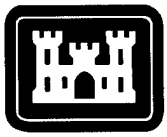
Contract Report CHL-99-1
January 1999



US Army Corps of Engineers
Waterways Experiment Station

User's Manual for LOCKSIM: Hydraulic Simulation of Navigation Lock Filling and Emptying Systems

by Gerald A. Schohl,
Tennessee Valley Authority

Prepared for Headquarters, U.S. Army Corps of Engineers

Contract Report CHL-99-1
January 1999

User's Manual for LOCKSIM: Hydraulic Simulation of Navigation Lock Filling and Emptying Systems

by Gerald A. Schohl

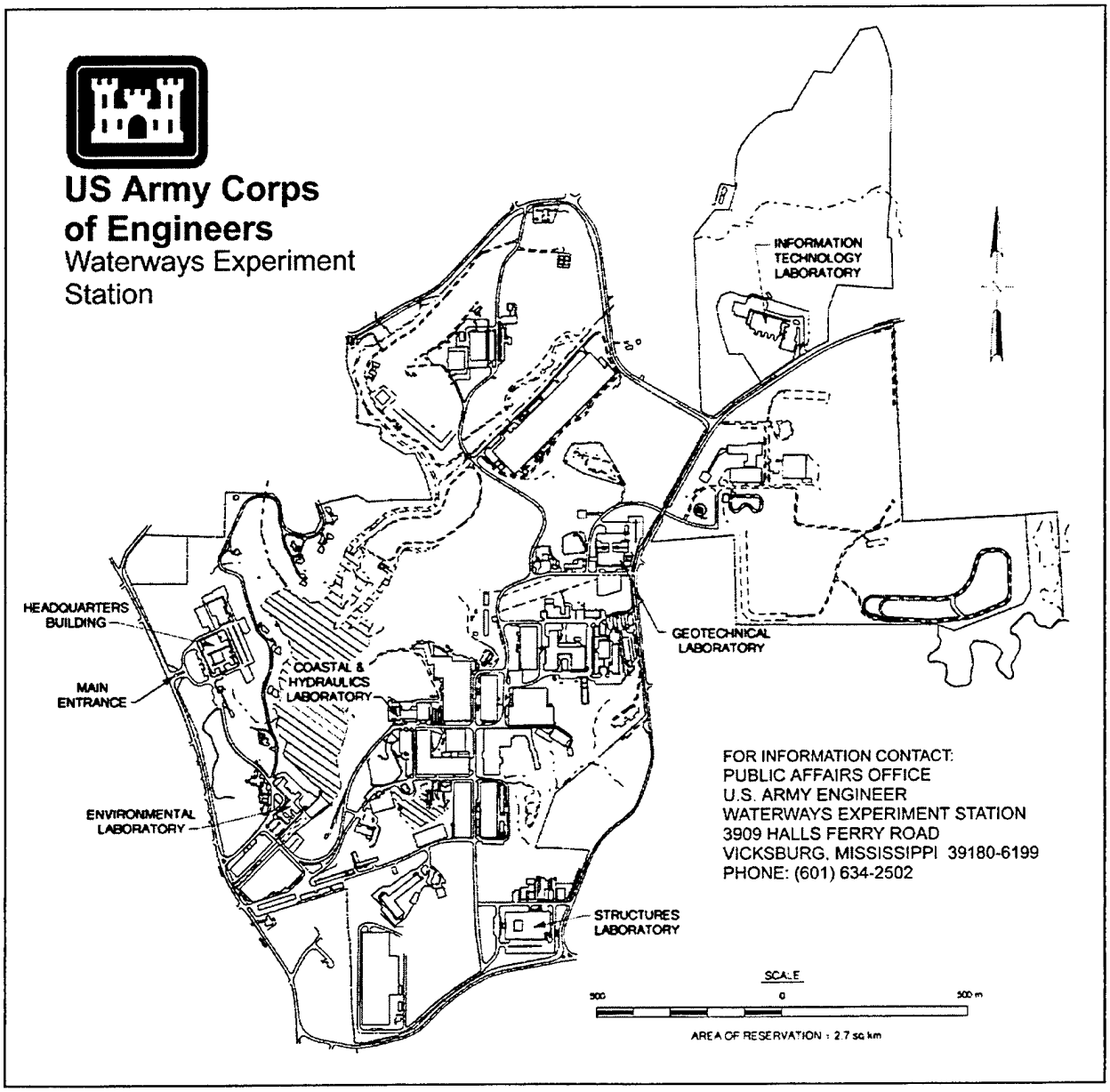
Tennessee Valley Authority
Engineering Laboratory
129 Pine Road
Norris, TN 37828

Final report

Approved for public release; distribution is unlimited

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Monitored by U.S. Army Engineer Waterways Experiment Station
3909 Halls Ferry Road, Vicksburg, MS 39180-6199



Waterways Experiment Station Cataloging-in-Publication Data

Schohl, Gerald Allen.

User's manual for LOCKSIM : hydraulic simulation of navigation lock filling and emptying systems / by Gerald A. Schohl ; prepared for U.S. Army Corps of Engineers ; monitored by U.S. Army Engineer Waterways Experiment Station.

190 p. : ill. ; 28 cm. -- (Contract report ; CHL-99-1)

Includes bibliographic references.

1. LOCKSIM (Computer program) 2. Locks (Hydraulic engineering) -- Computer programs -- Handbooks, manuals, etc. 3. Navigation. I. United States. Army. Corps of Engineers. II. U.S. Army Engineer Waterways Experiment Station. III. Coastal and Hydraulics Laboratory (U.S. Army Engineer Waterways Experiment Station) IV. Title. V. Title: Hydraulic simulation of navigation lock filling and emptying systems. VI. Series: Contract report (U.S. Army Engineer Waterways Experiment Station) ; CHL-99-1.

TA7 W34c no.CHL-99-1

CONTENTS

Page

PART 1: NUMERICAL MODELING OF NAVIGATION LOCKS WITH LOCKSIM 1

INTRODUCTION 2
Computer System Requirements 2
Brief History 3
Role Of LOCKSIM In Lock Evaluation And Design 3
Overview of LOCKSIM 5
Network Connectivity, Terminology, and Boundary Conditions 6
LOCKSIM Components 8
STEPS FOR APPLYING LOCKSIM 10
Schematic Representation 10
Input Data 13
Sources of Data on Loss Coefficients 14
Nodes 14
Culvert Segments 15
Inlets and Outlets 17
Reverse Tainter Valves 18
Portholes 19
Lock Chamber 21
Selection of Simulation Time Step 23
Input File 24
**CONSTANTS* Section 25
**COMPONENTS* Section 25
**NODES* Section 28
**FUNCTIONS* Section 30
**CROSS_SECTIONS* Section 31
**PLOT_VARIABLES* Section 32
Simulation 33
Command Line Input and Initial Commands 33
Run Steady 33
Run Unsteady 34
Solution Convergence Problems 37
Post Processing 38
Filling Results for Wannabe Main Lock 40
Emptying Results for Wannabe Main Lock 42
Estimating Longitudinal Hawser Forces 42
EXAMPLE APPLICATIONS 46
Multiport Lock 46
Bottom-Longitudinal Lock 59

PART 2: LOCKSIM REFERENCE: OPERATION, OUTPUT, AND INPUT 72
CONVENTIONS 73
OPERATION 74
Starting a LOCKSIM Simulation 74
Command Line Input 74
Initial Commands and Files 75
Report Periods and Report Destinations 76
Run Steady 77
Run Unsteady 79
Node Continuity Test 79
Simulation Suspension 81
Solution Convergence Problems 82
OUTPUT REPORTS 83
Error Messages 83
Nodes Report 83
Components Report 84
Continuity Report 85
INPUT FILE FORMAT 88
Overall Structure 88
General Input Rules 88
Input And Output Units 90
*Constants 91
Simulation *time_step* 91
Physical Constants 93
Calculation Options 93
Solution Convergence 93
Input and Output 95
Output for Plots 95
Cross Sections 96
*Components 97
moc_pipe and *imp_pipe* 98
General Description 98
Equations 98
Numerical Solution 99
Input 102
open_channel and *river_channel* 104
General Description 104
Equations 104
Numerical Solution 105
Input for *open_channel* 106
Input for *river_channel* 107
pipe_loss, valve, and *check_valve* 110
General Description 110
Equations 110
Loss, Discharge, and Flow Coefficients 111

Input for *pipe_loss* 112
Input for *valve* 113
Input for *check_valve* 116
rev_tainter 117
General Description 117
Equations 118
Contraction and Loss Coefficients 119
Input 120
storage 122
General Description 122
Equations 122
Numerical Solution 123
Input 123
converging_tee and *diverging_tee* 124
General Description 124
Equations 124
Tee Coefficients 125
Keywords 127
Input for *converging_tee* 128
Input for *diverging_tee* 130
Relationships between tee coefficients when one leg has zero discharge 131
converging_manifold and *diverging_manifold* 133
General Description 133
Equations 135
Tee Coefficients 139
Keywords 139
Input for *converging_manifold* 141
Input for *diverging_manifold* 142
Example 143
*Nodes 145
Boundary and Initial Conditions 145
Input Requirements 146
*Functions 148
General Description and Common Keywords 148
Discrete Functions 151
Polynomial Functions 154
Power Functions 154
Sinusoid Functions 155
Composite Functions 155
Summation Functions 157
*Cross_Sections 159
Trapezoidal Cross Section 160
Circular Cross Section 161
Riverine Cross Section 162
General Description 162
Subsections 163

Geometric Parameters 163
Discrete Elevation Levels 165
Input 166
Cross Section Output File 168
*Plot_Variables 170
Format Specifications for Plot File 170
Input Requirements for Plot Variables 173
Available Keywords 174
Available Variable Types 175
Example Input 177
REFERENCES 179
SF 298

PREFACE

The study reported herein was authorized by Headquarters, U.S. Army Corps of Engineers (HQUSACE), as part of the Inland navigation Research Program. The work was performed under Civil Works Research Work Unit 33072, "Model for Evaluation of Lock Filling and Emptying System." This research was completed under the direction of Dr. James R. Houston, Director, Coastal and Hydraulics Laboratory (CHL); Mr. Charles C. Calhoun, Jr., Assistant Director, CHL; and Dr. Phil G. Combs, Chief of the Rivers and Structures Division (CRD), CHL. Dr. Richard L. Stockstill was the Principal Investigator for this work unit under the direct supervision of Mr. James R. Leech, Acting Chief, Spillways and Channels Branch, CRD. Messrs. David B. Wingerd and Michael F. Kidby, HQUSACE, were the Program Monitors; Mr. Earl E. Eiker, HQUSACE, was the Area Coordinator; and Mr. R. Ray Bottin, CHL, was the Manager of the Inland Navigation Research Program.

This project was conducted by the Tennessee Valley Authority (TVA) during the period January 1997 to September 1998. Dr. Gerald A. Schohl, TVA, was the author.

At the time of publication of this report, Commander of WES was COL Robin R. Cababa, EN.

PART 1: NUMERICAL MODELING OF NAVIGATION LOCKS WITH LOCKSIM

INTRODUCTION

LOCKSIM (LOCK SIMulator) is a numerical model developed at the Tennessee Valley Authority's (TVA's) Engineering Laboratory for simulation of one-dimensional transient filling and emptying flow in navigation locks. In LOCKSIM, a filling and emptying system is represented by a network consisting of closed conduit and open channel components arranged in any desired combination. The geometry, hydraulic characteristics, and boundary conditions of the network are described in an ASCII input file, which is read by LOCKSIM at the start of each simulation. LOCKSIM is operated interactively, allowing the user to examine results, change parameters, and decide whether to quit or continue at any point during a simulation. LOCKSIM's primary output is custom-specified by the user and easily imported into spreadsheet software for further analysis and plotting. LOCKSIM was designed as a general-purpose simulator, applicable to nearly any type of filling and emptying system.

In addition to its generality, LOCKSIM has numerous valuable features. Its input file is structured to be readable and easily modified using any text editor. The input is mostly free format, depending heavily on keyword-value pairs entered in any desired order. A rich selection of input keywords provide many options for specifying components and boundary conditions. Distinguishing technical features of LOCKSIM include prediction of longitudinal hawser forces in the chamber, prediction of cavitation index and minimum pressure downstream from the reverse tainter valves, rigorous treatment of dividing and combining flows through tees and manifolds, and capability of including upstream and downstream approach channels in models of filling and emptying systems. Although a user's manual has not been previously available, LOCKSIM has proven easy enough to use that it has been a valuable tool within the U.S. Army Corps of Engineers since 1992 (particularly at the Cincinnati office).

Part 1 of this user's manual describes the application of LOCKSIM to navigation locks and provides two detailed example applications. Part 2 provides detailed descriptions for operating LOCKSIM, interpreting its output, and specifying its input.

Computer System Requirements

The present version of LOCKSIM is an MS-DOS® application, requiring an IBM®-compatible computer with a 80386 or better processor. LOCKSIM will run either with MS-DOS as the primary operating system or from within an MS-DOS window in Windows® 3.1 or Windows® 95. Two versions of the LOCKSIM executable code are available. Together, they require less than 1.5 megabytes of hard disk space. The 16-bit, standard version, named LOCKSIM.EXE, can access only conventional memory (640K), which can restrict the size of the output plot file. The 32-bit version, named LCKSIM32.EXE can access all available memory, but executes about 30 to 50 percent slower than LOCKSIM.EXE. In Windows 95, LCKSIM32.EXE requires no special arrangements (all memory options for MS-DOS windows can be set to "Auto"). However, for earlier operating systems it is necessary to use EMM386.EXE in the CONFIG.SYS file with the "ram" switch set, rather than the "noems" switch set.

Additional requirements include a text editor (a good free one for Windows is Programmer's File Editor, available for downloading at www.lancs.ac.uk/people/cpaap/pfe/) for creating and editing LOCKSIM input files and a plotting or spreadsheet software package for processing and presenting numerical results.

Brief History

The origin of LOCKSIM is a BASIC computer code written by the author in the mid-1980s primarily for water hammer applications in closed conduits. This code was first applied to a navigation lock in 1989, when it was used to study a transient condition at TVA's Wheeler Main Lock (Schohl, 1989), which is a multiport lock. By 1992, the code, by then named TFSIM, had been rewritten in C and had the capability of modeling both open channels and closed conduits in the same network. With this capability, it was possible to model a lock chamber as an open channel fed by a closed conduit network of culverts and portholes, and to estimate longitudinal hawser forces as a lock is filled or emptied from the computed differences in water surface elevation along the chamber length. Application of TFSIM to navigation locks with multiport (TVA's Pickwick Main Lock) and bottom-longitudinal (Bay Springs Main Lock) filling and emptying systems was described, verified, and demonstrated in a 1992 TVA report (Schohl, 1992). Later that year, an executable version of TFSIM, along with some example input files and notes, was provided to Gordon Lance and Lyn Richardson of the Ohio River Division of the U.S. Army Corps of Engineers. Mr. Lance has since used the code extensively, both for screening of potential innovative designs and for analysis of existing locks. His applications have included split lateral systems (Markland Main Lock and Barkley Lock), bottom lateral systems (Uniontown Lock and J.T. Meyers Auxiliary Lock), and central culvert systems (McAlpine Replacement Lock). Other district offices that have used TFSIM include Huntington and Pittsburgh. During the past year, the author has used TFSIM to evaluate manifold and culvert options for the proposed new multiport locks at TVA's Kentucky and Chickamauga Dams.

LOCKSIM is TFSIM without features that are unnecessary for simulation of filling and emptying systems. It is a mature computer code that has been in general use for over twelve years and applied to navigation locks for about nine years. It is a robust and reliable tool for evaluating navigation lock filling and emptying systems.

Role Of LOCKSIM In Lock Evaluation And Design

Table 1 lists the primary hydraulic performance factors of concern to navigation lock designers. Traditionally, information about all of these parameters is obtained from physical model studies. Most data are acquired from a relatively large scale (1:15 to 1:25 scale) model of the lock chamber, filling and emptying system, and near upstream and downstream approaches. Navigation data are acquired from a smaller scale (1:90 to 1:120 scale) model of the dam, including one or two miles of both the upstream reservoir and downstream tailwater. As indicated in Table 1, a one-dimensional numerical model such as LOCKSIM can provide

information on most of the key chamber, culvert, and tainter valve parameters. In addition, the discharge values provided by the model can be used to estimate velocities in the near upstream and downstream approaches. LOCKSIM can model the entire upstream and downstream approaches when they are narrow channels in which one-dimensional flow assumptions are reasonable.

Table 1 suggests that LOCKSIM is a supplement to, not a replacement for, traditional model studies. Certainly, the need for a physical navigation model is unaffected by the availability of the numerical model. The costs of testing a physical model of the filling and emptying system and lock chamber should be reduced if LOCKSIM is used to fine-tune the proposed design before the model is built. If sufficient confidence is gained in the numerical results, it should be possible to replace the traditional detailed lock model with a simpler model

Table 1: Primary Hydraulic Performance Factors and Sources of Performance Data for Navigation Locks

Hydraulic Parameters	LOCKSIM (1-D Num. Model)	1:25 Scale Chamber Model	1:100 Scale Navigation Model
Chamber:			
Fill and Empty Times	X	X	
Surface Turbulence		X	
Longitudinal Hawser Forces	X	X	
Transverse Hawser Forces		X	
Culverts:			
Pressures	X	X	
Discharges	X	X	
Air Pockets		X	
Tainter Valves:			
Cavitation Potential	X	X	
Strut Loads		X	
Air Vent Capacity	X		
Open Schedules	X	X	
Vibration Susceptibility		X	
Upstream Approach:			
Near Velocities	X	X	X
Intake Vortices		X	
Far Currents			X
Downstream Approach:			
Near Velocities	X	X	X
Hawser Forces		X	
Far Currents			X
Wave Heights		X	X

that permits testing of the intake structures, the discharge structures, and, perhaps, one reverse tainter valve, but does not include the port manifolds, the lateral or longitudinal culverts, or the lock chamber. This type of model would provide no information concerning chamber surface turbulence and transverse hawser forces. However, collection of these data from physical model studies may be unnecessary for established designs for which sufficient experience has been accumulated through past model studies. For example, past experience with multiport designs suggests that surface turbulence in a multiport lock decreases with increasing manifold length (Schohl, 1978), and transverse hawser forces are typically no larger than longitudinal hawser forces (Elder et al., 1964). Consequently, acceptable chamber performance should be achieved by a multiport lock designed by use of LOCKSIM with port manifolds that are as long as possible without violating longitudinal hawser force criteria.

LOCKSIM is also useful as an analysis tool for existing locks. For example, an earlier version of the model was used to determine the causes of violent releases of air and water through the downstream bulkhead slots at Wheeler Main Lock during emergency closure of the emptying valves (Schohl, 1989).

Overview of LOCKSIM

LOCKSIM simulates any lock filling and emptying system that can be described by a combination of closed conduit and free-surface components. Available closed conduit components include pipes (can represent any pipe, tube, duct, culvert, tunnel, or other closed conduit), reverse tainter valves, stop and throttling valves, check valves, pipe losses, tees, and manifolds. Available free-surface components include prismatic open channels, riverine channels, and water storage components (can represent surge tanks or reverse tainter valve wells in closed conduit systems and embayments or other discrete areas of water storage for free-surface components). Individual components from these lists are connected together at nodes, where they share a common piezometric head.

Discharge and piezometric head in the pipe and free-surface channel components are computed by numerically solving partial differential equations for one-dimensional unsteady flow. The water storage and air pocket components are governed by ordinary differential equations describing conservation of mass. The relationships between discharge and piezometric head difference for valves, check valves, and pipe losses are described by algebraic energy equations. The position of a valve is prescribed as a function of simulation time using tabulated data specified as a "function." Functions are used also by tee and manifold components, which simulate combining and dividing flow, to describe the variation of the branch headloss coefficients with the ratios of the individual branch discharges to the combined discharge.

Conservation of mass is enforced at every node. Vapor cavity formation, which should rarely or never occur in a navigation lock filling and emptying system, is simulated by allowing discrete vapor cavities to form, grow, and eventually collapse whenever the pressure drops to the liquid vapor pressure at any closed conduit computational section. External boundary conditions are applied at every node at every time step. A boundary condition is either a prescribed external

inflow or a prescribed pressure, either fixed or variable with simulation time. The default condition, which applies to most nodes, is one of zero inflow at all times.

Available time-varying numerical results include pressure, hydraulic gradeline elevation, and discharge at all computational points; stage, velocity, depth, top width, and channel area at all computational points within free-surface components; velocity, shear stress, and vapor cavity volume at all computational points within closed conduit components; and minimum pressure and cavitation index in the wakes of reverse tainter valves. Specified solution variables are saved at each time step to a disk file for later plotting. Alternatively, or in addition, comprehensive results for any time step are available through tabular reports directed to the user's monitor or printer, or to a file.

LOCKSIM is operated interactively from within an MS-DOS window: the user can suspend the simulation at any time step to examine results or change parameter values before continuing (or terminating) the simulation. Except for the interactive menu modules, the computer code is written in ANSI-standard C.

Network Connectivity, Terminology, and Boundary Conditions

For the purposes of LOCKSIM, a hydraulic network is described as a collection of interconnected *components* and *nodes* with imposed initial and boundary conditions. Each component, except for tee and manifold components, is bounded upstream by one node and downstream by another. Tee and manifold components are each bounded by three nodes. The user assigns labels, each which may consist of an arbitrary string of alphanumeric characters, to all nodes and designates upstream and downstream nodes for each component in the LOCKSIM input file. The labels permit each component to be identified by its bounding nodes. The designation of upstream and downstream nodes defines the direction of positive flow in each component.

Figure 1 illustrates a simple network consisting of five closed conduit components, indicated by single lines, and one free-surface component, indicated by a double line, connecting eight nodes, indicated by circles. The nodes in Figure 1 have been assigned simple two-character labels that uniquely identify each component. For example, for the indicated directions of positive flow in Figure 1 the pipe between nodes N1 and N2 is component "N1 N2" and the valve between nodes N7 and N8 is component "N7 N8." Similarly, the tee between nodes N2, N3, and N6 is component "N2 N3 N6" or "N2 N6 N3" (either is possible for the indicated directions of positive flow) depending on its specification in the LOCKSIM input file.

Figure 2 illustrates the terminology associated with a single component bounded by upstream and downstream nodes and subdivided into internal computational segments called *reaches*. Each reach is bounded upstream and downstream by computational *sections*, at which head, discharge, and other variables are computed and available for output. When sections are referred to by numbers, they are counted from upstream, which is always section 0, to downstream. Every component has at least one reach but only components with physical length,

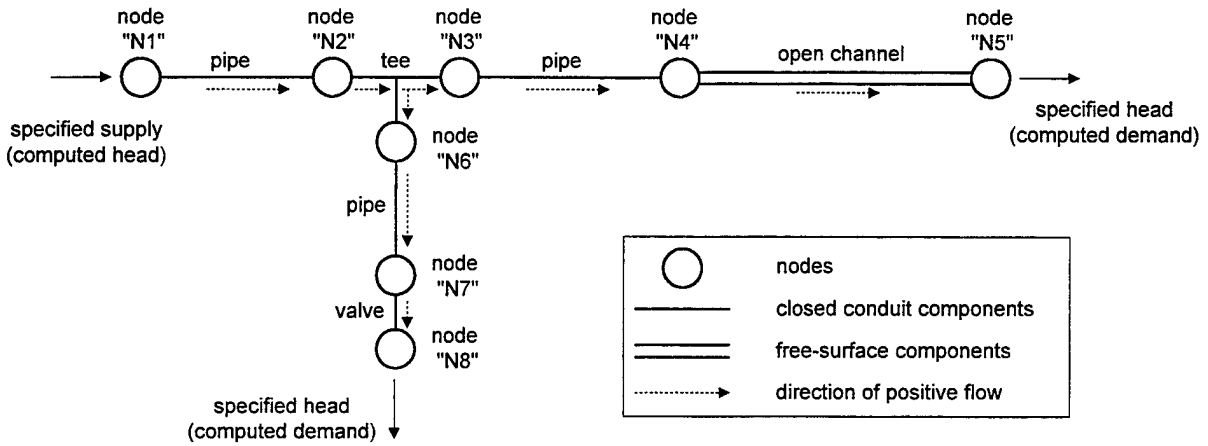


Figure 1: Simple network illustrating concept of linked components, nodes, and tees

which include only pipes and free-surface channels, can have multiple reaches All reaches within the same pipe or prismatic open channel component are the same length. However, reaches within one riverine channel component may each be a different length. For pipe components, the elevations of the upstream and downstream computational sections are set equal to the elevations of the upstream and downstream nodes, respectively. For prismatic open channel components, the elevations of the upstream and downstream sections are either specified explicitly or allowed to default to the adjacent node elevations. In both cases, linear interpolation is used to determine the elevations of internal computational sections For riverine channels, the elevations of all sections are explicitly specified. Computational reaches are discussed further in the descriptions of the *moc_pipe*, *imp_pipe*, *open_channel*, and *river_channel* components in the *Components section in Part 2.

A node is either a boundary point, in which case it is a terminal node, or a point of common piezometric head to two or more intersecting components, in which case it is an internal node. In Figure 1, nodes N1, N5, and N8 are all terminal nodes while nodes N2, N3, N4, N6, and N7 are all internal nodes. Although it is not indicated in Figure 1, every node is assigned an

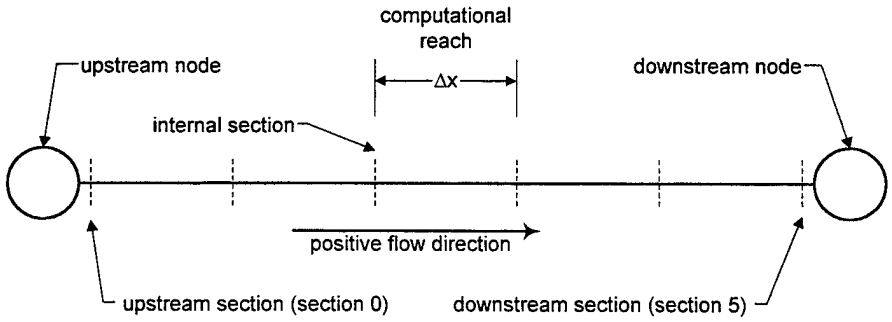


Figure 2: Definition sketch for component with multiple reaches

elevation. Also, every node requires a boundary condition for either head or external demand (never both), which may be either fixed or variable with simulation time. The boundary condition for an internal node is rarely specified explicitly because it defaults to zero demand (no external inflow or outflow), which is usually correct for internal nodes. However, terminal nodes always require explicitly specified boundary conditions. In Figure 1, supply (negative demand) is specified for terminal node N1 and head is specified at terminal nodes N5 and N8. Demand is determined at each time step as part of the solution for nodes with specified heads. Head is determined at each time step as part of the solution for nodes with specified demand. Node boundary conditions are discussed further in the *Nodes section in Part 2.

In LOCKSIM, the only components requiring prescribed boundary conditions are the *valve*, *rev_tainter*, *open_channel*, and *river_channel* components (these components are introduced in the following section). The *valve* and *rev_tainter* components represent valves with time-varying or fixed positions, which must be specified in the input data. The *open_channel* and *river_channel* components may have fixed or time-varying lateral inflows distributed along their length. Because the default lateral inflow is zero, it is usually unnecessary to explicitly specify lateral inflows for filling and emptying systems, which should rarely (if ever) need to simulate them.

LOCKSIM Components

The components that serve as the links between nodes in a hydraulic network are described in detail in the *Components section in Part 2. Each component type is assigned a keyword by which it is referenced. A brief summary of the available components, their identifying keywords, and their use in lock simulation is provided here.

imp_pipe	closed conduit solved using Preissmann's implicit scheme
moc_pipe	closed conduit solved using method of characteristics

The *imp_pipe* component is normally used to represent culvert segments and other full flowing, closed conduit sections of a filling and emptying system. The *moc_pipe* component may be used instead, when the conduit length is long enough and the simulation time step is small enough to satisfy the Courant number criterion described by Equation 10 in Part 2.

open_channel	prismatic open channel
river_channel	riverine channel

The *open_channel* component is normally used to represent segments of the free-surface lock chamber. However, the *river_channel* component may be used instead when the cross section varies significantly along the length of the chamber. Either or both of these components may be used to represent relatively narrow upstream and downstream approach channels that are adequately modeled by one-dimensional flow.

rev_tainter	reverse tainter valve in navigation lock culvert

The *rev_tainter* component is used to represent reverse tainter filling and emptying valves in lock culverts. Its computed results include downstream cavitation index and minimum pressure.

pipe_loss	minor loss in closed conduit

The *pipe_loss* component is used to represent various sources of closed conduit form loss in a filling and emptying system, including inlets, exits, culvert bends, and culvert transitions. If calibration data are available, *pipe_loss* components may also be used to represent the portholes through which the chamber is filled and emptied. However, the tee and manifold components discussed next are preferred for this purpose.

converging_tee	tee with combining flow defined as positive
diverging_tee	tee with dividing flow defined as positive
converging_manifold	manifold with combining flow defined as positive
diverging_manifold	manifold with dividing flow defined as positive

The *converging_tee* and *diverging_tee* components are used for simulation of combining and dividing flow through individual portholes or through tees and wyes that may be present in the filling and emptying culverts. The *converging_manifold* and *diverging_manifold* components are used to represent multiple portholes, when there are too many to specify a tee component for each. The only difference between the converging and diverging versions of these components is the defined directions of positive flow. Because diverging flow occurs through the portholes during filling, the diverging versions of these components are often preferred so that filling flow is positive in the output. These components represent the physics of flow through and past portholes much better than does the *pipe_loss* component. However, they require significant input data to define the variation of two headloss coefficients both with the ratio of flow through the porthole to flow past the porthole and with the different possible directions of flow. Default coefficients are available, but their applicability to lock portholes is uncertain.

storage	liquid storage with free surface

The *storage* component is often used to represent the water storage capability of the reverse tainter valve wells in filling and emptying culverts. In most cases, this storage effect is probably insignificant and could be neglected. However, it is sometimes simpler to include insignificant effects in a model than it is to later explain their insignificance. In models for which estimated hawser forces are not of interest, the *storage* component may also be used, rather than the *open_channel* or *river_channel* components, to represent the lock chamber.

valve	valve with time-varying position

The *valve* component is available to represent filling and emptying valves, or any other valves present, that are not reverse tainter valves. Downstream cavitation index and minimum pressure are not predicted for the *valve* component.

check_valve	nonreverse flow valve

The *check_valve* component is available (although rarely, if ever, needed) to represent any nonreverse flow valves that may be present in a filling and emptying system.

STEPS FOR APPLYING LOCKSIM

The steps for applying LOCKSIM to a navigation lock filling and emptying system are summarized as follows:

1. prepare schematic representation of hydraulic network,
2. gather input data,
3. create LOCKSIM input file,
4. perform simulation, and
5. process output results

The schematic representation (Figure 1, for example) shows how nodes and components are connected together to represent the flow paths of the filling and emptying system. The component types are indicated on the schematic and the nodes are assigned labels. The node boundary conditions may also be shown. The input data include node elevations and boundary conditions, and component lengths, areas, loss coefficients, and boundary conditions. The LOCKSIM input file describes, in a text format, the information depicted on the schematic representation and lists the input data. The input file also includes specifications of the variables for which time-varying output is desired at the end of the simulation. The simulation is performed for the completed input file and the time-varying results are saved to an output plot file. The plot file is loaded into spreadsheet software, where the results may be further analyzed, plotted, and printed. Additional simulations are performed with different boundary conditions or other input data after editing the input file to reflect the desired changes.

Figure 3 depicts the filling and emptying flow passages for a simple (fictional) side-port 110' by 600' navigation lock referred to as "Wannabe Main Lock." Filling and emptying flows for this lock are controlled by reverse tainter valves in two main culverts extending along the length of the chamber in each wall. Flow enters and leaves the lock chamber through five circular portholes in each wall. Using Wannabe Main Lock as an example, the steps for applying LOCKSIM to simulate filling and emptying of a navigation lock are described below. Two additional, more complex examples, are presented in the "EXAMPLE APPLICATIONS" section of this guide.

Schematic Representation

A schematic representation for Wannabe Main Lock is shown in Figure 4. Components representing physical segments of the hydraulic network are connected at labeled nodes to represent all possible flow paths. Beyond this basic information, a schematic can be as simple or as detailed as desired. Figure 4 shows the directions of positive flow, indicates the elevations of

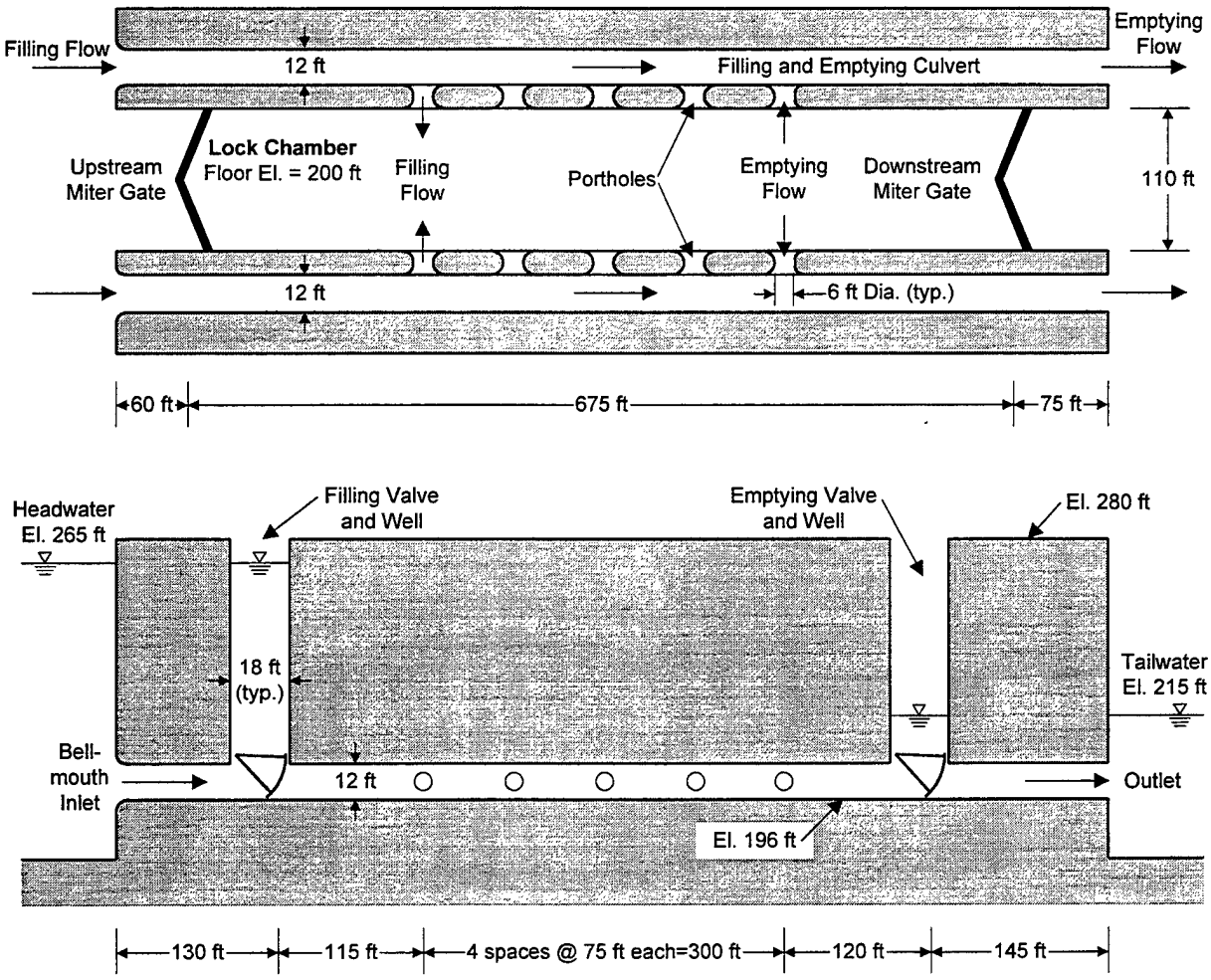


Figure 3: Plan and Elevation Sketch for Wannabe Main Lock

key nodes (in parenthesis under the node label), and assigns different symbols to different types of components. Schematics sometimes also include the lengths and cross section dimensions of individual culvert and chamber segments. Symbols used to represent nodes and different types of components are, of course, arbitrary and different users may prefer different notations.

The node labels assigned in Figure 4 are brief but somewhat descriptive. The labels for the river side nodes all begin with "R" for river. The labels for the corresponding land side nodes all begin with "L" for land. Descriptive phrases in the node labels include "HW" for headwater, "TW" for tailwater, "inlet," "outlet," "fill" for fill valves, "emp" for empty valves, "pintle" for miter gate pintles, "US" for upstream, and "DS" for downstream. The nodes bordering the porthole tee components all include a "p" for porthole. The "u" and "d" suffixes on the valve and porthole nodes indicate upstream and downstream, respectively. The "b" suffix on the porthole nodes indicates branch. The interior lock chamber nodes all begin with "C" for chamber.

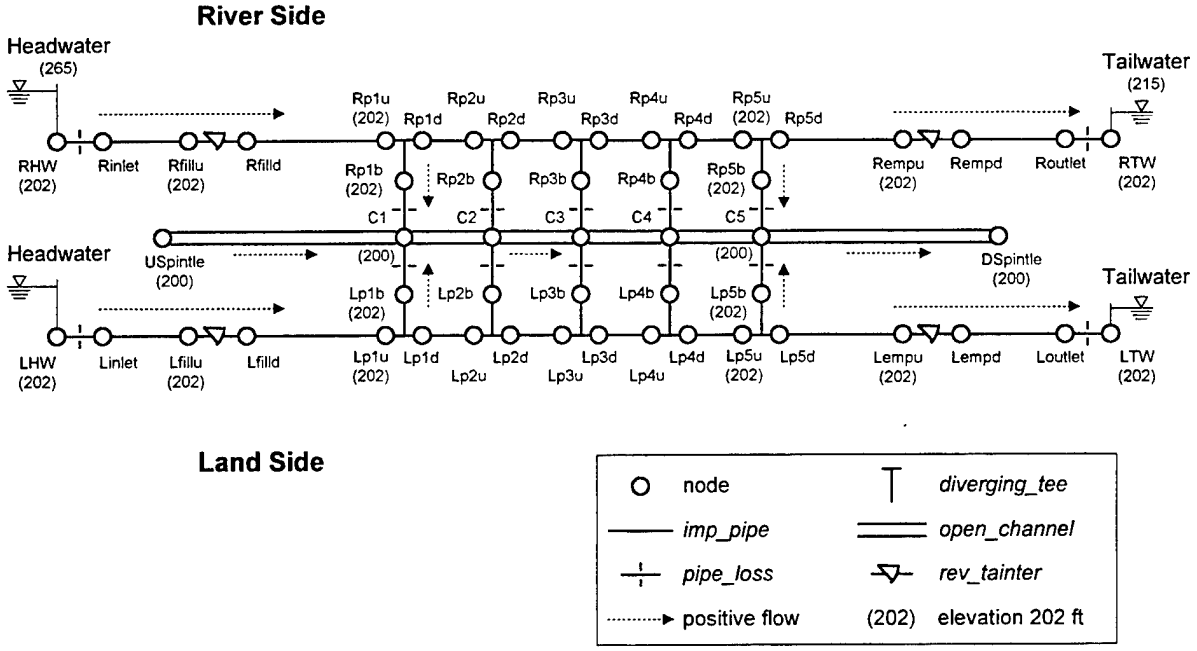


Figure 4: Schematic representation of Wannabe Main Lock using tees for portholes

The lock inlets and outlets are represented in Figure 4 by *pipe_loss* components, the filling and emptying valves by *rev_tainter* components, the culvert segments by *imp_pipe* components, and the chamber segments by *open_channel* components. Each porthole is represented by a *diverging_tee* component combined with a *pipe_loss* component. This approach effectively separates the energy change associated with diverging flow into the branch (during filling) from the energy loss associated with flow exiting the branch into the chamber. This separation is necessary when the default tee coefficients are used, which is the case in this example, because the default coefficients do not account for the exit loss associated with flow from the end of the tee branch into the chamber. The default coefficients apply to standard tees, which have velocity head in each tee leg based on the discharge and area of that leg. When data on combining and dividing loss coefficients are available, a porthole may be represented by a tee component by itself if the tee coefficients determined from the data are based on the branching flow losing its velocity head in the lock chamber. For further information see the "*converging_tee* and *diverging_tee*" section in Part 2.

Figure 5 shows an alternative schematic representation of Wannabe Main Lock, in which the portholes are represented by *pipe_loss* components without tees. Compared with the schematic in Figure 4, this simpler representation will simulate faster and will converge to a solution more readily. However, as explained by Schohl (1992), the physics of flow through and past portholes is much better represented by tee components than by branching *pipe_loss* components. Also, *pipe_loss* components assume a constant loss coefficient for all flow conditions while the loss coefficients for dividing and combining flow through a lock porthole vary with the ratio of the branch flow to the culvert flow. As a lock fills or empties, this flow ratio, and the tee coefficients that depend on it, vary along the length of the lock manifold and

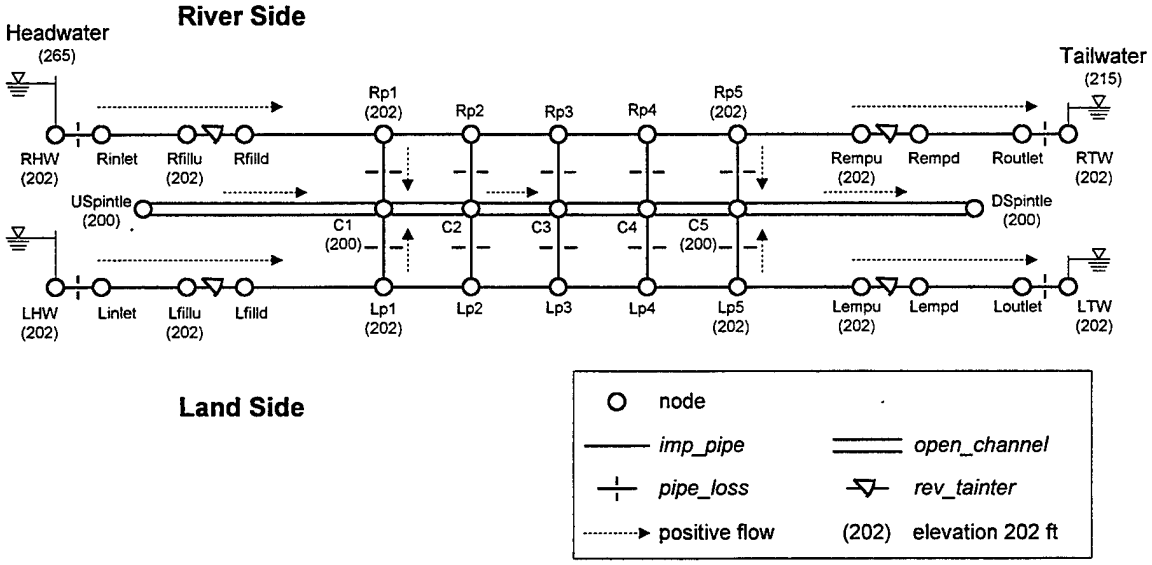


Figure 5: Schematic representation of Wannabe Main Lock using losses for portholes

with time. Nevertheless, the use of *pipe_loss* components with constant loss coefficients to represent portholes works reasonably well so long as the loss coefficients used are determined by calibration using available prototype or model data for the lock under study. This approach is used below for Wannabe Main Lock by using the results obtained with tees representing the portholes as the calibration standard.

Real filling and emptying systems include transition sections and bends in the culverts and may have too many portholes to represent each one using a separate *diverging_tee* or *pipe_loss* component. Bends and transitions are represented by additional *pipe_loss* components. For multiple portholes, the *diverging_manifold* component is used instead of the *diverging_tee* component or, when portholes are represented by *pipe_loss* components, each specified *pipe_loss* component represents the lumped influence of a number of actual portholes. Additionally, as mentioned above, it is often desirable to represent the filling and emptying valve wells using *storage* components. See the "EXAMPLE APPLICATIONS" section of this guide for schematics that includes these additional components.

Input Data

Input data for the nodes and components shown in Figure 4 must be gathered before the LOCKSIM input file can be completed. The required data is different for nodes than for components and varies with each type of component. Geometrical data are gathered primarily from project drawings. Hydraulic data, such as friction and form loss coefficients, are usually estimated from values for steady-state flow found in the literature or from available model or prototype data. The input requirements and data sources for nodes and each type of component

used in the schematic representation of Wannabe Main Lock are described below. For additional information on required inputs, see the corresponding sections in Part 2.

Sources of Data on Loss Coefficients

Except for reverse tainter filling and emptying valves (USACE, 1988), the published literature includes little data on loss coefficients specifically for components in navigation lock filling and emptying systems. More general data are available, however, in many sources including USACE (1988), Idelchik (1986), Miller (1990 and 1994), Blevins (1984), and Crane (1969). In addition, text books on elementary fluid mechanics (e.g., Streeter and Wylie, 1979) usually include some loss coefficient data. Engineering judgment is often required in applying loss coefficients found in the literature for inlets, outlets, conduit bends, and conduit transitions to the corresponding components in a filling and emptying system.

Nodes

Table 2 summarizes the required input data for the nodes defined for Wannabe Main Lock. Every node requires input values for elevation. Every terminal node requires an input value for boundary condition. Most internal nodes do not require an input value for boundary condition because the default usually applies. Every node requires an initial condition, which is initial head for nodes with demand boundary conditions and initial demand for nodes with head boundary conditions. Default initial conditions are available, but in most cases are incorrect for a filling and emptying system.

The elevations of a number of nodes are indicated in Figure 4. The elevations of the remaining nodes should be apparent by their locations relative to nodes with indicated elevations. In closed conduit components (the filling and emptying culverts and portholes), the node and internal section elevations are always centerline elevations. In free-surface channels (the lock

Table 2: Node Input Data for Wannabe Main Lock

Labels	Elev. (ft)	Boundary Condition	Initial Cond. (filling)	Initial Cond. (emptying)
RHW, LHW	202	head=265	demand=0	demand=0
Rinlet, Linlet	202	default	head=265	head=265
Rfillu, Lfillu	202	default	head=265	head=265
Rfilld, Lfilld	202	default	head=215	head=265
Rp1u ... Rp5u, Lp1u ... Lp5u	202	default	head=215	head=265
Rp1d ... Rp5d, Lp1d ... Lp5d	202	default	head=215	head=265
Rp1b ... Rp5b, Lp1b ... Lp5b	202	default	head=215	head=265
USpintle, DSpintle	200	demand=0	head=215	head=265
C1 ... C5	200	default	head=215	head=265
Rempu, Lempu	202	default	head=215	head=265
Rempd, Lempd	202	default	head=215	head=215
Routlet, Loutlet	202	default	head=215	head=215
RTW, LTW	202	head=215	demand=0	demand=0

chamber segments), the node elevations should be lower than the water surface elevation and must not be lower than the connecting channel bottom elevations. When they are not explicitly specified, the channel bottom elevations default to the connecting node elevations. The elevations of internal sections in free-surface channels always refer to channel bottom elevations.

In Figure 4 the elevations of the chamber nodes C1 through C5 are all specified as equal to the chamber bottom elevation. Because the bottom elevations default to the elevations of the connecting nodes, it is unnecessary to explicitly specify bottom elevations for the *open_channel* components representing the chamber segments. However, because nodes C1 through C5 are the downstream nodes for the *pipe_loss* components representing the porthole exit losses, it is also reasonable to specify their elevations as equal to the porthole centerline elevations, and to explicitly specify the chamber bottom elevations for the *open_channel* components. Either approach is acceptable and will result in equivalent solutions. In LOCKSIM, node elevations are used primarily for the purpose of determining local pressure values, which are important only for output and for determining the formation of vapor cavities. Filling and emptying flows are driven by differences in piezometric head, a variable that combines the effects of elevation and pressure, rather than by pressure alone or by elevation alone. Consequently, different choices in specifying node elevations usually do not affect solution results except for the reported pressure values.

All terminal nodes require specified boundary conditions, which are either piezometric head or external demand. The terminal nodes in Figure 4 and Figure 5 are those labeled RHW, LHW, RTW, LTW, USpintle, and DSpintle. For filling and emptying systems, the inlet node (RHW and LHW) and outlet node (RTW and LTW) boundary conditions are usually fixed piezometric heads equal to the headwater and tailwater elevations, respectively. Figure 3 indicates that for Wannabe Main Lock, the headwater elevation is 265 feet and the tailwater elevation is 215 feet. The appropriate boundary condition for the nodes at the upstream and downstream ends of the lock chamber is demand equal to zero. As already mentioned, the boundary condition for terminal nodes must be explicitly specified even though demand equal to zero is the default boundary condition for nodes.

All nodes also require initial conditions, which define the state of the hydraulic network at the start of a simulation, before filling or emptying has been initiated. For filling and emptying systems, the initial conditions are usually easily specified because all discharges are zero, the piezometric heads of all nodes upstream from the closed culvert valves are equal to the headwater elevation and the piezometric heads of all nodes downstream from the closed culvert valves are equal to the tailwater elevation. LOCKSIM has a steady-state solution feature that can be used to compute and automatically insert the initial, steady-state conditions into the input file. The default initial conditions provide a reasonable starting point for the steady-state solution but are not correct initial conditions for filling or emptying a navigation lock.

Culvert Segments

Table 3 summarizes the required input data for the Wannabe Main Lock culvert segments represented by *imp_pipe* components. The closed conduit length, cross-sectional area,

hydraulic diameter (equal to 4A/P, where A = area and P = wetted perimeter), wall roughness height (or fixed friction factor), and acoustic wavespeed are all required inputs.

The conduit lengths are taken from project drawings, which for Wannabe Main Lock are given in Figure 3. Because *rev_tainter*, *pipe_loss*, and other non-pipe components do not simulate conduit friction or fluid momentum, the lengths assigned to culvert segments should include the portions of conduit within attached non-pipe components. For example, the length of *imp_pipe* component "Rinlet Rfillu" is measured from the upstream side of the inlet to the center of the reverse tainter valve skin plate. Similarly, the length of "Rfilld Rp1u" is measured from the center of the skin plate to the center of the first porthole downstream. If a bend or transition existed in the culvert, the upstream culvert segments would extend to the center of the bend or transition (not to the upstream end) and the downstream segment would start at the center of the bend or transition (not at the downstream end). In summary, all physical culvert length should be accounted for by the culvert segments.

The conduit cross-sectional dimensions used to compute area and hydraulic diameter are also taken from project drawings. Only diameter is necessary for circular culverts, but both area and hydraulic diameter are required to properly compute friction losses in noncircular conduits.

The wall roughness height is used in the Colebrook-White equation to compute friction losses in closed conduits. The value of 0.003 feet given in Table 3 is a recommended design roughness height for rectangular concrete conduits (USACE, 1988). Because lock culverts are nearly always rectangular concrete conduits and filling and emptying is controlled more by valves and portholes than by culvert friction, this roughness height of 0.003 feet is almost always satisfactory (unless prototype data are available that indicate otherwise).

The wavespeed value of 3500 ft/s in Table 3 is a representative value for concrete conduits (Kennison, 1956). For the slow transients (surging rather than waterhammer) associated with lock filling and emptying, the value assigned for wavespeeds is not critical.

It is unnecessary to specify reach lengths for pipe components. LOCKSIM automatically determines the proper reach length to ensure numerical stability and accuracy. For further discussion, see the "*moc_pipe* and *imp_pipe*" section in Part 2.

Table 3: Input Data for Wannabe Main Lock Culvert Segments

imp_pipe components	length (ft)	area (ft^2)	hyd. dia. (ft)	roughness (ft)	wavespeed (ft/s)
Rinlet Rfillu, Linlet Lfillu	130	144	12	0.003	3500
Rfilld Rp1u, Lfillu Lp1u	115	144	12	0.003	3500
Rp1d Rp2u, Rp2d Rp3u, ...	75	144	12	0.003	3500
Lp1d Lp2u, Lp2d Lp3u, ...	75	144	12	0.003	3500
Rp5d Rempu, Lp5d Lempu	120	144	12	0.003	3500
Rempd Routlet, Lempd Loutlet	145	144	12	0.003	3500

Inlets and Outlets

Table 4 summarizes the required input data for the *pipe_loss* components representing the Wannabe Main Lock inlets and outlets. In general, three cross-sectional areas are required because the *pipe_loss* equation (Equation 28 in Part 2) accounts for both the energy loss and the change in velocity head that may occur across the component. The loss area indicated in Table 4 is the cross-sectional area for which the loss coefficient is defined. The upstream and downstream areas are those used to compute the velocity heads upstream and downstream from the component. For the inlet, the upstream velocity head is essentially zero (the upstream reservoir) so a large area is specified that results in negligible upstream velocity. Similarly, the downstream velocity at the outlet is essentially zero so a large downstream area is specified. Convergence problems may occur if the specified "reservoir area" is more than 100 to 1000 times the inlet or exit area.

The loss coefficient for a lock inlet depends on the geometry of the inlet structure and approach flow pattern, including whether or not vortices form at the inlet, and it is increased by the presence of trashracks and trash on the trashracks. There appears to be little, if any, published data on loss coefficients for lock inlets. The loss coefficient for any non-reentrant inlet without trashracks, vortices, or adverse approach flow conditions should lie between zero and 0.5 with the loss area equal to the downstream conduit area. Because lock inlets are typically streamlined to minimize entrance losses and model tested to minimize vortex formation, their loss coefficients should rarely exceed about 0.2, which is the value that is normally assumed when reliable data are unavailable (e.g., Hebler and Neilson, 1976). The additional loss due to trashracks is often negligible because the entrance cross-sectional area covered by the trashracks is usually significantly larger than the culvert area downstream. For example, the loss coefficient for the inlet trashracks for TVA's Pickwick Main Lock is about 0.2 to 0.3 with the loss area equal to the total cross-sectional area of the inlets at the trashrack location (810 ft^2). However, with the loss area equal to the cross-sectional area of the culvert fed by the inlets (225 ft^2), the trashrack loss coefficient drops to about 0.02 ($0.2 \times 225^2/810^2$), which is only about 10 percent of the nominal loss coefficient of 0.2 for the inlet alone.

In apparent contradiction to the above information, it is necessary assume a value of 0.7 for the inlet loss coefficient at TVA's Wheeler Main Lock in order to match prototype data on the water surface elevation in the land-side filling valve well (Schohl, 1994). The reason for this is not clear. Perhaps the trashracks were covered with debris during the prototype tests or the losses were increased by vortices (the inlets were not observed during the prototype tests).

An upper limit on the loss coefficient for a lock exit is 1.0 with the loss area equal to the area of the upstream culvert. In this case, 100 percent of the culvert velocity head would be

Table 4: Input Data for Wannabe Main Lock Inlets and Outlets

pipe_loss components	loss area (ft^2)	upstream area(ft^2)	downstream area (ft^2)	loss coefficient
RTW Rinlet, LTW Linlet	144	10^5	144	0.2
Routlet RTW, Loutlet LTW	144	144	10^5	1.0

dissipated at the exit. An outlet structure that gradually expands the flow, without losses, from the culvert area, A_c, to a larger exit area, A_e, will dissipate the velocity head at the larger exit area rather than in the upstream culvert. As a result, its loss coefficient based on the culvert area is reduced from 1.0 to the ratio $(A_c/A_e)^2$. Because the expansion is unlikely to take place without losses, the loss coefficient for a diffusing outlet structure actually will lie somewhere between $(A_c/A_e)^2$ and 1.0. In the absence of prototype or model data, published literature on outlet structures and diffusing closed conduit flows can provide guidance in estimating outlet loss coefficients.

Reverse Tainter Valves

Table 5 summarizes the input geometrical data for the *rev_tainter* components representing the Wannabe Main Lock filling and emptying valves. The culvert height and width (see Figure 3) are required input values. The culvert floor elevation is not required, but is necessary if downstream cavitation index and minimum pressure are desired outputs. Additional required inputs include the valve position as a function of time and the valve loss or discharge coefficient as a function of position. The contraction coefficient as a function of valve position must also be specified to obtain downstream cavitation index and minimum pressure as outputs. Refer to Figure 69 in Part 2 for a definition sketch showing flow passing under a partially open reverse tainter valve.

Figure 6 shows the valve opening pattern assumed for the filling and emptying valves in Wannabe Main Lock. An opening pattern with an initially slower opening rate is typical and desirable (lower hawser forces in chamber than for faster initial opening rates) for reverse tainter valves in lock culverts. The dimensionless position of the valve is measured as b/B, where b = vertical opening under the valve and B = height of the culvert (see Figure 69 in Part 2). Dimensionless time is defined as t/T, where t = simulation time and T = total opening time. The opening pattern for a reverse tainter valve is determined by the geometry of the valve and its linkage to the operating machinery, usually either an electric motor or hydraulic piston. Typically, the motor or piston operate at a constant rate of speed so that time is a linear function of the position of the piston or of the angle of rotation of the sector arm driven by the motor.

The opening times (T) for the filling and emptying valves in Wannabe Main Lock are assumed to be 3 minutes and 1.5 minutes, respectively. The LOCKSIM input file has format features that make it easy to change opening times without changing the valve opening pattern, so long as the pattern is specified in dimensionless form as in Figure 6. When both filling or both emptying valves are opened simultaneously, only one valve opening pattern must be specified. When one filling or emptying valve starts to open before the other, a valve opening pattern for each must be specified. Usually, the second pattern would simply be a copy of the first, with

Table 5: Input Data for Wannabe Main Lock Filling and Emptying Valves

rev_tainter components	culvert height (ft)	culvert width (ft)	culvert floor elev. (ft)
Rfillu Rfilld, Lfillu Lfilld	12	12	196
Rempu Rempd, Lempu Lempd	12	12	196

different shift and scale parameters (see the *Functions section in Part 2 for further information). It is, of course, also possible to open only one of the filling or emptying valves and leave the other closed. The filling and emptying valves for the Wannabe Main Lock example will be opened simultaneously.

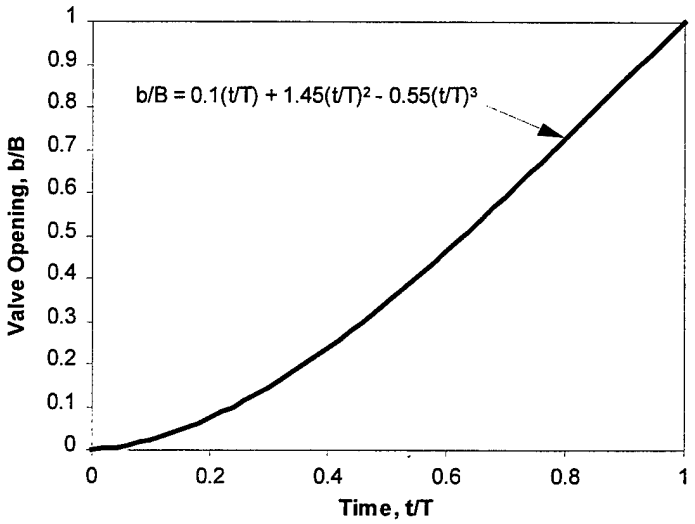


Figure 6: Valve opening pattern for Wannabe Main Lock filling and emptying valves

The curve illustrated in Figure 71 of Part 2, which is based on the best prototype data available for reverse tainter valves (USACE, 1988), is assumed to describe the variation of discharge coefficient with valve opening for Wannabe Main Lock, as well as for every other lock that the author has simulated. The primary uncertainty in using this curve is the value of the discharge coefficient for the fully opened valve, which varies from about 2.2 to 10 in the original data, corresponding to a variation in K_{vo} from about 0.2 to 0.01, where K_{vo} is the loss coefficient for the fully opened valve. In Figure 71, the discharge coefficient for the fully opened valve is assumed to be 3.16, corresponding to $K_{vo} = 0.1$. The author has sometimes adjusted the discharge coefficient for the fully opened valve within the 2.2 to 10 range to obtain better agreement between simulation results and model or prototype data. In these cases, a few additional discharge coefficient values were also adjusted (within the range of the prototype data) to maintain a smooth discharge coefficient versus valve opening curve.

The relationship described by Equation 37 and illustrated in Figure 70 of Part 2 is assumed to describe the variation of the contraction coefficient with valve opening for Wannabe Main Lock. This relationship, like the discharge coefficient relationship described above, is used almost always in the author's simulations of lock filling and emptying systems.

Portholes

Table 6 summarizes the required input data for the *diverging_tee* and *pipe_loss* components used to represent each porthole in the culverts for Wannabe Main Lock (see Figure 4). The reference numbers of the tee legs and angles in Table 6 are defined in Figure 81 in Part 2. Cross-sectional areas for each leg of each *diverging_tee* component must be specified. However, if a leg is circular its diameter may be specified instead. The angles between the tee legs are required because, as already mentioned, the default tee coefficients will be used. Otherwise it would be necessary to specify the loss coefficients for various flow directions through the tees as functions of the corresponding discharge ratios (see the "*converging_tee* and *diverging_tee*" section in Part 2). Convenience and simplicity are the motives for using the default loss coefficients for Wannabe Main Lock. In general, it is uncertain how well these loss coefficients derived from data for standard tees apply to lock culvert portholes.

Table 6: Input Data for Wannabe Main Lock Portholes

diverging_tee components	leg 1 area (ft^2)	leg 2 dia. (ft)	leg 3 area (ft^2)	angle 1 (deg)	angle 2 (deg)
Rp1u Rp1d Rp1b, ..., Rp5u Rp5d Rp5b	144	6	144	0	90
Lp1u Lp1d Lp1b, ..., Lp5u Lp5d Lp5b	144	6	144	0	90

pipe_loss components	loss dia. (ft)	u. s. dia. (ft)	d. s. area (ft^2)	loss coefficient for +Q	loss coefficient for -Q
Rp1b C1, ..., Rp5b C5	6	6	10^5	0.9	0.1
Lp1b C1, ..., Lp5b C5	6	6	10^5	0.9	0.1

Each *pipe_loss* component requires three cross-sectional areas, or diameters, as discussed in the "Inlets and Outlets" section above. Because the loss coefficient for positive, or filling, flow is different from the loss coefficient for negative, or emptying, flow, separate loss coefficients are specified for the *pipe_loss* components. Positive discharge from the porthole into the chamber experiences an exit loss, for which a loss coefficient of 0.9 is specified. A value less than 1.0 is used under the assumption that the bell-mouth exit partially expands the flow before its velocity head is lost. Negative discharge from the chamber into the porthole experiences an entrance loss, for which a coefficient of 0.1 is assumed.

Table 7 summarizes the required input when *pipe_loss* components alone are used to represent the portholes for Wannabe Main Lock (see Figure 5). The loss coefficients were determined by a trial and error calibration process using the results obtained with tees representing the portholes as the calibration standard. As illustrated in Figure 7, the loss coefficient for positive flow was successively adjusted until the computed chamber water surface elevations during filling reasonably matched those computed with tees representing the portholes. Similarly, as illustrated in Figure 8, the loss coefficient for negative flow was successively adjusted until the computed chamber water surface elevations during emptying reasonably matched those computed with tees representing the portholes. The calibrated loss coefficients apply only to Wannabe Main Lock and should not be applied to any other lock unless a similar calibration results in the same numbers. In addition to other factors, the values of calibrated loss coefficients depend on the type of portholes, the number of prototype portholes represented by each *pipe_loss* component, and the culvert geometry.

Table 7: Input Data for Portholes Modeled with *pipe_loss* Components

pipe_loss components	loss dia. (ft)	u. s. dia. (ft)	d. s. dia (ft)	loss coefficient for +Q	loss coefficient for -Q
Rp1 C1, ..., Rp5 C5	6	6	6	1.00	1.94
Lp1 C1, ..., Lp5 C5	6	6	6	1.00	1.94

The upstream, downstream, and loss diameters in Table 7 are assumed to all have the same value, which cancels out the upstream and downstream velocity heads in the governing equation for the *pipe_loss* component (Equation 28 in Part 2). This approach is as reasonable as any because the correct upstream and downstream velocities, those in the culvert and in the chamber, are not correctly predicted by dividing the porthole discharge by the culvert and chamber areas, respectively, and it is probably better to neglect changes in velocity head than to use incorrect values. In any case, the assumed upstream and downstream areas affect only the particular values determined by calibration for the loss coefficients. The computed results are the same as long as the specified loss coefficients are those determined for the specified upstream and downstream areas.

