



Waterways Experiment Station

HIVEL 2D v2.0 *Users Manual*

Hivel 2D is a product of the Coastal and Hydraulics Laboratory (CHL) of Waterways Experiment Station.

Last Revision: March 1, 1997

*The contents of this report are not to be used for advertising, publication or promotional purposes.
Citation of trade names does not constitute an official endorsement or approval of the use of such
commercial products.*

Contents

1—Introduction	1
Background	1
Purpose and Scope.....	1
2—HIVEL2D Overview	3
Governing Equations	3
Finite Element Model	6
Petrov-Galerkin Test Function	7
Shock Capturing	7
Temporal Derivatives	7
Solution of the Nonlinear Equations.....	8
Model Features	8
3—Developing an Application.....	11
Grid Generation	11
Hydrodynamic Input.....	12
Hot Start File	13
Running the Model	14
Hivel Interface.....	16
Super File	17
Hot Start	18
Model Control	18
Boundary Conditions.....	19
Model Checker	21
Run Hivel	22
Output Files	23
References.....	25

Appendix A: File Formats.....	27
HIVEL2D Super File.....	27
2D Geometry File.....	29
Hydrodynamic Flow File.....	31
Output Files.....	38
Appendix B: Boundary Conditions.....	41
Flow In and Out.....	41
Flow In, No Flow Out.....	43
No Flow In and Flow Out.....	43
No Flow In or Out.....	44
Appendix C: Troubleshooting.....	45
Appendix D: Example Problem.....	47
Appendix E: Interpolation Program.....	53
Program Description.....	53
Interpolation Scheme for Quadrilateral Elements.....	54
Interpolation Scheme for Triangular Elements.....	55
Interpolation.....	57

*The contents of this report are not to be used for advertising, publication or promotional purposes.
Citation of trade names does not constitute an official endorsement or approval of the use of such
commercial products.*

*The contents of this report are not to be used for advertising, publication or promotional purposes.
Citation of trade names does not constitute an official endorsement or approval of the use of such
commercial products.*

1 Introduction

Background

HIVEL2D is a free-surface, depth-averaged, two-dimensional model designed specifically for flow fields that contain supercritical and subcritical regimes as well as the transitions between the regimes. The model provides numerically stable solutions of advection-dominated flow fields containing shocks such as oblique standing waves and hydraulic jumps.

HIVEL2D has been verified by comparing computed model results with laboratory data. The findings of these tests are presented in Stockstill and Berger (1994).

Purpose and Scope

This report serves as a user's manual for HIVEL2D Version 2.0. This user friendly version of the program has been incorporated into SMS (Brigham Young University, 1997). First, a brief description of the model equations are presented. Next, step-by-step instructions are given to guide the user in creating the required input files. These input files include the geometry (numerical model computational mesh) file, the hydrodynamic parameters/boundary conditions file, and the initial conditions file. Finally, the numerical model output file is explained. Appendix A describes the file formats. Appendix B describes the boundary conditions. Appendix C describes problems with the model and solutions. Appendix D gives an example problem. Appendix E discusses the interpolation program.

2 HIVEL2D Overview

HIVEL2D (Stockstill and Berger 1994) is designed to simulate flow typical in high-velocity channels. The model is a finite element description of the two-dimensional shallow-water equations in conservative form. The model does not include Coriolis or wind effects as these are typically not important in high-velocity channels.

Governing Equations

Vertical integration of the equations of mass and momentum conservation for incompressible flow with the assumption that vertical accelerations are negligible compared to the acceleration of gravity results in the governing equations commonly referred to as the shallow-water equations. The dependent variables of the two-dimensional fluid motion are defined by the flow depth, h , the volumetric discharge per unit width in the x -direction, p , and the volumetric discharge per unit width in the y -direction, q . These variables are functions of the independent variables x and y , the two space directions, and time t . The shallow-water equations in conservative form are given as (Abbott 1979):

$$\frac{\partial Q}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + H = 0 \quad (1)$$

where

$$Q = \begin{pmatrix} h \\ p \\ q \end{pmatrix} \quad (2)$$

$$F_x = \begin{bmatrix} p \\ \frac{p^2}{h} + \frac{1}{2}gh^2 - \frac{h\sigma_{xx}}{\rho} \\ \frac{pq}{h} - \frac{h\sigma_{yx}}{\rho} \end{bmatrix} \quad (3)$$

$$F_y = \begin{bmatrix} q \\ \frac{pq}{h} - \frac{h\sigma_{xy}}{\rho} \\ \frac{q^2}{h} + \frac{1}{2}gh^2 - \frac{h\sigma_{yy}}{\rho} \end{bmatrix} \quad (4)$$

$$H = \begin{bmatrix} 0 \\ gh \frac{\partial z_0}{\partial x} + g \frac{n^2 p \sqrt{p^2 + q^2}}{C_0^2 h^{\frac{7}{3}}} \\ gh \frac{\partial z_0}{\partial y} + g \frac{n^2 q \sqrt{p^2 + q^2}}{C_0^2 h^{\frac{7}{3}}} \end{bmatrix} \quad (5)$$

where

$p = uh$, u being the depth-averaged x-direction component of velocity

$q = vh$, v being the depth-averaged y-direction component of velocity

g = acceleration due to gravity

$\sigma_{xx}, \sigma_{xy}, \sigma_{yx}, \sigma_{yy}$ = Reynolds stresses where the first subscript indicates the direction and the second indicates the axis direction normal to the face on which the stress acts

ρ = fluid density

z_0 = channel invert elevation

n = Manning's roughness coefficient

C_0 = dimensional constant ($C_0 = 1$ for SI units and $C_0 = 1.486$ for English units)

The Reynolds stresses are determined using the Boussinesq approach relating stress to the gradient in the mean currents:

$$\begin{aligned}\sigma_{xx} &= 2\rho\nu_t\frac{\partial u}{\partial x} \\ \sigma_{xy} = \sigma_{yx} &= \rho\nu_t\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \sigma_{yy} &= 2\rho\nu_t\frac{\partial v}{\partial y}\end{aligned}\quad (6)$$

where ν_t is the eddy viscosity, which varies spatially and is solved empirically as a function of local flow variables (Rodi 1980):

$$\nu_t = \frac{Cn\sqrt{8g}\sqrt{p^2 + q^2}}{C_0 h^{1/6}}\quad (7)$$

where C is a coefficient that varies between 0.1 and 1.0.

The typical high-velocity channel is rather wide and shallow, so that the bulk of the turbulence is bed-generated and therefore Equation 7 is reasonable. However, if the walls or other structures make a significant contribution to the overall turbulence, this equation will underpredict the magnitude of the eddy viscosity or turbulence. Even under typical channel configurations there is fairly large uncertainty in the parameter C . This is apparent since it has been found to range from 0.1 to 1.0. Therefore, the user should consider making a sensitivity check on the calculated water surface for a range of values of C and any other parameters having values that are largely uncertain.

Finite Element Model

This system of partial differential equations is solved using the finite element method. The finite element approach taken is a Petrov-Galerkin formulation that incorporates a combination of the Galerkin test function and a non Galerkin component to control oscillations due to convection. An integration by parts procedure is used to develop the weak form of the equations. The weak form which facilitates the specification of boundary conditions is:

$$\sum_e \left[\int_{\Omega_e} \left(\psi_i \frac{\partial Q}{\partial t} - \frac{\partial \phi_i}{\partial x} F_x - \frac{\partial \phi_i}{\partial y} F_y + \phi_i A \frac{\partial Q}{\partial x} \right. \right. \quad (8)$$

$$\left. \left. B \frac{\partial Q}{\partial y} + \psi_i H \right) d\Omega_e + \int_{\Gamma_e} \phi_i (F_x n_x + F_y n_y) dl \right] = 0$$

where the variables are understood to be discrete values and

e = subscript indicating a particular element

Ω = domain

$\psi_i = \phi_i I + \varphi_i$ = test function

ϕ_i = Galerkin part of the test function

I = identity matrix

φ_i = non-Galerkin part of the test function

$(n_x, n_y) = \hat{n}$ = unit vector outward normal to the boundary Γ_e

and

$$A = \frac{\partial F_x}{\partial Q} \quad (9)$$

$$B = \frac{\partial F_y}{\partial Q}$$

Natural boundary conditions are applied to the sidewall boundaries through the weak form. The side wall boundaries are “no-flux” boundaries; that is, there is no net flux of mass or momentum through these boundaries. These boundary conditions are enforced through the line integral in the weak form.

Petrov-Galerkin Test Function

The Petrov-Galerkin test function y_i is defined (Berger 1993) as:

$$\varphi_i = \phi_i I + \varphi_i \quad (10)$$

$$\varphi_i = \beta \left(\Delta x \frac{\partial \phi_i}{\partial x} \hat{A} + \Delta y \frac{\partial \phi_i}{\partial y} \hat{B} \right) \quad (11)$$

where β is a dissipation coefficient varying in value from 0 to 0.5, the Δ terms are the linear basis functions, and Δx and Δy are the grid intervals. A detailed explanation of this test function, in particular \hat{A} and \hat{B} , is given in Berger (1993).

Shock Capturing

The coefficient β scales the dissipation needed for numerical stability. More dissipation is needed in the vicinity of shocks such as hydraulic jumps than in smooth regions of the flow field. Because a lower value of β ($\beta = \beta_{SM}$) is more precise, a large value of β ($\beta = \beta_{SH} = 0.5$ where β_{SM} and β_{SH} are the Petrov-Galerkin parameters for smooth flow and for shocks, respectively) is applied only in regions in which it is needed. HIVEL2D employs a mechanism that detects shocks and increases β automatically. Therefore, β_{SH} is implemented only when needed as determined by evaluation of the element energy deviation. In a similar manner, the eddy viscosity coefficient C varies from C_{SM} to C_{SH} , the effect being that eddy viscosity is increased only in areas of greatest element energy deviation.

Temporal Derivatives

A finite difference expression is used for the temporal derivatives. The general expression for the temporal derivative of a variable Q_j is:

$$\left(\frac{\partial Q_j}{\partial t}\right)^{m+1} \approx \left(\frac{1+\alpha}{2}\right)\left(\frac{Q_j^{m+1} - Q_j^m}{t^{m+1} - t^m}\right) + \left(\frac{1-\alpha}{2}\right)\left(\frac{Q_j^m - Q_j^{m-1}}{t^m - t^{m-1}}\right) \quad (12)$$

where j is the nodal location and m is the time-step. An α equal to 1 results in a first order backward difference approximation, and an α equal to 2 results in a second-order backward difference approximation of the temporal derivative.

