.	NA	Real	Any real	1.0	0.3048
units	Units of the input.	NA	String	Any string	No default	
type	Valid DSS data type.	NA	String	Valid DSS data type	No default	
NA = Not Applicable; Req. = Required.						

Table 6.2: *Sample XML for specifying a constant* `<refet>`.

```
...
  <refet>
     <const dbintl="1440" value="0.14" mult="0.3048" </const>
  </refet>
...
```

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 6.2

6.1.2 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, 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 6.3. An example showing the creation and use of a rule curve shown in Table 6.4 is used in Benchmark 52.

Table 6.3: *Elements and attributes used to define a rule curve and to use it.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	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	NA	Integer	any integer	Req.	3
label	A label to describe the rule curve.	NA	String	Any string	Uns.	seasonal water level
xunits	Units for the first column (time) in a 1D lookup table.	NA	String	A valid DSS time interval	Req.	1DAY
cycle	The time length of the rule curve.	NA	String	A valid DSS time interval	Req.	1YEAR
yunits	Units of the variable.	NA	String	Any string	Req.	1YEAR
type	The type of data.	NA	String	INST-VAL PER-AVER PER-CUM	Req.	INST-VAL
A table of x and y values in two columns to define the rule curve. See the example in Table 6.4						
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.	NA	Integer	Any rule curve id	Req.	3
NA = Not Applicable; Req. = Required; Uns. = Unspecified.						

6.1.3 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 6.7. 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 6.5. An example of a DSS path name is shown below.

```
      A           B       C        D     E   F
 /RED RIVER/BEND MARINA/FLOW/01JAN1975/1DAY/OBS/
```

