FLOW-3D V11

WATER & ENVIRONMENT TRAINING

EXERCISES

FLOW Science®

Improving the world through accurate flow modeling

Guide 1: CFD *Project* Workflow Guide

CFD Project Workflow Guide

Contents

This project workflow guide is not an exhaustive decision-tree for all elements of a project. Rather, it is meant to be a checklist with a middling level of detail. Some possible project elements have been omitted (generating geometry, for example). The Flow Science support staff hopes you find this guide helpful. Please send suggestions to support@flow3d.com.

Select a problem with known results for proof-of-concept testing

Whenever you must model a new class of problems using CFD, it is good practice to check whether you can set up a CFD model that gives expected results. To check, you will need a case that is representative of the problem of interest and that has known results. Once you have tested one or more representative cases for a problem type you can skip this step for new problems of the same type.

☐ Check the online resources for similar cases

Some classes of problems (ogee spillways, for example) have already been thoroughly validated against physical experiments with ***FLOW-3D***. You can search to see how other users have modeled the same type of problem:

- ☐ www.flow3d.com > *Resources > Bibliography > html search (Ctrl-F) for keywords*
- ☐ www.flow3d.com > *Site Search (upper right input box) for keywords*
- ☐ users.flow3d.com > *Conference Proceedings*, *Validations*, *Example Simulations*, *Technical Notes* (for cases related to specific physics models), and *Training* (for tutorials)

☐ Compare the test case to the case of interest

- ☐ How were the test case experimental results collected?
- ☐ Can experimental uncertainty be quantified for the test case?

- ☐ Are the test case results from a reliable source?
- ☐ Does the test case include the same physical processes as the case of interest?
- ☐ Does the test case have similar physical and time scales as the case of interest?
- ☐ Are the dimensionless numbers (*Re*, *Fr*, *We*, etc.) similar between the cases?

A test case may still be valuable even if it doesn't meet all of the above criteria, but there will be relatively less certainty in how to parameterize the case of interest.

Set up and run a coarse test case

☐ Follow the simulation setup guide using the coarsest possible mesh

This setup mesh is purposely kept *as coarse as possible* to make it run *as fast as possible*. After completing the simulation setup guide (on this site) you will have:

- ☐ An annotated diagram with all necessary information in a consistent unit system,
- ☐ A simulation folder with:
  - all *.stl* files (checked and repaired)
  - a ready-to-run *prepin* file
- ☐ A mesh that resolves the thinnest flow sections with 3 or 4 cells,
- ☐ A mesh that follows best-possible-practices,
- ☐ Rationales for all specified physics models and coefficients.

☐ Run and post-process the coarse case

If deadlines are pressing, only run a portion of the case. Answer the following questions:

- ☐ Are the *solver warnings*  acceptable?
- ☐ Do the *solver text* and *runtime plots* indicate that the time step sizes, pressure iterations, mean and turbulent kinetic energies, and fluid volume are appropriate?
- ☐ Does the model run in a reasonable amount of time?
- ☐ Is it possible to get the necessary outputs from post-processing?
- ☐ Check *3-D* and *2-D* plots. Do the results look right?
- ☐ Check *Probe* and *Text* outputs. Do the results follow the expected trends?

If the answers to the above questions are yes, the simulation is configured adequately. If any answers are no, consider adjusting the simulation setup with the following checklist:

☐ Troubleshoot the coarse case if necessary

- ☐ Does the mesh follow best-possible-practices?
- ☐ Are geometry files checked and repaired and well-resolved by the mesh?
- ☐ Are the boundary conditions appropriate and realistic?
- ☐ Are the initial conditions appropriate and realistic?
- ☐ Do the boundary conditions and initial conditions match?
- ☐ Are the active physics models appropriate?

- ☐ Are physics model and component coefficients correct? Check units and values carefully.
- ☐ If all of the above considerations have been checked *carefully* and the problem is not resolved, refer to the *User Manual > Troubleshooting* chapter.

Continue adjusting and re-running the coarse case model until you are confident in the simulation and ready to commit to longer run times.

Select and calibrate numerical methods

In this step, you will test the model sensitivity to different numerical methods. Whenever a higher-order method gives significantly different results, you must make a choice as to whether or not to use it. This step is optional but recommended whenever the coarse test runs quickly and there is sufficient time in the budget. Test as many of the following as you can.

☐ Check sensitivity to the VOF method

Models with free surfaces have several options for volume-of-fluid methods. The default option includes the most robust surface reconstruction method, but higher-order methods are available that may be more accurate. In particular, if you have a curved free surface that moves (*e.g.*, waves), a higher-order option may be desirable. A rule of thumb is to try Split Lagrangian VOF and compare the results to the default One Fluid, Free-Surface VOF. If your output doesn't change significantly, then use the more robust and faster default option.

☐ Check sensitivity to the momentum advection method

Higher-order momentum advection methods are generally more accurate, but may be less stable and/or generate more noise in the solution.

- ☐ If your model has a free surface, try 2nd-order, monotonicity-preserving momentum advection. If there is no free surface, try pure 2nd-order momentum advection.
- ☐ Compare the results against the default 1st-order method, and if the results are significantly different use the higher-order method.

Special considerations:

- ☐ 1st-order should be used with wave boundaries (along with Split Lagrangian VOF).
- ☐ There is some anecdotal evidence that 1st-order momentum advection may give more accurate diffusion for highly turbulent flow than 2nd-order, monotonicity-preserving momentum advection. This has not been demonstrated for all cases.

☐ Check sensitivity to recommended implicit methods

Implicit methods are indicated whenever they will result in a significant speed-up of the run time. The *mentor*, for example, suggests an implicit method when the speed-up will be at least a factor of 10. If an implicit method will not result in a significant speed up, then explicit methods are preferred because they are more reliably accurate. Some physics models (*e.g.*, *Moving Objects*) may need implicit methods for stability.

- ☐ Compare results from indicated implicit methods to results from explicit methods.
- ☐ If the results are sufficiently similar and the implicit method(s) are noticeably faster, consider using the implicit method for the rest of the project.

☐ Check sensitivity to boundary conditions

- ☐ Move the boundary location (mesh block extents) away from the region of interest and re-run the coarse case until the results do not vary significantly.
- ☐ Each expansion should be an even multiple of the cell size in that direction. Check for each expansion that the cell size remains the same.
- ☐ Take the smallest model that is free from boundary effects as the working case.
- ☐ Turbulence at inlet boundaries can affect some models. Try a large turbulence intensity (*e.g.*, 5%) at the boundary to see if it has an effect. If it does, use values that correspond to physically-realistic turbulence intensities at the boundaries.

☐ Check results sensitivity to physics model coefficients

Physics models may include *numerical coefficients* (as opposed to physically-based *input values*) that affect the results. The default settings of these coefficients are often acceptable, but can be calibrated to better match experiment.

- ☐ For example, the two-equation turbulence models use a purely numerical *maximum turbulent mixing length* parameter (*TLEN*). Test the sensitivity of the results to constant vs. dynamic *TLEN*. Start with 7% or 10% of the controlling length scale (often max fluid depth) and increase the value (*e.g.*, to 15%, 30%, 50%, 100%) until the results cease to vary significantly. Take the setting that gives results that are closest to the experiment.
- ☐ Repeat the process for any other empirically-based physics models that have numerical knobs that can be shown to affect the model results.

Minimize & quantify the mesh dependency

Your model is working as expected and has been proven to run. Now you can begin running the case on more accurate meshes. Running the same case with different mesh resolutions demonstrates that the solution converges when the mesh and time step are refined. This step is sometimes called *solution verification* and is discussed at length in ASME (2009) and (ASCE) 2009. This step is optional, but highly recommended when time and resources allow it.

☐ Add copies or restarts with refined meshes

The easiest way to do this is to refine the cell sizes of *all mesh blocks*, in *all directions*, by the *same factor*. A refinement factor of 2 is common (increasing the cell count by a factor of 8), but any ratio larger than 1.3 is acceptable. A *minimum* of three grids is recommended.

- ☐ Create a copy or a restart copy of the simulation. If it ran to steady state, then a *restart* copy with a shorter finish time is recommended for faster results.
- ☐ Refine the mesh uniformly in all directions by the selected refinement factor.

- ☐ Adjust the finish time (if desired). The adjustment (if any) depends on (1) how long it took the coarse case to reach steady state and (2) if the new simulation is a *restart*.
- ☐ Repeat the process to create a third case with an even finer mesh. The third case may be set up as a *restart* from the second case even if the second case has not run yet.

☐ Run and post-process the finer-mesh cases

- ☐ Set Simulate > Set Max Simultaneous Simulations = 1 if the refinements are *restarts*.
- ☐ Add all the simulations to the run queue and run them in order.
- ☐ Post-process the results. Use the same variables of interest as in the coarse test case.

☐ Calculate the mesh-sensitivity of the results

- ☐ The results should be converging on a constant value. Demonstrating this means the solution is "verified". See ASCE (2009) and ASME (2009) for details and calculations.
- ☐ A simple rule of thumb: the model is "mesh-converged" when refining the mesh causes the output to change less than 3% (for output with magnitude greater than 1000) and less than 1% (for output magnitude less than 1000). This is a general guideline, and not always appropriate. Use more rigorous methods if in doubt.
- ☐ The design mesh is either the converged mesh or the finest mesh that is feasible.

Quantify the validation uncertainty

This step quantifies *input*, *numerical*, and *experimental uncertainty* for cases that involve experimental data. This step is optional. It is included here because it results in 95% and standard-deviation confidence intervals which are more defensible than simple error comparisons. The interested user is referred to ASCE (2009) and ASME (2005, 2009) for guidance and details.

☐ Numerical uncertainty

The results from mesh-dependency study described above can be used to calculate *numerical uncertainty*, *order of convergence*, *Grid Convergence Index*, and *extrapolated results*.

☐ Input uncertainty

Input parameters represent physical reality, which is to some extent unknowable. The way to deal with this is to quantify the uncertainty of the CFD results due to the uncertainty in the input parameters. This means estimating the distributions of likely input values, picking values from the distributions, and recording and interpreting the results when those values are used. This is time-consuming. It is preferable to find input uncertainty using the design mesh, but using a coarse mesh is much faster.

☐ Experimental uncertainty

You must make some estimate of the experimental data uncertainty whenever you want to compare CFD results to experimental results. Standard methods exist for quantifying experimental uncertainty; most involve estimating *measurement uncertainties* and combining

them to find *result uncertainties*. Detailed experimental uncertainties are ideal, approximate evaluations (*e.g.*, ±2%) are common in practice.

☐ Validation uncertainty

Validation uncertainty is the *estimated standard deviation of the parent population of the combination of errors* (ASME 2009), that is, a function of numerical, input, and experimental data uncertainty. The uncertainty u_{val} can be compared to the disagreement $|E|$ between the CFD model and the experiment:

- ☐ Wherever u_{val} is greater than or close to $|E|$, the disagreement is within the level of "noise" of the experiment, and the model can be considered "good".
- ☐ When the disagreement $|E|$ is greater than u_{val}, it suggests that the model setup could be improved (*e.g.*, by calibrating mesh-dependent parameters as described below) or that the CFD itself is of limited reliability (*e.g.*, turbulence models).

Calibrate mesh-dependent input parameters

CFD models often include some variables that must be calibrated *after* the design mesh is selected. An example is component surface roughness: when the physical roughness is used the results become realistic as the mesh is refined, but usually require resolutions that are not time-feasible. The usual working method is to instead use either the converged mesh or the best possible mesh and then *calibrate* component roughness to whatever value minimizes the difference between CFD and experiment. This value may not be physically realistic!

☐ Select parameters to calibrate or keep fixed

When possible, adjust only one parameter for each physics model (e.g., *surface roughness* for *turbulence*, *Critical Shields Parameter* for *sediment scour*, *rate coefficient* for *air entrainment*). Numerical options that were justified in previous steps should stay fixed.

☐ Vary the calibration parameters and record the results

When possible, adjust only one parameter per run. Use restart simulations to minimize the run time, if possible. Record the results and compare them to the experiment.

☐ Check for trends

This helps reduce the amount of guesswork that goes into selecting the next calibration value. For example, results for three roughness values can be fit logarithmically to predict the value that will minimize the model/experiment disagreement.

☐ Keep the values that minimize disagreement

It's important to remember that these design values are valid only for the given mesh, geometry, flow velocity, and dimensionless numbers (e.g., *Re*, *Fr*, *We*, k^+). The parameters may need to be recalibrated for each new case, unless they are similar. If the calibrated values are far from physically-realistic values, be prepared to explain why (*e.g.*, the physics model used is

mesh-dependent, and a physically-realistic input value requires a mesh resolution that is not feasible for the project time constraints).

Model the case of interest

Use everything you have learned from the test case.

- ☐ **Start with the coarsest-possible mesh**
- ☐ **Follow the simulation setup guide (on this site) and prove the case runs**
- ☐ **Use the numerical options and input parameters from test case**
- ☐ **Refine the mesh to minimize error**
  - ☐ If the test case and case of interest are of similar scales and dimensionless numbers, use the design mesh from the test case immediately.
  - ☐ If no test case was run first, or if the test case is significantly different from the design case, then perform a mesh dependency study as described above for the test case.
- ☐ **Post-process the results and check that they appear reasonable**

Report results

- ☐ **Describe the physical situation and why CFD modeling is needed**
- ☐ **Describe the test case & report results**
  - ☐ Describe the physical situation and experimental data.
  - ☐ List any non-standard numerical methods (and note if only default values were used).
  - ☐ Describe the selection of boundary conditions and their locations.
  - ☐ Describe the mesh dependency study, including sizes used and convergence results.
- ☐ **Describe the case-of-interest & report results**
  - ☐ Report model options, whether they were the same or different as in the test case.
  - ☐ Report results for the variables of interest from the design mesh resolution.
  - ☐ Justify the design mesh resolution (e.g., mesh dependency results or prior experience).
  - ☐ Report any conclusions drawn or lessons learned from the simulation.
- ☐ **Report validation uncertainty (optional)**
  - ☐ Report estimated *numerical*, *input*, and *experimental uncertainty* and methodology.
  - ☐ Report any additional calculated verification and validation results (*e.g.*, GCI).
  - ☐ If no uncertainties were estimated, justify the omission. Rigorous uncertainty estimation is an ideal that is not always necessary or feasible in practice (*e.g.*, when CFD is used to qualitatively compare potential designs).

Guide 2: CFD *Simulation* Workflow Guide

Simulation Setup Guide

Table of Contents

This setup guide is not an exhaustive decision-tree for all elements of a simulation. Rather, it is meant to be a checklist with a middling level of detail. Some possible model elements have been omitted (electromechanical physics models, for example). The Flow Science support staff hopes you find this guide helpful. Please send suggestions to support@flow3d.com.

Before starting

☐ Draw and annotate a diagram of the physical problem

The diagram should include:

- ☐ A consistent unit system (*e.g.*, SI, CGS, slug/foot/second) that matches the scale of the problem (very small and very large numbers can increase rounding error),
- ☐ Geometric dimensions,
- ☐ Locations of fluid and free surfaces,
- ☐ Notes on body forces (like gravity) and boundary forces (like shear stress),
- ☐ All heat and fluid fluxes and sources,
- ☐ All solid moving objects and their velocities, locations, and travel paths
- ☐ Unknown variables and the locations where you want to them.

The completed diagram should be an sketch of the physical problem that includes all of the relevant engineering information.

☐ Determine the general modeling approach

Decide how to approach the problem using the annotated diagram as a guide. Record your selections on the diagram as you proceed.

- ☐ Label the fluids in the problem (*e.g.*, air, water, oil),
- ☐ For liquid/gas systems, can local gas velocities be ignored?
  - Yes: consider a one-fluid model: decide if the gas pressure and temperature change with time (meaning one of the bubble & phase change physics models is needed).
  - No: consider a two-fluid model and decide if the gas can be treated as nearly incompressible or must be treated as fully compressible (costs more resources).
- ☐ Can a portion or all of the flow be treated as depth-averaged? If yes, consider using shallow water physics in one or more mesh blocks
- ☐ Does a fluid have a sharply-defined interface (free surface)?
  - Yes: consider free surface or sharp interface tracking
  - No: consider no sharp interface tracking (fluid separation is diffuse or the whole domain is submerged)
- ☐ Is the flow viscous? If yes, consider using viscous flow physics (almost always used).

☐ Determine the physics packages that are part of the modeling approach

Now decide which physical mechanisms are important. Add the relevant engineering information for each of these options to the diagram. Try to justify why physical mechanisms were included or ignored – this helps catch mistakes early. A common checklist is given below.

- ☐ Can gravity and reference-frame motion be ignored? If not, consider gravity and non-inertial reference frame physics.
- ☐ If heat transfer can't be ignored, consider heat transfer physics, and make fluid-to-solid heat transfer selections below:
  - Can all solid components be treated as having a constant, uniform temperature? If yes, consider no fluid-to-solid heat transfer.
  - Can all solids be treated as having a "lumped" (spatially-uniform) temperature? If yes, consider uniform component temperatures.
  - If some solids must have temperature that vary in space due to initial condition but are constant in time, consider the non-uniform, constant temperature option.
  - If some solids must have temperatures that vary in both space and time, consider the full energy equation option.
- ☐ If a fluid varies in density (*e.g.*, due to temperature gradients, salinity gradients, contaminants, or entrained air), consider variable density physics.
- ☐ If a fluid contains other phases (*e.g.*, contaminants, particles, or tiny bubbles):
  - consider scalar physics if the contaminant(s) can be treated as continuous concentrations,

- if a contaminant is added at free surfaces consider air entrainment physics or defect tracking physics,
- if contaminants should be treated as a finite number of particles (with or without drag) and not as continuous concentrations, consider particle physics,
- if they should be treated as continuous concentrations that *represent* many discrete particles (with drag effects), consider drift-flux physics,
- if they should be treated as packable (dense) concentrations that can take solid forms, consider granular flow physics, sediment scour physics, or dissolving object physics.

☐ Are there two pure-phase liquids with different densities and viscosities along with a free surface and a gas phase? If yes, consider using the two-liquids and free-surface workaround (set *RHOFS* & *MUS* as the density and viscosity of the second liquid in the *prepin* file namelist *$PROPS*, and then set all initial and boundary condition densities as liquid mixture densities between *RHOFS* and *RHOF*).

☐ Estimate the Weber number. If *We* is less than or close to 1, consider surface tension physics (computationally expensive).

☐ Estimate the Reynolds number *Re*. Decide if turbulence physics are necessary. If yes, decide which turbulence model to use, estimate the roughness Reynolds number k^+, and decide if surfaces should be hydraulically smooth (*ROUGH* = 0) or rough (*ROUGH* > 0).

☐ Draw a Cartesian or Cylindrical coordinate system on the diagram

☐ Decide whether to use Cartesian or Cylindrical coordinates based on the flow geometry.

☐ Decide where to place the origin.

☐ Draw the axes. Gravity is usually in the negative z direction, because some boundary conditions and hydraulic data output require it. Try to avoid placing axes that are mostly diagonal to the flow.

☐ Draw the computational domain on the diagram

Computational domains (mesh blocks) are 3-D rectilinear or axisymmetric cylindrical wedges. Their edges are always parallel to the coordinate system. They may be nested or linked.

☐ Sketch the block edges where they are not in regions of rapid change (high gradients of pressure, velocity, or temperature).

☐ If possible, sketch a single block. This minimizes interpolation error. Consider using automatic subdivision (available in v11) to locally resolve the mesh.

☐ Minimize the number of cells where there will be no flow. Consider adding domain-removing components to your sketch to turn off unused cells.

☐ If you know already that linked or nested blocks will be necessary to resolve geometry or flow features, sketch them now. Use as few blocks as possible.

☐ Special consideration: if you (a) have more than one mesh block, and (b) plan on using regularly-sized cells in each (no mesh stretching), then place the edge coordinates of

each block so that the x, y, and z distances between them are even multiples of the largest cell length of the same direction in all blocks. This will make perfectly aligned gridlines between blocks and minimize interpolation error.

- ☐ Write the edge coordinates of the blocks.

☐ Draw the boundary conditions on the diagram

Write which boundary condition type best represents the physical situation at each boundary. If none of the available boundary condition types are a reasonable approximation of the physical situation at a boundary then that boundary must be moved elsewhere.

☐ Generate, check, and repair all *.stl* files

Check all *.stl* files for errors and repair them using ***MiniMagics***, ***Admesh***, ***netfabb Studio***, ***MeshLab***, and/or similar *.stl* tool programs. Common errors include inverted normals, zero-area facets, non-manifold edges, and holes. These may not be detected by ***FLOW-3D*** but can cause bad solutions.

On the Model Setup: General tab

☐ Add some notes

Notes help explain what was done and why for future users or for later reference. Describe the problem, the purpose of the simulation, the case number, the approach, etc.

☐ Choose solver precision and number of processors to use

A good practice is to leave at least one core (either 1 or 2 processors, depending on whether hyper-threading is active) for the operating system and background programs. Another common practice is to use an even number of cores to reduce computational overhead.

☐ Select number of fluids, interface tracking, and flow mode options

Use the annotated diagram as a guide.