Solution of the Nonlinear Equations

The system of nonlinear equations is solved using the Newton-Raphson iterative method. Let \mathbf{R}_i be a vector of the nonlinear equations computed using a particular test function ψ_i and using an assumed value of \mathbf{Q}_j . \mathbf{R}_i is the residual error for a particular test function i . Subsequently, \mathbf{R}_i is forced toward zero as:

$$\frac{\partial R_i^k}{\partial Q_j^k} \Delta Q_j^k = -R_i^k \quad (13)$$

where the derivatives composing the Jacobian are determined analytically and k is the iteration number. This system of equations is solved for $\Delta \mathbf{Q}_j^k$ and then an improved estimate for \mathbf{Q}_j^{k+1} is obtained from:

$$Q_j^{k+1} = Q_j^k + \Delta Q_j^k \quad (14)$$

This procedure is continued until convergence to an acceptable residual error is obtained.

Model Features

Particular features of HIVE2D Version 2.0 are as follows:

- a. Combinations of linear-based triangular and rectilinear shape functions are used to represent p , q , and h .

- b.* The model is compiled to solve problems having meshes as large as 2000 nodes and 2000 elements and will run on a IBM-compatible personal computer having at least 8 MB of RAM.
- c.* A Newton-Raphson iteration is used within each time-step.
- d.* The mild slope assumption is invoked. This means the slope should be less than about 0.05. This assumes that the slope is geometrically mild, not hydraulically mild.
- e.* Boundary conditions are specified for any combination of supercritical and subcritical inflow and outflow.
- f.* Boundary conditions are constant over the simulation period.
- g.* Boundary roughness (Manning's n) is specified on an element type basis.
- h.* A user-specified parameter α produces either first-order or second-order backward temporal differences.
- i.* A Petrov-Galerkin approach in which the test function is weighted upwind along characteristics is employed. The degree of upwinding and thus stability is determined by the parameters β_{SM} and β_{SH} . Defaults are $\beta_{SM} = 0.25$ and $\beta_{SH} = 0.5$.
- j.* A shock detection mechanism based upon the energy variation per element is used to invoke β_{SH} .
- k.* Turbulent eddy viscosity is calculated based upon simple user-specified parameters C_{SM} and C_{SH} using velocity, depth, and roughness.

3 Developing an Application

The basic steps to developing an application of HIVEL2D are as follows:

- a. Generate the grid:
 - (1) Number the grid intelligently.
 - (2) Identify inflows and outflows.
- b. Develop a hydrodynamic input file.
- c. Develop a reasonable hot start file.
- d. Run the model (probably several times). If necessary, refine the grid and interpolate a new hot start file using INTRPL8.EXE (Appendix D).
- e. View the output files in Surface Water Modeling System (SMS) or in any other graphical program.
- f. Examine the solution for reasonableness.

Hivel 2D (version 2.0) uses SMS (Brigham Young University, 1997) as the pre- and postprocessor for programs involving two- dimensional finite element meshes. It was developed by the Engineering Computer Graphics Laboratory at Brigham Young University for the U.S. Army Engineer Waterways Experiment Station.

Grid Generation

To a large degree the quality of the grid determines the accuracy and stability of the model. The first step in grid generation is getting the necessary geometry information into the grid generator. Critical elements such as points on transitions, bridge piers, and curves should be put into the grid generator. One method involves drawing the structure to be modeled in three dimensions in a CAD program, and then extracting the critical points. These critical points are then put into a grid generator.

Remember the following general rules when generating the grid:

- a. HIVEL2D uses linear elements. Therefore, when generating a HIVEL2D grid, use only four-node quadrilaterals and three-node triangles.
- b. Keep the element aspect ratio less than 3:1 (the closer to 1:1, the better). The aspect ratio is the ratio of the longest element dimension to the shortest, i.e., the length-to-width ratio.
- c. Use gradual transitions in element size. Generally, an element's area should not be greater than 1½ times its smallest neighbor.
- d. Include at least five or six elements across a channel in the area of interest. If, for example, the channel has an island in the center, this resolution is needed on both sides of the island. Also, increase the resolution around grade breaks and wall transitions.

Once the grid generation is complete, be sure to renumber the grid using SMS renumber options. The best numbering scheme will give the smallest bandwidth. Normally, the nodes need to be numbered progressively across the narrowest dimension of the grid. This minimizes the bandwidth, which makes HIVEL2D run more efficiently.

Keep in mind that the model is expecting a relatively narrow (computationally) dimension across the channel. All input is free-field. The various lines can appear in any order. Any prefixes on each line that are not required by HIVEL2D are printed to the screen, but are otherwise ignored by the program.

Table 1 shows the geometry file format. If the geometry was generated using SMS, then the E3, E4 and ND lines are transparent to the user. To generate a grid from scratch, be sure to follow the correct format shown in Figure A4 (Appendix A).

Hydrodynamic Input

Starting the model is usually the most difficult task. The model should be started with small time-steps (i.e., a CFL number less than 1) where:

$$\text{CFL} = \frac{|U| + \sqrt{gh}}{\Delta l / \Delta t} \quad (15)$$

where

$$|U| = \sqrt{u^2 + v^2}$$

and

Δl = the element length

Δt = the time-step size

The time-step can then be gradually increased. If the steady-state solution is desired, then a fairly large time-step can be used. If the interest is in unsteady results, then, for accuracy, the CFL number should be held to a maximum of 2.

Appropriate boundary conditions must be supplied. For an inflow boundary condition when the flow is supercritical, the user must specify both x- and y-components of flow along with the depth. If the flow is subcritical here, then only the x- and y-components of flow are read. If the flow is subcritical at the outflow boundary, then the tailwater elevation must be specified. If the flow is supercritical at the outflow, then no boundary conditions need to be given. This is done in the input file by specifying a tailwater elevation that is lower than the bed elevation at the outflow boundary.

Starting the model is sometimes difficult when the flow is supposed to be supercritical at the outflow boundary. The model attempts to converge to a solution, but it may not be the desired solution. One way to avoid this is to start the model with a tailwater that is slightly subcritical (if the starting conditions are not known). Then after the model settles down, the boundary condition can be changed to supercritical. This takes a little experience.

Users should note that the Manning's n applies to each element type as well as the adjoining sidewalls. It is also used in conjunction with velocity, depth, and C to determine a turbulent eddy viscosity estimate.

The input to the hydrodynamics file is made through the list in Figure A5 (Appendix A). All variables are real except those beginning with I , J , K , L , M , or N , which are integer variables. All input is free-field. Each variable is described in Table 2 in the order in which it appears in the hydrodynamic input file. Units are designated by L and T , for length and time, respectively. If no units are specified, the variable is dimensionless.

Hot Start File

This file is the last two time-steps of information from the previous model run. Since the temporal derivative can be a second-order backward difference, two time-steps of old information are needed. If this is the initial run, then this information will have to be

generated based upon experience. One method is to start the model with zero velocity and a constant depth. The data are read as free-field.

This file will be overwritten by the model run, so if there is a chance that this information will be needed later, save a copy. If the model run finishes but the data are bad for some reason, the hot start will be bad as well. Again, this is a good reason to save a copy. The information in the hotstart file is:

(Line 1) TIME

(Line 2) $p^m, q^m, h^m, p^{m-1}, q^{m-1}, h^{m-1}$ (for each node, in order)

where

TIME = the time associated with values at time-step m

m = the last time-step

$m-1$ = the next to the last time-step

Remember that this file is free format.

The initial hot start file can also cause starting a run to be tricky. If the interest is in steady-state conditions, the accuracy of the initial guess will determine how long it will take to reach steady state. If the geometry is simple, then there is no trouble in guessing the discharge components. However, if it is complicated, it may be necessary to resort to zero discharge and a specified depth, which is a typical method. If the walls are sloping, there may be some difficulty in specifying the depth as well. It is fairly simple to write a program to subtract the bed elevation from some set of water-surface elevations. Typically, the user should set the old and the even older time-step information in this file as identical.

Running the Model

In this section a general overview for running HIVEL2D is described. The HIVEL interface in SMS is described in the following section. To run the model, the user must supply a super file which includes five file names, three input and two output. The file prefixes follow the name of the super file. HIVEL2D is developed and run within SMS.

- a. The executable of the model is called HIVEL2D and a copy of it should be in the users working directory.
- b. The model application and files are developed, launched and viewed within SMS.

- c. Each file name can be up to 15 characters for UNIX and IBM compatibles (e.g., EXAMPLE1.SUP).
- d. The file extensions of *SUP*, *GEO*, *FLO*, *HOT*, and *DAT* are ones that are typically used to designate files, but these suffixes are arbitrary.
- e. The geometry file contains the node locations and the element connection table.
- f. The hydrodynamic and computation parameter input file contains the hydraulic information about the run, the designation of boundary inflow and outflow nodes as well as the computation parameters. Examples of computation parameters are the specific boundary values of depth and flow rates, roughness, time-step size, number of iterations, etc.
- g. The output files contains the water-surface elevation and velocity fields for all nodes at each time location designated in the hydrodynamic input file. This information can then be converted to a form that can be viewed in SMS.
- h. The hot start file contains the time, p , q , and depth for the last two time-steps. This is used to start the calculation. This file is overwritten by each successful run; so, if it might be needed later, make a copy of it under another name before running the program.

After the file names are input, the program will begin running and several banners will appear on the screen.

- a. First, the program informs the user of any unused lines in the geometry file and prints the contents of the unused lines.
- b. Next, the contents of the hydrodynamic input file are displayed on the screen to allow the user to check the values that were read.
- c. If the coefficients C_{SM} , C_{SH} , or α are out of range, the program will report the error and halt.
- d. Then, as the program runs, the results for each time-step are displayed. These results include the number of iterations required, the maximum residual error and the node with which it is associated, the average energy, the minimum vorticity and the element where it occurred, and the maximum vorticity and the element where it occurred.

The following example illustrates the model development, launching and post processing of the model results using SMS. SMS can be launched by double clicking the *sms* icon found in the computer. The users are advised to read the SMS manual for installation and applications.