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//
```

Table 6.4: *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 6.5: *Elements and attributes used for specifying time series data in a* `<dss>` *file.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
`<dss>`	Designates a `<dss>` file.					
file	The name of the DSS file	NA	String	Any valid DSS file name	Req.	RRFlow.dss
pn	The DSS path name of the times series in the DSS file	NA	String	A valid DSS path name	No Default	/REDRIVER/BEND MARINA/FLOW/ 01JAN1975/1DAY/OBS/
mult	A multiplier for the data	NA	Real	Any Real	1.0	0.02831685
units	Units of the variable	NA	string	Any string	No Default	cfs
type	Type of data	NA	String	See Table 6.8	No Default	PER-CUM
dbintl	Interval in minutes between data points	T	Integer	1-525600	1440	
NA = Not Applicable; Req. = Required						

Table 6.6: *Elements and attributes used to define a TSnode and to use it.*

<Element> or Attribute	Definition	Dimensions	Variable type	Suggested range	Default	Example
Element and attribute for defining tsNodes.						
`<tsNodes>`	Designates tsNodes will be defined.					
`<tsentry>`	Designates that a particular time series node will be specified.					
id	The ID number of the tsNode	NA	Integer	any integer	Req.	3
label	A label to describe the tsNode.	NA	String	Any string	Uns.	Daily Pumping Rate
quantity	The type of data in the time series	NA	String	depth, stage, flow, volume, storage, depthfraction, transmissivity, nondimensional	Req.	storage
The data may be input under the elements `<const>`, `<dss>`, or `<rc>` as described in this chapter.						
Element and attribute for applying a tsNode.						
`<tsin>`	Designates the data specified by a tsNode.					
id	The id number of the tsNode.	NA	Integer	Any tsNode id	Req.	7
NA = Not Applicable; Req. = Required; Uns. = Unspecified.						

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 6.8.

Table 6.7: *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 6.8: *Default units used by the RSM model.*

Quantity	Unit	Type
Head	METERS	INST-VAL
Flow	CU_METER/SEC	INST-VAL
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
Definition of Unit Types		
Type	Definition	Example
PER-AVER	Period Average	Daily flow
PER-CUM	Period Cumulative	Monthly flow (volume)
INST-VAL	Instantaneous	Breakpoint Stage
INST-CUM	Instantaneous Cumulative	Rain mass curve

Chapter 7

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 7.1). A local and global water balance discussion is also provided in this chapter.
2. RSM output options (section 7.2) specified in the XML input files
3. RSM uncertainty analysis (section 7.3)
4. RSM graphical user interface (section 7.4 currently under development)

7.1 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.

$$\nabla \mathbf{A}.\frac{\partial \mathbf{H}}{\partial t} = \mathbf{Q}(\mathbf{H}) + \mathbf{S} \tag{7.1}$$

where $\mathbf{Q}(\mathbf{H})$ = 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 HPM 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. HPMs represent the local hydrology which may account for water above the water table. Water in the HPM 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.

$$V_s^{(n+1)} + V_p^{(n+1)} - V_s^{(n)} - V_p^{(n)} = Q_r + Q_d + Q_s + Q_i + Q_b \tag{7.2}$$

The components are defined in Table 7.1 and Table 7.2.

7.1.1 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 HPM. Only the saturated water is related to the head in the water body. The reported components of water are listed in Table 7.1.

7.1.2 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 7.1: *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 7.2: *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 HPM compartment as a result of the internal processes of the HPM. 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 HPMs. 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.

7.1.3 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.

7.2 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 under <globalmonitor>. This output is in GMS, netcdf, or dss format. The GMS file 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, dss, or GMS export is possible. The elements included are <budget>, <budgetpackage>, psbudgetpackage¿, and <cellreport>. The netcdf 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 of cells, pseudocells, segments, boundary conditions or other model components 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 7.3. The output file formats available under the various output options are listed in Table 7.4. Details on some of the most important options are described in the Tables that follow, with examples.

Table 7.3: *Model output options available using* `<output>`.

Element or Attribute	Description	Available Options	Default	Output File Format
`<globalmonitor>`	Used to dump all the values of a variable for the entire duration of the model run.			`<netcdf>` `<gms>` `<dss>`
`attr`	Name of the attribute to be reported.	See Table 7.5	Req	
`para`	Type of watermover to be monitored.	defwm	unspecified	
		cellbc		
		segmentbc		
`<budget>`	Budget of inflows/outflows to waterbodies.			
`file`	Name of the file to write data to.	Any file name	Req	
`<budgetpackage>`	The entire water budget is written to a netcdf file for post processing.			
`file`	Name of the file to write data to.	Any file name	Req	
`dbintl`	Interval in minutes to write data.	Standard intervals	-1	