Lock Chamber

Table 8 summarizes the required input data for the *open_channel* components representing the Wannabe Main Lock chamber. The lengths of each chamber segment are evident from Figure 3. The

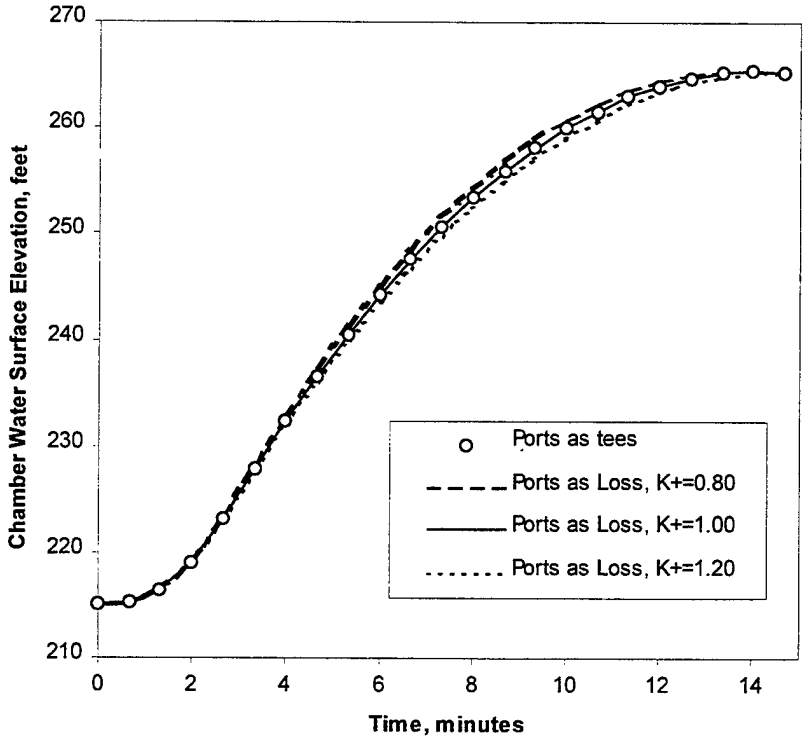


Figure 7: Filling calibration of portholes as *pipe_loss* components for Wannabe Main Lock

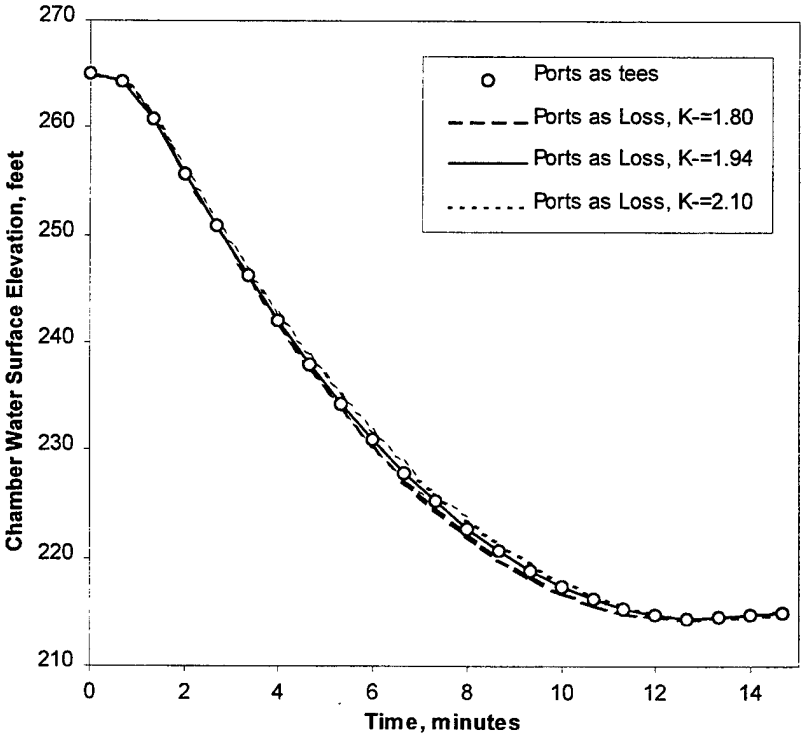


Figure 8: Emptying calibration of portholes as *pipe_loss* components for Wannabe Main Lock

Table 8: Input Data for Wannabe Main Lock Chamber

open_channel components	length (ft)	no. of reaches	width (ft)	floor elev. (ft)	Manning n
USpintle C1	185	5	110	200	0.02
C1 C2, C2 C3, C3 C4, C4 C5	75	2	110	200	0.02
C5 DSpintle	190	5	110	200	0.02

chamber length represented by *open_channel* components is, of course, the distance between the upstream and downstream miter gates (675 feet), rather than the nominal, or effective, length of the chamber (600 feet). Because the chamber cross section is assumed to be rectangular, its geometry is entirely defined by its width (110 feet) and floor elevation (200 feet).

The Courant number, defined by Equation 27 in Part 2, provides one criterion for selecting the number of computational reaches in each *open_channel* component. It is desirable for accuracy and resolution, although not necessary for stability, to specify enough reaches to ensure that the Courant number, C_r, remains near, or greater than, 1.0 during most of a simulation. This criterion is satisfied when the lengths of all reaches are less than or equal to Δx_{max}, which is determined from Equation 27 in Part 2 under the assumption that $C_r = 1.0$ and the cross section hydraulic depth is at its minimum value:

$$\Delta x_{max} = \Delta t \sqrt{g\left(\frac{A}{T}\right)_{min}} \tag{1}$$

in which Δt = simulation time step, g = acceleration of gravity, A = cross-sectional area, T = width of cross section at the water surface, and the subscript "min" indicates minimum value. For a rectangular cross section, Equation 1 reduces to

$$\Delta x_{max} = \Delta t \sqrt{g d_{min}} \tag{2}$$

in which d = water depth. For Wannabe Main Lock, $\Delta t = 2$ seconds, as discussed in the following section, and $d_{min} = 15$ feet, when the lock is empty. From Equation 2, the maximum reach length is about 44 feet. The numbers of reaches indicated in Table 8 are the minimum for each *open_channel* component that result in all reach lengths being less than 44 feet. A second criterion for selecting the number of reaches in each *open_channel* component is the need to define enough interior sections to adequately resolve the shape of the oscillating water surface in the chamber and to ensure that output is available at desired locations. This criterion may occasionally lead to use of a smaller reach length than suggested by Equation 2. In practice, the solution for the chamber water surface is not very sensitive to the number of reaches specified unless they are significantly too few.

The friction coefficient, n, used in the Manning equation (Equation 7 in Part 2) is assumed to be 0.02, a representative value for concrete-lined channels with excavated rock bottoms (Chow, 1959). However, higher values of n are sometimes used to increase physical

damping on the lock chamber water surface oscillations, as described in the "Estimating Longitudinal Hawser Forces" section of this guide. For the same reason, it is sometimes desirable to specify a value smaller than 1.0 for the momentum correction factor, β, which is an optional input parameter for describing cross sections (see the *Cross_Sections section of Part 2). The default value for β is 1.0 and, except to deliberately increase damping, there is no reason to specify a different value for a lock chamber cross section.

It is important to keep in mind that the *open_channel* component simulates *one-dimensional*, longitudinal flow in the lock chamber, which means that average velocity and depth vary over the length of the chamber, but not over the width. So, while longitudinal hawser forces can be estimated from differences in water surface elevation (depth) over the chamber length, transverse hawser forces cannot be estimated because differences in water surface elevation over the chamber width are not computed.

Also, all flow into or out of an *open_channel* component (lateral inflow, external supply at the upstream or downstream nodes, or flow from a component attached to the upstream or downstream nodes) is assumed to have no momentum in the direction of the channel flow. In other words, flows entering and leaving the lock chamber are assumed to be directed perpendicularly to the longitudinal axis of the chamber. When this is not the case, the longitudinal differences in water surface elevation computed using LOCKSIM (and the estimated hawser forces) are likely to be in error. Fortunately, the portholes in many locks do direct the chamber inflows and outflows in a direction approximately perpendicular to the chamber length axis.

When differences in longitudinal water surface elevation and estimated hawser forces are not of interest, the lock chamber may be represented by a *storage* component, as illustrated schematically in Figure 9, rather than by *open_channel* or *river_channel* components. Two input values are required to represent the Wannabe Main Lock chamber using a *storage* component. The first is the total surface area of the chamber, which is equal to 74,250 ft^2 (110 feet times 675 feet). The second is the maximum water surface elevation in the chamber before water flows over the top of the walls, which is 280 feet. In Figure 9, the *pipe_loss* components "LCP LC" and "RCP RC" each have loss area equal to the total porthole area on that side of the chamber and a negligible loss coefficient (e.g., 0.001). LOCKSIM cannot solve this network unless the *storage* component is separated from intersecting *diverging_tee* components on each side.

Selection of Simulation Time Step

LOCKSIM solves partial differential equations for unsteady flow in conduits and free-surface channels using finite difference numerical techniques. The term "finite difference" refers both to the finite length increments (computational reaches) into which the pipe and free-surface components are divided and to the finite time increments (time step) into which the simulation time domain is divided. Starting with time equal to zero and the specified initial conditions, the numerical solution progresses through simulation time in discrete steps, with the results at each new time step depending on the hydraulic conditions at the previous time and the updated boundary conditions. The size of the simulation time step, Δt, is specified by the user.

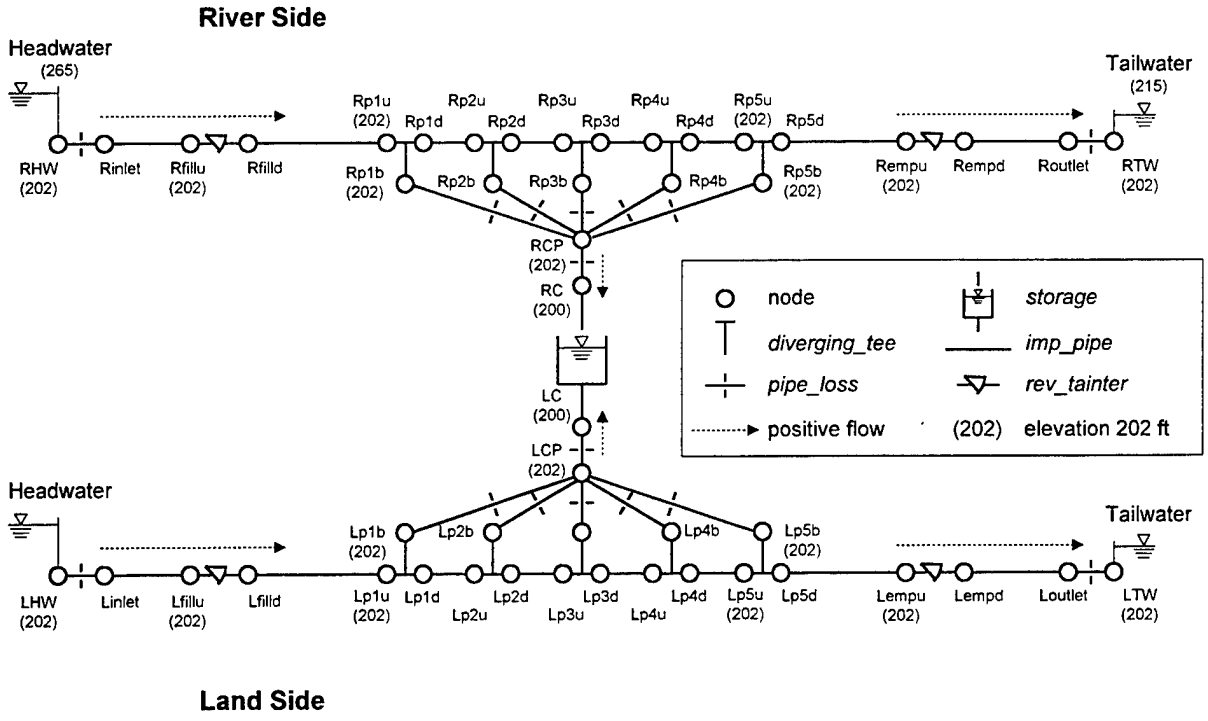


Figure 9: Schematic representation of Wannabe Main Lock using a *storage* component for the chamber

For simulations of filling and emptying systems a time step in the range of 1 to 5 seconds is usually appropriate, depending on how fast the valves are opened. A minimum criterion for selecting time step size is adequate resolution of the time-varying boundary conditions, which are typically only the valve opening patterns. Adequate resolution means that the curvature of the valve opening pattern is well represented by discrete points spaced Δt apart connected by straight lines. For Wannabe Main Lock, with the filling valves opening in 3 minutes, a time step as large as 10 seconds reasonably satisfies this criterion, but is otherwise not small enough. A time step of 2 seconds, which provides good output resolution, is chosen for this example. For valve opening times slower than about 4 minutes, a 5-second time step is usually adequate. For valve opening times of 1 to 1.5 minutes, a 1- or 2-second time step may be necessary to obtain a converged solution. When the best time step size is uncertain, it is useful to compare results obtained using different time steps. If the time step is small enough, making it even smaller will have little effect on the results (except for the hawser forces as discussed in the "Estimating Longitudinal Hawser Forces" section of this guide).

Input File

For LOCKSIM, a hydraulic network along with its boundary conditions is completely described in an ASCII input file that is read at the start of a simulation. Typically the user edits an existing input file to create a new input file, rather than start from scratch. Any text editor can

be used for this purpose. Figure 10 illustrates the input file, named "wannabe.sim," for Wannabe Main Lock. See Part 2 for detailed descriptions of all aspects of the input file.

The input file is divided into sections, each with its own input format. The start of each new section is indicated by its header title, which is one of the following:

**CONSTANTS*	Physical and simulation control constants
**COMPONENTS*	Component descriptions for network (pipes, valves, channels, losses, etc.)
**NODES*	Elevations, boundary and initial conditions for nodes
**FUNCTIONS*	Function descriptions
**CROSS_SECTIONS*	Definition of cross sections for channel components
**PLOT_VARIABLES*	List of variables for which values are saved for later plotting
**END*	Denotes end of input

The first three sections must be **CONSTANTS*, **COMPONENTS*, and **NODES* in that order. The header title **END* indicates the end of the input. Other sections may be included in any desired order. The **PLOT_VARIABLES* section is always optional. The **FUNCTIONS* and **CROSS_SECTIONS* sections are required only if components or nodes refer to particular functions or cross sections which must then be defined.

The first several lines of the input file for Wannabe Main Lock in Figure 10 are optional comments, which are ignored by LOCKSIM. Comments, which are indicated by an exclamation point (!), may be included anywhere in an input file. Blank lines and any text included after section header titles ("====" in Figure 10) also are ignored.

**CONSTANTS* Section

The **CONSTANTS* section of the input file in Figure 10 includes specifications for all of the constants, which are referenced by assigned keywords, normally required for simulation of navigation locks. The *Constants section in Part 2 describes many other constants, but all have default values that are usually acceptable. The values specified in Figure 10 for *wf_time* (implicit weighting factor, θ), *dQ_max* (solution discharge tolerance), *dH_max* (solution head tolerance), *dX_max* (tolerance for tee and manifold coefficients), *plot_line* (maximum length of a line in the plot file), *plot_field* (width for each value in plot file), and *plot_labels* (*row* specifies a plot file format that imports nicely into spreadsheet software) are normally applicable to any navigation lock simulation. The values specified for *time_step* (simulation time step), *plot_file* (name of plot file), and *io_units* (*English* or *SI* global input and output units), on the other hand, will often be changed by the user. For Wannabe Main Lock, a time step of 2 seconds (the default time units are seconds) is specified.

**COMPONENTS* Section

The **COMPONENTS* section of the input file in Figure 10 includes specifications for all of the components included in the schematic representation shown in Figure 4. In Figure 10, the components are specified in a convenient order, but any order is acceptable. A new line in

```
!Wannabe Main Lock Filling and Emptying System                    4/10/98

!  Example for LOCKSIM User's Manual
!  Portholes represented using diverging_tee and pipe_loss components
!  HW=265, TW=215
!  Filling valves open in 3 minutes
!  Emptying valves open in 1.5 minutes

*CONSTANTS ===================================================================
  time_step=2, wf_time=.55, io_units=English
  dQ_max=.001, dH_max=.0001, dX_max=.0001
  plot_file=wannabe.plt, plot_line=300, plot_field=11, plot_labels=row

*COMPONENTS ==================================================================

!------------ Inlet and fill valves -----------------------------------
LHW    Linlet pipe_loss us_area=1e5, area=144, K+=0.2
Linlet Lfillu imp_pipe dia=12, area=144, len=130, rough=.003, wave=3500
Lfillu  Lfilld rev_tainter  b=12, w=12, el_bottom=196
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    b/B_vs_t=cubic_open    ! filling
   !fixed_b/B=0            ! emptying

RHW    Rinlet pipe_loss us_area=1e5, area=144, K+=0.2
Rinlet Rfillu imp_pipe dia=12, area=144, len=130, rough=.003, wave=3500
Rfillu  Rfilld rev_tainter  b=12, w=12, el_bottom=196
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    b/B_vs_t=cubic_open    ! filling
   !fixed_b/B=0            ! emptying

!------------ Land wall culverts and ports ---------------------------
Lfilld Lp1u imp_pipe dia=12, area=144, len=115, wave=3500, rough=.003
Lp1d Lp2u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp2d Lp3u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp3d Lp4u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp4d Lp5u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp5d Lempu  imp_pipe dia=12, area=144, len=120, wave=3500, rough=.003

Lp1u Lp1d Lp1b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Lp2u Lp2d Lp2b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Lp3u Lp3d Lp3b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Lp4u Lp4d Lp4b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Lp5u Lp5d Lp5b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90

Lp1b C1 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp2b C2 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp3b C3 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp4b C4 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp5b C5 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10

!------------ River wall culverts and ports --------------------------
Rfilld Rp1u imp_pipe dia=12, area=144, len=115, wave=3500, rough=.003
Rp1d Rp2u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Rp2d Rp3u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Rp3d Rp4u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003

Rp4d Rp5u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Rp5d Rempu  imp_pipe dia=12, area=144, len=120, wave=3500, rough=.003

Rp1u Rp1d Rp1b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Rp2u Rp2d Rp2b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Rp3u Rp3d Rp3b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Rp4u Rp4d Rp4b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Rp5u Rp5d Rp5b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90

Rp1b C1 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Rp2b C2 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Rp3b C3 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Rp4b C4 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Rp5b C5 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10

!------------ Lock chamber modeled as series of open channels ---------
USpintle C1   open_channel  cross_section=chamber, len=185, reaches=5
C1       C2   open_channel  cross_section=chamber, len=75,  reaches=2
C2       C3   open_channel  cross_section=chamber, len=75,  reaches=2
C3       C4   open_channel  cross_section=chamber, len=75,  reaches=2
C4       C5   open_channel  cross_section=chamber, len=75,  reaches=2
C5   DSpintle open_channel  cross_section=chamber, len=190, reaches=5

!------------------------Empty valves and outlet-----------------------
Lempu  Lempd rev_tainter  b=12 , w=12, el_bottom=196
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    fixed_b/B=0             ! filling
    !b/B_vs_t=cubic_open    ! emptying
Lempd Loutlet imp_pipe dia=12, area=144, len=145, rough=.003, wave=3500
Loutlet LTW pipe_loss us_area=area=144, ds_area=1e5, K+=1.0

Rempu  Rempd rev_tainter  b=12 , w=12, el_bottom=196
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    fixed_b/B=0             ! filling
    !b/B_vs_t=cubic_open    ! emptying
Rempd Routlet imp_pipe dia=12, area=144, len=145, rough=.003, wave=3500
Routlet RTW pipe_loss us_area=area=144, ds_area=1e5, K+=1.0

*NODES =======================================================================

!------ initial conditions for filling ------------
LHW     elev=202   head=265 idemand=0
```

Figure 10: LOCKSIM input file for Wannabe Main Lock (file "wannabe.sim")

```
Linlet  elev=202  ihead=265
Lfillu  elev=202  ihead=265
Lfilld  elev=202  ihead=215
Lp1u    elev=202  ihead=215
Lp2u    elev=202  ihead=215
Lp3u    elev=202  ihead=215
Lp4u    elev=202  ihead=215
Lp5u    elev=202  ihead=215
Lp1d    elev=202  ihead=215
Lp2d    elev=202  ihead=215
Lp3d    elev=202  ihead=215
Lp4d    elev=202  ihead=215
Lp5d    elev=202  ihead=215
Lp1b    elev=202  ihead=215
Lp2b    elev=202  ihead=215
Lp3b    elev=202  ihead=215
Lp4b    elev=202  ihead=215
Lp5b    elev=202  ihead=215

Lempu   elev=202  ihead=215
Lempd   elev=202  ihead=215
Loutlet elev=202  ihead=215
LTW     elev=202   head=215 idemand=0

RHW     elev=202   head=265 idemand=0
Rinlet  elev=202  ihead=265
Rfillu  elev=202  ihead=265
Rfilld  elev=202  ihead=215
Rp1u    elev=202  ihead=215
Rp2u    elev=202  ihead=215
Rp3u    elev=202  ihead=215
Rp4u    elev=202  ihead=215
Rp5u    elev=202  ihead=215
Rp1d    elev=202  ihead=215
Rp2d    elev=202  ihead=215
Rp3d    elev=202  ihead=215
Rp4d    elev=202  ihead=215
Rp5d    elev=202  ihead=215
Rp1b    elev=202  ihead=215
Rp2b    elev=202  ihead=215
Rp3b    elev=202  ihead=215
Rp4b    elev=202  ihead=215
Rp5b    elev=202  ihead=215

Rempu   elev=202  ihead=215
Rempd   elev=202  ihead=215
Routlet elev=202  ihead=215
RTW     elev=202   head=215 idemand=0
USpintle elev=200  demand=0   ihead=215
```

```
C1   elev=200   ihead=215
C2   elev=200   ihead=215
C3   elev=200   ihead=215
C4   elev=200   ihead=215
C5   elev=200   ihead=215
DSpintle elev=200  demand=0   ihead=215

*FUNCTIONS ============================================================

!cubic_open  polynomial xscale=90, xmax=1  ! emptying
cubic_open  polynomial xscale=180, xmax=1  ! filling
   coefficients={ 0, .1, 1.45, -.55 }

tainterCdv  discrete  interpolation=spline
  x_values={0,   0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35,
            0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75,
            0.8, 0.85, 0.9, 0.95,   1}
  y_values={0, 0.0443, 0.0836, 0.1195, 0.1534, 0.196, 0.243, 0.295,
            0.349, 0.415, 0.494, 0.587, 0.69, 0.8165, 0.976, 1.155,
            1.387,  1.69,  2.02, 2.582, 3.162}

tainterCc  polynomial coefficients={.948, -1.396, 2.98, -2.918, 1.385 }

*CROSS_SECTIONS =======================================================

  chamber  trapezoidal  MN_n=.02, bed_w=110

*PLOT_VARIABLES =======================================================

!------ plot variables for filling -------------
 Lfillu Lfilld valve_position
 Lfillu Lfilld cav_index
 Lfillu Lfilld vena_head
 Lfillu Lfilld discharge

! Lengths: a->c=600,  a->b=298, b->c=302

 USpintle C1 depth sectno=1 ! (a)
 C3  depth                  ! (b)
 C5 DSpintle depth sectno=4 ! (c)

 Lp1b C1 discharge
 Lp2b C2 discharge
 Lp3b C3 discharge
 Lp4b C4 discharge
 Lp5b C5 discharge

*END ==================================================================
```

Figure 10 (continued): LOCKSIM input file for Wannabe Main Lock (file "wannabe.sim")

the input file is started for each component. The components are referenced by the labels assigned to their bounding nodes, as described in the "Network Connectivity, Terminology, and Boundary Conditions" section above. The component type is then indicated by the appropriate keyword (e.g., *imp_pipe*) following the component reference. Component properties, which are taken from the tables in the "Input Data" section above, are specified using keyword=value pairs following the component type. Valid keywords depend on the component type and are listed in the *Components section in Part 2. Numerical values are specified in *English* units, which are described in the "Input and Output Units" section in Part 2. The keyword=value pairs may be spread over as many lines as desired before the next component is specified.

Comments are included for clarity and, in the case of the reverse tainter valves, to disable one component option in favor of another. The latter practice permits one input file to be used for both filling and emptying simulations, with some editing in between. In Figure 10, operation of the filling and emptying valves is specified one way for filling simulations (filling valves gradually opened while emptying valves remain closed) and a different way for emptying simulations (emptying valves gradually opened while filling valves remain closed) by moving the exclamation point to disable the appropriate keyword=value pair.

The reverse tainter valve (*rev_tainter*) and open channel (*open_channel*) specifications in Figure 10 include keyword=value pairs in which the value is a descriptive string rather than a number. For the reverse tainter valves, these values specify the labels of "functions," which are defined in the **FUNCTIONS* section of the input file, describing the variation of one variable relative to another. For example, the keyword=value pair "b/B_vs_t=cubic_open" specifies that the function labeled "cubic_open" describes the valve opening pattern as a function of time that is depicted in Figure 6. The other keyword=value pairs specify functions for the discharge coefficient, C_{dv}, and contraction coefficient, C_c, both as functions of dimensionless valve opening, b/B. The function labeled "tainterCdv" describes the relationship depicted in Figure 71 of Part 2. The function "tainterCc" specifies the relationship described by Equation 37 in Part 2. For the *open_channel* components representing the lock chamber, the descriptive string values specify the labels of "cross sections," which are defined in the **CROSS_SECTIONS* section of the input file. The cross section and function descriptions are separated from the component definitions to permit one cross section or function definition, and the computer memory in which it is stored, to be shared by multiple components.

Figure 11 shows how the description of the land-side portholes in the **COMPONENTS* section of the input file would change if the portholes were represented only by *pipe_loss* components without tees, as illustrated in Figure 5. Similarly, Figure 12 shows the required changes to represent the lock chamber using a *storage* component, as illustrated in Figure 9. In both cases, corresponding changes would be required also for the river-side portholes.

**NODES* Section

The **NODES* section of the input file in Figure 10 includes specifications for all of the nodes included in the schematic representation shown in Figure 4. In Figure 10, the nodes are specified in a convenient order, but any order is acceptable. A new line in the input file is started

In Figure 10 replace

```
Lfilld Lp1u imp_pipe dia=12, area=144, len=115, wave=3500, rough=.003
Lp1d Lp2u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp2d Lp3u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp3d Lp4u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp4d Lp5u   imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp5d Lempu  imp_pipe dia=12, area=144, len=120, wave=3500, rough=.003

Lp1u Lp1d Lp1b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Lp2u Lp2d Lp2b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Lp3u Lp3d Lp3b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Lp4u Lp4d Lp4b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90
Lp5u Lp5d Lp5b diverging_tee area1=area3=144 dia2=6 angle1=0, angle2=90

Lp1b C1 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp2b C2 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp3b C3 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp4b C4 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp5b C5 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
```

with

```
Lfilld Lp1 imp_pipe dia=12, area=144, len=115, wave=3500, rough=.003
Lp1 Lp2    imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp2 Lp3    imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp3 Lp4    imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp4 Lp5    imp_pipe dia=12, area=144, len=75,  wave=3500, rough=.003
Lp5 Lempu  imp_pipe dia=12, area=144, len=120, wave=3500, rough=.003

Lp1 C1 pipe_loss dia=6, K+=1.0, K-=1.94
Lp2 C2 pipe_loss dia=6, K+=1.0, K-=1.94
Lp3 C3 pipe_loss dia=6, K+=1.0, K-=1.94
Lp4 C4 pipe_loss dia=6, K+=1.0, K-=1.94
Lp5 C5 pipe_loss dia=6, K+=1.0, K-=1.94
```

Figure 11: Change in **COMPONENTS* section when portholes are represented by *pipe_loss* components without tees

for each node. The first item on each new line is the node label. Node properties, which are taken from Table 2, are specified using keyword=value pairs following the node label. The valid keywords are listed in the *Nodes section in Part 2. In Figure 10, the keyword *head* specifies a fixed piezometric head boundary condition. The keywords *idemand* and *ihead* specify initial conditions.

A comment line at the beginning of the **NODES* section in Figure 10 indicates that the specified node initial conditions are for filling simulations. For emptying simulations, the initial conditions for all nodes between the filling and emptying valves change from initial head equal to the headwater elevation to initial head equal to the tailwater elevation. For convenience, a copy of the node definitions with the initial conditions set for emptying can be kept below the **END* header in the input file (all lines that follow the **END* header are ignored). Then before an emptying simulation is conducted, the nodes for filling simulations are cut from the text and pasted below the **END* header and the nodes for emptying simulations are cut from the text and pasted into the **NODES* section.

In Figure 10 replace

```
Lp1b C1 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp2b C2 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp3b C3 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp4b C4 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp5b C5 pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
```

with

```
Lp1b LCP pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp2b LCP pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp3b LCP pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp4b LCP pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
Lp5b LCP pipe_loss us_dia=dia=6, ds_area=1e5, K+=.9, K-=.10
LCP  LC  pipe_loss area=141.37, K+=.001
```

and replace

```
USpintle C1   open_channel  cross_section=chamber, len=185, reaches=5
C1       C2   open_channel  cross_section=chamber, len=75,  reaches=2
C2       C3   open_channel  cross_section=chamber, len=75,  reaches=2
C3       C4   open_channel  cross_section=chamber, len=75,  reaches=2
C4       C5   open_channel  cross_section=chamber, len=75,  reaches=2
C5   DSpintle open_channel  cross_section=chamber, len=190, reaches=5
```

with

```
LC RC surface_area=74250, max_wsel=280
```

Figure 12: Change in **COMPONENTS* section when chamber is represented by *storage* component

Figure 13 shows how the description of the land-side porthole nodes in the **NODES* section of the input file would change if the portholes were represented only by *pipe_loss* components without tees, as illustrated in Figure 5. Similarly, Figure 14 shows the required changes to represent the lock chamber using a *storage* component, as illustrated in Figure 9. In both cases, corresponding changes would be required also for the river-side nodes.

**FUNCTIONS* Section

The **FUNCTIONS* section of the input file in Figure 10 includes specifications for all of the functions referenced in the **COMPONENTS* and **NODES* sections above. For Wannabe Main Lock, and for most lock simulations, the **NODES* section does not reference any functions. Like the components and nodes in the **COMPONENTS* and **NODES* sections, functions may be specified in any desired order. A new line in the input file is started for each function. The first item on each new line is the function label, which has been referenced above. The second item is the function type indicated by a keyword, which in Figure 10 is either *polynomial* or *discrete*. Function properties are then specified using keyword=value and keyword=data cluster (a "value"

In Figure 10 replace

```
Lp1u    elev=202  ihead=215
Lp2u    elev=202  ihead=215
Lp3u    elev=202  ihead=215
Lp4u    elev=202  ihead=215
Lp5u    elev=202  ihead=215
Lp1d    elev=202  ihead=215
Lp2d    elev=202  ihead=215
Lp3d    elev=202  ihead=215
Lp4d    elev=202  ihead=215
Lp5d    elev=202  ihead=215
Lp1b    elev=202  ihead=215
Lp2b    elev=202  ihead=215
Lp3b    elev=202  ihead=215
Lp4b    elev=202  ihead=215
Lp5b    elev=202  ihead=215
```

with

```
Lp1    elev=202  ihead=215
Lp2    elev=202  ihead=215
Lp3    elev=202  ihead=215
Lp4    elev=202  ihead=215
Lp5    elev=202  ihead=215
```

Figure 13: Change in **NODES* section when portholes are represented by *pipe_loss* components without tees

consisting of a series of items enclosed in brackets) pairs. See the *Functions section of Part 2 for a description of other types of functions and for an explanation of the function property keywords and their values.

In Figure 10, two functions are described for the reverse tainter valve opening patterns, one for filling and one for emptying. For filling simulations, the emptying valve opening pattern is commented out and for emptying simulations the filling valve opening pattern is commented out. The only difference between the two is the value assigned to the *xscale* keyword, which is 90 (90 seconds or 1.5 minutes) for emptying and 180 (180 seconds or 3 minutes) for filling.

**CROSS_SECTIONS* Section

The **CROSS_SECTIONS* section of the input file in Figure 10 includes a specification for the one cross section referenced in the **COMPONENTS* section above. When several cross sections are referenced and must be defined, their definitions may be included in any desired order. A new line in the input file is started for each cross section. The first item on each new line is the cross section label, which has been referenced above. The second item is the cross

In Figure 10 replace

```
USpintle elev=200  demand=0   ihead=215
C1   elev=200   ihead=215
C2   elev=200   ihead=215
C3   elev=200   ihead=215
C4   elev=200   ihead=215
C5   elev=200   ihead=215
DSpintle elev=200  demand=0   ihead=215
```

with

```
LCP   elev=200   ihead=215
LC    elev=200   ihead=215
RCP   elev=200   ihead=215
RC    elev=200   ihead=215
```

Figure 14: Change in **NODES* section when chamber is represented by *storage* component

section type indicated by a keyword, which in Figure 10 is *trapezoidal*. Cross section properties, such as Manning n (*MN_n*) and bed width (*bed_w*), are then specified using keyword=value pairs. See the *Cross_Sections section of Part 2 for a description of other types of cross sections and other available keywords.

**PLOT_VARIABLES* Section

LOCKSIM produces tabular output but does not produce plots. LOCKSIM will, however, save data needed for plotting to an ASCII file in a columnar format that can be imported into an external plotting package or spreadsheet. The name of the file is specified using the *plot_file* keyword in the **CONSTANTS* section of the input file, or it may be entered interactively during the initial phase of a simulation. The variables to be saved during a simulation for later plotting are specified in the **PLOT_VARIABLES* section of the input file.

A new line in the **PLOT_VARIABLES* section of the input file is started for each plot variable. The first one or two items on each new line identify a node by its label or a component by the labels of its bounding nodes. The next item is a keyword identifying the desired variable type. The *depth* plot variable specifications in Figure 10 include a keyword=value pair identifying the component computational section (numbered starting at zero on the upstream end) for which output is desired. When the computational section is not specified, output for the upstream section (section 0) of a component is provided by default. See the *Plot_Variables section of Part 2 for a description of other variable types and available keywords.

The plot variables defined in Figure 10 include the valve position (*valve_position*), cavitation index (*cav_index*), head at the vena contracta downstream (*vena_head*), and the discharge (*discharge*) under the land-side filling valve. Because the lock is symmetrical, results for the river-side filling valve should be exactly the same as those for the land-side valve. The next three plot variables are water depths (*depth*) in the lock chamber. The depth at node C3, in the center of the lock, is used for plotting the lock water surface elevation as a function of time. The difference between the other two depths, one near each end of the lock, is used for estimating hawser forces. As indicated in the comment above, these variable specifications, the distance between computational section 1 (*sectno*=1) in *open_channel* component "USpintle C1" and computational section 4 (*sectno*=4) in *open_channel* component "C5 DSpintle," is 600 feet. Each of these components are divided into five, equal length, reaches and have six (numbers 0 through 5) computational sections. The remaining five plot variables specified in Figure 10 are the discharges through each of the land-side portholes. If the portholes were represented by *pipe_loss* components without tees, as illustrated in Figure 5, then the "b" would be removed from the upstream node names in these plot variables.

As in the **NODES* section in Figure 10, a comment line at the beginning of the **PLOT_VARIABLES* section indicates that the specified plot variables are for filling simulations. For emptying simulations, the plot variables related to the land-side (or river-side) emptying valve are specified rather than those for the filling valve. It is convenient to keep a copy of the plot variables for emptying below the **END* header in the input file so that they can be cut and copied as described above for the **NODES* section.

Simulation

LOCKSIM is operated interactively, allowing the user to examine results, change parameters, and decide whether to quit or continue at any point during a simulation. Operation of LOCKSIM is described in detail in the OPERATION section of Part 2. This section focuses on operating LOCKSIM to obtain results for filling of Wannabe Main Lock.

Command Line Input and Initial Commands

The completed input file, named "wannabe.sim," and a copy of the LOCKSIM executable file, LOCKSIM.EXE, are assumed to reside in the hard disk directory "C:\Wannabe." The first step in performing the simulation is to open an MS-DOS window, or boot to the MS-DOS command line, and set the current directory, using the MS-DOS CD command, to "C:\Wannabe." LOCKSIM is then started by issuing the following command:

```
C:\Wannabe\>LOCKSIM wannabe.sim
```

Figure 15 shows the screen displayed by LOCKSIM. The first several lines are introductory information. The next four nonblank lines list the available "Initial Commands," which are summarized in Table 9 in Part 2. The name of the current input file is correctly indicated as "wannabe.sim" near the bottom of Figure 15. The name of the current plot file is unknown until after either the "Run Steady" command (**s** or **S**) or the "Run Unsteady" command (**u**) is issued, both of which cause the input file to be read and processed.

For filling and emptying system simulations, the only initial commands frequently selected are "Run Steady" and "Run Unsteady."

Run Steady

As already mentioned, LOCKSIM assumes that the initial node heads and demands specified in the input file define a steady-state condition. The "Run Steady" command starts the

```
|====================== LOCKSIM version 1.00, July 1998 ======================|

     Hydraulic simulation of navigation lock filling and emptying systems.

Copyright 1998 -- G.A. Schohl, Tennessee Valley Authority. All rights reserved.

Initial Commands (Press Letter):
 (q) Quit              (i) Specify Input File    (P) Specify Plot File
 (s) Run Steady        (S) Run Steady (NC)       (u) Run Unsteady
 (p) Report Periods    (d) Report Destinations

Current Input File = "wannabe.sim"
Current Plot File  = ""
_
```

Figure 15: Initial screen displayed for Wannabe Simulation

```
Opening input file "wannabe.sim". . .
Reading *CONSTANTS. . .
Reading *COMPONENTS. . .
Reading *NODES. . .
Reading *FUNCTIONS. . .
Reading *CROSS_SECTIONS. . .
   "chamber". . .
Reading *PLOT_VARIABLES. . .
Ordering components for forward sweep. . .
Setting initial conditions and reordering links if necessary. . .
Balancing steady flows and pressures. . .

Iter =   1  MaxDQ =  0.00e+000  MaxDP =  0.00e+000  *CONVERGED*

Save Steady State (Y/N) ?
```

Figure 16: Text displayed when "Run Steady" is selected with input file "wannabe.sim"

steady-state simulator in LOCKSIM and gives the user the option to automatically write the steady-state node heads and demands into the input file as initial conditions for subsequent unsteady simulations. This feature can be used for filling and emptying systems but is usually unnecessary because the initial conditions are known (zero flow everywhere and known piezometric head elevations) and may be entered directly into the input file, as indicated in Figure 10. However, even in this case, the "Run Steady" command is useful as a means of checking that the specified initial conditions define a correct steady-state condition.

Figure 16 shows the text displayed by LOCKSIM when the "Run Steady" command is selected with "wannabe.sim" as the input file. The first several printed lines indicate progress in processing the input file. After the last section of the input file is read, the component, or link, order for the forward sweep, which refers to the first half of the matrix solution process implemented in LOCKSIM, is established. Finally, the specified starting, or initial, conditions for each component are computed based on the specified initial node heads and demands, and the iterative solution stage is entered. For the initial conditions specified in "wannabe.sim," only one iteration is necessary, with maximum discharge correction (MaxDQ) of zero and maximum pressure correction (MaxDP) of zero, which indicates that the specified initial node heads and demands define a steady-state condition. Otherwise, several iterations would be indicated with significant maximum corrections in the first iteration.

Because the initial conditions are already correct in "wannabe.sim," the appropriate response to the yes or no question at the bottom of Figure 16 is no (N). A no answer also to the next question (Steady State Report (Y/N)?) causes the text shown in Figure 15 to be redisplayed, at which time the "Run Unsteady" command can be selected.

Run Unsteady

Figure 17 shows the text displayed by LOCKSIM when the "Run Unsteady" command is selected with "wannabe.sim" as the input file. As for the steady-state simulator, the first several lines are printed during the preparation stage to indicate progress in processing the input file.

```
Opening input file " wannabe.sim ". . .
Reading *CONSTANTS. . .
Reading *COMPONENTS. . .
Reading *NODES. . .
Reading *FUNCTIONS. . .
Reading *CROSS_SECTIONS. . .
   "chamber". . .
Reading *PLOT_VARIABLES. . .
Ordering components for forward sweep. . .
Setting initial conditions and reordering links if necessary. . .
Checking initial flow balance at nodes. . . OK

Time = 0        sec, Step = 0          .....Simulation Suspended.....

Commands (Press Letter):
 (q) Quit           (s) Show Variable     (C) Change Timestep
 (c) Continue       (n) Next Suspension   (D) Debug Level
 (g) Go and Quit    (P) Print Reports     (d) Destinations      (p) periods
```

Figure 17: Text displayed when "Run Unsteady" is selected with input file "wannabe.sim"

After the last section of the input file is read, the component, or link, order for the forward sweep is established and the initial conditions for each component are computed based on the specified initial node heads and demands. The computed initial discharges into and out of each node are then added to check that inflow equals outflow. If all nodes pass this continuity test then "OK" is printed, as indicated in Figure 17. Then the current simulation time and time step (both initially zero), along with the phrase "Simulation suspended," is printed and a list of commands is displayed. Table 10 in Part 2 summarizes the available commands at a "simulation suspension."

If one or more nodes fail the continuity test, then the label for each node is listed along with the size of its discharge imbalance. The user is asked whether to continue the simulation or to quit. Unless the flow imbalance was deliberately imposed, the simulation should be terminated and restarted after the problem is fixed.

Two commands, "Next Suspension" (**n**) and "periods" (**p**), are usually issued before the unsteady simulation is continued beyond the initial steady-state. The "Next Suspension" (**n**) command is issued to set the time for the next simulation suspension, which is usually a time greater than the expected filling or emptying time of the lock but may be any time greater than zero. A simulation can always be continued from a simulation suspension after issuing a new "Next Suspension" command. For Wannabe Main Lock the next suspension time is specified as 900 (time units are seconds). The "periods" command (**p**) is issued to set the period of the Status report for Wannabe Main Lock to 30 time steps. The default value of 1 time step specifies printing of this report every two seconds, which is more information than needed and the frequent output can slow down the simulation. The Status report indicates the current simulation time and time step and lists commands that are available during the calculation phase of a simulation.

The unsteady calculations are started by issuing the "Continue" (**c**) command to continue the simulation. The text shown in Figure 18 is displayed as the calculations progress. A Status

```
Time = 60       sec, Step = 30

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports

Time = 120      sec, Step = 60

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports

Time = 180      sec, Step = 90

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports
       .
       .
       .
Time = 360      sec, Step = 180

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports

------ Simulation Time = 374      sec, Time Step = 187

Iter =  30  MaxDQ =  2.45e-002  Comp = C4 C5  Sect = 3
  ...Maximum convergence achieved.

Time = 420      sec, Step = 210

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports
       .
       .
       .
Time = 900      sec, Step = 450

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports

Time = 900      sec, Step = 450        .....Simulation Suspended.....

Commands (Press Letter):
 (q) Quit           (s) Show Variable     (C) Change Timestep
 (c) Continue       (n) Next Suspension   (D) Debug Level
 (g) Go and Quit    (P) Print Reports     (d) Destinations      (p) periods
```

Figure 18: Text displayed during filling simulation of Wannabe Main Lock

report is printed every 30 time steps (every 60 seconds). The commands that are available during the calculation phase of a simulation include the **s** command, which causes a simulation suspension to occur at the end of the current time step, and the **n**, **c**, and **y** commands, which cause the Nodes, Components, or Continuity reports (see the "OUTPUT REPORTS" section of Part 2), respectively, to be printed to the currently specified report destinations at the end of the current time step.

Figure 18 shows a warning message indicating that at simulation time equal to 374 seconds the solution did not converge to the specified solution tolerances. This message is not uncommon during simulations of networks containing tee and manifold components, which sometimes have convergence problems. The warning can be ignored when the maximum correction indicated is judged to be insignificantly small, even though it is larger than the

specified tolerance. Certainly a discharge error of 0.0245 cfs in the lock chamber component "C4 C5" is not enough error for alarm. As explained in the "OUTPUT REPORTS" section of Part 2, warning and error messages are printed both to the monitor and to a disk file named LOCKSIM.MSG, which can be viewed after the simulation is completed (but before the next one is started, at which time LOCKSIM.MSG is automatically deleted).

At time equal to 900 seconds, the simulation may be continued or terminated, as the user wishes. Because Wannabe Main Lock is full at 900 seconds, as verified by examination of the Nodes report (using the **P** command) or by use of the "Show Variable" (**s**) command to display the value of, for example, "C3 head," the simulation will be terminated by issuing the "Quit" (**q**) command. At this time, the plot file is saved. If the file already exists, the new plot data are written over the existing data, appended to the end of the existing data, or not written, depending on the value assigned to the *overwrite* keyword in the **CONSTANTS* section of the input file. With *overwrite* set to the default value of *warn*, the user is given the opportunity to provide a different file name before the existing file is overwritten.

Solution Convergence Problems

As already mentioned with respect to the "Maximum convergence achieved" message in Figure 18, the presence of tee and manifold components in a network sometimes leads to solution convergence problems. Nevertheless, these components are essential elements of filling and emptying system networks and useful results nearly always can be obtained.

Figure 19 shows an example of text that is often displayed at the end of a filling or emptying simulation of a network containing tees or manifolds. First, a message or messages indicate that convergence has not been achieved. Then an error described as "Negative or vapor pressure ..." occurs because the solution is diverging and the simulation is automatically suspended. The error, which sometimes occurs without the prior nonconvergence messages, is fatal and the simulation cannot be continued. Fortunately, this problem usually occurs only at the very end of a filling or emptying simulation, when the discharges through the tee and manifold branches are very small, with some negative and some positive. The simulation results are accurate for all time steps before the time step at which the convergence problem occurred and the plot file can be saved after the "Quit" (**q**) command is issued.

Another convergence problem often occurs when the filling or emptying valves are opened very fast (typically a minute or less). In this case, at some point after the valves are opened, and well before the lock is full or empty, "Maximum convergence achieved" messages are received with unacceptably large maximum corrections. Results can sometimes be obtained after restarting the simulation by suspending it the time step just before the error and using the "Change Timestep" (**C**) command to temporarily reduce the time step size. However, when this is necessary, the hawser forces estimated from the results inevitably indicate a very poor chamber water surface or, in other words, that the simulated valve opening time is too fast for good lock performance.

```
      .
      .
      .
Time = 600       sec, Step = 300

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports

Time = 660       sec, Step = 330

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports

------ Simulation Time = 668       sec, Time Step = 334

Iter = 150  MaxDQ =  6.89e+001  Comp = Lp6d Lmand1  Sect = 1
  ...Maximum number of iterations reached without convergence...

Iter = 150  MaxDQ =  6.89e+001  Comp = Lp6d Lmand1  Sect = 1
  ...Maximum convergence achieved.

<1> Negative or vapor pressure computed in cross section "chamber"
    for free surface component: "LC4 LC5".

Time = 672       sec, Step = 336        .....Simulation Suspended.....

Commands (Press Letter):
 (q) Quit            (s) Show Variable      (C) Change Timestep
 (c) Continue        (n) Next Suspension    (D) Debug Level
 (g) Go and Quit     (P) Print Reports      (d) Destinations       (p) periods

Write plot variables before quitting (Y/N) ?
```

Figure 19: Example display showing convergence problems at end of a filling simulation

Post Processing

LOCKSIM's most useful outputs are the time-varying data specified in the **PLOT_VARIABLES* section of the input file and saved at the end of the simulation to a plot file. Figure 20 shows the contents of the plot file "wannabe.plt" that was saved at the end of the filling simulation for Wannabe Main Lock. This plot file loads easily into spreadsheet software such as Microsoft Excel® for further analysis and plotting. When using Excel, load the plot file as "delimited" with "space" as the delimiter. Then select row 2, choose "Cells" from the "Format" menu, and, under the "Alignment" tab, select "Wrap Text." The columns can then be plotted with row 2 as the "series names," or "legend labels," row. Additional columns, with formulas to compute chamber water surface elevation and estimated hawser forces can be added. A completed spreadsheet can be used as a template for other plot files.

The following two sections present results from filling and emptying simulations of Wannabe Main Lock. The third section provides a detailed discussion of hawser force estimation, with emphasis on the effects of numerical and physical damping.

sec	(-)	(-)	(f)	(cfs)	(f)	(f)	(f)	(cfs)	(cfs)	(cfs)	(cfs)	(cfs)
Time	"Lfillu Lfilld valve_position sectno=0"			"Lfillu Lfilld cav_index sectno=0"				"Lfillu Lfilld vena_head sectno=0".				
0	0	0	215	0	15	15	15	0	0	0	0	0
2	.001289369	1.23044	218.9921	9.533626	15	15.00001	15	9.108212	0.4056315	0.01455752	.000605264	2.64006E-5
4	.002932236	1.21213	218.6933	21.73019	15.00004	15.00014	15	19.69694	1.892353	0.1362876	.009299717	.000575694
6	.004924074	1.211089	218.5891	36.36007	15.00041	15.00095	15	30.8392	4.805366	0.629842	0.07080252	.006242939
8	.007260357	1.194015	218.2797	53.64936	15.00237	15.00386	15	42.14694	9.568886	1.580985	0.3165329	0.04234851
10	.009936557	1.187442	218.0912	73.20081	15.00756	15.01015	15	52.7187	16.4996	2.865187	0.9105049	0.1971327
12	0.01294815	1.171444	217.7775	95.32414	15.01485	15.01785	15.00001	62.87115	25.23254	4.835251	1.737185	0.6532906
14	0.0162906	1.161545	217.5378	119.5726	15.0247	15.02723	15.00005	72.32104	34.93134	8.178343	2.539206	1.593287
16	0.0199594	1.147051	217.2378	146.2141	15.03535	15.03801	15.00027	81.56343	44.90557	13.3032	3.457295	2.987774
18	0.02395	1.13626	216.9901	174.8405	15.04705	15.04969	15.00106	90.2739	54.68497	20.25061	4.916201	4.706144
20	0.02825789	1.124113	216.7307	205.6032	15.06004	15.06217	15.00336	98.74277	64.11072	28.64879	7.365063	6.736838
22	0.03287853	1.114716	216.5207	238.1657	15.07434	15.07549	15.00861	106.7081	73.1422	37.87642	11.29245	9.138472
24	0.03780741	1.106077	216.336	272.5443	15.09	15.09011	15.01809	114.473	81.69031	47.29589	17.03421	12.0499
26	0.04303999	1.100113	216.209	308.465	15.10676	15.10649	15.03217	121.9736	89.79527	56.4255	24.50419	15.75897
28	0.04857174	1.096646	216.1437	345.8026	15.12457	15.12476	15.05048	129.1279	97.65924	65.12021	33.25339	20.63886
30	0.05439815	1.096438	216.1539	384.3189	15.14334	15.14429	15.07271	135.7893	105.3279	73.54808	42.6781	26.96802
32	0.06051468	1.098831	216.2256	423.905	15.16322	15.16481	15.09829	142.0757	112.8201	81.90242	52.3508	34.75135
34	0.06691681	1.103165	216.3421	464.47	15.18414	15.18673	15.12739	148.1159	120.1254	90.26032	62.22523	43.73502
36	0.0736	1.10866	216.485	505.9679	15.20606	15.21053	15.16	154.1305	127.2143	98.71447	72.26758	53.63465
38	0.08055974	1.114766	216.6413	548.3583	15.22888	15.23701	15.19619	160.1546	134.3175	107.2802	82.37647	64.22047
40	0.0877915	1.121093	216.8017	591.6031	15.2526	15.26681	15.23598	166.1154	141.6185	116.0109	92.58187	75.26785
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
860	1	14527.18	265.1644	-91.08618	65.42031	65.41338	65.41099	-86.88551	-62.67622	-1.276239	27.29644	32.45522
862	1	11456.46	265.15	-102.5613	65.41187	65.40811	65.40907	-93.82364	-57.87028	-7.898733	25.93237	31.10069
864	1	9397.382	265.1393	-113.2347	65.40219	65.40241	65.40668	-97.41737	-56.53281	-13.4976	24.53418	29.67914
866	1	7926.995	265.1301	-123.284	65.3923	65.39564	65.40344	-99.31414	-56.5527	-18.6206	22.99128	28.21367
868	1	6830.386	265.1221	-132.8064	65.38345	65.38871	65.39872	-100.2509	-57.16328	-23.4251	21.336	26.69734
870	1	5984.022	265.1151	-141.8824	65.3756	65.3815	65.39211	-100.6979	-57.93623	-28.02731	19.61382	25.1666
872	1	5313.204	265.1086	-150.5675	65.36832	65.37349	65.38348	-100.9406	-58.72206	-32.34981	17.75255	23.69317
874	1	4770.295	265.1027	-158.8995	65.36201	65.36513	65.37282	-101.1348	-59.48737	-36.32977	15.76356	22.29011
876	1	4320.218	265.0985	-166.9676	65.35649	65.35688	65.36056	-100.0187	-63.59752	-37.41905	13.12588	20.94215
878	1	3940.046	265.0945	-174.8335	65.35091	65.34805	65.34766	-98.7065	-68.83978	-37.15858	10.18797	19.685
880	1	3618.518	265.0886	-182.4298	65.34453	65.33823	65.33473	-98.48452	-72.00166	-37.91721	7.424486	18.54983
882	1	3347.225	265.082	-189.6719	65.3368	65.3282	65.32216	-98.72997	-74.08041	-39.02226	4.648437	17.51384
884	1	3118.394	265.0745	-196.4997	65.32736	65.31784	65.31053	-99.10559	-75.52146	-40.19312	1.751392	16.57003
886	1	2925.645	265.0665	-202.8604	65.31637	65.30711	65.30012	-99.37074	-76.56515	-41.2428	-1.408053	15.72775
888	1	2763.281	265.0582	-208.7256	65.30393	65.29606	65.29052	-99.42661	-77:27751	-42.12118	-4.82781	14.92852
890	1	2626.44	265.05	-214.0843	65.28995	65.28461	65.28145	-99.22572	-77.70049	-42.83764	-8.402922	14.08384
892	1	2510.996	265.042	-218.9406	65.27499	65.27293	65.2728	-98.7629	-77.87086	-43.41072	-12.03031	13.13526
894	1	2413.329	265.0345	-223.3177	65.25993	65.26078	65.26401	-97.98888	-77.07201	-45.5958	-14.63363	11.97355
896	1	2330.364	265.0277	-227.2496	65.24536	65.24857	65.25425	-96.89127	-75.90273	-48.51038	-16.56912	10.62534
898	1	2259.629	265.0214	-230.771	65.23169	65.23644	65.24316	-95.62218	-75.10309	-50.53892	-18.65415	9.148422
900	1	2199.206	265.0155	-233.9121	65.21912	65.22395	65.2307	-94.32587	-74.43916	-51.97537	-20.72879	7.558447

Figure 20: Contents of plot file "wannabe.plt" after filling simulation for Wannabe Main Lock

Filling Results for Wannabe Main Lock

Figure 21 summarizes the results of the filling simulation for Wannabe Main Lock. The filling valves were opened in 3 minutes. The filling time, defined as the time between initiation of filling valve opening to the time at which the water surface elevation in the center of the lock first passed the headwater elevation, was 12.93 minutes (linear interpolation between time steps). Discharges through the first, second, and fourth portholes from the upstream end on the land-side of the lock are plotted. Discharges from the third and fifth portholes were in between those from the first and fourth portholes. The minimum cavitation index was about 0.87 (criterion for acceptable minimum varies between 0.6 and 1.0, depending on reference). Estimated longitudinal hawser forces, based on a tow weight of 15600 tons, were less than 4 tons (in physical model studies, the usual criterion for acceptable performance is maximum hawser force of 5 tons).

The results in Figure 21 indicate that the fictional Wannabe Main Lock has acceptable filling performance, as measured by longitudinal hawser forces and cavitation index, but these results should not be taken too seriously. The use of five large portholes in each wall to fill and empty a navigation lock is not a good design. The primary reason that Wannabe Main Lock appears acceptable is that the loss coefficients assumed for the combining and dividing flow through the portholes are unrealistically large compared to those that would be measured for the real portholes. For convenience, the default loss coefficients, which apply to standard tees in circular piping systems, were assumed for the portholes. In reality, the short, well-rounded portholes sketched in Figure 3 would restrict flow less than standard tees, resulting in faster filling of the lock, a poorer distribution of flow into the chamber, and lower values of the cavitation index below the filling valves. To obtain realistic results, it is important to use realistic data in specifying loss coefficients for dividing and combining flow through the lock portholes. The "EXAMPLE APPLICATIONS" section of this guide provides realistic tee coefficients for multiport locks and for one other specific porthole geometry. In general, laboratory experiments are required to obtain loss coefficients for particular portholes.

The limitations of LOCKSIM should also be remembered when interpreting results such as those in Figure 21. An acceptable lock from the point of view of longitudinal surface slope may still have unacceptable currents and turbulence within the chamber, which is likely for a lock fed by five large portholes on each side. Also, the longitudinal surface slopes predicted for filling through large diameter, relatively short, portholes are subject to error because the flow into the chamber is likely to enter with significant downstream momentum, which, as already discussed (under the heading "Lock Chamber" in the "Input Data" section of this guide), is not represented by the *open_channel* and *river_channel* components in LOCKSIM.

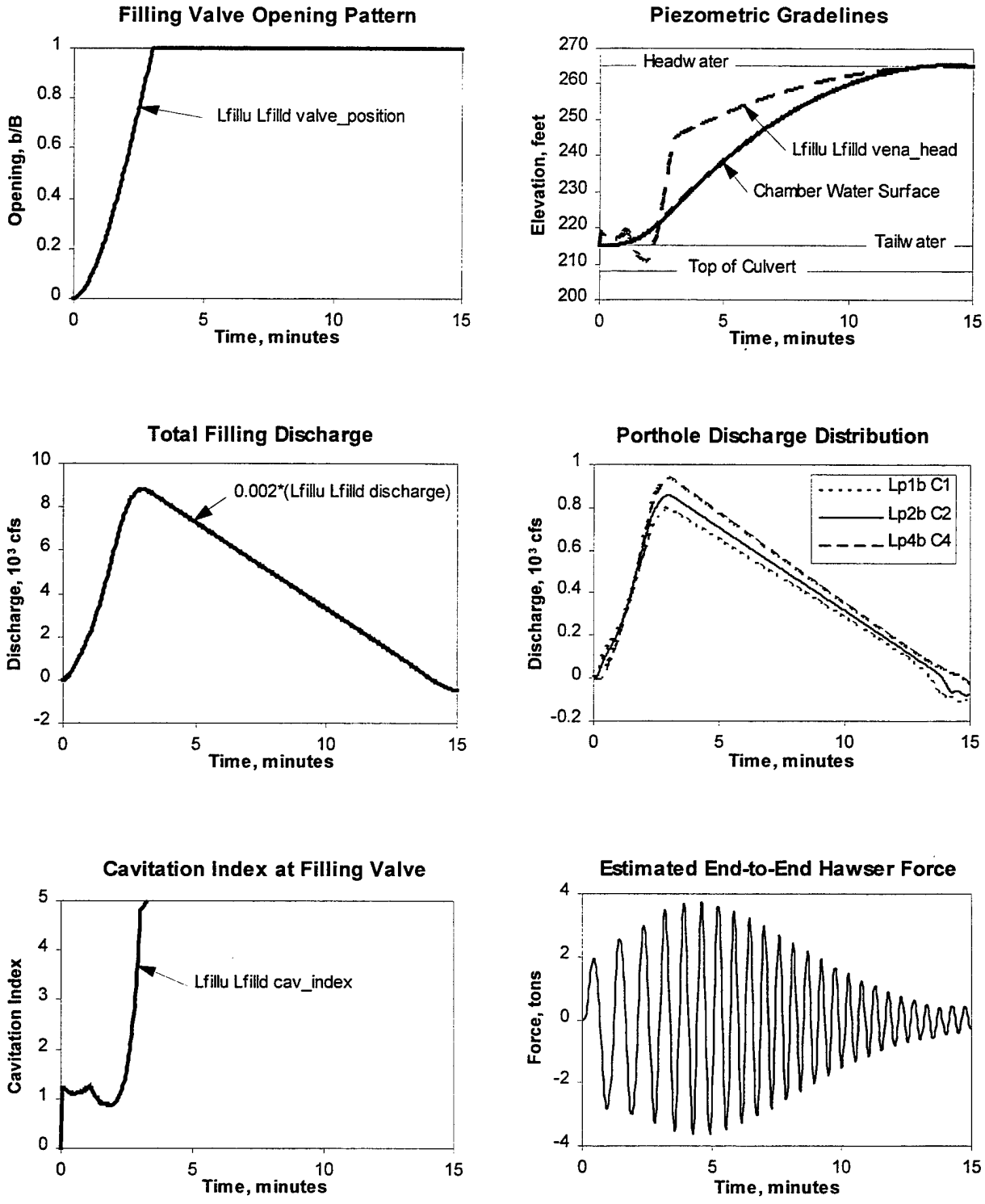


Figure 21: Filling results for Wannabe Main Lock

Emptying Results for Wannabe Main Lock

Figure 22 summarizes the results of the emptying simulation for Wannabe Main Lock. The emptying valves were opened in 1.5 minutes. The emptying time, defined as the time between initiation of emptying valve opening to the time at which the water surface elevation in the center of the lock first passed the tailwater elevation, was 11.65 minutes (linear interpolation between time steps). Discharges through the first, third, and fifth portholes from the upstream end on the land-side of the lock are plotted. Discharges from the second and fourth portholes were in between those from the first and fifth portholes. The minimum cavitation index was about 0.66. Estimated longitudinal hawser forces were less than 1.5 tons. For simulation times greater than about 10 minutes, the hawser force trace in Figure 22 appears to be physically unrealistic. This behavior, which occurs only after the lock is empty (or full), is apparently a numerical effect and can be ignored.

The comments in the preceding section, concerning the interpretation of the results in Figure 21 and the limitations of LOCKSIM, apply also to the results in Figure 22.

Estimating Longitudinal Hawser Forces

When a lock chamber is modeled using *open_channel* or *river_channel* components, longitudinal hawser forces on a tow moored in the chamber may be estimated from time-varying differences in water surface elevation computed by LOCKSIM. The estimated forces, like measured hawser forces, typically oscillate at the frequency of a first mode longitudinal standing wave in the chamber. The computed hawser forces are approximate, not only because they are, by definition, estimates, but also because, except for the first few oscillatory peaks, their magnitudes depend significantly on the physical damping modeled by LOCKSIM compared with the numerical damping present in the implicit solution technique. The physical damping is under-predicted by LOCKSIM (see discussion below) and the numerical damping varies with the time step size, resulting in hawser force magnitudes that depend on the time step size. Fortunately, the magnitudes of the first few oscillatory peaks, which are usually the largest when hawser forces are measured, are physically correct, repeatable, and converge to common values as the time step size is reduced. These magnitudes are useful for comparing different design and operation options for a filling and emptying system.

Hawser force is estimated by resolving the weight of a tow down the slope of the water surface under the tow (Li, 1965; Schohl, 1978). For a tow moored in a lock chamber, the longitudinal hawser force, F_H, is

$$F_H = W\frac{H_u - H_d}{L_{ud}} \tag{3}$$

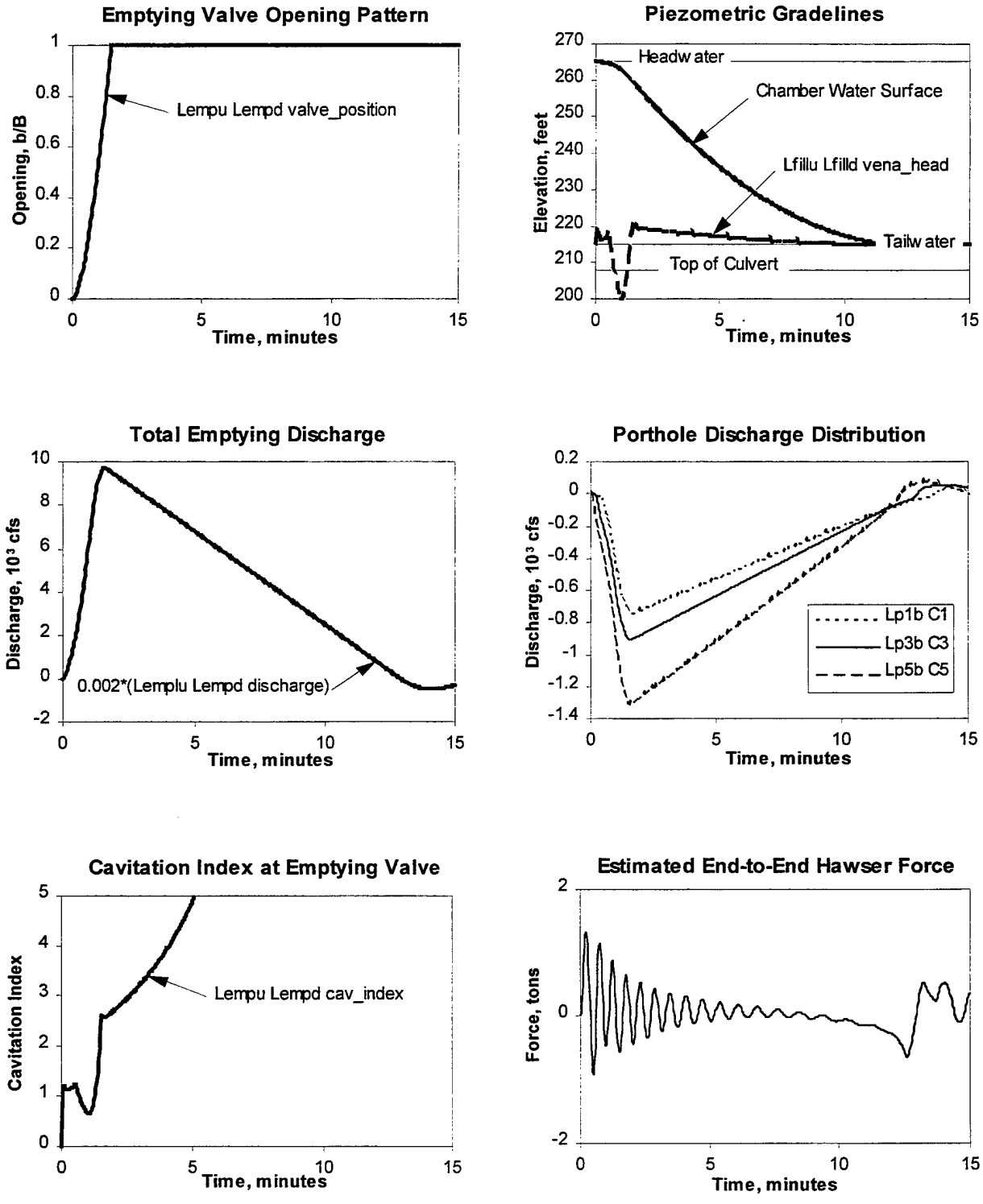


Figure 22: Emptying results for Wannabe Main Lock

in which W = weight of tow (or weight of water displaced by tow); H_u = water surface elevation, or piezometric head, at point u near upstream end of tow; H_d = water surface elevation, or piezometric head, at point d near downstream end of tow; and L_{ud} = horizontal distance between points u and d. Because of computer precision limitations in calculating small differences in large numbers, an alternative expression to Equation 3 is recommended when results from LOCKSIM are used for estimating hawser forces:

$$F_H = W\frac{(d_u - d_d) + (z_{bu} - z_{bd})}{L_{ud}} \tag{4}$$

in which d_u = chamber depth at point u, d_d = chamber depth at point d, z_{bu} = chamber floor elevation at point u, and z_{bd} = chamber floor elevation at point d. Because simulation results are available only at nodes and interior computational section, points u and d must correspond to the location of a node or interior computational section. This should be kept in mind when the *open_channel* components representing the lock chamber are divided into computational reaches. However, it is not necessary to make sure that points u and d are at precise locations relative to the ends of a tow. The water slopes computed using water surface elevations 600 feet apart in the lock chamber will be nearly the same as those computed using water surface elevations 630 or 570 feet apart. Remember also that the computed forces are used primarily for comparison purposes rather than as accurate predictions of hawser forces.