HIVEL Interface

SMS (version 5.0) includes a graphical interface to HIVEL2D (*fig. 3.1*). Several steps are involved in performing a HIVEL simulation within SMS. These steps are listed here to provide a summary of the overall process. Some of the steps are described in more detail in the following sections. The steps are as follows:

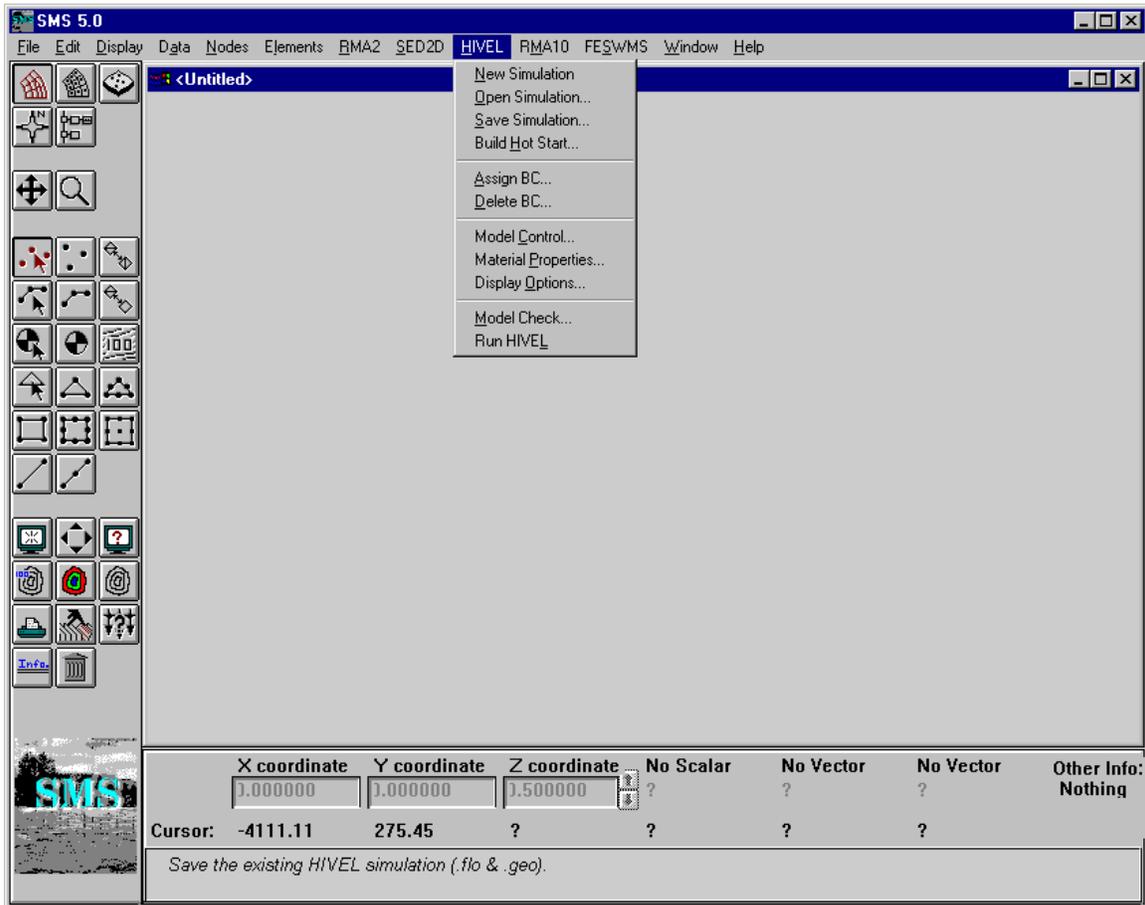


Figure 3.1 The SMS interface.

1. Create a 2D finite element grid to be modeled. The tools provided in SMS for constructing a 2D finite element grid are described in SMS users manual. This

chapter should be read entirely before generating a 2D grid for use with HIVEL2D.

2. Use the commands in the HIVEL menu to define the material properties, assign the boundary conditions, and to enter the other model parameters.
3. Run the HIVEL *Model Checker* by selecting the *Model Check* command. Fix any errors that are found in your model.
4. Select the *Save Simulation* command to save the model to disk.
5. Select the *Run HIVEL* command to run the simulation.
6. Select the *Data Browser* command from the *Data* menu and use the *Import* button to read in the solution files generated by HIVEL.
7. Use the *SMS* post-processing tools such as contours and velocity vectors to view the solution

Super File

A HIVEL2D super file is a special type of file that organizes the set of files used in a simulation. The names of all of the input and output files associated with a simulation are saved in the super file. When HIVEL2D is executed, it prompts the user for the name of the super file.

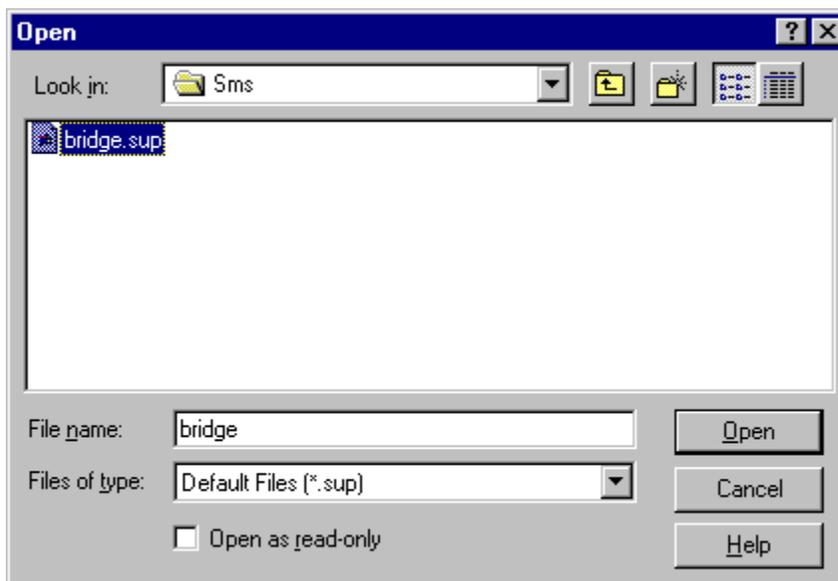


Figure 3.2 Open Simulation

The super file provides HIVEL2D with all of the file names involved in the simulation and no more prompts are necessary. Likewise, an entire simulation can be imported to SMS at once by opening the super file, using the *open simulation* dialog (fig 3.2). The super file name and full path are specified by selecting the *Super File* button at the top of the *Save Simulation* dialog which brings up the *File Browser*. The name and location (path) of the super file are specified. The super file and all of the other HIVEL2D input files are stored in the selected directory (fig. 3.3).

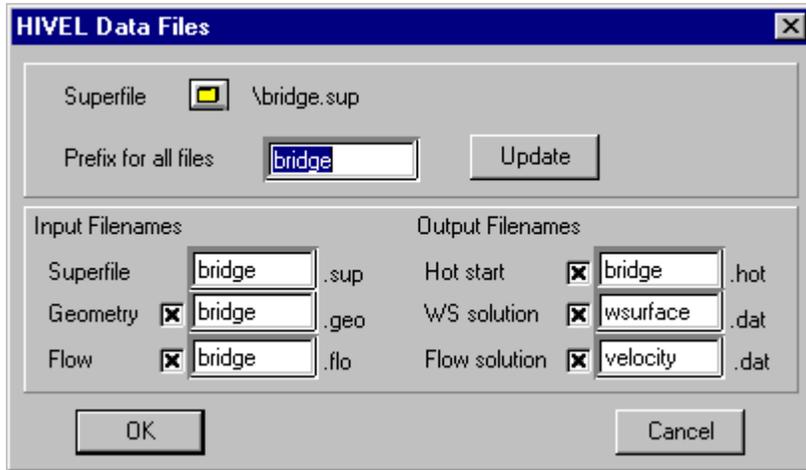


Figure 3.3 Save Simulation

Hot Start

The hot start file necessary for the simulation is created using the *Build Hot Start* dialog (fig 3.4).

Model Control

The hydrodynamic and computation parameter input file contains the hydraulic information about the run, the designation of boundary inflow and outflow nodes as well as the computation parameters. The dialog *Model control* (fig. 3.5) assigns the various hydrodynamic and computation parameter for the model run.

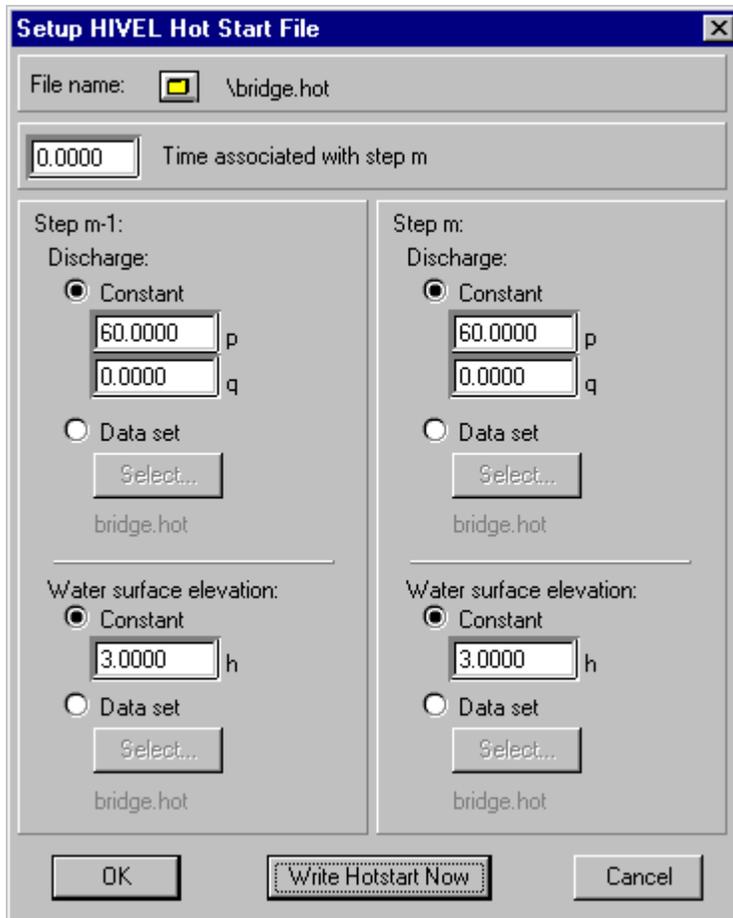


Figure 3.4 Hot Start

Boundary Conditions

The inflow Boundary conditions can be assigned using the *Assign Boundary Conditions* dialog, by selecting a specific node for assigning nodal boundary conditions (fig 3.6) or by selecting a node string for assigning the node string boundary conditions (fig 3.7). The outflow boundary conditions are assigned only using the node string option (fig 3.7). A detailed description of the boundary conditions are given in Appendix B. Existing nodal or node string boundary conditions can be deleted using the *Delete Boundary Conditions* dialog.