- ☐ The number of fluids refers to where momentum equations are to be solved. In one-fluid models momentum is solved in regions where $F > 0$. In two-fluid models, $F = 0$ represents *Fluid 2*, and momentum is solved in all open and partially-open cells.
- ☐ The interface tracking options define whether changes in the fluid fraction should be sharp or diffuse. The sharp interface option tracks the location and orientation of the fluid interface within cells, and permits free surfaces in one fluid models. The no sharp interface option treats the interface between fluids as a mixture, and forces all open volume to be full of fluid at all times.
- ☐ The flow mode option applies only to two-fluid problems. It treats regions where $F = 0$ as either compressible (ideal gas law) or incompressible.

☐ Define the finish condition(s)

Set the finish time. Additional finish conditions can be based on time, fill fraction, or steady-state criteria.

☐ Define how to restart the simulation from existing results (optional)

Restart options allow initialization from an existing results file. The restart options define what information is taken from the results and what information is reset to the current simulation initial conditions.

On the Model Setup: Physics tab

☐ Activate the relevant physics models based on the annotated diagram

- ☐ Be aware of dependencies: some physics models can't be activated without others.
- ☐ On the diagram or an attached sheet, list all empirical values and coefficients used.
- ☐ Have a rationale for each value on the list.
- ☐ Make notes of any values that you are unsure of.

On the Model Setup: Fluids tab

☐ Define the properties of fluid 1 and (if used) fluid 2

Define the appropriate physical properties for the active physical models using the annotated diagram as a guide. Parameters that are associated with inactive physics models will be grayed out.

☐ Define any necessary properties of the interface

Define the properties of the interface between regions of $F = 1$ and $F = 0$. These include the properties for surface tension physics, phase change physics, and two-fluid diffusion.

☐ Record values their rationales/sources/citations

As before, note any values on the diagram that you are uncertain of.

On the Model Setup: Meshing & Geometry tab

☐ Import and create geometry 

- ☐ Add subcomponents to create the solid geometry described in the annotated diagram using *.stl* files and primitives. Include any shapes that will be mass sources.
- ☐ Apply transformations to the subcomponents: magnifications, rotations, and translations. The transformations will be applied in the order just described. Consider defining transformation centers for subcomponents that need multiple transformations.

- ☐ Associate the subcomponents with components. Define components so that all their subcomponents share the same material properties (e.g., surface, solid, and motion properties). Give descriptive names to each component.
- ☐ Define domain-removing components to block off open regions where flow will *never* occur, and where the open void does not need to vent to a boundary condition. This reduces the active cell count, and makes the simulation faster. If in doubt, skip this step.

☐ Define the mesh 

- ☐ Following your diagram, create the mesh blocks and specify their extents.
  - Specify the Cartesian or Cylindrical coordinate system.
  - Check that linked blocks have edges that align exactly.
  - Check nested blocks are inside their containing blocks (they may share edges).
  - Add mesh planes in containing and linked blocks to match the edge extents of their nested or neighboring blocks.
  - Check that inter-block boundaries are not located in high flow-gradient regions.
- ☐ Specify target cell sizes or counts for each block and direction. Consider the following:
  - Cubic cells are the most accurate.
  - The ratio between the longest and shortest side of each cell should be less than 3:1. Exceptions increase uncertainty in the results, but may be necessary.
  - All solid cross-sections should be resolved by at least 2 or 3 cells.
  - Fluid cross-sections should be resolved by at least 3 or 4 cells. Accurate results often require 10 to 12 cells across the thinnest flow section.
  - If necessary to get these resolutions, add mesh planes, mesh plane cell counts, and mesh plane cell sizes. This will make the mesh irregular (stretched). The stretching ratio between adjacent cells should be less than 1.25 or 1.3.
  - As much as possible, the cell face size in nested blocks should be ½ that of the cells in the containing block. This minimizes interpolation error.
  - Adjust mesh block edge locations so that the side-to-side total length of *each block* is an even multiple of the largest expected cell length in *all blocks* in the same direction. This ensures that the gridlines of regular (not stretched) cells in neighboring blocks will line up exactly, minimizing interpolation error.

☐ Check the mesh and geometry resolution

- ☐ Check the mesh:
  - Visually check the grid line matching between neighboring mesh blocks.
  - Check Mesh Information for each block, especially total number of cells, maximum adjacent cell size ratio, and maximum aspect ratios.
  - Check Mesh Block Summary Information for total number of real cells in all blocks. Check the following rules of thumb (for single-machine parallel processing):
    - cell counts in the hundreds will run in seconds
    - cell counts in the thousands will run in minutes

- hundreds of thousands of cells run in hours or days
- millions of cells take days or weeks to run

☐ ***FAVORize*** the domain and check the geometry. It should appear well-resolved, especially around gaps and thin sections.

☐ If there are multiple mesh blocks, preprocess the simulation and check *Diagnostics > Preprocessor Summary* (*prpout.xxx*) for the search term 'mismatch' to find the difference in resolved open area on either side of mesh block interfaces. The total error should usually be much less than 5%.

☐ Adjust and check the mesh until the following criteria are satisfactory:

- geometry resolution
- resolution of flow sections
- total cell count (as a function of desired run time)
- grid line matching between neighboring blocks
- aspect ratios for single cell sides
- aspect ratios between neighboring cells (due to stretching)
- aspect ratios between cells in neighboring blocks
- Open area mismatch (from the preprocessor)

☐ Make note of the meshing methodology and rationale.

☐ If moving solids with prescribed motion are used, consider running the simulation without fluid to check the movement definition.

☐ Define all necessary component properties

Follow the annotated diagram and update it where necessary.

☐ Go through the list of Component Properties for each component and identify the available selections. Grayed-out selections require physics that have not been activated.

☐ Hold the mouse cursor over selection boxes to find the variable code for unfamiliar available selections. Search the User Manual's Input Variable Summary chapter for the variable code to identify the definition, default value (if left blank) and units of the selection.

☐ Note a rationale for each value entered or left at default setting. The goal is to be able to explain each property value used by the selected physics of the simulation.

☐ Define mass source flow rates and properties and moving object motion types and properties, if present.

☐ Add and define springs and ropes and valves and vents

☐ Spring/rope objects connect moving components to each other or to stationary anchor points. Identify attachment locations for each spring/rope.

☐ Identify the spring/rope type to be used (see *User Manual* for details)

☐ Identify the Hookean spring coefficient for each.

☐ Give each spring/rope a descriptive name, and mark it on the diagram.

- ☐ Valves allow gas (aka "void") bubble pressure to escape the domain in a realistic way. Identify the location point of each. They must be in open areas, not solids.
- ☐ Estimate the valve loss coefficient and external pressure for each valve (see *User Manual*).
- ☐ Give each valve a descriptive name, and mark it on the diagram.

☐ Add and define baffles and flux surfaces 

Follow the annotated diagram to locate solid (flow-deflecting) baffles and measurement planes.

- ☐ Each baffle region has one definition and as many limiters and transformations as necessary.
- ☐ Baffles will be moved to the nearest grid line during preprocessing. Try to place baffles normal to the axes to avoid "stair-stepping".
- ☐ Specify baffle porosity (0 = solid to 1 = transparent to flow).
- ☐ If the baffle is porous (0 < porosity < 1), specify baffle drag coefficients.
- ☐ Specify baffle heat transfer coefficients (when heat transfer physics are active)
- ☐ Activate the flux-surface option for baffles that will measure flow properties.
- ☐ If particle physics are activated, define particle-counting bins manually in the prepin file namelist $PARTS (see *User Manual > Input Variable Summary and Units chapter > Particle Setup: Flux Surface and Sampling Volume Data section*).

☐ Add and define history probes  and sampling volumes 

Follow the annotated diagram for locations to measure output variables and/or check known variables.

- ☐ Give each probe and sampling volumes a descriptive name for post-processing.
- ☐ Define locations and coupled motion parameters for history probes.
- ☐ Define particle counting bins for sampling volumes (see flux surfaces section above).

☐ Add and define mass-momentum sources 

Mass-momentum sources are boundary conditions located inside the mesh (as are mass sources, valves, and vents). Use the annotated diagram as a guide.

- ☐ Specify each mass-momentum source's shape. The source is defined around the origin, with flow vector pointing toward z-max.
- ☐ Rotate the mass-momentum source to get the desired flow direction normal.
- ☐ Translate the mass-momentum source to the desired location in the domain.
- ☐ Specify source, sink, and flow rate properties.
- ☐ Specify motion properties only if the source moves during the simulation.

☐ Define all boundary conditions 

Follow the annotated diagram. Check the following:

- ☐ Boundaries should usually be located far enough from the region of interest that increasing the boundary distance further does not change the results in the region of interest. This can be checked later with multiple versions of the same simulation set up.
- ☐ Boundary types should be appropriate for the situation. For example:
    - Pressure boundaries set reference pressure and free surface elevation, temperature, contaminant concentration, etc. These flow parameters are communicated into the domain. In free-surface flows, Pressure boundaries are appropriate where the Froude number $Fr \leq 1$. They do not specify velocity (except for tangential components).
    - Outflow boundaries do not communicate any information into the domain, unless they unintentionally reflect angular momentum. Free-surface elevation cannot be specified at Outflow boundaries, which are appropriate where $Fr > 1$.
    - Velocity and Volumetric-flow-rate boundaries apply a uniform velocity below a free surface elevation, and do not specify pressure.
    - Grid-overlay boundaries are used to interpolate a flow solution from a previous simulation as a spatially-varied, time-constant velocity profile. They do not specify pressure.

☐ Add and define fluid and void/gas initial conditions

Initial conditions prescribe the solution everywhere in the domain at t = 0, and set up the entry-values for all equations: flow velocity, heat transfer, electric potential, contaminant concentration, etc.

- ☐ Follow the annotated diagram and specify all available initial conditions.
- ☐ If gravity physics are active, consider the hydrostatic pressure distribution option, which will also apply to boundary conditions.
- ☐ Check that fluid initial conditions, boundary conditions, and component initial conditions match. Check velocity, free-surface elevation, temperature, and turbulence.
- ☐ Check that the initial conditions are good approximations of reality.
- ☐ Consider adding initial fluid regions to minimize splashing on solid surfaces.
- ☐ Note the rationale for each initial condition, including default values.

On the Model Setup: Output tab

- ☐ **Choose the basis for output frequency (time, fill, or solidified fraction)**
- ☐ **Choose additional output of interest**
- ☐ **Choose selected data to output more frequently**
- ☐ **Define output intervals for all data types**

  The default intervals are fractions of the finish time: $\frac{10\ outputs}{simulation\ end\ time}$ for restart and long print data and $\frac{100\ outputs}{simulation\ end\ time}$ for selected, history, and short print data. Specified intervals are in units of time between outputs.

On the Model Setup: Numerics tab

The numerical options in ***FLOW-3D*** are intended for advanced users and give significant control over numerical methods used to solve the governing equations. When used improperly some of these options can cause problems with the solution. In general, do not change these settings without understanding what the adjustment does and having justification for the change. Some calibrations are discussed in the General Project Workflow Checklist. See User Manual > Troubleshooting for more information on adjusting numerical methods for specific problems.

Final Check after setup

- ☐ **Check that the simulation pre-processes without crashing**
- ☐ **Check the pre-processed output in 3-D**
  - ☐ Do the fluid and geometry locations and resolutions "look right"?
  - ☐ Are all specified baffles in the expected location? Are they acceptably resolved?
- ☐ **Check the pre-processed output in 2-D slices**
  - ☐ Are fluid and geometry initialized in the correct locations?
  - ☐ Is geometry well-resolved? Are there sliver cells?
  - ☐ Are all initial fluid values (*e.g.*, depth, velocity, pressure, density) as expected?
  - ☐ Do initial fluid values match the boundary conditions?
  - ☐ Will there be unnecessary splashing on solid surfaces?

  Adjust the model setup if necessary and repeat the check. The preprocessed simulation results should give you a good idea of how the run will start.
- ☐ **Check the pre-processed text output files**
  - ☐ Were any errors noted in Diagnostics > Preprocessor Errors (*prperr.xxx*)?

- ☐ Check each section in Diagnostics > Preprocessor Summary (*prpout.xxx*). Some commonly checked items are given for demonstration here:
  - In section /OBS/, do specified component surface area and calculated component surface area agree reasonably well? If not, consider adjusting the mesh resolution.
  - In section summary of open areas at inter-block boundaries, is the percent difference less than 1% (ideal), 3% (typical) or 5% (usually the max acceptable)? If not, consider adjusting the mesh resolution.
  - In section /PARTS/, were any fluid history probes eliminated unexpectedly? If yes, consider adjusting the history probe location.
  - In section /PARTS/, is the total number of particles generated less than the maximum number of particles requested (max possible)? Will the simulation run out of available particles during the run?

Run the simulation

☐ Check solver text output and solver warnings 

- ☐ What are the warning messages?
- ☐ Is the time step reasonable? Will the simulation finish in your lifetime?
- ☐ What solution is limiting the time step? Does the mentor suggest an implicit method?

If necessary, refer to troubleshooting.

Exercise 1: Running & Post-Processing an Example

Exercise 1: Running & Post-Processing an Example

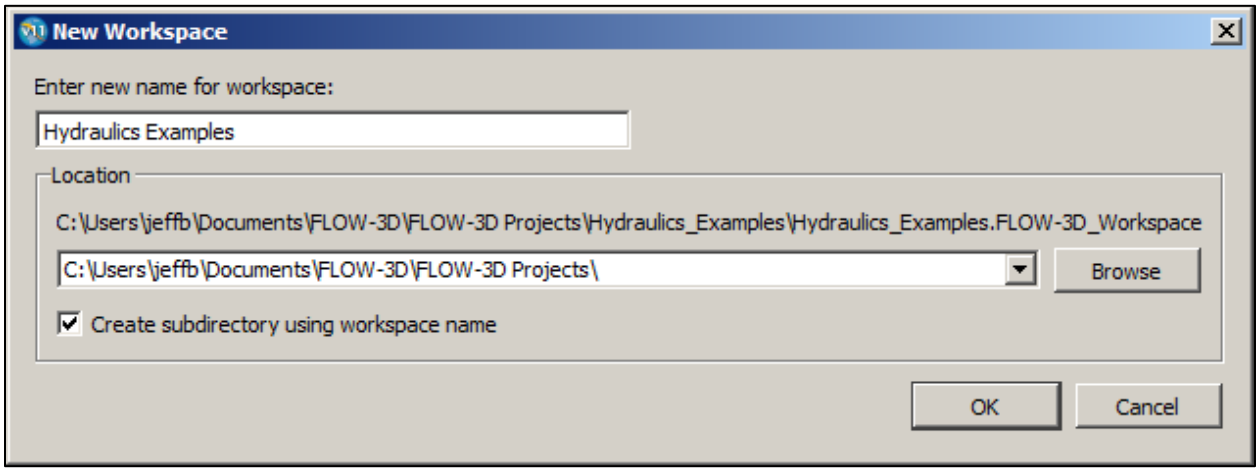
In this exercise, you will learn how to use ***FLOW-3D***® and its native post-processor to:

1. Create a new workspace and add an example project file included with the software,
2. View the geometry and mesh,
3. Select and add parameters to be output more frequently than the default,
4. Preprocess and run the simulation and interpret the runtime output,
5. Load results files,
6. Understand the options for post-processing in 1-D, 2-D, and 3-D,
7. Reflect results around a plane of symmetry,
8. Make *.avi* videos and *.bmp* images of the results,
9. Make a simulation copy and perform a restart.

The example problem is of flow over a bench-scale sharp-crested weir. Only half of the weir is modeled, and symmetry along the centerline of the weir is assumed.

Create a New Workspace

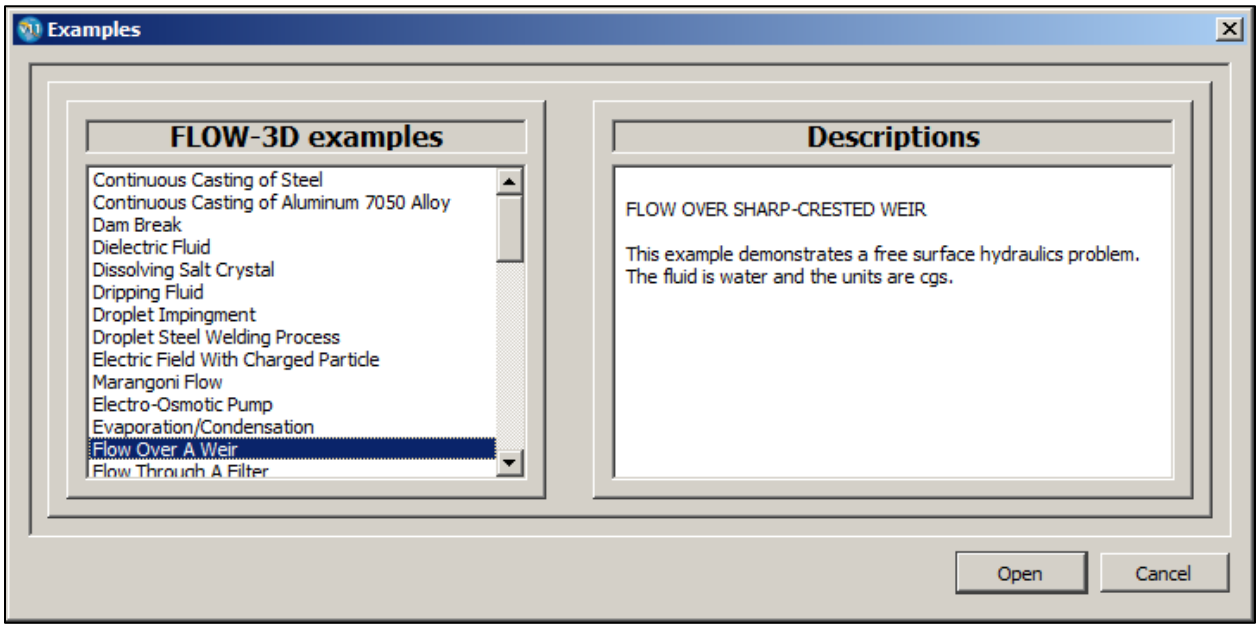
1. Launch ***FLOW-3D***® by double-clicking the ***FLOW-3D***® icon on your desktop.
2. On the **Simulation Manager** tab, create a new workspace by selecting **File > New Workspace** from the menu at the top.
3. Enter the workspace name: **Hydraulics Examples**. Keep **Create subdirectory using workspace name** checked so that the workspace and all its simulations will be under the same directory.

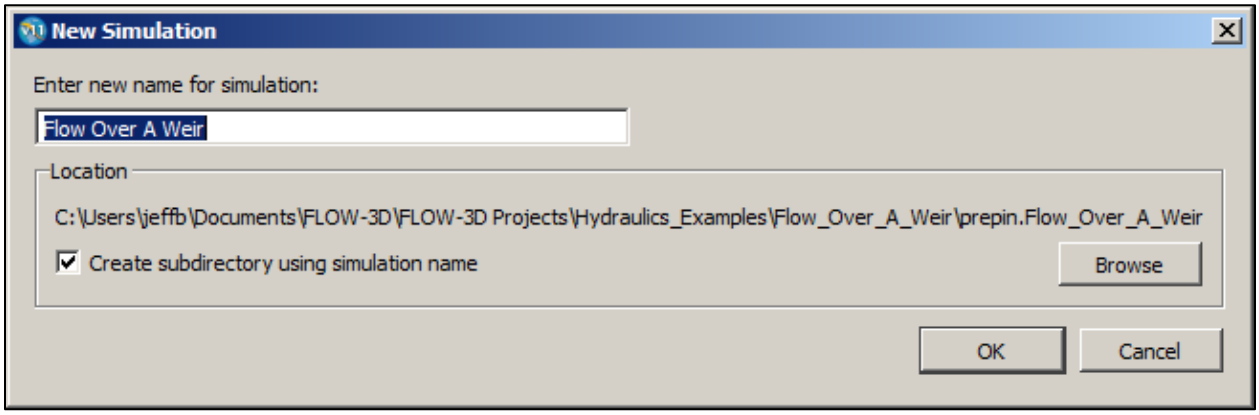


Add an Example Simulation

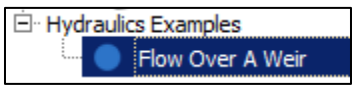
1. Select **File > Add Example**… from the top menu or right-click the workspace name and select **Add Example…** The dialog shown below will appear.

2. Select **Flow Over a Weir** from the alphabetically-ordered examples list and **Open**. Keep the default name and **Create subdirectory using simulation name** option. Click **OK** to finish importing.





3. The project has now been imported into the **Hydraulics Examples** workspace. *Load* the project by double-clicking its name:

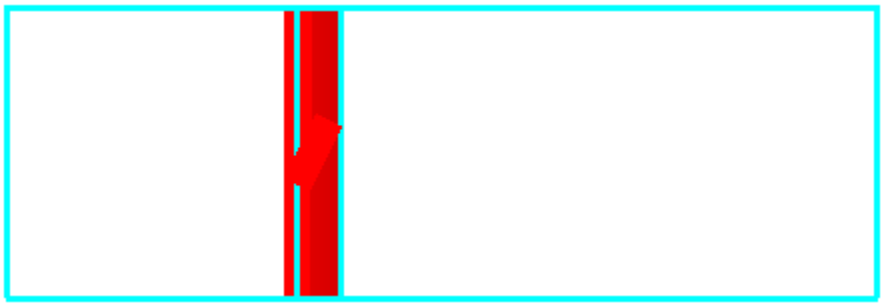


Description of the Primitive Geometry

The model is a very old validation case of a sharp-crested weir. It was built using ***FLOW-3D*** primitives (simple shapes). Some of the shapes (subcomponents) are solids, and some are holes to cut away the mitered edge. All the subcomponents are visible in this view, regardless of whether they are solids or holes. Later we will view the model as the solver will see it, and it will appear as half of a sharp-crested weir. Models are easier to visualize during setup using *.stl* geometry files, which will be covered in a later exercise.