While this discussion has focused on longitudinal hawser forces in a lock chamber, it should be obvious that Equation 4 can be applied also to tows moored in an upstream or downstream approach channel for which computed or measured water surface slopes are available.

Figure 23 shows hawser forces computed for filling of Wannabe Main Lock with different values of the simulation time step. Unfortunately, except for the first few peaks, the predicted hawser force traces depend on the time step value, with no convergence to a common trace as the time step is reduced to smaller and smaller values. In contrast, other important results for Wannabe Main Lock, for example cavitation index and minimum pressure downstream from the filling valves, porthole discharge distribution, and mean lock water surface elevation, are nearly identical for all values of the time step under 10 seconds.

The behavior exhibited in Figure 23 occurs because the numerical damping present in the implicit solution is reduced as the time step is reduced. The effect would be much less significant if the physical damping modeled by LOCKSIM were larger compared with the numerical damping. In LOCKSIM, the only physical damping mechanism modeled in a lock chamber is bed friction based on steady-state, turbulent flow (e.g., the Manning Equation). This treatment of friction is standard for unsteady flow modeling because better techniques are generally unavailable, but it under-predicts physical damping of oscillatory, standing waves in a lock chamber.

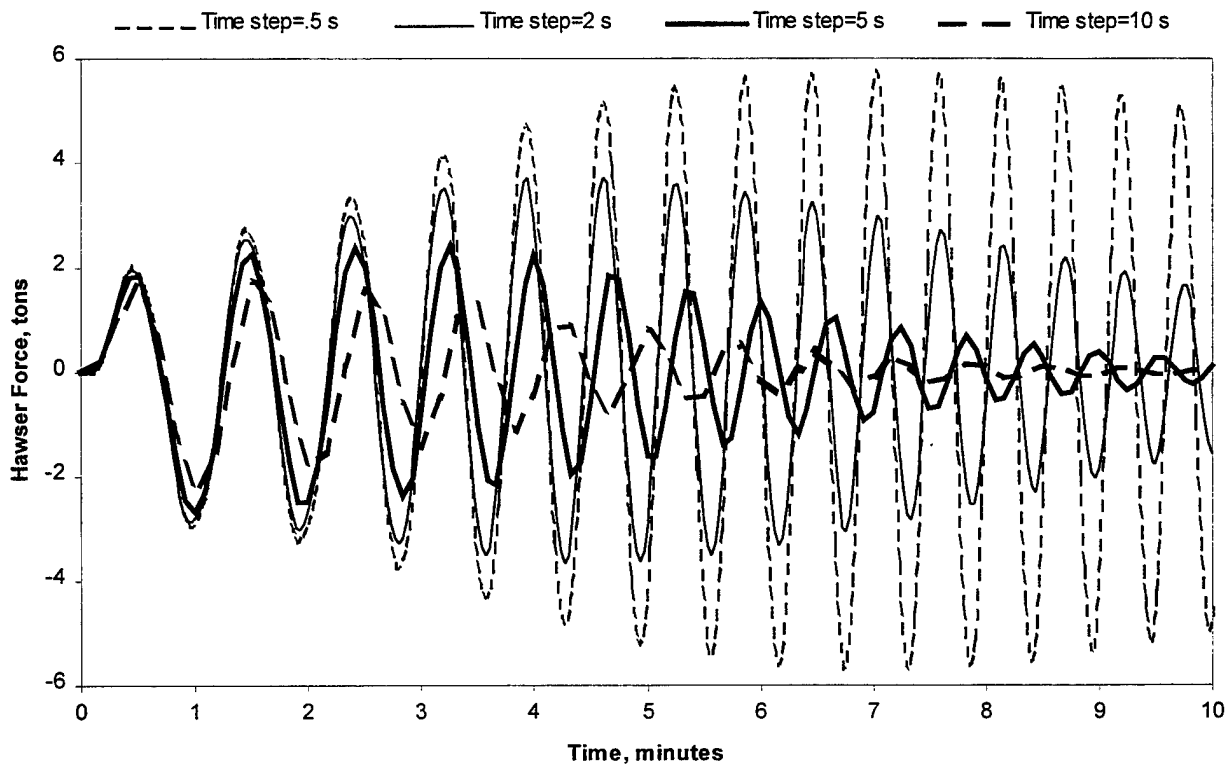


Figure 23: Dependence of estimated hawser forces on simulation time step (Wannabe Main Lock)

The recommended approach to dealing with the problem of different hawser forces for different time steps is to use the same time step for every simulation so that all are affected by similar numerical damping, and to use only the magnitudes of the first few oscillatory peaks for comparing hawser forces. As evident in Figure 23, the magnitudes of the first several peaks depend only slightly on the time step size, and data often indicate that the first few peaks are the largest. If desired, numerical damping can be increased by using a larger time step or implicit weighting factor, θ (see the "*moc_pipe* and *imp_pipe*" section in Part 2). The author has often used a time step of five seconds, because for that time step the damping affecting the computed hawser force traces usually appears about right compared to available model and prototype measurements. Physical damping may be increased by using a larger friction coefficient (e.g., Manning n) or a smaller momentum correction factor, β (setting β to zero is equivalent to assuming that all changes in longitudinal velocity head in the chamber are dissipated). The effects of increased physical damping are illustrated in Figure 24 for a fixed time step of 0.5 seconds. At these levels of physical damping, the results are still sensitive to the time step size but not as much as indicated in Figure 23. Increasing physical damping is a reasonable calibration strategy when prototype or model data are available for comparison. It is good practice to compare computed hawser forces for different time steps before arbitrarily increasing physical damping.

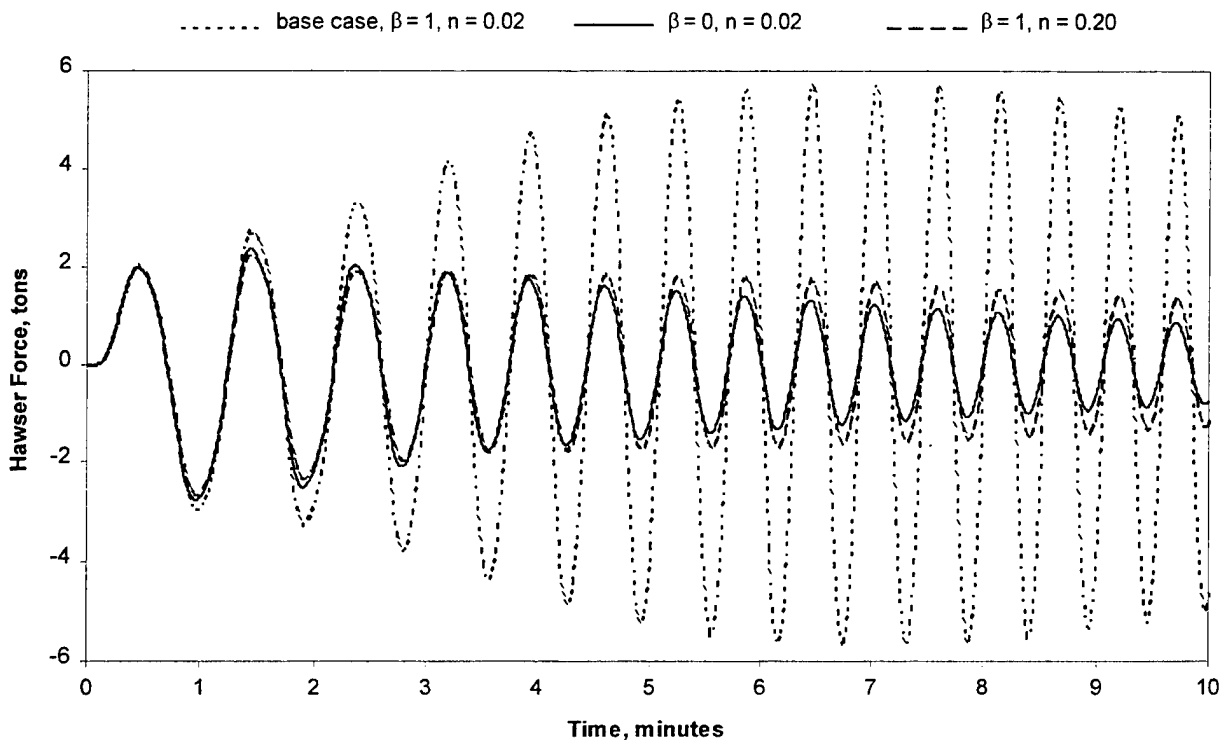


Figure 24: Estimated hawser forces with increased physical damping (Δt=0.5 seconds; Wannabe Main Lock)

EXAMPLE APPLICATIONS

Multiport Lock

TVA's Wheeler Main Lock is used for the multiport example application because field data are available for comparison to simulation results (Schohl, 1994) and because tee coefficients for the Wheeler porthole and manifold geometry are known from laboratory data.

Figure 25 approximately depicts the multiport filling and emptying system for Wheeler Main Lock. The clear surface area of the lock chamber is 110 feet wide by 600 feet long, large enough to accommodate nine jumbo barges. The maximum lift of the lock is approximately 52 feet, which corresponds to the normal maximum headwater elevation of 556.3 feet and the minimum tailwater elevation of 504.5 feet. Upstream, water for filling the lock enters longitudinal wall culverts through bell-mouth inlets covered by trashracks. Both filling and emptying flows are controlled by reverse tainter valves in the wall culverts. Bulkhead slots are provided both upstream and downstream from the valves for de-watering purposes. Flow enters

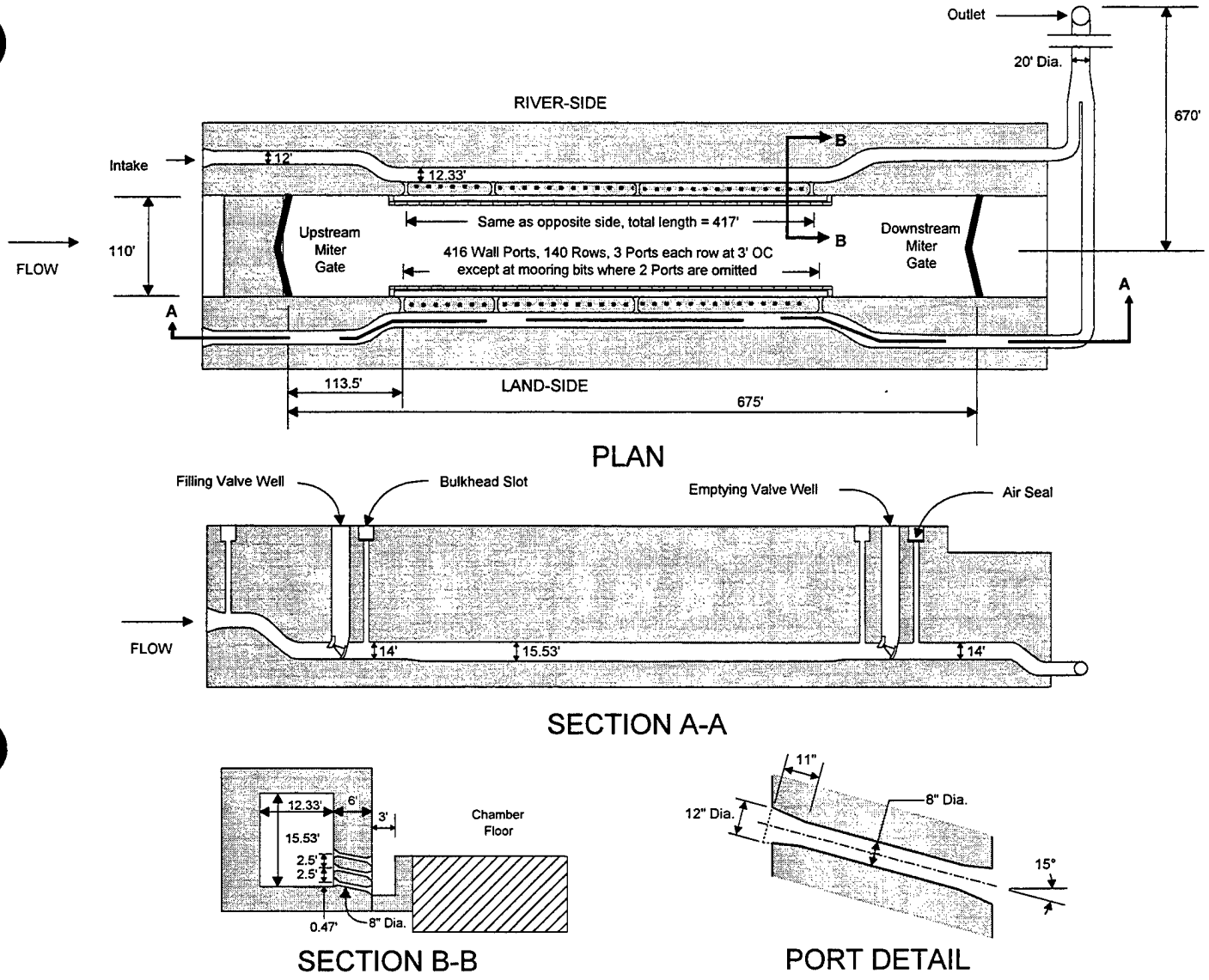


Figure 25: Conceptual illustration of multiport filling and emptying system for Wheeler Main Lock

and exits the lock chamber through 416 8-inch diameter portholes distributed along the length of each longitudinal wall culvert. Downstream, water emptying the lock through the wall culverts combines into one 20-foot diameter tunnel that discharges vertically into the tailwater about 450 feet from the river-side of the lock.

Figure 26 illustrates the schematic network of components and nodes that represents Wheeler Main Lock in LOCKSIM. The numbers in parentheses near the node labels are centerline elevations (for closed conduit components) or bottom elevations (for free-surface components) in feet. The numbers above the *imp_pipe* and *open_channel* components are lengths. The numbers below many of the *imp_pipe* components indicate cross-sectional dimensions (width times height). The filling and emptying port manifolds are represented between nodes "Lmanu" and "Lmand" on the land-side and "Rmanu" and "Rmand" on the river-

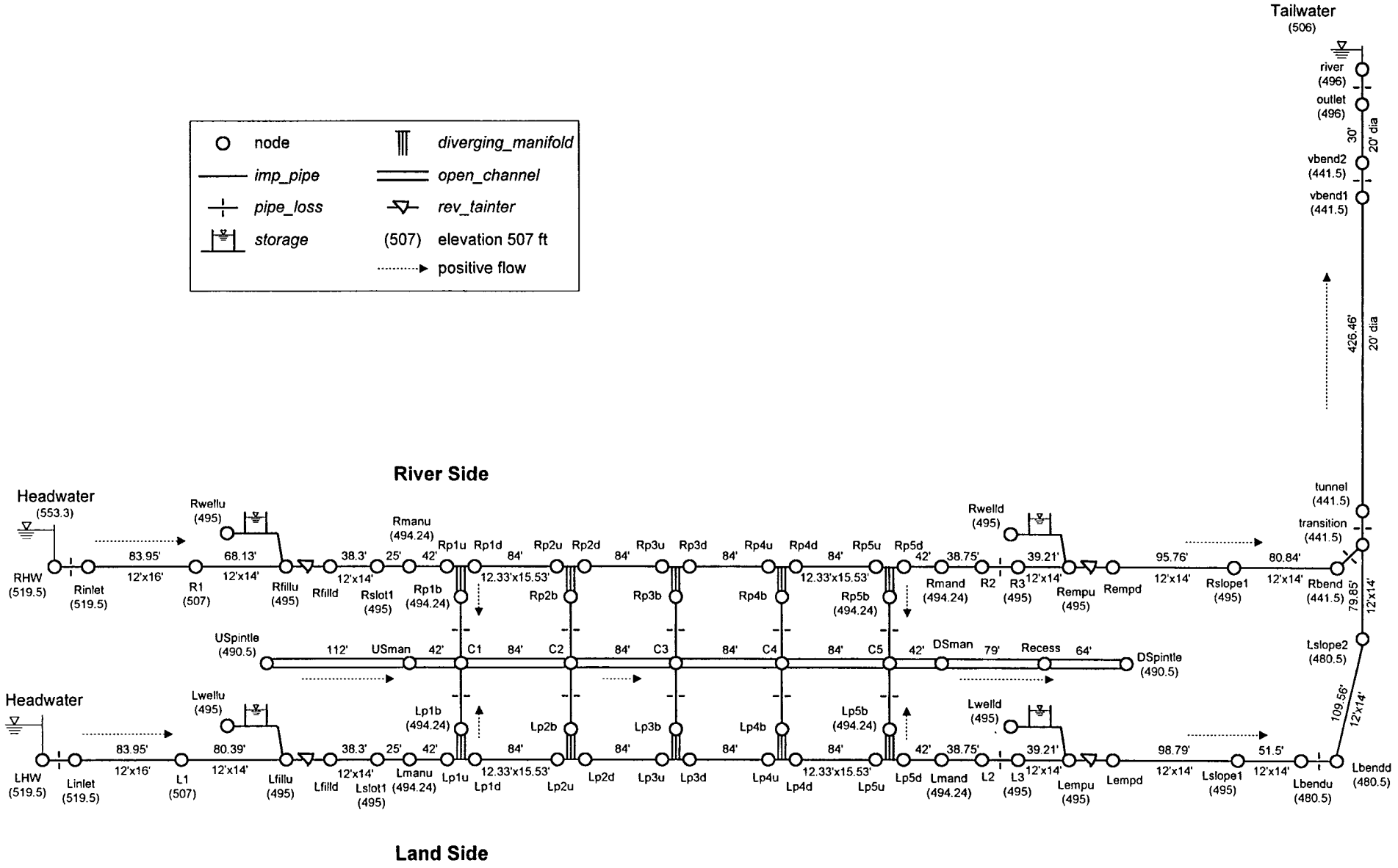


Figure 26: Schematic representation of Wheeler Main Lock

side. On each side, the 416 portholes are represented by five *diverging_manifold* components connected to *pipe_loss* components. Each *diverging_manifold* component is located at the center of the 83.2 portholes that it represents. The lock chamber is represented by a series of *open_channel* components between nodes "USpintle" and "DSpintle." The filling valves are included between nodes "Lfillu" and "Lfilld" and between nodes "Rfillu" and "Rfilld." The emptying valves are included between nodes "Lempu" and "Lempd" and between nodes "Rempu" and "Rempd." Water *storage* components, each with cross-sectional area equal to 336 ft^2, are connected to the upstream nodes of each valve to represent the valve wells.

Figure 27 illustrates the input file "wheeler.sim." The loss coefficient for the inlets was determined from calibration to measured water surface elevations in the land-side filling valve well. The opening profiles for the land-side and river-side filling valves are different because during the field tests the river-side filling valve opened about 4 seconds faster than the land-side filling valve. The opening profiles for the land-side and river-side emptying valves are the same because these valves opened in approximately the same time. The *open_channel* component "Recess DSpintle" uses a cross section with a larger surface area than the other *open_channel* components to account for the gate recesses at the downstream end of the lock chamber. An initial condition for filling, either initial head (*ihead*) or initial demand (*idemand*), is specified for every node. A boundary condition, either *head* or *demand* is specified for every terminal node. Functions are specified for opening the filling and emptying valves and for their discharge and contraction coefficients. The remaining functions specify the tee coefficients for the *diverging_manifold* components.

The *pipe_loss* components "L2 L3" and "R2 R3" each represent the combined effect of a slight flow contraction plus two 20-degree bends. The *pipe_loss* component "Lbendu Lbendd" represents the combined effect of a 90-degree bend, two 5-degree bends, and two 28-degree bends. The *pipe_loss* component "Rbend transition" represents the combined effect of a 90-degree bend and two 35-degree bends. The *pipe_loss* component "transition tunnel" represents the flow contraction from two 12-foot by 14-foot culverts to one 20-foot diameter conduit. The *pipe_loss* component "vbend1 vbend2" represents the 90-degree vertical bend near the outlet.

Each *diverging_manifold* component has 28 branches with the cross-sectional area of each branch equal to 1.0372 ft^2 for a total branch area of 29.042 ft^2. Each *pipe_loss* component connecting a *diverging_manifold* component to the lock chamber has a cross-sectional area of 29.042 ft^2. Because each *diverging_manifold* component represents 83.2 portholes and each "branch" of the Wheeler porthole manifold consists of three portholes (see Figure 25), the precise number of manifold branches is one-third of 83.2, or 27.73. However, the number of branches must be an integer so 27.73 branches is rounded up to 28 branches and the cross-sectional area of 83.2 portholes divided by 28 is assigned to each branch.

The most important tee and loss coefficients for the *diverging_manifold* and *pipe_loss* components that represent the porthole manifolds were measured in laboratory experiments of combining and dividing flow in a model of the Wheeler Main Lock porthole and culvert geometry. The loss coefficients for each *pipe_loss* component are equal to the exit loss coefficient for flow from a porthole into the lock chamber, which averaged about 0.64 in the experiments, and to the entrance loss coefficient for flow from the lock chamber into a porthole,

```
!Wheeler Main Lock Filling and Emptying System                4/29/97

!  Field Test Conditions
!           Filling, 4.5 minute valve, HW=553.3, TW=506.0
!           Emptying, 1.5 minute valve, HW=553.3, TW=506.0
!           Comparison with prototype data
!  Use port model data with 5 manifold components

*CONSTANTS ============================================================
  time_step=2, wf_time=.55, gravity=32.146
  dQ_max=.001, dH_max=.0001, dX_max=.0001
  plot_field=11, plot_line=200, plot_file=wheeler.plt, plot_labels=row

*COMPONENTS ===========================================================

!----------- Inlet and fill valves ----------------------------------
LHW Linlet  pipe_loss us_area=1e5, area=192, K+=0.70  !from calibration
Linlet  L1  imp_pipe dia=13.71 area=192 len=83.95 rough=.003 wave=3500
L1   Lfillu imp_pipe dia=12.92 area=168 len=80.39 rough=.003 wave=3500
Lfillu Lwellu  storage  surface_area=336  max_wsel=566.3
Lfillu Lfilld  rev_tainter   b=14, w=12, el_bottom=488
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    b/B_vs_t=Open_LS_Fill   ! filling
   !fixed_b/B=0             ! emptying

RHW Rinlet  pipe_loss us_area=1e5, area=192, K+=0.70  !from calibration
Rinlet  R1  imp_pipe dia=13.71 area=192 len=83.95 rough=.003 wave=3500
R1   Rfillu imp_pipe dia=12.92 area=168 len=68.13 rough=.003 wave=3500
Rfillu Rwellu  storage  surface_area=336  max_wsel=566.3
Rfillu Rfilld  rev_tainter   b=14, w=12, el_bottom=488
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    b/B_vs_t=Open_RS_Fill   ! filling
   !fixed_b/B=0             ! emptying

!------------ Land wall culverts and ports ---------------------------
Lfilld Lslot1 imp_pipe dia=12.92 area=168 len=38.3 rough=.003 wave=3500
Lslot1 Lmanu imp_pipe dia=13.33 area=180   len=25  rough=.003 wave=3500

Lmanu Lp1u  imp_pipe dia=13.75 area=191.5 len=42 wave=3500 rough=.003
Lp1d  Lp2u  imp_pipe dia=13.75 area=191.5 len=84 wave=3500 rough=.003
Lp2d  Lp3u  imp_pipe dia=13.75 area=191.5 len=84 wave=3500 rough=.003
Lp3d  Lp4u  imp_pipe dia=13.75 area=191.5 len=84 wave=3500 rough=.003
Lp4d  Lp5u  imp_pipe dia=13.75 area=191.5 len=84 wave=3500 rough=.003
Lp5d  Lmand imp_pipe dia=13.75 area=191.5 len=42 wave=3500 rough=.003

          ! each manifold represents 83.2 portholes
Lp1u Lp1d Lp1b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10
Lp2u Lp2d Lp2b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10
Lp3u Lp3d Lp3b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10
Lp4u Lp4d Lp4b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10
Lp5u Lp5d Lp5b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10

Lp1b C1 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10
Lp2b C2 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10
Lp3b C3 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10
Lp4b C4 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10
Lp5b C5 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10

Lmand L2 imp_pipe dia=13.33, area=180, len=38.75, rough=.003, wave=3500
L2    L3 pipe_loss  us_area=191.5, area=168, K+=0.12
L3 Lempu imp_pipe dia=12.92, area=168, len=39.21, rough=.003, wave=3500
Lempu   Lwelld  storage surface_area=336  max_wsel=566.3

!------------ River wall culverts and ports --------------------------
Rfilld Rslot1 imp_pipe dia=12.92 area=168 len=38.3 rough=.003 wave=3500
Rslot1 Rmanu imp_pipe dia=13.33 area=180   len=25  rough=.003 wave=3500

Rmanu Rp1u  imp_pipe dia=13.75 area=191.5 len=42 wave=3500 rough=.003
Rp1d  Rp2u  imp_pipe dia=13.75 area=191.5 len=84 wave=3500 rough=.003
Rp2d  Rp3u  imp_pipe dia=13.75 area=191.5 len=84 wave=3500 rough=.003
Rp3d  Rp4u  imp_pipe dia=13.75 area=191.5 len=84 wave=3500 rough=.003
Rp4d  Rp5u  imp_pipe dia=13.75 area=191.5 len=84 wave=3500 rough=.003
Rp5d  Rmand imp_pipe dia=13.75 area=191.5 len=42 wave=3500 rough=.003

          ! each manifold represents 83.2 portholes
Rp1u Rp1d Rp1b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
```

Figure 27: LOCKSIM input file for Wheeler Main Lock (file "wheeler.sim")

```
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10
Rp2u Rp2d Rp2b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10
Rp3u Rp3d Rp3b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10
Rp4u Rp4d Rp4b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10
Rp5u Rp5d Rp5b diverging_manifold area1=191.5 area2=1.0372 branches=28
  Kd31_vs_Qr=Kd13_vs_Qr=kd31   Kd32_vs_Qr=Kd12_vs_Qr=kd32  iq1=0  iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31   Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=.70
  fixed_Kc32=fixed_Kc12=.10

Rp1b C1 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10
Rp2b C2 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10
Rp3b C3 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10
Rp4b C4 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10
Rp5b C5 pipe_loss us_area=area=29.042, ds_area=1e5, K+=.64, K-=.10

Rmand R2 imp_pipe dia=13.33, area=180, len=38.75, rough=.003, wave=3500
R2    R3 pipe_loss  us_area=191.5, area=168, K+=0.12
R3 Rempu imp_pipe dia=12.92, area=168, len=39.21, rough=.003, wave=3500
Rempu   Rwelld  storage surface_area=336  max_wsel=566.3

!------------ Lock chamber modeled as series of open channels ---------
USpintle USman    open_channel cross_section=chamber len=112 reaches=6
USman    C1       open_channel cross_section=chamber len=42  reaches=3
C1       C2       open_channel cross_section=chamber len=84  reaches=4
C2       C3       open_channel cross_section=chamber len=84  reaches=4
C3       C4       open_channel cross_section=chamber len=84  reaches=4
C4       C5       open_channel cross_section=chamber len=84  reaches=4
C5       DSman    open_channel cross_section=chamber len=42  reaches=2
DSman    Recess   open_channel cross_section=chamber len=79  reaches=4
Recess   DSpintle open_channel cross_section=rchamber len=64 reaches=4

!------------------------Empty valves and outlet-----------------------
Lempu   Lempd  rev_tainter   b=14, w=12, el_bottom=488
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
   !b/B_vs_t=Open_empty    ! emptying
    fixed_b/B=0            ! filling

Lempd Lslope1 imp_pipe dia=12.92 area=168 len=98.79 rough=.003 wav=3500
Lslope1 Lbendu imp_pipe dia=12.92 area=168 len=51.5 rough=.003 wav=3500
Lbendu  Lbendd  pipe_loss  area=168  K+=0.33
Lbendd  Lslope2 imp_pipe dia=12.92 area=168 len=109.56 rough=.003
  wave=3500
Lslope2 transition imp_pipe dia=12.92 area=168 len=79.85 rough=.003
  wave=3500

Rempu   Rempd  rev_tainter   b=14, w=12, el_bottom=488
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
   !b/B_vs_t=Open_empty    ! emptying
    fixed_b/B=0            ! filling

Rempd Rslope1 imp_pipe dia=12.92 area=168 len=95.76 rough=.003 wav=3500
Rslope1 Rbend imp_pipe dia=12.92 area=168 len=80.84 rough=.003 wav=3500
Rbend   transition  pipe_loss  area=168  K+=0.33

!------------------------ Outlet Tunnel -------------------------------

transition tunnel pipe_loss  us_area=336 area=314.2  K+=.006
tunnel vbend1 imp_pipe dia=20 area=314.2 len=426.46 rough=.003 wav=3500
vbend1 vbend2 pipe_loss  area=314.2  K+=0.16
vbend2 outlet imp_pipe dia=20 area=314.2 len=30 rough=.003 wave=3500
outlet river  pipe_loss  us_area=area=314.2  ds_area=1e8  K+=1  K-=.5

*NODES ==========================================================

!------ initial conditions for filling ------------
LHW     elev=519.5    head=553.3  idemand=0
Linlet  elev=519.5   ihead=553.3
L1      elev=507     ihead=553.3
Lwellu  elev=495     ihead=553.3   demand=0
Lfillu  elev=495     ihead=553.3
Lfilld  elev=495     ihead=506
Lslot1  elev=495     ihead=506

Lmanu  elev=494.24   ihead=506
Lp1u   elev=494.24   ihead=506
Lp2u   elev=494.24   ihead=506
Lp3u   elev=494.24   ihead=506
Lp4u   elev=494.24   ihead=506
Lp5u   elev=494.24   ihead=506
Lp1d   elev=494.24   ihead=506
Lp2d   elev=494.24   ihead=506
Lp3d   elev=494.24   ihead=506
Lp4d   elev=494.24   ihead=506
Lp5d   elev=494.24   ihead=506
```

Figure 27 (continued): LOCKSIM input file for Wheeler Main Lock (file "wheeler.sim")

```
Lp1b    elev=494.24   ihead=506
Lp2b    elev=494.24   ihead=506
Lp3b    elev=494.24   ihead=506
Lp4b    elev=494.24   ihead=506
Lp5b    elev=494.24   ihead=506
Lmand   elev=494.24   ihead=506

L2      elev=495   ihead=506
L3      elev=495   ihead=506
Lempu   elev=495   ihead=506
Lwelld  elev=495   ihead=506  demand=0

Lempd   elev=495   ihead=506
Lslope1 elev=495   ihead=506
Lbendu  elev=480.5 ihead=506
Lbendd  elev=480.5 ihead=506
Lslope2 elev=480.5 ihead=506

RHW     elev=519.5  head=553.3  idemand=0
Rinlet  elev=519.5 ihead=553.3
R1      elev=507   ihead=553.3
Rwellu  elev=495   ihead=553.3   demand=0
Rfillu  elev=495   ihead=553.3
Rfilld  elev=495   ihead=506
Rslot1  elev=495   ihead=506

Rmanu elev=494.24   ihead=506
Rp1u  elev=494.24   ihead=506
Rp2u  elev=494.24   ihead=506
Rp3u  elev=494.24   ihead=506
Rp4u  elev=494.24   ihead=506
Rp5u  elev=494.24   ihead=506
Rp1d  elev=494.24   ihead=506
Rp2d  elev=494.24   ihead=506
Rp3d  elev=494.24   ihead=506
Rp4d  elev=494.24   ihead=506
Rp5d  elev=494.24   ihead=506
Rp1b  elev=494.24   ihead=506
Rp2b  elev=494.24   ihead=506
Rp3b  elev=494.24   ihead=506
Rp4b  elev=494.24   ihead=506
Rp5b  elev=494.24   ihead=506
Rmand elev=494.24   ihead=506

R2    elev=495   ihead=506
R3    elev=495   ihead=506
Rempu elev=495   ihead=506
Rwelld elev=495   ihead=506  demand=0

Rempd   elev=495   ihead=506
Rslope1 elev=495   ihead=506
Rbend   elev=441.5 ihead=506

USpintle elev=490.5   ihead=506  demand=0
USman    elev=490.5   ihead=506
C1       elev=490.5   ihead=506
C2       elev=490.5   ihead=506
C3       elev=490.5   ihead=506
C4       elev=490.5   ihead=506
C5       elev=490.5   ihead=506
DSman    elev=490.5   ihead=506
Recess   elev=490.5   ihead=506
DSpintle elev=490.5   ihead=506  demand=0

transition elev=441.5  ihead=506
tunnel     elev=441.5  ihead=506
vbend1     elev=441.5  ihead=506
vbend2     elev=441.5  ihead=506
outlet     elev=496    ihead=506
river      elev=496     head=506 idemand=0

*FUNCTIONS ======================================================================

!Open_empty   discrete  xscale=90   ! emptying
Open_LS_Fill  discrete  xscale=269  ! filling
 interpolation=linear, xshift=0
 xy_pairs={  0,0      .03,.013  .06,.0274 .10,.0491 .15,.080  .20,.115
           .25,.1541 .35,.2434 .45,.3461 .55,.4599 .65,.5829 .75,.7138
           .85,.8525 .95,1       1, 1 }

Open_RS_Fill  discrete  xscale=265  ! filling
 interpolation=linear, xshift=0, share_xy=Open_LS_Fill

tainterCdv  discrete  interpolation=spline
  x_values={0,    0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35,
            0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75,
            0.8, 0.85, 0.9, 0.95,   1}
  y_values={0, 0.0443, 0.0836, 0.1195, 0.1534, 0.196, 0.243, 0.295,
            0.349, 0.415, 0.494, 0.587, 0.69, 0.8165, 0.976, 1.155,
            1.387,  1.69,  2.02, 2.582, 3.162}

tainterCc  polynomial coefficients={.948, -1.396, 2.98, -2.918, 1.385 }

kd31 discrete  interpolation=spline   ! Miller's Internal Flow
  x_values={ 0   .1  .2  .3  .4  .5   .6  .7 .8  .9 1.0}
  y_values={.50 .40 .30 .21 .13 .061 .019  0  0   0   0}

kd32a discrete interpolation=spline    ! From port model data
```

Figure 27 (continued): LOCKSIM input file for Wheeler Main Lock (file "wheeler.sim")

```
  x_values={
     0      .01     .1      .2      .3      .4      .5      .6
     .7     .8      .9      1       1.1     1.35    1.7     2
     2.4    2.6     3.3     5       6.8     10      13 }
  y_values={
    .85     .851    .848    .84     .827    .805    .783    .753
    .707    .656    .616    .595    .595    .620    .651    .686
    .785    .875    1.265   2.805   5.129   10.573  16.9 }

kd32b power coefficients={.10} powers={2}

kd32 composite xscale=.0054684 ! A2/A3, Q2/Q3=1 at x=182.9 (x=V2/V3)
  xtransitions={13} functions={kd32a kd32b}

kc31  discrete  interpolation=spline    ! Stockstill et al. paper
  x_values={ 0   .1  .2   .3   .4   .5   .6  .7   .8  .9  1.0}
  y_values={1.7 2.5 2.35 1.74 1.13 .65  .28  .18  .15 .13   0}

kc21a_wheeler discrete interpolation=spline  ! From port model data
  x_values={
     0    .001  .121  .217    .3    .4    .5     .6     .7    .79    .9
     1     1.2   1.4  1.65     2   2.5     3      4      5     10 }
  y_values={
    -.850 -.845 -.719 -.599 -.494 -.378 -.261  -.149  -.043   .048  .162
     .301  .587  .954 1.478 2.370 3.990 5.946 10.916 17.350     70 }

kc21b_wheeler power coefficients={.70} powers={2}

kc21 composite xscale=.0054684 ! A2/A3, Q2/Q3=1 at x=182.9 (x=V2/V1)
  xtransitions={10} functions={kc21a_wheeler kc21b_wheeler}

*CROSS_SECTIONS ========================================================

  chamber   TRAPEZOIDAL  MN_n=.02, bed_w=110.083

    !  representation of D.S. gate recessess -- width gives desired
    !  total lock surface area (not necessarily recess depth)
  rchamber  TRAPEZOIDAL  MN_n=.02, bed_w=118.94

*PLOT_VARIABLES ========================================================

!------ plot variables for filling -------------

 Lfillu head
 Lempu head
 Lfillu Lfilld valve_position
 Lfillu Lfilld cav_index
 Lfillu Lfilld vena_head
 Lfillu Lfilld discharge
 Rfillu Rfilld discharge

! Length: a->c=573.67, a->b=284.67, b->c=289

 USpintle USman depth sectno=2  ! (a)
 C3 depth                       ! (b)
 DSman  Recess  depth sectno=4  ! (c)

 Lp1b  C1  discharge
 Lp2b  C2  discharge
 Lp3b  C3  discharge
 Lp4b  C4  discharge
 Lp5b  C5  discharge

*END ===================================================================
```

Figure 27 (continued): LOCKSIM input file for Wheeler Main Lock (file "wheeler.sim")

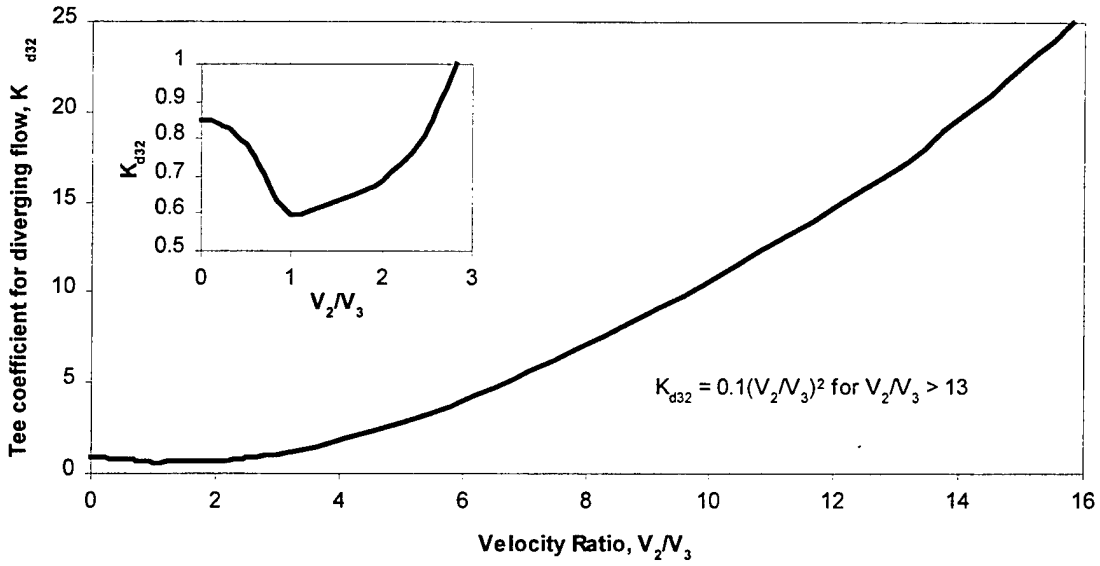


Figure 28: Tee coefficient K_{d32} for Wheeler multiport geometry

which averaged about 0.10 in the experiments. The tee coefficient K_{d32} (see definition sketch in Figure 88 in Part 2) for dividing flow from the culvert into the portholes (filling) is illustrated in Figure 28 and described by function "kd32" in the **FUNCTIONS* section of "wheeler.sim." The tee coefficient K_{c21} for combining flow from the portholes into the culvert (emptying) is illustrated in Figure 29 and described by function "kc21." For both of these functions, the *xscale* parameter is used to convert the independent variable from velocity ratio to the required discharge ratio.

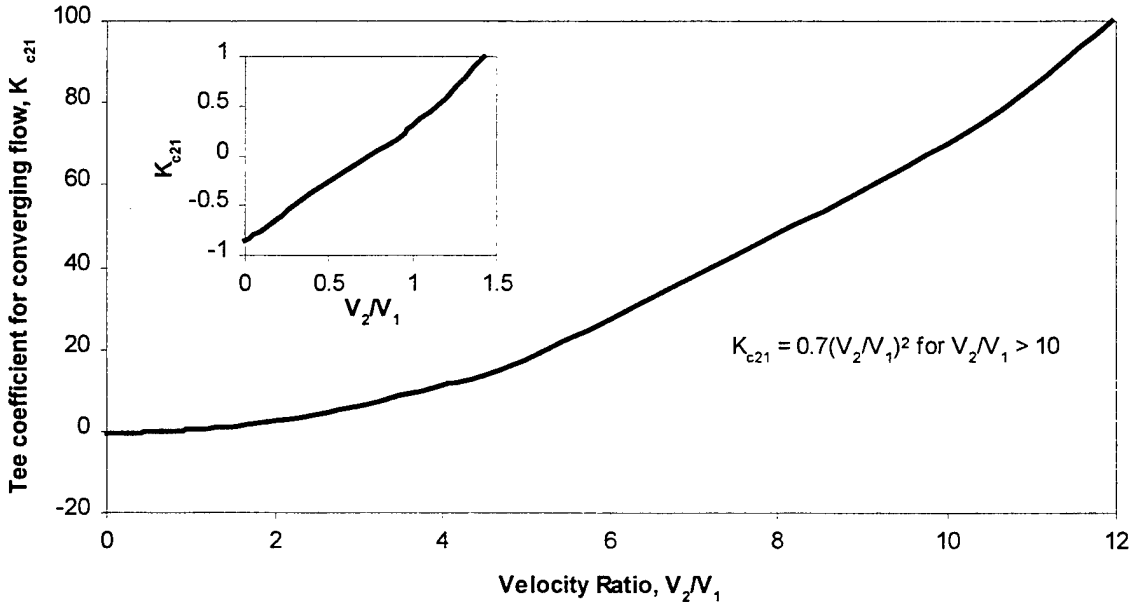


Figure 29: Tee coefficient K_{c21} for Wheeler multiport geometry

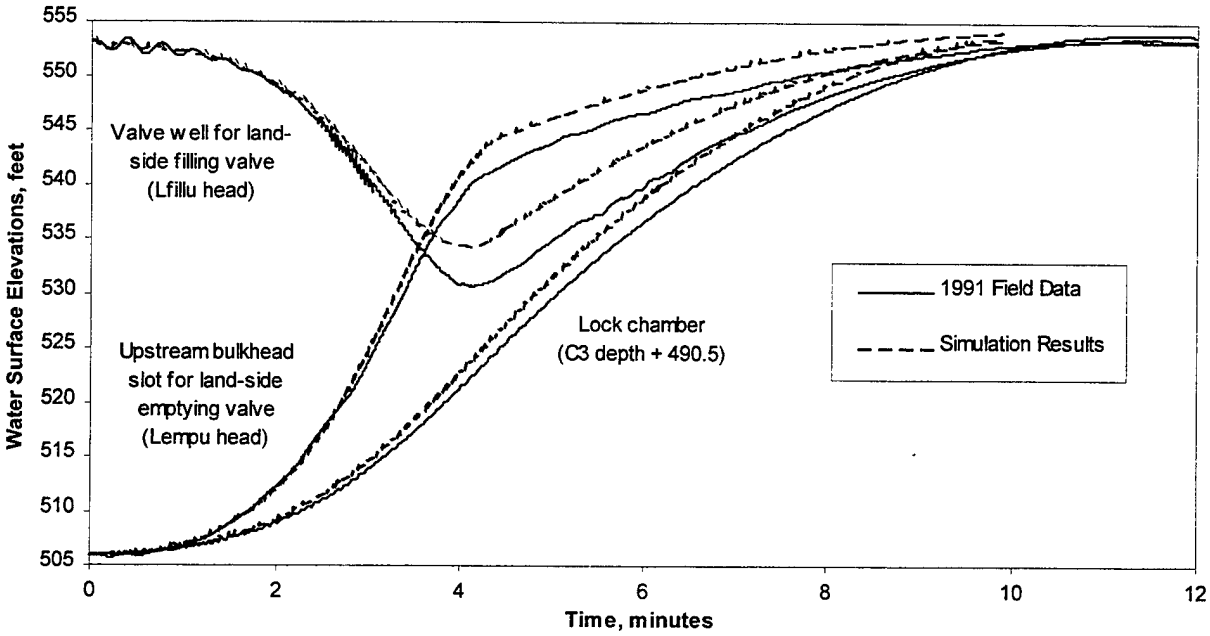


Figure 30: Simulation results before calibration compared with field data for filling at Wheeler Main Lock

The remaining tee coefficients were not measured in the laboratory experiments. The functions "kd31" and "kc31," which describe the tee coefficients K_{d31} and K_{c31} for culvert flow past the portholes, are based on data available in the literature (Miller, 1990; Stockstill et al., 1991). Symmetry determines that $K_{d13} = K_{d31}$, $K_{d12} = K_{d32}$, $K_{c13} = K_{c31}$, $K_{c23} = K_{c21}$, $K_{d23} = K_{d21}$, and $K_{c12} = K_{c32}$. Equations 49 and 50 in Part 2 determine that $K_{d21}(0) \approx K_{d21}(1) = 0.70$ and that $K_{c32}(0) \approx K_{c32}(1) = 0.10$. Both of these tee coefficients are assumed to be constants for discharge ratios between zero and 1.

Figure 30 compares water surface elevations measured during field tests of filling to simulation results obtained before the inlets were "calibrated" by increasing their loss coefficients from 0.2, a typically reasonable value, to 0.7. Figure 31 shows the same comparison after the inlets were "calibrated." The agreement of simulation results with measured results is significantly better in Figure 31 than in Figure 30 for all three of the water levels compared. The inlet loss coefficient was adjusted rather than some other coefficient because lowering both the computed valve well water surface elevation and the total inflow discharge (to fill the chamber slower) required an increase in resistance upstream from the filling valves. Further improvement in the agreement between computed results and field test results would require adjustments in the relationship describing discharge coefficient as a function of opening position for the filling valves. Figure 32 compares water surface elevations measured during field tests of emptying to simulation results obtained without calibration. The agreement between computed and measured results is excellent for both water surface elevations compared.

Figure 33 and Figure 34 summarize the results of filling and emptying simulations for Wheeler Main Lock.

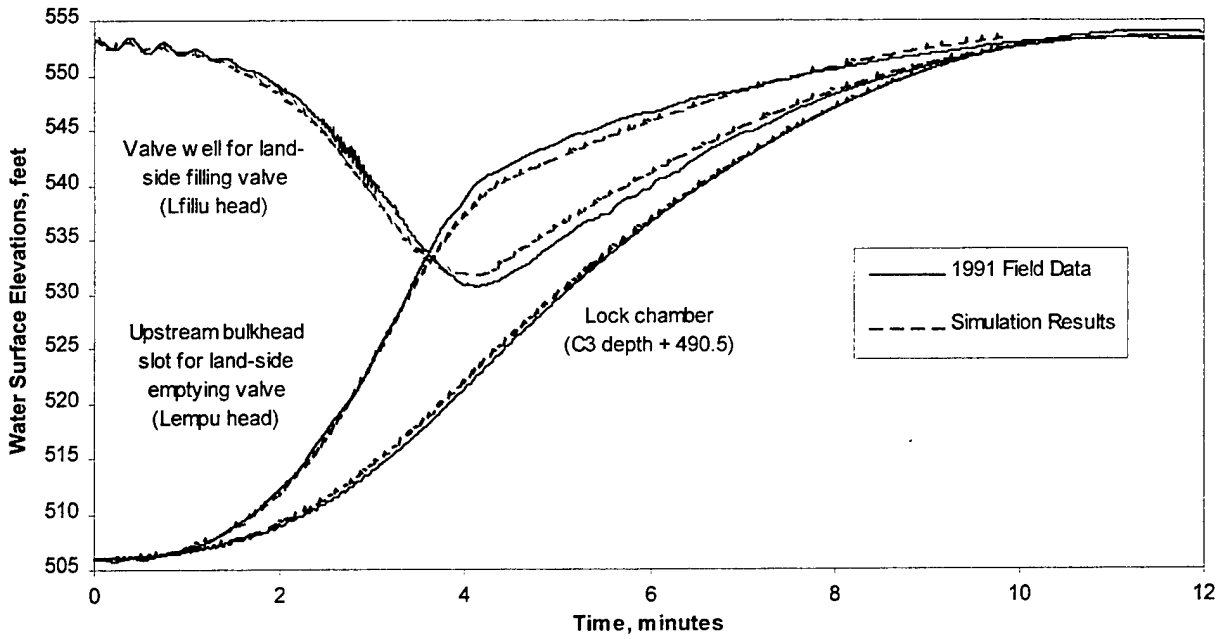


Figure 31: Simulation results after calibration compared with field data for filling at Wheeler Main Lock

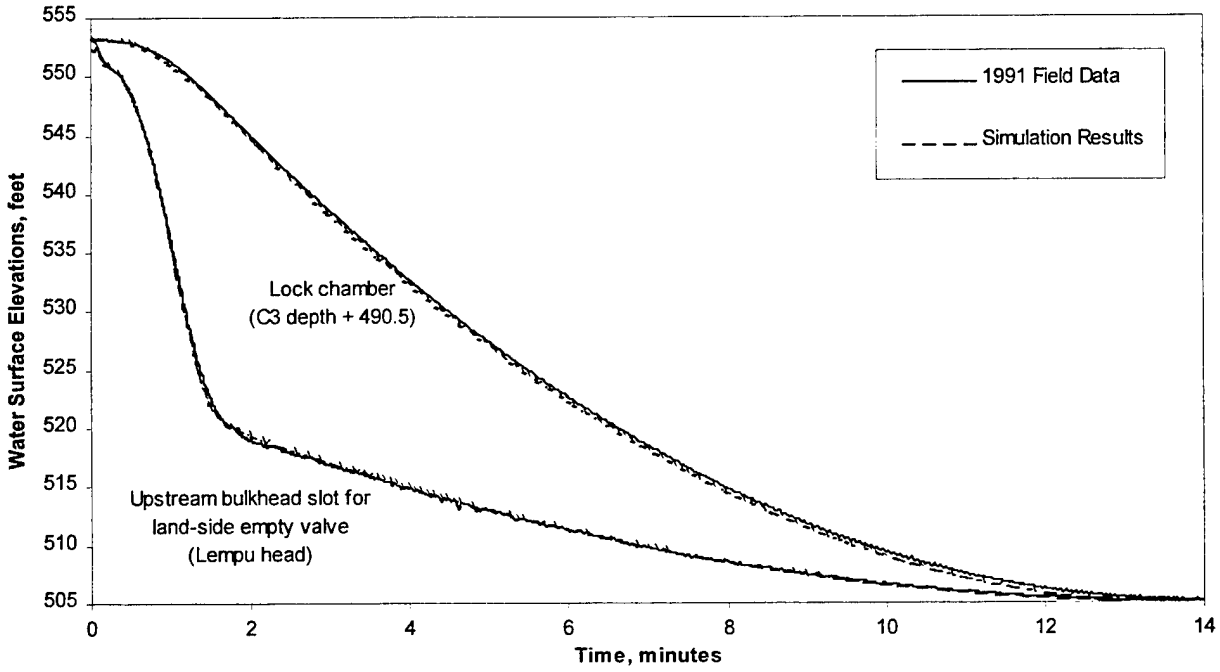


Figure 32: Simulation results compared with field data for emptying at Wheeler Main Lock

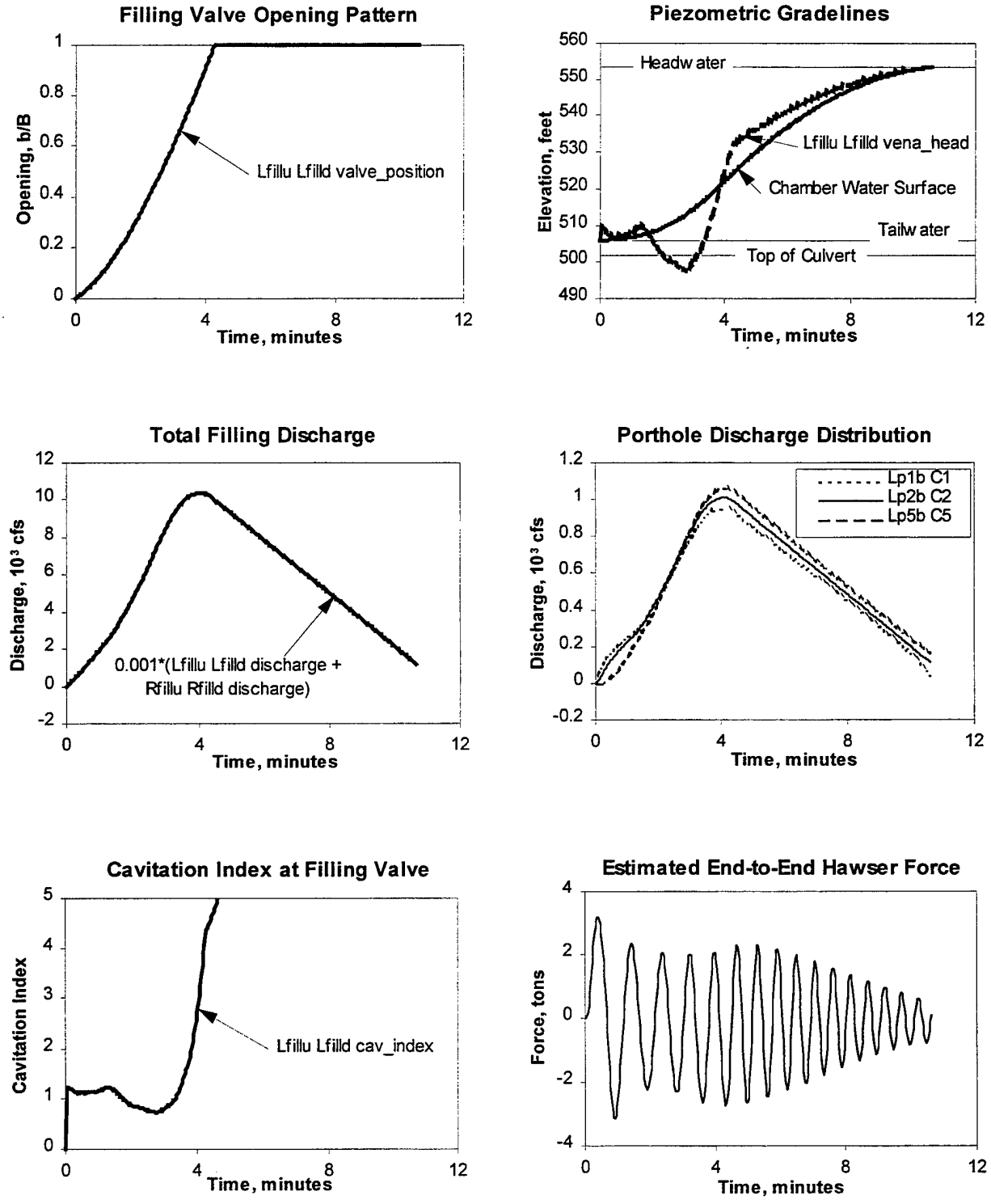


Figure 33: Filling results for Wheeler Main Lock (calibrated inlets)

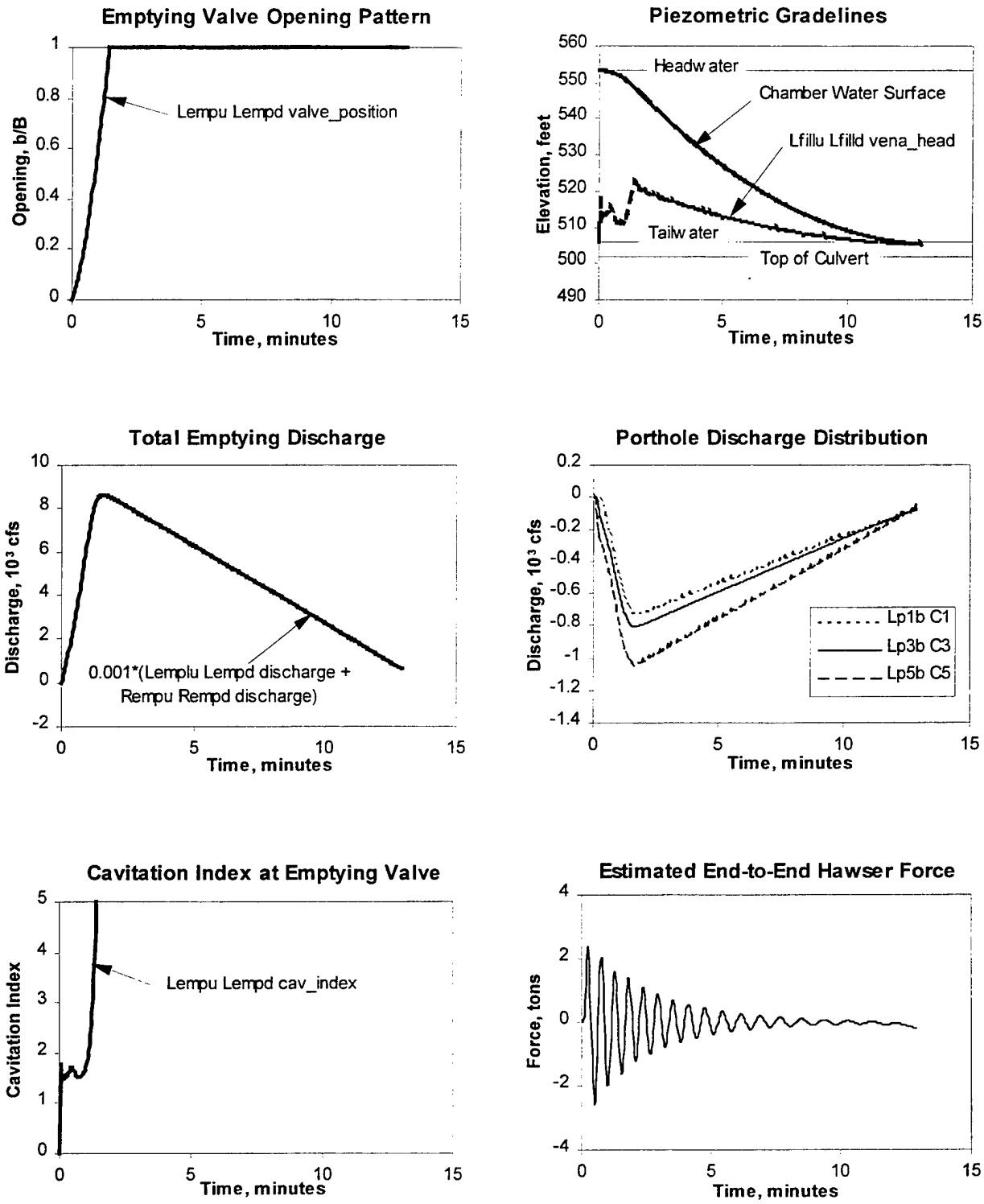


Figure 34: Emptying results for Wheeler Main Lock

Bottom-Longitudinal Lock

Bay Springs Lock is used for the bottom-longitudinal example. As for Wheeler Main Lock, field data are available for comparison to simulation results (McGee, 1989). Data are also available from a 1:25 scale physical model (Ables, 1978).

Tee coefficients for dividing, or filling, flow obtained in laboratory tests by the U. S. Army Corps of Engineers Waterways Experiment Station on portholes that are similar to the Bay Springs portholes, are presented in this section and used for the example. Because the tested portholes are more rounded in shape than the Bay Springs portholes, the tee coefficients under-predict energy losses for Bay Springs. For the example, energy losses were increased by adding *pipe_loss* components to the branch end of each *diverging_manifold* component. The loss coefficient assigned to each *pipe_loss* component was determined by iterative adjustment until reasonable agreement was obtained between the computed and measured lock water surface elevations. A converged solution for an entire filling cycle could not be obtained for values of the loss coefficient smaller than the value determined by calibration. This convergence difficulty is not uncommon for LOCKSIM when energy losses through tee and manifold components are too small to significantly control the discharge through them (over-sized or under-restrictive tees and manifolds).

This example also demonstrates the effects on predicted hawser forces of LOCKSIM's neglect of the upstream- or downstream-directed component of momentum associated with flow entering the lock chamber from the portholes (see the "Lock Chamber" section of this guide). This is, at least, a probable explanation for the disagreement between the predicted end-to-end hawser forces (less than 0.1 tons) and those measured in the 1:25 scale model (3 to 4 tons).

Figure 35 approximately depicts the bottom-longitudinal filling and emptying system for Bay Springs Lock. The clear surface area of the lock chamber is 110 feet wide by 600 feet long, large enough to accommodate nine jumbo barges. The maximum lift of the lock is approximately 84 feet. Water for filling the lock enters longitudinal wall culverts through intake manifolds in the lock walls upstream from the upper miter gates. Both filling and emptying flows are controlled by reverse tainter valves in the wall culverts. Bulkhead slots are provided both upstream and downstream from the valves for de-watering purposes. Flow passes through "crossover" and "tuning fork" culverts before discharging into the lock through 96 1.5-feet by 3.5 feet portholes distributed along the lengths of upstream and downstream longitudinal floor culverts. Water empties the lock through the downstream wall culverts which discharge into the tailwater through outlet laterals, one positioned between the lock approach walls and the other positioned outside the approach walls.

Figure 36 illustrates the schematic network of components and nodes that represents Bay Springs Lock for filling simulations using LOCKSIM. For emptying simulations, the *diverging_manifold* components were not included in the schematic because tee coefficients for converging, or emptying flow, were not available from the laboratory data. The numbers in parentheses near the node labels in Figure 36 are centerline elevations (for closed conduit

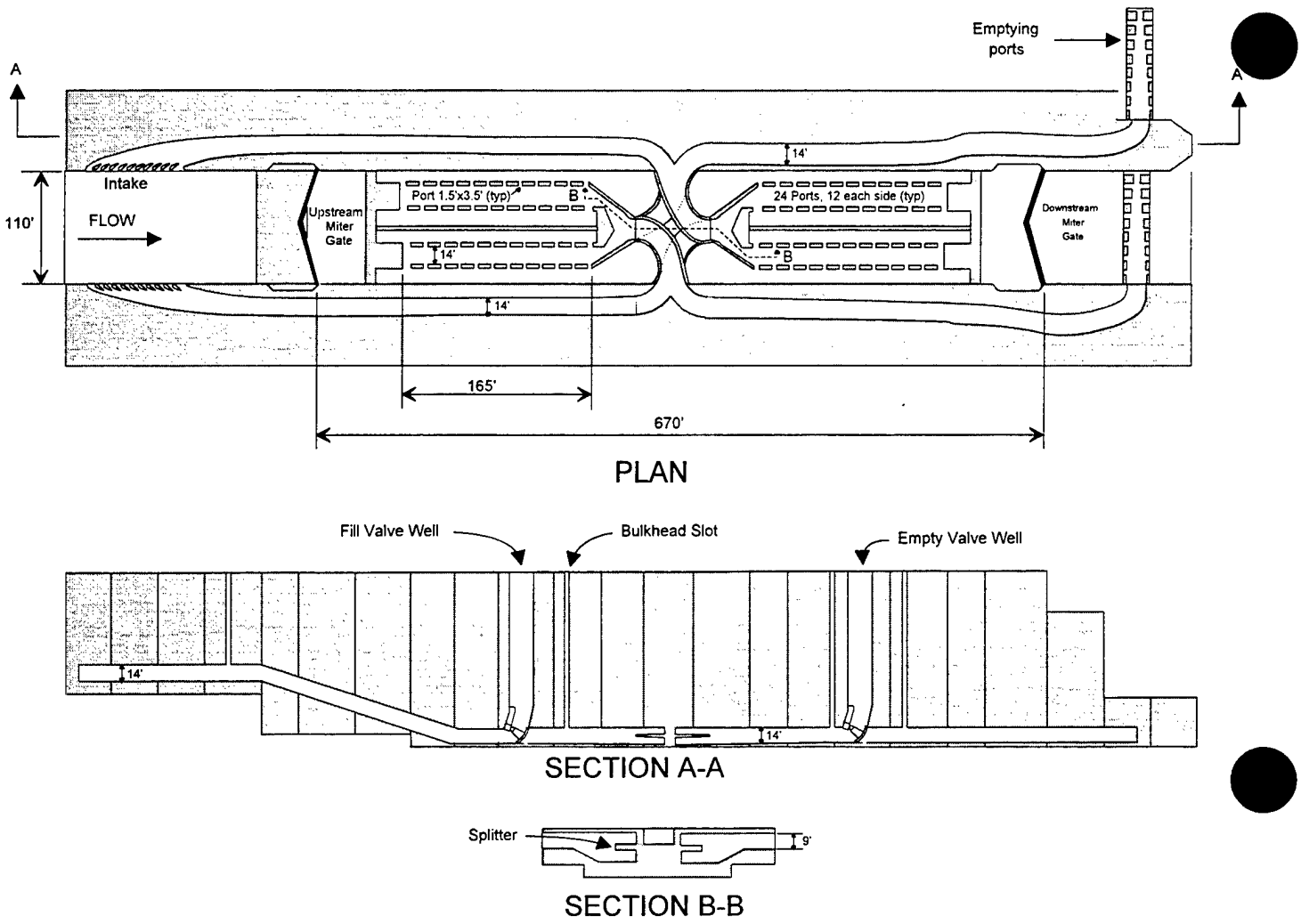


Figure 35: Conceptual illustration of bottom longitudinal filling and emptying system for Bay Springs Lock

components) or bottom elevations (for free-surface components) in feet. The numbers above the components indicate cross-sectional dimensions (width times height). The filling and emptying port manifolds are represented between nodes "U1" and "U2" upstream and "D1" and "D2" downstream. Four *diverging_manifold* components connected to *pipe_loss* components represent the 48 porthholes in the upstream half of the chamber. Another four *diverging_manifold* components connected to *pipe_loss* components represent the 48 porthholes in the downstream half of the chamber. Each *diverging_manifold* component is located at the center of the 12 portholes that it represents (6 from the left branch and 6 from the right branch of the upstream or downstream "tuning fork"). The lock chamber is represented by a series of *open_channel* components between nodes "USpintle" and "DSpintle." The filling valves are included between nodes "Lfillu" and "Lfilld" and between nodes "Rfillu" and "Rfilld." The emptying valves are included between nodes "Lempu" and "Lempd" and between nodes "Rempu" and "Rempd." Water *storage* components, each with cross-sectional area equal to 414 ft^2, are connected to the upstream nodes of each valve to represent the valve wells.