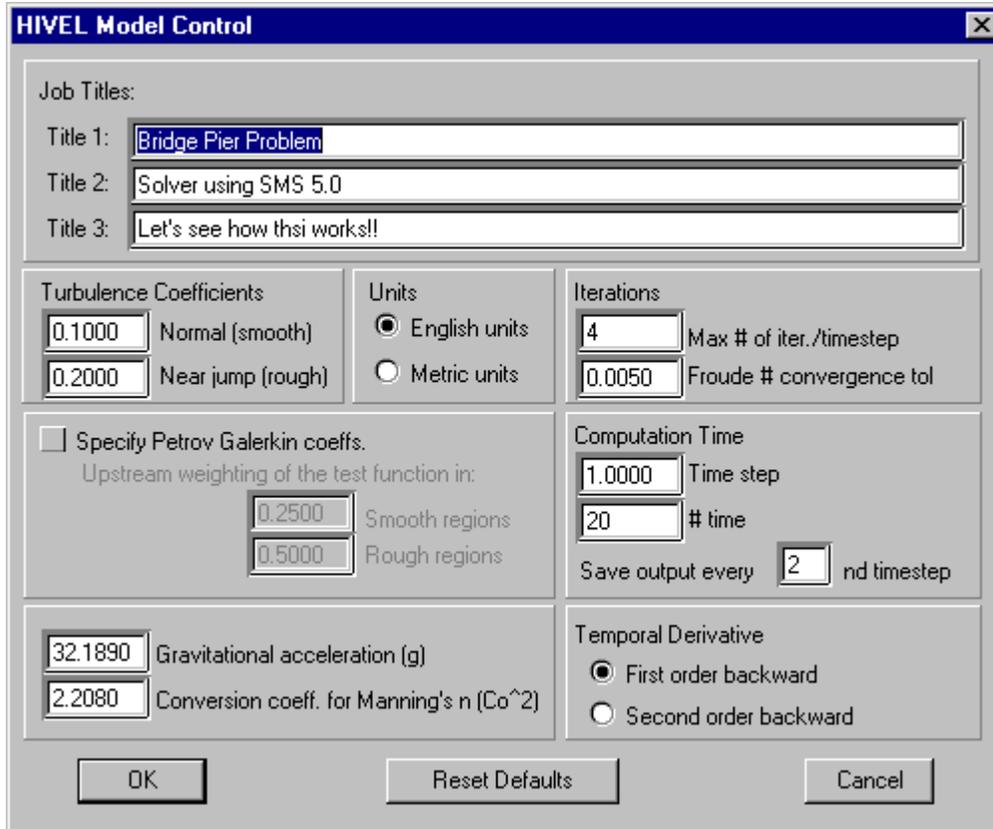


Figure 3.5 Model control

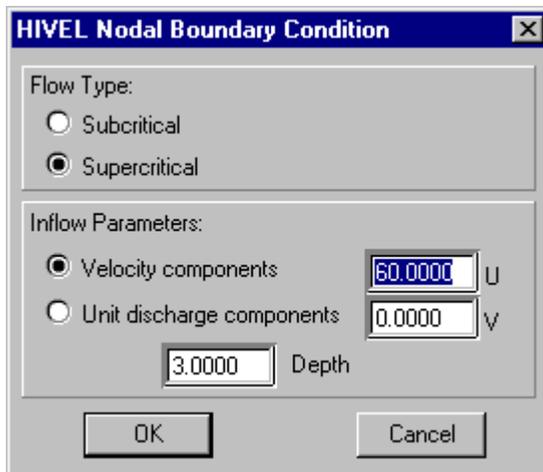


Figure 3.6 Nodal Boundary Conditions

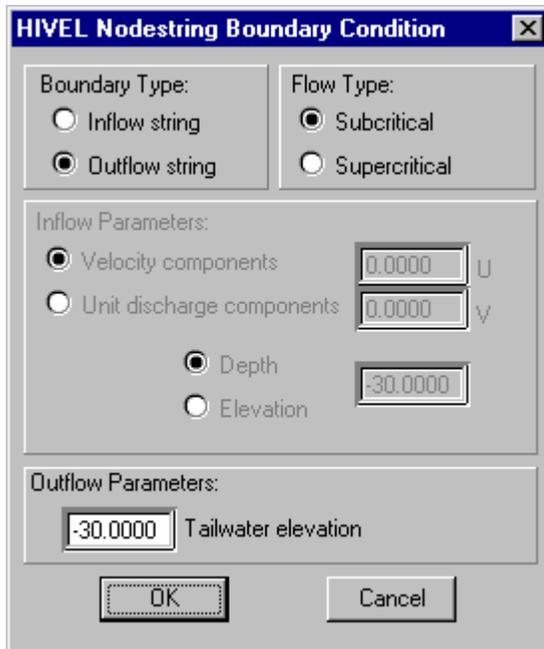


Figure 3.7 Node String Boundary Conditions

Model Checker

Once a grid is generated and all of the analysis options and boundary conditions have been specified, the next step is to save the simulation to disk and *Run HIVEL*. However, before saving the simulation and running HIVEL, the model should be checked with the *Model Checker*. Because of the significant amount of data required for a HIVEL2D simulation, it is often easy to neglect important data or to define inconsistent or incompatible options and parameters. Such errors will either cause HIVEL2D to crash or to generate an erroneous solution. The purpose of the *Model Checker* is to analyze the input data currently defined for the model simulation and report any obvious errors or potential problems.

Running the *Model Checker* successfully does not guarantee that a solution will be correct. It simply serves as an initial check on the input data and can save a considerable amount of time that would otherwise be lost tracking down input errors.

To check the current HIVEL2D data, select the *Model Check* command from the *HIVEL* menu. The *Model Check* dialog shown in Figure 3.8 will appear. To run the *Model Checker*, select the button labeled *Run Check* at the bottom of the dialog. This generates a list of possible errors and warning messages in the top scrolling window.

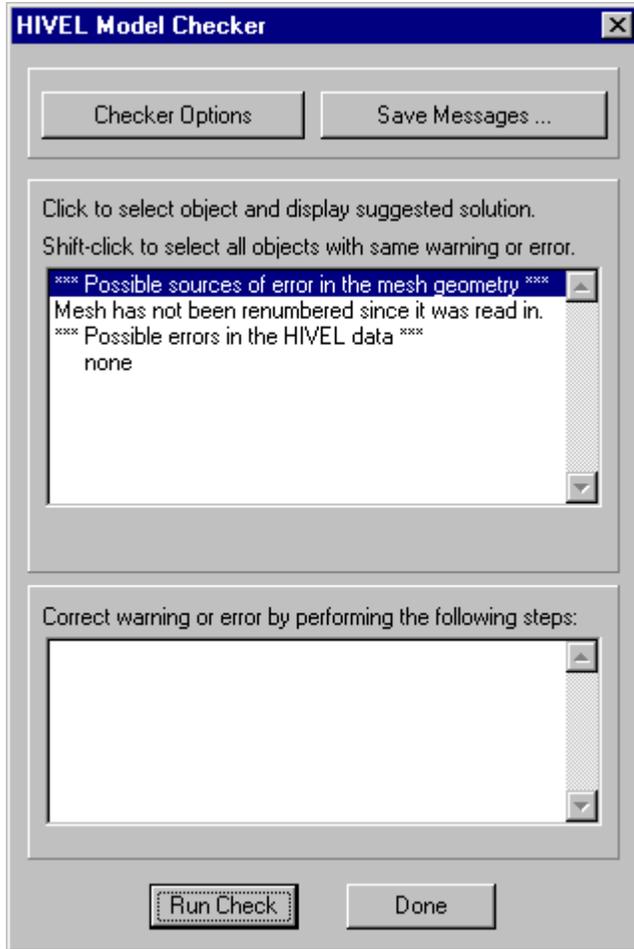


Figure 3.8 Model Check

Run HIVEL

Once a simulation has been created, checked, and saved, the HIVEL2D executable program can be run. HIVEL2D can be executed one of two ways: from the HIVEL menu in SMS or from the DOS or UNIX command line.

Launching HIVEL2D from the Menu

HIVEL2D can be run from the *Hivel* menu in *SMS* using the following procedure:

1. Make sure you have saved the simulation using the *Save Simulation* command.
2. Select the *Run Hivel* command from the *Hivel* menu.

3. At this point *HIVEL2D* is launched in a new window, prompting the user to type the name of the super file. Once the super file name is typed, *HIVEL2D* opens the file and begins the simulation

Launching HIVEL2D from the Command Prompt

HIVEL2D can be run from DOS/UNIX command prompt using the following procedure:

1. Make sure you have saved the simulation using the *Save Simulation* command.
2. Open a DOS/UNIX shell (and the appropriate directory) and type the HIVEL2D executable.
3. HIVEL2D is launched, prompting the user to type the name of the super file. Once the super file name is typed, *HIVEL2D* opens the file and begins the simulation.

Output Files

Two ASCII solution files (Appendix A for file formats) are generated as the model output. The *Data Browser* command from the *Data* menu can be used to *Import* the solution files for viewing. The output file contains the following:

- a. The number of time-steps saved, number of elements, and number of nodes.
- b. The time.
- c. Velocity components of u and v , and water-surface elevation for each node.

Note: these are repeated for each time-step that was output.

The number of time-steps saved is determined by the request in the hydrodynamic input file for calculated time-steps and NRVL (the interval to save time-steps to this file). This file can get quite large, so assign NRVL with discretion.

References

Abbott, M. B. (1979). *Computational hydraulics, elements of the theory of free surface flows*. Pitman Advanced Publishing Limited, London.

Berger, R. C. (1993). "A finite element scheme for shock capturing," Technical Report HL-93-12, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

Brigham Young University. (1997). "SMS 5.0 MS-Windows Version," Engineering Computer Graphics Laboratory, Brigham Young University, Provo, UT.

Rodi, W. (1980). "Turbulence models and their application in hydraulics - A state of the art review," State-of-the Art Paper, International Association for Hydraulic Research, Delft, The Netherlands.