Mouse Modes & Viewing Options in Display Panes

4. Select the **Model Setup** tab and then the **Meshing & Geometry** tab below that. Locate the large display window showing the model and domain illustrated below.



2. **Left Button – Rotate**. Click and hold the left-mouse button and move the mouse in the Meshing_Geometry window. The model will rotate accordingly.

3. **Middle Buttom or Wheel – Zoom**. Click and hold the middle-mouse button or wheel while moving the mouse vertically, or rotate the wheel, to zoom.

4. **Right Button – Pan**. Click and hold the right-mouse button and move the mouse in the window. The model will move with the mouse.

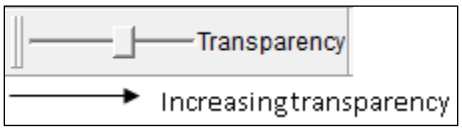
5. Experiment with the  buttons above the view pane. The first four buttons reset the view. The last button toggles between orthographic and perspective views.

6. Experiment with probing surface coordinates with *Shift+click*. When you hold the Shift key a black cross target appears (if it doesn't try releasing and re-pressing *Shift*). When you click the mouse button, the XYZ coordinate of the solid surface at that point appears in the lower-left corner of the GUI. You can access a list of probed points by selecting *View > Probe Points List…* in the menu above the view pane.

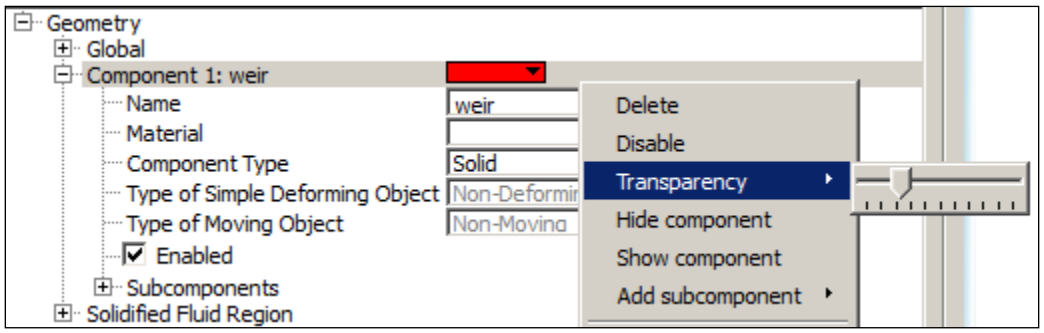
7. Experiment with setting a pivot point with *Ctrl+click* (or toggle the  button) A gray lozenge marks the pivot point, which can be hidden by toggling *View > Hide (or Show) Pivot Point* in the menu above the view pane.

8. Close the *Probe Points* list and toggle the  button to turn off the pivot point option. Click the  button to reset the view and then toggle the  button until the x-axis points to the right and the z-axis points up. You can toggle the axis and control its location by selecting the options in *View > Draw Axis* on the toolbar menu above the view pane.

9. **Global Transparency**: The **Global Transparency** slider in the toolbar controls the transparency of *all* objects in the display window.

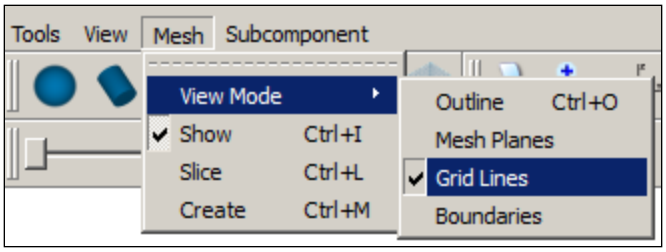


5. **Component- and Subcomponent-specific Transparency**: these can be set in the **Geometry tree** (toggle ). Right-click on a **component** name or open the **Subcomponents** list and right-click on a **subcomponent** to adjust the transparency. Try out the transparency options.

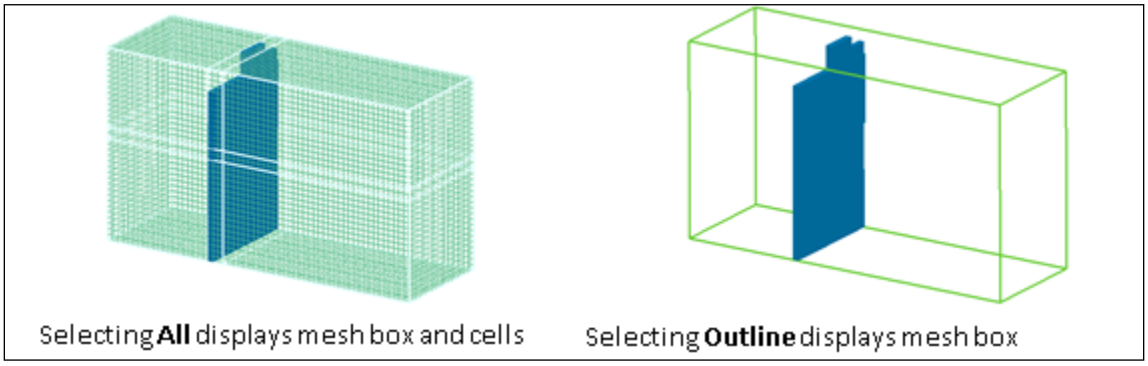


6. **Other View Options**: experiment with the following. The  button **prints** the current view, which can be used with a .pdf printer program to generate images of the mesh and geometry setup (hint: use the *Print Screen* keyboard button and a ***Paint*** program as an alternative). The  button allows you to draw a box and places a pivot point in its center before zooming to the selection. The  button fits the domain and geometry to the view window.

7. **Mesh Display**: It's important to examine the mesh to assess resolution. It's also important to be able to turn it off to see the geometry. The mesh can be viewed from the **Mesh** menu item. Check the **Mesh > Show** option so the mesh is displayed, and select **Mesh > View Mode > Grid Lines**.



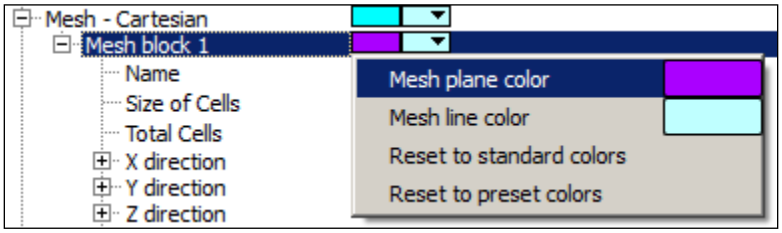
Mesh Planes (user-specified grid lines) are displayed in a different color and/or weight than the regular **Grid Lines**, which are generated automatically by ***FLOW-3D***.



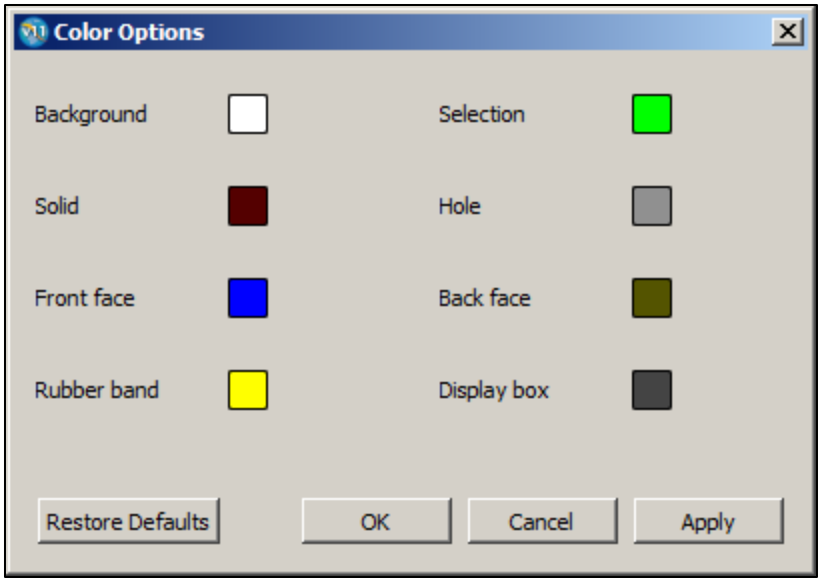
8. **Color Options**: Colors can be assigned to **components**, **subcomponents**, and **mesh blocks**. The colors are to make identifying parts of the simulation easier when editing: they do not affect the solution.

   A. Open the **Mesh tree** ( to toggle) and expand the tree with the **+ icon**.

   B. Click on the **color palette** next to **Mesh Block 1** and experiment with different grid line and mesh plane colors.



   C. Select **Tools > Options** from the menu above the display pane. On the **Color** tab, select the **Background** color swatch and make the background white. Adjust your mesh colors if necessary to be able to visualize both mesh planes and grid lines.



Assessing Mesh Resolution with FAVOR

One of the most important aspects of simulation setup is choosing an appropriate computational mesh. If the mesh is too coarse the geometry and flow may not be resolved enough and the simulation will not give good results. If the mesh is too fine the runtime may be unnecessarily long. The goal of mesh setup is to use just enough cells to resolve the geometry and the flow features of interest.

There are two ways of judging how well a computational mesh resolves the geometry. One way is to run the **preprocessor**, and this gives the most information short of running the simulation. A quicker way is to ***FAVORize*** the geometry. **FAVOR** embeds the geometry and initialized fluid in the current computational mesh using the same routines as the preprocessor.

1. Click the  **icon** in the window toolbar. The **FAVOR** dialog will appear.

2. Check that **Geometry Surface = Solid Volume** and **ISO Surface Value = 0.5** (more or less). Click **Render** to view a gray *iso-surface* fitted to the solver-embedded solid geometry.

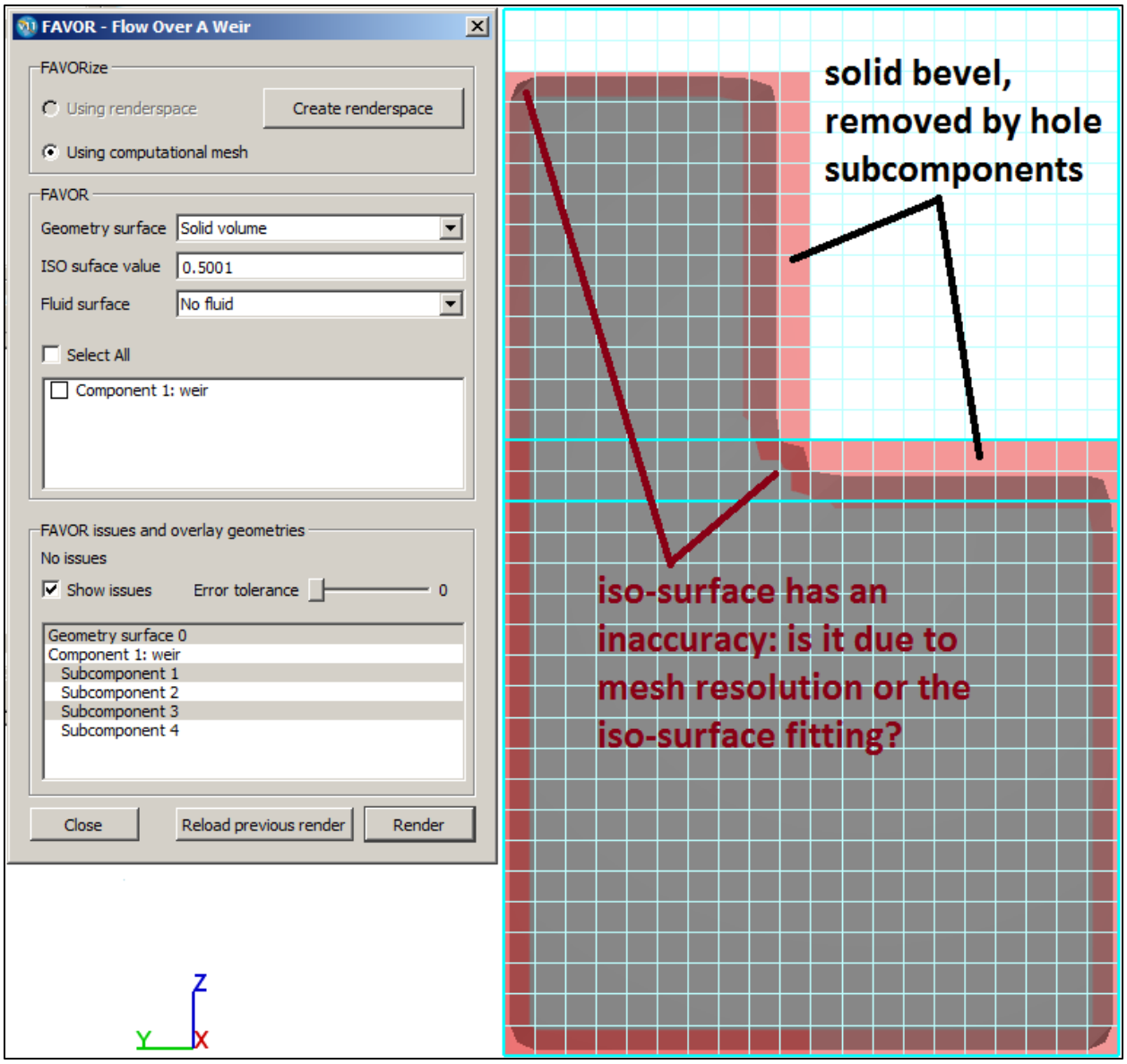
3. The checkboxes associated with individual Components allow all or some components to be ***FAVORized***. There is only one component, so ignore the checkboxes in the middle of the dialog.

4. At the bottom of the dialog, activate the **Show Issues** checkbox. Cells where ***FAVORize*** will have known problems will be highlighted in orange. Check that there are no known issues when the relative **Tolerance = 0** and **100**.

5. The component/subcomponent list at the bottom of the dialog lets you overlay the geometry that was imported with iso-surfaces that are fit to the embedded geometry. First de-select **Geometry Surface 0** to turn off the ***FAVORized*** iso-surface. Then highlight **Subcomponent 1** and **Subcomponent 3** to see the solid part of the geometry. The remaining subcomponents are holes that cut bevels into the visible subcomponents. Now turn **Geometry Surface 0** back on to see the contrast between the embedded geometry iso-surface and the original defined subcomponents. Play with **Transparency** and viewing the mesh **Grid Lines** while you're looking

at this view. Increasing mesh resolution will usually improve the match between iso-surface and original geometry. It's important to understand that the way the solver will "see" the embedded geometry most accurately viewed in *2-D plots* (which you'll do later). The 3-D ***FAVORize*** view is not precise because the iso-surface fitting is imperfect and approximate.

6. **Close** the dialog to return to the normal setup view.

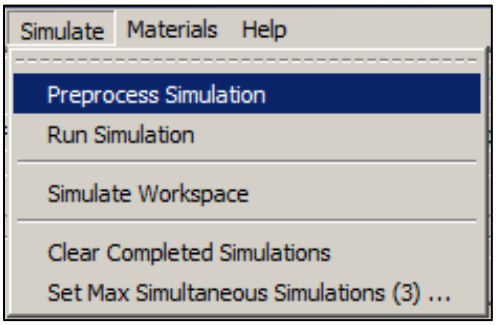


The image above on the right shows the weir structure. The sharp crest of the weir is visible and it appears to be adequately resolved. There is a rounding at the weir corner that could be examined in 2-D plots to see if it's an artifact of the iso-surface approximation or actually poor geometry resolution.

Preprocessing the Simulation

The ***FAVORize*** process gave you some good information, but there is more you want. 2-D plots are more accurate than the 3-D rendering because they do not include the approximately fitted iso-surfaces drawn by ***FAVORize*** and 3-D rendering. 2-D plots use cell-by-cell *area* and *volume fractions* to give the precise surface locations computed by the solver.

1. Select **Simulate > Preprocess Simulation > Local** from the top menu bar.



2. The interface will prompt you to **Save** the ***FAVORize*** selections you made (click **Yes**) and will then switch back to the **Simulation Manager** and the preprocessor will run. It should complete within a few seconds, and a message will appear in the **Solver text** indicating "Preprocessor Done".

3. In the **Solver text**, scroll up and find the **total, fluid, and solid sub-domain cell counts**. The total and fluid counts are both around 50,000 cells. This is important information, since the number of active cells directly relates to the simulation run time.

Loading a Results File

1. The preprocessor generates a results file named ***prpgrf."project name"***. To load the results file, select the **Analyze tab.**

2. If no results file is currently loaded you will be prompted to select a file. Keep the default **Custom** option**. Custom** file types are *flsgrf.xxx* (created by the solver during a run) and *prpgrf.xxx* (created by the preprocessor) that require you to select output options yourself. **Existing** file types are *flsplt.xxx* and *prpplt.xxx* which contain non-alterable plots that are pre-defined in the prepin file (i.e., the simulation instructions). The prp prefix means a file is preprocessor output, and the fls prefix indicates flow solver output.

3. Select ***prpgrf.Flow_Over_A_Weir*** in the dialog box, and **OK** to load the file.

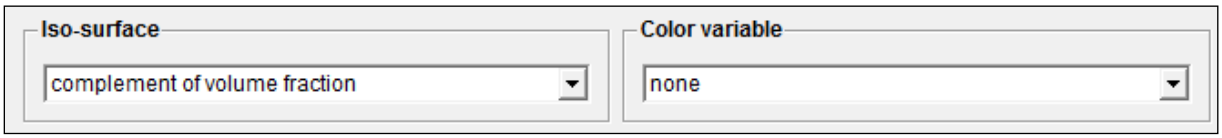
Viewing Preprocessed Geometry and Initial Fluid Configuration in 3-D and 2-D

The **Analyze** panel will now be displayed. Although all sub-tabs are available, typically only 2-D and 3-D plots are necessary to check the model setup.

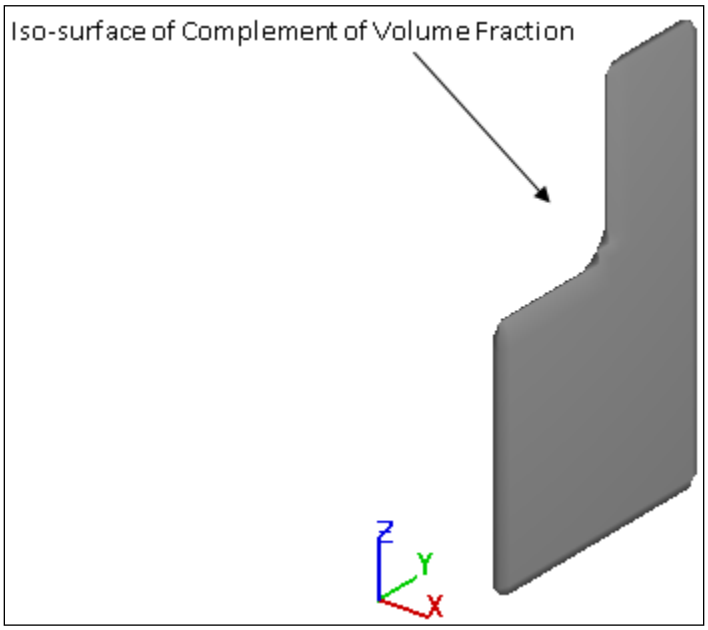
The **3-D subtab** will be displayed initially when the **Analyze** tab is selected. First, you'll generate the same display that was generated by the **FAVORize** function.

1. On the **Analyze > 3-D** tab, find the **Iso-surface** dropdown. Select **Complement of Volume Fraction** so the *surface* that is plotted is the *interface* between *solid geometry* and *open space*.

2. Find the **Color Variable** dropdown, and select **None**. You do not need to color the surface by a flow parameter right now.



3. Select **Render** in the lower right corner of the screen. The tab at top will change to the **Display** tab, and will show the same image as you saw in the **FAVORize** display.



The next step will be to generate a 2-D plot along the weir centerline to show the initial fluid pressure and location with the mesh.

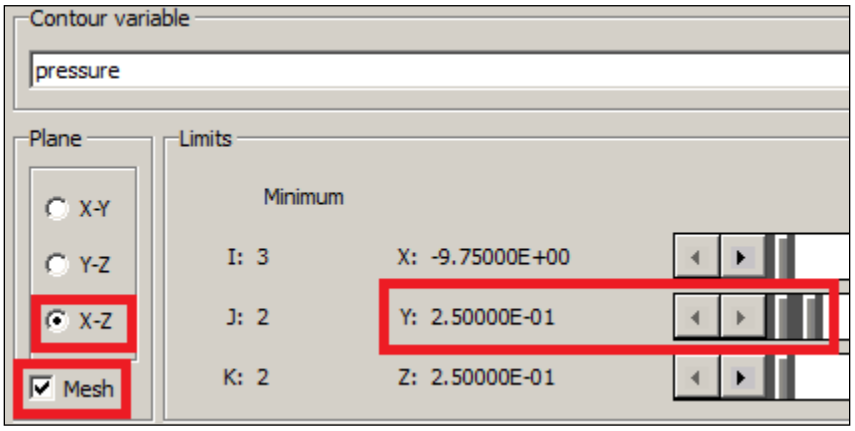
4. Select the **Analyze > 2-D** tab.