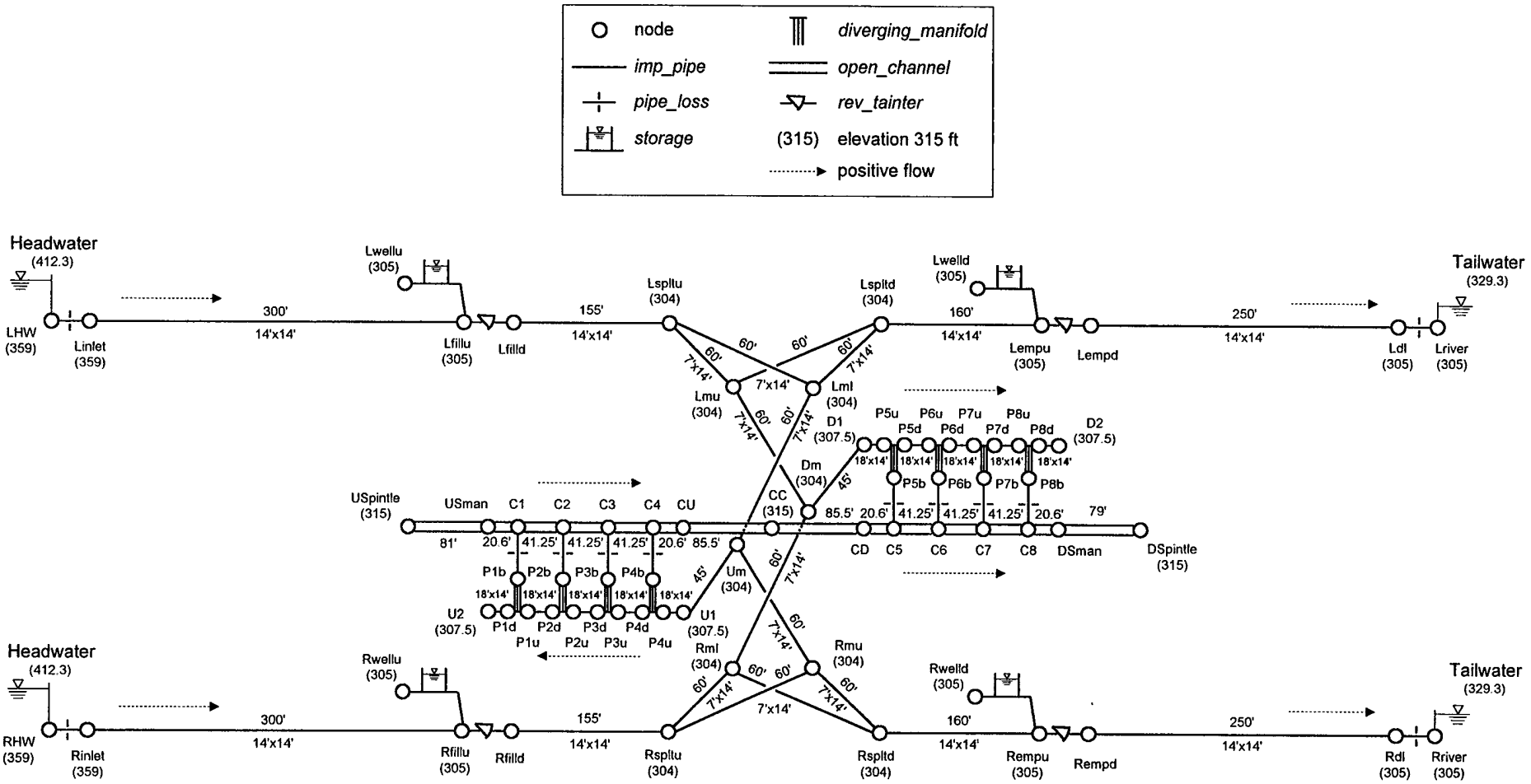


Figure 36: Schematic representation of Bay Springs Lock

Figure 37 illustrates the input file "bayspr_f.sim," where "_f" is used to distinguish this input file from a separate input file for emptying simulations, which is named "bayspr_e.sim." In Figure 37, an initial condition for filling, either initial head (*ihead*) or initial demand (*idemand*), is specified for every node. A boundary condition, either *head* or *demand* is specified for every terminal node. Functions are specified for opening the filling valves and for their discharge and contraction coefficients. The remaining functions specify the tee coefficients for the *diverging_manifold* components. The *pipe_loss* components "LHW Linlet" and "RHW Rinlet" each represent the combined effect of an inlet and two 22-degree bends. The *pipe_loss* components "Ldl Lriver" and "Rdl Rriver" each represents the combined effect of a 90-degree bend and an exit into the tailwater. Each *diverging_manifold* component has 3 branches, each with cross-sectional area equal to 21 ft^2, which is the combined area of 4 portholes. Each *pipe_loss* component connecting a *diverging_manifold* component to the lock chamber has a cross-sectional area of 63 ft^2, which is the total area of 3 branches, and a loss coefficient of 0.35 for positive, filling flow. As already mentioned, this loss coefficient was determined by calibration with field data.

Figure 38 illustrates the tee coefficient K_{d32} for dividing flow from the culvert (point 3) into the chamber (point 2) that was measured in laboratory experiments on the portholes illustrated above the graph. The function "kd32" defined in the **FUNCTIONS* section of "bayspr_f.sim" reproduces the curve fit shown in Figure 38. However, for the Bay Springs portholes, which are illustrated in Figure 39, the function "kd32" is assumed to represent the energy losses between the culvert (point 3 in Figure 39) and the portholes (point 2 in Figure 39). Additional losses, determined by calibration, are simulated between the portholes (point 2) and the chamber (point 4) using *pipe_loss* components. For function "kd32," the *xscale* parameter is used to convert the independent variable from velocity ratio to the required discharge ratio. The remaining tee coefficients were not measured in the laboratory experiments. The functions "kd31" and "kc31," which describe the tee coefficients K_{d31} and K_{c31} for culvert flow past the portholes, are based on data available in the literature (Miller, 1990 and Stockstill et al., 1991). The tee coefficient K_{d21}, which is not important for filling flows, is assumed to be constant and equal to 1.0. Symmetry determines that $K_{d13} = K_{d31}$, $K_{d12} = K_{d32}$, $K_{c13} = K_{c31}$, $K_{c23} = K_{c21}$, $K_{d23} = K_{d21}$, and $K_{c12} = K_{c32}$. Equations 49 and 50 in Part 2 determine that $K_{d21}(0) \approx K_{d21}(1) = 1.0$ and that $K_{c32}(0) \approx K_{c32}(1) = 0.333$. Like K_{d21},the tee coefficient K_{c32} is assumed to be constant for discharge ratios between zero and 1. Equations 49 and 50 also determine that $K_{c21}(0) = -0.456$ and $K_{c21}(1) \approx 144$. For simplicity and because it is not important for filling flows, K_{c21} is assumed to vary linearly between -0.456 and 144 for discharge ratios between 0 and 1.

For reference, Figure 40 illustrates the tee coefficient K_{d32} measured after "port extensions" were added to the portholes illustrated in Figure 38. The curve fit for the data in Figure 40 differs only slightly from the curve fit for the data in Figure 38. The later is defined by the function "kd32" in Figure 37.

```
!Bay Springs Main Lock Filling and Emptying System          8/21/98

!  Field Test Conditions
!           Filling, 3.65-minute valve,  FE1 -- HW=412.3, TW = 329.3
!           Comparison with prototype data
!           Ports represented by manifold components

*CONSTANTS ============================================================
  time_step=2, wf_time=.55, gravity=32.146
  dQ_max=.001, dH_max=.0001, dX_max=.0001
  plot_field=11, plot_line=200, plot_file=bayspr_f.plt, plot_labels=row

*COMPONENTS ===========================================================
!----------- Inlet and fill valves -----------------------------------
      ! inlet and 2 22-degree bends
LHW     Linlet  pipe_loss  us_area=1e5, area=196, K+=0.25
Linlet  Lfillu  imp_pipe dia=14 area=196 len=300 rough=.003 wave=3500
Lfillu  Lwellu  storage surface_area=414  max_wsel=424
Lfillu  Lfilld  rev_tainter   b=14, w=14, el_bottom=298
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    b/B_vs_t=Open_bstainter  ! filling

      ! inlet and 2 22-degree bends
RHW     Rinlet  pipe_loss  us_area=1e5, area=196, K+=0.25
Rinlet  Rfillu  imp_pipe dia=14 area=196 len=300 rough=.003 wave=3500
Rfillu  Rwellu  storage surface_area=414  max_wsel=424
Rfillu  Rfilld  rev_tainter   b=14, w=14, el_bottom=298
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    b/B_vs_t=Open_bstainter  ! filling

!------------ Culverts Leading to Ports -------------------------------
Lfilld Lspltu imp_pipe dia=14.00 area=196 len=155 rough=.003 wave=3500
Lspltu Lmu    imp_pipe dia=9.333 area=98  len=60  rough=.003 wave=3500
Lspltu Lml    imp_pipe dia=9.333 area=98  len=60  rough=.003 wave=3500
Rfilld Rspltu imp_pipe dia=14.00 area=196 len=155 rough=.003 wave=3500
Rspltu Rmu    imp_pipe dia=9.333 area=98  len=60  rough=.003 wave=3500
Rspltu Rml    imp_pipe dia=9.333 area=98  len=60  rough=.003 wave=3500

Lmu     Dm    imp_pipe dia=9.333 area=98 len=60 rough=.003 wave=3500
Lml     Um    imp_pipe dia=9.333 area=98 len=60 rough=.003 wave=3500
Rmu     Um    imp_pipe dia=9.333 area=98 len=60 rough=.003 wave=3500
Rml     Dm    imp_pipe dia=9.333 area=98 len=60 rough=.003 wave=3500

Um      U1    imp_pipe dia=10.96 area=252 len=45 rough=.003 wave=3500
Dm      D1    imp_pipe dia=10.96 area=252 len=45 rough=.003 wave=3500

!------------ Port Manifolds ------------------------------------------
U1  P4u imp_pipe dia=10.96 area=252 len=20.625 rough=.003 wave=3500
P4d P3u imp_pipe dia=10.96 area=252 len=41.25  rough=.003 wave=3500
P3d P2u imp_pipe dia=10.96 area=252 len=41.25  rough=.003 wave=3500
P2d P1u imp_pipe dia=10.96 area=252 len=41.25  rough=.003 wave=3500
P1d U2  imp_pipe dia=10.96 area=252 len=20.625 rough=.003 wave=3500

D1  P5u imp_pipe dia=10.96 area=252 len=20.625 rough=.003 wave=3500
P5d P6u imp_pipe dia=10.96 area=252 len=41.25  rough=.003 wave=3500
P6d P7u imp_pipe dia=10.96 area=252 len=41.25  rough=.003 wave=3500
P7d P8u imp_pipe dia=10.96 area=252 len=41.25  rough=.003 wave=3500
P8d D2  imp_pipe dia=10.96 area=252 len=20.625 rough=.003 wave=3500

          ! each manifold represents 12 portholes (3 rows of four)
P1u P1d P1b diverging_manifold area1=252, area2=21, branches=3
  Kd31_vs_Qr=Kd13_vs_Qr=kd31 Kd32_vs_Qr=Kd12_vs_Qr=kd32 iq1=0 iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31 Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=1
  fixed_Kc32=fixed_Kc12=.333
P1b C1 pipe_loss area=63 ds_area=1e5 K+=.35

P2u P2d P2b diverging_manifold area1=252, area2=21, branches=3
  Kd31_vs_Qr=Kd13_vs_Qr=kd31 Kd32_vs_Qr=Kd12_vs_Qr=kd32 iq1=0 iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31 Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=1
  fixed_Kc32=fixed_Kc12=.333
P2b C2 pipe_loss area=63 ds_area=1e5 K+=.35

P3u P3d P3b diverging_manifold area1=252, area2=21, branches=3
  Kd31_vs_Qr=Kd13_vs_Qr=kd31 Kd32_vs_Qr=Kd12_vs_Qr=kd32 iq1=0 iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31 Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=1
  fixed_Kc32=fixed_Kc12=.333
P3b C3 pipe_loss area=63 ds_area=1e5 K+=.35

P4u P4d P4b diverging_manifold area1=252, area2=21, branches=3
  Kd31_vs_Qr=Kd13_vs_Qr=kd31 Kd32_vs_Qr=Kd12_vs_Qr=kd32 iq1=0 iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31 Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=1
  fixed_Kc32=fixed_Kc12=.333
P4b C4 pipe_loss area=63 ds_area=1e5 K+=.35

P5u P5d P5b diverging_manifold area1=252, area2=21, branches=3
  Kd31_vs_Qr=Kd13_vs_Qr=kd31 Kd32_vs_Qr=Kd12_vs_Qr=kd32 iq1=0 iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31 Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=1
  fixed_Kc32=fixed_Kc12=.333
P5b C5 pipe_loss area=63 ds_area=1e5 K+=.35

P6u P6d P6b diverging_manifold area1=252, area2=21, branches=3
  Kd31_vs_Qr=Kd13_vs_Qr=kd31 Kd32_vs_Qr=Kd12_vs_Qr=kd32 iq1=0 iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31 Kc21_vs_Qr=Kc23_vs_Qr=kc21
```

Figure 37: LOCKSIM input file for Bay Springs Lock (file "bayspr_f.sim")

```
  fixed_Kd21=fixed_Kd23=1
  fixed_Kc32=fixed_Kc12=.333
P6b C6 pipe_loss area=63 ds_area=1e5 K+=.35

P7u P7d P7b diverging_manifold area1=252, area2=21, branches=3
  Kd31_vs_Qr=Kd13_vs_Qr=kd31 Kd32_vs_Qr=Kd12_vs_Qr=kd32 iq1=0 iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31 Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=1
  fixed_Kc32=fixed_Kc12=.333
P7b C7 pipe_loss area=63 ds_area=1e5 K+=.35

P8u P8d P8b diverging_manifold area1=252, area2=21, branches=3
  Kd31_vs_Qr=Kd13_vs_Qr=kd31 Kd32_vs_Qr=Kd12_vs_Qr=kd32 iq1=0 iq2=0
  Kc31_vs_Qr=Kc13_vs_Qr=kc31 Kc21_vs_Qr=Kc23_vs_Qr=kc21
  fixed_Kd21=fixed_Kd23=1
  fixed_Kc32=fixed_Kc12=.333
P8b C8 pipe_loss area=63 ds_area=1e5 K+=.35

!------------ Empty culverts and valves ------------------------------
Lmu    Lspltd imp_pipe dia=9.333 area=98  len=60 rough=.003 wave=3500
Lml    Lspltd imp_pipe dia=9.333 area=98  len=60 rough=.003 wave=3500
Lspltd Lempu  imp_pipe dia=14   area=196 len=160 rough=.003 wave=3500
Lempu  Lwelld  storage surface_area=414  max_wsel=424
Lempu  Lempd  rev_tainter   b=14, w=14, el_bottom=298
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    fixed_b/B=0                ! filling
Lempd  Ld1    imp_pipe dia=14   area=196 len=250 rough=.003 wave=3500
      ! exit and 90-degree bend
Ld1    Lriver pipe_loss ds_area=1e5, area=196, K+=1.16

Rmu    Rspltd imp_pipe dia=9.333 area=98  len=60 rough=.003 wave=3500
Rml    Rspltd imp_pipe dia=9.333 area=98  len=60 rough=.003 wave=3500
Rspltd Rempu  imp_pipe dia=14   area=196 len=160 rough=.003 wave=3500
Rempu  Rwelld  storage surface_area=414  max_wsel=424
Rempu  Rempd  rev_tainter   b=14, w=14, el_bottom=298
    Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc
    fixed_b/B=0                ! filling
Rempd  Rd1    imp_pipe dia=14   area=196 len=250 rough=.003 wave=3500
      ! exit and 90-degree bend
Rd1    Rriver pipe_loss  ds_area=1e5, area=196, K+=1.16

!------------ Lock chamber modeled as series of open channels ---------
USpintle USman open_channel cross_section=chamber len=81     reaches=2
USman    C1    open_channel cross_section=chamber len=20.625 reaches=1
C1       C2    open_channel cross_section=chamber len=41.25  reaches=1
C2       C3    open_channel cross_section=chamber len=41.25  reaches=1
C3       C4    open_channel cross_section=chamber len=41.25  reaches=1
C4       CU    open_channel cross_section=chamber len=20.625 reaches=1
CU       CC    open_channel cross_section=chamber len=85.5   reaches=2
CC       CD    open_channel cross_section=chamber len=85.5   reaches=2
CD       C5    open_channel cross_section=chamber len=20.625 reaches=1
C5       C6    open_channel cross_section=chamber len=41.25  reaches=1
C6       C7    open_channel cross_section=chamber len=41.25  reaches=1
C7       C8    open_channel cross_section=chamber len=41.25  reaches=1
C8       DSman open_channel cross_section=chamber len=20.625 reaches=1
DSman DSpintle open_channel cross_section=chamber len=79     reaches=2

*NODES ==========================================================

!------ initial conditions for filling ------------
LHW       elev=359    head=412.3 idemand=0
Linlet    elev=359   ihead=412.3
Lfillu    elev=305   ihead=412.3
Lwellu    elev=305   ihead=412.3  demand=0
Lfilld    elev=305   ihead=329.3

RHW       elev=359    head=412.3 idemand=0
Rinlet    elev=359   ihead=412.3
Rfillu    elev=305   ihead=412.3
Rwellu    elev=305   ihead=412.3  demand=0
Rfilld    elev=305   ihead=329.3

Lspltu    elev=304   ihead=329.3
Lmu       elev=304   ihead=329.3
Lml       elev=304   ihead=329.3

Lspltd    elev=304   ihead=329.3
Lempu     elev=305   ihead=329.3
Lwelld    elev=305   ihead=329.3  demand=0
Lempd     elev=305   ihead=329.3
Ld1       elev=305   ihead=329.3
Lriver    elev=305    head=329.3

Rspltu    elev=304   ihead=329.3
Rmu       elev=304   ihead=329.3
Rml       elev=304   ihead=329.3

Rspltd    elev=304   ihead=329.3
Rempu     elev=305   ihead=329.3
Rwelld    elev=305   ihead=329.3  demand=0
Rempd     elev=305   ihead=329.3
Rd1       elev=305   ihead=329.3
Rriver    elev=305    head=329.3

Um     elev=304     ihead=329.3
U2     elev=307.5   ihead=329.3  demand=0
P1u    elev=307.5   ihead=329.3
P2u    elev=307.5   ihead=329.3
```

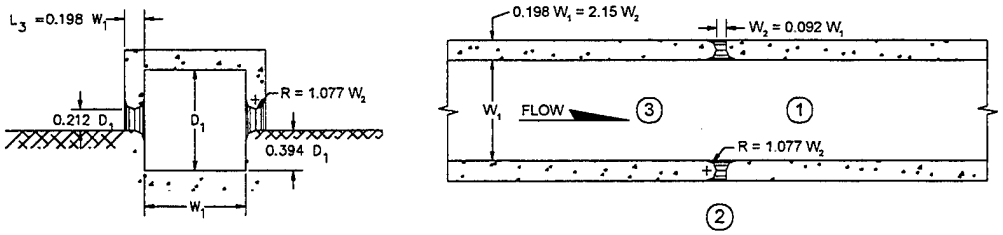
Figure 37 (continued): LOCKSIM input file for Bay Springs Lock (file "bayspr_f.sim")

```
P3u     elev=307.5   ihead=329.3
P4u     elev=307.5   ihead=329.3
P1d     elev=307.5   ihead=329.3
P2d     elev=307.5   ihead=329.3
P3d     elev=307.5   ihead=329.3
P4d     elev=307.5   ihead=329.3
P1b     elev=307.5   ihead=329.3
P2b     elev=307.5   ihead=329.3
P3b     elev=307.5   ihead=329.3
P4b     elev=307.5   ihead=329.3
U1      elev=307.5   ihead=329.3

Dm      elev=304     ihead=329.3
D1      elev=307.5   ihead=329.3
P5u     elev=307.5   ihead=329.3
P6u     elev=307.5   ihead=329.3
P7u     elev=307.5   ihead=329.3
P8u     elev=307.5   ihead=329.3
P5d     elev=307.5   ihead=329.3
P6d     elev=307.5   ihead=329.3
P7d     elev=307.5   ihead=329.3
P8d     elev=307.5   ihead=329.3
P5b     elev=307.5   ihead=329.3
P6b     elev=307.5   ihead=329.3
P7b     elev=307.5   ihead=329.3
P8b     elev=307.5   ihead=329.3
D2      elev=307.5   ihead=329.3  demand=0

USpintle elev=315    ihead=329.3  demand=0
USman    elev=315    ihead=329.3
C1       elev=315    ihead=329.3
C2       elev=315    ihead=329.3
C3       elev=315    ihead=329.3
C4       elev=315    ihead=329.3
CU       elev=315    ihead=329.3
CC       elev=315    ihead=329.3
CD       elev=315    ihead=329.3
C5       elev=315    ihead=329.3
C6       elev=315    ihead=329.3
C7       elev=315    ihead=329.3
C8       elev=315    ihead=329.3
DSman    elev=315    ihead=329.3
DSpintle elev=315    ihead=329.3  demand=0

*FUNCTIONS ============================================================
Open_bstainter  discrete xscale=219   ! filling
  interpolation=linear, xshift=0
  x_values={
     0     .025   .05    .075   .10   .15    .20    .25    .30    .35
     .40   .45    .50    .60    .70   .80    .90     1 }
  y_values={
     0    .006   .0194  .0299  .0408  .0644 .090   .130   .170   .210
   .255   .310   .360   .475   .595   .725  .855     1 }

tainterCdv  discrete  interpolation=spline
  x_values={0,   0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35,
            0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75,
            0.8, 0.85, 0.9, 0.95,   1}
  y_values={0, 0.0443, 0.0836, 0.1195, 0.1534, 0.196, 0.243, 0.295,
            0.349, 0.415, 0.494, 0.587, 0.69, 0.8165, 0.976, 1.155,
            1.387,  1.69,  2.02, 2.582, 3.162}

tainterCc  polynomial coefficients={.948, -1.396, 2.98, -2.918, 1.385 }

kd31 discrete  interpolation=spline   ! Miller's Internal Flow
  x_values={ 0   .1  .2  .3  .4  .5   .6  .7 .8  .9 1.0}
  y_values={.50 .40 .30 .21 .13 .061 .019  0  0   0   0}

kd32a discrete  ! Assumed constant value for V2/V3<1.5
  x_values={  0   1.5}
  y_values={.456 .456}

kd32b discrete interpolation=spline    ! From curve fit of WES data
 x_values={ 1.5  2.25     3     4     5      6      7      8
            9     10     11     12}
 y_values={.456 1.213 2.402 4.728 7.985 12.222 17.357 23.117
      29.078 34.879 40.684 47.952}

kd32c power coefficients={.333} powers={2}

! Fit through WES data for port shape "A" (Stockstill)
kd32 composite xscale=.083333 ! A2/A3, Q2/Q3=1 at x=12 (x=V2/V3)
  xtransitions={1.5 12} functions={kd32a kd32b kd32c}

kc31  discrete  interpolation=spline    ! Stockstill et al. paper
  x_values={ 0   .1  .2   .3   .4   .5   .6  .7   .8  .9  1.0}
  y_values={1.7 2.5 2.35 1.74 1.13 .65  .28  .18  .15 .13   0}

kc21 discrete  ! Made up; based on assumption that Kd21=1.0
  x_values={   0    1 }
  y_values={-.456  144}

*CROSS_SECTIONS ========================================================

  chamber  TRAPEZOIDAL  MN_n=.02, bed_w=110

*PLOT VARIABLES ========================================================
```

Figure 37 (continued): LOCKSIM input file for Bay Springs Lock (file "bayspr_f.sim")

```
!------ plot variables for filling -------------

 Lfillu head
 Lempu  head
 Lfillu Lfilld valve_position
 Lfillu Lfilld cav_index
 Lfillu Lfilld vena_head
 Lfillu Lfilld discharge
 Rfillu Rfilld discharge

! Length: a->c=581, a->b=291, b->c=290

 USpintle USman depth sectno=1  ! (a)
 CC depth                       ! (b)
 DSman DSpintle depth sectno=1  ! (c)

 P1b  C1  discharge
 P2b  C2  discharge
 P3b  C3  discharge
 P4b  C4  discharge

*END ==================================================================
```

Figure 37 (continued): LOCKSIM input file for Bay Springs Lock (file "bayspr_f.sim")



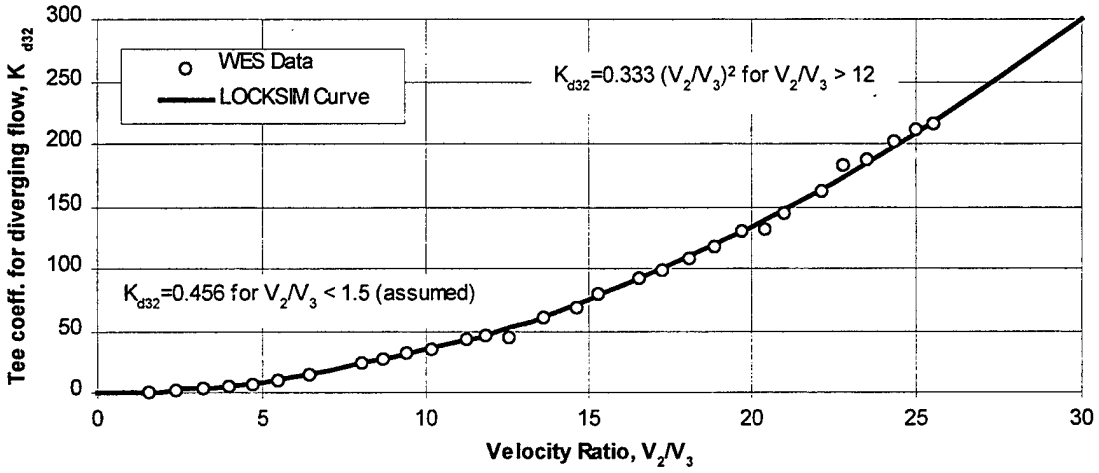


Figure 38: Tee coefficient K_{d32} for portholes illustrated above

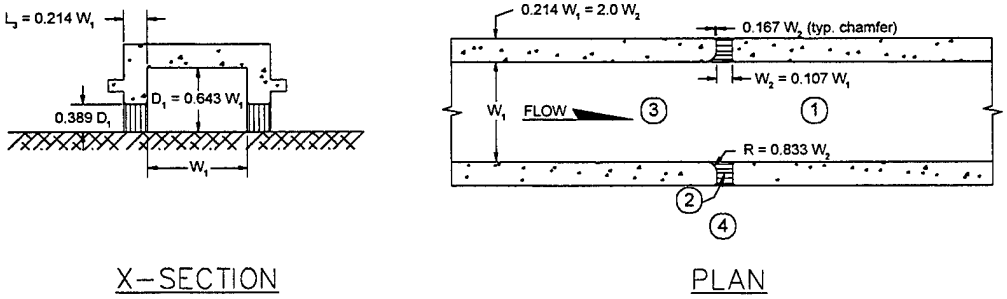


Figure 39: Geometry of Bay Springs Lock portholes

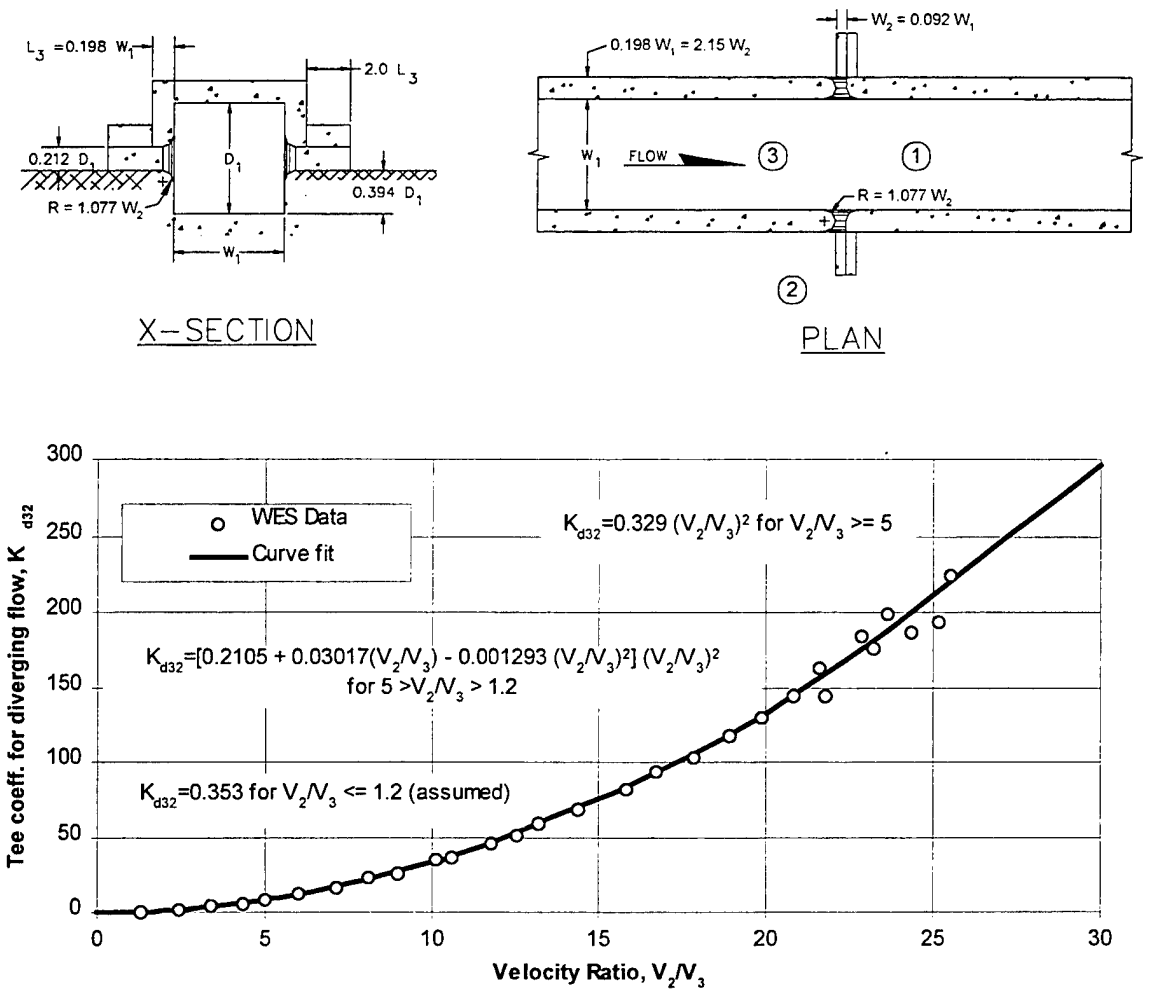


Figure 40: Tee coefficient K_{d32} for portholes with extensions illustrated above

Figure 41 compares piezometric head elevations measured during field tests of filling to simulation results obtained using a loss coefficient of 0.35 for the *pipe_loss* components connecting the *diverging_manifold* components to the lock chamber. Figure 42 compares piezometric head elevations measured during field tests of emptying to simulation results obtained by removing the *diverging_manifold* components from the schematic and using a loss coefficient for negative flow (flow from the chamber into the culverts) of 0.31 for the *pipe_loss* components representing the portholes.

Figure 43 and Figure 44 summarize the results of filling (same headwater, tailwater, and valve opening time as for Figure 41) and emptying (same headwater, tailwater, and valve opening time as for Figure 42) simulations for Bay Springs Lock. As already mentioned, the predicted hawser forces for filling are less than 0.1 tons.

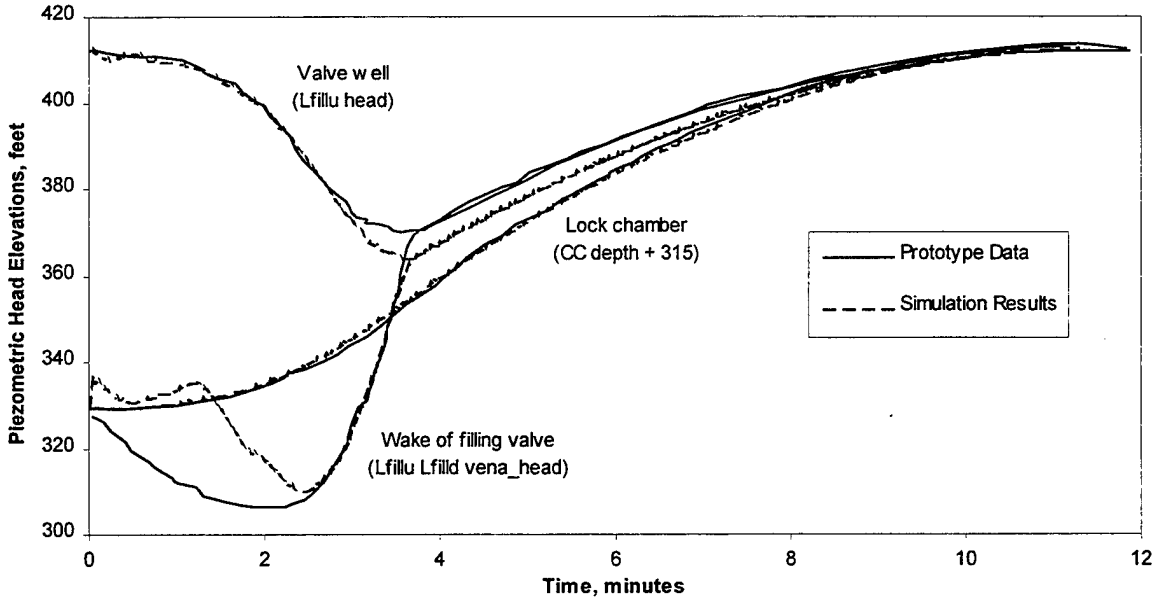


Figure 41: Simulation results compared with field data for filling at Bay Springs Lock (headwater = 412.3 feet, tailwater = 329.3 feet, valve opening time = 3.65 minutes)

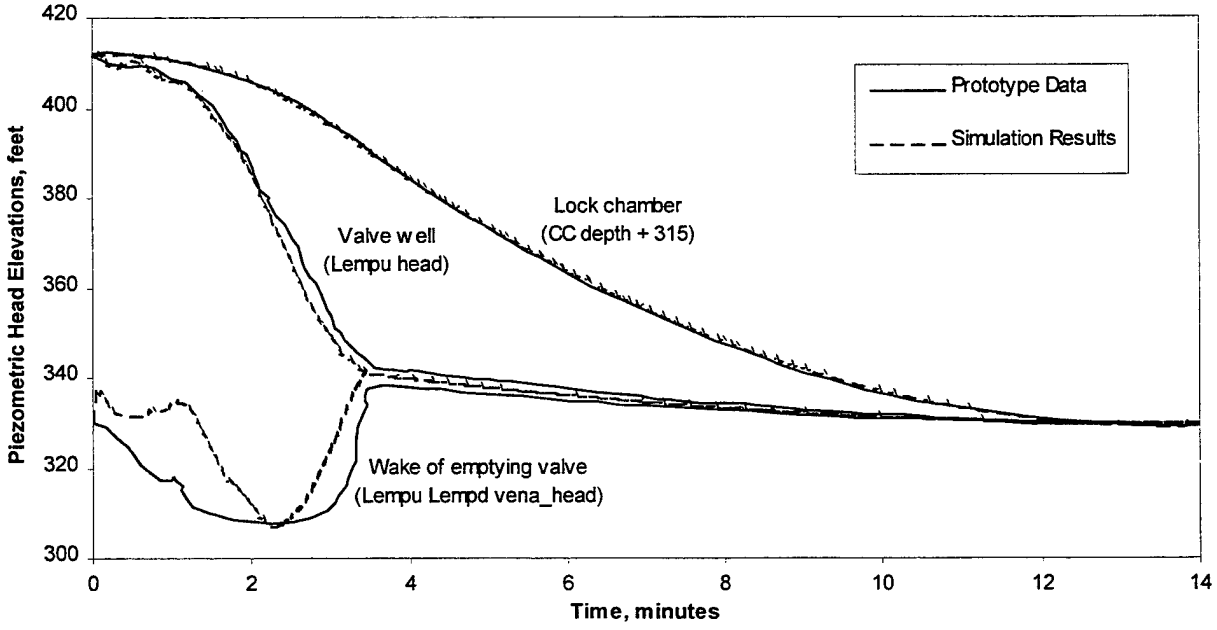


Figure 42: Simulation results compared with field data for emptying at Bay Springs Lock (headwater = 412.1 feet, tailwater = 330.4 feet, valve opening time = 3.42 minutes)

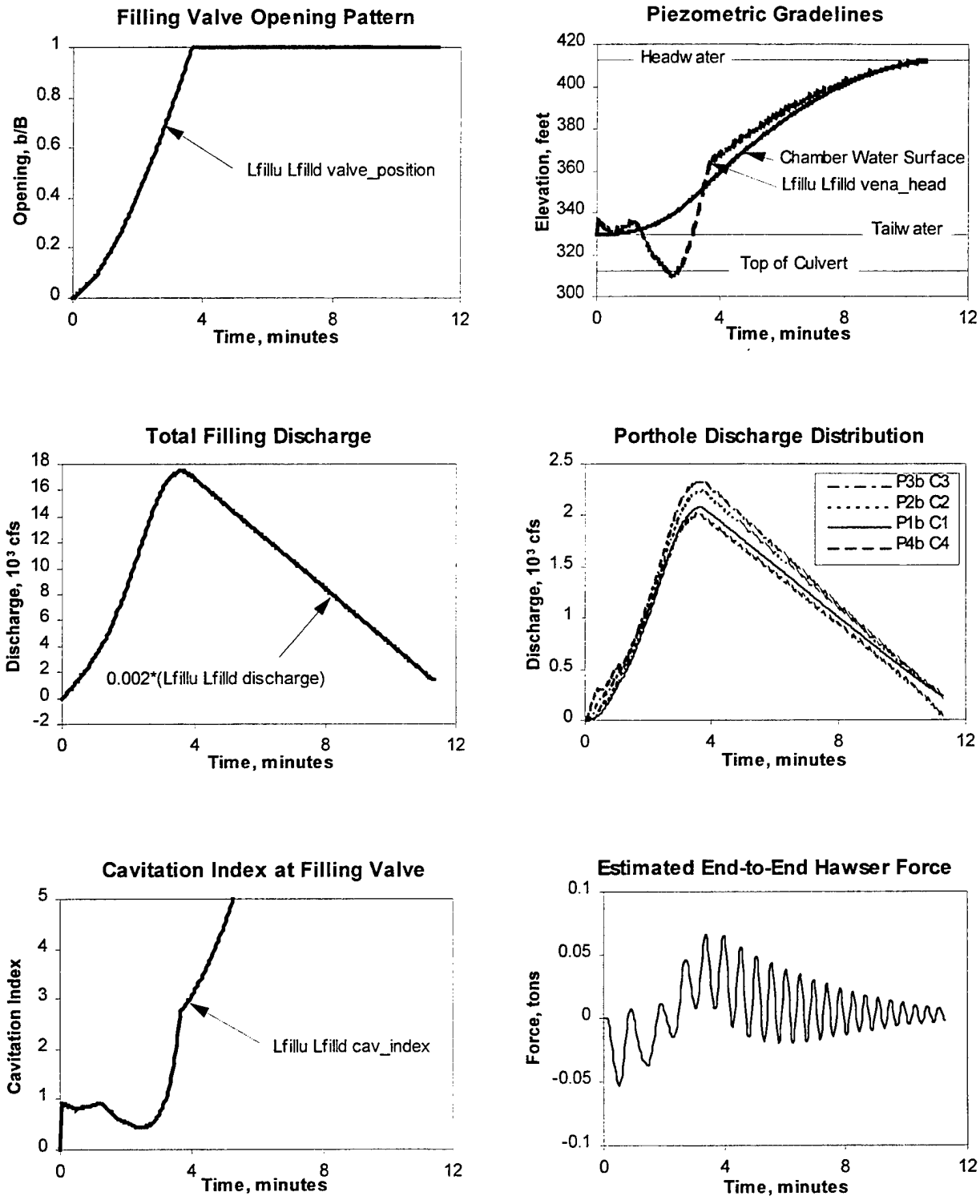


Figure 43: Filling results for Bay Springs Lock

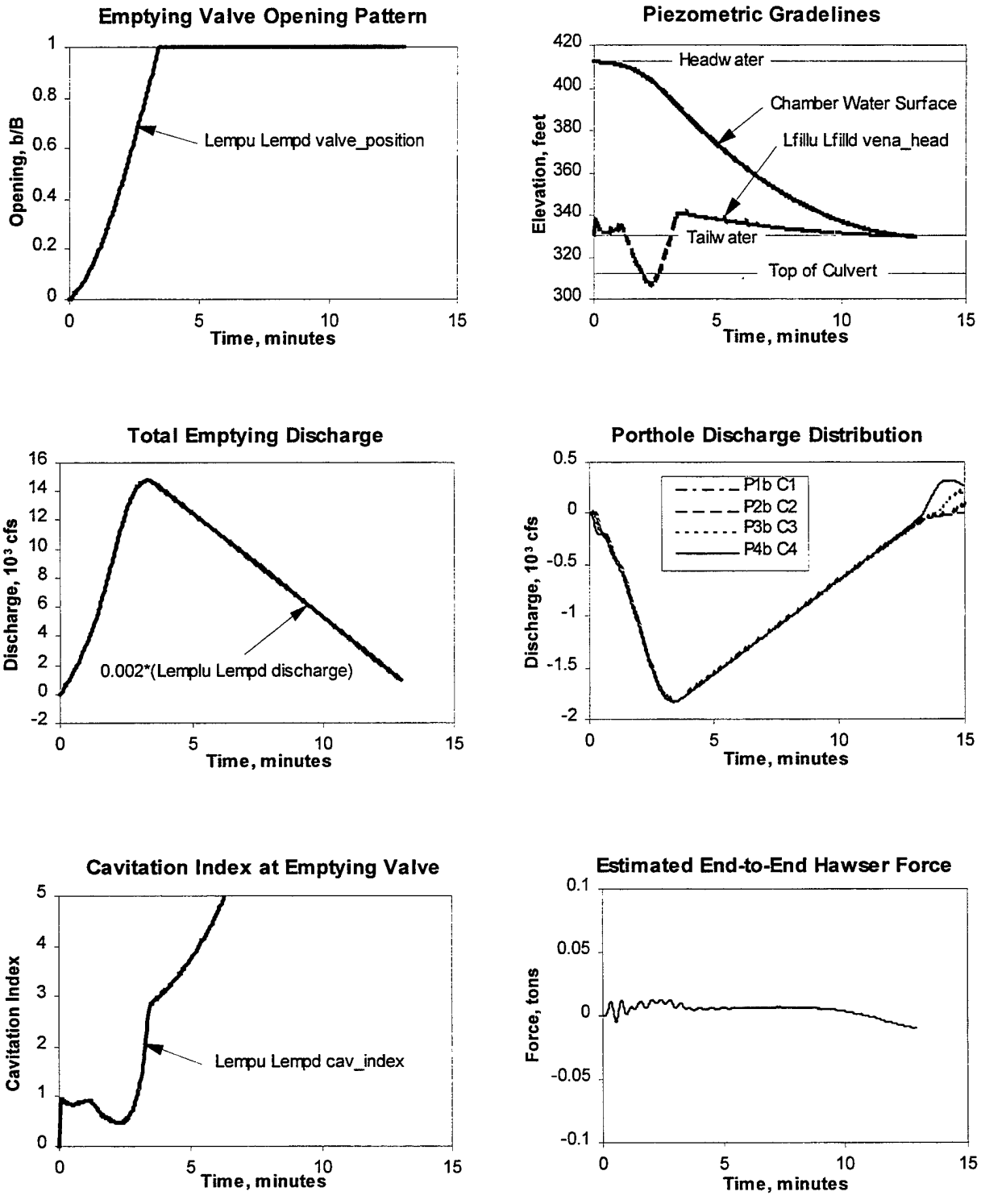


Figure 44: Emptying results for Bay Springs Lock

PART 2: LOCKSIM REFERENCE: OPERATION, OUTPUT, AND INPUT

CONVENTIONS

In this document, commands entered by the user when operating LOCKSIM are given in bold. Input file keywords, header titles and values are italicized. Graphics showing keyword requirements, referred to as "keyword charts," use left-hand brackets ([ ) to enclose keywords that are required, left-hand braces ({) to enclose keywords from which one choice must be made, and left-hand angle brackets (⟨) to enclose keywords that are optional. Consequently, in the example given in Figure 45, *keyword1* and *keyword2* are always required. Either *keyword3*, *keyword4*, or *keyword5* must be provided. Likewise, either *keyword6* or *keyword7* must be provided. *Keyword8*, *keyword9*, and *keyword10* are always optional. If *keyword3* is provided, then both *keyword3a* and *keyword3b* must be provided. If *keyword5* is provided, then *keyword5a* may optionally be provided. If *keyword6* is provided, then *keyword6a* and *keyword6b* may optionally be provided, but both must be provided. If *keyword9* is provided, then either *keyword9a* or *keyword9b* must be provided.

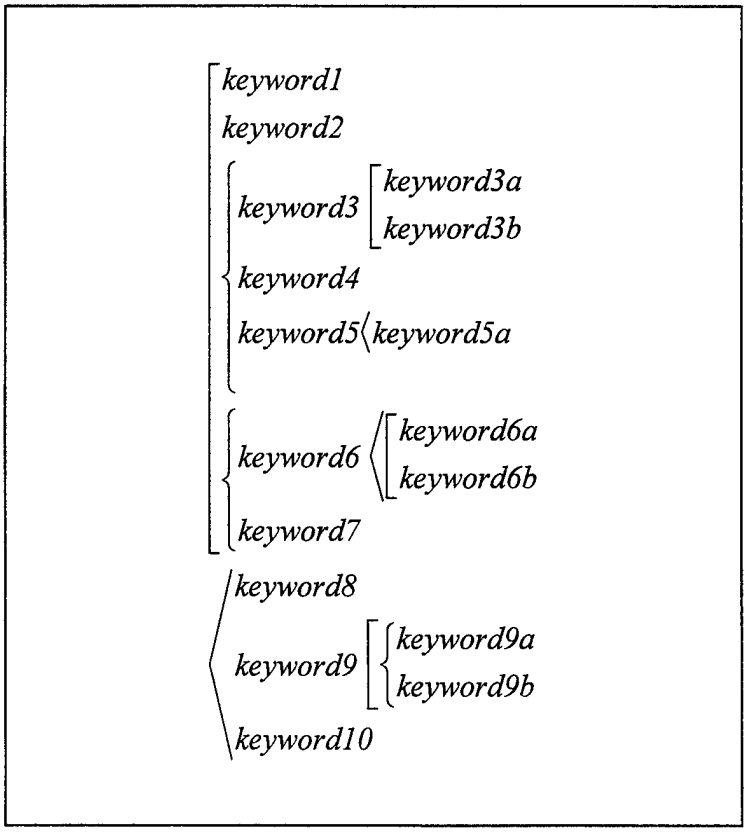


Figure 45: Example graphic (keyword chart) defining keyword requirements

OPERATION

LOCKSIM is operated interactively, allowing the user to examine results, change parameters, and decide whether to quit or continue at any point during a simulation. LOCKSIM's primary output is custom-specified by the user and easily imported into spreadsheet software for further analysis and plotting. The geometry, hydraulic characteristics, and boundary conditions of a network are described in an ASCII input file, which is read by LOCKSIM at the start of each simulation. The contents and format of the input file are described in detail in the "INPUT FILE FORMAT" section of this guide.

Because it is referenced in the sections below, the simulation "time step" is briefly described here. LOCKSIM solves partial differential equations for unsteady flow in conduits and free-surface channels using finite difference numerical techniques. The term "finite difference" refers both to the finite length increments (computational reaches) into which the pipe and free-surface components are divided and to the finite time increments (time step) into which the simulation time domain is divided. Starting with time equal to zero and the specified initial conditions, the numerical solution progresses through simulation time in discrete steps, with the results at each new time step depending on the hydraulic conditions at the previous time and the updated boundary conditions. The size of the simulation time step is specified in the input file as described in the *Constants section of this guide. The time step may be changed during the simulation when certain conditions are met, as described below in the "Run Unsteady" section.

The following descriptions assume that the user is familiar with basic MS-DOS commands and conventions, and with operating MS-DOS applications.

Starting a LOCKSIM Simulation

Command Line Input

LOCKSIM is typically operated after setting the current disk directory, using the MS-DOS CD command, to the directory containing the input file describing the network to be simulated. The current directory must also contain a copy of LOCKSIM.EXE (or LCKSIM32.EXE). Under the assumption that the current directory is C:\MyNetwork, the command for starting LOCKSIM at an MS-DOS prompt is

```
C:\MyNetwork\>LOCKSIM [input file name]
```

or

```
C:\MyNetwork\>LCKSIM32 [input file name]
```

```
|====================== LOCKSIM version 1.00, July 1998 ======================|

     Hydraulic simulation of navigation lock filling and emptying systems.

Copyright 1998 -- G.A. Schohl, Tennessee Valley Authority. All rights reserved.

Initial Commands (Press Letter):
 (q) Quit              (i) Specify Input File    (P) Specify Plot File
 (s) Run Steady        (S) Run Steady (NC)       (u) Run Unsteady
 (p) Report Periods    (d) Report Destinations

Current Input File = "mynet.sim"
Current Plot File  = ""
_
```

Figure 46: Initial screen displayed by LOCKSIM

where the command is shown in bold characters and the brackets ([]) indicate that the name of the input file is an optional command line parameter.

Initial Commands and Files

Figure 46 shows the text that is displayed on the monitor when LOCKSIM is started as follows

```
C:\MyNetwork\>LOCKSIM mynet.sim
```

The first several lines in Figure 46 are introductory information. The next four nonblank lines list the available "Initial Commands," which are summarized in Table 9. These lines are followed by the names of the input file and plot file, after these names are specified. For this example, the name of the input file has been specified on the command line. Otherwise, it would be necessary to specify its name using the initial command activated by pressing the letter **i** (or **I**)

Table 9: Initial Commands

Command letter (s)	Description
q or **Q**	Quit and return to MS-DOS prompt.
i or **I**	Enter name of network input file.
P	Enter name of output plot file.
s	Run steady-state simulator to obtain initial node heads and demands.
S	Run steady-state simulator with convective terms in free-surface component equations initially neglected.
u or **U**	Run unsteady-state simulator.
p	Specify period at which to generate tabular reports during unsteady simulation.
d or **D**	Specify destinations (monitor, file, or printer) for tabular reports.

before the simulation could continue. The specified input file must be an existing file on the disk.

The plot file is the file to which plot variables are saved during a simulation (see the *Constants and *Plot_Variables sections of this guide). If no plot variables are to be saved, then the plot file name is not required. Otherwise, the name of the plot file is usually specified within the input file. Alternatively, it may be specified by using the initial command activated by pressing the letter **P** (must be uppercase because lowercase **p** specifies a different initial command). The name of the plot file is not included in Figure 46 because the input file in which it is specified is not processed until the "Run Steady" or "Run Unsteady" command is selected. The specified plot file may already exist or may be a new file to be created.

The extensions (letters following the period in the file name) on the plot and input file names have no special significance in LOCKSIM. However, it is good practice to always use the same extension on files of the same type. For example, the extension "plt" can be used for all plot files and the extension "sim" can be used for all input files.

The designated plot and input files may be located in a directory different from the current directory if desired. In this case the directory path to the file must precede the file name. However, the length of the character string defining the path and file name must not exceed 40 characters.

Report Periods and Report Destinations

The available tabular reports that may be generated at every time step of an unsteady simulation are referred to as the "Nodes," "Components," "Continuity," and "Status" reports. The Nodes, Components, and Continuity reports are described in the "OUTPUT REPORTS" section of this guide. The Status report is described in the "Run Unsteady" section below. The periods at which these reports are automatically generated and the destinations to which they are generated (except for the Status report, which is displayed on the monitor only) may be specified using the **p** and **d** initial commands, respectively. In addition, the periods and destinations may be specified or changed at any time step during an unsteady simulation using the **p** and **d** commands at a simulation suspension, as described in the "Run Unsteady" section.

Figure 47 shows the text displayed by LOCKSIM when the "Report Periods" command (**p**) is selected. The periods are specified as number of time steps between automatic generation of reports. The periods shown in Figure 47 are the default values. A period of zero specifies that the report is never automatically generated. A period of 1 specifies that the report is generated

```
Report Periods (Press number; press z when done):
    (1) Nodes       =     0 timesteps
    (2) Components  =     0 timesteps
    (3) Continuity  =     0 timesteps
    (4) Status      =     1 timesteps
```

Figure 47: Input form for report periods

```
Report Destinations (Press number or letter; press z when done):

                    CRT        Printer        File         File Name
    Nodes       (1)  Y    (4)     N      (7)  N    (a)
    Components  (2)  Y    (5)     N      (8)  N    (b)
    Continuity  (3)  Y    (6)     N      (9)  N    (c)
```

Figure 48: Input form for report destinations

every time step, a period of 2 specifies that the report is generated every other time step, and so on. To change a report period, press number **1**, **2**, **3**, or **4** on the keyboard and follow the displayed directions. Press **z** to exit the input form and return to the initial commands display.

Figure 48 shows the text displayed by LOCKSIM when the "Report Destinations" command (**d**) is selected. The Nodes, Components, and Continuity reports may be directed to the monitor (CRT), printer, or to a text disk file. The destinations indicated in Figure 48 by either a Y (yes) or N (no) are the default values (all reports to the monitor only). The Y and N indications toggle from one to the other when a number in the range **1** to **6** is pressed on the keyboard. The N indications next to numbers (7) through (9) will toggle only after one or more file names are specified using the **a**, **b**, or **c** commands. Specification of only one file name is required because unspecified file names will default to the last one entered. For example, if a file name is specified for the Nodes report using the **a** command, then pressing **8** for the Components report will both toggle the N next to (8) to a Y and fill the file name field next to (b) with the name specified for the Nodes report. In this example, however, a Nodes report is not yet specified because both a Y next to (7) and a file name next to (a) are required before a Nodes report will be generated. It is necessary to press **7**, which toggles the N next to (7) to a Y, before a Node report will be sent to the file specified next to (a). Press **z** to exit the input form and return to the initial commands display.

Run Steady

A network is assumed to be in a steady-state condition at the start of an unsteady simulation. Initial conditions specified in the input file consist of initial heads for nodes with supply or demand boundary conditions and initial demands for nodes with head or pressure boundary conditions (see the *Nodes section of this guide). These values are usually determined from a steady-state simulation conducted before the unsteady simulation. LOCKSIM includes a steady-state simulator with an option to automatically write the steady-state results into the input file as initial conditions for the subsequent unsteady simulation. This simulator is invoked by the initial commands "Run steady" (**s**) or "Run Steady (NC)" (**S**).

The "Run Steady (NC)" initial command is different from the "Run Steady" initial command only for networks that include *open_channel* or *river_channel* components. It tells the steady-state simulator to solve the network first with the convective terms (velocity head terms) in the governing equations for *open_channel* and *river_channel* components neglected. Then, using the nonconvective solution as the starting point, the steady-state simulation is repeated with

```
Opening input file " mynet.sim ". . .
Reading *CONSTANTS. . .
Reading *COMPONENTS. . .
Reading *NODES. . .
Reading *FUNCTIONS. . .
Reading *PLOT_VARIABLES. . .
Ordering components for forward sweep. . .
Setting initial conditions and reordering links if necessary. . .
Balancing steady flows and pressures. . .

Iter =   1  MaxDQ =  7.26e+000  Comp = node1 node4  Sect = 2
Iter =   2  MaxDQ = -3.61e+000  Comp = node1 node4  Sect = 1
Iter =   3  MaxDQ = -1.79e+000  Comp = node1 node4  Sect = 2
Iter =   4  MaxDQ =  1.09e+000  Comp = node2 node3  Sect = 1
Iter =   5  MaxDQ = -7.35e-001  Comp = node4 node5  Sect = 1
Iter =   6  MaxDQ = -2.04e-001  Comp = node4 node5  Sect = 1
Iter =   7  MaxDQ = -2.26e-002  Comp = node4 node5  Sect = 1
Iter =   8  MaxDQ = -8.64e-004  Comp = node4 node5  Sect = 1
Iter =   9  MaxDQ = -2.51e-005  Comp = node4 node5  Sect = 1
Iter =  10  MaxDQ = -7.22e-007  Comp = node5 node6  Sect = 2
Iter =  11  MaxDQ = -2.07e-008  Comp = node4 node5  Sect = 1
Iter =  12  MaxDQ = -5.95e-010  Comp = node5 node6  Sect = 2
Iter =  13  MaxDQ = -1.71e-011  MaxDP = -3.19e-011  *CONVERGED*
Save Steady State (Y/N) ?
```

Figure 49: Text displayed when "Run Steady" is selected

the convective terms included. This approach to obtaining a balanced steady-state solution is sometimes necessary when free-surface components are used. It is used by default, whether the **s** or **S** command is selected, when initial, or starting, conditions are unspecified for each node in the input file (see "Boundary and Initial Conditions" in the *Nodes section of this guide).

Figure 49 shows the text that is displayed on the monitor when the **s** initial command is selected. The first several lines are printed during the preparation stage to indicate progress in processing the input file. After the last section of the input file is read, the component, or link, order for the forward sweep, which refers to the first half of the matrix solution process implemented in LOCKSIM, is established. Finally, the specified starting, or initial, conditions for each component are computed based on the specified initial node heads and demands, and the iterative solution stage is entered. The results of each iteration are displayed as they are generated. Each line shows the iteration number, the maximum correction to an unknown variable (discharge correction, dQ; pressure correction, dP; or auxiliary variable correction, dX) during that iteration, and the component and computational section at which the maximum correction was applied. As indicated in Figure 49, the maximum correction should get smaller and smaller with each iteration until it becomes negligible, at which point the solution has converged. The steady-state solution uses the default tolerances in LOCKSIM. The solution tolerances specified using the *dh_max*, *dq_max*, *dx_max* keywords in the **CONSTANTS* section of the input file are used only for the unsteady solution.

As indicated in Figure 49, once a converged steady-state solution is obtained, the user is asked whether or not to save the steady-state conditions to the input file, or to a copy of the input

```
Choose Report(s) (Press number or letter):
 (n) Nodes       (c) Components      (y) Continuity     (1) n and c
 (2) n and y     (3) c and y         (4) n, c, and y    (z) done
```

Figure 50: Input form for selecting reports

file (press letter **Y** or **N**). The results will not be available as initial conditions to a subsequent unsteady simulation until they are saved to a file (steady results are not maintained in computer memory for the unsteady solution; it gets its initial conditions by rereading the input file).

The user is next asked whether or not steady-state reports are desired. If the answer is no (**N**), then the initial commands are again displayed, as shown in Figure 46. If the answer is yes (**Y**) then the report destinations input form is displayed, as shown in Figure 48. After this form is terminated by pressing **z**, the user is asked to select the desired reports using the input form shown in Figure 50. After this form is terminated by pressing **z**, the initial commands are again displayed. The "Run Unsteady" (**u** or **U**) command is typically selected next.

It is often worthwhile to rerun the steady-state simulator using the saved results from the earlier steady-state simulation as starting conditions. The succeeding solution often converges with smaller last corrections then the earlier solution, providing more precise steady-state initial conditions for the subsequent unsteady simulation. The steady-state simulator may be invoked as many times as desired.

Run Unsteady

The **u** or **U** initial commands invoke the unsteady simulator. An unsteady simulation should not be started until the initial conditions provided in the input file describe a steady-state condition (see the "Run Steady" section above and the *Nodes section of this guide).

Figure 51 shows the text that is displayed on the monitor when the **u** initial command is selected. As for the steady-state simulator, the first several lines are printed during the preparation stage to indicate progress in processing the input file. After the last section of the input file is read, the component, or link, order for the forward sweep is established and the initial conditions for each component are computed based on the specified initial node heads and demands. The computed initial discharges into and out of each node are then added to check that inflow equals outflow. If all nodes pass this continuity test then "OK" is printed, as indicated in Figure 51. Then the current simulation time and time step (both initially zero), along with the phrase "Simulation suspended," is printed and a list of commands is displayed. Table 10 summarizes the available commands.

Node Continuity Test

If one or more nodes fail the continuity test, then the label for each node is listed along with the size of its discharge imbalance. The user is asked whether to continue the simulation or

```
Opening input file " mynet.sim ". . .
Reading *CONSTANTS. . .
Reading *COMPONENTS. . .
Reading *NODES. . .
Reading *FUNCTIONS. . .
Reading *PLOT_VARIABLES. . .
Ordering components for forward sweep. . .
Setting initial conditions and reordering links if necessary. . .
Checking initial flow balance at nodes. . . OK

Time = 0        sec, Step = 0          .....Simulation Suspended.....

Commands (Press Letter):
 (q) Quit           (s) Show Variable     (C) Change Timestep
 (c) Continue       (n) Next Suspension   (D) Debug Level
 (g) Go and Quit    (P) Print Reports     (d) Destinations      (p) periods
```

Figure 51: Text displayed when "Run Unsteady" is selected

to quit. One reason to continue is to use the **P** command when the simulation is suspended at time equal to zero to examine the Nodes and Components reports, which may show the source of the flow imbalance. Otherwise, unless the flow imbalance was deliberately imposed, the simulation should be terminated and restarted after the problem is fixed.

One or more flow imbalances usually indicate that the specified initial node heads and demands do not define a steady-state condition. Sometimes, however, flow imbalances occur because, for some components (tees and free-surface components for example), known upstream and downstream heads do not always determine a unique flow direction. Consequently, the computed initial steady-state discharge for one or more of these components may flow in the wrong direction (see "Boundary and Initial Conditions" in the *Nodes section of this guide). This problem is addressed by specifying starting flows using the *iQ* keyword (see the individual component write ups in the *Components section of this guide). The steady-state simulator in

Table 10: Available Commands at Simulation Suspension

Command letter (s)	Description
q or **Q**	Quit, write plot file, and return to MS-DOS prompt.
s or **S**	Show value of specified plot variable.
C	Change size of computational time step.
c	Continue simulation to time of next suspension.
n or **N**	Enter time for next simulation suspension.
D	Specify level of output printed to the monitor during each iteration.
g	Continue simulation to time of next suspension and then quit.
P	Print tabular reports.
d	Specify destinations (monitor, file, or printer) for tabular reports.
p	Specify period at which to generate tabular reports during unsteady simulation.

```
Time = 5          sec, Step = 10

Commands: (s) suspend, (n,c,y) nodes, components, or continuity reports
```

Figure 52: Status report showing current time and time step number

LOCKSIM always specifies the *iQ* keyword for those components for which it is available when it automatically saves results to the input file.

Simulation Suspension

An unsteady simulation may be suspended at any simulation time step, and commands may be issued. The first "simulation suspension" occurs automatically at time step equal to zero, as indicated in Figure 51. The next simulation suspension occurs at a time specified by the user using the "Next Suspension" command (**n** or **N**). In addition, as indicated in Figure 52, if the letter **s** is pressed at any point during the calculations, the simulation will be suspended at the next time step. A simulation is terminated by using the **q** command at any simulation suspension. The **c** command is used to continue the calculations to the next simulation suspension. The **g** command is used to continue the calculations to the next simulation suspension and then quit. If the current next suspension time is less than or equal to the current simulation time, then the **c** or **g** command will cause the simulation to progress one time step before pausing at another simulation suspension.

Tabular reports are printed to specified destinations at specified periods using the **P**, **d**, and **p** commands, which display the input forms illustrated in Figure 50, Figure 48, and Figure 47, respectively. It is usually desirable to specify a period greater than one for the Status report before continuing an unsteady simulation beyond time step zero. The Status report is illustrated in Figure 52, where it is shown for simulation time equal to 5 seconds. It is printed just before the calculations for the indicated time step begin. The Status report also lists commands that are available during the calculation phase of a simulation. These include the **s** command, which causes a simulation suspension to occur at the end of the current time step, and the **n**, **c**, and **y** commands, which cause the Nodes, Components, or Continuity reports (see Figure 50), respectively, to be printed to the currently specified report destinations at the end of the current time step.