Stockstill, R. L., and Berger, R. C. (1994). "HIVEL2D: A two-dimensional flow model for high-velocity channels," Technical Report REMR-HY-12, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

Appendix A

File Formats

This section contains the file formats for files used by HIVEL2D. The files used by HIVEL2D use a "card" type format. With this format, the components of the file are grouped into logical groups called "cards." The first component of each card is a short name which serves as the card identifier. The remaining fields on the line contain the information associated with the card. In some cases, such as lists, a card can use multiple lines.

There are many advantages associated with the card type approach to formatting files. Some of the advantages are:

1. Card identifiers make the file easier to read. Each input line has a label which helps to identify the data on the line.
2. The card names are useful as text strings for searching in a large file. All input lines of a particular type can be located quickly using a find command in a text editor.
3. Cards allow the data to be input in any order in many cases, i.e., the order that the cards appear in the file is usually not important.
4. Cards make it easy to modify a file format. New data can be included simply by defining a new card type. If the new card is optional (which is often the case for new cards) old files are still compatible. If an old card type is no longer used, the card can simply be ignored without causing input errors.

HIVEL2D Super File

A HIVEL2D super file is a file which contains a list of other files. Each of the files in the list must be one of the basic HIVEL2D file types (geometry, hydrodynamic flow,..). If a super file is selected using the *Read* command in the *File* menu, each of the files listed in the super file are opened and read. This makes it possible to quickly read in several files at once without having to identify each file individually in the file browser. The file format for a super file is shown in Figure A1. The first line in the file is a SUPER card which identifies the file as a super file. Each of the other cards shown are optional. Each of the cards has a card identifier representing the type of file. The

identifier is followed by a file name. The file name should be a complete path if the file is not in the same directory as the super file. Any suffix may be used with the file name. A sample super file is shown in Figure A2.

GEOM	filename	/* Geometry file */
FLOW	filename	/* Hydrodynamic flow file name */
HOTS	filename	/* Hot Start file */
WSOL	filename	/* First solution file */
VSOL	filename	/* Second solution file */

Figure A1 Super File Format.

GEOM	bridge.geo
FLOW	bridge.flo
HOTS	bridge.hot
WSOL	wsol.dat
VSOL	vsol.dat

Figure A2 Sample Super File

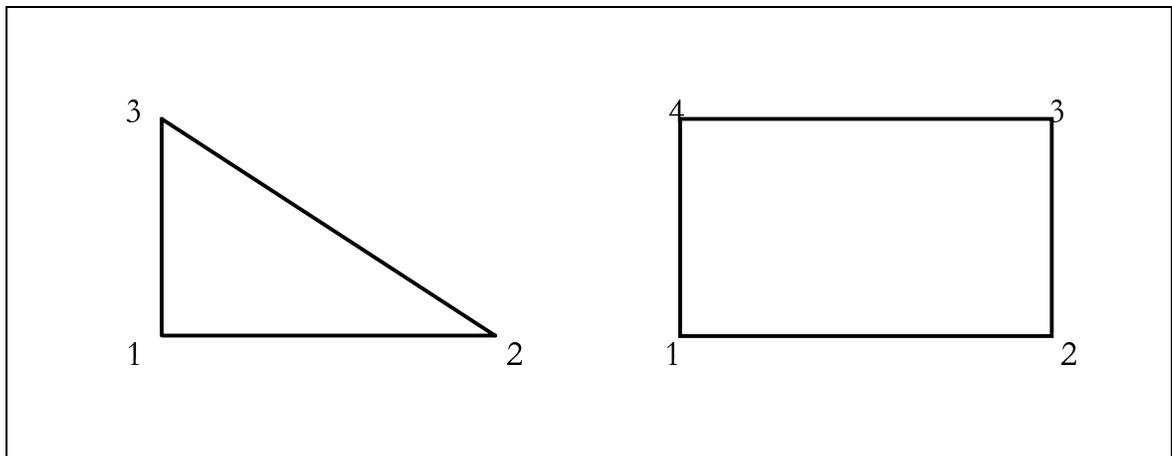


Figure A3. Numbering sequence for quadrilateral and triangular elements

2D Geometry File

Two-dimensional finite element geometry files are stored in a 2D *GEO* file. The types of elements supported in HIVEL2D and the numbering sequence for the elements are shown in Figure A3. The file format is shown in Figure A4. A sample geometry file is shown in figure D3 (Appendix D).

T1	/* Title Lines */
T2	/* Title Lines */
T3	/* Title Lines */
E3 id n1 n2 n3 mat	/* 3 node triangle */
E4 id n1 n2 n3 n4 mat	/* 4 node quad */
ND id x y z	/* Nodal coordinates */

Figure A4. 2D Geometry File Format.

The card type used in the 2D geometry file are as follows:

<i>Card Type</i>	T1, T2, T3		
<i>Description</i>	Title Lines		
<i>Required</i>	No		
<i>Format</i>	T1 textT2 textT3 text		
<i>Sample</i>	T1 Sample Problem T2 Charlie Berger T3 3/1/97		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	text		Any text string

<i>Card Type</i>	E3		
<i>Description</i>	Defines a three node (linear) triangle element		
<i>Required</i>	No		
<i>Format</i>	E3 id n1 n2 n3 mat		
<i>Sample</i>	E3 283 13 32 27 3		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	id	+	The id of the element
2-4	n1-n3	+	The nodal indices of the element ordered counterclockwise
5	mat	+	The material id of the element

<i>Card Type</i>	E4		
<i>Description</i>	Defines a four node (Bi-linear) quadrilateral element		
<i>Required</i>	No		
<i>Format</i>	E4 id n1 n2 n3 n4 mat		
<i>Sample</i>	E4 283 13 32 27 30 3		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	id	+	The id of the element
2-5	n1-n4	+	The nodal indices of the element ordered counterclockwise
6	mat	+	The material id of the

			element
<i>Card type</i>	ND		
<i>Description</i>	Defines the coordinates of a node		
<i>Required</i>	No		
<i>Format</i>	ND id x y z		
<i>Sample</i>	ND 84 120.4 380.3 5632.0		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	id	+	The id of the element
2-4	x,y,z	±	The nodal coordinates

Hydrodynamic Flow File

Two-dimensional finite element geometry files are stored in a 2D *GEO* file. The file format is shown in Figure A5. A sample geometry file is shown in figure D3 (Appendix D).

GRAV	/* Gravitational Acceleration */
MCON	/* Conversion coefficient for manning's n*/
TURB	/* Turbulence coefficients for smooth and rough region* /
TIME	/* Time step and temporal derivative order */
STEP	/* Number of time steps and interval */
ITER	/* Number of iterations and convergence criteria */
PGWC	/* Petrov Galerkin coefficients for model stability */
BIN	/* Inflow locations by nodes */
BIS	/* Inflow locations by string */
BOS	/* Outflow locations by string */
MTYP	/*Material type and manning's n for each */

Figure A5. 2D Geometry File Format.

The card type used in the hydrodynamic flow file are as follows:

<i>Card Type</i>	GRAV		
<i>Description</i>	Gravitational acceleration		
<i>Required</i>	Yes		
<i>Format</i>	GRAV g		
<i>Sample</i>	GRAV 32.2		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	g	+	Gravitational acceleration in the units of your choice

<i>Card Type</i>	MCON		
<i>Description</i>	Conversion coefficient for Manning's <i>n</i> formulation 2.208 for English units and 1.0 for Metric units		
<i>Required</i>	Yes		
<i>Format</i>	MCON c2		
<i>Sample</i>	MCON 2.208		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	c2	+	(1.486) ² for English (1.000) ² for Metric

<i>Card Type</i>	TURB		
<i>Description</i>	Turbulence coefficients for smooth and rough regions		
<i>Required</i>	Yes		
<i>Format</i>	TURB csb csbshk		
<i>Sample</i>	TURB 0.10 0.50		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	csb	0.1-1.0	coef. for smooth regions
2	csbshk	0.1-1.0	coef. for rough regions

<i>Card Type</i>	TIME		
<i>Description</i>	Time step size and order of temporal derivative		
<i>Required</i>	Yes		
<i>Format</i>	Time dt a		
<i>Sample</i>	Time 1.25 2.00		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	dt	+	time step size
2	a	1.00-2.00	1.0 first order backward 2.0 second order

			backward
--	--	--	----------

<i>Card Type</i>	STEP		
<i>Description</i>	Number of time steps, and interval for storing output		
<i>Required</i>	Yes		
<i>Format</i>	STEP nsteps nrvl		
<i>Sample</i>	STEP 50 5		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	nsteps	+	Number of time steps to run
2	nrvl	+	Interval for saving output, ie 5 means save time step 5,10,15,..

<i>Card Type</i>	ITER		
<i>Description</i>	Number of iterations per time step and convergence parameter		
<i>Required</i>	Yes		
<i>Format</i>	ITER itmax tol		
<i>Sample</i>	ITER 5 0.0005		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	itmax	+	Maximum number of iterations allowed per time step
2	tol	+	If maximum relative change in nodal Froude number is less than

			tolerance the code will advance to the next time step
<i>Card Type</i>	PGWC		
<i>Description</i>	Petrov-Galerkin coefficients for model stability		
<i>Required</i>	No (default values will be used)		
<i>Format</i>	PGWC sm sh		
<i>Sample</i>	PGWC 0.10 0.50		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	sm	0.00-0.50	“upstream” weighting of the test function in smooth regions
2	sh	0.00- 0.50	same but for rough areas like shocks

<i>Card Type</i>	MTYP		
<i>Description</i>	Number of material types and manning’s n for each		
<i>Required</i>	Yes		
<i>Format</i>	MTYP k 1 fric1 2 fric1 . . k fric1		
<i>Sample</i>	MTYP 3 1 0.014 2 0.020 3 0.025		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	k	+	Number of material types
1 (continued)	mat	+	Material number
2 (continued)	fric	+	Manning’s n for the

			material
--	--	--	----------

<i>Card Type</i>	BOS		
<i>Description</i>	Outflow location (list of nodes)		
<i>Required</i>	No - if you intend no outflow		
<i>Format</i>	BOS ibost iscsc tail nbo n1 n2 . . nn		
<i>Sample</i>	BO 2 0 120.4 4 470 369 242 333		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	ibost	+	id for outflow node string
2	iscsc	0 or 1	0 subcritical outflow, read specified tailwater elevation
3	tail	+	tailwater elevation to be applied along this string of nodes
4	nbo	+	number of nodes that make up this outflow string
1 (continuation)	n1 n2 . . nn	+	string of node ids that makeup this outflow

The inflow nodes can be specified by either a line of nodes (BIS) or by individual nodes (BIN). For the BIS specification the flow vector is defined as a perpendicular to the BIS line (say it has to be straight) and into the model.