`<psbudgetpackage>`	All necessary components to do a pseudocell budget water are written to a netcdf file.			
`file`	Name of the file to write data to.	Any file name	Req	
`dbintl`	Interval in minutes to write data.	Standard intervals	-1	
`<cellreport>`	Output description needs to be written. ??			
`file`	Name of the file to write data to.	Any file name	Req	
`<cellmonitor>`	Writes cell attributes to output file.			`<dss>` `<asciiform>` `<csv>`

Table 7.3 continued on next page

Element or Attribute	Description	Available Options	Default	Output File Format
id	Cell id	Valid cell id	Req	
attr	Attribute to save to file.	See Table 7.6	Req	
label	Description of the data.	Any string	No default	
<segmentmonitor>	Writes canal segment attributes to output file.			<dss> <asciiform> <csv>
id	Segment id	Valid segment id	Req	
attr	Attribute to be monitored.	See Table 7.7	Req	
label	Description of the data.	Any string	No default	
<junctionmonitor>	Writes information about flow at canal segment junctions to output file.			<dss> <asciiform> <csv>
id1	Upstream segment id	Valid segment id	Req	
id2	Downstream segment id	Valid segment id	Req	
attr	Attribute to be monitored.	See Table 7.8	Req	
<wmmonitor>	Writes information about flow at canal segment junctions to output file.			<dss> <asciiform> <csv>
wmID	Water Mover ID	Valid water mover id	Req	
attr	Attribute to be monitored	See Table 7.9	Req	
label	Description of the output.	Any string	No default	
<bcmonitor>	Writes information about boundary conditions to output file.			<dss> <asciiform> <csv>

Table 7.3 continued on next page

Element or Attribute	Description	Available Options	Default	Output File Format
bcID	Boundary condition ID	Valid boundary condition id	Req	
attr	Attribute to be monitored	See Table 7.10	Req	
<lakemonitor>	Writes information about lakes to output file.			<dss> <asciiform> <csv>
id	Lake ID	Valid lake id	Req	
attr	Attribute to be monitored	See Table 7.11	Req	
label	Description of the output.	Any string	No default	
<lkstressmonitor>	Needs description - not in DTD.			<dss> <asciiform> <csv>
id	Lake ID	Valid lake id	Req	
attr	Attribute to be monitored	See Table 7.12	Req	
<assessormonitor>	Used to monitor assessors.			<dss> <asciiform> <csv>
asmtID	Assessor ID	Valid assessor id	Req	
attr	Attribute to be monitored	See Table 7.13.	Req	
name	Needs a description??	????	No default	
label	Description of the data.	Any string	No default	
<ctrlmonitor>	Used to monitor attributes of controllers.			<dss> <asciiform> <csv>

Table 7.3 continued on next page

Element or Attribute	Description	Available Options	Default	Output File Format
cID	Controller ID	Valid controller id	No Default	
attr	Attribute to be monitored	See Table 7.14.	Req	
montype	Monitor Type ???	scalar or vector	scalar	
var	Only for montype = vector	??	No default	
<flowgage>	Used to monitor flow across flow lines.			<dss> <asciiform> <csv>
section	Specifies whether groundwater flow, overland flow, or both are monitored.	See Table 7.15	Req	
label	Description of the flow	Any string	No default	
<nodelist>	A list of nodes defining the flow line for <flowgage>. See section for details	A list of valid node numbers	1 3 6 8	
<pseudomonitor>	Used to write out attributes within an HPM.			<dss> <asciiform> <csv>
id	Pseudocell ID	Valid pseudocell id	Req	
attr	Attribute to be monitored	See Table 7.16	Req	
label	Description of the data	Any string	No default	
<reachmonitor>	Useful in monitoring reaches.			<dss> <asciiform> <csv>

Table 7.3 continued on next page

Element or Attribute	Description	Available Options	Default	Output File Format
id	Pseudocell ID	Valid pseudocell id	Req	
attr	Attribute to be monitored	See Table 7.17	Req	
<segmentlist>	A list of segments to define the reach for <reachmonitor> is the next text	A list of valid segment numbers.	4 5 6 7	
<wcumonitor>	Needs description.			<dss> <asciiform> <csv>
ntwkID	Network ID??	A valid network ID??	Req	
name	??	??	Req	
attr	Attribute to be monitored	See Table 7.18	Req	
label	Descriptive label	Any string	No Default	
<spvrmonitor>	Not yet implemented.			
spvrID	Supervisor ID	A valid supervisor ID	Req	
wmID	Water mover ID	A valid water mover ID	Req	
attr	Attribute to be monitored	See Table 7.19	Req	
montype	Monitor Type ???	scalar or vector	scalar	
var	Only for montype = vector	????	No default	
<flashmonitor>	Not yet implemented, but in DTD.			
<statmonitor>	Same as <statemonitor>			<dss> <asciiform> <csv>

Table 7.3 continued on next page

Element or Attribute	Description	Available Options	Default	Output File Format
asmtID	??		Req	
name	??	??	No default	
attr	Attribute to be monitored	See Table 7.20	No default	
label	Descriptive label	Any string	No Default	
`<statemonitor>`	Same as statmonitor??.			`<dss>` `<asciiform>` `<csv>`
asmtID	??		Req	
name	??	??	No default	
attr	Attribute to be monitored	See Table 7.20	No default	
asmtID	??		Req	
`<trmonitor>`	Needs description??.			`<dss>` `<asciiform>` `<csv>`
name	??	??	Req	
attr	Attribute to be monitored	See Table 7.21	No default	
label	Descriptive label	Any string	No Default	
`<CBmonitor>`	Not yet implemented, but in DTD.			
`<ormNodemonitor>`	Not yet implemented but in DTD.			
name	??	??	No default	
attr	Attribute to be monitored	See **??**	No default	
label	Descriptive label	Any string	No Default	
`<basinmonitor>`	Need description.			`<dss>` `<asciiform>` `<csv>`

Table 7.3 continued on next page

Element or Attribute	Description	Available Options	Default	Output File Format