5. Select the **X-Z** radio button in the **Plane** group. Note that the sliders in the center of the screen reset so that the full extents of the X and Z domain is plotted, and the **Y sliders** come together so that by default only one slice is displayed (at y = 5.25, the geometric center of the y-domain).

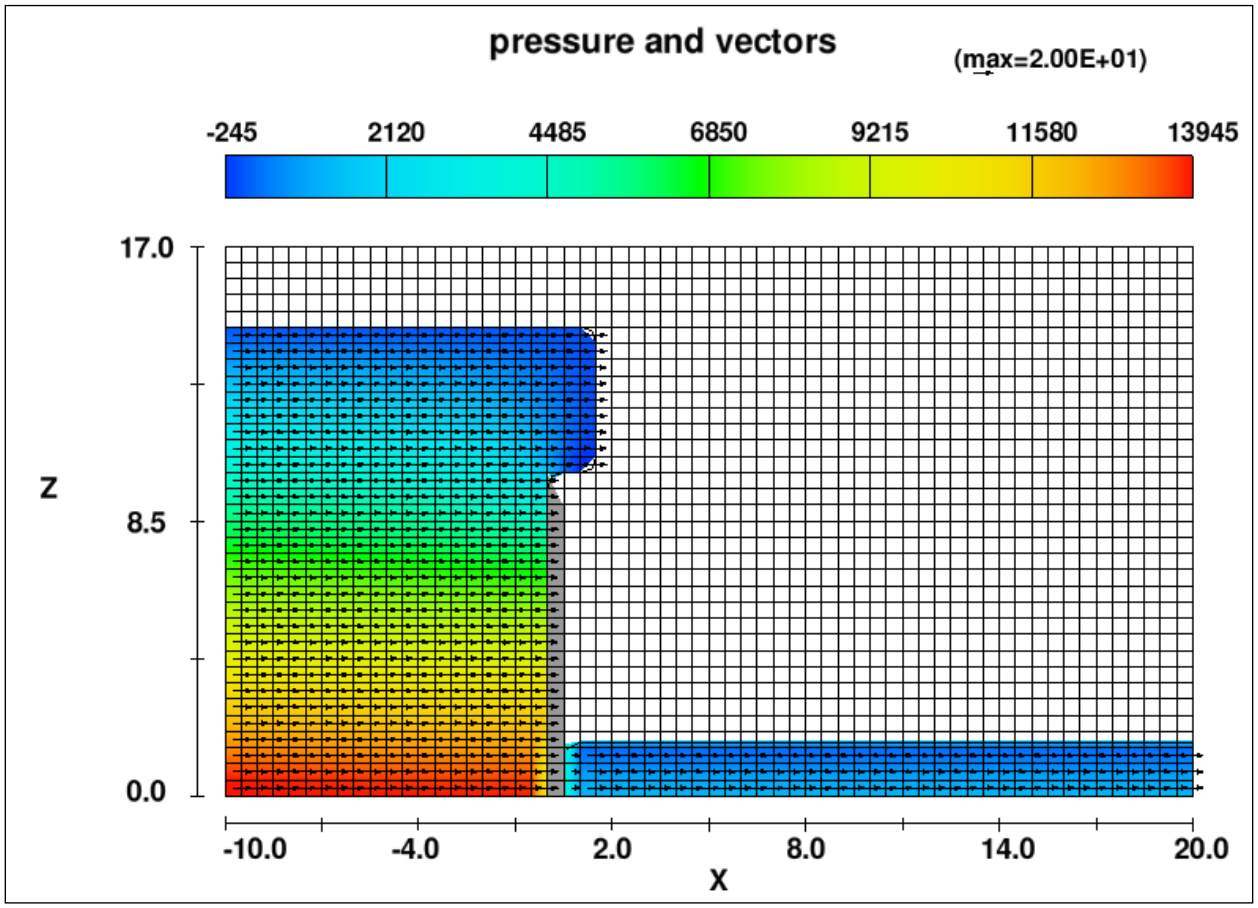
6. Move both of the **Y-direction sliders** to the **left-most position**. The numbers on either side of the slider show you the Cartesian coordinate of the slider. We want both sliders to be at **Y = 0.25**, the cell center of the first cell inside the domain. Note that the cell number **J = 2**. This indicates that the first cell in the domain is actually the second cell in the mesh block created by

the solver: the first cell at J = 1 is a *ghost cell* (outside of the domain) used to compute boundary conditions.

7. Select the **Mesh** checkbox to overlay the mesh on the results.



8. Click **Render** in the lower left to generate the graphics. You will see the image shown below.



Zoom in and out with the mouse roller and pan with the right mouse button. The sharp crest of the weir indicates the mesh resolution is probably adequate for capturing the important geometric features, and

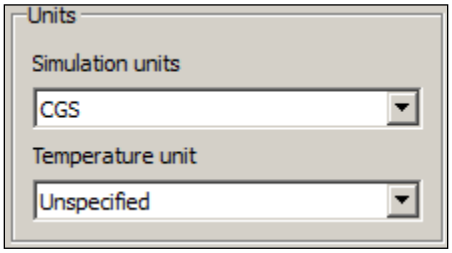
that the rounded edges in the ***FAVORize*** view were due to poor iso-surface fitting. The initial fluid pressure (the **color variable**) and velocity (the **vectors**) are shown. They are important for checking the setup. If other flow quantities such as *density* or *scalar concentration* had also been initialized, they could be checked by selecting them in the **Contour Variable** dropdown list back on the **Analyze > 2-D** tab.

Requesting Additional Output Detail from the Solver

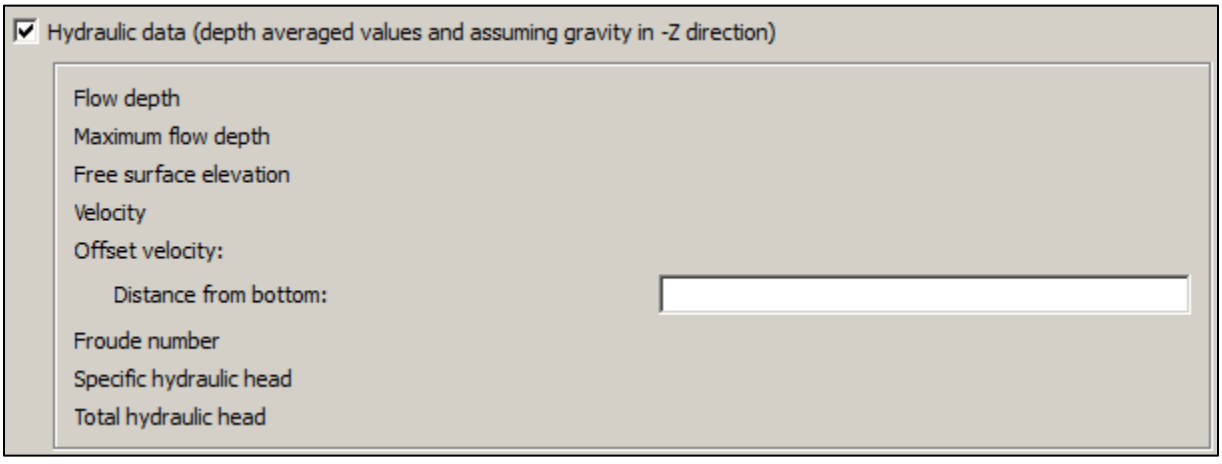
Before running the simulation, there are a few details that should be checked and added:

- Check the **Simulation Units** are specified to allow labels on output plots
- Request **Hydraulic Data** (includes Froude number, fluid depth, free-surface elevation, near-bottom and depth-averaged velocity)
- Specify **Selected Data** output for more frequent plots to see details and make movies

1. Check the **Simulation Units**: Select the **Model Setup > General** tab. On the right-hand side you will see a group box named **Units**. **CGS** (centimeters, grams, seconds) is selected for the **Simulation Units**. There is no **Temperature Unit**, and none are needed since there are no **Heat Transfer** physics models activated on the **Physics** tab.

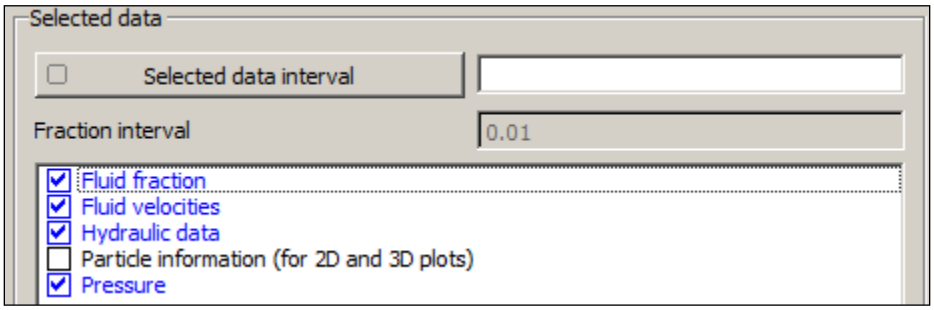


2. Request **Hydraulic Data**: Select the **Model Setup > Output** tab. At the right, under **Additional Output**, select the checkbox for **Hydraulic Data** as shown below. This will cause the fluid elevation, fluid depth, and Froude number to be computed and stored in the results file. These data are not computed unless this option is selected because they are secondary parameters derived from other values.



3. Specify **Selected Data** output: **Selected** data are user-specified output parameters that are written to the results more frequently than **Restart** data. By default **Selected data** and **Restart data** are output every 1/100th and 1/10th of the simulation **finish time**, respectively. **Selected data** is useful for creating smooth animations and seeing detailed flow patterns develop and change.

   On the **Model Setup > Output** tab, select the following **Selected data**: **Fluid Fraction**, **Fluid Velocities**, **Hydraulic Data** (which is available now because you selected it as **Additional Output**), and **Pressure**.



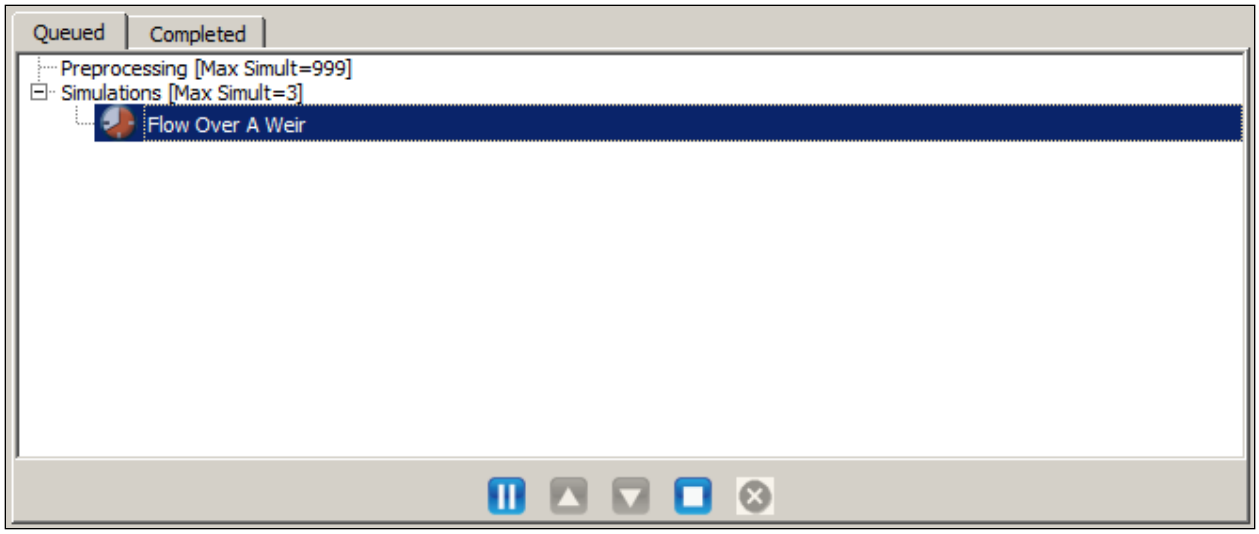
Rule of Thumb

Only specified selected data variables will be written to the results file. If you decide later that you need a variable which was not specified in the **Selected Data** list, the simulation will need to be re-run to get the output you need! On the other hand, too many selected outputs will increase the size of your results file. *It's usually better to select too many than not enough*.

Running the Simulation

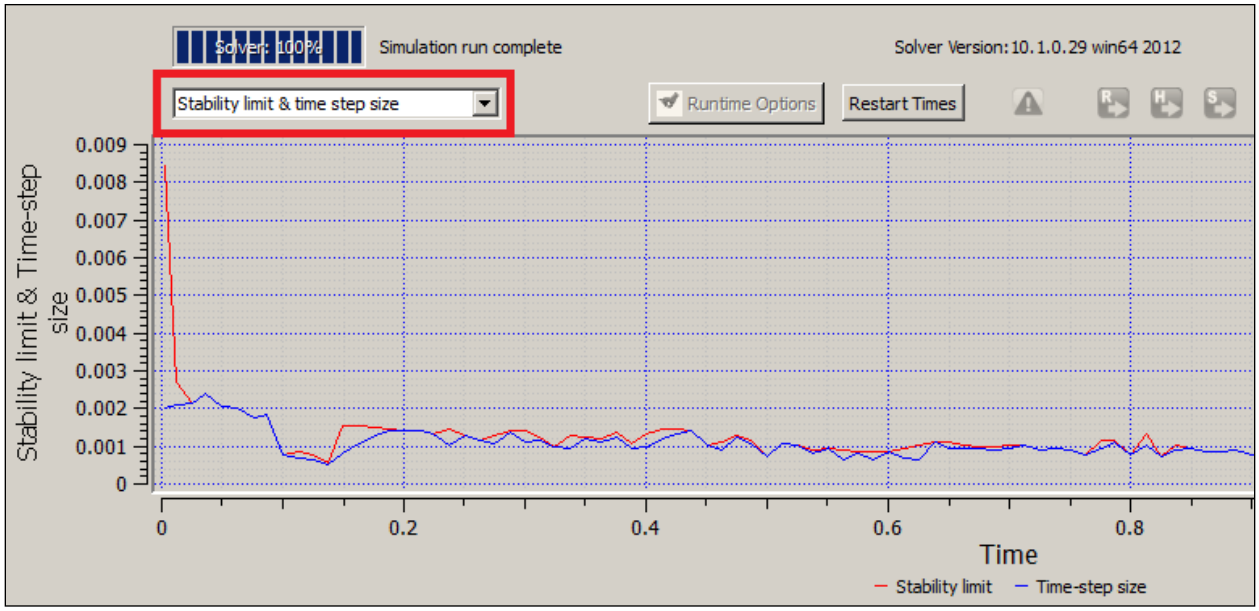
1. **Save** the simulation from the **File** menu at the top. Note before you save that there is an asterisk following the simulation title at the very top title bar of the screen. The asterisk means there are unsaved changes you've made to the simulation.

2. Start the simulation by selecting **Simulate > Run Simulation > Local** from the menu bar at top. The view in the interface will automatically switch to the **Simulation Manager** tab and the preprocessor will start, followed by the runtime solver. Simulations that are running or waiting to run can be paused  , terminated  or rearranged in the queue in the lower left window:



Checking and Diagnosing the Running Simulation

The **Simulation Manager** tab can be thought of as a dashboard for the run. The *efficiency* and *accuracy* of the solution are indicated by the **runtime plots** and **solver text** messages. Familiarize yourself with the various runtime diagnostic plots available in the dropdown list above the plot window:



The plots are generated at every **History Data** output step, which is, by default, 1/100th of the simulation finish time. All the data shown here is also available as **General History** data in the output ***flsgrf*** file. You can draw a box to zoom in on the plots by holding the *left mouse button* and dragging a zoom window. Use *Ctrl+left-click* to zoom back out.

1. **Stability limit & time step size**: Compares the time step stability limit (the smallest time step required in any cell to meet any explicit solver that is active) and *dt*, the actual time step being

used. Ideally the time step *dt* is the same as the stability limit but it may be smaller when there are too many pressure iterations or a stability limit is unexpectedly exceeded (e.g., due to splashing).

2. **Time-step size**: A simple plot of the dynamically-selected time step over time.

3. **Epsi & maximum pressure residual**: *Epsi* represents the pressure iteration convergence criteria that the solver uses to determine if the pressure solution is accurate. At every time step, pressures and velocities are solved in all cells with fluid until the *residual* $\nabla\bullet\bar{U}$ (a measure of potential error) in each cell is less than some small cutoff value *epsi*. The max residual is for the cell with the maximum value of $\nabla\bullet\bar{U}$ after the pressure solution has either converged or reached the maximum allowed number of iterations. The pressure iteration fails if the *max residual* is larger than *epsi* after the maximum number of pressure solver iterations, otherwise it is *converged*.

4. **Pressure convergence ratio**: the ratio of maximum residual to epsi. Less than one means the pressure solution converged, greater than one means it did not. An "ideal" solution maintains *res/epsi* less than one at all times. A few iteration failures are unlikely to damage the simulation accuracy, especially when they are isolated or occur at the beginning of the run. When the maximum residual is much lower than *epsi*, it means that the solution is converging very quickly (the solution is "easy"). When it is close to *epsi*, it implies that the convergence criteria selected is an efficient choice. Non-default options for controlling the pressure solution are rarely needed or desirable, but are available on the **Model Setup > Numerics** tab and will be discussed in a later lecture.

5. **Pressure iteration count**: The number of iterations of the pressure solver. Different pressure solvers have different "best" values. In general, less than ten is a desirable value.

6. **Convective volume error (% lost)** and **Multi-block volume error (% lost)**: Represents the amount of fluid gained (negative value) or lost (positive value) due to advection errors or inter-block interpolation errors, in percent. Usually much less than 1%, and should always be less than 3%.

7. **Fill Fraction, Volume of fluid 1**, and **Fluid 1 surface area**: self-explanatory. In this simulation, the units of volume are cm^3 and surface area is in cm^2. Constant fluid volume and surface area are two indicators that the simulation has reached steady state. Fill fraction is the dimensionless ratio of fluid volume to total open (non-solid) volume in all blocks of the domain.

8. **Mass-averaged mean kinetic energy** and **Mass-averaged mean turbulent energy**: the domain-averaged, time-mean, kinetic and turbulent kinetic energies. These are two more indicators of the steadiness of the flow. Average TKE is not shown in this case because the flow is being modeled as laminar according to the selections on the **Model Setup > Physics** tab.

9. Other plots are available depending on the physics models used: these include thermal energy, number of particles, and other output.

Take a few moments to look over the available **runtime plots** for the simulation. See if they make sense to you.

Also look at the **solver text** output results at the bottom of the screen. This is a copy of the *hd3msg* file that the solver writes to the simulation directory while it runs. **Even if the user interface (GUI) crashes, the solver will continue to run and write all the important output files**. The text output lists a summary of the simulation at each **Short Print Data Interval**. Each line gives:

1. output time **t** (usually in seconds, refers to time elapsed in the model)
2. output **cycle** (the number of time steps solved to date, including the current time step),
3. actual time step **delt** used during the cycle (less than or equal to **dtstbl**),
4. limiting time step required for stability by any explicit solver process **dtstbl**,
5. solver process that requires the limiting time step, as a two-letter **code** (*cx* = *x-direction convective fluid flux*, for example)
6. number of pressure iterations during the cycle **iter**,
7. the ratio **res/epsi** of pressure solution maximum residual (any cell) to convergence criteria epsi
8. **volume** of fluid in the simulation domain,
9. the known **%loss** due to convective flux error and inter-block interpolation error
10. the fraction of the open domain volume that is filled with fluid **frac**,
11. elapsed real-world time and computer clock time **el_time** and **clk_time** (hh:mm:ss),
12. multi-processor efficiency ratio **%PE**, as percent

Loading Results While the Solver is Running

The results of a simulation can be viewed while the simulation is running or after the simulation is complete. It is often useful to visualize the results early to ensure that it is running correctly.

1. Select the **Analyze** tab. A message appears indicating that the ***prpgrf*** file no longer exists. The ***prpgrf*** file was generated during preprocessing and was deleted to save space when the simulation ran. The ***flsgrf*** file is the simulation output file and contains the same information as the ***prpgrf*** file, plus the output of the run. Select **Continue** and a file selection dialog will be presented.
2. Select **Custom** output files and the ***flsgrf*** file, and then select **OK**.

Understanding Plot Types and Data Sources

The **Analyze** tab will now be displayed. There a number of ways in which to visualize the results of a simulation. The available plot types are:

Custom: a rarely-used way to get "under the hood" of the post-processor. Text commands used for this tab are given in the *User Manual* chapter on *Post-Processing*.