<i>Card Type</i>	BIS		
<i>Description</i>	Inflow location (list of nodes)		
<i>Required</i>	No - if you intend no inflow		
<i>Format</i>	BIS ibist nbi iuorp vxi vyi ihe hi n1 n2 . nn		
<i>Sample</i>	BIS 2 4 2 21.23 1.67 1 27.32 111 112 113 114		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	ibist	+	string id
2	nbi	+	Number of nodes that make up this inflow string
3	iuorp	-2 , 2, -1, 1	-2 Supercritical boundary specify u, v, h 2 Supercritical boundary specify p, q, h -1 Subcritical boundary specify u,v 1 Subcritical boundary specify p, q
4,5	vxi, vyi	±, ±	x and y components of either velocity or unit discharge depending on iuorp
6	ihe	0,1	0 Depth specified 1 Elevation specified If this is a subcritical boundary this variable is not used.
7	hi	+	Either the depth or water surface elevation of this string. If this is a subcritical boundary

			this variable is not used.
1 (continuation)	n1 . nn	+	Node ids that makeup this inflow node string.

<i>Card Type</i>	BIN		
<i>Description</i>	Inflow location and amount by node		
<i>Required</i>	No		
<i>Format</i>	BIN nd iuorp vxni vyni ihe hni		
<i>Sample</i>	BIS 363 2 21.23 1.67 1 27.32		
<i>Field</i>	<i>Variable</i>	<i>Value</i>	<i>Description</i>
1	nd	+	node id
2	iuorp	-2 , 2, -1, 1	-2 Supercritical boundary specify u,v,h 2 Supercritical boundary specify p,q,h -1 Subcritical boundary specify u,v 1 Subcritical boundary specify p, q
3.4	vxni, vyni	±, ±	x and y components of either velocity or unit discharge depending on iuorp
7	hni	+	Depth at the node is specified at supercritical inflows and at subcritical inflows no specification is needed (or used).

Output Files

The output data sets can be stored in ASCII files. The ASCII scalar and vector data set formats are shown in Figures A6 and A7. A sample data set file is shown in Figure A8 and A9. For scalar data set files, one value is listed per node. For vector data set files, one set of XY vector components is listed per node.

```

DATASET          /* File type identifier */
OBJTYPE type     /* Type of object data set is associated with */
BEGSCL          /* Beginning of scalar data set */
OBJID id        /* Object id */
ND numdata      /* Number of data values */
NC numcells     /* Number of elements */
NAME "name"     /* Data set name */
TS istat time   /* Time step of the following data. */
stat 1          /* Status flags */
stat 2
.
.
stat numcells
val 1 /* Scalar data values */
val 2
.
.
val numdata
/* Repeat TS card for each time step */
ENDDDS /* End of data set */
    
```

Figure A6 ASCII Scalar Data Set File Format.

```

DATASET          /* File type identifier */
BEGVEC          /* Beginning of vector dataset */
VECTYPE type    /* Vector at node/grid or element/cell */
ND numdata      /* Number of data values */
NC numcells     /* Number of elements */
NAME "name"     /* Data set name */
TS istat time   /* Time step of the following data. */
stat 1          /* Status flags */
stat 2
.
.
stat numcells
v x1 v y1
v x2 v y2
.
.
v numdata v numdata v numdata
    
```

```
/* Repeat TS card for each time step */  
ENDDDS /* End of data set */  
/* Repeat BEGSCL and BEGVEC sequences for each data set */
```

Figure A7 ASCII Vector Data Set File Format.

```
DATASET  
OBJTYPE mesh2d  
BEGSCL  
ND 8  
NC 8  
NAME water surface  
TS 0 1.00000000e+00  
0.00000000e+00  
0.00000000e+00  
0.00000000e+00  
3.24000000e+00  
4.39000000e+00  
2.96000000e+00  
7.48000000e+00  
0.00000000e+00  
ENDDDS
```

Figure A8 Sample ASCII Scalar Data Set.

```
DATASET  
OBJTYPE mesh2d  
BEGVEC  
OBJID 27211  
ND 8  
NC 8  
NAME velocity  
TS 0 1.00000000e+00  
5.692781 -5.649887  
5.726238 -5.683091  
5.768243 -5.724780  
5.780255 -5.736701  
5.769043 -5.725574  
5.728620 -5.685455  
5.698828 -5.655888  
5.682499 -5.573179  
ENDDDS
```

Figure A9 Sample ASCII Vector Data Set.

Appendix B

Boundary Conditions

Flow In and Out

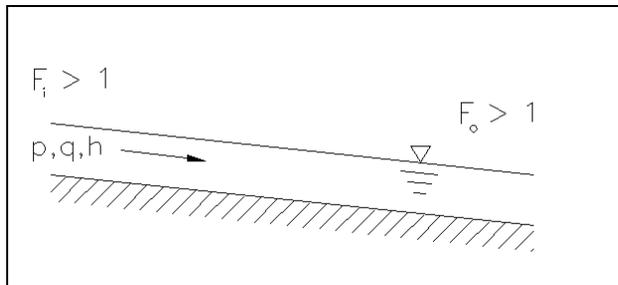
Case 1

Flow regime

Supercritical upstream
 Supercritical downstream

Boundary conditions

p, q, h specified upstream
 (defined in Chapter 2)



Comments

It can be difficult to get the outflow boundary to converge at startup. One method is to specify a tailwater such that the flow is barely subcritical, then after some time, when the resulting jump is nearly gone, remove the downstream boundary condition. Note: F_i = inflow Froude number; F_o = outflow Froude number.

Figure B1. Case 1 sketch

Case 2

Flow regime

Supercritical upstream
 Subcritical downstream

Boundary conditions

p, q, h specified upstream
 h specified downstream

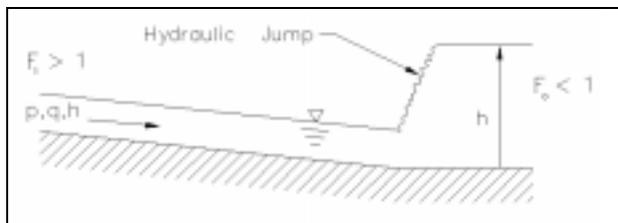


Figure B2. Case 2 sketch

Case 3

Flow regime

Subcritical upstream
Supercritical downstream

Boundary conditions

p, q specified upstream

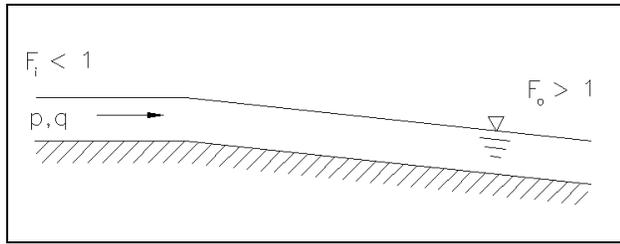


Figure B3. Case 3 sketch

Case 4

Flow regime

Subcritical upstream
Subcritical downstream

Boundary conditions

p, q specified upstream
 h specified downstream

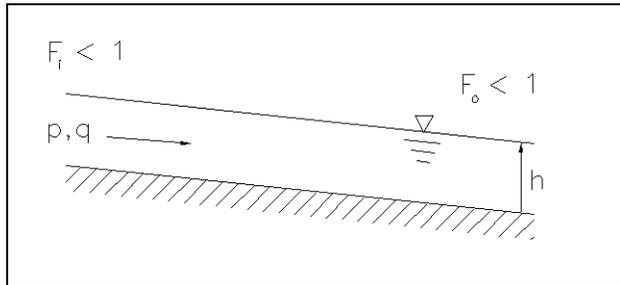


Figure B4. Case 4 sketch

Comments

This problem may take a long time to reach steady state.

Case 5

Flow regime

Subcritical upstream
Subcritical downstream

Boundary conditions

p, q specified upstream
 h specified downstream

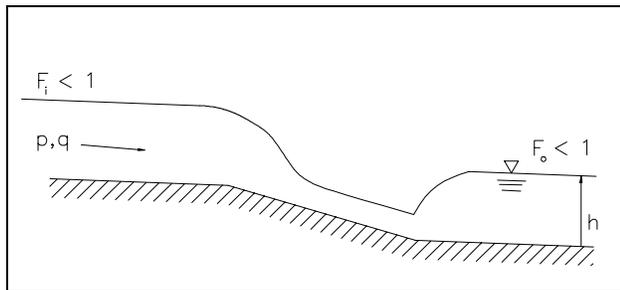


Figure B5. Case 5 sketch

Comments

Even though this contains a steep section in which the flow is supercritical, the boundary conditions are the same as for Case 4. The tailwater specified downstream causes the hydraulic jump.

Flow In, No Flow Out

Case 6

Flow regime

Supercritical upstream
Wall downstream

Boundary conditions

p, q, h upstream
Leave out BO lines
(BI and BO defined in Table 1)

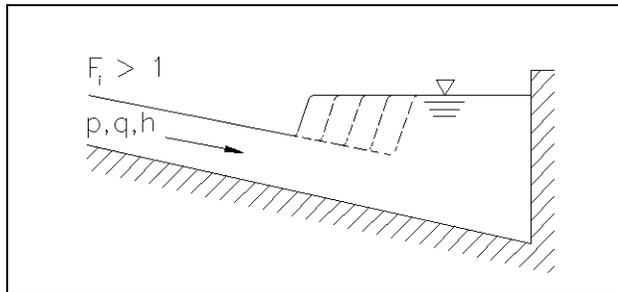


Figure B6. Case 6 sketch

Comments

No steady state, keeps filling. Note the moving jump that forms.

Case 7

Flow regime

Subcritical upstream
Wall downstream

Boundary Conditions

p, q specified upstream
Leave out BO lines in geo file

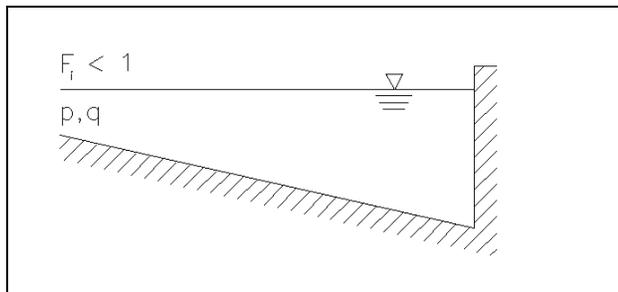


Figure B7. Case 7 sketch

Comments

There is no steady-state answer, just keeps filling.