id	??		Req	
attr	Attribute to be monitored	See Table 7.22	Req	
`<trigzonemonitor>`	Need description.			`<dss>` `<asciiform>` `<csv>`
zoneID	??		Req	
trigcat	??		unspecified	
attr	Attribute to be monitored	See Table 7.23	Req	
`<rcmonitor>`	Need description			`<dss>` `<asciiform>` `<csv>`
id	rule curve id	Any valid rule curve id	Req	
`<const>`	Need description - Not in DTD			
`<tsmonitor>`	Need description.			`<dss>` `<asciiform>` `<csv>`
id	time series id	a valid time series id	Req	
label	Descriptive label	Any string	unspecified	
`<tkprmonitor>`	Need description. - Not in DTD			
attr	Attribute to be monitored	See Table 7.24	Req	
`<filter>`	Need description.			`<dss>` `<asciiform>` `<csv>`

Table 7.3 continued on next page

Element or Attribute	**Description**	**Available Options**	**Default**	**Output File Format**
label	Descriptive label	Any string	unspecified	
type			mult	
mult			1.0	
offset			0.0	
numAvg			0	
numSum			0	
order			0	

7.2.1 Saving Model Output <globalmonitor>

The head, velocity and a number of other variables (Table 7.5) in the entire domain for the entire duration can be output into a <dss> or <netcdf> format or an ASCII file in GMS format using the <globalmonitor> option. Table 7.4 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.

Table 7.5: *Attributes available with* <globalmonitor>.

Table 7.4: *Time series formats available within the output options in Table 7.3.*

Element or Attribute	Description	Default
`<gms>`	The output will be written to a gms file.	
`file`	Name of the output file.	Req.
`<netcdf>`	The output will be written to a netcdf file.	
`file`	Name of the output file.	Req.
`dbintl`	Interval in minutes at which output will be written.	-1
`<dss>`	The output will be written to a dss file.	
`file`	Name of the output file.	Req.
`pn`	dss path name to identify the output.	No default
`dbintl`	Interval in minutes at which output will be written.	No Default
`<csv>`	The output will be written to a file in comma delimited format.	
`file`	Name of the output file.	Req.
`dbintl`	Interval in minutes at which output will be written.	1440
`label`	Description of the output data.	Req.
`<asciiform>`	The output will be written to an ASCII formatted file.	
`file`	Name of the output file.	Req.
`format`	The format of the output data. Any c-style formats are allowable, but floating point formats are ormally used as %lg, %10.2lf, %f	%lf

Attribute	Description	Dimensions
`ponding`	Ponding depth (Head - Land Surface Elevation).	L
`wtdepth`	Depth of head below land surface.	L
`head`	Water head in cells	L
`recharge`	Recharge into cells	L
`runoff`	Runoff from cells	L
`wsupply`	Water supply in cells	L
`rain`	Rain in cells	L
`refet`	Reference ET in cells	L
`et`	Computed ET in cells	L
`etvol`	Computed ET vol in cells	L^3
`rainvol`	Rainfall volume in cells	L^3
`rchgvol`	Recharge volume in cells	L^3
`watercontent`	Water content in cells	L
`wcvol`	Water content in cells	L^3
`inflow`	Inflow into cells	L
`waterlevel`	Water level in cells	L
`sy`	Storage coefficient	NA
`transmissivity`	Transmissivity	L^2T^{-1}
ps_et	depth of evapotranspiration in HPM	L
ps_etvol	volume of evapotranspiration	L^3
ps_rechargevol	volume of recharge to homecell	L^3
ps_seepagevol	volume of seepage	L^3
ps_cuvol	HPM consumptive use volume	L^3
ps_septicvol	volume of consumptive use return flow – septic	L^3
ps_sewervol	volume of consumptive use return flow – sewer	L^3
ps_rchgvol	In ProcessGlobalMonitors, but not implemented	L^3
ps_celldeltavol	volume of inflow to HPM	L^3

Table 7.5 continued on next page

Attribute	**Description**	**Dimensions**
ps_rain	depth of rainfall	L
ps_refet	depth of Reference ET	L
ps_rainvol	volume of rainfall	L^3
ps_watercontent	depth of storage	L
ps_wcvol	Pvolume of storage	L^3
ps_wsupply	Pdepth of water imported for water supply	L
ps_wsupplyvol	Volume of water imported for water supply	L^3
ps_wslocal	depth of water withdrawn from homecell for water supply	L
ps_wsimport	volume of water imported for water supply	L^3
ps_runoff	depth of runoff	L
ps_runoffvol	volume of runoff	L^3
ps_runofflocal		L^3
ps_runoffexport		L^3
ps_initvol	initial volume of water in HPM	L^3
topo	cell land surface elevation	L
initvol	initial volume of water in cells	L^3
swflow	cell land surface elevation	L
initvol	initial volume of water in cells	L^3
swflow	Volume of surface water flow through walls	L^3
gwflow	Volume of groundwater flow through walls	L^3
totalvector	need description??	L^3
gwvector	need description??	L^3
swvector	need description??	L^3
segmenthead	head in canal segments	L
segmentdepth	depth of water in canal segment	L
sbflow	sum of the flow rate of all streambank flow in segment	L^3T^{-1}
seepageflow	streambank flow rate – aquifer seepage	L^3T^{-1}
overbankflow	streambank flow rate – overbank flow	L^3T^{-1}
lev1flow	streambank flow rate – type 1 levee seepage	L^3T^{-1}
lev2flow	streambank flow rate – type 2 levee seepage	L^3T^{-1}
segsto	volume of storage in segment	L^3
initsegsto	initial volume of storage in segment	L^3
segwsupply	Water supply volume delivered by segment ??	L^3
segrunoff	Runoff volume received by segment	L^3
junctionvolume	Volume of flow through junction	L^3

Attribute	**Description**	**Dimensions**
cellbc	Volume of cell boundary condition flow	L^3
segmentbc	Volume of segment boundary condition flow	L^3
lakebc	Volume of lake boundary condition flow	L^3