The **s** or **S** command in Table 10 displays the current value of any simulation variable that can be identified using the plot variable format described in the *Plot_Variables section of this guide. The **C** command permits changing the value of the simulation time step. However, the time step cannot be changed at a simulation suspension for networks containing *moc_pipe* components (see the *Components section of this guide). The **D** command permits changing the level of output printed to the monitor during each iteration. A nonzero value is specified when additional output may give clues as to why a solution is difficult or impossible to obtain. Figure 53 shows the additional printed output for each iteration when the debug level is set to 1. Users of LOCKSIM should never need to specify values of *debug* greater than 1, which provide output useful only during code development and debugging of new features.

```
Iter =   1  MaxDQ = -1.53e-002  Comp = node5 node6  Sect = 2
Iter =   2  MaxDQ = -6.07e-006  Comp = node5 node6  Sect = 2
Iter =   3  MaxDP =  4.96e-011  Comp = node5 node6  Sect = 1
Iter =   4  MaxDQ =  6.70e-017  MaxDP = -5.50e-016  *CONVERGED*
```

Figure 53: Additional output at each iteration with debug level set to 1

Solution Convergence Problems

In LOCKSIM all components except for the *moc_pipe* are solved using an implicit, iterative solution technique. The implicit solution is obtained using the Newton-Raphson iterative technique (see, for example, Carnahan et al., 1969) on the governing nonlinear equations. This technique requires solution of a matrix equation for corrections to the unknown variables at each iteration. The double-sweep method (Cunge et al., 1980), which is a modified tridiagonal approach, is used to efficiently solve the matrix equation.

Occasionally, solution convergence problems, indicated by the run-time messages "Maximum number of iterations reached without convergence" and "Maximum convergence achieved," may occur during an unsteady simulation. The keywords *dh_max*, *dq_max*, *dx_max*, *iter_max*, *converge_freq*, and *converge_mult*, which are specified in the **CONSTANTS* section of the input file, all are related to convergence issues. The first step in resolving convergence problems is to make sure that the specified solution tolerances (*dh_max*, *dq_max*, and *dx_max*) are reasonable. The second step is to set the *debug* keyword equal to 1, which causes the results of each iteration to be written to the monitor. If it appears that the solution is converging but needs more iterations, then specification of the keywords *iter_max*, *converge_freq*, or *converge_mult* should be considered. If the solution is diverging rather than converging, then specification of these keywords is unlikely to make a difference. See "Solution Convergence" in the *Constants section of this guide for additional information.

OUTPUT REPORTS

LOCKSIM's most useful outputs are the time-varying data specified in the **PLOT_VARIABLES* section of the input file. The specified solution variables are saved to an ASCII disk file in a columnar format that is readily imported into spreadsheet software for further analysis and plotting. The name of the disk file is specified using the *plot_file* keyword in the **CONSTANTS* section of the input file (see the *Constants section of this guide.). Because it is described thoroughly in the *Plot_Variables section of this guide, the plot variables feature is not discussed further in this section.

Comprehensive tabular results for any time step are available through the Nodes, Components, and Continuity reports, which may be directed to the user's monitor or printer, or to a file. The user selects the report destinations and the frequency at which they are printed using the "Report Destination" and "Report Period" commands as described in the OPERATION section of this guide. Additional outputs include various error messages that may be printed both as the input file is processed and during a simulation.

Error Messages

As the input file is read and processed, nearly every item on every line is checked in some manner. Numerical values must be reasonable (or at least possible) and within valid ranges defined for them. Keywords must be valid choices for the component, node, function, cross section, or plot variable for which they are used. Every keyword must have an assigned value. Certain keywords with values must be provided for certain components. A discrete function must have the same number of y-values as x-values. The list goes on and on. In all, over 100 error messages are defined within LOCKSIM, with most related to processing of the input file.

Error messages are printed both to the monitor and to a disk file named "LOCKSIM.MSG," which may be viewed using any text editor. Each time LOCKSIM is started, the file LOCKSIM.MSG is deleted if it resides in the current directory. It is then recreated if needed and is available for inspection until the next time LOCKSIM is started.

Nodes Report

Figure 54 shows a sample Nodes report. The first text column lists the node labels, in the order in which they are defined in the **NODES* section of the input file. The remaining columns give the node elevation, hydraulic (piezometric) gradeline elevation (H), pressure (P), demand, vapor volume, and status. When node pressure drops to vapor pressure, the status column, which is normally blank, contains the text "VAPOR."

>>========== Results: Time = .1 sec, Time Step = 1 ==========<<

NODES	Elev(f)	H(f)	P(psig)	Demand(cfs)	Vapor(f^3)	Status
sump	60	68	3.4652	-23.6093	0	
300	62	64.04724	.8867599	0	0	
301	62	97.51198	15.38201	0	0	
305	73	68	-2.16575	0	0	
306	82.09	94.0118	5.163929	0	0	
310	82.09	89.36555	3.151405	0	0	
320	82.09	88.79964	2.90628	0	0	
321	82.09	88.5887	2.814913	0	0	
125	86	88.5887	1.121296	0	0	
322	82.09	88.4796	2.767654	0	0	
323	82.09	88.11591	2.610125	0	0	
501	80.9	86.99622	2.640577	0	0	
601	74.3	86.99622	5.499367	0	0	
603	74.3	82.04295	3.353861	0	0	
551	73	82.09782	3.94072	0	0	
552	73	86.7671	5.963219	0	0	
553	73	86.99622	6.062462	0	0	
557	73	68	-2.16575	0	0	
return	66	68	.8663	0	0	
333	82.09	80.07979	-.870723	0	0	
334	78	79.19934	.519494	0	0	
hopper	78	78	0	23.60925	0	

Figure 54: Sample Nodes report

Components Report

Figure 55 shows a Components report with example output for every type of component. Because different components require different outputs, the Components report is divided into separate sections. The "PIPES" section includes output for both *moc_pipe* and *imp_pipe* components. The remaining sections include output for only one type of component each. For brevity, Figure 55 includes sample output for only one component of each type except that output for two tees is included.

The first two columns in the Components report identify each component by the labels of their upstream and downstream nodes. For the tees and manifolds, the two flow paths through each are identified as if they were two separate components. For "PIPES," "OPEN CHANNELS," and "RIVER CHANNELS," the third column indicates the computational section number (counting from section zero upstream) or label. Other outputs in Figure 55 are identified by the labels defined in Table 11. See the individual component descriptions in the *Components section of this guide for additional information on the individual outputs.

```
>>========== Results:  Time = .1       sec, Time Step = 1      ==========<<

PIPES              Sect  Q(cfs)  V(f/s)  E(f)  P(psig)  Vapor(f^3)  Re      DW-f
 us       valve      0  10.001       1  200.05  43.32          0   295157
                     1  9.9989  .9999  200.06  43.32          0   295108
                     2  10.004       1  200.01   43.3          0   295249
                     3  9.9873  .9987  200.18  43.37          0   294764

VALVES             Q(cfs)   Eu(f)   Ed(f)  Eu-Ed(f) Pu(psig) Pd    Position
 309      320      23.609  90.478  89.912   .56591   3.151  2.906       1

CHECK VALVES       Q(cfs)  VH(f)   Eu(f)   Ed(f)  Eu-Ed(f) Pu(psig) Pd  Position
 PDis     CVDis    1.8053  .0336  620.56  620.54   .01682   95.52  95.51   OPEN

PIPE LOSSES        Q(cfs)  VHu(f) VHd(f)  Eu(f)   Ed(f)  Eu-Ed(f) Pu(psig) Pd
 sump     300      23.609  1.E-7  3.593      68  67.641   .35934  3.465  .8868

STORAGES           Qu(cfs)  Qd    Qin   WSEL(f) Pu(psig) Pd     Status  Vu(f^3)
 Lempu    Lwelld   -72.14      0 -72.14  378.39  38.46  38.46   NORMAL        0

REVERSE TAINTERS  Q(cfs)   Eu(f)   Ed(f) Eu-Ed(f) Hvc(f) Pvc(psig)   Ki    b/B
 Lempu    Lempd    225.75   378.4  302.95  75.454  305.16   9.926   .7689  .0158

OPEN CHANNELS  Sect   Q(cfs)  V(f/s)  E(f)     H(f)    d(f)   Fr   Crmin  DW-f
 upstream        downstream
                  0    1211  4.509  305.687  305.371  5.371  .34  5.2643 .0172
                  1  1213.7  4.545  305.662  305.341  5.341  .35  5.2207 .0173
                  2  1216.1  4.578  305.638  305.312  5.312  .35  5.1785 .0173

RIVER CHANNELS Sect   Q(cfs)  V(f/s)  E(f)     H(f)    d(f)   Fr   Crmin  DW-f
 flume          us_weir
         STA_14+10   451.11  .4293  302.369  302.366  32.37  .01  1.9352
         STA_14+40    356.6  .3407  302.262   302.26  32.26  .01  2.3259
         sta_14+65   232.03  .1934  302.146  302.145  32.15  .01   2.198

TEES               Qu(cfs) Qd(cfs)   Eu(f)   Ed(f)  Eu-Ed(f)   K      Status
 4        8             30  19.731  773.45  765.49   7.9569  .5689  DIVERGING
 4        9             30  10.269  773.45  765.46   7.9938  .5715

 8        3         19.731      20  765.24   759.4   5.8376  .9391  CONVERGING
 8        7         19.731  -.2693  765.24  763.82   1.4215  .7104

MANIFOLDS          Qu(cfs) Qd(cfs)   Eu(f)   Ed(f)  Eu-Ed(f)   Ku     Kd
 2        3         37.474  22.245  766.16  755.11    11.05      0    .01
 2        4         37.474  15.229  766.16  758.11    8.046    .01  1.141
                        Status = DIVERGING     Qb(cfs) = .03046
```

Figure 55: Sample Components report

Continuity Report

The Continuity report is examined typically only for debugging purposes, to make sure that mass continuity is satisfied at all nodes. Figure 56 shows a sample Continuity report. Current imbalances in flow into and out of each node and in accumulated vapor volume are

Table 11: Labels Used to Identify Outputs in Components Report

label	Description
Q, Qu, Qd	Discharge, upstream Q, downstream Q.
Qi	Net inflow.
Qb	Q in single branch of a manifold.
E, Eu, Ed	Energy gradeline elevation (piezometric gradeline elevation plus velocity head), upstream E, downstream E.
Eu-Ed	Energy difference.
H	Hydraulic (piezometric) gradeline elevation.
P, Pu, Pd	Pressure, upstream P, downstream P.
Hvc, Pvc	H and P at vena contracta of *rev_tainter* component.
V	Velocity.
Vh, Vhu, Vhd	Velocity head ($V^2/2g$), upstream Vh, downstream Vh.
Vapor	Vapor cavity volume.
Re	Reynolds number.
DW-f	Darcy-Weisbach friction factor (no value if alternative friction option is specified).
WSEL	Water surface elevation.
Vu	Underflow volume for *storage* component.
d	Liquid depth.
Fr	Froude number $[V(T/gA)^{0.5}]$.
Crmin	Courant number, C_r, based on Δx of longest adjacent reach.
K, Ku, Kd	Loss coefficient, upstream K, downstream K.
Ki	Cavitation index.
b/B, Position	Valve opening position.
Status	Condition report.

listed. Normally, all outputs are zero or nearly zero. A significant imbalance indicates a probable bug in the LOCKSIM computer code, which should be brought to the author's attention.

```
>>========== Results:  Time = .1       sec, Time Step = 1      ==========<<

>>====== Node Continuity Report ======<<

Simulation Time = .3       sec
      Time Step = 3

                ------ Imbalances ------
  Node          Flow (cfs)    Vapor (f^3)

 sump                    0          ----
 300                     0          ----
 301                     0          ----
 305                     0          ----
 306                     0          ----
 310                     0          ----
 320                     0          ----
 321                     0          ----
 125                     0          ----
 322                     0          ----
 323                     0          ----
 501                     0          ----
 601                     0          ----
 603                     0          ----
 551                     0          ----
 552                     0          ----
 553                     0          ----
 557                     0          ----
 return                  0          ----
 333                     0          ----
 334                     0          ----
 hopper                  0          ----
```

Figure 56: Sample Continuity report

INPUT FILE FORMAT

Overall Structure

The input file is divided into sections, each with its own input format. The start of each new section is indicated by its header title, which is one of the following:

**CONSTANTS*	Physical and simulation control constants
**COMPONENTS*	Component descriptions for network (pipes, valves, channels, losses, etc.)
**NODES*	Elevations, boundary and initial conditions for nodes
**FUNCTIONS*	Function descriptions
**CROSS_SECTIONS*	Definition of cross sections for channel components
**PLOT_VARIABLES*	List of variables for which values are saved for later plotting
**END*	Denotes end of input

The first three sections must be **CONSTANTS*, **COMPONENTS*, and **NODES* in that order. The header title **END* indicates the end of the input. Other sections may be included in any desired order. The **PLOT_VARIABLES* section is always optional. The **FUNCTIONS* and **CROSS_SECTIONS* sections are required only if components or nodes refer to particular functions or cross sections which must then be defined.

A header title must be the first item on its line. Any text following the header on the same line is ignored.

General Input Rules

Each line of the input file consists of a number of data "items" that define header titles, labels, type identifiers, keyword-value pairs, comments, and data clusters. Items are included in free format separated by spaces or any number of the following characters, which are considered equivalent to spaces:

, ( ) [ ] \ |

or by one of the special characters:

= ! { }

The equal sign (=) separates and identifies keywords and values, the exclamation point (!) introduces a comment, and the curly braces ({ and }) bracket and define data clusters. A maximum of thirty total items may be included on any one input line; for this purpose, each keyword with its value counts as two items, a comment (the whole comment) counts as one item, each curly brace counts as one item, and each value within a data cluster counts as one item. Except for the thirty item limit, the number of items included on any one line is arbitrary. However, labels and associated type identifiers must be on the same line, and every keyword must be on the same line as its value. The maximum length of an input line is 256 characters.

Header titles, type identifiers, and keywords are case-independent and may be included in lower case, upper case, or mixed case as desired. Also, these items may be abbreviated as desired as long as enough characters are supplied to uniquely identify the item within its list.

Component, node, function, and cross section labels, on the other hand, are not case-independent and may not be abbreviated. Labels are strings which may include any characters except the separation characters listed above. The only restriction on the length of a label is the maximum input line length of 256 characters. However, long node and component labels may be abbreviated in the output reports.

The format for values associated with keywords is

KEYWORD=value

The equal sign indicates that the preceding item is a keyword and the following item is a value to be assigned to the *KEYWORD*. Values are usually numbers, but are sometimes character strings or data clusters. Two keywords may be assigned the same value by using the following format:

KEYWORD1=KEYWORD2=value

Similarly, more than two keywords may be assigned the same value. Note that spaces may be included as additional separators if desired, but the equal sign is sufficient to separate keywords from their values.

A data cluster is a series of items within a set of curly braces. Once a left curly brace is detected, all items which follow are considered part of the data cluster until a right curly brace is detected. The left curly brace must be on the same line as the keyword. The right curly brace need not be on the same line as the left curly brace, which means that a data cluster may be input over multiple lines. Data clusters are used, for example, to input the pairs of x-y points which define a function:

xy_pairs={ (0,5) (3,9) (8,14) (10,22) }

In this example the keyword *xy_pairs* is set equal to a data cluster. Note the liberal use of optional spaces, commas, and parentheses to separate individual items. This practice improves the readability of the data.

Comments may be included anywhere in an input file. These are indicated by an exclamation point (!). All text following an exclamation point on the same line is considered a comment and is ignored. An entire line is made a comment by including an exclamation point as the first non-space character on the line.

Blank lines are ignored and may be included at will in the input file.

Input And Output Units

Units for input and output values are specified as either *SI* or *English* in the **CONSTANTS* section of the input file. Table 12 shows the units associated with each base quantity. The units for other input and output quantities are the same as for the equivalent quantity in Table 12. For example, the units for diameter, elevation, and pipe roughness are all the same as the units for length. Likewise, the units for wave speed are the same as the units for velocity. The units for simulation time are specified separately in the **CONSTANTS* section and do not depend on the choice of *SI* or *English* for the base units.

The unit symbols in Table 12 should be mostly familiar. Input and output pressures are gage pressures (e.g., psig) except where otherwise indicated.

Table 12: Input and output units

Base Quantity	English Units	SI Units
acceleration	f/s^2	m/s^2
area	f^2	m^2
discharge	cfs	m^3/s
kinematic viscosity	f^2/s	m^2/s
length	f	m
pressure	psi	kPa
time	s,min,hrs, or days	s,min,hrs, or days
unit weight	lb/f^3	N/m^3
velocity	f/s	m/s
volume	f^3	m^3

*Constants

Physical and simulation control constants and calculation and input-output options are assigned in the **CONSTANTS* section using keyword-value pairs as illustrated below:

**CONSTANTS*
keyword1=value1, keyword2=value2, ...
keyword3=value3, keyword4=value4, ...
.
.
.
**COMPONENTS ! next section header title*

Because the **CONSTANTS* section must be the first section included in the input file, its header title is actually optional. The header title may be preceded by blank lines or lines containing comments if desired. Recall that keyword specifications are case-independent and that keywords may be abbreviated.

Table 13 lists the available keywords and default values, and provides brief descriptions for each. For many keywords, additional description is provided in the following paragraphs. Default values for physical constants are given in internal units in Table 13. However, the units for user-specified values must be those listed in Table 12, depending on whether *SI* or *English* units are activated and on the choice of *time_units* (for the keyword *time_step).* "Valid range or choices" indicates acceptable values for the keywords. To maintain as much flexibility as possible, ranges are in most cases specified only where values outside the range could cause run-time errors during simulations, rather than to ensure that physically realistic values are provided. This approach permits the user to, for example, turn off vapor cavity calculations by specifying a large negative absolute vapor pressure.

The default values are used for constants that are not assigned values. Simulation *time_step* is the only keyword that does not have a default value and must be specified in every input file. Keywords that usually are specified include *dh_max*, *dq_max*, *dx_max*, and *plot_file*. Keywords that are specified frequently include *autofix_k's*, *barometric_pressure*, *io_units*, *kin_viscosity*, *overwrite*, *plot_field*, *plot_line*, *plot_labels*, *report_line*, *time_units*, *unit_weight*, *vapor_pressure*, *wf_time*, and *xsect_file*. The remaining keywords are specified infrequently or rarely.

Simulation *time_step*

The *time_step* keyword must be specified in every input file. Its value is usually a constant but a data cluster may be used to specify a time step that varies with simulation time. In this case, the first item in the data cluster is the initial time step. The following items are pairs of values, in which each pair consists of a simulation time value and a new time step value. An example follows:

Table 13: Keywords Specified in **CONSTANTS* Section

KEYWORD	DESCRIPTION	DEFAULT	VALID RANGE OR CHOICES
autofix_k's	automatic conveyance fix for negative dK/dd's in riverine cross sections	off	on, off
autofix_xsects	automatic fix for dry beds (negative depth in cross section)	off	on, off
barometric_pressure	barometric pressure (absolute pressure)	2116 psfa	any absolute pressure
converge_freq	check progress towards convergence every *converge_freq* iterations	10 iterations	> 0
converge_mult	check progress by comparing largest current correction multiplied by *converge_mult* with largest correction the last time progress was checked	5	> 0
debug	print debug reports and messages	0	0, 1, 2, or 3
depth_min	minimum depth in open or river channel	0.1 f	> 0
dh_max	head correction tolerance	1.e-10 f	> default
dq_max	flow correction tolerance	1.e-10 cfs	> default
dx_max	auxiliary variable correction tolerance	1.e-8	> default
gravity	gravitational acceleration, g	32.174 f/s^2	> 0
io_units	global units choice	English	English or SI
iter_max	maximum double sweep iterations per time step	150	> 0
kin_viscosity	kinematic viscosity, ν	1.206e-5 f^2/s	> 0
loss_re	laminar Reynolds number for losses and valves	1000.	> 0
overwrite	overwrite or append existing output files	warn	on, off, or warn
pipe_re	laminar Reynolds number for pipes and channels	2000.	> 0
plot_field	width in plot file for each value (maximum significant figures)	10 columns	> 0
plot_file	name of file to which plot variables are saved	none	valid filename
plot_labels	format for writing plot variable labels into plot file	column	column, row
plot_line	width in plot file for each line of saved data	80 columns	> 0
pv_end_time	simulation time to stop saving plot variables	-1 s	>=0
pv_period	simulation time between plot variable saves	0	>=0
pv_start_time	simulation time to start saving plot variables	0	>=0
report_line	maximum width of lines in tabular reports	80 columns	>= 80 and <= 256
time_step	constant or time-varying computational time step, Δt	none	> 0
time_units	simulation time units	seconds	seconds, minutes, hours, days
unit_weight	unit weight of fluid, $\gamma=\rho g$	62.3736 lb/f^3	> 0
vapor_pressure	vapor pressure (absolute pressure)	52.33 psfa	any absolute pressure
wf_friction	friction weighting factor for implicit pipe and channel components	wf_time	>= 0 and <= 1
wf_moc_friction	friction weighting factor for MOC pipe	0.5	>=0 and <= 1
wf_time	weighting factor in Preissmann's scheme for implicit pipes and channels	0.55	> 0.5 and <= 1
wf_volume	volume accumulation weighting factor for storage	0.5	> 0 and <= 1
xsect_file	name of file to which cross section tables are written	none	valid filename
xsect_warning	print warning when depth is out of range of cross section definition	on	on, off

time_step={0.1 (5,.2) (10,.5)}

This specifies an initial time step of 0.1 seconds (assuming *time_units* is set to seconds), which is changed to 0.2 seconds at simulation time 5 seconds and then to 0.5 seconds at simulation time 10 seconds. Another way to vary the time step is to change it at a simulation suspension (see the OPERATION section of this guide). However, if the simulated network contains *MOC_pipe* components, the time step may be specified only as a constant and cannot be changed at a simulation suspension.

Physical Constants

The keywords for specifying physical parameters include *barometric_pressure*, *gravity*, *kin_viscosity*, *unit_weight*, and *vapor_pressure*. The default values for kinematic viscosity, unit weight, and vapor pressure apply to water at atmospheric pressure and a temperature of 60 degrees F. Both the barometric pressure and vapor pressure are specified using absolute pressure units (either psia or kPaa).

Calculation Options

The following keywords may be classified as calculation options: *loss_re*, *pipe_re*, *wf_friction*, *wf_moc_friction*, *wf_time*, and *wf_volume*.

The keywords *loss_re* and *pipe_re* are used to define the transition Reynolds number for loss-type components (*pipe_loss*, *valve*, *check_valve*, *rev_tainter*, *converging_tee*, *diverging_tee*, *converging_manifold*, and *diverging_manifold*) and pipe and channel components (*moc_pipe*, *imp_pipe*, *open_channel*, and *river_channel*), respectively. For Reynolds numbers below the transition, laminar flow is assumed. For Reynolds numbers above the transition, turbulent flow is assumed. The default values are reasonable and it should rarely be necessary to specify values for these keywords.

The keywords *wf_friction*, *wf_moc_friction*, *wf_time*, and *wf_volume* are all weighting factors in the numerically integrated governing equations for various components. Except for *wf_time*, the default values normally should be used for these four keywords, which are provided primarily for flexibility and research purposes. The keyword *wf_time* specifies the weighting factor θ used in Preissmann's four-point implicit scheme for *imp_pipe*, *open_channel*, and *river_channel* components. For solution stability, θ must lie between 0.5 and 1.0. For best accuracy θ should be near 0.5 but values too close to 0.5 can result in solutions with undesirable numerical oscillations. Increasing θ above 0.5 increases numerical damping, which eliminates numerical oscillations at the expense of reduced accuracy. As a compromise, θ is usually specified in the range 0.55 to 0.7. The default value for *wf_time* is 0.55, but in some cases a value of 0.6 or higher may give more acceptable results or improve solution convergence.

Solution Convergence

The keywords *dh_max*, *dq_max*, *dx_max*, *iter_max*, *converge_freq*, and *converge_mult* all specify convergence criteria for the implicit, iterative solution technique used for all

components except the *moc_pipe*, for which an explicit solution technique is used. The implicit solution is obtained using the Newton-Raphson iterative technique (see, for example, Carnahan et al., 1969) on the governing nonlinear equations. This technique requires solution of a matrix equation for corrections to the unknown variables at each iteration. The double-sweep method (Cunge et al., 1980), which is a modified tridiagonal approach, is used to efficiently solve the matrix equation.

Solution tolerances for head and flow are specified using *dh_max*, and *dq_max*, respectively. Iterations are continued until the largest computed corrections for head and flow are less than the specified tolerances. The solution tolerance for the few other variables that are adjusted each iteration, which include the loss coefficients for tees and manifolds, is specified using *dx_max*. The tolerance specified by *dx_max* is in most cases interpreted as a fraction of the variable value. For example, if *dx_max* were 0.01, then convergence for a tee loss coefficient would be assumed when the correction divided by the loss coefficient is less than 0.01 (correction is less than 1 percent of the current value).

It is advisable to specify reasonable values for *dh_max*, *dq_max*, and *dx_max* rather than to accept the default values. The default values for these tolerances are conservatively low, which can result in excessive iterations and "Maximum convergence achieved" run-time messages. A reasonable value for *dh_max* is a small percentage, say 0.1 or 0.01 percent, of the largest available head difference in the network (for example, upstream head minus downstream head). A reasonable value for *dq_max* is a small percentage of the steady or maximum flow expected during the simulation. Because *dx_max* is usually interpreted as a fractional amount, a reasonable value for it is 0.001 or 0.0001, which is equivalent to 0.1 or 0.01 percent. However, the tolerance defined by *dx_max* is used only in networks containing tees or manifolds.

Solution convergence problems are indicated when the run-time messages "Maximum number of iterations reached without convergence" and "Maximum convergence achieved" are received. The first step in resolving these messages is to make sure that the specified solution tolerances (*dh_max*, *dq_max*, and *dx_max*) are reasonable. The second step is to set the *debug* keyword equal to 1, which causes the results of each iteration to be written to the monitor. If it appears that the solution is converging but needs more iterations, then specification of the keywords *iter_max*, *converge_freq*, or *converge_mult* should be considered. If the solution is diverging rather than converging, then specification of these keywords is unlikely to make a difference.

The maximum number of iterations for each time step is specified by *iter_max*. The default value should normally be acceptable. In most cases, three to twenty iterations are required to obtain a solution. However, additional iterations are sometimes required when boundary conditions change very suddenly (sharp transient) or when the convergence tolerances (specified by *dh_max*, *dq_max*, and *dx_max*) are set very tight. As already mentioned, in most cases, reducing convergence tolerances is a more effective strategy for eliminating convergence problems than is increasing *iter_max*.

It should rarely be necessary to specify *converge_freq* and *converge_mult*. These parameters define the criteria for deciding whether or not "maximum convergence" has been

achieved, which occurs when the solution has converged as far as possible but has not satisfied the specified convergence criteria (tolerances). Again, before specifying *converge_freq* and *converge_mult*, make sure that the specified convergence tolerances are reasonable.

Input and Output

The keywords for specifying input and output options include *debug*, *io_units*, *overwrite*, *report_line*, and *time_units*. The remaining keywords related to input and output are described in the following two sections.

The keyword *io_units* is used to select either English or SI global units. The units for time are separately specified using the *time_units* keyword.

The *overwrite* keyword is used to specify the manner in which new plot and cross section data are written to existing plot and cross section files. If *overwrite* is set to *on*, then the existing file is simply overwritten, which destroys the previous contents. If *overwrite* is set to *off*, then the new data is appended to the end of the existing file, which preserves the previous contents. If *overwrite* is set to *warn*, then the user is warned and given the opportunity to provide a different file name before the existing file is overwritten.

The *report_line* keyword specifies the maximum column width of the tabular, detailed output report. The default value of 80 columns is convenient for viewing and printing the report contents, but often requires abbreviation of node labels.

The *debug* keyword specifies the level of output printed to the monitor during each iteration. A nonzero value is specified when additional output may give clues as to why a solution is difficult or impossible to obtain. As mentioned above under "Solution Convergence," useful output is provided after each iteration when *debug* is set to 1. However, users of LOCKSIM should never need to specify values of *debug* greater than 1. These provide output useful only during code development and debugging of new features.

Output for Plots

The keywords *plot_field*, *plot_file*, *plot_line*, *plot_labels*, *pv_start_time*, *pv_period*, and *pv_end_time* specify plot file characteristics, which are discussed in detail in the *Plot_Variables section of this guide. The most important of these keywords is *plot_file* which is used to specify the name of the file to which time-varying data are saved. The keywords plot_field and plot_line control the precision of the saved data and limit the length of each data line in the plot file. The keyword *plot_labels* specifies the format used to label the data columns in the plot file. The keywords *pv_start_time*, *pv_period*, and *pv_end_time* specify default values for, respectively, the simulation time at which to start saving plot variables, the incremental simulation time between plot variable saves, and the simulation time at which to stop saving plot variables. These three parameters can be specified also for individual plot variables in the **PLOT_VARIABLES* section of the input file.

Cross Sections

The keywords *autofix_k's*, *autofix_xsects*, *depth_min*, *xsect_file*, and *xsect_warning* apply only to cross sections, which are defined in the **CROSS_SECTIONS* section of the input file for *open_channel* and *river_channel* components. These keywords are discussed further in the *Cross_Sections section of this guide.

The keyword *xsect_file* is used to specify the name of the file to which interpreted cross section information is written. The input for each cross section is echoed to this file followed by calculated properties based on the input. This information is most useful for *riverine* cross sections for which the calculated output is a table of top width, area, wetted perimeter, and conveyance for each increment of water surface elevation.

The keyword *autofix_k's*, in which "k" refers to conveyance, is a switch that applies only to *riverine* cross sections. If the conveyance function for a *riverine* cross section does not monotonically increase with increasing water surface elevation (if the conveyance gradient, dK/dd, is negative for some d), solution convergence problems can occur. When *autofix_k's* is set to *on*, any decrease in the conveyance function for a riverine cross section as water surface elevation increases is automatically fixed by interpolation between the previous conveyance value and the next value that is larger. The tabulated results printed into the cross section file defined by *xsect_file* will include the interpolated values. When *autofix_k's* is set to *off*, a warning message is printed to the monitor and to the error file (LOCKSIM.MSG). In this case, the problem can be ignored in the hope that the solution will converge anyway, or the cross section parameters can be manually modified to ensure increasing conveyance with increasing water surface elevation. Setting of *autofix_k's* to *on* is highly recommended. The automatic interpolation is a reasonable and highly effective method of removing negative conveyance gradients from the cross section property tables.

The *xsect_warning* keyword specifies whether or not to print a warning message to the monitor and error file when a converged solution has been obtained by extrapolating a *riverine* cross section's tabulated data or by setting a minimum depth in a dry cross section. The extrapolation, which is automatic, occurs when the depth in a *riverine* cross section exceeds the maximum depth for which data were prescribed.

The *depth_min* keyword specifies a minimum cross section depth. When the number of elevation increments in the property tables for *riverine* cross sections is explicitly specified, the first increment is set equal to *depth_min* or the elevation range divided by the number of increments, whichever is smaller. The value of *depth_min* is also used in an attempt to deal with dry cross sections that, at this time, is still experimental and so far largely unsuccessful. When the keyword *autofix_xsects* is set to *on*, dry beds are handled by setting a minimum depth equal to *depth_min*.

*Components

The **COMPONENTS* section of the input file is always the first section following the **CONSTANTS* section. Except for tees and manifolds, each component, or "link," is bounded upstream by one node and downstream by another. Labels assigned to all nodes permit each components to be identified by its bounding nodes. For example, a component located between nodes "usnode" and "dsnode" is referred to as component "usnode dsnode," where, for the purpose of defining the direction of positive flow through a component, "usnode" is understood to be the upstream node and "dsnode" is understood to be the downstream node. Except for tees and manifolds, the input format for a component is

usnode ds_node component_type keyword1=value1, keyword2=value2
keyword3=value3, keyword4=value4, keyword5=value5, ...

where keyword-value pairs are provided to describe the component properties. Valid keywords depend on the identifier *component_type*, which is the name of one of the available components listed below:

moc_pipe	closed conduit solved using method of characteristics
imp_pipe	closed conduit solved using Preissmann's implicit scheme
open_channel	prismatic open channel
river_channel	riverine channel
valve	valve with time-varying position
check_valve	nonreverse flow valve
pipe_loss	minor loss in closed conduit
converging_tee	tee with combining flow defined as positive
diverging_tee	tee with dividing flow defined as positive
converging_manifold	manifold with combining flow defined as positive
diverging_manifold	manifold with dividing flow defined as positive
rev_tainter	reverse tainter valve in navigation lock culvert
storage	liquid storage with free-surface

Because they have two flow paths that either combine or divide, tee and manifold components are bounded by three nodes. Converging tee and manifold components, for which combining flow is positive, are bounded upstream by two nodes and downstream by one. Diverging tee and manifold components, for which dividing flow is positive, are bounded upstream by one node and downstream by two. The input formats are described below, in the specific sections describing these components.

As already mentioned, type identifiers are case-independent and may be abbreviated. Components may be included in the input file in any desired order. The order in which the components are included in the input file determines the order in which they are arranged in the output tabular reports.

moc_pipe and *imp_pipe*

General Description

The *moc_pipe* and *imp_pipe* components model one-dimensional, unsteady flow in closed conduits. The *moc_pipe* uses the method of characteristics (Wylie and Streeter, 1993), which is an explicit numerical technique, while the *imp_pipe* uses a four-point, weighted implicit method, which is often referred to as Preissmann's scheme (Cunge et al., 1980). A LOCKSIM network may include both *moc_pipe* and *imp_pipe* components in any desired configuration. For a given closed conduit segment, the choice of *moc_pipe* or *imp_pipe* depends on the segment length, the acoustic wavespeed, and the size of the selected time step. Typically, the *moc_pipe* is most appropriate for longer closed conduits and simulations of rapid transient events requiring very small time steps to resolve the boundary conditions. Because it imposes no limitations on the time step size, the *imp_pipe*, on the other hand, is most appropriate for shorter closed conduits and simulations of gradually varying flow events for which larger time steps are sufficient to resolve the boundary conditions.

The *moc_pipe* has greater theoretical accuracy (zero numerical dispersion because interpolations are not supported) than the *imp_pipe* and its explicit solution scheme is more efficient. However, to ensure stability and accuracy, the time step size is limited by the Courant condition as described below. In practice, this usually means that either a very small time step must be specified or only the longest closed conduits in a network can be modeled using the *moc_pipe*. The *imp_pipe* is effected more by numerical dispersion than the *moc_pipe* but it is stable for all time steps and reach lengths. Consequently, relatively short closed conduits can be modeled and the time step can be selected to accurately resolve the boundary conditions rather than to satisfy a numerical stability constraint. Typically, if the time step is small enough to accurately resolve the boundary conditions, results obtained using the *imp_pipe* are nearly indistinguishable from results obtained using the *moc_pipe* and a significantly smaller time step.

Equations

The following continuity and momentum equations are solved to compute one-dimensional unsteady flow in uniform closed conduit components (Wylie and Streeter, 1993):

$$\frac{\partial p}{\partial t}+\frac{\rho a^2}{A}\frac{\partial Q}{\partial x}=0 \tag{5}$$

$$\frac{\partial Q}{\partial t}+\frac{A}{\rho}\frac{\partial p}{\partial x}+gA\frac{dz}{dx}+\frac{4A\tau_o}{\rho D_h}=0 \tag{6}$$

in which p = pressure, Q = discharge, t = time, x = longitudinal coordinate, a = acoustic wavespeed, g = acceleration of gravity, A = cross-sectional area, D_h = hydraulic diameter, ρ = fluid density, z = centerline elevation, and τ_o = wall shear stress. Hydraulic diameter, D_h, refers to the quantity (4A/P) in which P = wetted perimeter. Representation of the shear stress, τ_o, in Equation 6 depends on whether the conduit flow is turbulent or laminar. The flow is assumed turbulent when the Reynolds number, defined as $R_e = QD_h/A\nu$ in which ν = kinematic

viscosity, exceeds the laminar Reynolds number, which is defined in the **CONSTANTS* section of the input file using the keyword *pipe_re*. In turbulent flow, the shear stress is represented using either the Darcy-Weisbach friction factor, f, the Hazen-Williams coefficient, C, or the Manning coefficient, n:

$$\tau_o = \begin{cases} \dfrac{\rho f Q^2}{8A^2} & \cdots \text{Darcy Weisbach} \\ \dfrac{3\rho g Q^{1.85}}{4D^{1/6}(CA)^{1.85}} & \cdots \text{Hazen Williams} \\ \dfrac{\rho g n^2 Q^2}{1.39 D^{1/3} A^2} & \cdots \text{Manning} \end{cases} \tag{7}$$

The Darcy Weisbach f may either be constant or variable with Reynolds number, R_e. When it is specified as variable with Reynolds number, the following explicit approximation to the Colebrook equation is used (Zigrang and Slyvester, 1982):

$$\frac{1}{\sqrt{f}} \approx -2.0 \log\left[\frac{\varepsilon/D_h}{3.7} - \frac{5.02}{R_e}\log\left(\frac{\varepsilon/D_h}{3.7} + \frac{13}{R_e}\right)\right] \tag{8}$$

in which ε = wall roughness height. Over a broad range of roughness values and Reynolds numbers, values of the friction factor computed using Equation 8 differ by less than 1 percent from those obtained using Colebrook's equation. The expressions in Equation 7 for Hazen-Williams and Manning shear stress both assume *English* units. However, LOCKSIM makes the proper conversions when *SI* units are specified so that correct results are achieved. In laminar flow, the shear stress is represented by the following relationship:

$$\tau_o = \frac{8\rho\nu Q}{AD_h} \tag{9}$$

Numerical Solution

Equations 5 and 6 are numerically solved using the explicit, method of characteristics for the *moc_pipe* or the implicit, Preissmann's scheme for the *imp_pipe*. Figure 57 serves as a definition sketch. The length of a computational reach, labeled Δx in Figure 57, is assigned automatically in LOCKSIM to ensure numerical stability and accuracy. For the *moc_pipe* the reach length is chosen to exactly satisfy the Courant condition:

$$C_r = \frac{a\Delta t}{\Delta x} = 1 \tag{10}$$

in which C_r = Courant number and Δt = time step. The wavespeed, a, is adjusted if necessary to ensure that the total length of the *moc_pipe* component is divided into an integral number of reaches, each of length Δx. A warning message is printed if the required adjustment exceeds 0.5 percent of the specified wavespeed. To avoid significant adjustment in physical wavespeeds,

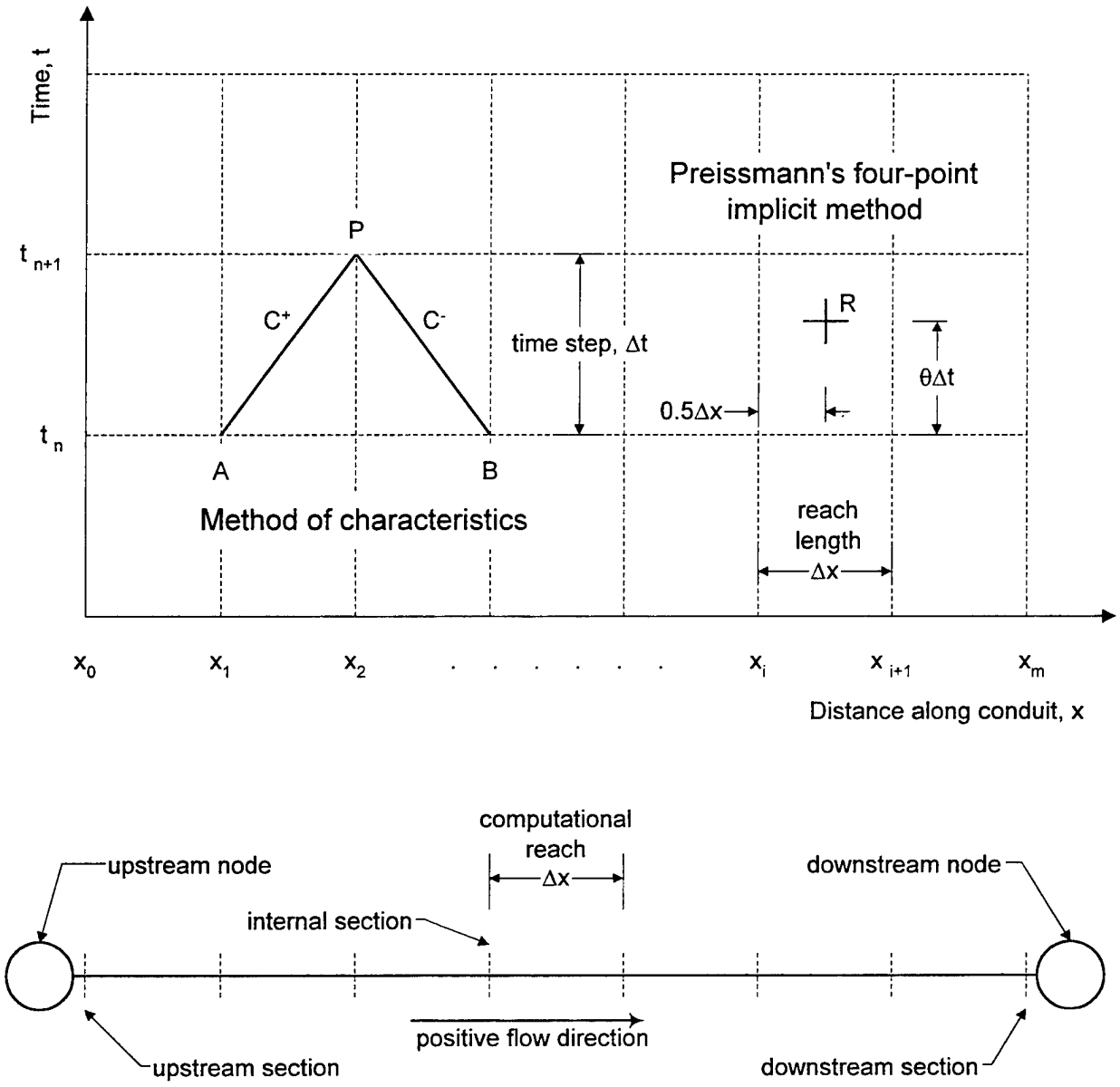


Figure 57: Definition sketch for closed conduit numerical solution techniques

the total length of each *moc_pipe* component and the time step size must be carefully specified. For the *imp_pipe*, which is stable for all values of C_r, a reach length is chosen that results in C_r greater than or equal to one but otherwise as close to one as possible, given the specified time step and wavespeed. Optionally for the *imp_pipe*, the user may directly specify the number of reaches rather than accept the automatic determination of reach length.

In the method of characteristics (for details see Wylie and Streeter, 1993), partial differential Equations 5 and 6 are transformed into four ordinary differential equations, two of which define C^+ and C^- characteristic lines as illustrated in Figure 57. The remaining two ordinary differential equations are integrated along the characteristic lines, one from point A to

point P in Figure 57 and the other from point B to point P, resulting in two equations in the two unknowns p_P, and Q_p at time t_{n+1}. The values of all variables are known for all grid points at time t_n. After solving for the pressure at point P, the integrated equations become:

$$C^+: \quad p_P = p_A - \rho g(z_P - z_A) - \frac{\rho a}{A}(Q_P - Q_A) - \frac{4}{D_h}\int_A^P \tau_o dx \tag{11}$$

$$C^-: \quad p_P = p_B + \rho g(z_B - z_P) + \frac{\rho a}{A}(Q_P - Q_B) + \frac{4}{D_h}\int_P^B \tau_o dx \tag{12}$$

which are solved simultaneously to determine p_P and Q_P. Using Equations 11 and 12 with points A, B, and P shifted along the conduit to adjacent grid points, the values for p and Q are evaluated explicitly for all grid points at time t_{n+1}, except those coinciding with the upstream and downstream boundaries. Each boundary has only one characteristic equation available, the C^- equation (point B at $x = x_1$) upstream and the C^+ equation (point A at $x = x_{m-1}$) downstream. Boundary conditions specifying p, Q, or a relationship between the two unknowns provide a second equation at each end, permitting the solution to be completed. At internal nodes in a network the boundary conditions are handled automatically.

For turbulent flow, the shear stress integral in Equation 11 may be approximated as

$$\int_A^P \tau_o dx = K\int_A^P Q|Q|^m dx \approx K\Delta x\left[2\theta_f Q_P|Q_A|^m + (1-2\theta_f)Q_A|Q_A|^m\right] \tag{13}$$

in which K and m are defined by Equation 7, depending on friction choice, and θ_f = friction weighting factor, which is assigned using the *wf_moc_friction* keyword in the **CONSTANTS* section of the input file. The trapezoidal integration used in Equation 13 is first order for $\theta_f = 0$ and second order for $\theta_f = 0.5$. The absolute value signs are required to give the friction term the proper sign when the discharge, Q, is negative. For turbulent flow, the shear stress integral in Equation 12 is similarly evaluated.

For laminar flow, the shear stress integrals in Equations 11 and 12 are evaluated by trapezoidal integration with θ_f equal to 0.5 (regardless of the specified value of *wf_moc_friction*).

In Preissmann's four-point implicit method (for details see Cunge et al., 1980), partial differential Equations 5 and 6 are transformed into algebraic, finite-difference equations applied to each reach along the conduit. The difference equations are weighted to approximate conditions at a point within each "cell" bounded by four surrounding grid points. Point R in Figure 57 is an example of such a point, surrounded by grid points (x_i, t_n), (x_{i+1}, t_n), (x_i, t_{n+1}), and (x_{i+1}, t_{n+1}). In Preissmann's scheme, point R is midway between x_i and x_{i+1}, and weighted between t_n and t_{n+1} depending on the value of θ. The set of difference equations for each cell, along with the upstream and downstream boundary conditions, must be solved simultaneously for the unknown pressures and discharges at time t_{n+1}.

The finite-difference forms of Equations 5 and 6 are written as follows:

$$\Delta x\left(\overline{\overline{Q_x}}+\frac{A}{\rho a^2}\overline{\overline{P_t}}\right)=0 \tag{14}$$

$$2\Delta t\left(\overline{\overline{Q_t}}+\frac{A}{\rho}\overline{\overline{P_x}}+gAz'+\frac{4A}{\rho D_h}\overline{\overline{\tau_o}}\right)=0 \tag{15}$$

in which the double-barred and primed quantities are finite difference approximations to the base quantities. With f representing any real variable, the double-barred and primed approximations to f and its derivatives are defined below:

$$f\approx\overline{\overline{f}}=\frac{1}{2}\left[\theta\left(f_{i+1}^{n+1}+f_i^{n+1}\right)+(1-\theta)\left(f_{i+1}^{n}+f_i^{n}\right)\right] \tag{16}$$

$$\frac{\partial f}{\partial t}\approx\overline{\overline{f_t}}=\frac{1}{2\Delta t}\left[\left(f_{i+1}^{n+1}-f_{i+1}^{n}\right)+\left(f_i^{n+1}-f_i^{n}\right)\right] \tag{17}$$

$$\frac{\partial f}{\partial x}\approx\overline{\overline{f_x}}=\frac{1}{\Delta x}\left[\theta\left(f_{i+1}^{n+1}-f_i^{n+1}\right)+(1-\theta)\left(f_{i+1}^{n}-f_i^{n}\right)\right] \tag{18}$$

$$\frac{df}{dx}\approx f'=\frac{1}{\Delta x}\left(f_{i+1}-f_i\right) \tag{19}$$

The subscripts i and i+1 indicate grid point position along the x-axis while superscripts n and n+1 indicate grid point position along the time axis. The weighting factor, θ, is assigned using the *wf_time* keyword in the **CONSTANTS* section of the input file. For solution stability, θ must lie between 0.5 and 1.0. For best accuracy θ should be near 0.5 but values too close to 0.5 can result in solutions with undesirable numerical oscillations. Increasing θ above 0.5 increases numerical damping, which eliminates numerical oscillations at the expense of reduced accuracy. As a compromise, θ is usually specified in the range 0.55 to 0.7. The default value for *wf_time* is 0.55, but in some cases a value of 0.6 or higher may give more acceptable results or improve solution convergence.

Input

The following keywords are available for describing closed conduit components (input units are given in parentheses; see Table 12):

Keyword	Symbol	Description
diameter	D_h	hydraulic diameter (length)
area	A	cross-sectional area (area)
length	---	total length between upstream and downstream nodes (length)

wavespeed	a	acoustic wavespeed (velocity)
DW_f	f	Darcy-Weisbach friction factor
HW_c	C	Hazen-Williams friction coefficient
MN_n	n	Manning resistance coefficient
roughness	ε	wall roughness height for computing f as function of R_e (length)
reaches	---	number of computational reaches for *imp_pipe*
SectPrintFreq	---	specifies computational sections included in reports (see text)

Figure 58 illustrates the keyword requirements. The keyword *DW_f* specifies a constant Darcy-Weisbach friction factor while the keyword *roughness* specifies Darcy-Weisbach friction that varies with Reynolds number according to Equation 8. If the *area* keyword is not provided, then the conduit is assumed to be circular. If the *SectPrintFreq* keyword is not provided, then output for every computational section is included in table reports. If *SectPrintFreq* is set equal to two, then output for every other section is included. If *SectPrintFreq* is set equal to three, then output for every third section is included, and so on.

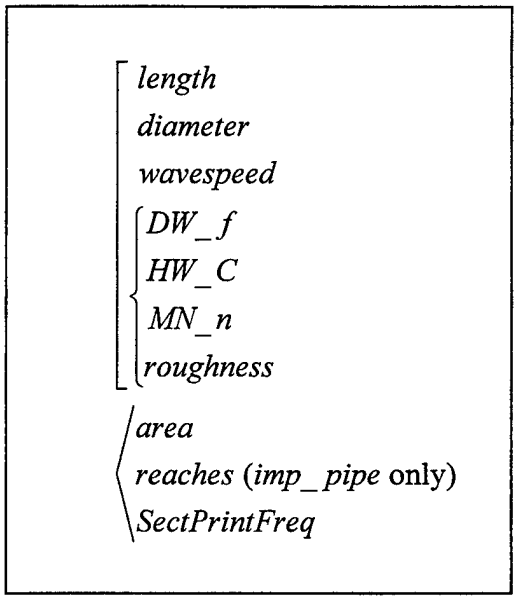


Figure 58: Keyword chart for *moc_pipe* and *imp_pipe* components

As described previously, the number of computational reaches into which a closed conduit component is divided is automatically determined by LOCKSIM. The *reaches* keyword may be provided for imp_pipes when it is desired to override the automatic determination. This keyword is invalid, however, for *moc_pipe* components because the computational reach length must exactly satisfy the Courant condition.

The centerline elevations of the upstream and downstream computational sections are set equal to the elevations of the upstream and downstream nodes, respectively. Linear interpolation is used to determine the centerline elevations of internal computational sections.

The following example input describes an *moc_pipe* between nodes "upstream" and "downstream" with friction computed using Equation 8:

upstream downstream moc_pipe dia=.5 length=1750 wavespeed=3500 rough=.0005

in which the keywords *diameter* and *roughness* have been abbreviated. For an assumed time step of 0.05 seconds, Equation 10 gives the reach length for this *moc_pipe* component as 175 feet, which divides it evenly into 10 reaches. If this component were defined as an *imp_pipe* rather than an *moc_pipe*, then its number of reaches could be specified, if desired. Otherwise, LOCKSIM would automatically assign 10 reaches in accordance with its efforts to maintain C_r as near to 1.0 as possible but not less than 1.0.

open_channel and *river_channel*

General Description

The *open_channel* and *river_channel* components model one-dimensional, subcritical, unsteady flow in free-surface channels. Both components use the four-point, weighted implicit method often referred to as Preissmann's scheme (Cunge et al., 1980). The *open_channel* component assumes a uniform cross section throughout its length, which makes it best for artificial channels of uniform cross section. The *river_channel* component permits different cross sections to be defined for each computational section, which makes it usually more appropriate for natural channels such as rivers and reservoirs. In both cases, the channel length is divided into computational reaches for the numerical solution. For the *open_channel* component, the reaches are all the same length. For the *river_channel* component, the reaches may each be a different length.

Channel cross sections, which are described in the **CROSS_SECTIONS* section of the input file, may be defined as either *trapezoidal*, *circular*, or *riverine*. The shape of a trapezoidal cross section is defined by specifying the bottom width and side slopes of a trapezoid (side slopes both equal to zero define a rectangular section). The shape of a *circular* cross section is defined by specifying a diameter. The shape of a *riverine* cross section is defined by specifying distance-elevation or elevation-width pairs. A riverine cross section, which can represent a section of arbitrary shape, may be laterally subdivided with different roughness coefficients applied to each subdivision. Automatic interpolation between given cross sections is available for *river_channels* when computational points are desired between known cross sections.

Lateral inflows may be specified as constants or as prescribed variations with time for both *open_channel* and *river_channel* components.

Equations

The following continuity and momentum equations are solved to compute one-dimensional unsteady flow in *open_channel* and *river_channel* components (e.g.; Cunge et al., 1980; Wylie and Streeter, 1993; Chow, 1959; Henderson, 1966):

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}-q=0 \tag{20}$$

$$\frac{\partial Q}{\partial t}+\frac{Q}{A}\left(2\beta\frac{\partial Q}{\partial x}+q\right)-\beta\frac{Q^2}{A^2}\frac{\partial A}{\partial x}+gA\frac{\partial H}{\partial x}+gAS_f=0 \tag{21}$$

in which H = water surface elevation or piezometric head defined by $H = p/\rho g+z$, Q = discharge, t = time, x = longitudinal coordinate, g = acceleration of gravity, A = cross-sectional area, β = momentum correction factor, q = lateral inflow per unit length, S_f = friction slope,

p = pressure, ρ = fluid density, and z = bed elevation. For solution in LOCKSIM, Equations 20 and 21 are rewritten in terms of pressure, discharge, and shear stress as follows:

$$\frac{\partial p}{\partial t}+\frac{\rho g}{T}\left(\frac{\partial Q}{\partial x}-q\right)=0 \tag{22}$$

$$\frac{\partial Q}{\partial t}+\frac{Q}{A}\left(2\beta\frac{\partial Q}{\partial x}+q\right)-\beta\frac{Q^2}{A^2}\frac{\partial A}{\partial x}+\frac{A}{\rho}\frac{\partial p}{\partial x}+gA\frac{dz}{dx}+\frac{4A\tau_o}{\rho D_h}=0 \tag{23}$$

in which T = width of cross section at the water surface, D_h = hydraulic diameter, and τ_o = bed shear stress. Hydraulic diameter, D_h, refers to the quantity (4A/P) in which P = wetted perimeter. Representation of the shear stress, τ_o, in Equation 23 depends on whether the channel flow is turbulent or laminar. The flow is assumed turbulent when the Reynolds number, defined as $R_e = QD_h/A\nu$ in which ν = kinematic viscosity, exceeds the laminar Reynolds number, which is defined in the **CONSTANTS* section of the input file using the keyword *pipe_re*. In turbulent channel flow, the shear stress is computed using Equation 7 with either the Darcy-Weisbach friction factor, f, or the Manning coefficient, n (Hazen-Williams friction is not used for free-surface flow). The Darcy Weisbach f may either be constant or variable with Reynolds number, R_e. When it is specified as variable with Reynolds number, it is computed using Equation 8. In laminar channel flow, shear stress is represented by the following relationship:

$$\tau_o=\frac{8\rho\nu Q}{AD_h} \tag{24}$$

which is equivalent to Equation 9.

Numerical Solution

Equations 22 and 23 are numerically solved using the Preissmann's four-point implicit, scheme, which was discussed above in the description of the *moc_pipe* and the *imp_pipe*. The comments there about the weighting factor, θ, also apply to the *open_channel* and *river_channel*. The finite-difference forms of Equations 22 and 23 are written as follows:

$$\Delta x\left(\overline{\overline{Q_x}}-\overline{q}+\frac{\overline{T}}{\rho g}\overline{p_t}\right)=0 \tag{25}$$

$$2\Delta t\left[\overline{\overline{Q_t}}+2\overline{\left(\frac{\beta Q}{A}\right)}\,\overline{\overline{Q_x}}+\frac{\overline{A}}{\rho}\overline{p_x}+g\overline{A}z'+\frac{4}{\rho}\overline{\left(\frac{A\tau_o}{D_h}\right)}+\overline{\beta}_x\overline{\left(\frac{Q^2}{A}\right)}-\overline{\beta}\,\overline{\left(\frac{Q}{A}\right)}^2\overline{A_x}+\overline{\left(\frac{Qq}{A}\right)}\right]=0 \tag{26}$$

in which Δt = time step, Δx = reach length (see Figure 57), and the double-barred and primed quantities are finite difference approximations to the base quantities as defined by Equations 16 through 19. The variation of cross section properties such as A, T, β, and D_h with x and water depth complicate the numerical solution for free-surface channels compared with that for closed conduits.

The number of computational reaches into which *open_channel* and *river_channel* components are divided is specified by the user. It is desirable for accuracy, although not necessary for stability, to specify enough reaches to ensure that the solution Courant number, C_r, defined by

$$C_r = \frac{\Delta t}{\Delta x}\sqrt{\frac{gA}{T}} \qquad (27)$$

remains near, or greater than, 1.0 during most of a simulation.

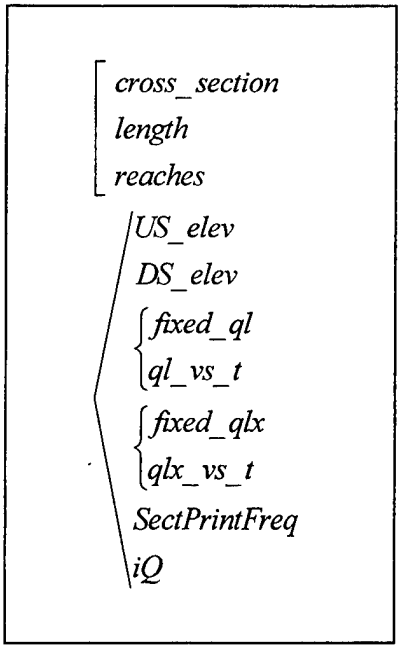


Figure 59: Keyword chart for *open_channel* component

Input for *open_channel*

The following keywords are available for describing *open_channel* components (input units are given in parentheses; see Table 12):

Keyword	Description
cross_section	label of cross section
length	total length of component (length)
reaches	number of computational reaches
US_elev	upstream bed elevation (length)
DS_elev	downstream bed elevation (length)
fixed_ql	fixed, or constant, total lateral inflow
ql_vs_t	label of function specifying total lateral inflow (discharge) vs. time (time)
fixed_qlx	fixed, or constant, lateral inflow per unit length of channel
qlx_vs_t	label of function specifying lateral inflow per unit length (discharge/length) vs. time (time)
SectPrintFreq	specifies computational sections included in reports (see text)
iQ	estimate of initial, steady-state discharge

Figure 59 illustrates the keyword requirements. All computational sections for an open_channel component have the same *cross_section* and all computational reaches have the same length. The cross section, along with its bed friction and momentum correction factor, is defined in the **CROSS_SECTIONS* section of the input file. The default values for *US_elev* and *DS_elev* are the adjacent node elevations (specified in the **NODES* section of the input file). Values for *US_elev* and *DS_elev* must be less than or equal to the adjacent node elevations. Linear interpolation between *US_elev* and *DS_elev* is used for the bed elevations of internal sections. Fixed or time-varying lateral inflow may be modeled by specifying either the total inflow over

the length of the component or the inflow per unit length. Lateral outflow is specified as negative lateral inflow. Functions describing lateral inflow variations with time are specified in the **FUNCTIONS* section of the input file. If the *SectPrintFreq* keyword is not provided, then output for every computational section is included in table reports. If *SectPrintFreq* is set equal to two, then output for every other section is included. If *SectPrintFreq* is set equal to three, then output for every third section is included, and so on. Specification of *iQ* is sometimes necessary to avoid an incorrect component discharge initial condition. See "Boundary and Initial Conditions" in the *Nodes section of this guide for additional information.

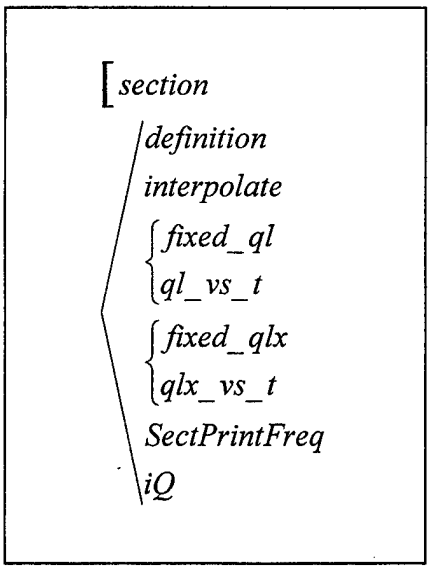


Figure 60: Keyword chart for *river_channel* component

The following example input describes an *open_channel* between nodes "upstream" and "downstream" with a fixed lateral inflow:

upstream downstream open_channel cross_section=square_one length=800, reaches=16
fixed_qlx=80

Input for *river_channel*

The following keywords are available for describing *river_channel* components (input units are given in parentheses; see Table 12):

Keyword	Description
section	data cluster with several keywords defining a computational cross section
definition	data cluster defining an intermediate cross section used for interpolating
interpolate	data cluster defining an interpolated cross section
fixed_ql	fixed, or constant, total lateral inflow
ql_vs_t	label of function specifying total lateral inflow (discharge) vs. time (time)
fixed_qlx	fixed, or constant, lateral inflow per unit length of channel
qlx_vs_t	label of function specifying lateral inflow per unit length (discharge/length) vs. time (time)
SectPrintFreq	specifies computational sections included in reports (see text)
iQ	estimate of initial, steady-state discharge

Figure 60 illustrates the keyword requirements. The data cluster *section* or the data cluster *interpolate* is specified for every computational section in the *river_channel*. Computational sections are defined in order from the first section upstream to the last section downstream. The first and last sections may not be interpolated sections. The data cluster *definition*, which is rarely used, is available to define an intermediate cross section used only for interpolating and not as a computational section. Fixed or time-varying lateral inflow may be modeled by specifying either the total inflow over the length of the component or the inflow per unit length. Lateral outflow is specified as negative lateral inflow. Functions describing lateral inflow variations with time are specified in the **FUNCTIONS* section of the input file. If the

SectPrintFreq keyword is not provided, then output for every computational section is included in table reports. If *SectPrintFreq* is set equal to two, then output for every other section is included. If *SectPrintFreq* is set equal to three, then output for every third section is included, and so on. Specification of *iQ* is sometimes necessary to avoid an incorrect component discharge initial condition. See "Boundary and Initial Conditions" in the *Nodes section of this guide for additional information.

The following keywords are available for use within the *section* and *definition* data clusters:

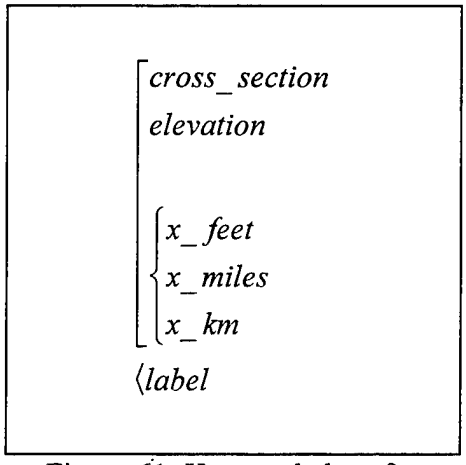


Figure 61: Keyword chart for *section* and *definition* data clusters

Keyword	Description
cross_section	label of cross section
elevation	bed elevation (length)
x_feet	longitudinal coordinate in feet
x_miles	longitudinal coordinate in feet
x_km	longitudinal coordinate in feet
label	label of this computational section in output

Figure 61 illustrates the keyword requirements for each *section* or *definition* data cluster. The number of computational reaches for a *river_channel* are equal to the number of defined computational sections minus one. Reach lengths are computed from the specified longitudinal coordinates for each section, which are defined in sequential order from upstream to downstream. The specified longitudinal coordinates must either monotonically increase or monotonically decrease from the upstream section to the downstream section. The cross section, along with its bed friction and momentum correction factor, is defined in the **CROSS_SECTIONS* section of the input file. The optional *label* is used in output reports and may be referenced in the **PLOT_VARIABLES* section of the input file. It is ignored when included in *definition* data clusters.

Available keywords for the *interpolate* data cluster are the same as those for the *section* and *definition* data clusters except for the keyword *cross_section*, which is not available. Figure 62 illustrates the keyword requirements. The interpolated cross section is automatically created from the properties of the nearest defined sections upstream and downstream. The linear interpolation is weighted according to the ratio of the distance to the upstream defined section to the distance between the upstream and downstream defined sections. If elevation is not specified, then it too is interpolated from the upstream and downstream defined sections.

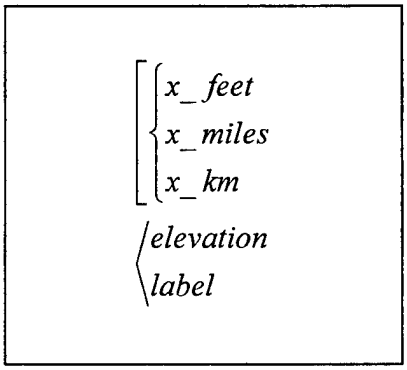


Figure 62: Keyword chart for *interpolate* data clusters

The following example input describes a *river_channel* between nodes "upstream" and "downstream" with both defined and interpolated computational sections:

upstream downstream river_channel iQ=14000
sect = {label=STA_14+10, x_feet=1410, cross_sect=XS1440, elev=270}
sect = {label=STA_14+40, x_feet=1440, cross_sect=XS1440, elev=270}
interp={label=sta_14+90, x_feet=1490}
sect = {label=STA_15+40, x_feet=1540, cross_sect=XS1540, elev=270}
sect = {label=STA_15+65, x_feet=1565, cross_sect=XS1565, elev=270}
interp={label=sta_16+19, x_feet=1619}
interp={label=sta_16+73, x_feet=1673}
sect = {label=STA_17+00, x_feet=1700, cross_sect=XS1700, elev=270}

For convenience and readability, keyword abbreviation has been liberally used to permit each computational section to be defined on one line. This *river_channel* component has seven reaches, most of different lengths. For each computational section, the keyword *label* refers to a physical river station. The cross section labels refer to the river stations at which cross section data are available. The computational section at river station 14+10 uses the cross section data for station 14+40 as an approximation because data are unavailable for station 14+10.

pipe_loss, valve, and *check_valve*

General Description

The *pipe_loss* component is used to represent various sources of geometry related energy loss in closed conduit networks. These sources of energy loss, also referred to as form loss or minor loss, include bends, expansions, contractions, inlets, exits, orifices, and many other pipe fittings and conduit obstructions. The *valve* and *check_valve* components are *pipe_loss* components with special features. The *valve* component represents a valve with specified, time-varying opening position. The *check_valve* component represents a simple nonreturn valve that remains fully open for positive flow and closes instantaneously to prevent negative flow.