No Flow In, Flow Out

Case 8

Flow regime

Wall upstream
Supercritical downstream

Boundary conditions

Leave out BI lines

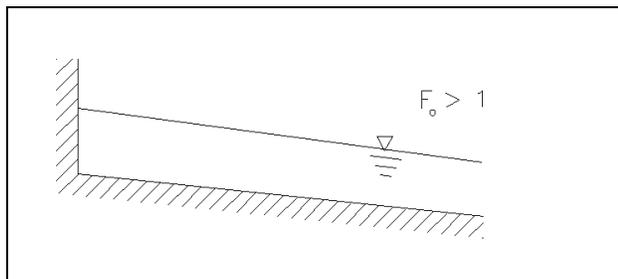


Figure B8. Case 8 sketch

Comments

No steady state, goes dry. This is probably a very hard problem to run.

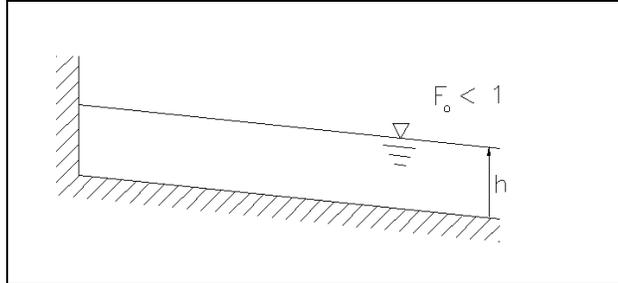
Case 9

Flow regime

Wall upstream
Subcritical downstream

Boundary conditions

Leave out BI lines
Specify h downstream



Comments

No steady state, goes dry.
This is also a very hard
problem to run.

Figure B9. Case 9 sketch

No Flow In or Out

Case 10

Flow regime

Wall upstream
Wall downstream

Boundary conditions

Leave out BI and BO lines

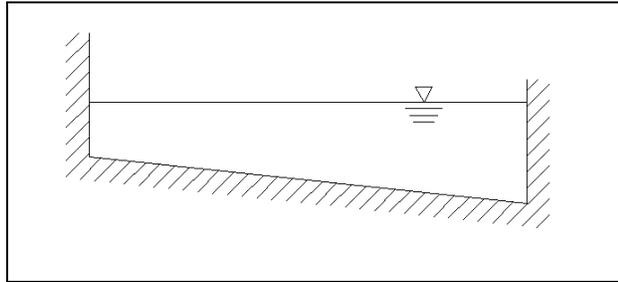


Figure B10. Case 10 sketch

Appendix C

Troubleshooting

Table C1 Problems and Solutions	
Problem	Possible Solutions
1. Model won't converge	Try more iterations per time-step. Reduce time-step. If lower boundary is migrating away from solution: First set a tailwater, wait until jump is nearly gone, then remove tailwater boundary condition. Call for help: be prepared to send the geometry, hydraulic input, output, and hot start files.
2. Model keeps filling up (water level continuously rises).	Make sure you didn't leave out BO lines in geometry file. Use <i>BO</i> and not <i>B0</i> (zero) (Table 1).
3. Converges to a bizarre solution.	Make sure resolution is adequate.
4. Problems reading hot start file.	Verify that the time is included as the first line of the file. Verify the number of lines. May have to put a carriage return at the end of the last line of the hot start file.

Appendix D

Example Problem

The example grid is shown in Figure D1. The flow enters on the left and exits to the right. The 2,000-ft (610-m)-long model is supercritical at both boundaries, so all information is specified at the upstream end. The example contains a transition from a 100-ft (30.5-m) width to a 90-ft (27.4-m) width. The slope is 0.01 except in the 100-ft (30.5-m) length of transition, where it is 0.02. The transition is to have a bed roughness higher than the rest of the model. The model includes a 50-ft (15.2-m)-long by 18-ft (5.5-m)-wide bridge pier. Note that the resolution is concentrated around and downstream of any abrupt changes (Figure D2). Since this flow is supercritical, all information including shocks and rough conditions is swept downstream. Therefore, this is where the resolution is needed.

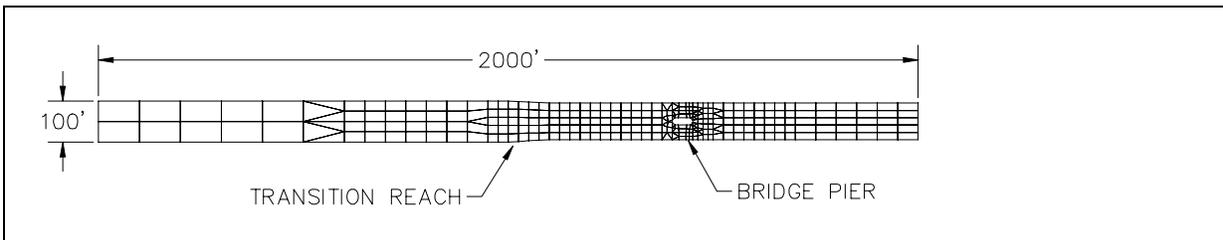


Figure D1. Complete example grid and geometry

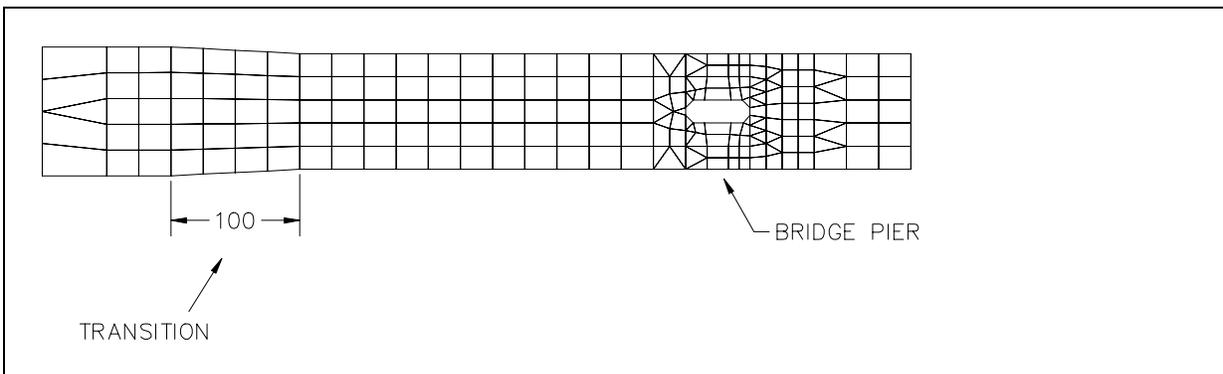


Figure D2. Detail of example grid showing transition and bridge pier

The complete geometry file includes 338 nodes and 302 elements. An abbreviated version of the geometry file is shown in Figure D3.

```

T1 example geometry file, contains 338 nodes, representing a bridge pier
T2 and a contraction, the contraction is steeper and rougher than
T3 the rest of the model
E4 1 5 2 1 4 1
E4 2 6 3 2 5 1
E4 3 8 5 4 7 1
E4 4 9 6 5 8 1
E4 5 11 8 7 10 1
E4 6 12 9 8 11 1
.
.
.
E4 299 335 329 328 334 1
E4 301 337 331 330 336 1
E4 302 338 332 331 337 1
ND 1 0.0000 0.0000 0.0000
ND 2 0.0000 50.0000 0.0000
ND 3 0.0000 100.0000 0.0000
ND 4 100.0000 0.0000 -1.0000
ND 5 100.0000 50.0000 -1.0000
ND 6 100.0000 100.0000 -1.0000
.
.
.
ND 334 2000.0000 23.0000 -20.0000
ND 335 2000.0000 41.0000 -20.0000
ND 336 2000.0000 59.0000 -20.0000
ND 337 2000.0000 77.0000 -20.0000
ND 338 2000.0000 95.0000 -20.0000
    
```

Figure D3. Example geometry file (example.geo)

The initial hot start file can be tricky. This case involves a straight reach so it is fairly easy to set up a file with p, q, h of 60.,0., 3. as a constant throughout the domain. If the domain is curved one might have to start with pooled flow at a single depth. The beginning of the initial hot start file is shown in Figure D4. The initial input file is shown in Figure D5.

The text in the hydrodynamic input file that describes each numerical input is not actually read by the program so it can be left out or changed. The initial time-step was chosen to produce a CFL number (Equation 15) of roughly 1 (so this time-step is chosen as 1.0 sec). The temporal derivative

```

0.000
60. 0. 3. 60. 0. 3.
60. 0. 3. 60. 0. 3.
60. 0. 3. 60. 0. 3.
60. 0. 3. 60. 0. 3.
60. 0. 3. 60. 0. 3.
60. 0. 3. 60. 0. 3.
60. 0. 3. 60. 0. 3.
.
.
.
(This line is repeated at least 338 times, one line for each node)
    
```

Figure D4. Example hot start file (example.hot).

```

grav 32.18900          | GRAVITY (FT/(SEC*SEC))
mcon 2.20800          | MANNING CONVERSION
turb 0.1000 0.2000    | CB FOR VISCOSITY SMOOTH, AND JUMP
time 1.0000 1.0000    | TIME STEP, TEMPORAL DIFF
step 20 20           | TIME STEPS, OUTPUT INTERVAL
iter 4 0.00500        | NUMB. OF ITER, CONV CRITERIA
bin 1 2 60.000 0.000 1 3.000
bin 2 2 60.000 0.000 1 3.000
bin 3 2 60.000 0.000 1 3.000
bos 2 0 -14.8 6
333
334
335
336
337
338
mtyp 2                | NUMBER OF ELEMENT TYPES(FOR ROUGHNESS)
1 0.0150
2 0.0250

```

Figure D5. Initial hydrodynamic input file (example1.flo).

was chosen to be first-order since the interest is in steady-state results only. Even though the downstream boundary is intended to be supercritical and no boundary condition is needed, experience has shown that the model sometimes will converge to a different solution or be unstable unless a very good starting guess at the solution was made. Therefore, until the model settles down, a tailwater elevation is specified corresponding to flow that is slightly subcritical. Two different element types and Manning’s roughness coefficients have been designated. The convergent section is steeper and rougher in this example. The model will run 20 time-steps resulting in a simulation time of 20 sec. The time-step size is then increased. Later the downstream tailwater condition is effectively removed by specifying an elevation that is below the bed elevation. The nodes 1, 2, and 3 are the upstream boundary nodes (designated by BIN in the hydrodynamic flow file), and nodes 333-338 are the downstream boundary nodes (designated by BOS in the hydrodynamic flow file).