lakeERS	Volume of lake recharge through surface (rain and ET)	L^3
basinsto	Volume of storage in basin	L^3
initbasinsto	initial volume of storage in basin	L^3
basinrecharge	volume of recharge in basin	L^3
basinrunoff	volume of runoff into basin	L^3
basinwsupply	volume of water supply from basin	L^3
lakerecharge	volume of recharge to lake	L^3
lakerunoff	volume of runoff to lake	L^3
lakewsupply	volume of water supply from lake	L^3
borrowvolume	to be defined	L^3
lakewm	to be defined	L^3
lakesto	volume of water stored in lake	L^3
initlakesto	volume of water stored in lake at beginning of simulation	L^3
defwm	Flow volume in defined water movers	L^3
verticalmover	Flow volume in vertical water movers	L^3
verticaltransfer	To be defined	L^3

Table 7.6: *Variables that can be monitored using* `<cellmonitor>`*. The usage is: <cellmonitor id="cellid#" attr="attribute"> <filetype in Table 7.4 > < /cellmonitor>*

Cell Monitor

Attribute	Name	Description	Dimensions
`ponding`	PondDepth	depth of water above land surface	L
`wtdepth`	WTDepth	depth to water table	L
`head`	ComputedHead	elevation of water table	L
`recharge`	Recharge	volume of recharge received from HPM	L^3
`runoff`	RunoffVolume	volume of runoff received from HPM	L^3
`wsupply`	WSupplyVolume	volume of water supply withdrawals from cell	L^3
`rain`	Rainfall	depth of rainfall on cell	L
`rainvol`	RainfallVolume	volume of rainfall on cell	L^3
`rchgvol`	RechargeVolume	volume of recharge received from HPM	L^3
`watercontent`	WaterContent	Saturated water content ion depth	L
`initvol`	InitVol	volume of initial water content in cell	L^3
`wcvol`	WCVolume	volume of saturated water content in cell	L^3
`topo`	Topography	land surface elevation	L
`inflow`	Inflow	volume of inflow from adjacent water bodies in previous time step	L
`waterlevel`	WaterLevel	head - land surface	L
`sy`	SpecificYield	specific yield (depth fraction)	NA
`transmissivity`	Transmissivity	aquifer transmissivity	L^2T^{-1}
`routedrunoff`	RoutedRunoffVolume	Volume of routed runoff from the HPM	L^3
`refet`	RefET	Reference evapotranspiration for the cell	L
`et`	ET	Computed ET	L
`etvol`	EtVolume	Calculated volume of ET from the HPM	L^3

Table 7.7: *Variables that can be monitored using* `<segmentmonitor>`.

Segment Monitor

Attribute	Name	Description	Dimensions
head	segmenthead	elevation of water level	L
segmenthead	segmenthead	elevation of water level	L
depth	segmentdepth	depth of water below land surface	L
segmentdepth	segmentdepth	depth of water below land surface	L
sbflow	sbflow	sum of the flow rate of all streambank flow in segment	L^3T^{-1}
seepageflow	seepageflow	streambank flow rate – aquifer seepage	L^3T^{-1}
overbankflow	overbankflow	streambank flow rate – overbank flow	L^3T^{-1}
lev1flow	lev1flow	streambank flow rate – type 1 levee seepage	L^3T^{-1}
lev2flow	lev2flow	streambank flow rate – type 2 levee seepage	L^3T^{-1}
sbvolume	sbvolume	sum of all streambank flow volumes	L^3
seepagevolume	sbvolume	streambank flow volume – aquifer seepage	L^3
overbankvolume	sbvolume	streambank flow volume – overbank flow	L^3
lev1volume	sbvolume	streambank flow volume – type 1 levee seepage	L^3
lev2volume	sbvolume	streambank flow volume – type 2 levee seepage	L^3
initsegsto	InitSegSto	initial volume of storage	L^3
segsto	SegmentStorage	volume of storage	L^3
segwsupply	SegmentWSupply	Water supply volume delivered by segment ??	L^3
wsupply	SegmentWSupply	Water supply volume delivered by segment ??	L^3
segrunoff	SegmentRunoff	Runoff volume received by segment	L^3
runoff	SegmentRunoff	Runoff volume received by segment	L^3
freestor	SegmentFreeStor	Volume of storage between the water surface and the top of the canal bank.	L^3

Table 7.8: *Variables that can be monitored using* `<junctionmonitor>`.

Network Junction Monitor

Attribute	Name	Description	Dimensions
flow	JunctionFlow	Flow rate through junction	L^3T^{-1}

Table 7.9: *Variables that can be monitored using* `<wmmonitor>`.

Watermover Monitor

Attribute	Name	Description	Dimensions
`flow`	WaterMoverFlow	Flow rate through water mover	L^3T^{-1}
`volume`	WaterMoverFlow	Flow volume through water mover	L^3
`control`	WaterMoverControl	The output value of the water mover controller.	NA
`maxflow`	WaterMoverMaxFlow	Flow rate through water mover with controller output = 1.0 ????	L^3T^{-1}

Table 7.10: *Variables that can be monitored using* `<bcmonitor>`.

Boundary Condition Monitor

Attribute	Name	Description	Dimensions
`flow`	BCVolume	Flow rate through boundary	L^3T^{-1}
`head`	BCHead	Head applied at boundary	L

Table 7.11: *Variables that can be monitored using* `<lakemonitor>`.

Lake Monitor

Attribute	Name	Description	Dimensions
lakesto	LakeStorage	volume of lake storage	L^3
initlakesto	InitLakeSto	initial volume of lake storage	L^3
head	LakeHead	elevation of water level	L
lakerecharge	LakeRecharge	Volume of recharge from HPMs	L^3
lakerunoff	LakeRunoff	Volume of runoff from HPMs	L^3
lakewsupply	LakeWSupply	Water supply volume taken form lake??	L^3

Table 7.12: *Variables that can be monitored using* `<lkstressmonitor>`.

Segment Monitor

Attribute	Name	Description	Dimensions
stressorvol	LakeStressorVolume	volume of net (rain - ET) to the lake	L^3

Table 7.13: *Variables that can be monitored using* `<assessormonitor>`.

Assessor Monitor

Attribute	Name	Description	Dimensions
`depth`	Depth	depth of water in waterbody	L
`volume`	Volume	volume stored in waterbody	L^3
`flow`	Flow	Flow rate into waterbody	L^3T^{-1}