Probe: Displays values for *individual cells* as well as *non-cell* data like integrated global values, boundary condition values, and component-specific integrated output. Plots charts or text of values vs. time.

1-D: Flow parameters and other data can be viewed along a line of cells in the X, Y, or Z direction. Extents (limits) can be applied both in space and in time.

2-D: Results can be viewed in an X-Y, Y-Z, or X-Z plane. Plot limits can be applied both spatially and in time. Solids are always displayed, and fluid is colored by selected flow parameters. Velocity vectors and particles (if present) can be displayed. Solid surfaces and fluid free surfaces are drawn using all available data, and are more accurate than in 3-D plots.

3-D: Both fluid and solid surfaces can be visualized together or separately. The fluid surface can be colored by flow quantities, which show the state at that surface. Additional information such as velocity vectors, particles (if present), and streamlines can be added. Plot limits can be applied both spatially and in time. The surfaces are drawn using a 3-D iso-surface method that does not locate them as accurately as in 2-D plots.

Text Output: Restart, Selected, and Solidification data can be written to text files for every cell within selectable extents in space and time. Use this tab to get many cell-by-cell numerical values. If just a few locations are of interest, it is easier to place *History Probes* before the run and get their output from the **Probe** tab.

Neutral File: Restart and Selected Data can be interpolated to give results at a set of user-defined points. This output is useful for plotting time-varied results along a flow path that curves within the mesh.

FSI/TSE: Only used for visualizing deforming solids output from the finite-element stress-analysis physics package.

Once a plot type (tab) is selected, the next step is to choose the data source. There are five sources of data in ***FLOW-3D***:

Restart: All cell-by-cell flow variables necessary for a complete solution, plus any **Additional Output**. Output frequency = $1/10^{th}$ of the finish time by default.

Selected: User-selected cell-by-cell flow variables. Output frequency = $1/100^{th}$ of the finish time by default.

General History: Non-cell-specific data. Only available on the **Probe** sub-tab. Includes global output like time step size, epsi, and mean kinetic energy. Also includes output for moving objects, history probes, flux planes (measurement windows), control volumes, history particles, and more.

Mesh Dependent History: Only available on the **Probe** sub-tab. Includes boundary condition information like flow rates.

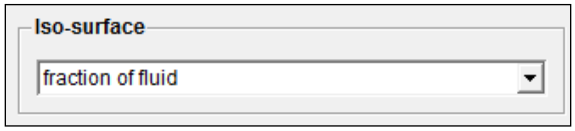
Solidification: Only available if the solidification model is active.

Examples of some of the available plot types will be generated in the next step.

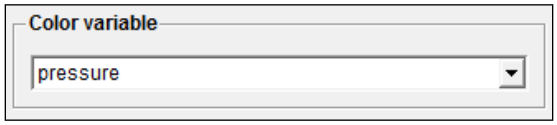
Plotting 3-D Graphics

Select the **Analyze > 3-D** tab. There are 4 choices which always need to be made on this tab.

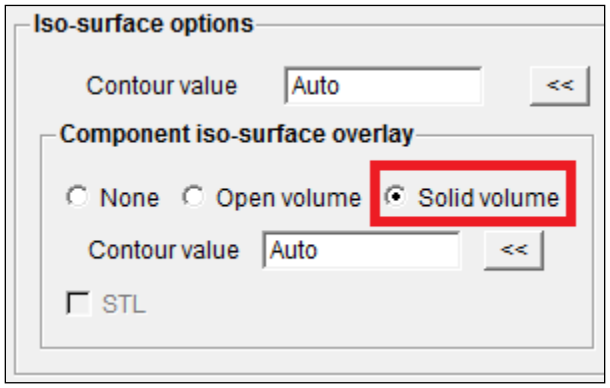
1. **Iso-surface**: The surface (interface) to draw in three dimensions. Select **Fraction of Fluid** to plot the fluid free surface.



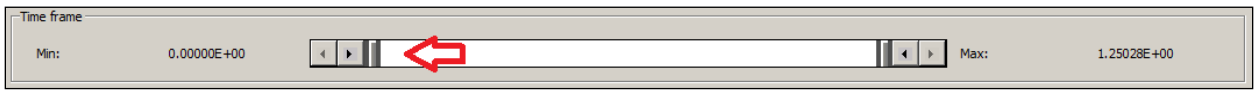
2. **Color variable**: The selection here will determine what variable to color the fluid surface with. Select **Pressure**.



3. **Iso-surface options > Component Iso-Surface Overlay**: options for plotting the interface (surface) of solids. Select **Solid Volume** to plot the solid components (weir).

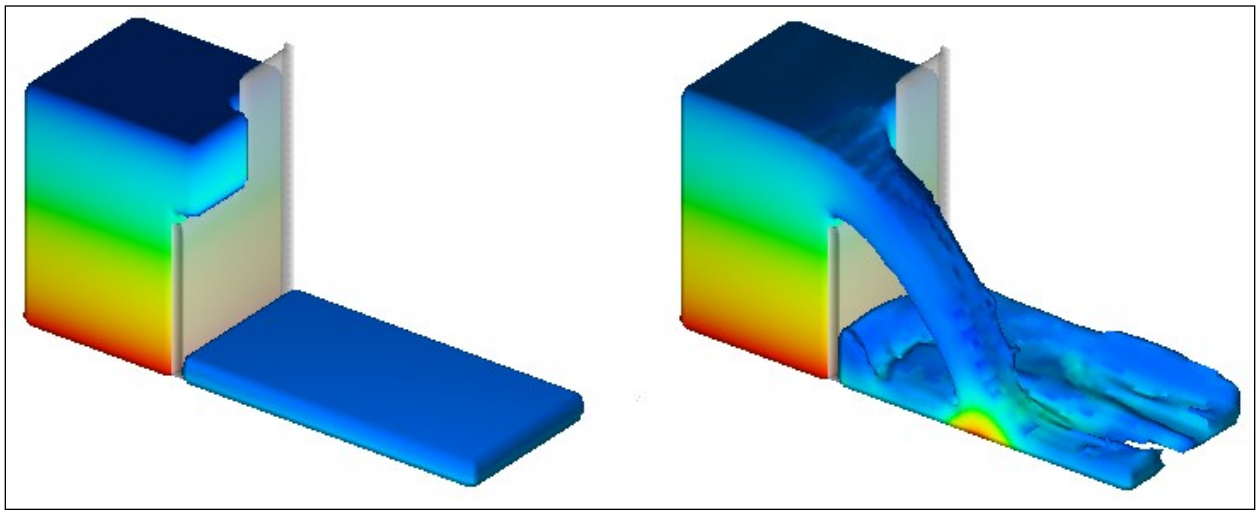


4. **Time**: Click and drag the **Left (Min) Time Frame Slider** so that **Min = 0** and **Max = 1.25** seconds.



5. Click the **Render** button to generate a series of 11 plots between t = 0.0 and 1.25 seconds which contain the weir and fluid surfaces, with the fluid surface colored by pressure. If the geometry

isn't visible, use  to reset the view. Step through the images by selecting time steps from the list at the right. The first and last output times should be similar to these:



6. Return to the **Analyze > 3-D** tab. Choose **Selected** data as the **Data Source** to access the more frequent data specified earlier in the exercise.

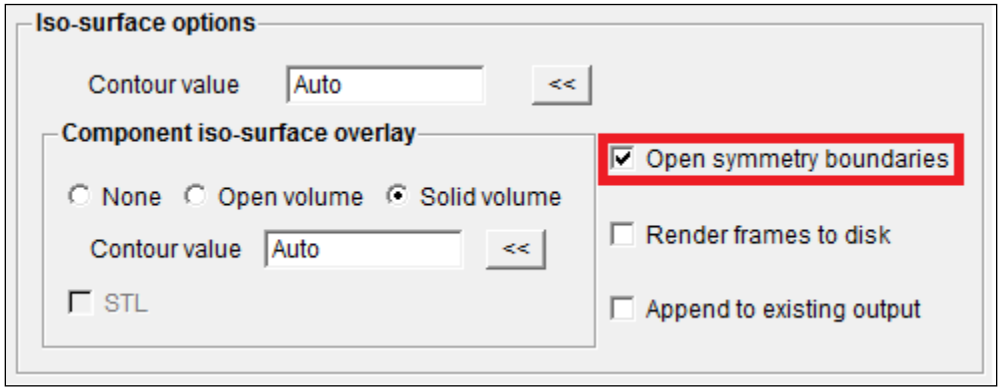
7. Set the **Min Time Slider** all the way to the left (**t=0.0**)

8. Now click the **Render** button. The view will switch to the **Display** tab and 101 plots will be listed in the **Available Time Frames** list. Click **Next** to advance through the frames.

Reflecting Results Across a Symmetry Boundary

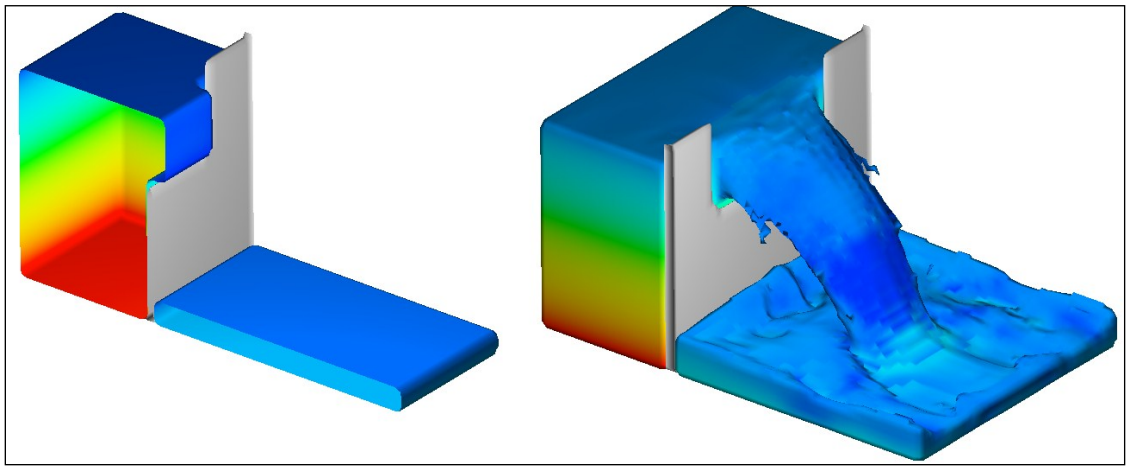
The simulation was set up with a symmetry plane as a boundary condition along the centerline of the weir. Only half of the weir structure is being modeled and only that half is displayed. For presentations it would be helpful to show both halves of the weir and explain the *assumption* of symmetric flow.

1. Return to the **Analyze > 3-D** tab and select the **Open Symmetry Boundaries** checkbox.



2. Click **Render**. The fluid surface should now appear open (transparent with sharp edges) at the y-min symmetry boundary, which is at y = 0 cm.

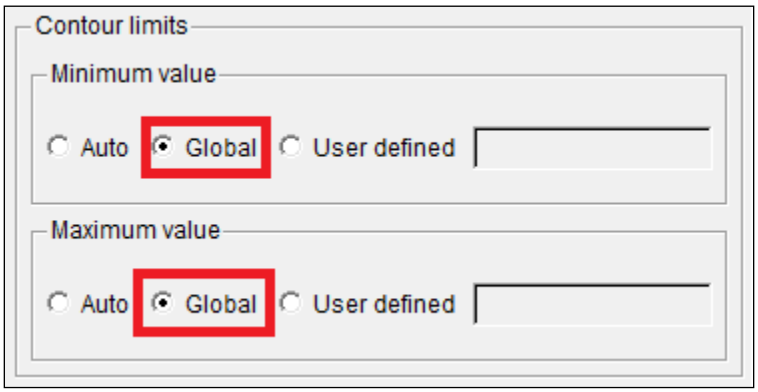
3. To mirror the results across the y=0 plane, select **Tools > Symmetry** from the menu above the display and select the **Y Direction** checkbox in the dialog. Click **Apply** and **Close** to exit the dialog. The display show a full weir structure as shown below.



Creating a 3-D Animation

The next step will be to create an animation of the 3-D fluid surface. Animations are movies created from the frames in the **Available Time Frames** list. To improve the visual effect of animations, it is recommended that a common color scale be applied to all frames.

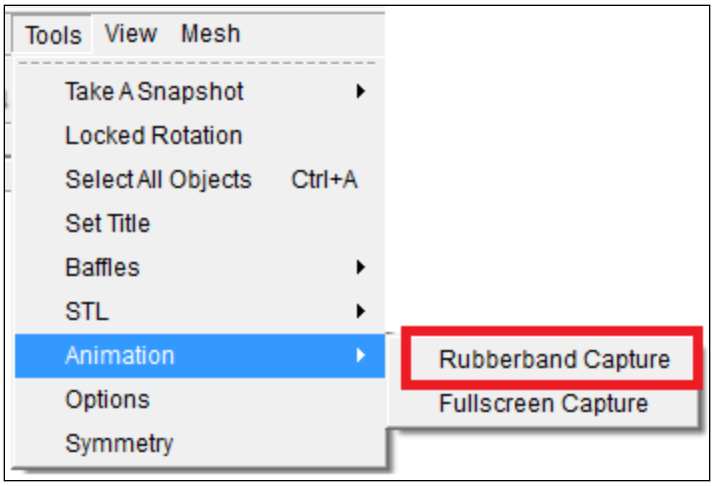
1. Return to the **Analyze > 3-D** tab and select the **Global** radio buttons under **Contour Limits**.



2. Make sure you have selected all time steps with the **Time Frame Sliders** and click **Render**.

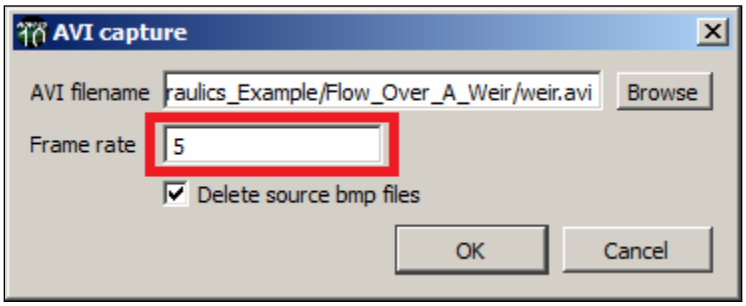
3. Again, select **Tools > Symmetry > Y-Direction** to mirror the results across the Y=0 plane. Select **Apply** and **Close**. **Pan** (using the right mouse button) until the weir is just below and to the right of the color bar.

4. To create the animation, select **Tools > Animation > Rubberband Capture**. Instructions will appear to explain the next step; select **OK** to continue.

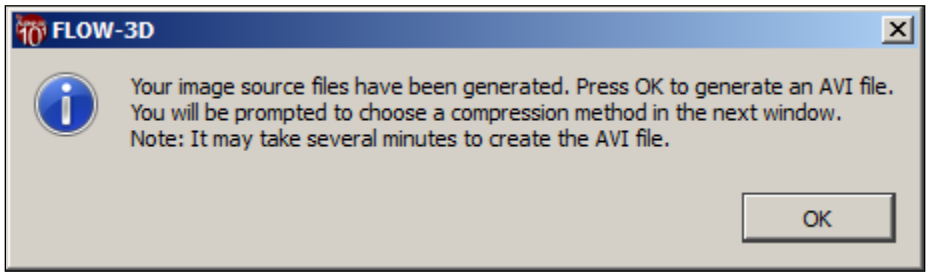


5. **Click and hold** the left mouse button somewhere to the upper-left of the weir and color bar, and drag the selection box that appears so it encloses the weir and color bar. Repeat the process if necessary. Click the **Capture** button to accept the selection.

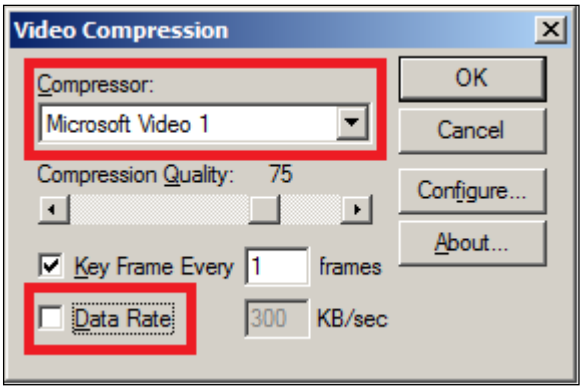
6. A dialog will appear allowing the animation to be named. The default name for animations is ***out.avi***. A more descriptive name is always recommended. The default frame rate is 10 frames per second. This may be too fast for some simulations, so enter **frame rate = 5**. Keep the option to delete the individual frame images after they've been combined into a movie.



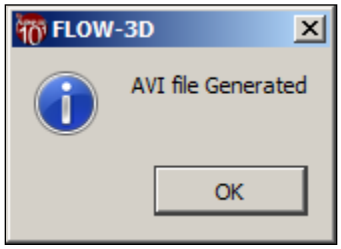
7. Each time frame will be rendered to the Display window and bitmaps will be written (and then deleted) in the simulation directory. The camera view may wobble a little during this process: this is a known issue and will be addressed in a later exercise on using ***FlowSight*** post-processing. Once the rendering is complete, the following dialog will appear. Click **OK** to begin the next step of the process. The **Video Compression** dialog will open. Note that the message below stays open: do not click OK on it again.



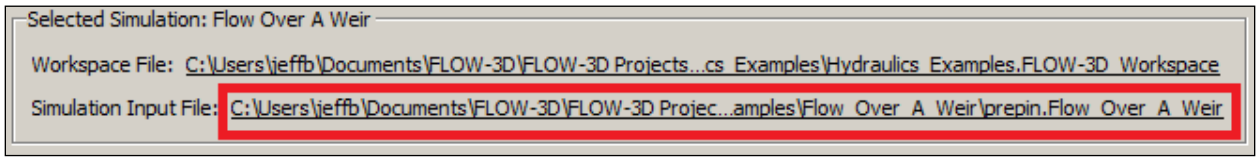
8. The default compression for animations is **Full Frames (Uncompressed)** and is not recommended since the file size can be quite large and difficult to load in video players and slide show presentations. Select **Microsoft Video 1** since the codec is universally available on Windows platforms. Unselect the **Data Rate** checkbox so that the quality of animations is not limited by the data rate.



9. Click **OK** to begin the compression process. When the compression is complete, the following dialog will appear. Click **OK** to continue.



10. To play the animation, use the operating system to browse to the directory where the simulation is located. You can do this quickly by going to the **Simulation Manager** tab and clicking on the path listed for **Simulation Input File**. Find the ***.avi*** file you created and play it by double-clicking on it.



Plotting Options in 2-D

1. Go to the **Analyze > 2-D** tab.

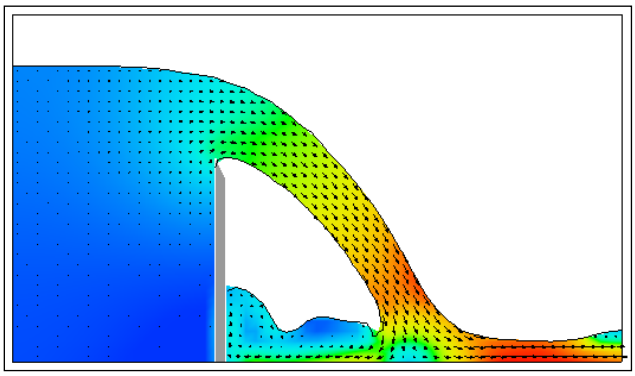
2. Choose **Selected** data. Note that the **Particle Type** option grays out because there is no particle data in the **Selected** data set (it wasn't activated on the **Output** tab).

3. The most useful plane to view results for this simulation is the centerline of the weir. Choose the **X-Z** radio button in the **Plane** group and set both of the **Y Limits sliders = 0.25** cm, which is

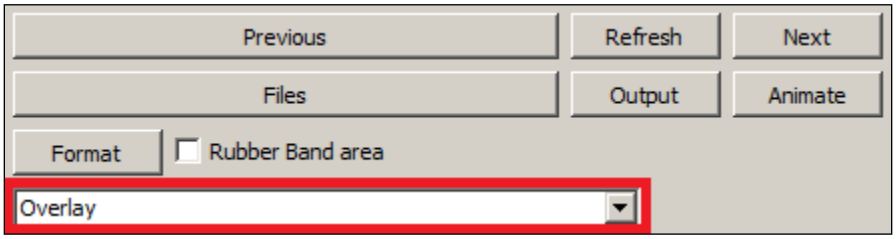
the location of the cell centers closest to the centerline at Y = 0. Set the **Time Frame sliders** to include all output times.

4. Select **Contour Variable = Velocity Magnitude**. Geometry is always displayed on 2-D plots, so nothing needs to be selected to view it.