Wyes and tees may be approximately represented by *pipe_loss* components but the *converging_tee*, and *diverging_tee* components provide more rigorous representations. Closed conduit friction is more properly modeled using the *moc_pipe* or *imp_pipe* component.

Equations

The *pipe_loss*, *valve*, and *check_valve* components satisfy the following energy equation:

$$\frac{Q^2}{2gA_u^2} + H_u = \frac{Q^2}{2gA_d^2} + H_d + h_l \tag{28}$$

in which Q = discharge, A_u = upstream flow area, H_u = upstream piezometric head defined by $H = p/\rho g + z$, A_d = downstream flow area, H_d = downstream piezometric head, h_l = energy loss, p = pressure, z = centerline elevation, ρ = fluid density, and g = acceleration of gravity. Both the energy loss and the change in velocity head that may occur across a *pipe_loss* , *valve*, or *check_valve* component are represented in Equation 28. The energy loss, h_l, is defined by

$$h_l = \begin{cases} K\dfrac{Q|Q|}{2gA^2} & \cdots \text{turbulent} \\ K\dfrac{\nu R_{lam}\sqrt{\pi}\,Q}{4gA^{3/2}} & \cdots \text{laminar} \end{cases} \tag{29}$$

in which K = a loss coefficient, A = flow area for which K applies, ν = kinematic viscosity, and R_{lam} = transition Reynolds number between laminar and turbulent flow, defined in the **CONSTANTS* section of the input file using the keyword *loss_re.* Flow is assumed turbulent when the Reynolds number defined by $QD/A\nu$ where $D = (4A/\pi)^{1/2}$ exceeds R_{lam}. For the *pipe_loss* and *check_valve* components, the loss coefficient, K, and the flow area, A, are both constants. For the *valve* component, K may be a function of A and A may vary with time in a user-specified manner. The absolute value sign in Equation 29 is required to give the turbulent energy loss term the proper sign when the discharge, Q, is negative. Of course, Q is always greater than or equal to zero for the *check_valve* component.

In Equation 29, the laminar expression is an approximation derived from the turbulent expression by analogy with the Colebrook equation for pipe friction. It is assumed that the loss coefficient in the turbulent expression varies with Reynolds number as illustrated in Figure 63, where K_{turb} refers to the loss coefficient used in the turbulent expression. The default value for R_{lam} in Figure 63 is 1000, based on a discussion and figure provided by Miller (1990). This transition value is lower than the transition value for pipe friction because of the abrupt changes in area and direction experienced by flows through pipe fittings and valves.

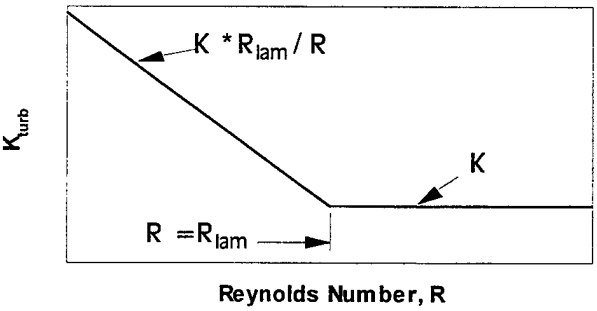


Figure 63: Assumed variation of turbulent K with Reynolds number

The laminar approximation in Equation 29 provides two significant benefits in LOCKSIM. First, it gives a more reasonable estimate of laminar form loss than would result from applying the turbulent expression for all Reynolds numbers. Second, it provides a *physically based* linearization of the energy loss term for small values of discharge, which is absolutely necessary to avoid solution convergence problems. Figure 64 illustrates the behavior of the turbulent energy loss term for discharge near zero. As the discharge goes to zero, the slope or derivative, dh_l/dQ, goes to zero, which causes the implicit solution procedure to diverge. By linearizing the h_l-Q relationship for discharge between plus and minus ε, where ε is some small value of discharge, a minimum slope greater than zero is established. However, it is difficult to determine a constant value of ε that is large enough to achieve solution convergence in all cases but small enough to minimize inaccuracy. The approach defined by Equation 29 neatly sidesteps this problem by indirectly specifying ε as a function of Reynolds number, which results in a physically reasonable and effective value of ε.

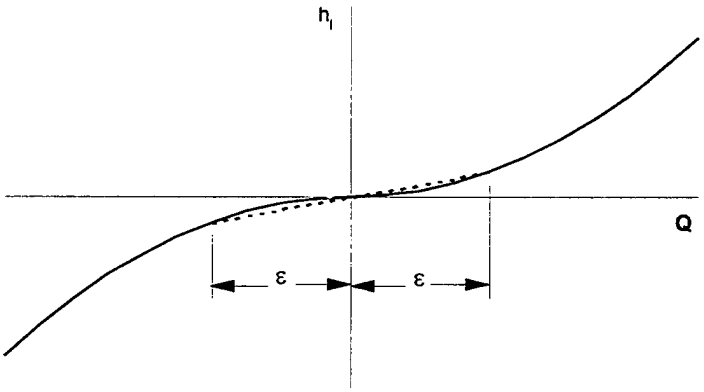


Figure 64: Energy loss for small discharge

Loss, Discharge, and Flow Coefficients

For the *pipe_loss*, *valve*, and *check_valve* components, the loss coefficient in Equation 29 can be specified directly as K or indirectly as the discharge coefficient C_d or the flow coefficient C_v, defined as follows:

$$K = \frac{1}{C_d^2} = g\left(\frac{964.4A}{C_v}\right)^2 \qquad (30)$$

The flow coefficient, C_v, is often used in industry to specify the discharge characteristics of valves. Its value specifies the discharge in gpm (gallons per minute) for a 1 psi pressure drop. The constant 964.4 in its definition results from conversion of gpm and psi units with the acceleration of gravity, g, in f/s^2 units.

In available references the loss coefficient, K, and the discharge coefficient, C_d, are not always defined the same as they are implemented in the *pipe_loss*, *valve*, and *check_valve* components. Before using these coefficients, it is important to compare their definitions with Equations 28 and 30.

Loss coefficients for closed conduit fittings, valves, and obstructions are available in most fluid mechanics textbooks (e.g., Streeter and Wylie, 1979) as well as many other references including Idelchik (1986), Miller (1990 and 1994), Crane (1969), Blevins (1984), and USACE (1988).

Input for *pipe_loss*

The following keywords are available for describing *pipe_loss* components (input units are given in parentheses; see Table 12):

Keyword	Description
area	A, the flow area for which K applies (area)
dia	diameter for computing A (length)
us_area	A_u, upstream flow area; defaults to A (area)
us_dia	upstream diameter for computing A_u (length)
ds_area	A_d, downstream flow area, defaults to A (area)
ds_dia	downstream diameter for computing A_d (length)
K+	K, loss coefficient for positive flow
Cd+	discharge coefficient for positive flow (see Equation 30)
Cv+	flow coefficient for positive flow (see Equation 30)
K-	K, loss coefficient for negative flow
Cd-	discharge coefficient for negative flow (see Equation 30)
Cv-	flow coefficient for negative flow (see Equation 30)

Figure 65 illustrates the keyword requirements. For convenience, different options are provided for specifying the cross-sectional areas and loss coefficients. Minimally, a cross-sectional area, either *area* or *dia*, and a loss coefficient, either *K+*, *Cd+*, or *Cv+*, define a *pipe_loss* component. If unspecified, the upstream area, A_u, and the downstream area, A_d, are assumed to be the same as A (no change in velocity head) and the loss coefficient for negative, or reverse, flow is assumed to be the same as for positive flow.

The following examples describe *pipe_loss* components between nodes "upstream" and "downstream." The loss coefficients are taken from Streeter and Wylie (1979). The first example is for a 90-degree standard elbow in a 6-inch pipe (English units):

upstream downstream pipe_loss dia=0.5 K+=0.9

There is no need to specify *us_dia* and *ds_dia* because they are the same and both are equal to *dia*. Also, there is no need to specify *K-* because it would typically be the same as *K+* for a pipe elbow. The next example describes an expansion from a 6-inch pipe to a 10-inch pipe:

upstream downstream pipe_loss us_dia=dia=0.5
ds_dia=0.833 K+=0.41 K-=0.283

When flow reverses, this fitting acts as a contraction with K=0.283. A well-rounded inlet from a reservoir to a 6-inch pipe could be represented as follows:

upstream downstream pipe_loss us_dia= 100
ds_dia=dia=0.5 K+=0.03 K-=1.0

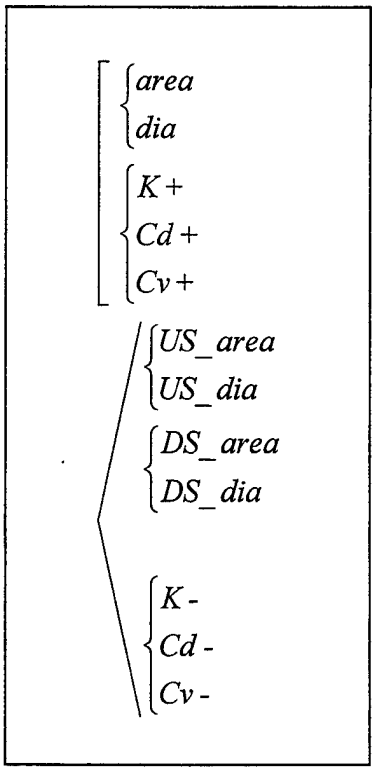


Figure 65: Keyword chart for *pipe_loss* component

The upstream diameter is assigned a large enough number to result in negligible upstream velocity head. Convergence problems may occur if the upstream flow area is more than 100 to 1000 times the downstream flow area. When the flow reverses, the inlet acts as an exit with K=1.0. Similarly, an exit from a 6-inch pipe to a reservoir could be represented as

upstream downstream pipe_loss us_dia= dia=0.5 ds_dia=100 K+=1.0 K-=0.5

where the exit is assumed to be square edged.

Input for *valve*

The following keywords are available for describing *valve* components (input units are given in parentheses; see Table 12):

Keyword	Description
area	A_{vo}, the flow area for the fully open valve (area)
dia	diameter for computing A_{vo} (length)
us_area	A_u, upstream flow area; defaults to A_{vo} (area)
us_dia	upstream diameter for computing A_u (length)
ds_area	A_d, downstream flow area, defaults to A_{vo} (area)
ds_dia	downstream diameter for computing A_d (length)
fixed_RVO	fixed valve position in terms of relative valve opening (RVO)
RVO_vs_t	label of function specifying RVO variation with time (time)

fixed_POS	fixed valve position in terms of arbitrary parameter POS
POS_vs_t	label of function specifying POS variation with time (time)
Ko+	K, loss coefficient for positive flow when valve is fully open
Cdo+	C_d for positive flow when valve is fully open (see Equation 30)
Cvo+	C_v for positive flow when valve is fully open(see Equation 30)
Ko-	K, loss coefficient for negative flow when valve is fully open
Cdo-	C_d for negative flow when valve is fully open (see Equation 30)
Cvo-	C_v for negative flow when valve is fully open (see Equation 30)
Kv+_vs_POS	label of function specifying variation of Kv+ with valve position
Cdv+_vs_POS	label of function specifying variation of Cdv+ with valve position
Cv+_vs_POS	label of function specifying variation of Cv+ with valve position
Kv-_vs_POS	label of function specifying variation of Kv- with valve position
Cdv-_vs_POS	label of function specifying variation of Cdv- with valve position
Cv-_vs_POS	label of function specifying variation of Cv- with valve position

Figure 66 illustrates the keyword requirements. For convenience, different options are provided for specifying cross-sectional areas, valve positions, and loss coefficients. If unspecified, the upstream area, A_u, and the downstream area, A_d, are assumed to be the same as the fully open valve area, A_{vo}, and the loss coefficients for negative, or reverse, flow are assumed to be the same as for positive flow. The time-varying or fixed valve opening and loss characteristics are input either by specifying "relative valve opening," RVO, along with the loss coefficient for the fully open valve, denoted by K_o, C_{do}, or C_{vo}, or by specifying "valve position," POS, along with a function describing the relationship between the loss coefficient denoted by K_v, C_{dv}, or C_v and valve position. The distinctions between RVO and POS and between the loss coefficients K_o and K_v and the discharge coefficients C_{do} and C_{dv} are discussed in the following paragraphs.

As illustrated in Figure 67 for a gate valve, valve flow characteristics are typically described by specification of one of the coefficients K_v, C_{dv}, or C_v as a function of opening position. These coefficients reflect the combined effect of varying flow area and loss coefficient with varying valve position. Consequently, energy loss is computed using Equations 29 and 30 with K and C_d replaced by K_v and C_{dv}, and A equal to the area of the fully open valve, even for partial valve

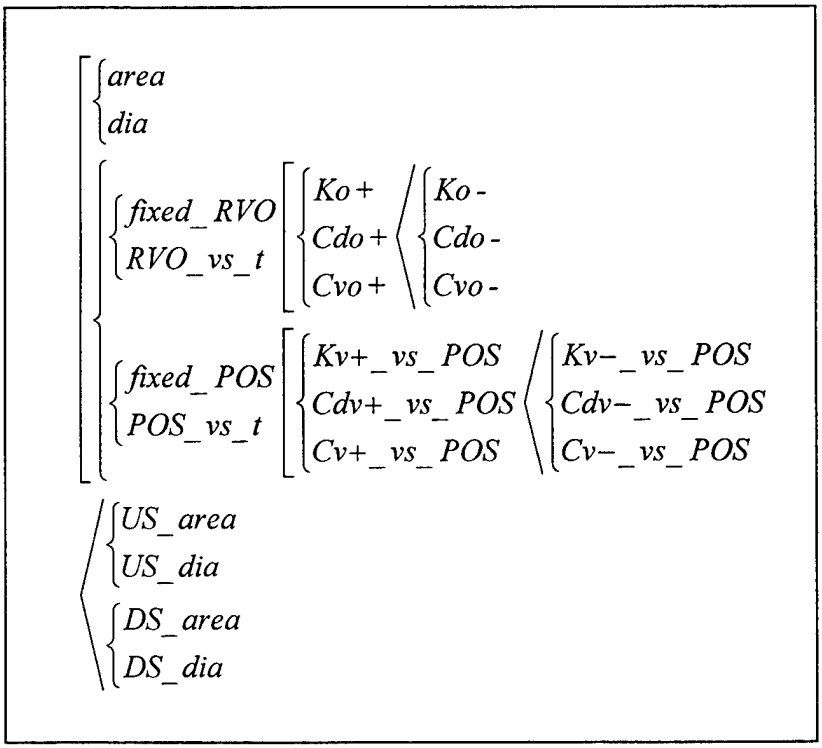


Figure 66: Keyword chart for *valve* component

openings. The coefficients K_o, C_{do}, and C_{vo}, on the other hand, refer to the particular values of K or K_v, C_d or C_{dv}, and C_v for the fully open valve. For example, C_{do} for the gate valve with the flow characteristics illustrated in Figure 67 is equal to 2.5.

Relative valve opening (RVO) combines the effects of time-varying valve flow area, $A_v(t)$ where t = time, and corresponding time-varying loss or discharge coefficient, $C_d[A_v(t)]$, in a single parameter defined as follows:

$$RVO(t) = \frac{C_d[A_v(t)]A_v(t)}{C_{do}A_{vo}} \tag{31}$$

RVO varies between 0, for a closed valve, and 1, for a fully open valve. LOCKSIM assumes the valve is closed whenever RVO is less than $1x10^{-6}$. This is the traditional method for specifying transient valve operation as described in references such as Wylie and Streeter (1993), for example. In those references, RVO is usually denoted by the symbol τ. Because only one value of the loss or discharge coefficient is required (K_o, C_{do}, or C_{vo}), this method is particularly convenient when it is desired to simply open or close a valve suddenly, or when the loss characteristics as a function of opening for a valve are unknown.

When the required data are available, transient valve operation is more conveniently specified using valve position (POS) as a function of time along with the loss coefficient (K_v, C_{dv}, or C_v) as a function of valve position. In this approach the valve motion is separated from the valve flow characteristics, which, as expressed by Equation 31, is not the case when RVO is specified. Various valve operation strategies may be simulated simply by modifying the function describing POS, rather than recomputing RVO for each one. The valve position, POS, is an arbitrary measure of valve opening which may be percent open (as in Figure 67), valve angle, area ratio, or any other convenient measure. However, a valve is assumed to be closed whenever POS is less than $1x10^{-6}$.

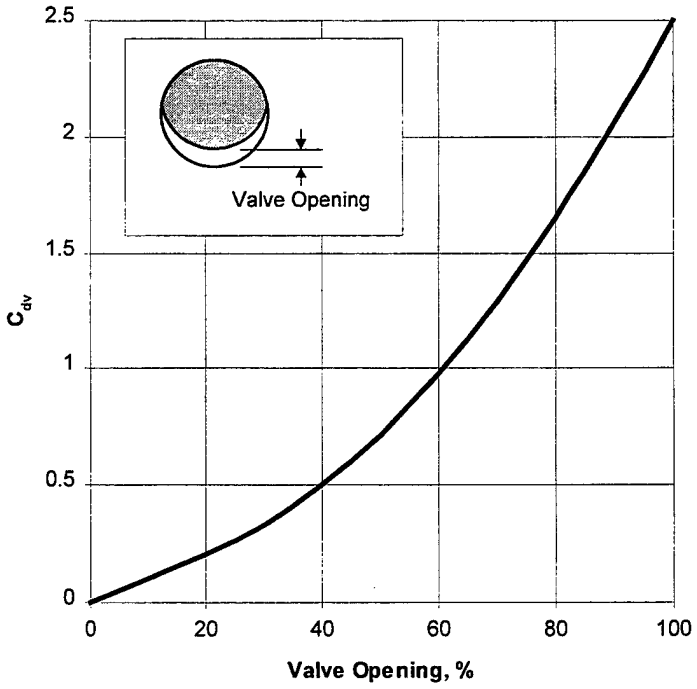


Figure 67: Discharge coefficient versus valve opening for a gate valve (data from USACE, 1988)

The following examples describe *valve* components between nodes "upstream" and "downstream." In the first example, a 6-inch gate valve with the flow characteristics illustrated in Figure 67 is closed in

0.05 seconds, starting at simulation time 10 seconds:

upstream downstream valve dia=0.5, Cdo+=2.5, RVO_vs_t=close_it_fast

and in the **FUNCTIONS* section of the input file:

close_it_fast discrete xshift=10, xscale=0.05 xy_pairs={0,1 1,0}

The relative valve opening varies linearly from 1 (open) to 0 (closed) as the simulation time increases from 10 and 10.05 seconds. Use of the keywords *xshift* and *xscale* in the function specification make it easy to change the time at which the valve starts to close and the time over which the closure takes place. In the second example, the 6-inch gate valve is initially closed and then gradually opened over a 20-second period, starting at simulation time 10 seconds:

upstream downstream valve dia=0.5
Cdv+_vs_POS=figure_12_data, POS_vs_t=gradual_opening

Two functions are specified in the **FUNCTIONS* section of the input file:

figure_12_data discrete interpolation=spline
x_values={0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100} ! POS in % open
y_values={0, .0985, .201, .328, .5, .718, .981, 1.29, 1.65, 2.07, 2.5} ! Cdv

gradual_opening discrete xshift=10, xscale=20, interpolation=spline
x_values={0, .1, .2, .3, .4, .5, 6, .7, .8, .9, 1} ! fractional time
y_values={0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100} ! POS in % open

As in the first example, the starting time and opening time for the valve can be adjusted by changing the values of *xshift* and *xscale*.

Input for *check_valve*

The following keywords are available for describing *check_valve* components (input units are given in parentheses; see Table 12):

Keyword	Description
area	A, the flow area for which K applies (area)
dia	diameter for computing A (length)
Ko+	K, loss coefficient for positive flow when valve is fully open
Cdo+	C_d for positive flow when valve is fully open (see Equation 30)
Cvo+	C_v for positive flow when valve is fully open(see Equation 30)

Figure 68 illustrates the keyword requirements. For convenience, different options are provided for specifying the cross-sectional area and loss coefficient. For the *check_valve* component, the upstream area, A_u, and the downstream area, A_d, in Equation 28 are assumed to be the same (no change in velocity head).

The following example describes a *check_valve* component between nodes "upstream" and "downstream:"

upstream downstream check_valve dia=0.5 Cdo+=1.4

$$\left[\begin{Bmatrix} area \\ dia \end{Bmatrix} \begin{Bmatrix} Ko+ \\ Cdo+ \\ Cvo+ \end{Bmatrix}\right]$$

Figure 68: Keyword chart for *check_valve* component

rev_tainter

General Description

The *rev_tainter* component is a specialized *valve* component used to represent reverse tainter valves in filling and emptying systems for navigation locks. Figure 69, depicts flow under a partially opened reverse tainter valve in a rectangular lock culvert, B units high by W units wide. The discharge accelerates as it passes under the valve until it attains a maximum velocity

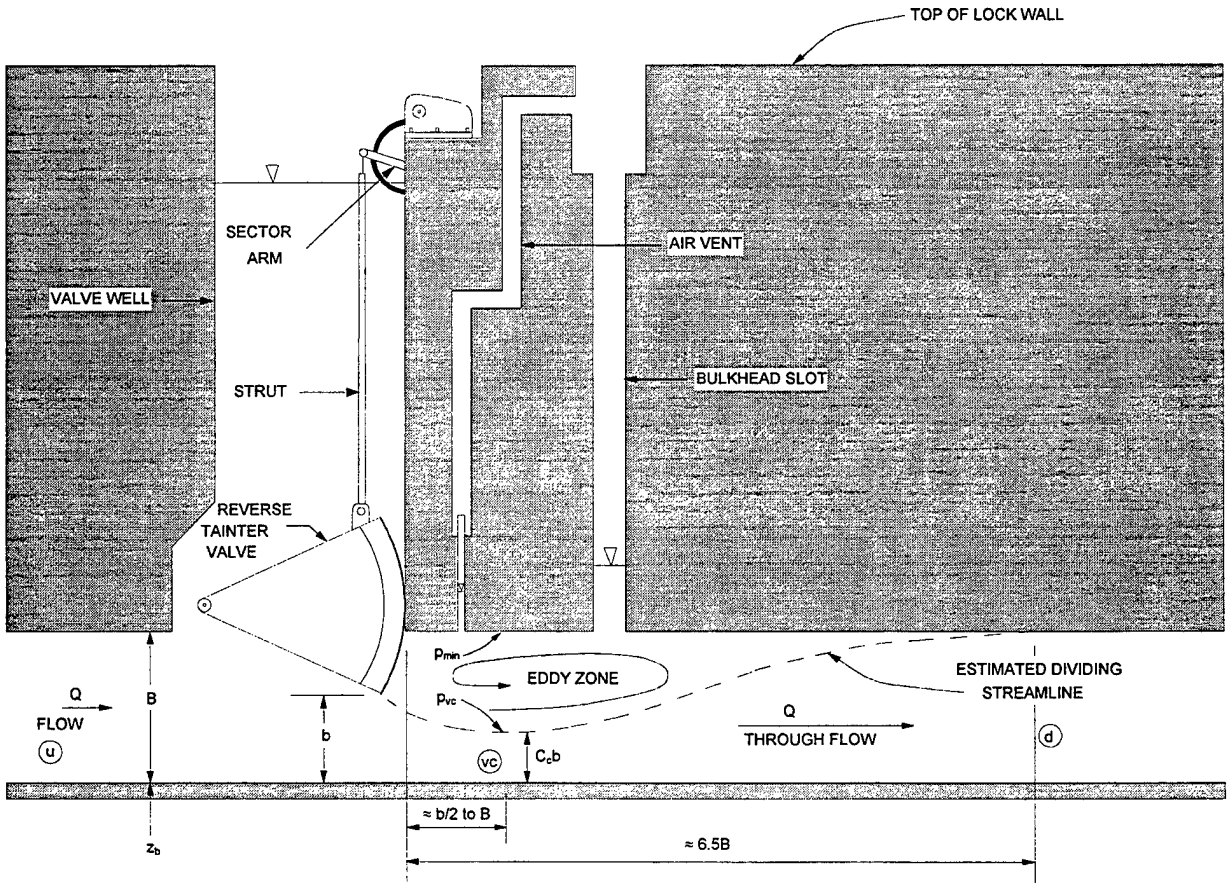


Figure 69: Flow under a partially open reverse tainter valve

and minimum cross-sectional area at the point labeled vc in Figure 69, which is the vena contracta of the jet downstream from the valve. This is also the point of minimum pressure and maximum cavitation potential downstream from the valve. The through-flow expands downstream from the vena contracta, gradually increasing in pressure and decreasing in velocity, until it fills the entire culvert cross section at point d, which is approximately 6.5 times the culvert height, B, downstream from the valve (USACE, 1975).

Like the *valve* component, the *rev_tainter* component has a specified time-varying position, b/B, and a specified position-dependent loss coefficient, K_v, C_{dv}, or C_v. Unlike the *valve* component, output for the *rev_tainter* component can include the cavitation index, piezometric head, and pressure at the vena contracta downstream from the valve. This additional output is available when the contraction coefficient, C_c, is specified as a function of relative valve opening position, b/B.

Equations

The *rev_tainter* component satisfies the following energy equation:

$$H_u = H_d + h_l \tag{32}$$

in which H_u and H_d = piezometric head at points u and d, respectively, in Figure 69 and h_l = energy loss. Piezometric head is defined by $H = p/\rho g + z$ where p = pressure, z = centerline elevation, g = acceleration of gravity, and ρ = fluid density. Equation 32 neglects changes in velocity head because the culvert in which the reverse tainter valve sits is assumed to be uniform between points u and d. The energy loss, h_l, is identical to the energy loss for the *pipe_loss*, *valve*, and *check_valve* components, defined by Equations 29 and 30. The piezometric head, H_{vc}, at the vena contracta is computed by assuming no head loss between points u and vc in Figure 69:

$$H_{vc} = H_u - \frac{V_{vc}^2}{2g}\left[1 - \left(\frac{C_c b}{B}\right)^2\right] \tag{33}$$

in which C_c = contraction coefficient, V_{vc} = velocity at the vena contracta ($Q/C_c bW$), B = culvert height, b = vertical opening under the valve, W = culvert width, and Q = discharge. The cavitation index, K_i, at the vena contracta is computed using V_{vc} and p_{vc} as reference values of velocity and pressure, where p_{vc} is the pressure on the top surface of the jet passing under the valve. The equation for K_i is

$$K_i = \frac{p_{vc} + p_{bar} - p_{vap}}{\rho g\, V_{vc}^2/2g} \tag{34}$$

in which p_{bar} = barometric pressure and p_{vap} = water vapor pressure, both of which are specified in the **CONSTANTS* section of the input file using the keywords *barometric_pressure* and *vapor_pressure*, respectively. The equation for p_{vc} is

$$p_{vc} = \rho g\left(H_{vc} - z_b - C_c b\right) \tag{35}$$

in which z_b = culvert bottom elevation. The minimum pressure, p_{min}, downstream from a reverse tainter valve is assumed to occur at the top of the culvert above the vena contracta. It is computed from H_{vc} as

$$p_{min} = \rho g\left(H_{vc} - z_b - B\right) \tag{36}$$

Contraction and Loss Coefficients

Data for C_c and K_v for reverse tainter valves in lock culverts are available in publications of the USACE Waterways Experiment Station. Figure 70 shows two representations of the function $C_c(b/B)$. The solid line plot, described by the polynomial

$$C_c = 0.948 - 1.396\frac{b}{B} + 2.98\left(\frac{b}{B}\right)^2 - 2.918\left(\frac{b}{B}\right)^3 + 1.385\left(\frac{b}{B}\right)^4 \tag{37}$$

is a curve-fit to prototype data reported by USACE (1979). The dashed line plot, described by the two equations

$$C_c = \begin{cases} 0.65 + 0.15\cos\left(5.236\frac{b}{B}\right) & \cdots \frac{b}{B} < 0.3 \\ 0.9 - 0.25\cos\left[2.244\left(\frac{b}{B} - 0.3\right)\right] & \cdots \frac{b}{B} \geq 0.3 \end{cases} \tag{38}$$

is the curve applied in H5320, symmetrical lock system numerical model (Hebler and Neilson, 1976). LOCKSIM function specifications for both Equations 37 and 38 are provided in the example input below.

Figure 71 shows a representation of the function $C_{dv}(b/B)$ based on K_v data provided by USACE (1988). The prototype data are sparse and scattered for b/B values less than 0.2 and greater than 0.9. In order to obtain reasonable values of the cavitation index for small valve openings, the curve in Figure 71 was extended to b/B equal to 0 by assuming that the entire velocity head at the vena contracta is

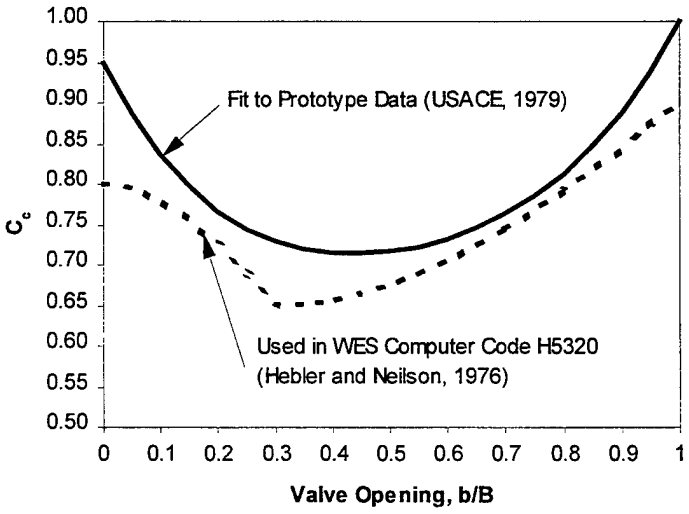


Figure 70: Variation of contraction coefficient with reverse tainter valve opening

lost when the flow expands to fill the culvert. The velocity head at the vena contracta was computed using C_c from Equation 37. The prototype data suggest that the value of K_v for a fully open (b/B equal to 1) reverse tainter valve is between 0.01 and 0.2. In Figure 71, K_v is assumed to be 0.1, which by Equation 30 results in C_{dv} equal to 3.16. The LOCKSIM function specification for the curve in Figure 71 is provided in the example input below.

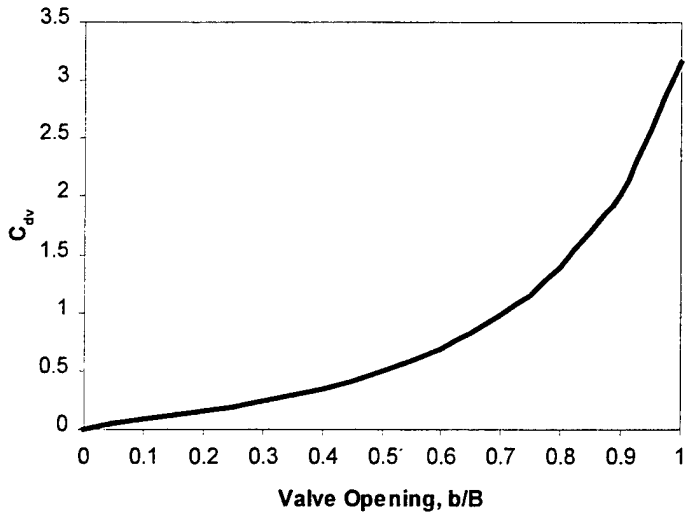


Figure 71: Variation of discharge coefficient with reverse tainter valve opening

Input

The following keywords are available for describing *rev_tainter* components (input units are given in parentheses; see Table 12):

Keyword	Description
B	height of rectangular culvert in which valve sits (length)
W	width of rectangular culvert in which valve sits (length)
fixed_b/B	fixed valve position in terms of fractional opening b/B
b/B_vs_t	label of function specifying b/B variation with time (time)
Kv+_vs_b/B	label of function specifying variation of Kv+ with b/B
Cdv+_vs_b/B	label of function specifying variation of Cdv+ with b/B
Cv+_vs_b/B	label of function specifying variation of Cv+ with b/B
EL_bottom	z_b, floor elevation of culvert, default = elevation of upstream node - .5*B
Cc_vs_b/B	label of function specifying variation of C_c with b/B

Figure 72 illustrates the keyword requirements. For convenience, different options are provided for specifying the loss coefficient. The loss coefficients for negative, or reverse, flow, which should never occur under a filling or emptying valve in a lock culvert, are assumed to be the same as for positive flow. As for the *valve* component, flow characteristics for a *rev_tainter* component are described by specification of one of the coefficients K_v, C_{dv}, or C_v as a function of opening position. These coefficients reflect the combined effect of varying flow area and loss coefficient with varying valve position. Consequently, energy loss is computed using Equations 29 and 30 with K and C_d replaced by K_v and C_{dv}, and A equal to the area of the fully open valve, BW, even for partial valve openings. Specification of the contraction coefficient, C_c, is required when output for the cavitation index, piezometric head, or pressure at the vena contracta downstream from the valve are desired. These output also require the bottom, or floor, elevation of the culvert, which defaults to the elevation of the upstream node minus one-half of the culvert height (node elevations are assumed to equal the centerline elevations of attached closed conduits and closed conduit fittings).

The following example describes a *rev_tainter* component between nodes "upstream" and "downstream" in a 15-foot high by 17-foot wide culvert (English units). The valve is opened in 7 minutes (420 seconds) according to a cubic opening profile.

upstream downstream rev_tainter B=15, W=17,
el_bottom=282, b/B_vs_t=cubic_fill,
Cdv+_vs_b/B=tainterCdv, Cc_vs_b/B=tainterCc

Three functions are specified in the **FUNCTIONS* section of the input file. One function specifies the valve opening sequence:

cubic_fill polynomial xscale=420, xshift=0, xmax=1
coefficients={0, .1, 1.45, -.55}

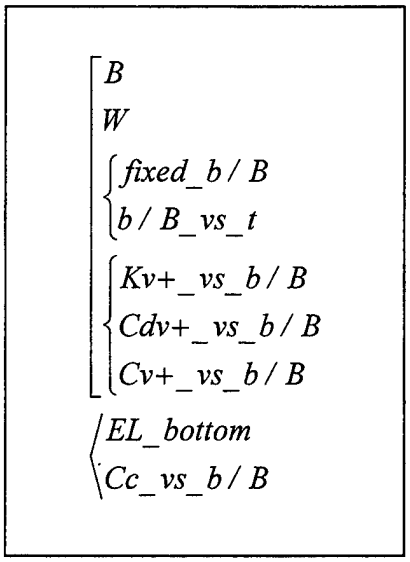


Figure 72: Keyword chart for *rev_tainter* component

Another function specifies C_c(b/B). Equation 37 would be specified as

tainterCc polynomial coefficients={.948, -1.396, 2.98, -2.918, 1.385 }

A composite function is necessary if Equation 38 is preferred:

tainterCc composite functions={Cc1 Cc2} xtransitions={.3}
Cc1 sinusoid yshift=.65 amplitude=.15 frequency=.8333 phase_shift=.25
Cc2 sinusoid yshift=.9 xshift=.3 amplitude=-.25 frequency=.3571 phase_shift=.25

A third function specifies C_{dv}(b/B). The curve shown in Figure 71 is specified by

tainterCdv discrete interpolation=spline
xy_pairs={

0, 0	.05, 0.0443	.10, 0.0836	.15, 0.1195	.20, 0.1534
.25, 0.196	.30, 0.243	.35, 0.295	.40, 0.349	.45, 0.415
.50, 0.494	.55, 0.587	.60, 0.69	.65, 0.8165	.70, 0.976
.75, 1.155	.80, 1.387	.85, 1.69	.90, 2.02	.95, 2.582
1.0, 3.162 }				

storage

General Description

The liquid *storage* component is used to represent simple surge tanks in closed conduit networks, embayments in free-surface networks, and other liquid volumes with variable free surfaces in which simulation of dynamic effects is not required.

Equations

Figure 73 serves as a definition sketch for the *storage* component. As liquid volume accumulates or depletes, depending on the balance between inflow and outflow, the liquid surface elevation, z_s, varies between a minimum value, equal to the larger of the upstream (u) and downstream (d) node elevations (the liquid level cannot fall below the elevations, z_u or z_d, of either of the connecting nodes), and a maximum value, z_{max}, specified by the user. Between the minimum and maximum values, the liquid surface elevation varies according to the following continuity equation:

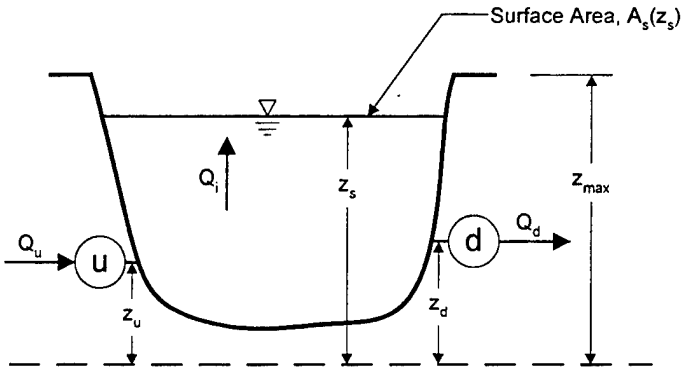


Figure 73: Definition sketch for *storage* component

$$A_s(z_s)\frac{dz_s}{dt} = Q_u - Q_d = Q_i \tag{39}$$

in which z_s = liquid surface elevation (equal to the piezometric head at each node, $H = p/\rho g + z$ where p = pressure, z = elevation, g = acceleration of gravity, and ρ = liquid density), t = time, Q_u = upstream discharge into component, Q_d = downstream discharge leaving component, and Q_i = net discharge into component. The liquid surface area, A_s, is defined either by a constant or by a prescribed variation with liquid surface elevation, z_s. When the liquid surface elevation rises to z_{max} it is fixed at that level and overflow discharge Q_o is computed as

$$Q_o = Q_u - Q_d = Q_i \tag{40}$$

The overflow discharge is treated as an external demand, meaning that the overflow volume is permanently removed from the network. When the liquid surface elevation falls to the larger of z_u or z_d it is fixed at that level (z_d in Figure 73) and an "underflow" volume, V_u, is accumulated as follows:

$$V_u = -\int Q_o dt \tag{41}$$

in which the overflow discharge, Q_o determined by Equation 40, is negative for underflow. The liquid surface elevation cannot rise above the minimum level again until the underflow volume is filled (V_u becomes zero) by sufficient positive inflow, Q_i (which by Equation 40 is equal to Q_o).

Numerical Solution

Equations 39 and 41 are integrated numerically using the trapezoidal rule. Equation 39 becomes

$$\int_{z_s^n}^{z_s^{n+1}} A_s(z)dz = \left[\theta_v Q_i^{n+1} + (1-\theta_v)Q_i^n\right](t^{n+1} - t^n) \qquad (42)$$

in which superscript n+1 refers to the new time step, superscript n refers to the past time step, and θ_v = volume weighting factor, which is assigned using the *wf_volume* keyword in the **CONSTANTS* section of the input file. The default value of θ_v is 0.5 and, as explained in the *Constants section of this guide, it should rarely be necessary to specify a different value. When A_s is constant, the integral on the left-hand side of Equation 42 reduces to $A_s(z_s^{n+1}-z_s^n)$. Otherwise, the function that describes $A_s(z_s)$ is integrated exactly. Trapezoidal integration of Equation 41 results in

$$V_u^{n+1} = V_u^n - \left[\theta_v Q_o^{n+1} + (1-\theta_v)Q_o^n\right](t^{n+1} - t^n) \qquad (43)$$

For solution in LOCKSIM, Equations 42 and 43 are rewritten in terms of pressure (or piezometric head) and the inflow and outflow discharges, Q_u and Q_d.

Input

The following keywords are available for describing *storage* components (input units are given in parentheses; see Table 12):

Keyword	Description
max_wsel	z_{max}, maximum elevation to which liquid surface can rise (length)
surface_area	A_s, liquid surface area when it is a constant (area)
area_vs_elev	label of function specifying variation of surface area with elevation

Figure 74 illustrates the keyword requirements. The maximum liquid surface elevation and either a constant or elevation-dependent liquid surface area must be specified. As already mentioned, the minimum liquid surface elevation is the larger of the upstream and downstream connecting node elevations.

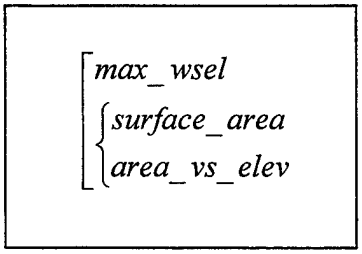


Figure 74: Keyword chart for *storage* component

The following example describes a *storage* component between nodes "upstream" and "downstream" representing a 10-foot diameter surge tank with a top elevation at 925 feet (English units):

upstream downstream storage max_wsel=925
surface_area=78.54

converging_tee and *diverging_tee*

General Description

The *converging_tee* and *diverging_tee* components simulate combining and dividing flow through tees, wyes, and other three-way junctions in closed conduit networks. Figure 75 illustrates the six possible configurations for flow through a tee. Configurations 1, 2, and 3 are converging, or combining, flows while configurations 4, 5, and 6 are diverging, or dividing, flows. Flows in all six configurations are correctly simulated by both the *converging_tee* and the *diverging_tee* components, which differ only in their assumed directions for positive flow, corresponding to configuration 1 for the *converging_tee* component and to configuration 4 for the *diverging_tee* component.

Equations

A tee has two separate flow paths, each governed by a quasi-energy equation, or "tee equation," with a coefficient that varies with the split in discharge and flow path geometry. In Figure 75, each tee leg is labeled as 1, 2, or 3 with leg 3 carrying the combined discharge. For converging flows, the flow paths are from leg 1 to leg 3 and from leg 2 to leg 3, with the following tee equation applied to each flow path:

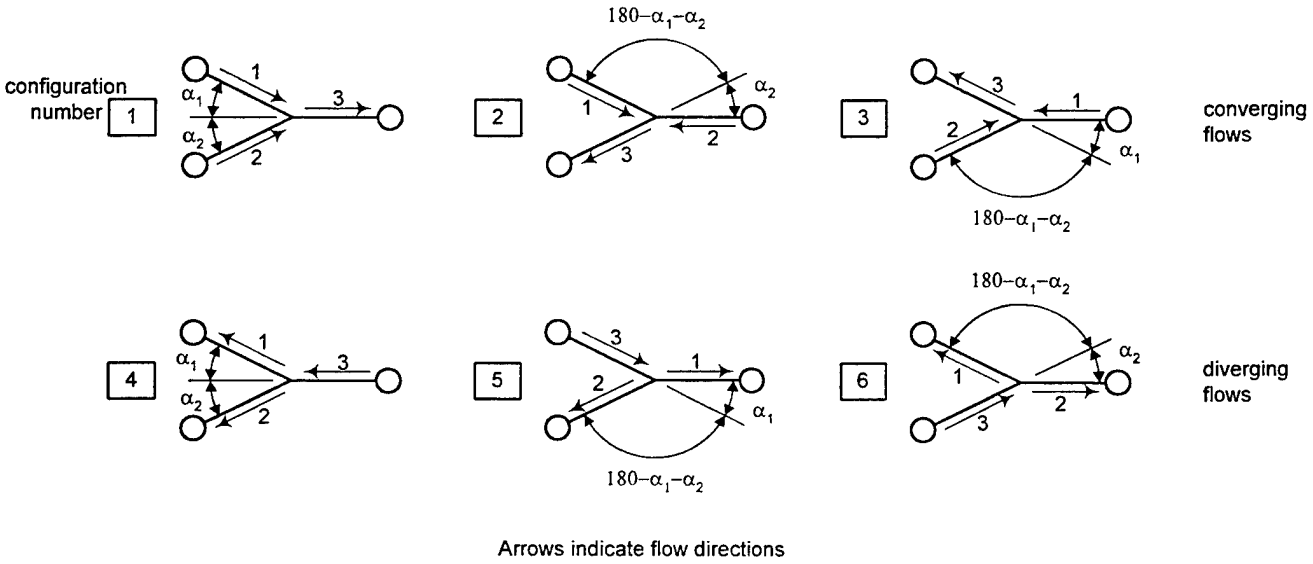


Figure 75: Six possible configurations for division of flow at a tee

$$\frac{Q_j^2}{2gA_j^2} + H_j = \frac{Q_3^2}{2gA_3^2} + H_3 + h_{cj3} \tag{44}$$

in which j = index of upstream leg (1 or 2), Q_j = discharge in leg j, A_j = flow area of leg j, H_j = piezometric head in leg j defined by $H = p/\rho g + z$, h_{cj3} = energy change for converging flow from leg j to leg 3, p = pressure, z = centerline elevation, ρ = fluid density, and g = acceleration of gravity. For diverging flows, the two flow paths are from leg 3 to leg 1 and from leg 3 to leg 2, with the following tee equation applied to each flow path:

$$\frac{Q_3^2}{2gA_3^2} + H_3 = \frac{Q_j^2}{2gA_j^2} + H_j + h_{d3j} \tag{45}$$

in which j = index of downstream leg (1 or 2) and h_{d3j} = energy change for diverging flow from leg 3 to leg j. The energy change terms in Equations 44 and 45 are each defined by a tee coefficient times the velocity head in leg 3, which always carries the combined flow:

$$h_{cj3} = \begin{cases} K_{cj3}\dfrac{Q_3^2}{2gA_3^2} & \cdots \text{turbulent} \\ K_{cj3}\dfrac{\nu R_{lam}\sqrt{\pi}\,Q_3}{4gA_3^{3/2}} & \cdots \text{laminar} \end{cases} \tag{46}$$

and

$$h_{d3j} = \begin{cases} K_{d3j}\dfrac{Q_3^2}{2gA_3^2} & \cdots \text{turbulent} \\ K_{d3j}\dfrac{\nu R_{lam}\sqrt{\pi}\,Q_3}{4gA_3^{3/2}} & \cdots \text{laminar} \end{cases} \tag{47}$$

in which K_{cj3} = tee coefficient for converging flow from leg j (1 or 2) to leg 3, K_{d3j} = tee coefficient for diverging flow from leg 3 to leg j, ν = kinematic viscosity, and R_{lam} = transition Reynolds number between laminar and turbulent flow, defined in the **CONSTANTS* section of the input file using the keyword *loss_re*. Flow is assumed turbulent when the Reynolds number defined by $Q_3D_3/A_3\nu$ where $D_3 = (4A_3/\pi)^{1/2}$ exceeds R_{lam}. The origin of the laminar terms in Equation 46 and Equation 47 are discussed above, in the section entitled "*pipe_loss, valve,* and *check_valve.*"

Tee Coefficients

The tee coefficients, K_{cj3} and K_{d3j}, vary with the amount of flow in each upstream leg compared with the combined flow in leg 3 and with the physical geometry of each individual flow path. The variance with flow is indicated by expressing the coefficients as functions of the appropriate flow ratios, or $K_{cj3}(Q_j/Q_3)$ and $K_{d3j}(Q_j/Q_3)$. For the general case, in which the flow

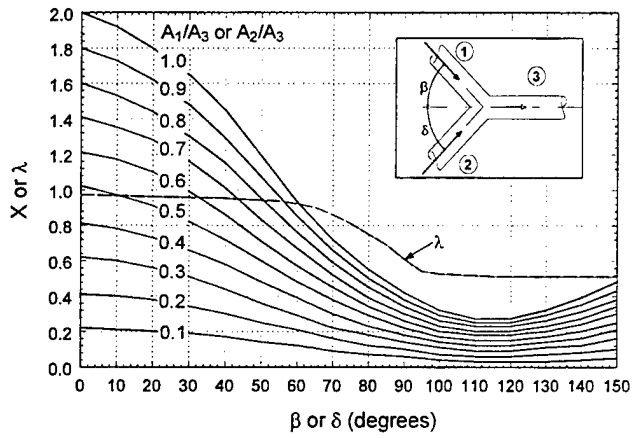


Figure 76: Parameters determining tee coefficients for converging flow (from Schohl et al., 1995)

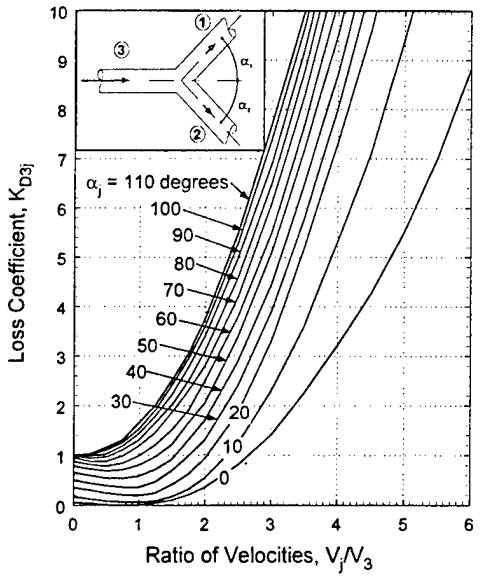


Figure 77: Tee coefficients for diverging flow (from Schohl et al., 1995)

areas of the tee legs and the angles between them are different, the coefficients $K_{cj3}(Q_j/Q_3)$ are different for each converging flow configuration and the coefficients $K_{d3j}(Q_j/Q_3)$ are different for each diverging flow situation. Figure 75 shows three converging flow configurations, each with two flow paths, and three diverging flow configurations, each with two paths. Consequently, six $K_{cj3}(Q_j/Q_3)$ functions and six $K_{d3j}(Q_j/Q_3)$ functions are required to completely define all possible flow paths through a given tee.

LOCKSIM includes internal, default functions for the tee coefficients that are available when the tee angles, denoted as α_1 and α_2 in Figure 75, are specified. Figure 76, for converging flow, and Figure 77, for diverging flow, illustrate the data defining the default tee coefficients. The loss coefficients K_{c13} and K_{c23} are computed as follows:

$$K_{c13} = 1 + [\lambda(\beta) - X(\beta, A_1/A_3)]\frac{V_1^2}{V_3^2} - X(\delta, A_2/A_3)\frac{V_2^2}{V_3^2}$$
$$K_{c23} = 1 + [\lambda(\delta) - X(\delta, A_2/A_3)]\frac{V_2^2}{V_3^2} - X(\beta, A_1/A_3)\frac{V_1^2}{V_3^2} \qquad (48)$$

in which V = velocity defined as Q/A, β and δ = angles between tee legs as illustrated in Figure 76, λ and X are parameters determined from Figure 76, and the numerical subscripts refer to the leg reference numbers in Figure 76. The loss coefficient K_{d3j} is determined directly from Figure 77. The default coefficients compare favorably with those for standard tees in circular piping given in other publications, including Miller (1990) and Idelchik (1986).

User options for the tee coefficients include using the default functions for all flow paths, some flow paths, or no flow paths. The defaults are used only for those individual tee coefficients that are not specified by the user.

Keywords

The following keywords are available for describing both *converging_tee* and *diverging_tee* components (input units are given in parentheses; see Table 12):

Keyword	Description
area1	A_1, the flow area for leg 1 (area)
dia1	diameter of leg 1 (length)
area2	A_2, the flow area for leg 2(area)
dia2	diameter of leg 2 (length)
area3	A_3, the flow area for leg 3 (area)
dia3	diameter of leg 3 (length)
angle1	α_1, angle between leg 3 and leg 1 (degrees)
angle2	α_2, angle between leg 3 and leg 2 (degrees)
fixed_Kd31	fixed value for K_{d31}
Kd31_vs_Qr	label of function describing $K_{d31}(Q_1/Q_3)$
fixed_Kd32	fixed value for K_{d32}
Kd32_vs_Qr	label of function describing $K_{d32}(Q_2/Q_3)$
fixed_Kd13	fixed value for K_{d13}
Kd13_vs_Qr	label of function describing $K_{d13}(Q_3/Q_1)$
fixed_Kd12	fixed value for K_{d12}
Kd12_vs_Qr	label of function describing $K_{d12}(Q_2/Q_1)$
fixed_Kd23	fixed value for K_{d23}
Kd23_vs_Qr	label of function describing $K_{d23}(Q_3/Q_2)$
fixed_Kd21	fixed value for K_{d21}
Kd21_vs_Qr	label of function describing $K_{d21}(Q_1/Q_2)$
fixed_Kc13	fixed value for K_{c13}
Kc13_vs_Qr	label of function describing $K_{c13}(Q_1/Q_3)$
fixed_Kc23	fixed value for K_{c23}
Kc23_vs_Qr	label of function describing $K_{c23}(Q_2/Q_3)$
fixed_Kc31	fixed value for K_{c31}
Kc31_vs_Qr	label of function describing $K_{c31}(Q_3/Q_1)$
fixed_Kc21	fixed value for K_{c21}
Kc21_vs_Qr	label of function describing $K_{c21}(Q_2/Q_1)$
fixed_Kc32	fixed value for K_{c32}
Kc32_vs_Qr	label of function describing $K_{c32}(Q_3/Q_2)$
fixed_Kc12	fixed value for K_{c12}
Kc12_vs_Qr	label of function describing $K_{c12}(Q_1/Q_2)$
iQ1	estimate of initial, steady-state discharge in leg 1
iQ2	estimate of initial, steady-state discharge in leg 2

The numerical subscripts indicate leg numbers as defined in the following sections describing specific input for the *converging_tee* and *diverging_tee* components. The leg numbers do not correspond to those in Figure 75 except when all leg discharges are positive.

Figure 78 illustrates the keyword requirements for *converging_tee* and *diverging_tee* components. The cross-sectional areas, or diameters, of each tee leg are required. Two options are provided for specifying the tee coefficients for combining and dividing flow. The first option is to specify the angles α_1 and α_2 between the tee legs, which makes the default tee coefficients available for all flow configurations. Any tee coefficients that are optionally specified are used instead of the default functions. The second option is to specify the tee coefficients for the primary flow directions, which, for *converging_tee* components, are those indicated as positive in Figure 79 and, for *diverging_tee* components, are those indicated as positive in Figure 81. The tee coefficients for other flow configurations default to zero if they are unspecified. It is recommended that *all* tee coefficients be specified unless it is certain that some flow configurations will not occur during the simulation. Specification of *iQ1* and *iQ2* is sometimes necessary to avoid an incorrect component discharge initial condition. See "Boundary and Initial Conditions" in the *Nodes section of this guide for additional information.

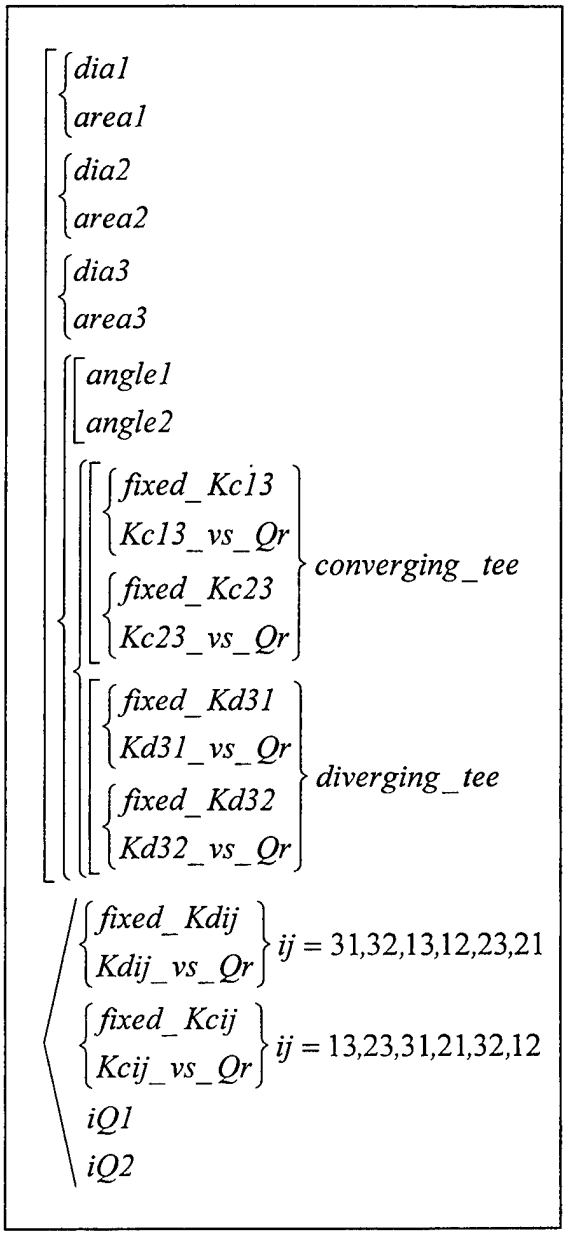


Figure 78: Keyword chart for *converging_tee* and *diverging_tee* components

Because tee components are bounded by three nodes, rather than two, their input differs from that of most components. The input formats for *converging_tee* and *diverging_tee* components are described in the following two sections.

Input for *converging_tee*

Figure 79 serves as a definition sketch for describing the input format for a *converging_tee* component. For input purposes, legs 1, 2, and 3 in Figure 79 are fixed regardless of the flow direction, meaning that the leg bounded upstream by node "U1" is leg 1 for all flow configurations, the leg bounded upstream by node "U2" is leg 2 for all flow configurations, and the leg bounded downstream by node "D" is leg 3 for all flow configurations. The discharges in each leg may be positive or negative. When all three are positive, as indicated in Figure 79, leg 3 carries the combined flow.

Referring to Figure 79, the input format for a *converging_tee* component is

U1 U2 D converging_tee keyword1=value1, keyword2=value2

keyword3=value3, keyword4=value4, keyword5=value5, ...

where "U1" and "U2" are the labels assigned to the upstream nodes and "D" is the label assigned to the downstream node. The order in which the upstream nodes are included determines which leg is number 1 and which is number 2, which is important to remember when specifying keywords. Keyword *area1*, for example, specifies the flow area for leg 1. Similarly, keyword *Kc13_vs_Qr* specifies a function describing the tee coefficient for converging flow from leg 1 to leg 3.

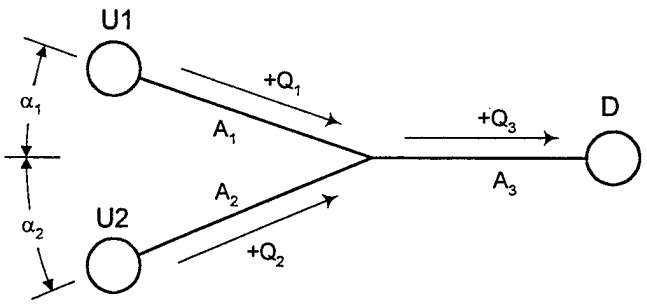


Figure 79: Definition sketch for *converging_tee* input

The following examples describe *converging_tee* components for the configurations depicted in Figure 80. The nodes in Figure 80 are labeled "A," "B," and "C." The numbers along each tee leg indicate diameter in inches (English units are assumed). For configuration 1, the following input relies on the internal, default tee coefficients:

A B C converging_tee dia1=.5, dia2=.75, dia3=1.0, angle1=25, angle2=45

Alternatively, the same tee could be specified as follows:

B A C converging_tee dia1=.75, dia2=.5, dia3=1.0, angle1=45, angle2=25

which simply assigns legs 1 and 2 differently. Similarly, configuration 2 could be specified as

A B C converging_tee dia1=dia2=.5, dia3=.75, angle1=0, angle2=90

and configuration 3 as

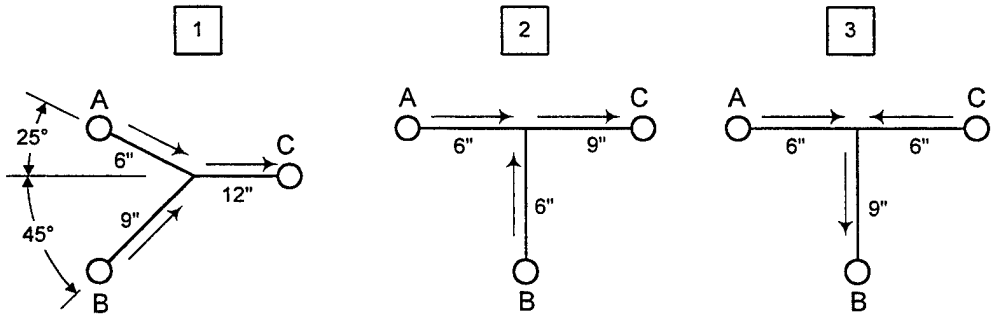


Figure 80: Configurations for *converging_tee* input examples

A C B converging_tee dia1=dia2=.5, dia3=.75, angle1=90, angle2=90

For configuration 2, the following input specifies tee coefficients for the primary flow paths and accepts the default value of zero for all other flow paths:

A B C converging_tee dia1=dia2=.5, dia3=.75
Kc13_vs_Qr=data_13, Kc23_vs_Qr=data_23

The functions "data_13" and "data_23" are specified in the **FUNCTIONS* section of the input file.

Input for *diverging_tee*

Figure 81 serves as a definition sketch for describing the input format for a *diverging_tee* component. For input purposes, legs 1, 2, and 3 in Figure 81 are fixed regardless of the flow direction, meaning that the leg bounded upstream by node "U" is leg 3 for all flow configurations, the leg bounded downstream by node "D1" is leg 1 for all flow configurations, and the leg bounded downstream by node "D2" is leg 2 for all flow configurations. The discharges in each leg may be positive or negative. When all three are positive, as indicated in Figure 81, leg 3 carries the combined flow.

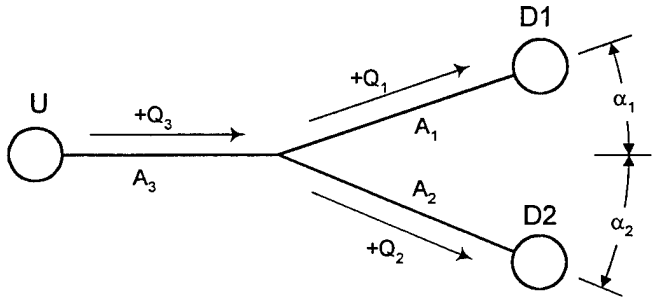


Figure 81: Definition sketch for *diverging_tee* input

Referring to Figure 81, the input format for a *diverging_tee* component is

U D1 D2 diverging_tee keyword1=value1, keyword2=value2
keyword3=value3, keyword4=value4, keyword5=value5, ...

where "U" is the label assigned to the upstream node and "D1" and "D2" are the labels assigned to the downstream nodes. The order in which the downstream nodes are included determines which leg is number 1 and which is number 2, which is important to remember when specifying keywords. Keyword *area1*, for example, specifies the flow area for leg 1. Similarly, keyword *Kd31_vs_Qr* specifies a function describing the tee coefficient for diverging flow from leg 3 to leg 1.

The following examples describe *diverging_tee* components for the configurations depicted in Figure 82. The nodes in Figure 82 are labeled "A," "B," and "C." The numbers along each tee leg indicate diameter in inches (English units are assumed). For configuration 1, the following input relies on the internal, default tee coefficients:

A B C diverging_tee dia1=.5, dia2=.75, dia3=1.0, angle1=25, angle2=45

Alternatively, the same tee could be specified as follows:

A C B diverging_tee dia1=.75, dia2=.5, dia3=1.0, angle1=45, angle2=25

which simply assigns legs 1 and 2 differently. Similarly, configuration 2 could be specified as

A B C diverging_tee dia1=dia2=.5, dia3=.75, angle1=0, angle2=90

and configuration 3 as

C A B diverging_tee dia1=dia2=.5, dia3=.75, angle1=90, angle2=90

For configuration 2, the following input specifies tee coefficients for the primary flow paths and accepts the default value of zero for all other flow paths:

A B C diverging_tee dia1=dia2=.5, dia3=.75
Kd31_vs_Qr=data_31, Kd32_vs_Qr=data_32

The functions "data_31" and "data_32" are specified in the **FUNCTIONS* section of the input file.

Relationships between tee coefficients when one leg has zero discharge

In principal, when the discharge in one tee leg is zero, the energy change for flow through the other two legs can be computed using either Equation 44 for converging flow or Equation 45 for diverging flow. However, in practice, these two equations give the same results only when the following equalities for tee coefficients, $K_{dij}(Q_j/Q_i)$ and $K_{cij}(Q_i/Q_j)$ are satisfied (all flow ratios are either zero or 1 when the discharge in one leg is zero):

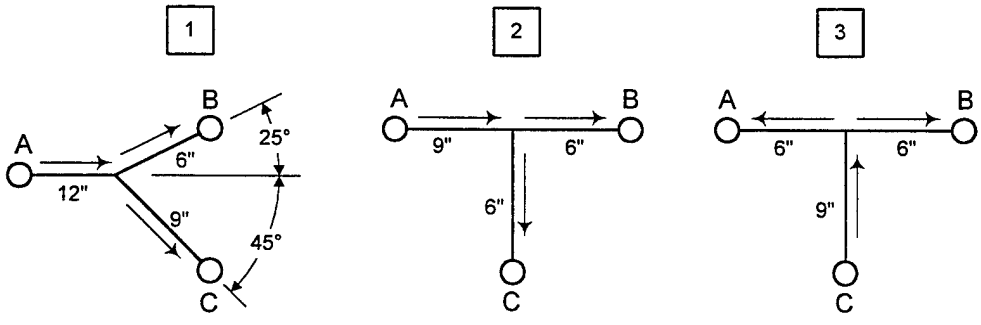


Figure 82: Configurations for *diverging_tee* input examples

$$K_{dij}(1) = K_{cij}(1)\left(\frac{A_i}{A_j}\right)^2 \quad \text{for ijk} = 312,132,321,231,123,213 \tag{49}$$

$$K_{dik}(0) = \left[K_{cij}(1) - K_{ckj}(0)\right]\left(\frac{A_i}{A_j}\right)^2 \quad \text{for ijk} = 312,132,321,231,123,213 \tag{50}$$

The subscripts i, j, and k refer to the leg numbers assigned in Figure 79 for the *converging_tee* component and in Figure 81 for the *diverging_tee* component. In Equations 49 and 50, leg k is the leg with zero discharge.