```
..
  <output>
    <globalmonitor attr="totalvector">
       <gms file="outvect.dat"> </gms>
    </globalmonitor>
    <globalmonitor attr="head">
       <gms file="outheads.dat"> </gms>
    </globalmonitor>
  </output>
```

Table 7.14: *Variables that can be monitored using* `<ctrlmonitor>`.

Control Monitor

Attribute	Name	Description	Dimensions
error	ControlError	error	NA
control	ControlOutput	output	NA
state	ControlState	state variable	NA
maxflow	ControlMaxFlow	Flow rate with controller output = 1.0	L^3T^{-1}

Table 7.15: *Variables that can be monitored using* `<flowgage>`.

Flowline Monitor

Attribute	Name	Description	Dimensions
ol	OverlandFlow	volume of overland flow across defined flowline	L^3
gw	GroundwaterFlow	volume of groundwater flow across defined flowline	L^3
ol_gw	TotalFlow	volume of total flow across defined flowline	L^3

7.2.2 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 `<cellreport>`, `<budget>`, `<budgetpackage>` or `<psbudgetpackage>` as shown in Table 7.3. These tags will create the netCDF files at the requested time interval in minutes specified 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>
```

where 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

Table 7.16: *Variables that can be monitored using* `<pseudomonitor>`.

Attribute	Name	Description	Dimensions
ps_celldeltavol	PS_CellDeltaVolume	volume of inflow to HPM	L^3
ps_rechargevol	PS_RechargeVolume	volume of recharge to homecell	L^3
ps_et	PS_Et	depth of evapotranspiration	L
ps_rain	PS_Rain	depth of rainfall	L
ps_refet	PS_RefET	depth of Reference ET	L
ps_etvol	PS_EtVolume	volume of evapotranspiration	L^3
ps_rainvol	PS_RainVolume	volume of rainfall	L^3
ps_watercontent	PS_WaterContent	depth of storage	L
ps_wcvol	PS_WCVolume	volume of storage	L^3
ps_wsupply	PS_WaterSupply	depth of water imported for water supply	L
ps_wslocal	PS_WSLocal	depth of water withdrawn from homecell for water supply	L
ps_wsimport	PS_WSImportVolume	volume of water imported for water supply	L^3
ps_runoff	PS_Runoff	depth of runoff	L
ps_runoffvol	PS_RunoffVolume	volume of runoff	L^3
ps_runoffexport	PS_RunoffExport	volume of runoff exported	L^3
ps_seepagevol	PS_SeepageVolume	volume of seepage	L^3
ps_wsupplyvol	PS_WSupplyVolume	volume of water imported for water supply	L^3
ps_cuvol	PS_CUVolume	volume of import consumptive use	L^3
ps_sewervol	PS_SewerVolume	volume of consumptive use return flow – sewer	L^3
ps_septicvo	PS_SepticVolume	volume of consumptive use return flow – septic	L^3
ps_wslocalvol	PS_WSLocalVolume	volume of water withdrawn from homecell for water supply	L^3
ps_initvol	PS_InitVol	initial volume of storage in HPM	L^3
ps_runofflocal	PS_RunoffLocal		L^3
ps_runoffroute	PS_RunoffRoutedVolume		L^3
ps_wsupplyroute	ps_wsupplyroute		L^3
ps_curoute	PS_CURoutedVolume		L^3
ps_sewerroute	PS_SewerRoutedVolume		L^3

Table 7.17: *Variables that can be monitored using* `<reachmonitor>`.