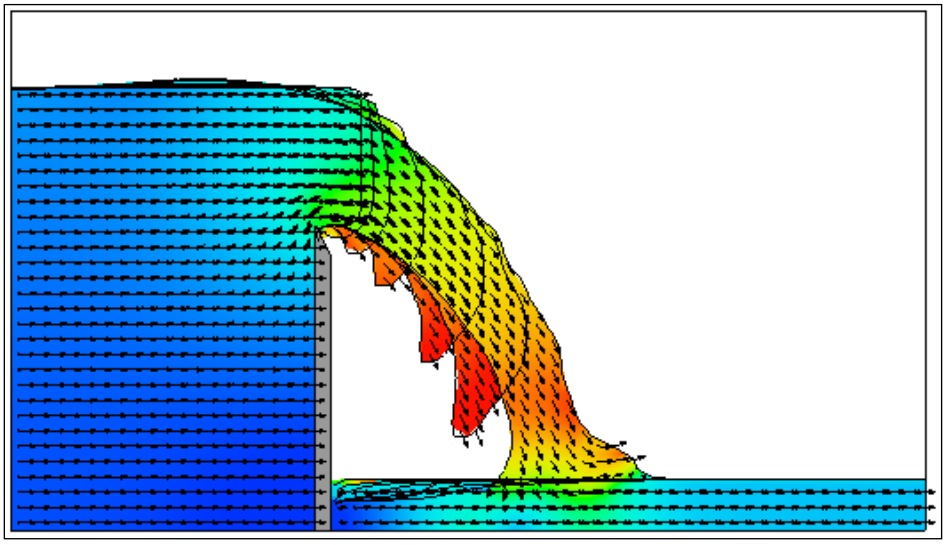
5. Click **Render** to generate a time sequence of 2-D plots of pressure in the plane. Graphics similar to following will appear on the **Display** tab.



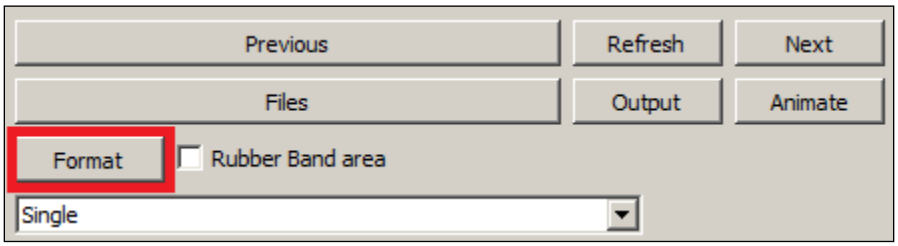
6. Make an overlay of the free surface with time: select **Overlay** from the dropdown plot type list in the upper right:



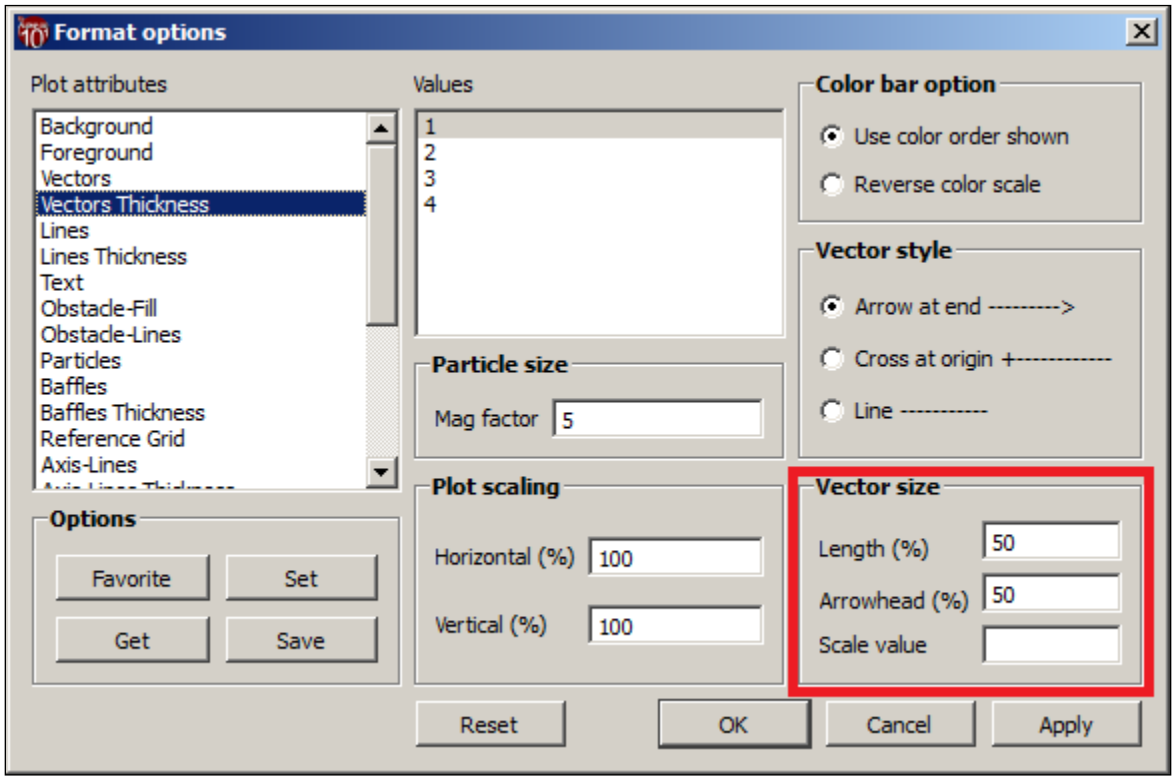
Click on several of the early time steps to make an overlay plot:



7. Return to **Single** plots and click on the **Format** button in the upper right-hand corner.



Experiment with the options there, especially the **Vector Size** options. Click Apply to change the display without closing the dialog. Take some time and experiment. When you are done, select **Reset**, then change the **Background** to **White**, and then click **OK** to close the dialog.



Plotting Time History Results in Probe

1. Select the **Analyze > Probe** tab. Time history plots of variables are created here. There are four types of time-dependent data that you're likely to use. **Solidification** data is not discussed here.

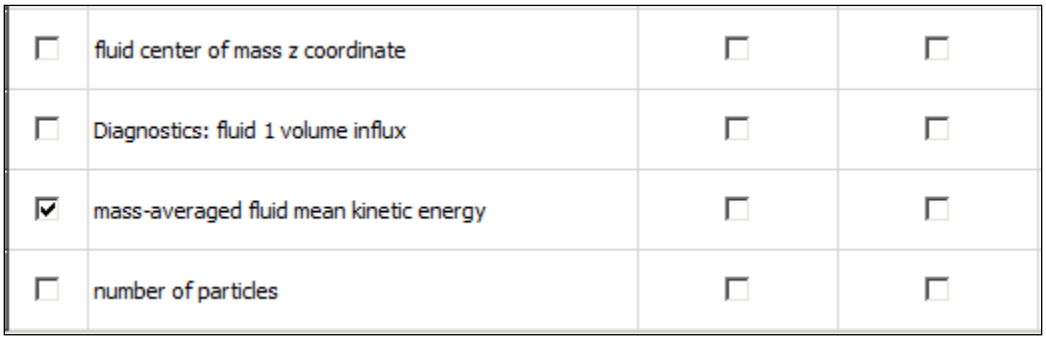
   A. **Restart data** and **Selected data**: cell-specific values of flow variables, plotted for a single spatial point over time when using **Probe** history.

   B. **General history data**: non-cell-specific quantities which vary with time. Typical examples are global quantities and component-specific quantities. Many **General History** variables are integrations of cell-by-cell values. Examples include integrations over the whole domain, over a solid surface, or over the plane of a baffle.

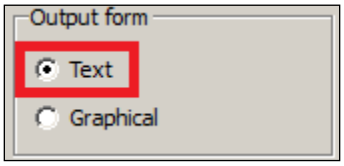
C. **Mesh dependent data**: time-dependent parameters computed at mesh boundaries. Typical quantities are flow rate and fluid height. Note that some parameters only report the specified value (like fluid height), while others report the computed value (like flow rate). You can usually tell the difference because computed values *fluctuate*, while specified values are *constant*.

2. Select the **General History** radio button under **Data Source**. Note that the **Data Point (location) sliders** gray out because the values for this source are global.

3. Select **Mass-Averaged Fluid Mean Kinetic Energy** from the list and **Render**. You can use the mouse to zoom and pan.



4. Return to the **Analyze > Probe** tab and select **Text** as the output form. Then **Render** again.



5. The text output can be written to a file by selecting the **Save As** button (don't forget to name the file). Click **Continue** to close the text output dialog.

Plotting Text Output

Text Output (on the **Analyze > Text Output** tab) is very similar to the text which can be output from the **Probe** tab except that output for many cells in a *3-D rectangle* can be output from **Text Output**, while only *one cell at a time* can be output from **Probe** spatial data. Only **Restart** and **Selected** data can be output as **Text Output**. The default spatial extents are set to the entire domain, and the default time extents are the last time step only. The spatial and time extents should be selected with care as the text files can become very large.

1. Experiment with outputting a single time step. Plot **Free Surface Elevation** and **Fluid Depth**. You can save the text file with **Save As**. Click **Continue** to close the text output dialog.

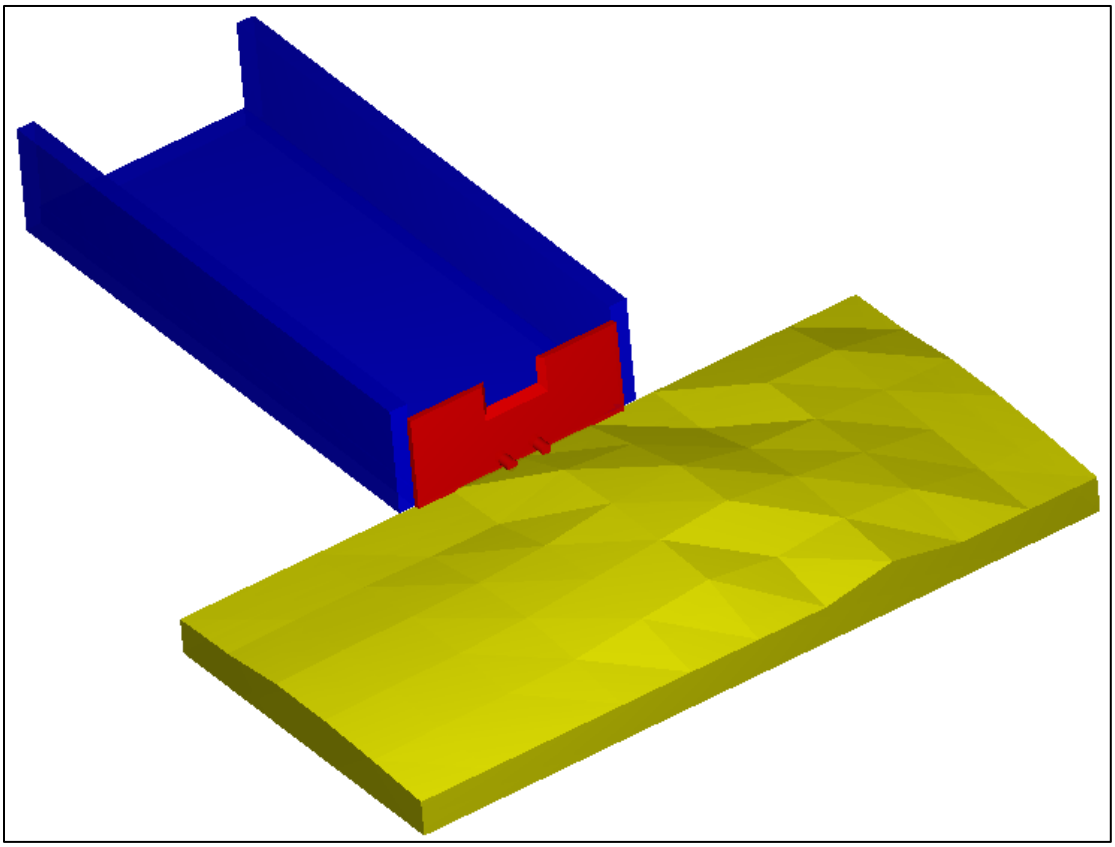
2. Experiment with outputting two or more time steps. Note the section breaks between time steps: these often must be removed outside of ***FLOW-3D*** for additional post-processing.

Exercise 2: Setting Up Geometry

Exercise 2: Setting Up Geometry

Problem Description & Objectives

The next three exercises focus on building a sharp-crested weir simulation. An ellipsoidal obstacle lying on its side will be embedded in the channel downstream of the weir. Topography will be added to the downstream channel bottom. This setup will be used in future exercises to simulate flow over the weir as well as outfall from pipes at the base of the weir.



Learning Objectives

In this exercise you will learn techniques to:

1. Check and fix stereolithography (*.stl*) geometry files using ***netfabb Basic*** and ***pyAdmesh***,
2. Add *.stl* geometry to a simulation,
3. Add simple geometry to a simulation using "primitives",
4. Create holes in geometry (Boolean operations),
5. Transform geometry (rotation, translation, magnification).

Create a New (Blank) Simulation File

1. Launch the ***FLOW-3D*** ® interface by double-clicking the ***FLOW-3D*** icon on your desktop.

2. If you have not done so already, create a workspace for this training titled **Hydraulics Examples** following the instructions in Exercise 1, p.1.

3. Go to the **Simulation Manager** tab.

4. Select (click to highlight) the workspace **Hydraulics Examples** in the **Portfolio**.

5. Create a new simulation named **Weir Geometry**.

    a. Select **File** > **Add New Simulation…** from the dropdown menu at the top of the screen.
    b. Enter **Weir Geometry** as the simulation name.
    c. Leave **Create Subdirectory Using Simulation Name** checked.
    d. Select **OK**.

Check & Fix the *.stl* File

The starting point for developing a new simulation is making the geometry. Simple geometry can be build with primitives in ***FLOW-3D***. Complex geometry must be imported in stereolithography (*.stl*) format. You will use *.stl* files in this exercise for a weir plate and some simple downstream topography.

.stl files are exported from computer-aided design (CAD) programs and contain a closed surface of triangular facets. Each triangle (facet) which makes up the surface is described by three Cartesian vertices and a normal vector pointing outward from the enclosed volume. In the author's experience, *all* CAD packages *always* add errors and artifacts during *.stl* creation. Simple errors can be fixed quickly with *.stl* repair tools. Systemic errors require correcting and re-exporting the file from CAD. Typical simple errors are missing facets (holes), zero-area facets, non-manifold vertices, and reversed or incorrect unit normals. **Geometry files must always be checked for errors and errors must be repaired**. ***netfabb Basic***, ***MeshLab***, and ***qAdmesh*** (which comes with ***FLOW-3D***) are free tools for repairing *.stl* files.

1. Install and/or launch ***netfabb Basic*** (http://www.netfabb.com/basic.php).

2. Remember from Exercise 1 how to open the simulation directory from inside the **Simulation Manager** tab? Do so, and copy the geometry files ***weir1.stl*** and ***Topo_Example_04_fixed.stl*** to the new **Weir Geometry** simulation folder. It's recommended (but not required except in this exercise) that you keep your *.stl* files in your project folder.

3. Check the *.stl* file using ***netfabb Basic***:
    a. Open **Windows Start Menu > All Programs > netfabb Basic > netfabb Basic**.
    b. Select **Part > Add** and navigate to **Libraries > Documents > FLOW-3D > FLOW-3D Projects > Hydraulics_Examples > Weir_Geometry > *Weir1.stl***. If you have trouble finding the file, ask your instructor for help.

c. The  image in ***netfabb Basic*** indicates that there are errors. Select **Extras > New Analysis > Standard Analysis**. A new branch of the tree appears in the *Context Area* at the right. You can switch between views and information by clicking on different branches of the tree.

d. Read the analysis in the *Context Area* and find where it reports that: (a) the surface is **not closed**, and (b) there is **1 hole** with **3 boundary edges** and **0 bad edges**.

e. Check the **volume** and **surface area** of the part. Note that mm just means 'unit of length': most *.stl* tools assume you are making *.stl* files for 3-D printing.

f. Rotate (right mouse button) and pan (center mouse button/wheel) the view until the front face of the weir is visible. Note the red facet on the left side of the crest. This is the missing facet.

4. Repair the *.stl* file using ***netfabb Basic***.

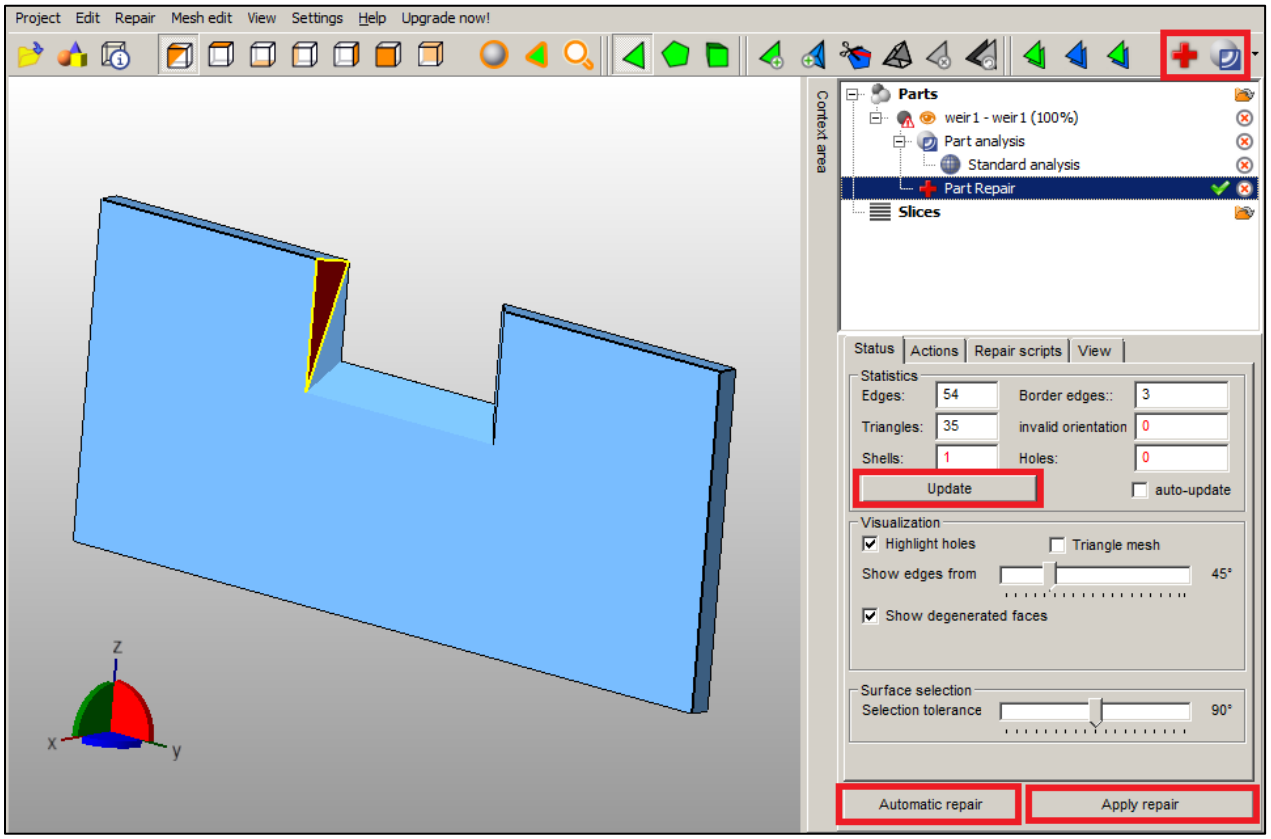
a. Select **Extras > Repair Part**. Select **Automatic Repair** at the bottom of the *Context Area*. Choose **Default Repair** and then **Execute**.

b. Click **Update** and note how the number of holes is changed to zero.

c. Click **Apply Repair** and then choose to **Remove Old Part**.

d. Don't use the **Save** option. Instead, choose **Part > Export Part > as STL**. Name the file **weir1_repaired.stl**, and make sure it's being saved into the *Weir Geometry* project folder. Click **Save**.

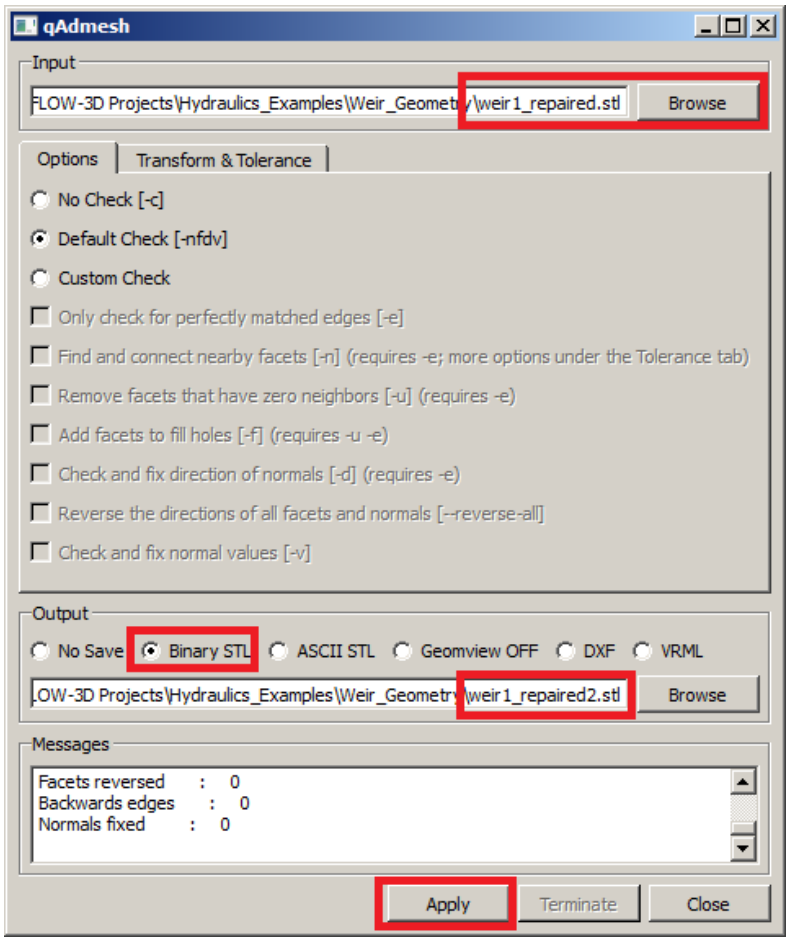
e. Repeat the process for ***Topo_Example_04_fixed.stl***. There should be no errors in that file.