The next input file is run to raise the time-step now that the model has settled down from startup. The time-step has been raised to 3.0 sec. The second input file is shown in Figure D6.

The final input file (Figure D7) removes the tailwater constraint. This run was repeated one time.

Figures D8 and D9 are the steady-state results of the model run for water-surface and velocity contours, respectively. These were generated by SMS. There are many options available such as vectors and time-series plots. The results show that the bridge pier is a

choke, so that the flow is subcritical in front of the pier. This is apparent since the shock contour is

```

grav 32.18900          | GRAVITY (FT/(SEC*SEC))
mcon 2.20800          | MANNING CONVERSION
turb 0.1000 0.2000    | CB FOR VISCOSITY SMOOTH, AND JUMP
time 3.0000 1.0000    | TIME STEP, TEMPORAL DIFF
step 20 20            | TIME STEPS, OUTPUT INTERVAL
iter 4 0.00500        | NUMB. OF ITER, CONV CRITERIA
bin 1 2 60.000 0.000 1 3.000
bin 2 2 60.000 0.000 1 3.000
bin 3 2 60.000 0.000 1 3.000
bos 2 0 -14.8 6
333
334
335
336
337
338
mtyp 2                | NUMBER OF ELEMENT TYPES(FOR ROUGHNESS)
1 0.0150
2 0.0250

```

Figure D6. Second hydrodynamic input file (example2.flo).

```

grav 32.18900          | GRAVITY (FT/(SEC*SEC))
mcon 2.20800          | MANNING CONVERSION
turb 0.1000 0.2000    | CB FOR VISCOSITY SMOOTH, AND JUMP
time 1.0000 1.0000    | TIME STEP, TEMPORAL DIFF
step 20 20            | TIME STEPS, OUTPUT INTERVAL
iter 4 0.00500        | NUMB. OF ITER, CONV CRITERIA
bin 1 2 60.000 0.000 1 3.000
bin 2 2 60.000 0.000 1 3.000
bin 3 2 60.000 0.000 1 3.000
bos 2 0 -30 6
333
334
335
336
337
338
mtyp 2                | NUMBER OF ELEMENT TYPES(FOR ROUGHNESS)
1 0.0150
2 0.0250

```

Figure D7. Final hydrodynamic input file (example3.flo).

perpendicular to the flow instead of being swept back. Note also that the flow returns to supercritical flow along the pier sides and a depression wave has the sweptback wake.

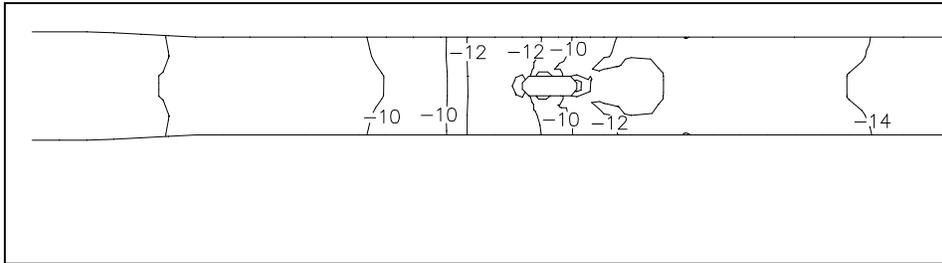


Figure D8. Final water-surface contours, ft

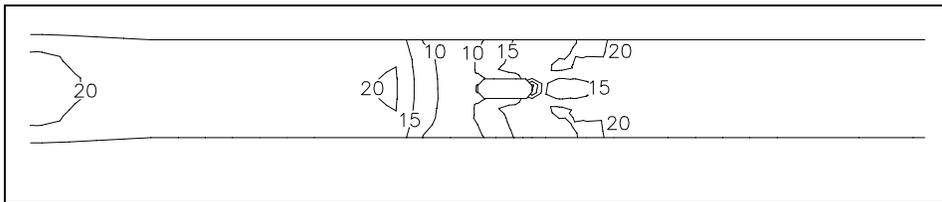


Figure D9. Final velocity contours, fps

Appendix E

Interpolation Program

The interpolation program INTRPL8.EXE allows the user to refine a grid for which answers (p , q , and h , defined in Chapter 2) have already been found. Using the old hot start information, the old geometry, and the new geometry, the user can generate a new hot start file for the new geometry. This new hot start file is then the starting point for a new run of HIVEL2D.

The program also outputs an interpolated plot output file that can be converted to binary and viewed in SMS. This is used to verify the inter-polated values. This feature can also be used to interpolate from a fine grid to a coarse grid, to better view vectors on a very fine grid, for example. Users should note that the program cannot interpolate answers for any nodes that lie outside the old geometry's boundary. For instance, refining a mesh around a curve will cause the nodes along the curve's outer radius to lie outside the old coarse mesh's boundary. The program will report the nodes that lie outside the domain.

Program Description

After input of the necessary data, the first routine the program performs is that of comparing the old grid with the new grid. Old node numbers are typically going to be different than new ones. For this reason, the old answers at each node are first "mapped" onto the corresponding nodes on the new grid by comparing the coordinates of the nodes.

Next comes the task of looking at each node, seeing if it is a new node, and, if it is a new node, finding out in which old element it is located. There are two possible locations for a new or "target" node: in a quadrilateral element or in a triangular element. A node on an element side is interpolated in the element in which it is first found.

Deciding in which element a target node is located is done by a cross product scheme, in which the cross product is taken between a vector lying on the side of an element and

a vector from a corner node on that side to the target point. If the cross products for all sides are greater than zero, then the node lies within that element.

Interpolation Scheme for Quadrilateral Elements

For convenience in interpolating, a local coordinate system (x,h) is used where $-1 \leq \xi \leq 1$ and $-1 \leq \eta \leq 1$ (Figure E1) to derive the bilinear interpolation functions. First the target point (x_p, y_p) is written in (x,h) coordinates.

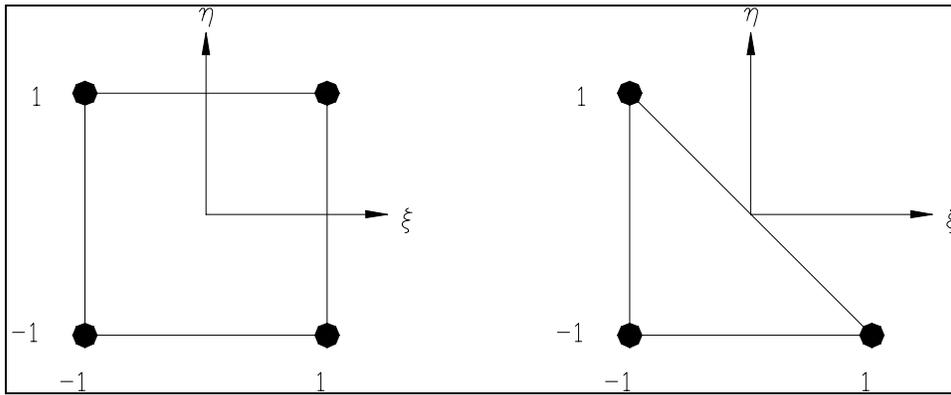


Figure E1. Local coordinates for quadrilateral and triangular elements

This is done using Newton-Raphson iteration to solve the following equations:

$$x = \sum_{i=1}^4 \phi_i x_i \quad (\text{E1})$$

and

$$y = \sum_{i=1}^4 \phi_i y_i \quad (\text{E2})$$

for the local coordinates x and h where the bilinear shape functions for a quadrilateral element are:

$$\phi_1 = \frac{1}{4}(1 - \xi)(1 - \eta)$$

$$\phi_2 = \frac{1}{4}(1 + \xi)(1 - \eta)$$

$$\phi_3 = \frac{1}{4}(1 + \xi)(1 + \eta)$$

$$\phi_4 = \frac{1}{4}(1 - \xi)(1 + \eta)$$

Interpolation Scheme for Triangular Elements

Again, the local coordinate system (x,h) where $-1 \leq \xi \leq 1$ and $-1 \leq \eta \leq 1$, (Figure E1) is used to derive the interpolation functions. Solving

$$x = \sum_{i=1}^3 \phi_i x_i \quad (\text{E3})$$

and

$$y = \sum_{i=1}^3 \phi_i y_i \quad (\text{E4})$$

simultaneously for x and h, where:

$$\phi_1 = -\frac{1}{2}(\xi + \eta)$$

$$\phi_2 = \frac{1}{2}(1 + \xi)$$

$$\phi_3 = \frac{1}{2}(1 + \eta)$$

are the bilinear shape functions for a triangular element, results in the following equations for x and h:

$$\xi = -\frac{\beta_1}{\alpha_1} + \frac{\alpha_2(-\alpha_3\beta_1 + \alpha_1\beta_2)}{\alpha_1(-\gamma_1 + \gamma_2 + \gamma_3 - \gamma_4 - \gamma_5 + \gamma_6)} \quad (\text{E5})$$

and

$$\eta = -\frac{(-\alpha_3\beta_1 + \alpha_1\beta_2)}{(-\gamma_1 + \gamma_2 + \gamma_3 - \gamma_4 - \gamma_5 + \gamma_6)} \quad (\text{E6})$$

where

$$\alpha_1 = x_1 - x_2$$

$$\alpha_2 = x_1 - x_3$$

$$\alpha_3 = y_1 - y_2$$

$$\beta_1 = 2x_p - x_2 - x_3$$

$$\beta_2 = 2y_p - y_2 - y_3$$

$$\gamma_1 = x_2y_1$$

$$\gamma_2 = x_3y_1$$

$$\gamma_3 = x_1y_2$$

$$\gamma_4 = x_3y_2$$

$$\gamma_5 = x_1y_3$$

$$\gamma_6 = x_2y_3$$

$$\begin{aligned} \gamma_5 &= x_1y_3 \\ \gamma_5 &= x_1y_3 \end{aligned}$$

Interpolation

Once the target node has been located, and mapped onto old grid element local coordinates x and h , the interpolated values of p , q , and h are found using the following equations:

$$h = \sum_{i=1}^n \phi_i h_i \quad (\text{E7})$$

$$p = \sum_{i=1}^n \phi_i p_i \quad (\text{E8})$$

$$q = \sum_{i=1}^n \phi_i q_i \quad (\text{E9})$$

where n is the number of nodes on the old grid element and f is the appropriate interpolation function.