Segment Monitor

Attribute	Name	Description	Dimensions
head	segmenthead	elevation of water level	L
segmenthead	segmenthead	elevation of water level	L
depth	segmentdepth	depth of water below land surface	L
segmentdepth	segmentdepth	depth of water below land surface	L
sbflow	sbflow	sum of the flow rate of all streambank flow in segment	L^3T^{-1}
seepageflow	seepageflow	streambank flow rate – aquifer seepage	L^3T^{-1}
overbankflow	overbankflow	streambank flow rate – overbank flow	L^3T^{-1}
lev1flow	lev1flow	streambank flow rate – type 1 levee seepage	L^3T^{-1}
lev2flow	lev2flow	streambank flow rate – type 2 levee seepage	L^3T^{-1}
sbvolume	sbvolume	sum of all streambank flow volumes	L^3
seepagevolume	sbvolume	streambank flow volume – aquifer seepage	L^3
overbankvolume	sbvolume	streambank flow volume – overbank flow	L^3
lev1volume	sbvolume	streambank flow volume – type 1 levee seepage	L^3
lev2volume	sbvolume	streambank flow volume – type 2 levee seepage	L^3
initsegsto	InitSegSto	initial volume of storage	L^3
segsto	SegmentStorage	volume of storage	L^3
segwsupply	SegmentWSupply	Water supply volume received by segment ??	L^3
wsupply	SegmentWSupply	Water supply volume received by segment ??	L^3
segrunoff	SegmentRunoff	Runoff volume received by segment	L^3
runoff	SegmentRunoff	Runoff volume received by segment	L^3
freestor	SegmentFreeStor	Volume of storage between the water surface and the top of the canal bank.	L^3

Table 7.18: *Variables that can be monitored using* `<wcumonitor>`.

Segment Monitor

Attribute	Name	Description	Dimensions
head	segmenthead	elevation of water level	L
segmenthead	segmenthead	elevation of water level	L
depth	segmentdepth	depth of water below land surface	L
segmentdepth	segmentdepth	depth of water below land surface	L
sbflow	sbflow	sum of the flow rate of all streambank flow in segment	L^3T^{-1}
seepageflow	seepageflow	streambank flow rate – aquifer seepage	L^3T^{-1}
overbankflow	overbankflow	streambank flow rate – overbank flow	L^3T^{-1}
lev1flow	lev1flow	streambank flow rate – type 1 levee seepage	L^3T^{-1}
lev2flow	lev2flow	streambank flow rate – type 2 levee seepage	L^3T^{-1}
sbvolume	sbvolume	sum of all streambank flow volumes	L^3
seepagevolume	sbvolume	streambank flow volume – aquifer seepage	L^3
overbankvolume	sbvolume	streambank flow volume – overbank flow	L^3
lev1volume	sbvolume	streambank flow volume – type 1 levee seepage	L^3
lev2volume	sbvolume	streambank flow volume – type 2 levee seepage	L^3
initsegsto	InitSegSto	initial volume of storage	L^3
segsto	SegmentStorage	volume of storage	L^3
segwsupply	SegmentWSupply	Water supply volume delivered by segment ??	L^3
wsupply	SegmentWSupply	Water supply volume delivered by segment ??	L^3
segrunoff	SegmentRunoff	Runoff volume received by segment	L^3
runoff	SegmentRunoff	Runoff volume received by segment	L^3
freestor	SegmentFreeStor	Volume of storage between the water surface and the top of the canal bank.	L^3

Table 7.19: *Variables that can be monitored using* `<spvrmonitor>`.

Time Keeper Monitor

Attribute	Name	Description	Dimensions
output	SupervisorOutput	To be defined ???	NA
state	SupervisorState	To be defined ???	NA

Table 7.20: *Variables that can be monitored using* `<statmonitor> or <statemonitor>`.

State Monitor

Attribute	Name	Description	Dimensions
flow	flow	Rate of flow in ????	L^3T^{-1}
head	head	Head in ???	L

Table 7.21: *Variables that can be monitored using* `<trmonitor>`.