You can load multiple parts at once and toggle between them using the tree to check that they are oriented correctly and in the same coordinate system. There are many other features of ***netfabb Basic*** that you can explore on your own.

A second *.stl* tool, called ***qAdmesh***, is included with ***FLOW-3D***. The default mode of operation for ***qAdmesh*** is to check a file and report any errors found, without fixing the errors. ***qAdmesh*** fixes normals more precisely than ***netfabb Basic***, but generally does not fill holes as well. It is recommended that you use ***qAdmesh*** after ***netfabb Basic*** as a final check.

5. Re-check and repair the *.stl* file using ***qADMesh***:
   a. In ***FLOW-3D***, go to the **Model Setup > Meshing & Geometry** tab.
   b. Select **Tools > qAdmesh**.
   c. Select **Browse** (next to the **Input** box) and then find your local copy of ***weir1_repaired.stl***. The simulation folder should already be open in Windows; look at the address bar at the top of that window to guide your browsing. Select **Open** when you've selected ***weir1_repaired.stl***.
   d. Under **Output**, select the **Binary STL** radio button. The **Output** location box is now activated.
   e. Change the name of the output file (at the end of the path) to ***weir1_repaired2.stl***.
   f. Click **Apply** to **Default check** (and fix) the file.
   g. Check the **Messages** pane and carefully review the details of the repair.



The information in the **Messages** pane indicates the results. The status listed under **Final** indicates no errors, so the file has been fixed.

Under some conditions, *.stl* tools may have difficulty fixing a file. If multiple repairs do not solve the problems then re-export the *.stl* file from CAD. Different *.stl* tools use different repair algorithms and may produce different results. It's recommended that you repeat the process of checking and fixing files until both ***netfabb Basic*** and ***qAdmesh*** report no errors.

If an *.stl* file is larger than about 420 MB, neither software will be able to open it. In that case use a 64-bit software like ***MeshLab*** or a professional version of ***netfabb***. The use of ***MeshLab*** is discussed in a separate exercise on topography.

Import the Fixed Weir *.stl* File

1. Go to the **Model Setup > Meshing & Geometry** tab.

2. Select **Subcomponent > Geometry File(s)** from the dropdown menu above the display pane.
   a. **Add to Component** = **New Component (1)** since there are currently no components.
   b. Click the **Add** button.
   c. Select ***weir1_repaired2.stl*** and click **Open**.
   d. Set **Subcomponent Name = weir plate** and then click **OK**.
   e. Give the new **Component 1 Name = weir** and select **OK** to finish adding it as a **Solid** component.

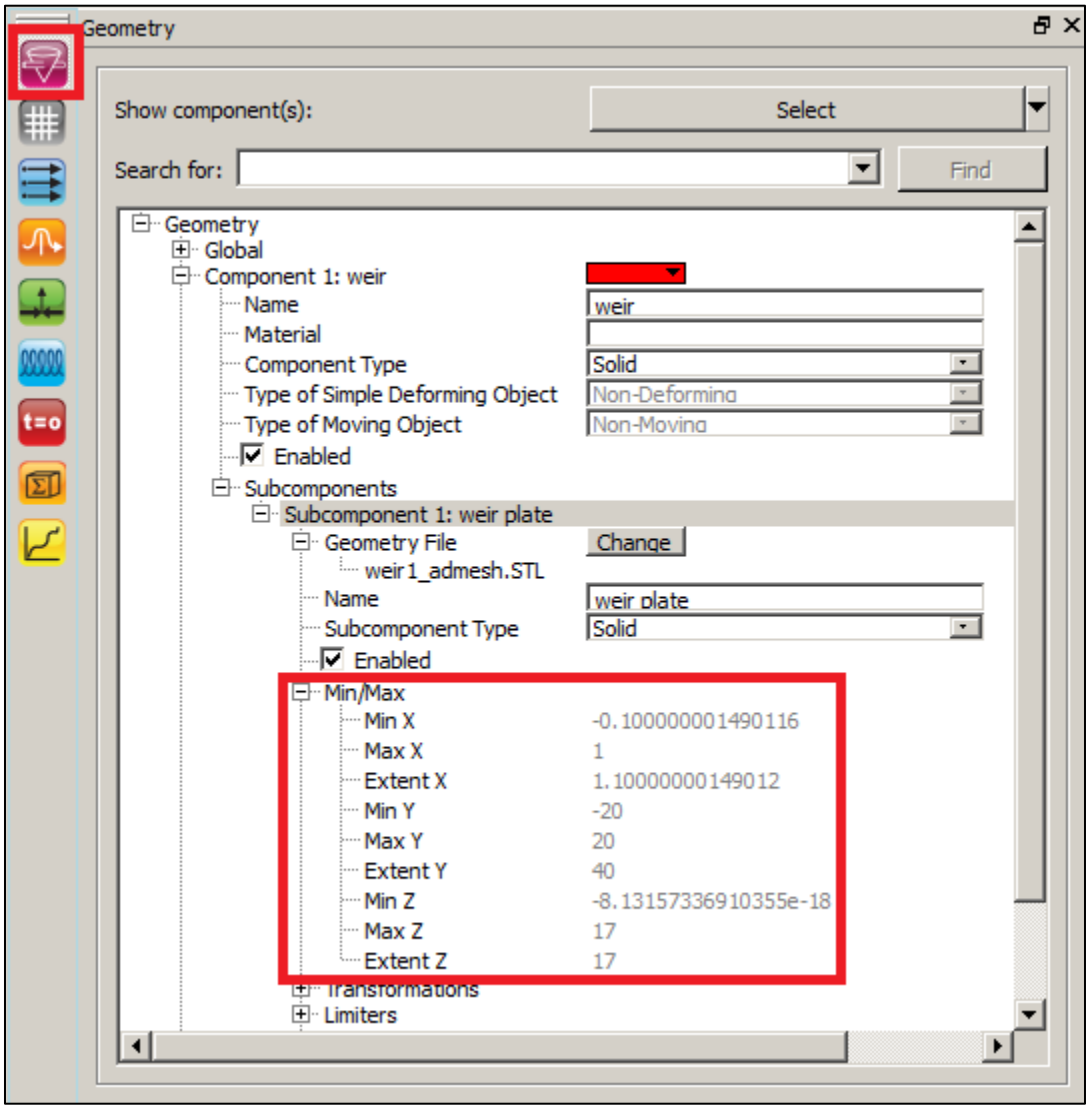
3. Rotate (left mouse button), zoom (mouse wheel), and pan (right mouse button) the camera to view the weir.

4. **Save** the simulation.

Determine Weir Subcomponent Dimensions and Unit System

Sharp-crested rectangular weirs are a relatively well understood method of measuring flow in channels. They measure flow accurately only when certain approach and downstream conditions are met and when their construction follows established guidelines. You need to know the dimensions of the plate itself, the width and elevation of the crest notch, and the angle of the blade edge to guarantee proper flow measurement. The US Bureau of Reclamation (USBR) *Water Measurement Manual (3rd edition revised 2001)* and *Water Measurement Structures (3rd edition revised 1989)* by M.G. Bos are guides. Now you will use ***FLOW-3D*** to find the dimensions of the weir, although you could do so more easily back in ***netfabb Basic***.

1. Toggle the Geometry tree using the  button.
2. Expand tree branches to open **Geometry > Component 1: weir > Subcomponents > Subcomponent 1: weir plate > Min/Max**. Examine the **Extents**, **Min**, and **Max** coordinates of the weir subcomponent**.** The weir is **40** units wide (centerline at y=0), **17** units tall (base at **z = 0**), and about **1.1** units wide (upstream face at **x = -0.1**).

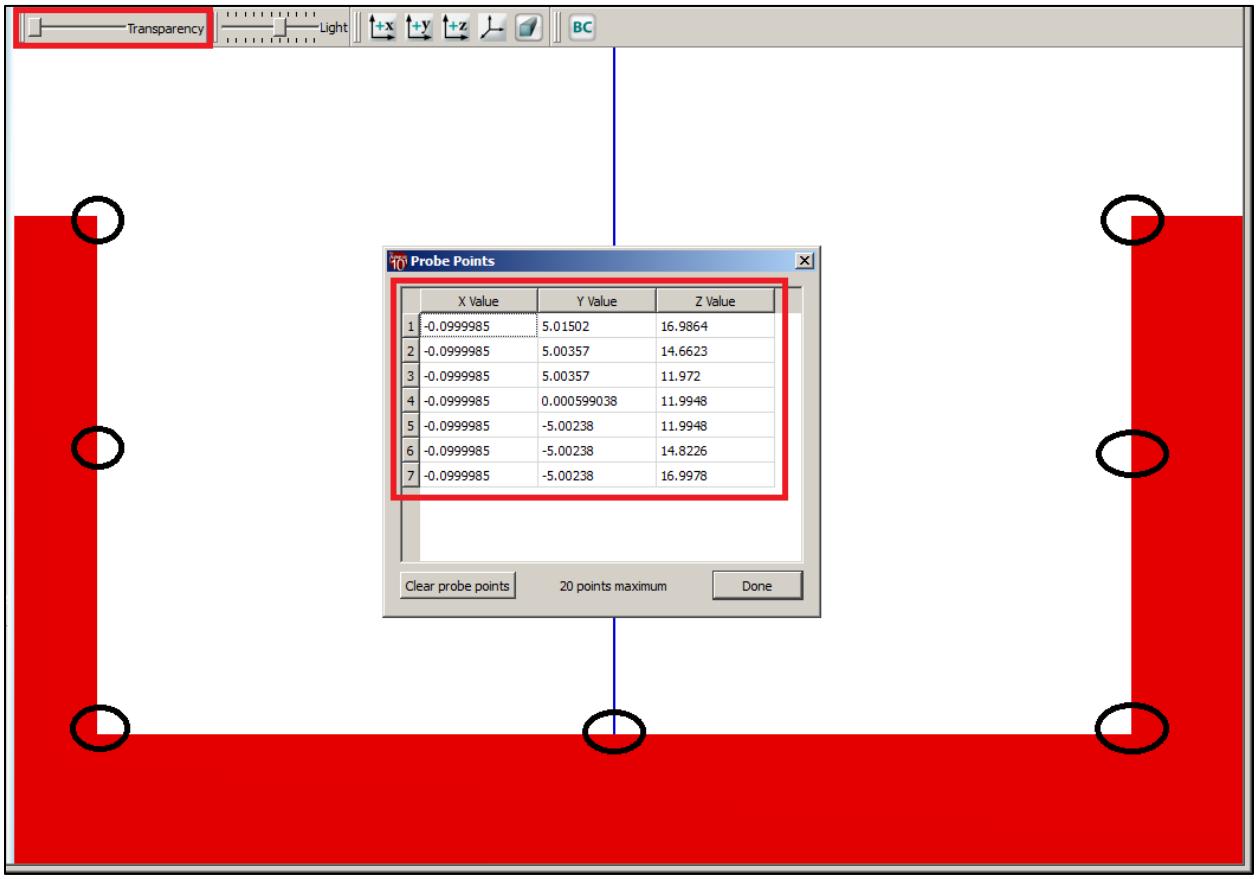


3. Toggle the  button until the y-axis vector points to the left.

4. Select **View > Draw Axis > Origin** from the menu above the display to set the axis at the origin.

5. Set the **Transparency** slider all the way to the left so the weir is opaque.

6. **Zoom** and **pan** the view so the crest fills the screen as shown below. Do not rotate the view.



7. Select **View > Probe Points List …** from the display pane menu.

8. *Left-click* the mouse button and then press and hold *Shift*. Repeat if necessary until the mouse arrow becomes a cross-hairs when you hold *Shift*.

9. Carefully *Shift-click* on the edges of the notch at the points shown.

    a. *No point is recorded* when you click on *blank space*.
    b. Take multiple readings to estimate the uncertainty of your mouse location.
    c. Zoom in and pan if necessary to get more precise readings (rotating is not recommended).
    d. The most recent point probe value is also shown in the bottom left margin of the ***FLOW-3D*** window. When the probed point is in empty space, the X-value displayed there is an estimate (around -42) of the *domain extent*.

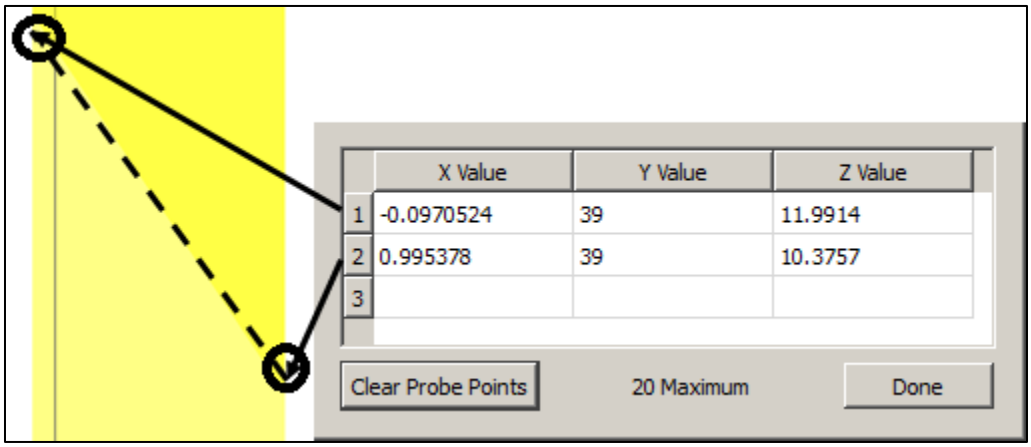
10. Use your measurements to find the edge coordinates of the crest notch. The notch is open from **y = -5** to **5** (10 units wide), and begins at **z = 12** and continues to the top of the plate at z = 17 (5 units tall).

11. The list remains populated until you select **Clear Probe Points**. Do this now.

12. Click on the  icon to switch to a profile view from the side. Pan and zoom to view the weir.

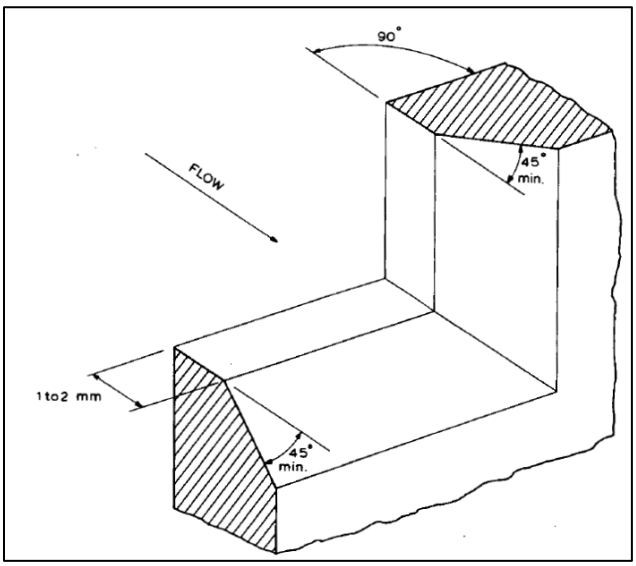
13. Use the **Transparency** slider to increase transparency until you can see the blade edge. You may see two, in which case the one of interest is the lower one.

14. Repeat the measuring process to find the coordinates of the blade endpoints.



15. Find the blade angle using trigonometry:
    a. $\tan \theta = \Delta x/\Delta z = (0.995378 + 0.0970524)/(11.9914 - 10.3757) = 0.676$.
    b. $\text{Tan}^{-1}(0.676) = 34°$, the internal angle of the blade.
    c. Subtract the internal angle from 90° to find the external angle = ~ 66°.
    d. The external angle must be at least 45° as shown in the schematic below.

16. Zoom in very close to the upper edge of the blade and identify the flat crest bevel. Bos (1989) and others recommend bevel thickness between 1 and 2 mm (0.03 to 0.08 feet). A series of closely-zoomed and careful measurements show that *the blade crest is almost exactly 0.08 units wide in the x direction*. **This strongly suggests that the *.stl* file was made in units of feet**. However, the *total thickness* of the plate is about 1.1 feet, which is much wider than recommended for sharp-crested weirs. The total thickness may be a design error.



Schematic enlargement of rectangular sharp-crested weir (Bos 1989)

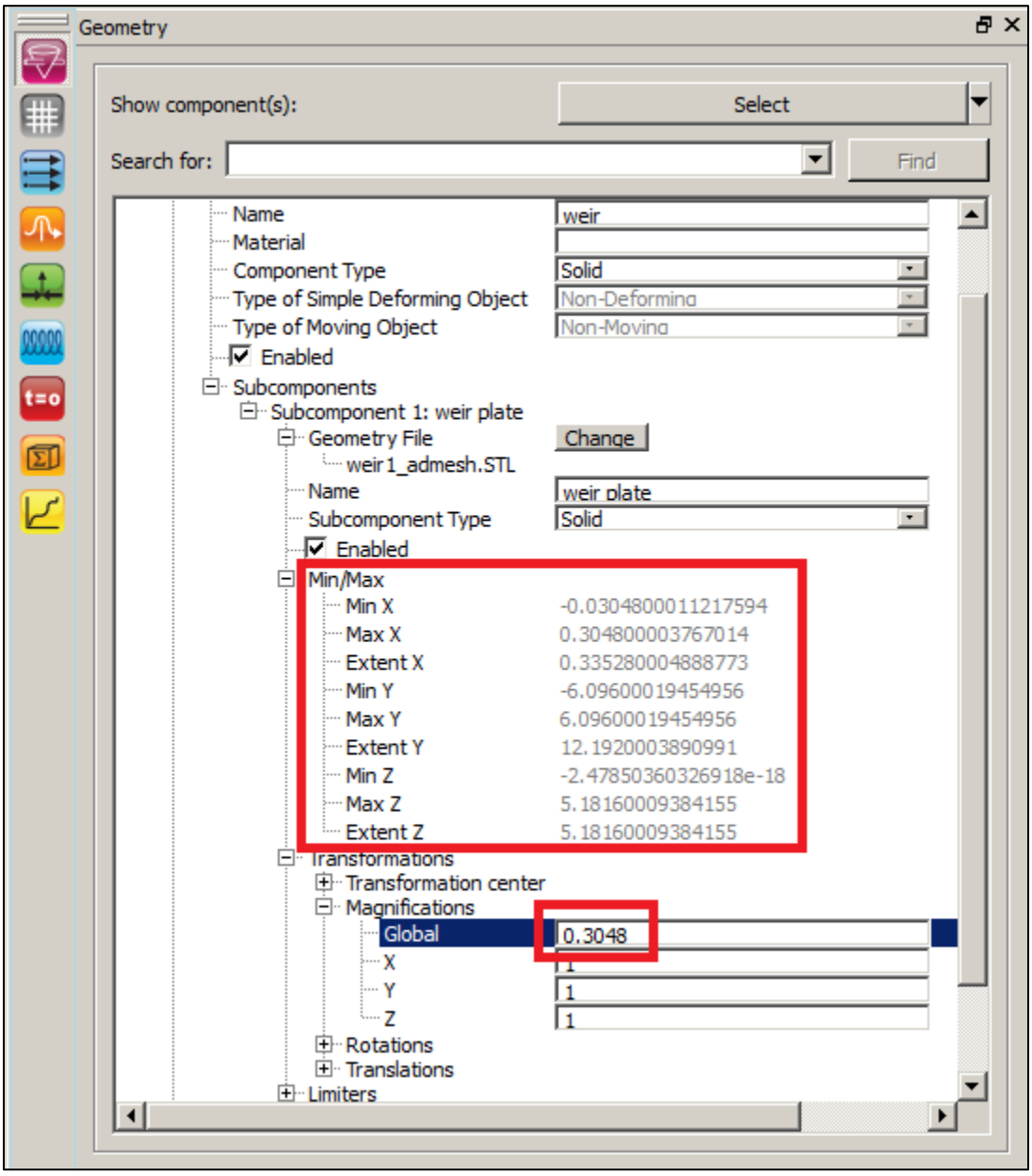
Scale Weir Subcomponent From Feet to Meters

1. Clear and close the probe points dialog. Open the **Geometry > Component 1: weir > Subcomponents > Subcomponent 1: weir plate > Transformations > Magnifications** branch of the tree in the **Geometry Window**.