Equations 49 and 50 must be satisfied to ensure a continuous relationship between energy change and discharge for flows through tee legs. However, while it is good practice to define tee coefficients that satisfy Equations 49 and 50, it is not critical (for example, the internal, default tee coefficients do not satisfy Equations 49 and 50) because LOCKSIM makes automatic adjustments to force the tee coefficients to satisfy Equations 49 and 50 when the flow ratios are near zero and 1.

Equations 49 and 50 can be used to determine reasonable values for unknown tee coefficients when tee coefficients for the more important flow directions are known. For example, if $K_{d32}(0) = 0.85$, $K_{c21}(0) = -0.5$, and $A_1 = A_3$ then Equation 50 (with ijk = 312) specifies that $K_{c31}(1) = 0.35$. For lack of better information, this value of K_{c31} can be used for all flow ratios between zero and 1. Alternatively, if $K_{c31}(0)$ can also be determined from Equation 50, then $K_{c31}(Q_3/Q_1)$ can be assumed to vary linearly between $K_{c31}(0)$ and $K_{c31}(1)$.

converging_manifold and *diverging_manifold*

General Description

The *converging_manifold* and *diverging_manifold* components, which are special tee components, simulate combining and dividing flow through manifolds in closed conduit networks. Figure 83 serves as a definition sketch for discussing the attributes of manifolds as represented by the *converging_manifold* and *diverging_manifold* components. A manifold consists of a main conduit with multiple branches. The main conduit is assumed to be straight and to have constant flow area throughout its length. All of the branches are assumed to be identical, with the same flow area and the same angle of intersection with the main conduit (not necessarily 90 degrees as depicted in Figure 83). For the purpose of determining tee coefficients, each branch is assumed to pass the same flow, equal to the average branch flow, and to have the same piezometric head (head at point 2 in Figure 83). The tee coefficients for all flow paths through and past the branches are assumed to be the same for every branch. Their dependence on discharge ratio is determined using the average branch discharge and a suitable average of the main conduit discharge.

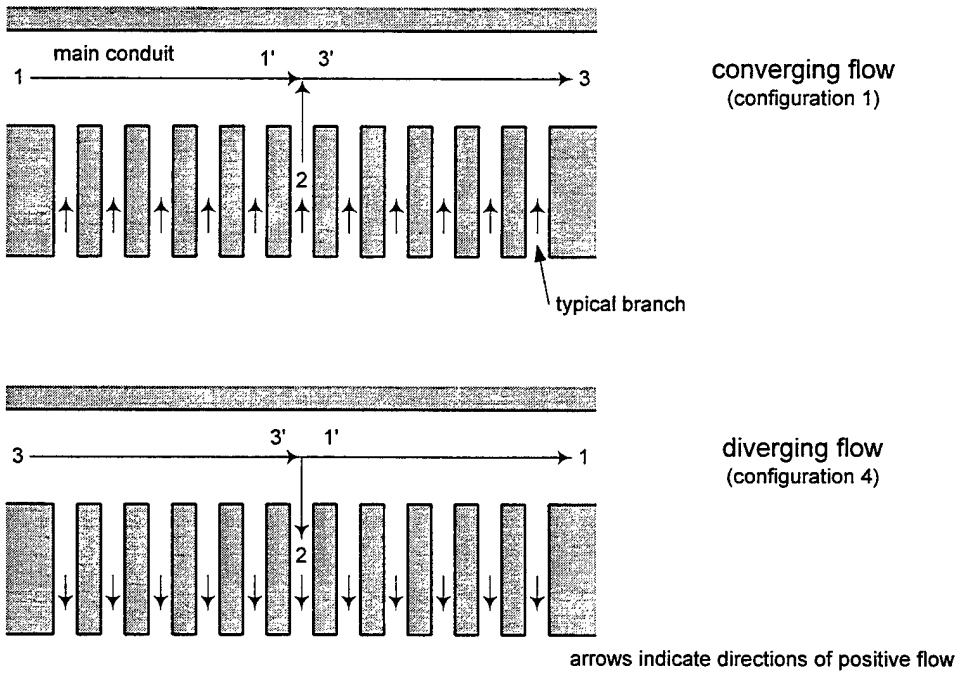


Figure 83: Definition sketch for *converging_manifold* and *diverging_manifold* components

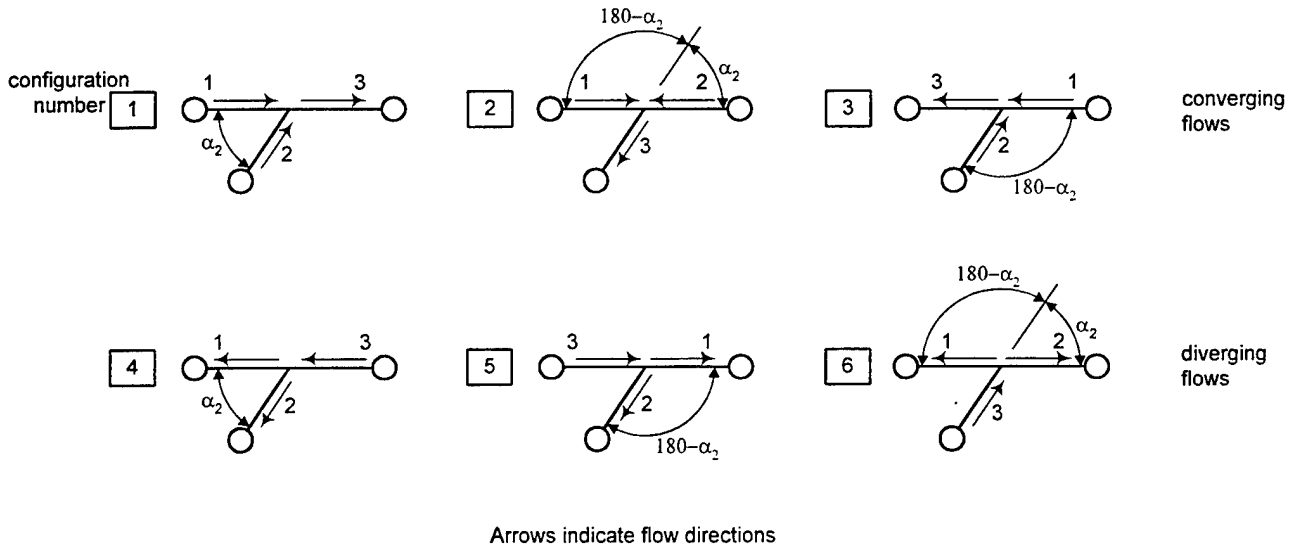


Figure 84: Six possible configurations for division of flow at a manifold

As for tees, flow through a manifold follows two separate paths in one of the six possible configurations illustrated in Figure 84. Configurations 1, 2, and 3 are converging, or combining, flows while configurations 4, 5, and 6 are diverging, or dividing, flows. For configurations 1, 3, 4, and 5, one flow path (2 to 3 for converging flow and 3 to 2 for diverging flow in Figure 83) is assumed to pass through the center-most manifold branch, while the other flow path (1 to 3 for converging flow, 3 to 1 for diverging flow) passes from one end of the main conduit to the other end. For configurations 2 and 6, both flow paths (1 to 3 and 2 to 3 for configuration 2, 3 to 1 and 3 to 2 for configuration 6) pass through a branch, but not the same branch. Figure 85 illustrates the assumed flow paths for configuration 2. Flow paths 1 to 3, for configuration 2, and 3 to 1, for configuration 6, are assumed to pass through the branch that is midway from point 1 to the zero discharge location in the main conduit. Similarly, flow paths 2 to 3, for configuration 2, and 3 to 2, for configuration 6, are assumed to pass through the branch that is midway from point 2 to the zero discharge location in the main conduit. The two points labeled 3 in Figure 85 are treated as the same point in LOCKSIM. By assumption they share the same branch discharge and piezometric head.

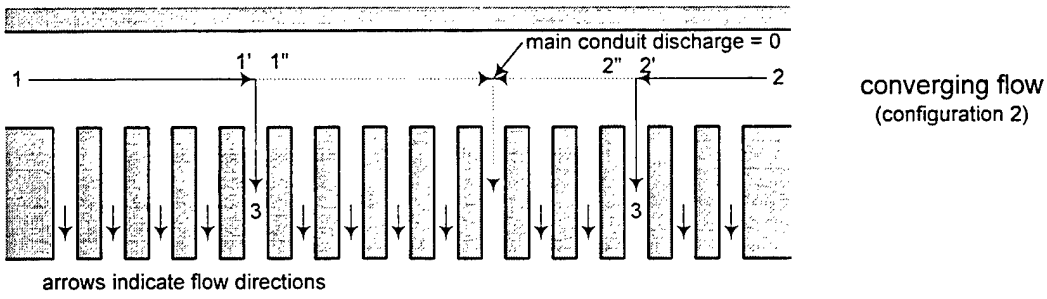


Figure 85: Assumed flow paths for configuration 2 manifold flow

Flows in all six configurations depicted in Figure 84 are correctly simulated by both the *converging_manifold* and the *diverging_manifold* components, which differ only in their assumed directions for positive flow, corresponding to configuration 1 for the *converging_manifold* component and to configuration 4 for the *diverging_manifold* component

The *converging_manifold* and *diverging_manifold* components do not simulate friction and momentum in the main conduit or branches. Like the tee and most other non-pipe, closed conduit components, changes in energy along the flow paths through a manifold are treated as if they occur at a *point* in the network, rather than as distributed over a finite length of conduit. Friction and momentum effects, which may be important in the main conduit, are simulated by attaching *imp_pipe* or *moc_pipe* components to the appropriate nodes of the manifold component.

Equations

The two separate flow paths through a manifold are each governed by a quasi-energy equation, or "tee equation," with tee coefficients that vary with the split in discharge and flow path geometry as described in the "*converging_tee and diverging_tee*" section of this guide. In Figure 83, Figure 84, and Figure 85, each manifold leg is labeled as 1, 2, or 3 with leg 3 carrying the combined discharge. Unlike tees, the discharge and piezometric head vary as flow passes each branch along the length of the two main conduit legs of a manifold. With reference to Figure 83, this means that the discharges and heads at points 1' and 3' are different from those at points 1 and 3, respectively. Consequently, the tee equation for flow between the branch (point 2) and main conduit (point 3) contains two separate energy change terms, one to express the energy change associated with combining or dividing flow between point 2 and point 3' and the other to express the energy change associated with combining or dividing flow in the main conduit between points 3' and 3. The two separate terms use different tee coefficients and different reference velocity heads. Similar considerations apply to the flow paths for all configurations.

For brevity, the expressions for energy change in laminar flow are not provided here. They are analogous to those listed for the *converging_tee* and *diverging_tee* components and, similarly, flow is assumed turbulent when the Reynolds number for the combined flow exceeds R_{lam}.

With reference to Figure 83 and Figure 84, the following tee equations apply for configurations 1 and 3:

$$\frac{Q_1^2}{2gA_1^2} + H_1 = \frac{Q_3^2}{2gA_3^2} + H_3 + h_{c13} \tag{51}$$

$$\frac{Q_2^2}{2gA_2^2} + H_2 = \frac{Q_3^2}{2gA_3^2} + H_3 + h_{c23'} + h_{c3'3} \tag{52}$$

in which the subscripts 1, 2, 3, and 3' refer to the labeled locations in Figure 83 and Figure 84; Q = discharge (Q_2 = total discharge from all branches); A = flow area (A_2 = total area of all branches); H = piezometric head defined by $H = p/\rho g + z$, where p = pressure, z = centerline elevation, ρ = fluid density, and g = acceleration of gravity; and h_{cij} = energy change associated with combining flow between points i and j. For turbulent flow, the energy change terms are each defined by a tee coefficient times the velocity head in leg 3 (at point 3' or 3), which carries the combined flow:

$$h_{c13} = nK_{c1'3'}(Q_{1'}/Q_{3'})\frac{Q_{3'}^2}{2gA_3^2} \tag{53}$$

$$h_{c23'} = K_{c23'}(q_2/Q_{3'})\frac{Q_{3'}^2}{2gA_3^2} \tag{54}$$

$$h_{c3'3} = \frac{n-1}{2}K_{c1'3'}(Q_{1'}/Q_{3'})\frac{Q_{3'}^2}{2gA_3^2} \tag{55}$$

in which n = number of branches, $K_{cij}(Q_i/Q_j)$ = tee coefficient as function of the appropriate flow ratio for converging flow from point i to point j and q_2 = discharge through a single branch. The discharges in Equations 53, 54, and 55 are defined as follows:

$$q_2 = \frac{Q_2}{n} \tag{56}$$

$$Q_{1'} = \frac{(n+1)Q_1 + (n-1)Q_3}{2n} \tag{57}$$

$$Q_{3'} = \frac{(n+1)Q_3 + (n-1)Q_1}{2n} \tag{58}$$

Equations 57 and 58 reduce to $Q_{1'} = Q_1$ and $Q_{3'} = Q_3$, respectively, when n = 1. Both discharges approach $(Q_1 + Q_3)/2$ as n increases to infinity. The continuity condition that $q_2 = Q_{1'} - Q_{3'}$ is also satisfied by Equations 56 through 58.

The following tee equations apply for configurations 4 and 5:

$$\frac{Q_3^2}{2gA_3^2} + H_3 = \frac{Q_1^2}{2gA_1^2} + H_1 + h_{d31} \tag{59}$$

$$\frac{Q_3^2}{2gA_3^2} + H_3 = \frac{Q_2^2}{2gA_2^2} + H_2 + h_{d33'} + h_{d3'2} \tag{60}$$

in which and h_{dij} = energy change associated with dividing flow between points i and j. For turbulent flow, the energy change terms are

$$h_{d31} = nK_{d3'1'}(Q_{1'}/Q_{3'})\frac{Q_{3'}^2}{2gA_3^2} \tag{61}$$

$$h_{d33'} = \frac{n-1}{2}K_{d3'1'}(Q_{1'}/Q_{3'})\frac{Q_{3'}^2}{2gA_3^2} \tag{62}$$

$$h_{d3'2} = K_{d3'2}(q_2/Q_{3'})\frac{Q_{3'}^2}{2gA_3^2} \tag{63}$$

in which $K_{dij}(Q_j/Q_i)$ = tee coefficient as function of the appropriate flow ratio for diverging flow from point i to point j. The discharges q_2, $Q_{1'}$, and $Q_{3'}$ are the same as those defined by Equations 56 through 58.

With reference to Figure 85, both tee equations for configuration 2 are defined by the following expression, the first with j = 1 and the second with j = 2:

$$\frac{Q_j^2}{2gA_j^2} + H_j = \frac{Q_3^2}{2gA_3^2} + H_3 + h_{djj'} + h_{dj'3} \tag{64}$$

in which $h_{djj'}$ = energy change associated with dividing flow between points j and j', $h_{dj'3}$ = energy change associated with dividing flow between points j' and 3, and A_3 = total flow area of all branches. Notice that although configuration 2 is a combining flow configuration, its energy change terms are all for dividing flow. For turbulent flow, the energy change terms, again for both j = 1 and j = 2, are

$$h_{djj'} = \frac{k_j - 1}{2}K_{dj'j''}(Q_{j''}/Q_{j'})\frac{Q_{j'}^2}{2gA_j^2} \tag{65}$$

$$h_{dj'3} = K_{dj'3}(q_3/Q_{j'})\frac{Q_{j'}^2}{2gA_j^2} \tag{66}$$

in which $K_{dj'j''}(Q_{j''}/Q_{j'})$ = tee coefficient for diverging flow from point j' to point j" as function of the appropriate flow ratio, $K_{dj'3}(q_3/Q_{j'})$ = tee coefficient for diverging flow from point j' to point 3 as function of the appropriate flow ratio and q_3 = discharge through a single branch defined by Equation 56 with subscript 2 replaced by subscript 3. The variable k_j references the zero discharge location in the main conduit in terms of the number of branches from the j^{th} end of the manifold. It is defined as follows:

$$k_j = \left| \frac{Q_j}{q_3} \right| \tag{67}$$

The branch numbers k_1 and k_2 are not necessarily integers, but their sum is n, the total number of branches. The discharges $Q_{j'}$, and $Q_{j''}$ are defined as follows:

$$Q_{j'} = \frac{(k_j + 1)Q_j}{2k_j} \tag{68}$$

$$Q_{j''} = \frac{(k_j - 1)Q_j}{2k_j} \tag{69}$$

which are similar to Equations 57 and 58 when the downstream discharge (Q_3 in those equations) is equal to zero.

One of the tee equations represented by Equation 64 changes if either k_1 or k_2 is less than 1.0. If k_j is less than 1.0, then the j^{th} tee equation is

$$\frac{Q_j^2}{2gA_j^2} + H_j = \frac{Q_3^2}{2gA_3^2} + H_3 + h_{cj3} \tag{70}$$

in which h_{cj3} = energy change associated with converging flow between point j and point 3:

$$h_{cj3} = K_{cj3}(Q_j / q_3)\frac{Q_3^2}{2gA_3^2} \tag{71}$$

Configuration 6 is similar, but opposite, to configuration 2. Its two tee equations are defined by the following expression with j = 1 and j = 2:

$$\frac{Q_j^2}{2gA_j^2} + H_j = \frac{Q_3^2}{2gA_3^2} + H_3 + h_{c3j'} + h_{cj'j} \tag{72}$$

in which $h_{c3j'}$ = energy change associated with converging flow between points 3 and j' and $h_{cj'j}$ = energy change associated with converging flow between points j' and j. Notice that although configuration 6 is a dividing flow configuration, its energy change terms are all for combining flow. For turbulent flow, the energy change terms, again for both j = 1 and j = 2, are

$$h_{c3j'} = K_{c3j'}(q_3 / Q_{j'})\frac{Q_{j'}^2}{2gA_j^2} \tag{73}$$

$$h_{cj'j} = \frac{k_j - 1}{2} K_{cj''j'}(Q_{j''} / Q_{j'}) \frac{Q_{j'}^2}{2gA_j^2} \tag{74}$$

in which $K_{c3j'}(q_3/Q_{j'})$ = tee coefficient for converging flow from point 3 to point j' as function of the appropriate flow ratio, $K_{cj''j'}(Q_{j''}/Q_{j'})$ = tee coefficient for converging flow from point j" to point j' as function of the appropriate flow ratio. As for configuration 2, the variable k_j references the zero discharge location in the main conduit and is defined by Equation 67. Likewise, the discharges $Q_{j'}$, and $Q_{j''}$ are defined by Equations 68 and 69. When k_j is less than 1.0, the j^{th} tee equation is

$$\frac{Q_j^2}{2gA_j^2} + H_j = \frac{Q_3^2}{2gA_3^2} + H_3 + h_{d3j} \tag{75}$$

in which h_{d3j} = energy change associated with diverging flow between point 3 and point j:

$$h_{d3j} = K_{d3j}(Q_j / q_3) \frac{Q_3^2}{2gA_3^2} \tag{76}$$

Tee Coefficients

The discussion of tee coefficients provided for the *converging_*tee and *diverging_tee* components applies also to the tee coefficients for the *converging_manifold* and *diverging_manifold* components. User options for the tee coefficients include using the default functions for all flow paths, some flow paths, or no flow paths. The defaults are used only for those individual tee coefficients that are not specified by the user. Because the angle between the main conduit legs of a manifold is assumed to be always zero, only the angle between the branches and the main conduit must be specified to use the default functions.

The combining and dividing energy changes for flow paths through a manifold branch are always defined in terms of the area and discharge for a *single* branch, not for the sum of all the branches. Several of the equations above reflect this fact, showing the dependence of tee coefficients on the ratio (or its reciprocal) of the main conduit discharge to the discharge in a single branch. Also, in Equation 48 the area of a single branch is used to compute the default tee coefficients for flow paths through manifold branches.

Keywords

The following keywords are available for describing both *converging_manifold* and *diverging_manifold* components (input units are given in parentheses; see Table 12):

Keyword	Description
area1	A_1, the flow area of the main conduit (area)
dia1	diameter of the main conduit (length)
area2	a_2, the flow area for one of the manifold branches (area)

dia2	diameter of one of the manifold branches (length)
branches	the number (integer) of manifold branches, each of area *area2*
angle2	α_2, angle between the main conduit and each branch (degrees)
fixed_Kd31	fixed value for K_{d31}
Kd31_vs_Qr	label of function describing $K_{d31}(Q_1/Q_3)$
fixed_Kd32	fixed value for K_{d32}
Kd32_vs_Qr	label of function describing $K_{d32}(q_2/Q_3)$
fixed_Kd13	fixed value for K_{d13}
Kd13_vs_Qr	label of function describing $K_{d13}(Q_3/Q_1)$
fixed_Kd12	fixed value for K_{d12}
Kd12_vs_Qr	label of function describing $K_{d12}(q_2/Q_1)$
fixed_Kd23	fixed value for K_{d23}
Kd23_vs_Qr	label of function describing $K_{d23}(Q_3/q_2)$
fixed_Kd21	fixed value for K_{d21}
Kd21_vs_Qr	label of function describing $K_{d21}(Q_1/q_2)$
fixed_Kc13	fixed value for K_{c13}
Kc13_vs_Qr	label of function describing $K_{c13}(Q_1/Q_3)$
fixed_Kc23	fixed value for K_{c23}
Kc23_vs_Qr	label of function describing $K_{c23}(q_2/Q_3)$
fixed_Kc31	fixed value for K_{c31}
Kc31_vs_Qr	label of function describing $K_{c31}(Q_3/Q_1)$
fixed_Kc21	fixed value for K_{c21}
Kc21_vs_Qr	label of function describing $K_{c21}(q_2/Q_1)$
fixed_Kc32	fixed value for K_{c32}
Kc32_vs_Qr	label of function describing $K_{c32}(Q_3/q_2)$
fixed_Kc12	fixed value for K_{c12}
Kc12_vs_Qr	label of function describing $K_{c12}(Q_1/q_2)$
iQ1	estimate of initial, steady-state discharge in main conduit leg 1
iQ2	estimate of total initial, steady-state discharge in manifold branches

The numerical subscripts indicate leg numbers as defined in the following sections describing specific input for the *converging_manifold* and *diverging_manifold* components. The leg numbers do not correspond to those in Figure 84 except when all leg discharges are positive.

Figure 86 illustrates the keyword requirements for *converging_manifold* and *diverging_manifold* components. The cross-sectional area, or diameter, of the main conduit is required. The cross-sectional area, or diameter, of one branch is required and the total number of branches, which must be an integer, is required. As for the *converging_tee* and *diverging_tee* components, two options are provided for specifying the tee coefficients for combining and dividing manifold flow. The first option is to specify the angle α_2 between the main conduit and the manifold branches, which makes the default tee coefficients, discussed under the *converging_tee* and *diverging_tee* description, available for all flow configurations. Any tee coefficients that are optionally specified are used instead of the default functions. The second option is to specify the tee coefficients for the primary flow directions, which, for *converging_manifold* components, are those indicated as positive in Figure 87 and, for *diverging_manifold* components, are those indicated as positive in Figure 88. The tee

coefficients for other flow configurations default to zero if they are unspecified. It is recommended that *all* tee coefficients be specified unless it is certain that some flow configurations will not occur during the simulation. Specification of *iQ1* and *iQ2* is sometimes necessary to avoid an incorrect component discharge initial condition. See "Boundary and Initial Conditions" in the *Nodes section of this guide for additional information.

Because manifold components are bounded by three nodes, rather than two, their input differs from that of most components. The input formats for *converging_manifold* and *diverging_manifold* components are described in the following two sections.

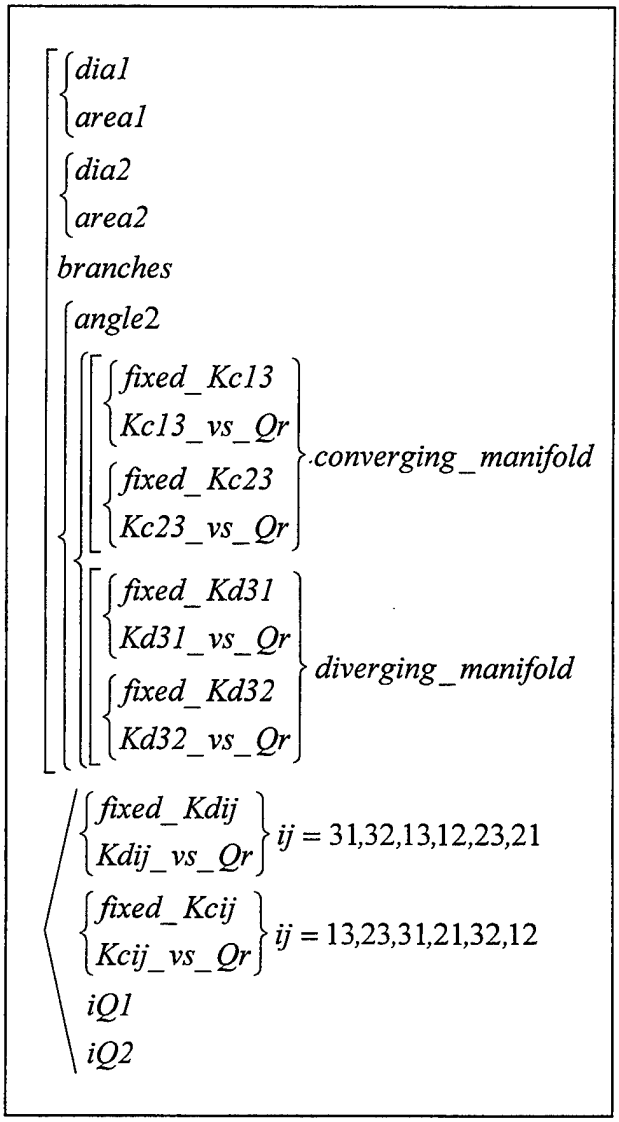


Figure 86: Keyword chart for *converging_manifold* and *diverging_manifold* components

Input for *converging_manifold*

Figure 87 serves as a definition sketch for describing the input format for a *converging_manifold* component. For input purposes, legs 1, 2, and 3 in Figure 87 are fixed regardless of the flow direction, meaning that the leg bounded upstream by node "U" is leg 1 for all flow configurations, the leg bounded upstream by node "B" is leg 2 for all flow configurations, and the leg bounded downstream by node "D" is leg 3 for all flow configurations. The discharges in each leg may be positive or negative. When all three are positive, as indicated in Figure 87, leg 3 carries the combined flow.

Referring to Figure 87, the input format for a *converging_manifold* component is

U B D converging_manifold keyword1=value1, keyword2=value2
keyword3=value3, keyword4=value4, keyword5=value5, ...

where "U" and "D" are the labels assigned to the upstream and downstream nodes for the main conduit and "B" is the label assigned to the node representing the upstream end of the manifold branches.

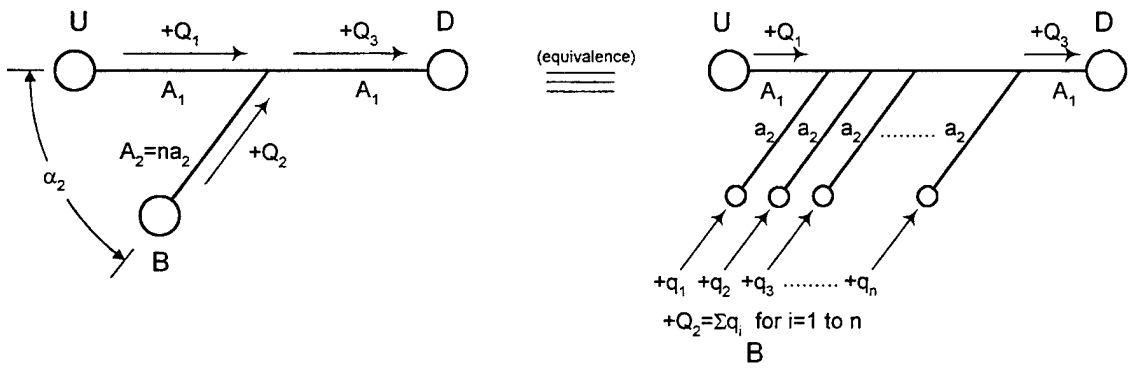


Figure 87: Definition sketch for *converging_manifold* input

Input for *diverging_manifold*

Figure 88 serves as a definition sketch for describing the input format for a *diverging_manifold* component. For input purposes, legs 1, 2, and 3 in Figure 88 are fixed regardless of the flow direction, meaning that the leg bounded upstream by node "U" is leg 3 for all flow configurations, the leg bounded downstream by node "D" is leg 1 for all flow configurations, and the leg bounded downstream by node "B" is leg 2 for all flow configurations. The discharges in each leg may be positive or negative. When all three are positive, as indicated in Figure 88, leg 3 carries the combined flow.

Referring to Figure 88, the input format for a *diverging_manifold* component is

U D B diverging_manifold keyword1=value1, keyword2=value2
keyword3=value3, keyword4=value4, keyword5=value5, ...

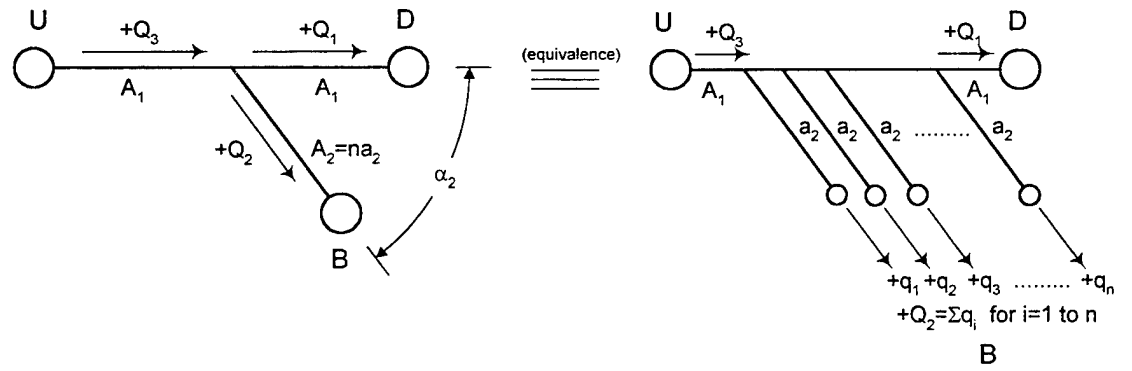


Figure 88: Definition sketch for *diverging_manifold* input

where "U" and "D" are the labels assigned to the upstream and downstream nodes for the main conduit and "B" is the label assigned to the node representing the downstream end of the manifold branches.

Example

Typically, a manifold is represented using several *converging_manifold* or *diverging_manifold* components in combination with several *imp_pipe* or *moc_pipe* components. The accuracy of the solution increases with the number of manifold components used, but usually "several" are all that are required for satisfactory results. In a given application, the sensitivity of the solution to the number of manifold components used can be determined by repeating simulations using different numbers of manifold components and comparing the results. Use of only one manifold component is rare because a manifold with only a few branches is usually better represented using *converging_tee* or *diverging_tee* components for each branch with *imp_pipe* components separating the branches.

The following example describes the manifold depicted in Figure 89 using a combination of pipe, manifold, and *pipe_loss* components as shown in Figure 90. The manifold branches are assumed to connect a main conduit to a "reservoir," or other area of constant pressure. The 36 manifold branches are represented by three manifold components with 12 branches each. The energy losses for flows between the branches and the reservoir are represented by *pipe_loss* components. As indicated by the assigned pipe lengths in Figure 90, each individual manifold component is located at the center of the section of manifold branches that it represents. The nodes in Figure 90 are labeled "A" through "N." The following input defines converging flow as positive and relies on the internal, default tee coefficients (English units are assumed):

```
A  B  imp_pipe  area=10,  length=17.5,  wavespeed=3500,  roughness=.0005
B  K  C  converging_manifold  area1=10,  dia2=.5,  angle2=90,  branches=12
L  K  pipe_loss  ds_area=area=2.356, us_area=1e5, K+=0.5, K-=1.0
C  D  imp_pipe  area=10,  length=35,  wavespeed=3500,  roughness=.0005
D  J  E  converging_manifold area1=10,  dia2=.5,  angle2=90,  branches=12
M  J  pipe_loss  ds_area=area=2.356, us_area=1e5, K+=0.5, K-=1.0
E  F  imp_pipe  area=10,  length=35,  wavespeed=3500,  roughness=.0005
F  I  G  converging_manifold area1=10,  dia2=.5,  angle2=90,  branches=12
G  H  imp_pipe  area=10,  length=17.5,  wavespeed=3500,  roughness=.0005
N  I  pipe_loss  ds_area=area=2.356, us_area=1e5, K+=0.5, K-=1.0
```

Assumed values are used for *wavespeed* and *roughness* in the *imp_pipe* components. The cross-sectional area of each *pipe_loss* component is equal to the total area of the 12 branches for the connecting *converging_manifold* component. The loss coefficients for the *pipe_loss* components are typical values for square-edged entrances and exits. The following input defines diverging flow as positive and relies on the internal, default tee coefficients:

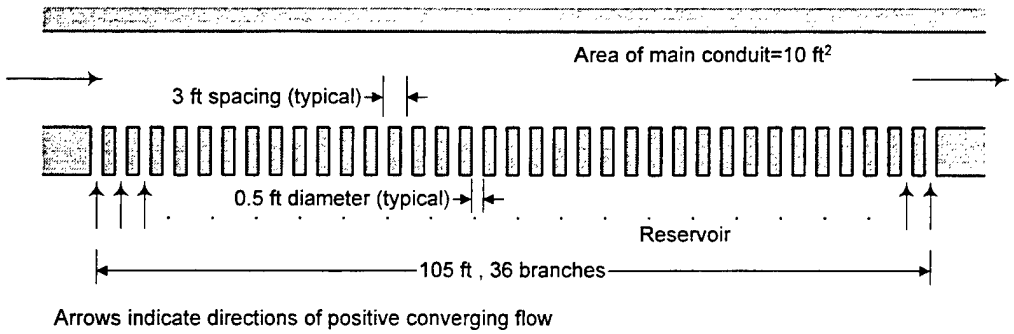


Figure 89: Input example for *converging_manifold* and *diverging_manifold*

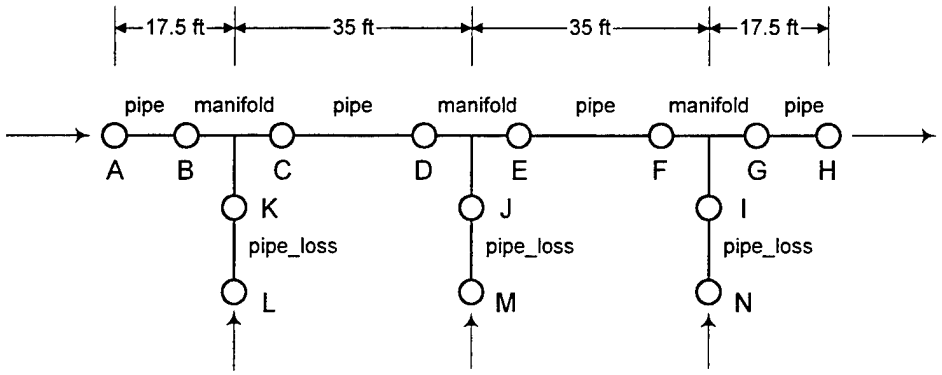


Figure 90: Node and link diagram for manifold example

A B imp_pipe area=10, length=17.5, wavespeed=3500, roughness=.0005
B C K diverging_manifold area1=10, dia2=.5, angle2=90, branches=12
K L pipe_loss us_area=area=2.356, ds_area=1e5, K+=1.0, K-=0.5
C D imp_pipe area=10, length=35, wavespeed=3500, roughness=.0005
D E J diverging_manifold area1=10, dia2=.5, angle2=90, branches=12
J M pipe_loss us_area=area=2.356, ds_area=1e5, K+=1.0, K-=0.5
E F imp_pipe area=10, length=35, wavespeed=3500, roughness=.0005
F G I diverging_manifold area1=10, dia2=.5, angle2=90, branches=12
G H imp_pipe area=10, length=17.5, wavespeed=3500, roughness=.0005
I N pipe_loss us_area=area=2.356, ds_area=1e5, K+=1.0, K-=0.5

The examples provided for *converging_tee* and *diverging_tee* components illustrate use of the tee coefficient specifications.

*Nodes

The **NODES* section must immediately follow the **COMPONENTS* section in the input file. Nodes are the upstream and downstream boundaries of each component and also serve as connection points for different components. A node is either a boundary point, in which case it is a terminal node, or a point of common piezometric head to two or more intersecting components, in which case it is an internal node. Most boundary conditions, initial conditions, and elevations are specified at nodes.

Boundary and Initial Conditions

Boundary conditions for either head or demand (never both) are required at every node. Boundary conditions may be either fixed or variable with simulation time. Variable boundary conditions are specified using functions, which are defined in the **FUNCTIONS* section of the input file. The boundary condition for internal nodes is usually fixed demand equal to zero, which serves as the default boundary condition when neither head nor demand are specified. The boundary conditions for terminal nodes must be specified -- the default condition will not be assumed. Demand is determined at each time step as part of the solution for nodes with specified heads. Head is determined at each time step as part of the solution for nodes with specified demand.

The initial conditions for a simulation consist of specified initial heads for nodes with supply or demand boundary conditions and specified initial demands for nodes with head or pressure boundary conditions. In addition, an initial vapor volume may be specified for any node although this feature is rarely used. Values for all of these initial conditions are required but because they all have default values their input is not required. For nodes with supply or demand boundary conditions, the default initial head is the largest specified boundary or initial head in the network (H_{max}). For nodes with head or pressure boundary conditions, the default initial demand is zero. For all nodes, the default initial vapor volume is zero.

Initial heads and demands for all nodes are usually determined from a steady-state simulation using the initial node heads and demands that are known as boundary conditions. For the remaining nodes, those without known initial heads or demands, the default initial values described above provide a reasonable starting point for the steady-state simulation. LOCKSIM includes a steady-state simulator with an option to automatically write the steady-state results into the input file as initial conditions for the subsequent unsteady simulation.

Initial discharges for each component are also needed at the start of an unsteady simulation. LOCKSIM assumes steady-state flow and uses the specified upstream and downstream initial node heads to compute these initial discharges. However, for some components, tees and free-surface components for example, known upstream and downstream heads do not always determine a unique flow direction. Consequently, the computed initial discharge may flow in the wrong direction. This problem is solved by using the *iQ* keyword for the component in the **COMPONENTS* section of the input file to specify an estimate for the

initial component discharge. When it automatically saves results to the input file, the steady-state simulator in LOCKSIM always specifies the *iQ* keyword for those components for which it is available.

Input Requirements

The input for each node includes a label followed by a list of keyword-value pairs:

node_label keyword1=value1, keyword2=value2, keyword3=value3, ...

The definition of a single node may extend over several lines if desired. The following keywords are supported (input units are given in parentheses; see Table 12):

elevation	node elevation (length) ------ **REQUIRED**
head	fixed head (length)
pressure	fixed pressure (gage pressure)
demand	fixed demand (discharge)
supply	fixed supply (discharge)
ihead	initial head; default is H_{max} (length)
ipressure	initial pressure; default determined from H_{max} (gage pressure)
idemand	initial demand; default is zero (discharge)
isupply	initial supply; default is zero (discharge)
ivapor_volume	initial vapor volume; default is zero (volume)
head_vs_t	label of function specifying head variation with time (time)
pressure_vs_t	label of function specifying pressure variation with time (time)
demand_vs_t	label of function specifying demand variation with time (time)
supply_vs_t	label of function specifying supply variation with time (time)

Note that the *elevation* keyword is required. All of the other keywords are optional because of the available default values. Figure 91 illustrates the keyword requirements for nodes with specified demand or supply while Figure 92 illustrates the keyword requirements for nodes with specified head or pressure. For convenience, different options are provided for specifying boundary and initial conditions. For example, *pressure* may be specified rather than *head* and *supply* may be specified rather than *demand*, where supply equals negative demand. For the time-varying boundary conditions, the units for time are those specified using the *time_units* keyword in the **CONSTANTS* section of the input file

Node elevations are equal to the centerline elevations of the attached closed conduits and closed conduit fittings. For free-surface components, the upstream and downstream node elevations must not be lower than the upstream and downstream bed elevations, respectively.

The pressure at a node is not permitted to be less than vapor pressure (specified using the *vapor_pressure* keyword in **CONSTANTS*). The pressure at nodes attached to free-surface components (*open_channel*, *river_channel*, and *storage*) is not permitted to be less than the ambient barometric pressure (zero gage). If the head and elevation specified at a node result in

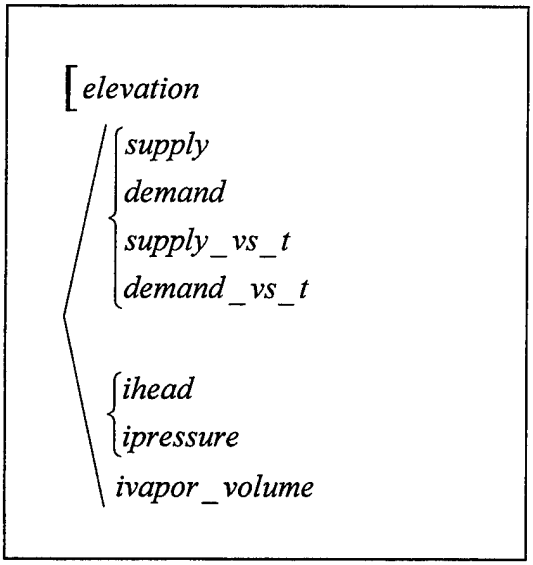


Figure 91: Keyword chart for nodes with specified (or default) demand

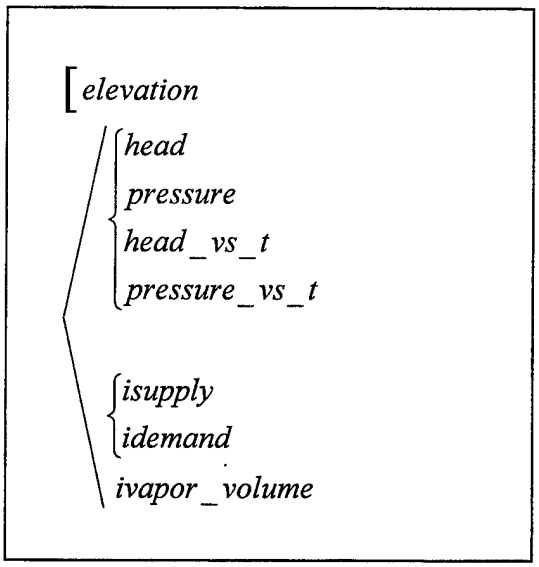


Figure 92: Keyword chart for nodes with specified (or default) head

pressure less than the allowed minimum then the specified head is ignored and the node pressure is set to either vapor pressure or barometric pressure (and a message to this effect is printed).

All nodes referenced in the **COMPONENTS* section of the input file must be specified in the **NODES* section. As with components, nodes may be included in the input file in any desired order, but the order of input determines the order of output in the tabular reports.

*Functions

LOCKSIM functions define variations in node demand or head with time, valve position with time, valve loss coefficient with position, storage surface area with elevation, and other required functional relationships. Four types of functions of a single variable are supported: *discrete*, *polynomial*, *power*, and *sinusoid*. In addition, two or more of these base functions may be combined to create a *composite* function, in which individual function segments join together to create one overall function, or a *summation* function, in which individual functions are added together to create one function.

Each function specification starts with a label, by which the function is referred to, followed by a type identifier followed by as many keyword-value pairs as needed:

function_label function_type keyword1=value1, keyword2 = value2, ...
keyword3=value3, keyword4=value4, ...

The type identifier *function_type* is either *discrete*, *polynomial*, *power*, *sinusoid*, *composite*, or *summation*. Each of these types has its own list of keywords as described separately below. In addition to the type-specific keywords, the following keywords are valid for all functions: *xshift*, *xscale*, *yshift*, *yscale*, *Xmin*, and X*max*. Both the common keywords and the type-specific keywords may be specified in any desired order.

All functions referenced in the **COMPONENTS* and **NODES* sections of the input file must be specified in the **FUNCTIONS* section. Functions may be included in any desired order.

General Description and Common Keywords

A function in LOCKSIM relates an independent variable y (a node demand or a valve position, for example) to a dependent variable x (most often time), expressed by the relationship $y = f(x)$. As an input convenience, however, the user provides a normalized form of this function, defined as

$$Y = f(X)$$

in which

$$X = (x - \beta_x) / \phi_x,$$
$$Y = (y - \beta_y) / \phi_y,$$

and β_x, β_y, ϕ_x, and ϕ_y shift and scale the function provided in X-Y coordinates to the x-y coordinates required in LOCKSIM. Use of these shift and scale parameters permits a valve position, for example, to be defined nondimensionally as percent open (Y) versus percent of opening time (X). Simulations for different opening times can then be performed by changing the value of ϕ_x to the desired total opening time for each simulation, rather than redefining the x-

coordinates of the function for each additional simulation. In using the shift and scale parameters it is important to remember that the function used by LOCKSIM is y = f(x), not Y = f(X). These two functions are equivalent only when the default values of zero for β_x and β_y and 1.0 for ϕ_x, and ϕ_y are used.

Figure 93 illustrates the effects of the shift and scaling parameters (one at a time) on a sample input function Y = f(X), which for simplicity is depicted as a simple linear relationship. The illustrations show that positive values of β_x and β_y shift the input function to the right and upward, respectively, while values of ϕ_x and ϕ_y multiply the X- and Y-coordinates.

The results of using two or more of the shift and scale parameters simultaneously are sometimes counter-intuitive, but they are predictable. The definitions for X and Y can be rearranged to define x and y as follows:

$$y = \phi_y Y + \beta_y,$$
$$x = \phi_x X + \beta_x$$

As indicated, the x and y coordinates are obtained by first multiplying the X and Y coordinates by ϕ_x and ϕ_y, respectively, and then adding the shift parameters β_x and β_y. Use these definitions to determine the effects of specific shift and scale parameters. Also, specify the independent variable, y, as a plot variable so that its variation during the simulation can be conveniently checked.

The keywords common to all functions are listed below with their default values. As already mentioned, the default values of the shift and scale parameters result in y = Y and x = X. Each function is defined within the range *Xmin* to *Xmax*. For X values less than *Xmin*, Y = f(*Xmin*) and for X values greater than *Xmax*, Y = f(*Xmax*).

Keyword	Symbol	Default
xshift	β_x	0
xscale	ϕ_x	1
yshift	β_y	0
yscale	ϕ_y	1
Xmin		-10^{10}
Xmax		10^{10}

The units of X and Y depend on the function's use. For example, if X represents simulation time then the units of X are those defined by *time_units* in **CONSTANTS*. If Y represents valve discharge coefficient then Y is dimensionless.

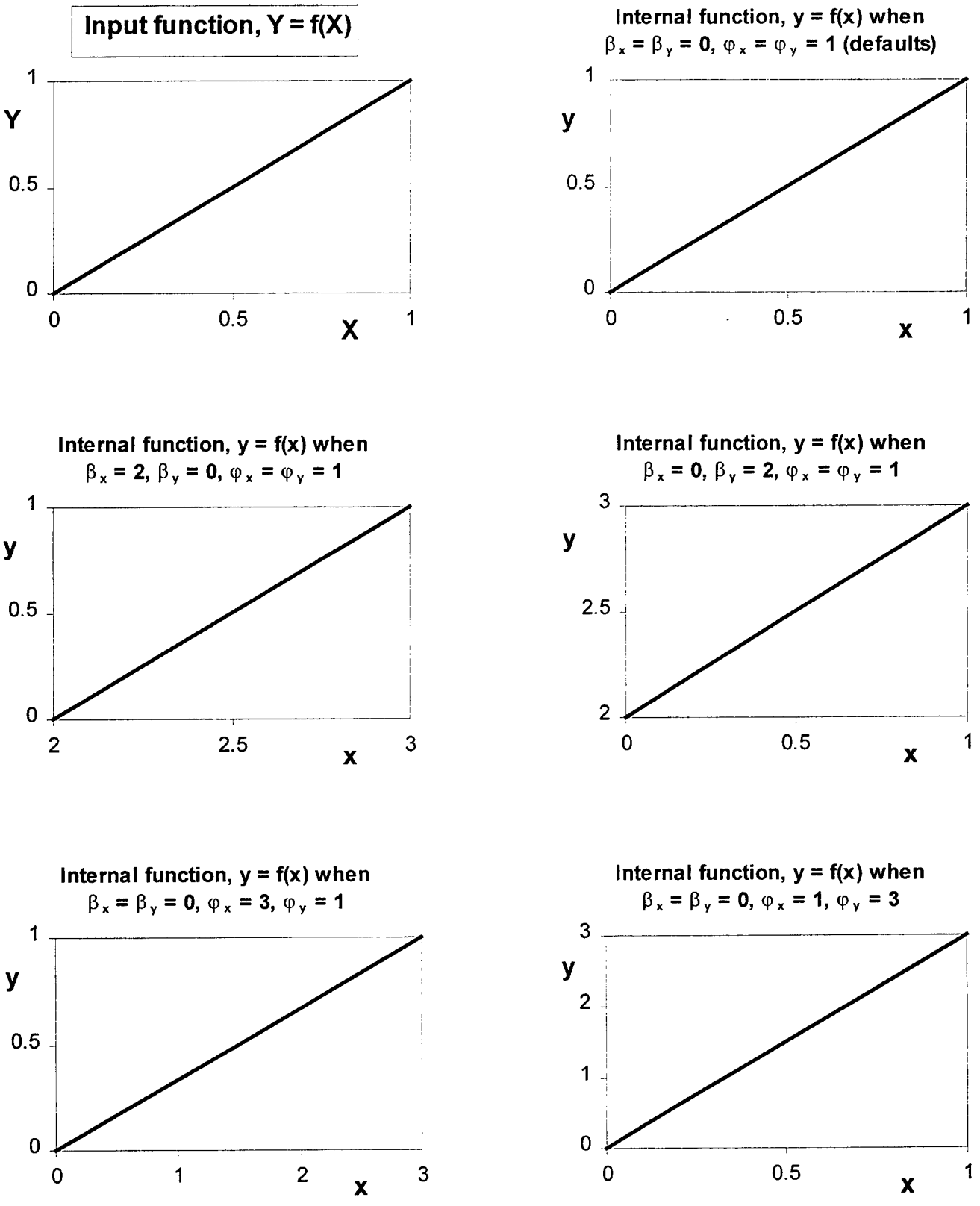


Figure 93: Effects of shift and scale parameters on input function Y = f(X)

Discrete Functions

Figure 94 shows a sample *discrete* function and illustrates the available interpolation and extrapolation options. A d*iscrete* function consists of a set of X-Y coordinate pairs (points) which define a curve. As many X-Y coordinate pairs as desired may be specified. A minimum of one X-Y pair must be specified, in which case Y is assumed to be constant, equal to its one given value, for all X.

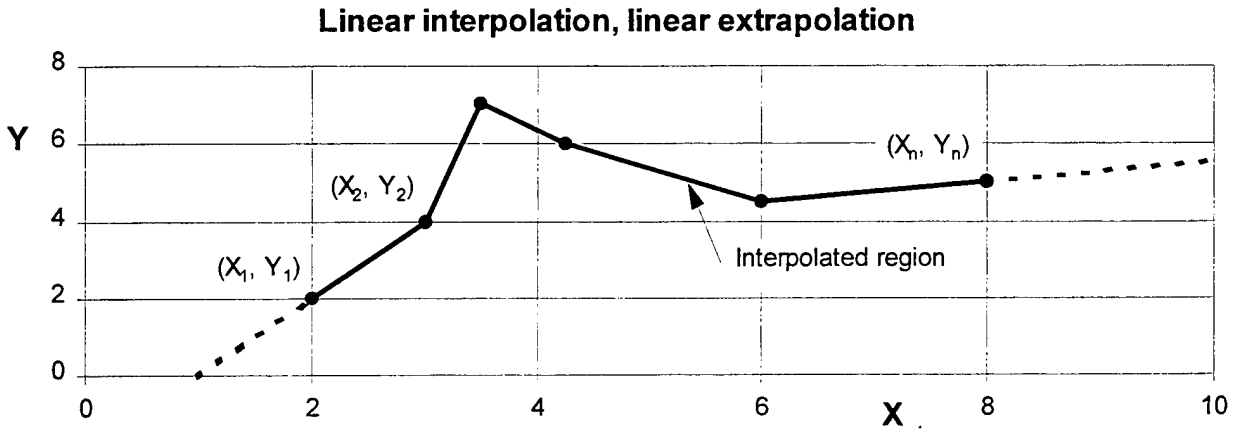
Interpolation, which may be *linear*, *spline*, or *step*, provides values of Y for values of X in between the specified points. As illustrated in Figure 94, *linear* interpolation connects adjacent points of a function using straight lines. *Spline* interpolation connects adjacent points of a function using cubic splines with "natural" endpoint conditions. *Step* interpolation maintains a constant Y value between points equal to the Y value at the beginning of the interval. The *spline* option should be used only when the discrete points define a relatively smooth curve. Even in these cases, it is advisable to check the results of the *spline* interpolation either by plotting the function using a software package that provides cubic spline interpolation or by saving the independent variable y as a plot variable during a simulation and eventually plotting it.

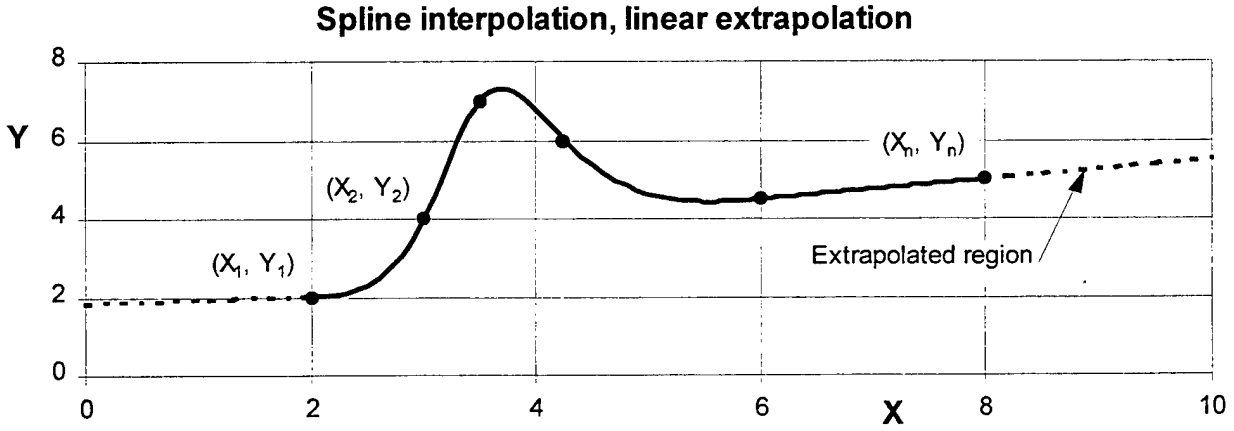
Extrapolation, which may be *constant* or *linear*, provides values of Y for values of X outside the range of the given points. *Constant* extrapolation specifies that $Y = f(X_1)$ for all values of X less than X_1, where X_1 is the smallest X value, and $Y = f(X_n)$ for all values of X greater than X_n, where X_n is the largest X value. As evident in Figure 94, the results of *linear* extrapolation depend on the *interpolation* choice. For *discrete* functions using *linear* or *spline* interpolation, Y values for out-of-range X values are determined by linear extrapolation from the endpoints using the slope of the interpolated curve at the endpoint. For *discrete* functions using *step* interpolation, *linear* extrapolation is treated the same as *constant* extrapolation.

The keywords specific to *discrete* functions consisting of *n* points are listed below:

Keyword	Default	Format of Value
xy_pairs	none	$\{ X_1, Y_1 \;\; X_2, Y_2 \;\; X_3, Y_3 \ldots\ldots X_n, Y_n \}$
x_values	none	$\{ X_1, X_2, X_3, \ldots\ldots, X_n \}$
y_values	none	$\{ Y_1, Y_2, Y_3, \ldots\ldots, Y_n \}$
share_x	none	label of another function
share_y	none	label of another function
share_xy	none	label of another function
interpolation	*linear*	*linear*, *step*, or *spline*
extrapolation	*constant*	*constant* or *linear*

Coordinate pairs and values are specified using keywords set equal to data clusters (see GENERAL INPUT RULES). X values, which do not need to be evenly spaced, must be input in monotonically increasing order. Because *discrete* functions require a set of X-Y coordinates, one of the following combinations of keywords must be specified:





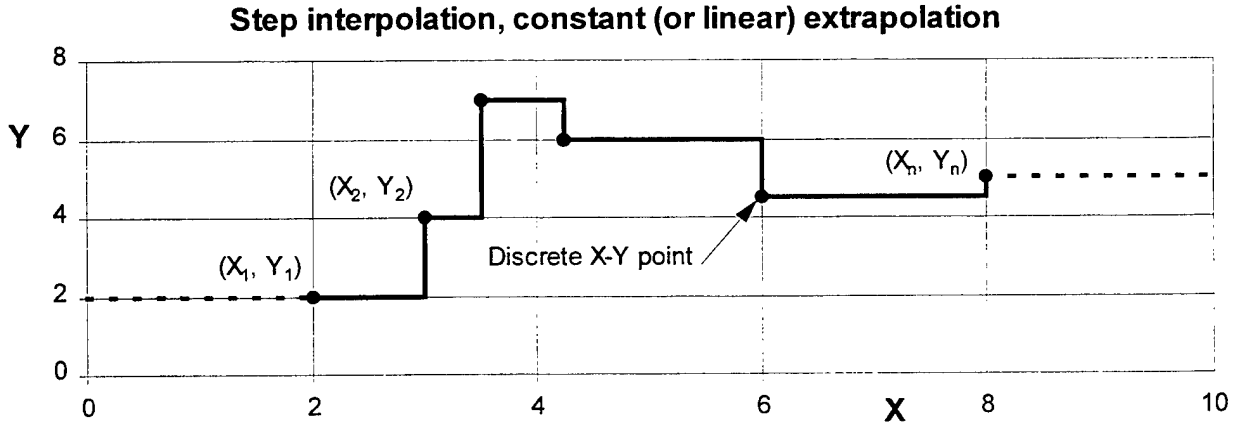


Figure 94: Interpolation and extrapolation options for a discrete input function, Y = f(X)

xy_pairs,
x_values and *y_values*,
share_xy,
share_x and *share_y*,
x_values and *share_y*, or
y_values and *share_x*

The *share* keywords permit different functions to share the same set of X and Y, or X, or Y coordinates. Each function sharing a set of coordinates can define its own shift (*xshift* and *yshift*) and scale (*xscale* and *yscale*) values without effecting any of the other functions. To share data, the coordinates are given in full for one function. The other functions then use the *share_x*, *share_y*, or *share_xy* keywords set equal to the name of the function for which the data they wish to share is defined. In the input file, the function defining the coordinates may either precede or follow the function sharing the coordinates. Figure 95 shows the keyword requirements for *discrete* functions.

The following examples illustrate a few ways in which the discrete functions illustrated in Figure 94 could be specified (the function labels indicate which function is described):

```
*FUNCTIONS
top_curve  discrete
    xy_pairs={ 2,2  3,4  3.5,7  4.25,6  6,4.5  8,5 }
    interpolation=extrapolation=linear

middle_curve  discrete  xscale=1, xshift=0
    interpolation=spline, extrapolation=linear
    x_values={ 2,  3,  3.5,  4.25,  6,  8 }
    y_values={ 2,  4,   7,    6,   4.5,  5 }

bottom_curve  discrete  interpolation=step,
    share_xy=middle_curve
```

In the first example, the keyword *interpolation* is unnecessary (but harmless) because the default value is *linear*. Similarly, the keywords *xscale* and *xshift* are unnecessary in the second example, but are included for illustration purposes. Because data clusters can be defined over as many lines as needed, the first example could also be specified as

```
top_curve  discrete
    interpolation=extrapolation=linear
    xy_pairs={
        2,2     3,4
```

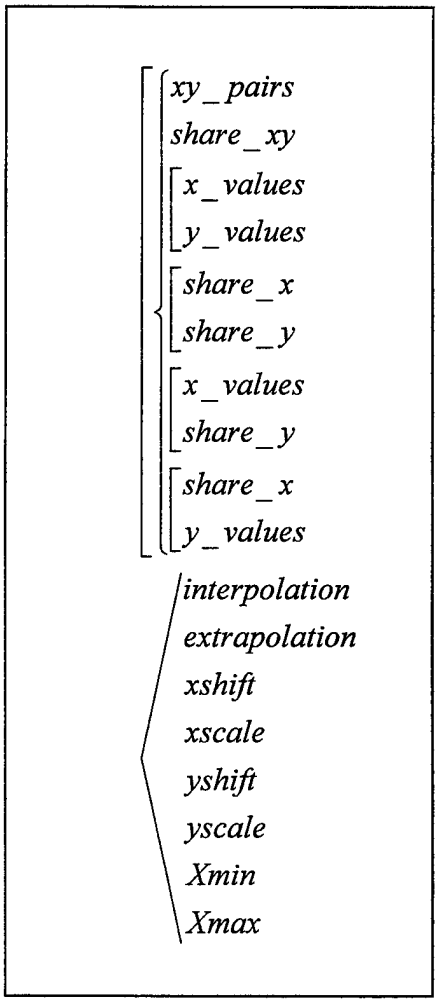


Figure 95: Keyword chart for *discrete* functions

3.5,7 4.25,6
6,4.5 8,5 }

Polynomial Functions

Polynomial functions are specified by a set of n+1 coefficients, a_o, a_1, a_2, ... , a_n, which are applied as follows to define Y = f(X):

$$Y = a_o + a_1X + a_2X^2 + a_3X^3 + ... + a_nX^n$$

The coefficients are input using a data cluster set equal to the keyword *coefficients*:

coefficients={ a_o, a_1, a_2, ..., a_n }

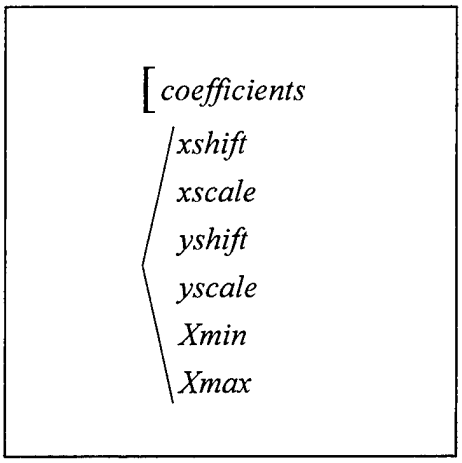


Figure 96: Keyword chart for *polynomial* functions

The number of coefficients, n+1, can be as large as desired to define Y = f(X). The minimum number of coefficients is one.

Figure 96 illustrates the keyword requirements for *polynomial* functions. Although it is not required, it is good practice to specify the keywords *Xmin* and *Xmax* when using *polynomial* functions, especially when the polynomial is the result of curve fitting over a finite range of X values. Curve-fit polynomials can give unexpected results when used with X values outside of the range over which they were fit.

Power Functions

Power functions are specified by a set of n+1 coefficients, a_o, a_1, a_2, ... , a_n and a set of n+1 powers, which are applied as follows to define Y = f(X):

$$Y = a_0X^{b_0} + a_1X^{b_1} + a_2X^{b_2} + a_3X^{b_3} + ... + a_nX^{b_n}$$

The coefficients and powers are input using data clusters set equal to the keywords *coefficients* and *powers*, respectively:

coefficients={ a_o, a_1, a_2, ..., a_n }
powers={ b_o, b_1, b_2, ..., b_n }

The number of powers provided must be equal to the number of coefficients provided. The number of coefficients and powers, n+1, can be as large as desired to define Y = f(X). The minimum number of coefficients and powers is one.

Figure 97 illustrates the keyword requirements for *power* functions. As with *polynomial* functions, it is good

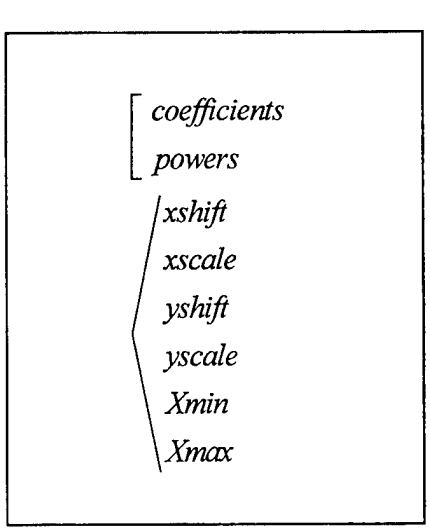


Figure 97: Keyword chart for *power* functions

practice to specify the keywords *Xmin* and *Xmax* when using *power* functions determined from curve fits.