State Monitor

Attribute	Name	Description	Dimensions
flow	flow	Rate of flow in ????	L^3T^{-1}
wsneed	wsneed	water supply need assigned to structure???	L^3
fcneed	fcneed	flood control need assigned to structure ???	L^3
wsrelease	wsrelease	water supply release ???	L^3
fcrelease	fcrelease	flood control release ???	L^3
envneed	wsrelease (Misnamed ???)	environmental need assigned to structure ???	L^3
envrelease	fcrelease (Misnamed ???)	environmental release ???	L^3

Table 7.22: *Variables that can be monitored using* `<basinmonitor>`.

Segment Monitor

Attribute	**Name**	**Description**	**Dimensions**
basinsto	BasinStorage	Volume of water in the basin	L^3
initbasinsto	InitBasinSto	Volume of water in the basin at the beginning of the simulation	L^3
head	BasinHead	Head in the basin	L
basinrecharge	BasinRecharge	Volume of recharge to the basin	L^3
basinrunoff	BasinRunoff	Volume of runoff to the basin	L^3
basinswsupply	BasinWSupply	Volume of water supply extracted from the basin??	L^3

Table 7.23: *Variables that can be monitored using* `<trigzonemonitor>`.

Segment Monitor

Attribute	**Name**	**Description**	**Dimensions**
phase	TriggerPhase	Volume of water in the basin	L^3
cutback	TriggerCutback	Volume of water in the basin at the beginning of the simulation	L^3

Table 7.24: *Variables that can be monitored using* `<tkprmonitor>`.

Time Keeper Monitor

Attribute	**Name**	**Description**	**Dimensions**
julday		Julian day	T
month		month of simulation??	T
day		day of simulation ???	T
year		year of simulation ???	T
minute		minute of simulation ???	T
interval	Model time step ???	T	

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.

7.2.3 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 7.3. The attributes available for monitoring using these elements are described in the tables listed in Table 7.3. 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>
```

7.3 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, 1998) 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, 2000b) 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.

7.3.1 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.

7.3.2 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.

7.3.3 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 are assumed as a standard data set against which all runs are compared.

7.4 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.

7.4.1 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

7.4.2 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

Bibliography

Brater, E. F., H. W. King, J. E. Lindell, and C. Y. Wei (1996, March 01). *Handbook of Hydraulics*. McGraw-Hill Professional.

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.

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. (1998, 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. (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). 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.

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.

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, 1998) 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 HPM 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 HPMs. Lal added structures, overland, groundwater, canal and lake interactions, various test cases, and certain types of HPMs. 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 are 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 are 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 are 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 (see subsubsection B.3.2.1) demonstrates a successful validation, which means that all of the XML data are of the proper data type and all required data are present. The second case (see subsubsection B.3.2.2) 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 (see subsubsection B.3.2.3) discusses the validation of model input data when more than one XML input file exists.

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]

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.

[6] http://www.w3.org/2001/03/webdata/xsv

```
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>
```

[7] http://www.w3.org/TR/xinclude/

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.

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 are 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 HPMs SYSTEM "pseudo.xml">`, and referred to later when necessary using `&HPMs;`. The file `pseudo.xml` should be placed within the directory. An example of this has been shown above in subsubsection 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 consists of 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.

[8] http://gwmftp.jacobs.com/xml_schema_corrected/hse_222_corrected.html

- 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.

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

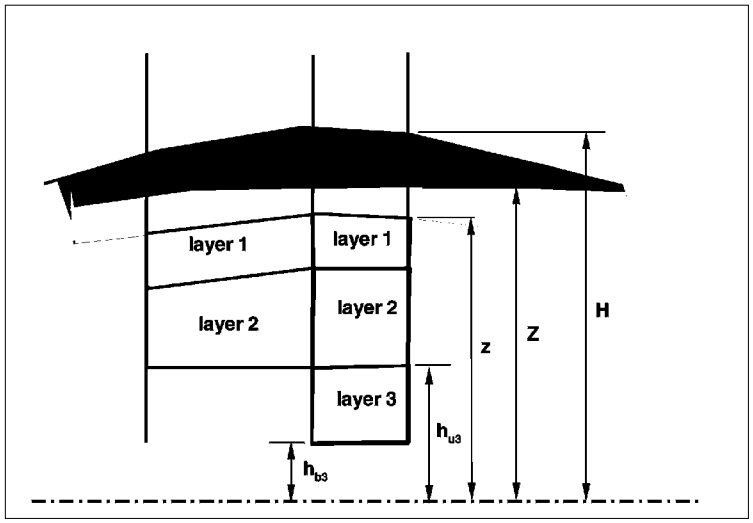


Figure C.1: *Sketch of the multi-layered grid used to solve 3-D groundwater flow.*

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 3.2 of this manual. This grid file will be extended from 2d to 3d based on the user-defined layering described in subsubsubsection 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 HPM Definition File `<HPM>`

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