Sinusoid Functions

Sinusoid functions describe a damped sinusoid defined by

$$Y = A_{mo}\, e^{-2\pi\lambda fX} \sin[2\pi\,(fX + \phi)]$$

in which A_{mo} = amplitude of Y at X = 0, λ = damping coefficient, f = frequency, and ϕ = phase shift (as a fraction of the period, T, which is equal to 1/f). The following keywords are used to define these parameters:

Keyword	Default	Range	Parameter
amplitude	0	any value	A_{mo}
frequency	0	any value	f
damping	0	any value	λ
phase_shift	0	any value	ϕ

As indicated in Figure 98, which illustrates the keyword requirements, all of these keywords are optional. However, the function described if no keywords are specified is simply Y = 0. The *phase_shift* keyword will accept any value but its reasonable range is 0 to 1. A cosine variation results when *phase_shift* is set to 0.25, which shifts the wave form 0.25 cycles to the left along the X axis.

The *sinusoid* function can be used to describe a wavy water level (head) boundary condition. The mean water level about which the water surface oscillates is specified using the *yshift* keyword.

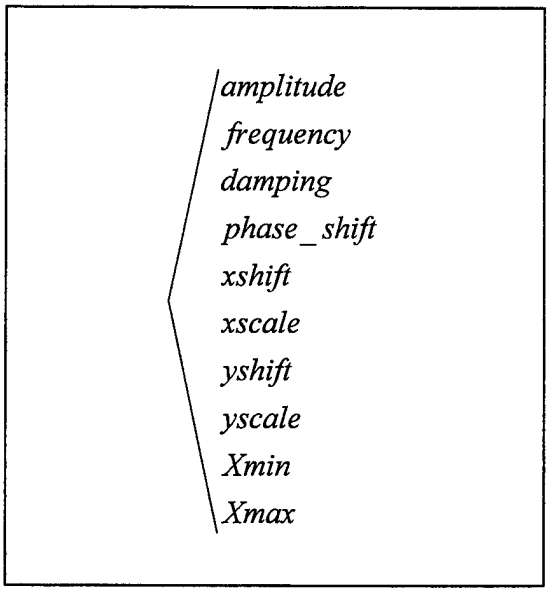


Figure 98: Keyword chart for *sinusoid* functions

Composite Functions

Figure 99 shows a sample *composite* function. A *composite* function represents one curve using a series of individual functions (f_1 through f_4 in Figure 99), each applicable over a discrete range of X values. The X range of the first function (f_1 in Figure 99) is the minimum X, specified by the keyword *Xmin*, to the first transition X value (X_{tr1} in Figure 99). The X range for the remaining functions, except for the last one, is the previous transition X value to the next transition X value. The X range for the last function (f_4 in Figure 99) is the last transition X value (X_{tr3} in Figure 99) to the maximum X, specified by the keyword *Xmax*. The *composite*

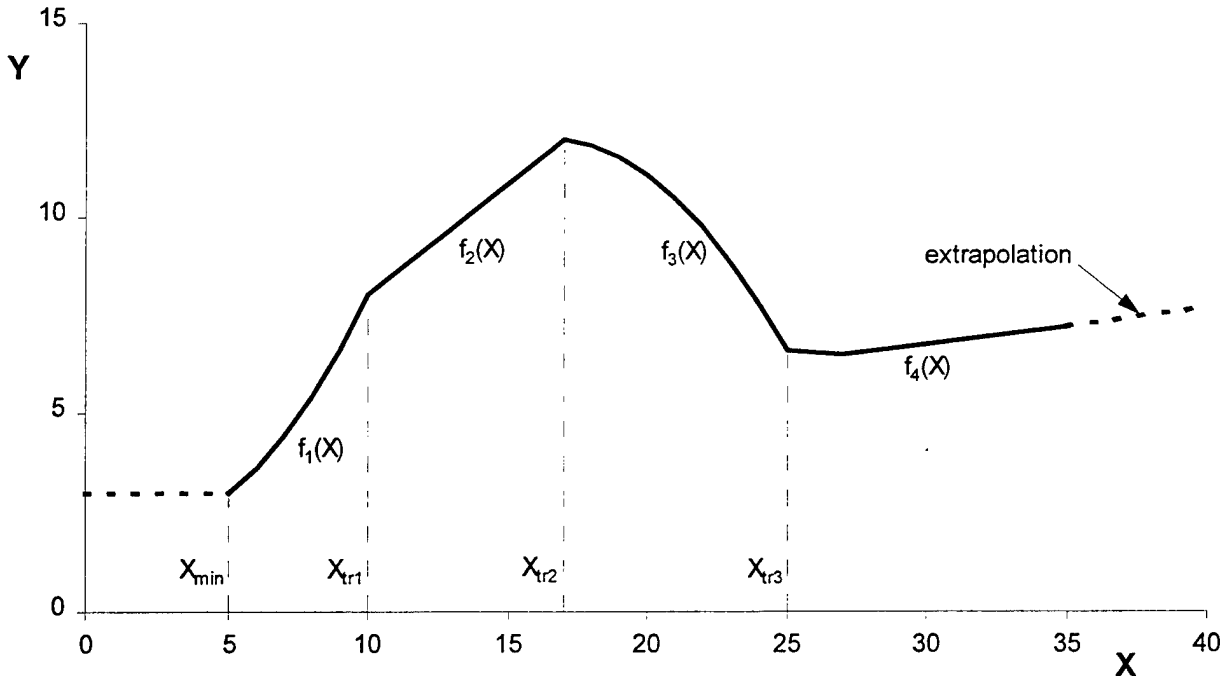


Figure 99: Composite function illustration and example

function is extrapolated based on the selected extrapolation options for the first and last individual functions.

The keywords *Xmin* and *Xmax* may be specified for the individual functions, for the composite function, or for both. When *Xmin* is specified for both, the effective minimum X is the larger of the two. When *Xmax* is specified for both, the effective maximum X is the smaller of the two.

For a *composite* function consisting of *n* function segments, the individual functions are specified using the *functions* keyword set equal to a data cluster of *n* function labels:

functions={ $label_1$, $label_2$, $label_3$, ..., $label_n$ }

As many functions as desired may be listed, but at least one must be specified. The individual functions, each a segment of the overall function, are of type *discrete*, *polynomial*, *power*, or *sinusoid*. Each function may be a different type, if desired. In the input file, the individual function definitions may either precede or follow the composite function definition.

For a *composite* function consisting of *n* function segments, *n*-1 transition X values are required. These are specified setting the *Xtransitions* keyword equal to a data cluster of transition X values:

Xtransitions={ X_{tr1}, X_{tr2}, X_{tr3}, ..., $X_{tr(n-1)}$ }

The transition X values must be input in monotonically increasing order. To ensure a continuous overall function, at each transition X value the Y values obtained from the adjoining functions are required to be within 0.01 percent of each other ($f_1(X_{tr1})=f_2(X_{tr1})$, $f_2(X_{tr2})=f_3(X_{tr2})$, etc.). The effective minimum and maximum X values are automatically adjusted if conflicts with the transition X values are detected. If X_{tr1} is less than the minimum X, then the effective minimum X is set equal to X_{tr1}. If $X_{tr(n-1)}$ is greater than the maximum X, then the effective maximum X is set equal to $X_{tr(n-1)}$.

Figure 100 illustrates the keyword requirements for *composite* functions. The example illustrated in Figure 99 is described by the following input:

```
*FUNCTIONS
example  composite  functions={f1, f2, f3, f4 }, xtransitions={ 10, 17, 25 }

f1  polynomial  coefficients={ 3, -0.5, 0.1 },  Xmin=5
f2  discrete  xy_pairs={ 10,8   17,12 }
f3  polynomial  coefficients={ -8.4, 2.475, -0.075 }
f4  discrete  xy_pairs={ 25,6.6  26,6.5  27,6.45  28,6.5  35,7.2 }, extrapolation=linear
```

Alternatively, the *Xmin* keyword could have been specified for the *composite* function, *example*, rather than for function *f1*. The keyword Xmax is not used in this example (the default value is accepted). For values of X greater than 35, function *f4* is extrapolated to obtain values of Y.

The shift (*xshift* and *yshift*) and scale (*xscale* and *yscale*) parameters can be used for both the *composite* and individual functions, but the results are sometimes difficult to predict. Shifts and scales specified for a *composite* function are applied after those for its individual functions. It may be helpful to think of the i^{th} individual function as describing $Y^* = f_i(X^*)$ before its shifts and scales are applied. The shifts and scales transform the functional relationship into $Y = f_i(X)$, which is a segment of the overall $Y = f(X)$ curve. The shifts and scales for the *composite* function then transform $Y = f(X)$ into $y = f(x)$, which is the functional relationship used by LOCKSIM.

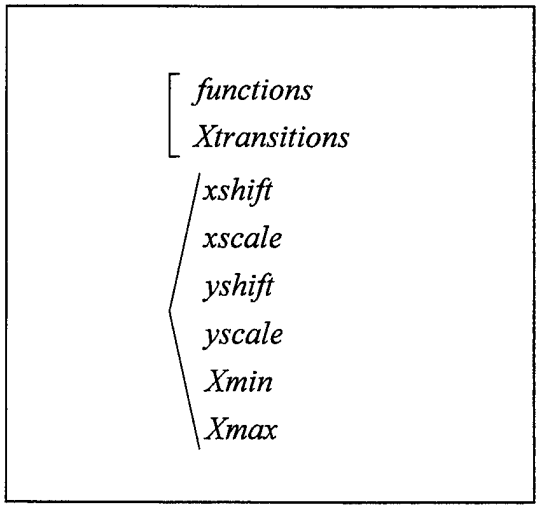


Figure 100: Keyword chart for *composite* functions

Summation Functions

A *summation* function superimposes two or more individual functions to define $Y = f(X)$ as follows:

$$Y = f_1(X) + f_2(X) + f_3(X) + \ldots\ldots + f_n(X)$$

where n is the total number of individual functions. For a given X, Y is determined by adding the Y values corresponding to the given X for each

individual function. The individual functions are specified using the *functions* keyword set equal to a data cluster of *n* function labels:

functions={ $label_1$, $label_2$, $label_3$, ..., $label_n$ }

As many functions as desired may be listed, but at least one must be specified. The individual functions are of type *discrete*, *polynomial*, *power*, *sinusoid*, *composite*, or *summation*. Each function may be a different type, if desired. In the input file, the individual function definitions may either precede or follow the *summation* function definition. A *summation* function cannot include itself as one of its individual functions, but it can include other *summation* functions. Figure 101 illustrates the keyword requirements for *summation* functions.

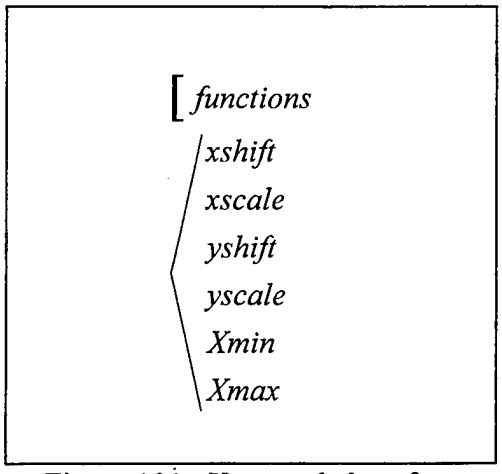


Figure 101: Keyword chart for *summation* functions

The shift (*xshift* and *yshift*) and scale (*xscale* and *yscale*) parameters can be used for both the *summation* and individual functions, but, like the *composite* function, the results are sometimes difficult to predict. Shifts and scales applied to the individual functions result in the following expression for Y = f(X):

$$Y = \sum_{i=1}^{n}\left[\beta_{yi} + \phi_{yi} f_i\left(\frac{X-\beta_{xi}}{\phi_{xi}}\right)\right]$$

in which β_{yi}, β_{xi}, ϕ_{yi}, and ϕ_{xi} are the shift and scale parameters for the i^{th} individual function and $X = (x - \beta_x) / \phi_x$, where β_x and ϕ_x are *xshift* and *xscale* specified for the *summation* function. The required y value is obtained after application of *yshift* and *yscale* specified for the *summation* function to Y.

*Cross_Sections

The geometry and friction characteristics of cross sections used by the *open_channel* and *river_channel* components are described in this section. The cross section descriptions are separated from the component definitions to permit one cross section definition, and the computer memory in which it is stored, to be shared by multiple free-surface components. For the same reason, functions are described in a separate section from where they are referenced.

Each cross section definition starts with a label, by which the cross section is referred to, followed by a type identifier followed by as many keyword-value pairs as needed:

cross_section_label cross_section_type keyword1=value1, keyword2 = value2, ...
keyword3=value3, keyword4=value4, ...

The type identifier *cross_section_type* must be *trapezoidal, circular,* or *riverine.* As for functions, each cross section type has its own list of keywords in addition to the keywords that are valid for all cross sections.

The following keywords are valid for all cross sections (input units are given in parentheses; see Table 12):

Keyword	Default	Parameter
DW_f	none	Darcy-Weisbach friction factor, *f*
MN_n	none	Manning's *n*
roughness	none	roughness height in Moody's equation (length)
beta	1.0	momentum correction factor, β

The friction characteristics (shear stress) of a cross section are defined by specifying either the Manning's *n*, the Darcy-Weisbach *f*, or the roughness height to be used in Moody's relationship describing *f* as a function of Reynolds number. For cross sections used in *open_channel* and *river_channel* components, one of the friction keywords, *DW_f, MN_n,* or *roughness* must be specified. In addition, the momentum correction factor, β, may be specified if desired.

The keywords *DW_f, MN_n,* and *beta* are assigned fixed values for *trapezoidal* and *circular* cross sections and fixed values or data clusters for *riverine* cross sections. Data clusters are used for *riverine* cross sections that are divided into lateral subsections, each with its own value of *DW_f, MN_n,* or *beta.* The keyword *roughness* can only be assigned a constant value and may not be used with subsections.

The bottom elevation of a cross section is specified in the **COMPONENTS* section of the input file each time the cross section is referenced. This is necessary to permit one cross section definition to be used at various locations with different bottom elevations.

The keywords *autofix_k's*, *autofix_xsects*, *depth_min*, *xsect_file*, and *xsect_warning*, which are specified in the **CONSTANTS* section of the input file, all apply to cross sections. In addition to the following sections, see the *Constants section of this guide for their descriptions and usage.

All cross sections referenced in the **COMPONENTS* section of the input file must be specified in the **CROSS_SECTIONS* section. Cross section definitions may be included in any desired order.

Trapezoidal Cross Section

The shape of a *trapezoidal* cross section is defined by its bottom width, w, and its two side slopes, m_1 and m_2. Additional parameters for a *trapezoidal* cross section containing water are depicted in Figure 102, in which T = top width, d = depth, and z_b = elevation of channel bottom. The top width and depth are determined during a simulation. The bottom elevation is specified in the **COMPONENTS* section of the input file when the cross section is referenced. There is no limit to the maximum depth in a *trapezoidal* cross section. The sides are assumed to extend as high as needed to contain the flow.

The following keywords describe a *trapezoidal* cross section:

Keyword	Symbol	Default	Parameter
bed_width	w	none	bed, or base, width (length) ------ **REQUIRED**
side_slope1	m_1	0	slope of one side wall, dx/dy
side_slope2	m_2	0	slope of opposite side wall, dx/dy

The *bed_width* keyword is required. The side slopes are defined as the incremental horizontal change for unit vertical change. This means that both side slopes equal to zero defines a rectangular section. Figure 103 illustrates the keyword requirements for *trapezoidal* cross sections.

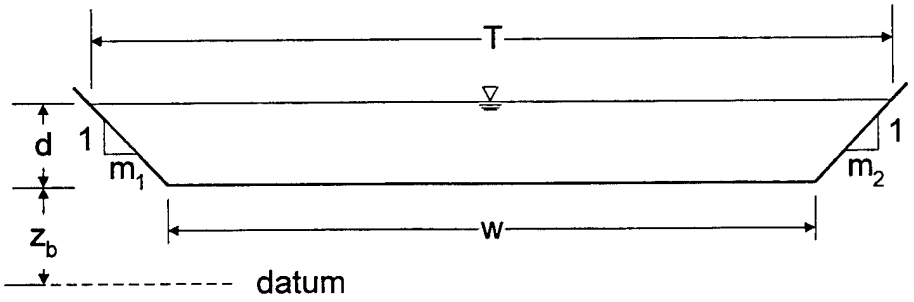


Figure 102: Definition sketch for *trapezoidal* cross section

Circular Cross Section

The shape of a *circular* cross section is entirely defined by its diameter, D. Additional parameters for a *circular* cross section containing water are depicted in Figure 104, in which T = top width, d = depth, and z_b = elevation of channel bottom. The top width and depth are determined during a simulation. The bottom elevation is specified in the **COMPONENTS* section of the input file when the cross section is referenced.

The following keyword describes a *circular* cross section:

Keyword	Symbol	Default	Parameter
diameter	D	none	diameter (length) ------ **REQUIRED**

Figure 105 illustrates the keyword requirements for *circular* cross sections.

The free-surface components for which *circular* cross sections are defined cannot handle closed-conduit flow. When the water depth exceeds the diameter of a *circular* cross section an error message is printed and the simulation is terminated.

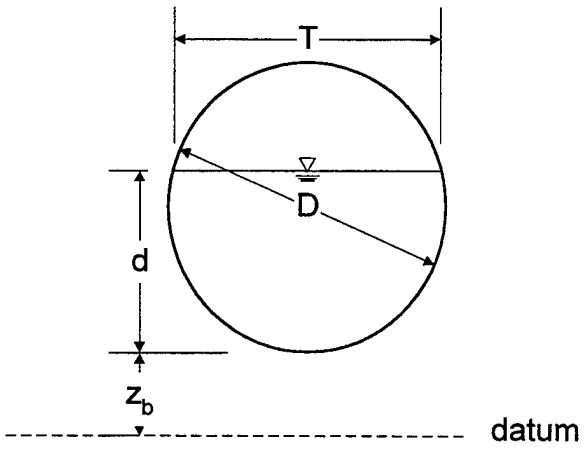


Figure 104: Definition sketch for *circular* cross section

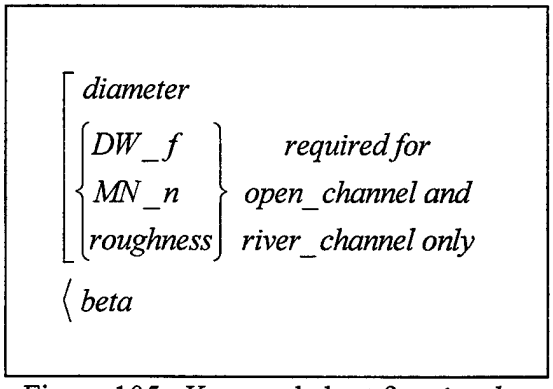


Figure 105: Keyword chart for *circular* cross sections

Riverine Cross Section

General Description

The shape of a *riverine* cross section is defined by a series of discrete points, either distance-elevation (x-z) or elevation-width (z-w) coordinate pairs, connected by straight lines (linear interpolation). As many x-z or z-w coordinate pairs as desired may be specified but a minimum of three coordinate pairs must be specified. When x-z coordinate pairs are specified, the cross section may be divided into lateral subsections, each with its own value of *DW_f* or *MN_n* and *beta*. As many subsections as desired may be specified. However, when z-w coordinate pairs are specified, subsections may not be defined.

Figure 106 shows a sample *riverine* cross section defined by *n* x-z coordinate pairs and divided into three subsections. The x values, which do not need to be evenly spaced, must be input in monotonically increasing order. Likewise, the z values must be input in monotonically increasing order for cross sections specified using z-w coordinate pairs. In both cases, the z values are not treated as actual elevations but only as indicating differences in elevation. The actual elevations are determined by adding the differences between the z values for each point and the minimum z value (700 for the example in Figure 106) to the bottom elevation, z_b, specified when the cross section is referenced in the **COMPONENTS* section of the input file. For cross sections specified using x-z coordinates, it is required that z_1 be greater than z_2 and z_n be greater than z_{n-1}, which means that the slopes of the channel banks must result in increasing channel width with increasing elevation.

Additional parameters for a *riverine* cross section containing water are depicted in Figure 106, in which T = top width and d = depth, which is defined as the water surface elevation minus the channel bottom elevation, z_b. The top width and depth are determined during a simulation.

For cross sections specified using x-z coordinates, Figure 106 also illustrates the default *extrapolation* option, which is *linear*. The alternative *extrapolation* option is *constant*, in which case, the sides of the cross section are assumed to extend vertically above z_1 at x_1 and z_n at x_n.

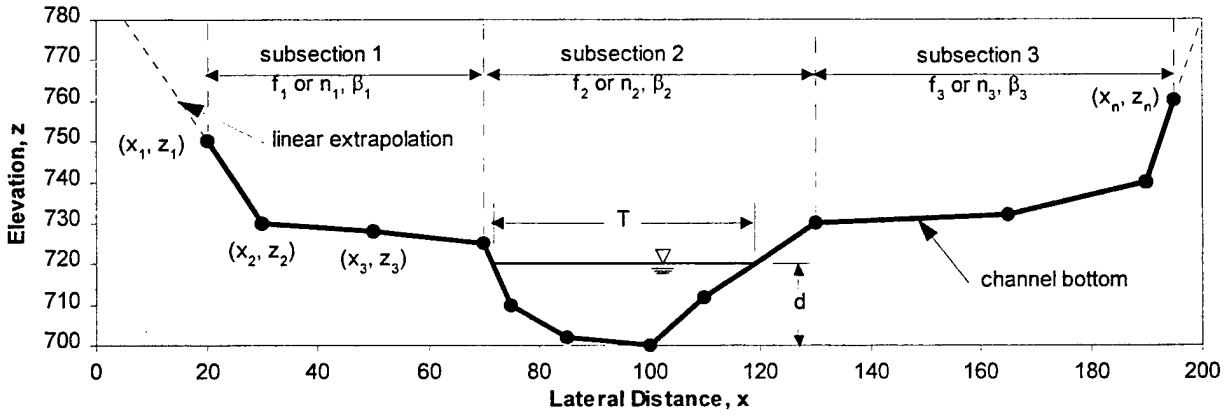


Figure 106: Sample *riverine* cross section defined by x-z coordinate pairs

Consequently, the channel width is constant for all z values greater than the maximum specified z-value (z_n in Figure 106). When *extrapolation=constant* is specified, an additional elevation value equal to the maximum elevation plus 10 percent of the difference between it and the minimum elevation is added to the table of geometric parameters reported in the cross section file specified using the *xsect_file* keyword in the **CONSTANTS* section of the input file. This additional point allows LOCKSIM to internally treat *constant* extrapolation in the same manner as *linear* extrapolation.

For cross sections specified using *n* z-w coordinates, the default *linear* extrapolation specifies that, for z values above z_n, the channel width continues to increase at the rate of increase between z_n and z_{n-1}. *Constant* extrapolation specifies that the channel width remains equal to w_n for all z values above z_n.

The *xsect_warning* keyword, which is defined in the **CONSTANTS* section of the input file, specifies whether or not to print a warning message to the monitor and error file (LOCKSIM.MSG) when a converged solution has been obtained by extrapolating a *riverine* cross section's tabulated data or by setting a minimum depth in a dry cross section.

Subsections

Subsections are defined to approximately account for the effects of lateral variations in cross-sectional geometry and roughness on conveyance, K, which is a measure of the water-carrying capacity of the cross section, and momentum correction coefficient, β. Given a water surface elevation, the hydraulic properties of each subsection are computed separately and combined to yield values of K and β for the entire cross section. Other important geometric parameters, such as top width, area, and wetted perimeter, are not affected by dividing the cross section into subsections. For further discussion of this topic see, for example, Chow (1959), Cunge et al. (1980), or Garbrecht and Brown (1991).

In LOCKSIM, subsections are defined by specifying the x coordinates of their boundaries within the cross section. The x coordinates of the left and right bank subsection boundaries are not input. The number of subsection x values specified should be one less than the number of subsections, and each subsection x value should lie between the left and right bank x values. It is not required that the subsection x values coincide with x values from the x-z coordinate pairs that define the channel bottom. For the example in Figure 106, the subsections are defined by specifying the x values 70 and 130, which are the boundaries between subsection 1 and 2 and subsection 2 and 3, respectively. The extrapolated regions to the outside of the left and right banks are assumed to have the same friction and momentum correction characteristics as the adjacent subsections.

Geometric Parameters

The specified x-z or z-w coordinates defining the channel bottom are used along with the friction specification to construct a discrete table of geometric parameters as functions of elevation, z. The required parameters at each discrete z include top width, T(z); area, A(z); wetted perimeter, P(z); and conveyance, K(z). In addition, when subsections are defined and the

momentum correction factor, β, is specified for each subsection, the function $\beta(z)$ is included in the table.

The parameter functions T(z) and P(z) are accurately determined from the specified cross-sectional shape, under the assumption that all coordinate points are connected by straight lines. When the cross-sectional shape is described by z-w coordinate pairs, T(z) is equal to w(z) with interpolation between the given w values, and P(z) is determined under the assumption that the cross section is laterally symmetrical.

Because the governing equations for the *open_channel* and *river_channel* components depend on the assumption that $(\partial A/\partial z)_x = T$, the area function, A(z), is determined for discrete level k by integrating the discrete T(z) function as follows:

$$A_k = A_{k-1} + \frac{1}{2}(T_k + T_{k-1})(z_k - z_{k-1}) \tag{77}$$

in which discrete level k-1 is the level immediately below discrete level k. The degree to which the areas computed using Equation 77 agree with the true areas determined from the details of the cross-sectional shape depends on how the discrete elevation levels are specified (see the following section). For example, if discrete levels are specified for every z value in the cross-sectional shape description, then the areas computed using Equation 77 will be the same as the true areas. However, if too few discrete levels are specified to adequately represent the cross section, then the areas computed using Equation 77 will differ somewhat from the true areas.

The conveyance, K_{kj}, for subsection j at discrete level k is defined as follows:

$$K_{kj} = \begin{cases} A_{kj}\sqrt{\dfrac{8gA_{kj}}{P_{kj}f_j}} & \cdots \text{Darcy Weisbach} \\ \dfrac{CA_{kj}^{5/3}}{P_{kj}^{2/3}n_j} & \cdots \text{Manning} \end{cases} \tag{78}$$

in which A_{kj} = area of subsection j at discrete level k (this area is determined from the details of the cross section when subsections are defined and from Equation 77 when subsections are not defined), P_j = portion of the wetted perimeter at discrete level k that bounds subsection j, f_j = Darcy-Weisbach friction factor for subsection j, n_j = Manning coefficient for subsection j, g = acceleration of gravity, and C = units conversion factor equal to 1.486 for *English* units and 1.0 for *SI* units. The total conveyance for a section at discrete level k is simply the sum of the individual subsection conveyance values.

Once the parameter tables have been constructed, linear interpolation is used to obtain values for T(z), P(z), K(z), and $\beta(z)$ at elevation, z, in between the discrete tabulated elevations. In-between values of A are determined using Equation 77 with A_k, T_k, and z_k replaced by A(z), T(z), and z, respectively.

Discrete Elevation Levels

Two options are provided for defining the specific discrete elevation levels for which values of the geometric parameters are computed. The default option, which requires no specific input, is to use all of the z values included in the x-z or z-w coordinates that define the cross-sectional shape. The second option, which is illustrated in Figure 107, is to specify the number of even increments to use between the maximum and minimum z values in the x-z or z-w coordinate pairs. The increment size is computed as the maximum z value minus the minimum z value divided by the specified number of increments. In addition to the evenly spaced elevation levels, a small increment equal to the value of the *depth_min* keyword, which is specified in the **CONSTANTS* section of the input file, is used next to the channel bottom, unless the even increment size is smaller. As discussed above, when the extrapolation option is set to *constant* an additional elevation level is also added above the maximum z value.

Figure 107 illustrates the discrete elevation levels that result for the sample *riverine* cross section given in Figure 106 when ten even increments are specified and *depth_min* is set equal to 2.0 in the **CONSTANTS* section of the input file. Figure 107 also illustrates the relationship between the bottom elevation z_b and the input, or relative, elevations that describe the cross-sectional shape. It is necessary to point out that while specifying ten increments leads to an

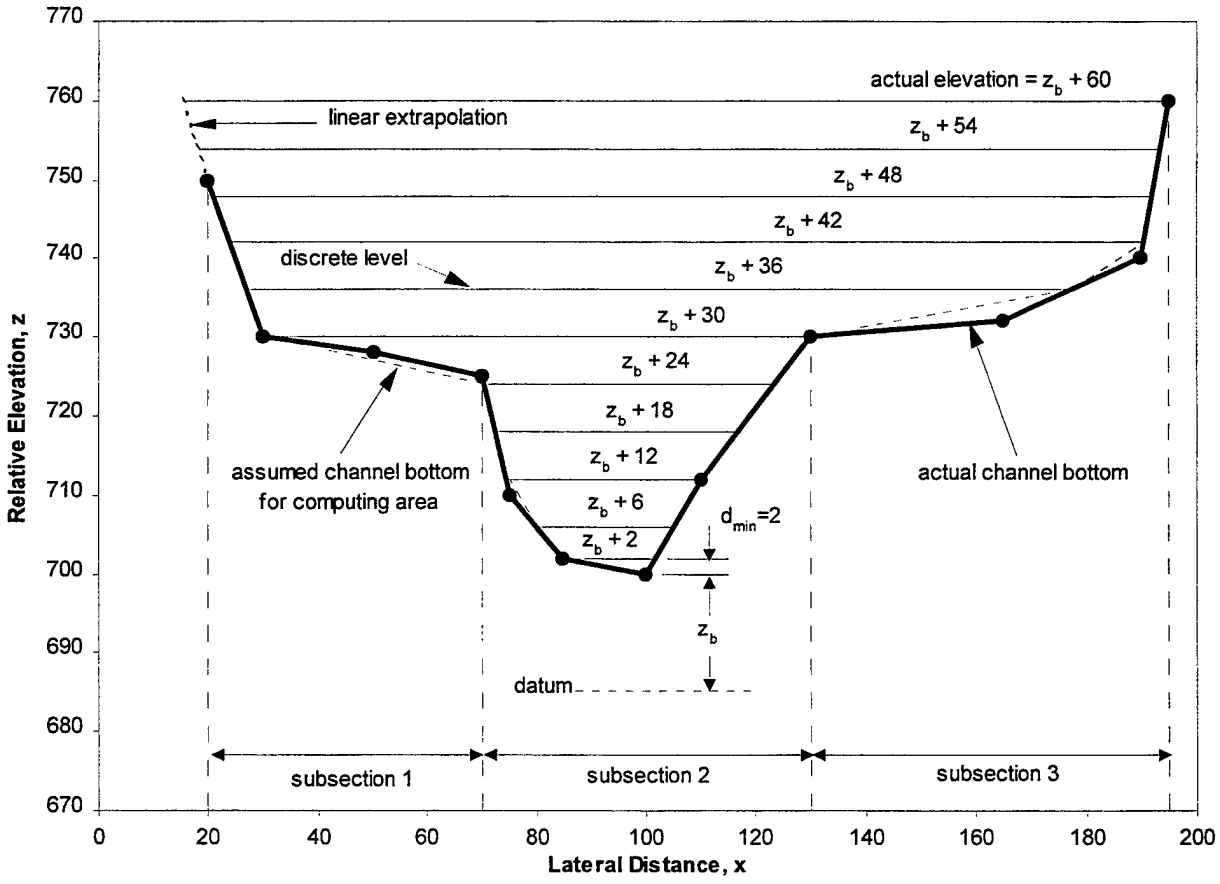


Figure 107: Discrete levels resulting from specification of ten elevation increments for sample *riverine* cross section

effective illustration, it is not really enough increments for this cross section, as indicated by the dashed lines showing the cross-sectional shape used for computing cross-sectional area. Similarly, the keyword *depth_min*, which defaults to 0.1, is normally set to a lower value than 2.0. The example input and example cross section output file provided below illustrate the differences in the geometric parameter tables that occur for this sample when all z values included in the cross-sectional shape definition are used as discrete elevation levels as opposed to specifying ten even increments.

The *z_increments* keyword is used to specify the number of even increments in which to divide the elevation range of a cross section. When the number of z values provided in the cross-sectional shape description is large, this keyword is used to limit the size of the parameter tables (25 to 50 increments are usually enough). When the number of z values provided in the cross-sectional shape description is small, this keyword is used to make sure that enough discrete elevation levels (usually 25 to 50) are included in the parameter tables to adequately represent the conveyance function. The conveyance function varies nonlinearly with elevation between coordinate points connected by straight lines, so linear interpolation can yield poor results if the discrete elevation levels are too far apart. Additional levels are not needed for the top width, wetted perimeter, and area functions, which are all exactly represented (under the assumption that coordinate points are connected by straight lines) by using only the input z values for the discrete levels. In fact, for these parameters, additional levels can lead to *less* accurate interpolated values, depending on how well the evenly spaced discrete levels describe the cross-sectional shape.

The *autofix_k's* keyword, which is defined in the **CONSTANTS* section of the input file, is a switch that may affect the tabulated conveyance values, the k's, for a *riverine* cross section. If the conveyance function does not monotonically increase with increasing water surface elevation (if the conveyance gradient, dK/dd, is negative for some d), solution convergence problems can occur. When *autofix_k's* is set to *on*, any decrease in the conveyance function for a riverine cross section as water surface elevation increases is automatically fixed by interpolation between the previous conveyance value and the next value that is larger. The tabulated results printed into the cross section file defined by *xsect_file* in the **CONSTANTS* section of the input file will include the interpolated values. When *autofix_k's* is set to *off*, a warning message is printed to the monitor and to the error file (LOCKSIM.MSG). In this case, the problem can be ignored in the hope that the solution will converge anyway, or the cross section parameters can be manually modified to ensure increasing conveyance with increasing water surface elevation. Setting of *autofix_k's* to *on* is highly recommended. The automatic interpolation is a reasonable and highly effective method of removing negative conveyance gradients from the cross section parameter tables.

Input

The following keywords are specific to describing *riverine* cross sections defined by *n* coordinate points and divided into *m* subsections:

Keyword	Default	Description
z_increments	none	number of evenly spaced discrete elevation levels
extrapolation	*linear*	*constant* or *linear*
subsections	no subsections	data cluster { x_{s1}, x_{s2}, x_{s3}, ..., $x_{s(m-1)}$ } (length)
xz_pairs	none	data cluster { x_1, z_1 x_2, z_2 x_3, z_3 ..... x_n, z_n } (length)
zw_pairs	none	data cluster { z_1, w_1 z_2, w_2 z_3, w_3 ..... z_n, w_n } (length)
x_values	none	data cluster { x_1, x_2, x_3, ..., x_n } (length)
z_values	none	data cluster { z_1, z_2, z_3, ..., z_n } (length)
w_values	none	data cluster { w_1, w_2, w_3, ..., w_n } (length)
xshift	0	shift applied to all x and x_s values (length)
xscale	1	scale applied to all x and x_s values
zshift	0	shift applied to all z values (length)
zscale	1	scale applied to all z values
wshift	0	shift applied to all w values (length)
wscale	1	scale applied to all w values

Figure 108 and Figure 109 illustrate the keyword requirements when x-z coordinate pairs and z-w coordinate pairs, respectively, are used to describe a *riverine* cross section. The value for *z_increments*, when it is specified, is usually between 25 and 50. *Subsections* may be specified only when x-z coordinate pairs are used to describe the cross-sectional shape. When *subsections* are defined, then friction must be specified (*DW_f* or *MN_n*), whether or not the cross section

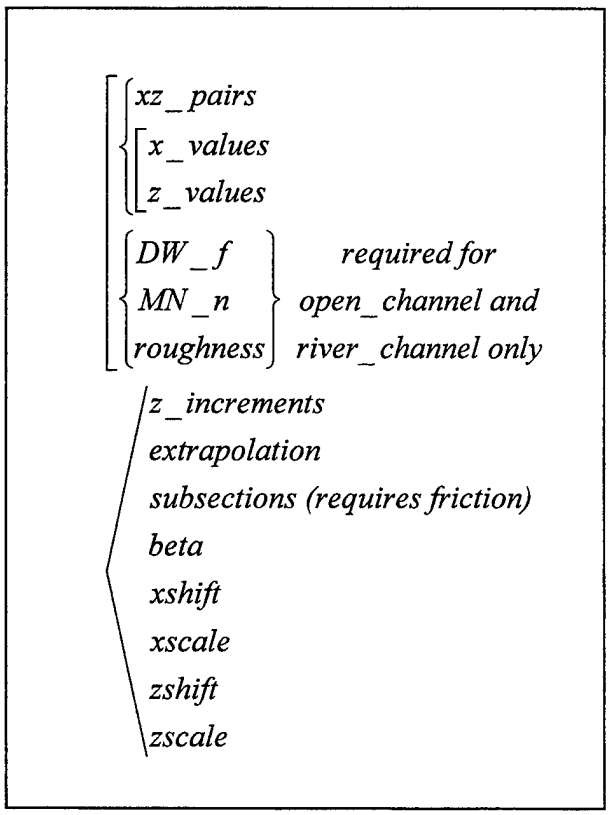


Figure 108: Keyword chart for *riverine* cross sections described using x-z coordinates

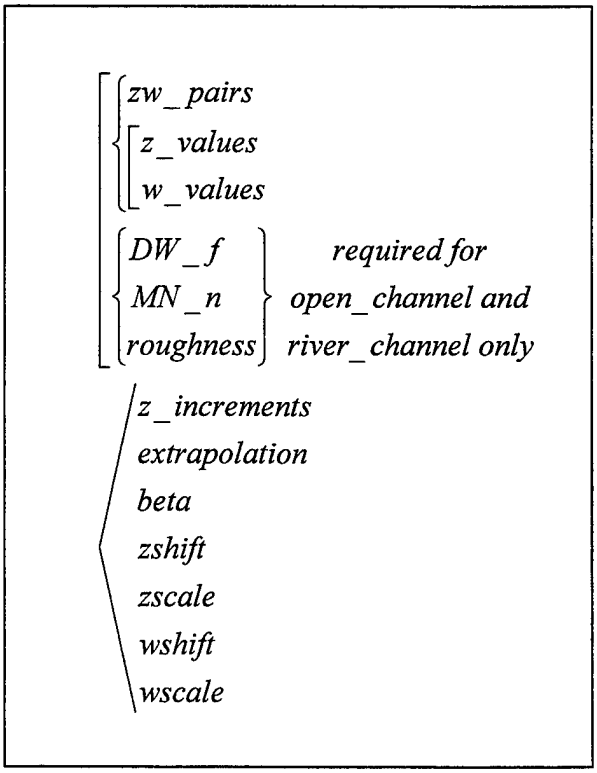


Figure 109: Keyword chart for *riverine* cross sections described using z-w coordinates

will be used by *open_channel* or *river_channel* components. As indicated above, the subsection and coordinate keywords all expect data clusters as input. Note that the *subsections* data cluster contains one less value than the number of subsections. Data clusters defining subsection values of *DW_f, MN_n,* or *beta* always contain the same number of values as the number of subsections. The shift and scale parameters affect the coordinate values in the same manner as described for functions in the *Functions section of this guide.

The following example input describes the sample *riverine* cross section illustrated in Figure 107, which has three subsections and is divided into ten equal elevation increments:

```
sample riverine z_increments=10
  x_values={  20    30    50    70    75    85    100   110
              130   165   190   195 }
  z_values={  750   730   728   725   710   702   700   712
              730   732   740   760 }

  subsections={  70  130 }
  mn_n={  .035  .026  .033 }
```

The following section shows the cross section output file for this input and compares it with output obtained when *z_increments* is not specified.

Cross Section Output File

The keyword *xsect_file*, which is specified in the **CONSTANTS* section of the input file, is used to specify the name of the file to which interpreted cross section information is written. The input for each cross section is echoed to this file followed by calculated properties based on the input. This information is most useful for *riverine* cross sections for which the calculated output is a table of top width, area, wetted perimeter, and conveyance for each increment of water surface elevation. The momentum correction coefficient is also included in the table when subsection-dependent values have been specified.

Figure 110 shows the contents of the cross section output file for the example input given above, with *z_increments=10*. For comparison, Figure 111 shows the cross section output for the same input with *z_increments* unspecified. Where elevations are the same, the top width, T, and wetted perimeter, P, columns are identical in the two outputs. The conveyance columns are also identical because, as mentioned above, when subsections are defined the area used to compute conveyance is based on the detailed cross-sectional shape rather than Equation 77. The area columns, based on Equation 77, are not identical because of the difference between the cross-sectional shape represented by the top width function and the actual cross-sectional shape.

```
========== CROSS SECTION: sample ==========

INPUT:
  sample   RIVERINE  z_increments=10 !
    x_values={ 20   30  50  70  75  85 100 110 130 165 190 195 }
    z_values={ 750 730 728 725 710 702 700 712 730 732 740 760 }
    subsections={     70     130    }
    mn_n={       .035    .026     .033  }

COMPUTED PROPERTIES:
  beta = 1.0000

      z(f)          T(f)       A(f^2)        P(f)       K(cfs)
    700.0000       0.0000       0.0000      0.0000       0.0000
    702.0000      16.6667      16.6667     17.7362     913.5217
    706.0000      25.0000     100.0000     29.3461   12937.2979
    712.0000      35.6667     282.0000     45.6677   55407.7734
    718.0000      44.3333     522.0000     60.9613  126286.9688
    724.0000      53.0000     814.0000     76.2550  227156.8438
    730.0000     100.0000    1273.0000    126.6016  379898.5000
    736.0000     150.5000    2024.5000    181.4913  663289.1250
    742.0000     166.5000    2975.5000    203.3855 1065409.3750
    748.0000     171.0000    3988.0000    216.2784 1571854.0000
    754.0000     175.5000    5027.5000    229.1712 2160553.7500
    760.0000     180.0000    6094.0000    242.0641 2822942.5000
```

Figure 110: Cross section output for *z_increments=10*

```
========== CROSS SECTION: sample ==========

INPUT:
  sample   RIVERINE
    x_values={ 20   30  50  70  75  85 100 110 130 165 190 195 }
    z_values={ 750 730 728 725 710 702 700 712 730 732 740 760 }
    subsections={     70     130    }
    mn_n={       .035    .026     .033  }

COMPUTED PROPERTIES:
  beta = 1.0000

      z(f)          T(f)       A(f^2)        P(f)       K(cfs)
    700.0000       0.0000       0.0000      0.0000       0.0000
    702.0000      16.6667      16.6667     17.7362     913.5217
    710.0000      33.3333     216.6667     40.9561   37582.4688
    712.0000      35.6667     285.6667     45.6677   55407.7734
    725.0000      54.4444     871.3889     78.8039  247095.9531
    728.0000      77.7778    1069.7222    103.5122  321312.2500
    730.0000     100.0000    1247.5000    126.6016  379898.5000
    732.0000     136.0000    1483.5000    163.8948  460009.2813
    740.0000     165.0000    2687.5000    199.0879  917213.6250
    750.0000     172.5000    4375.0000    220.5760 1759461.5000
    760.0000     180.0000    6137.5000    242.0641 2822942.5000
```

Figure 111: Cross section output with *z_increments* unspecified

*Plot_Variables

Results from unsteady simulations are most easily interpreted when presented graphically in plots. While LOCKSIM produces tabular output, it does not produce plots. LOCKSIM will, however, save data needed for plotting to an ASCII file in a columnar format that can be imported into an external plotting package or spreadsheet. The name of the file is specified using the *plot_file* keyword in the **CONSTANTS* section of the input file, or it may be entered interactively during the initial phase of a simulation. The variables to be saved during a simulation for later plotting are specified in the **PLOT_VARIABLES* section of the input file.

Format Specifications for Plot File

The *plot_labels* keyword specifies the format, either *column* or *row*, used to write the plot variable labels into the plot file. The *column* format is best for visual inspection using a text editor while the *row* format is best for loading the plot data into spreadsheet software. The *column* format is the default primarily because it was the only format supported in previous versions of LOCKSIM. The saved numerical data are written in the same format in either case.

The example in Figure 112 illustrates the format of the plot file into which data are saved when the *plot_labels* keyword is specified as *column*. At the top of the file is a numbered list of the saved variables, written in the format used to define them in the **PLOT_VARIABLES* section of the input file, as described below. The numbers assigned to each variable are used as reference labels in the first column heading row below. The second column heading row gives the units associated with each variable.

The example in Figure 113 illustrates the format of the plot file when the *plot_labels* keyword is specified as *row*. The units associated with each variable are listed in the first column heading row. The second column heading row lists the plot variable descriptions (in quotes), using the format used to define them in the *PLOT_VARIABLES section of the input file. In Figure 113, the second column heading line is truncated because there is not enough room on the page to show all of the plot variables. Note that the length of this heading line is not affected by the value assigned to the *plot_line* keyword (discussed below).

As indicated in Figure 112 and Figure 113, the first data column is always simulation time. The remaining data columns are the plot variables. The data format is controlled using two keywords in the **CONSTANTS* section of the input file. The keyword *plot_line* specifies the maximum length, in characters, of a single line of data in the plot file. The keyword *plot_field* specifies the width, in characters, reserved for each variable including a separating space. Consequently, output precision is controlled by the value of *plot_field* and the number of variable values per line is controlled by the value of *plot_line*. The number of plot variables per line is equal to the integer value of (*plot_line* / *plot_field* - 1), where one is subtracted for the time column. For the examples in Figure 112 and Figure 113, *plot_line* was set equal to 60 and *plot_field* was set equal to 8, resulting in 6 plot variables per line.

There are no limits, except those imposed by computer memory and hard disk space, to the number of plot variables that may be defined and saved during a simulation. As illustrated in Figure 112 and Figure 113, if *n* plot variables fit on one line, then the values of the first *n* variables are written in parallel columns, followed below by the values of the next *n* variables

```
PLOT VARIABLES LIST:
     (1) node5 node6 valve_position sectno=0
     (2) node5 head
     (3) node4 head
     (4) node3 head
     (5) node5 node6 discharge sectno=0
     (6) node4 node5 discharge sectno=0
     (7) node1 supply
     (8) node2 supply

Time     (1)      (2)      (3)      (4)      (5)      (6)
sec      (-)      (f)      (f)      (f)      (cfs)    (cfs)
0        1        112.502  119.153  143.829  1.15988  1.15988
0.5      0.75     114.332  119.153  143.829  1.14459  1.15988
1        0.5      119.082  119.153  143.829  1.10489  1.15988
1.5      0.25     137.109  119.153  143.829  .954459  1.15988
2        0        251.363  119.153  143.829  0        1.15988
2.5      0        251.53   119.153  143.829  0        1.15988
3        0        252.467  119.154  143.829  0        1.15988
3.5      0        252.626  121.507  143.829  0        1.15032
4        0        253.57   127.625  143.831  0        1.12544
4.5      0        253.729  151.373  145.148  0        1.02954
5        0        254.674  301.376  148.58   0        .421071
5.5      0        254.833  301.566  162.015  0        .421118
 .       .           .        .                .        .
 .       .           .        .                .        .
 .       .           .        .                .        .
200      0        159.301  160.233  153.533  0        -.00532


Time     (7)      (8)
sec      (cfs)    (cfs)
0        .517245  .642634
0.5      .517245  .642634
1        .517245  .642634
1.5      .517245  .642634
2        .517245  .642634
2.5      .517245  .642634
3        .517245  .642634
3.5      .517245  .642634
4        .517245  .642634
4.5      .517245  .642634
5        .517245  .642634
5.5      .517245  .642634
 .          .        .
 .          .        .
 .          .        .
200      .355751  -.36648
```

Figure 112: Sample plot file with *plotlabels=column*

written in parallel columns, followed below by the next *n* variables, and so on until all variables have been written. During a simulation the variables are all saved in dynamic arrays in computer memory. The plot file is not written to disk until the simulation is terminated. Consequently, a large number of plot variables during a long (several thousand time steps) simulation require a significant amount of computer memory.

The keywords *pv_start_time*, *pv_end_time*, and *pv_period* specified in the **CONSTANTS* section of the input file set default limits on the amount of data saved for all plot variables. (as described below, similar keywords specified for individual plot variables may be used to override the default values). Data will not be saved until *pv_start_time* is reached in the simulation. Data will no longer be saved after *pv_end_time* is reached in the simulation (the default value of -1 is interpreted as the end of the simulation). Data will be saved when the elapsed simulation time since the last save is greater than or equal to *pv_period.*

```
sec     (-)     (f)     (f)     (f)     (cfs)   (cfs)
Time "node5 node6 valve_position sectno=0" "node5 head" "node4 head". . . .
0       1       112.502 119.153 143.829 1.15988 1.15988
0.5     0.75    114.332 119.153 143.829 1.14459 1.15988
1       0.5     119.082 119.153 143.829 1.10489 1.15988
1.5     0.25    137.109 119.153 143.829 .954459 1.15988
2       0       251.363 119.153 143.829 0       1.15988
2.5     0       251.53  119.153 143.829 0       1.15988
3       0       252.467 119.154 143.829 0       1.15988
3.5     0       252.626 121.507 143.829 0       1.15032
4       0       253.57  127.625 143.831 0       1.12544
4.5     0       253.729 151.373 145.148 0       1.02954
5       0       254.674 301.376 148.58  0       .421071
5.5     0       254.833 301.566 162.015 0       .421118
 .      .          .       .                .      .
 .      .          .       .                .      .
 .      .          .       .                .      .
200     0       159.301 160.233 153.533 0       -.00532


sec     (cfs)   (cfs)
Time "node1 supply" "node2 supply"
0       .517245 .642634
0.5     .517245 .642634
1       .517245 .642634
1.5     .517245 .642634
2       .517245 .642634
2.5     .517245 .642634
3       .517245 .642634
3.5     .517245 .642634
4       .517245 .642634
4.5     .517245 .642634
5       .517245 .642634
5.5     .517245 .642634
 .         .       .
 .         .       .
 .         .       .
200     .355751 -.36648
```

Figure 113: Sample plot file with *plotlabels=row*

The plot file illustrated in Figure 113, with *plot_labels* specified as *row*, loads easily into spreadsheet software such as Microsoft Excel®. When using Excel, load the plot file as "delimited" with "space" as the delimiter. Then select row 2, choose "Cells" from the "Format" menu, and, under the "Alignment" tab, select "Wrap Text". The columns can then be plotted with row 2 as the "series names," or "legend labels," row.

Input Requirements for Plot Variables

Each plot variable specification consists of a *variable definition* optionally followed by keyword-value pairs as needed. A variable definition consists of a node or component designation, followed by a type identifier indicating the variable choice. Nodes are designated by their labels, which are assigned in the **NODES* section of the input file. Consequently, a plot variable specification for a node has the following form:

node_label variable_type keyword1=value1, keyword2=value2, ...

Components are designated by their upstream (us) and downstream (ds) nodes, in the same manner in which they are defined in the **COMPONENTS* section of the input file. Consequently, a plot variable specification for a component has the following form:

us_node_label ds_node_label variable_type keyword1=value1, keyword2=value2, ...

The two flow paths of a tee or manifold are specified as if they were two separate components. For a *diverging_tee* or *diverging_manifold* defined by *us_node ds_node1 ds_node2* in the **COMPONENTS* section of the input file, the following two specifications are valid:

us_node ds_node1 variable_type keyword1=value1, keyword2=value2, ...

us_node ds_node2 variable_type keyword1=value1, keyword2=value2, ...

For a *converging_tee* or *converging_manifold* defined by *usnode1 usnode2 ds_node* in the **COMPONENTS* section of the input file, the following two specifications are valid:

us_node1 ds_node variable_type keyword1=value1, keyword2=value2, ...

us_node2 ds_node variable_type keyword1=value1, keyword2=value2, ...

Available Keywords

The following optional keywords are available for plot variables:

Keyword	Default	Description
sectno	0	number of computational section for which output is desired (section 0 is upstream section)
sectlbl	*sectno*=0	label of computational section for which output is desired (river channels only)
start	0	simulation time to start saving data for this plot variable (time units)
end	-1	simulation time to stop saving data for this plot variable (time units)
period	0	simulation time between saves for this plot variable (time units)

Figure 114 and Figure 115 illustrate the keyword requirements for plot variables.

The *sectno* and *sectlbl* keywords are used for components only. Each component has a minimum of two computational sections, an upstream section next to the upstream node and a downstream section next to the downstream node. In addition, the *imp_pipe*, *moc_pipe*, *open_channel*, and *river_channel* components may have internal computational sections between the upstream and downstream sections. Component variables, such as discharge, head, pressure, and velocity, will in general have different values at each computational section. The *sectno* and *sectlbl* keywords are used to specify the computational section at which output is desired. The computational sections are numbered sequentially upstream to downstream, with the upstream section, which is the default section, designated as section 0 (zero). For *river_channel* components only, each computational section may optionally be assigned a label, which can then be specified using the *sectlbl* keyword as an alternative to specifying the *sectno* keyword.

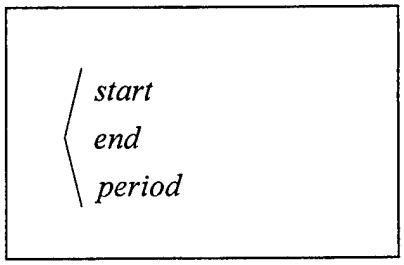


Figure 114: Keyword chart for node plot variable

As a convenience, an alternative method is available for specifying the upstream and downstream computational sections for components. The *variable_type* type identifier may be prefixed by "*us_*" or "*ds_*" to designate the upstream or downstream section, respectively. For example, the upstream discharge for a component may be specified as *us_discharge*.

Some component variables, for example *valve_position*, have only one value for the component rather than values for each computational section. These plot variables will return the same value regardless of how *sectno* or *sectlbl* are specified. As indicated in Figure 112 for *valve_position*, section-independent component variables are, nevertheless, associated with a computational section number in the plot file variable list.

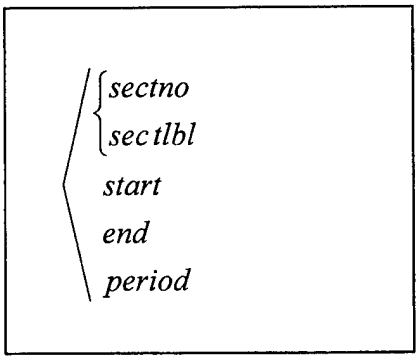


Figure 115: Keyword chart for component plot variable

The *start*, *end*, and *period* keywords are used to override, for a particular plot variable, the global values of *pv_start_time*, *pv_end_time*, and *pv_period,* assigned in the **CONSTANTS* section of the input file. The *start*, *end*, and *period* keywords should be used sparingly, to minimize the required computer memory for saved plot variables, if at all. A plot file containing data columns with different start and end times and different save periods, will have numerous blank fields in many columns. External plotting software and spreadsheets will frequently misinterpret the resulting plot file.

Available Variable Types

The acceptable type identifiers for *variable_type* are different for nodes than for components, for different components, and even for the different flow paths of a manifold. Table 14 and Table 15 define type identifiers available for all nodes and all components, respectively. Table 16 through Table 22 define additional type identifiers available for specific components. For additional details on the component plot variables, see the individual component descriptions in the *Components section of this guide.

Table 14: Plot Variables for Nodes

Identifier	Units	Description
demand	discharge	external discharge leaving the node and network
supply	discharge	external discharge entering the network at the node
head	length	hydraulic gradeline elevation
pressure	pressure	gage pressure
vapor	volume	volume of accumulated vapor
depth	length	water depth (hydraulic gradeline minus node elevation)

Table 15: Plot Variables for *All* Components

Identifier	Units	Description
discharge	discharge	component discharge at specified computational section
head	length	hydraulic gradeline elevation at specified section
energy	length	total energy (hydraulic gradeline plus velocity head) at section
pressure	pressure	gage pressure at specified section
All except *river_channel* and *open_channel*:		
vapor	volume	accumulated vapor volume at specified section

Table 16: Additional Plot Variables for *moc_pipe* and *imp_pipe*

Identifier	Units	Description
velocity	velocity	average cross-sectional velocity
re	none	Reynolds number
shear	pressure	wall shear stress
dwf	none	Darcy-Weisbach friction factor

Table 17: Additional Plot Variables for *open_channel* and *river_channel*

Identifier	Units	Description
velocity	velocity	average cross-sectional velocity
re	none	Reynolds number
shear	pressure	bed shear stress
dwf	none	Darcy-Weisbach friction factor
depth	length	water depth (hydraulic gradeline minus bed elevation)
fr	none	Froude number
crmin	none	Courant number = $\lvert V-c \rvert \ \Delta t/\Delta x$
crmax	none	Courant number = $\lvert V+c \rvert \ \Delta t/\Delta x$
qlateral	discharge	lateral inflow for reach between *sectno* and *sectno*+1
top_width	length	cross-sectional top width
area	area	cross-sectional area
hyd_depth	length	hydraulic depth = *area* / *top_width*

Table 18: Additional Plot Variables for *pipe_loss*

Identifier	Units	Description
velocity	velocity	average velocity at upstream or downstream section
re	none	Reynolds number at upstream or downstream section

Table 19: Additional Plot Variables for *valve* and *check_valve*

Identifier	Units	Description
velocity	velocity	average velocity at upstream or downstream section
re	none	Reynolds number at upstream or downstream section
valve_position	none	valve opening position; 0 (closed) or 1 (open) for *check_valve*

Table 20: Additional Plot Variables for *rev_tainter*

Identifier	Units	Description
velocity	velocity	water velocity at upstream or downstream section
re	none	Reynolds number at upstream or downstream section
valve_position	none	reverse tainter valve opening position (b/B)
cav_index	none	cavitation index (Eq. 34)
vena_pressure	pressure	gage pressure at top surface of vena contracta (Eq. 35)
vena_head	length	hydraulic gradeline at vena contracta (Eq. 33)

Table 21: Additional Plot Variables for *storage*

Identifier	Units	Description
ws_elevation	length	water surface elevation
over_flow	discharge	over flow out of *storage* component and out of network
in_flow	discharge	net flow into *storage* component from network
under_volume	volume	accumulated void volume from emptied *storage* component

Table 22: Additional Plot Variables for Tees and Manifolds

Identifier	Units	Description
Valid for the upstream and downstream ends of both flow paths through a tee or manifold:		
velocity	velocity	water velocity
re	none	Reynolds number
Valid only for the flow path through a manifold branch (not valid for tees):		
branch_flow	discharge	discharge through an individual branch (use *discharge* for total)

Example Input

The plot file example illustrated in Figure 112 and Figure 113 was generated by the following input:

```
*PLOT_VARIABLES
 node5 node6 valve_position
 node5 head
 node4 head
 node3 head
 node5 node6 discharge
 node4 node5 us_discharge
 node1 supply
 node2 supply
```

As demonstrated in this example, it is generally more convenient to accept the default section (upstream) or to use the *us_* or *ds_* prefix than to specify the computational section using *sectno* or *sectlbl*. These keywords typically are used only for components with internal sections.

REFERENCES

Ables, J. H., Jr., 1978, "Filling and Emptying System for Bay Springs Lock, Tennessee-Tombigbee Waterway, Mississippi," U. S. Army Engineer Waterways Experiment Station, Technical Report H-78-19.

Blevins, R. D., 1984, *Applied Fluid Dynamics Handbook*, Van Nostrand Reinhold.

Carnahan, B., H. A. Luther, and J. O. Wilkes, 1969, *Applied Numerical Methods*, John Wiley and Sons, New York, NY.

Chow, V. T., 1959, *Open-Channel Hydraulics*, McGraw-Hill.

Crane, 1969, *Flow of Fluids Through Valves, Fittings, and Pipe*, Technical Paper No. 410, Crane Company, Joliet, Illinois.

Cunge, J. A., F. M. Holly, Jr., and A. Verwey, 1980, *Practical Aspects of Computational River Hydraulics*, Pittman.

Elder, R. A., J. T. Price, and W. W. Engle, 1964, "Navigation Locks: TVA's Multiport Lock Filling and Emptying System," *J. of the Waterways and Harbors Division*, ASCE, Vol. 90, No. WW1, Proc. Paper 3788, pp. 31-46.

Garbrecht, J., and G. O. Brown, 1991, "Calculation of Total Conveyance in Natural Channels," *J. of Hydraulic Engineering*, ASCE, Vol. 117, No. 6, pp. 788-798.

Hebler, M. T., and F. M. Neilson, 1976, "Lock Filling and Emptying -- Symmetrical Systems," U. S. Army Engineer Waterways Experiment Station, Miscellaneous Paper H-76-13.

Henderson, F. M., 1966, *Open Channel Flow*, Macmillan.

Idelchik, I. E., 1986, *Handbook of Hydraulic Resistance*, 2nd edition, Hemisphere Publishing Co., Houston, Texas.

Kennison, H. F., 1956, "Surge-Wave Velocity -- Concrete Pressure Pipe," *Trans. ASME*, pp. 1323-1328.

Li, S. T., 1965, "Hydromechanics of Inland Navigation," *Advances in Hydroscience*, Vol. 2, edited by V. T. Chow, Academic Press, New York, NY, pp. 131-208.

McGee, R. G., 1989, "Prototype Evaluation of Bay Springs Lock, Tennessee-Tombigbee Waterway, Mississippi," U. S. Army Engineer Waterways Experiment Station, Technical Report HL-89-15.

Miller, D. S., 1990, *Internal Flow Systems*, 2nd edition, Gulf Publishing Co., Houston, Texas.

Miller, D. S. (editor), 1994, *Discharge Characteristics*, International Association for Hydraulic Research Hydraulic Structures Design Manual No. 8, Balkema, Rotterdam, Netherlands.

SAE (Society of Automotive Engineers), 1969, *SAE Aerospace Applied Thermodynamics Manual*, SAE Inc., New York, NY.

Schohl, G. A., 1978, "Model Test Results of Various Lock Design Modifications, Pickwick Landing 1000-Foot Lock," Tennessee Valley Authority, Water Systems Development Branch, Report No. WM28-2-4-100.

Schohl, G. A., 1989, "Transient Flow and Pressures Resulting from Emergency Closure of Empty Valves at Wheeler Main Lock," Tennessee Valley Authority, Engineering Laboratory, Report No. WR28-1-3-103.

Schohl, G. A., 1992, "Numerical Model for Navigation Lock Filling and Emptying Systems," Tennessee Valley Authority, Engineering Laboratory, Report No. WR28-1-720-100.

Schohl, G. A., 1994, "Comparisons of Field Data from Wheeler Main Lock with Predictions from a Numerical Model," Tennessee Valley Authority, Hydraulic Engineering, Engineering Laboratory, Report No. WR28-1-3-107.

Schohl, G. A., J. D. Hubble, D. J. Benton, and C. F. Bowman, 1995, "EZFLOW, a Quality Assured Intuitive Computer Code for Hydraulic Analysis of Pipe Networks," International Joint Power Generation Conference and Exposition, ASME, Minneapolis, MN, Oct. 9-11.

Stockstill, R. L., F. M. Neilson, and V. L. Zitta, 1991, "Hydraulic Calculations for Flow in Lock Manifolds," *J. of Hydraulic Engineering*, ASCE, Vol. 117, No. 8, pp. 1026-1041.

Streeter, V. L., and E. B. Wylie, 1979, *Fluid Mechanics*, 7th edition, McGraw-Hill.

USACE (U. S. Army Engineer Waterways Experiment Station), 1975, "Hydraulic Design of Lock Culvert Valves," EM 1110-2-1610.

USACE (U. S. Army Engineer Waterways Experiment Station), 1979, "Hydraulic Design of Navigation Locks," EM 1110-2-1604 (Draft).

USACE (U. S. Army Engineer Waterways Experiment Station), 1988, "Hydraulic Design Criteria," Eighteenth issue, Vicksburg, MS.

Vazsonyi, A., 1944, "Pressure Loss in Elbows and Duct Branches," *Trans. ASME*, Vol. 66, pp. 177-183.

Wylie, E. B., and V. L. Streeter, 1993, *Fluid Transients in Systems*, Prentice-Hall.

Zigrang, D. J., and N. D. Sylvester, 1982, "Explicit Approximations to the Solution of Colebrook's Friction Factor Equation," *AIChE Journal* ,The American Institute of Chemical Engineers, Vol. 28, No. 3, pp. 514-515.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1999	Final report

4. TITLE AND SUBTITLE
User's Manual for LOCKSIM: Hydraulic Simulation of Navigation Lock Filling and Emptying Systems

5. FUNDING NUMBERS

6. AUTHOR(S)
Gerald A. Schohl

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Tennessee Valley Authority
Engineering Laboratory
129 Pine Road, Norris, TN 37828

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Corps of Engineers
Washington, DC 20314-1000;
U.S. Army Engineer Waterways Experiment Station
3909 Halls Ferry Road, Vicksburg, MS 39180-6199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
Contract Report CHL-99-1

11. SUPPLEMENTARY NOTES
Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

LOCKSIM (LOCK SIMulator) is a numerical model for simulation of one-dimensional transient filling and emptying flow in navigation locks. Part 1 of this user's manual describes the application of LOCKSIM to navigation locks and provides two detailed example applications. Part 2 provides detailed descriptions for operating LOCKSIM, interpreting its output, and specifying its input.

LOCKSIM was designed as a general-purpose simulator, applicable to nearly any type of filling and emptying system. In LOCKSIM, a filling and emptying system is represented by a network consisting of closed conduit and open channel components arranged in any desired combination. The geometry, hydraulic characteristics, and boundary conditions of the network are described in an ASCII input file, which is read by LOCKSIM at the start of each simulation. LOCKSIM is operated interactively, allowing the user to examine results, change parameters, and decide whether to quit or continue at any point during a simulation. LOCKSIM's primary output is custom-specified by the user and easily imported into spreadsheet software for futher analysis and plotting.

Distinguishing technical features of LOCKSIM include prediction of longitudinal hawser forces in a lock chamber, prediction of cavitation index, and minimum pressure downstream from reverse tainter valves, rigorous treatment of

(Continued)

14. SUBJECT TERMS

Closed conduit	Navigation
Locks (waterways)	Numerical model
Manifold	Open channel

15. NUMBER OF PAGES
190

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED		

13. (Concluded).

dividing and combining flows through tees and manifolds, and capability of including upstream and downstream approach channels in models of filling and emptying systems.