2. Set **Global = 0.3048** to scale the subcomponent to meters.

3. Check in the **Min/Max** tree for the subcomponent as before: the new extents show the scaling.

4. **Save** your work.

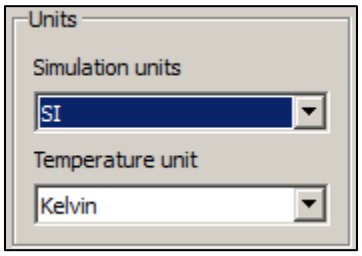


Specify the Simulation Units as SI

1. Go to the **Model Setup > General** tab.

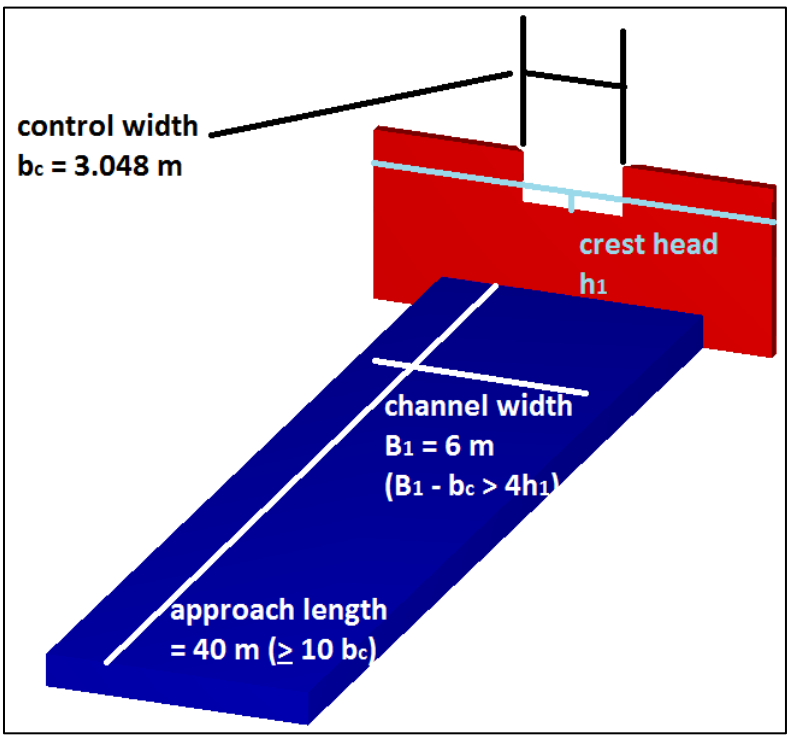
2. The default **Simulation Units** are **Custom**, which means they are unspecified (units can be anything, as long as the system is consistent for all parameters).
   a. Change the **Simulation Units** to **SI**.
   b. When prompted for the **Temperature Unit**, select **Kelvin**. This selection is unimportant since there will be no heat transfer in this simulation.

3. **Save** your work, and return to the **Model Setup > Meshing & Geometry** tab.



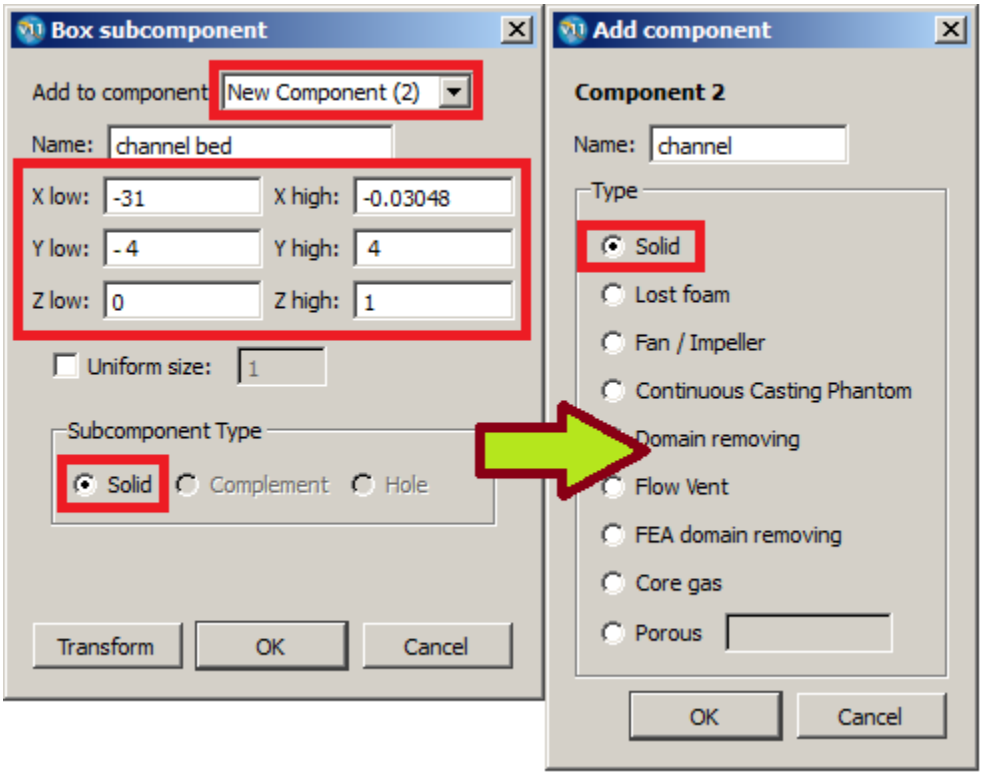
Add the Upstream Channel as Primitive Boxes

The upstream channel will be added using box primitives. The floor will be defined according to the sketch drawn below and using the dimensions determined from the **Min/Max Y** coordinates and the **Min X** coordinate of the weir subcomponent found in the tree.

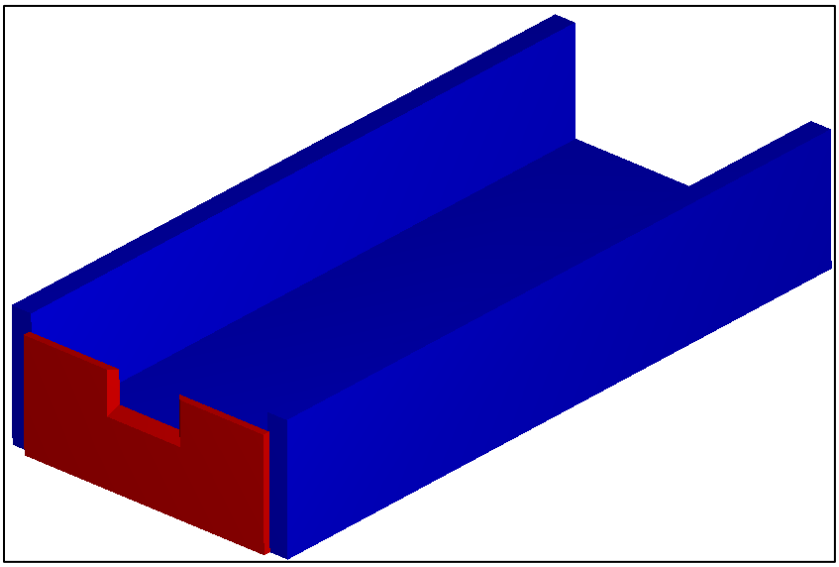


1. Select **Subcomponent > Box** from the view pane menu.

2. Enter the values shown below in the dialogs and click **OK** to accept the values. Note that the floor will be part of a new subcomponent, so it can have a different surface roughness than the weir. Name the subcomponent **channel bed** and the new component **channel**.



3. The **Geometry** tree now has two components, each with one subcomponent. Note that the new box has been added as **Subcomponent 2**: the subcomponents are numbered in the order they appear in the tree, regardless of which component they are part of. Next you will add the walls.

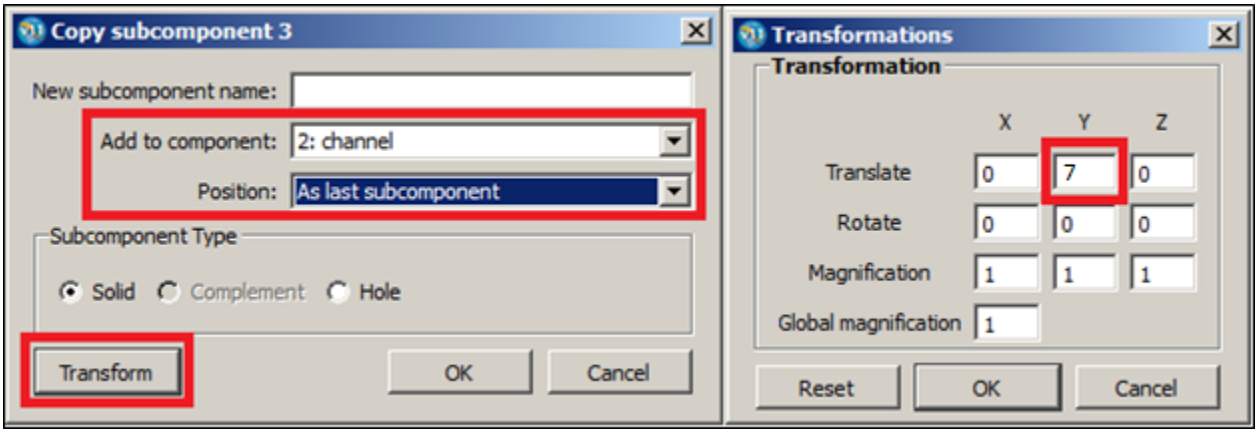


4. Add the first wall as another solid box like before but with the following specifications:
   a. **Add** the box to **Component 2: channel**.
   b. **X low = -31**, **X high = 0.3048**.
   c. **Z low = 0**, **Z high = 6**.
   d. **Y low = -4**, **Y high = -3**.

5. Expand the **Min/Max** tree under **Subcomponent 3** and verify that the values match what the input above. If there are any mistakes, fix them with the following steps:
   a. To *adjust the X,Y, and Z extents*, change the values under the **Limiters** branch.
   b. To *associate a subcomponent with a different component*, *right-click* the subcomponent name and select **Reassign**.
   c. Locate these two options now for future reference.

6. *Double-click* the wall subcomponent in the view to select it (it will turn green). *Right-click* the selected subcomponent to bring up a context menu. The title of the menu should be **Subcomponent 3**.
   a. Select **Copy…**
   b. Make sure that **Add to Subcomponent = 2: channel**.
   c. Set **Position = As Last Subcomponent**.
   d. Click **Transform**, and set **Translate Y = 7**.
   e. Click **OK**, check that you're adding the new wall to **Subcomponent 2**, and then **OK** again.
   f. *Double click* anywhere in blank space to deselect the wall.

7. **Save** your work.

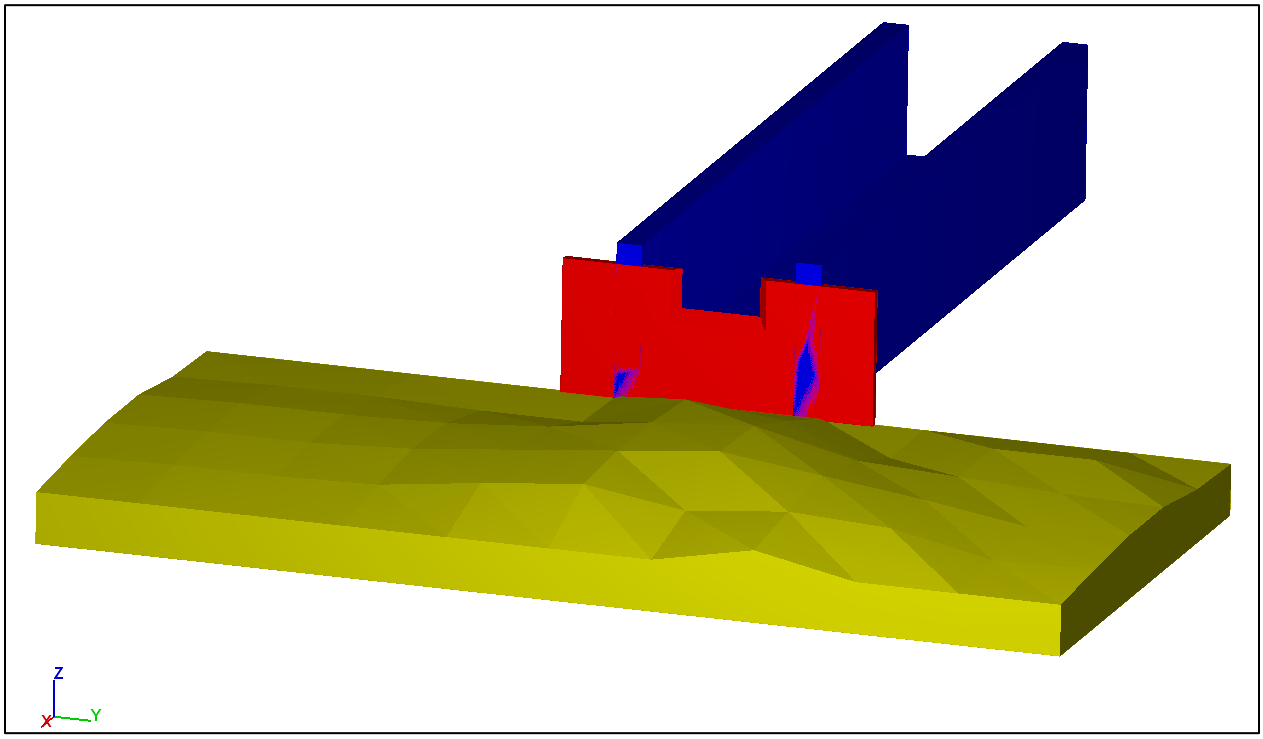


Import the Downstream Bed *.stl* File

1. Practice the *.stl* import process by importing ***Topo_Example_04_fixed.stl*** as follows:
   a. Make it a **solid Subcomponent 5** named **topo**.
   b. Make it part of a *new* **solid Component 3**, also named **topo**.

2. Check the **Geometry** tree to make sure the topography is part of **Component 3**. If it is, it will also be a unique color in the display. If it's not, *right-click* the topo subcomponent and **Reassign** it to a new component.

3. **Save** your work so far.



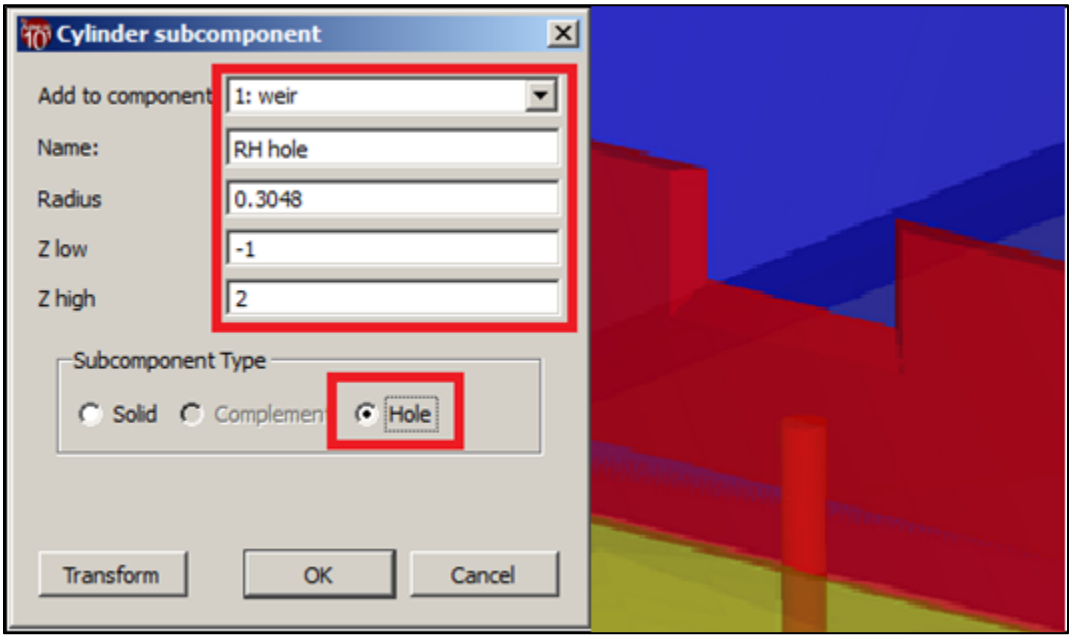
Add Subcomponent Holes to Represent Pipe Outfalls

We will now add the final pieces of the geometry by creating two cylindrical holes in the weir face. These holes will represent outflow pipes; you will specify the flow rate later using **Mass/momentum sources**. The sources will add flow without needing to extend the pipes to a domain boundary, so you don't need to make the pipes extend very far into the weir and upstream floor.

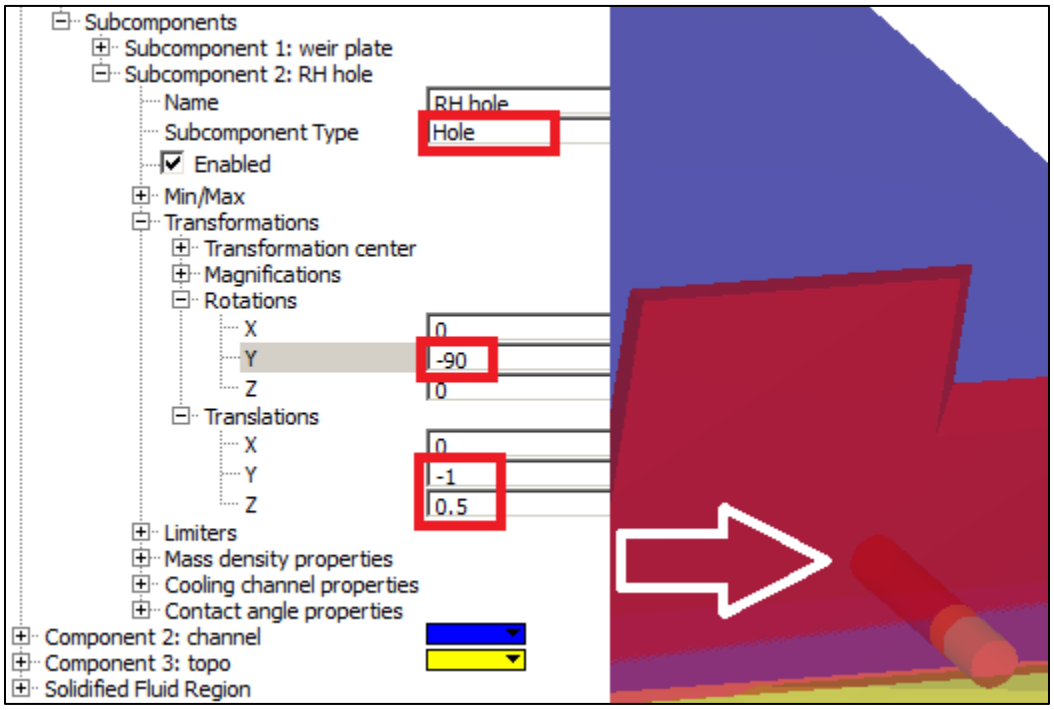
Tip
Holes replace solids in their own component. Holes can only be made as *subcomponents*. They remove volume from *other subcomponents* that come *before* them in their *component* tree. Use more than one hole subcomponent to cut through more than one component.

1. Add the hole subcomponent.
   a. Select **Subcomponent > Cylinder** from the menu above the display.
   b. Enter the values shown below, including **Add to Component**. *Cylinders are defined around the z-axis*. The next step will be to rotate and move it.

2. Check your work.
   a. Use the **Transparency** slider above the view pane to set transparency to about 75% so you can see the cylinder.
   b. Locate the new **Subcomponent 2** added to **Component 1** in the **Geometry** tree.
   c. Check that the **Subcomponent Type = Hole**. Change the type if you made a mistake.



3. Rotate and move the cylinder.
   a. Expand the **Transformation** branch and then the **Rotations** and **Translations** sections.
   b. Enter the values shown below to **rotate** and **translate** the pipe.



4. Check your work. If you made a mistake defining the cylinder size, use the **Limiters** branch of tree to fix it. If you made mistakes rotating or translating, use the **Transformations** branch.

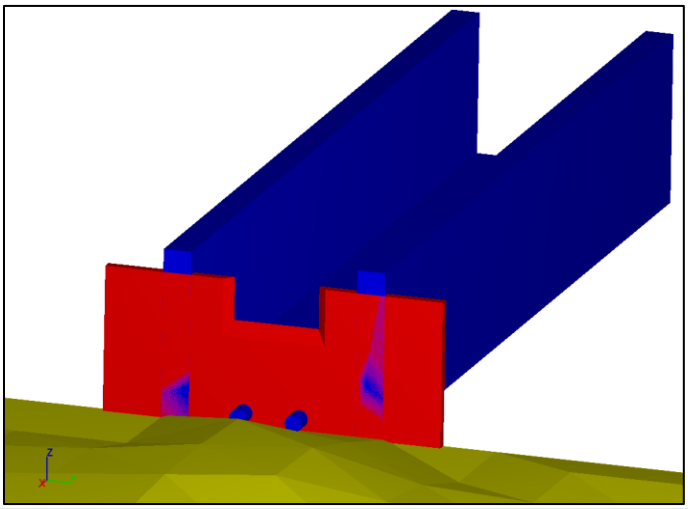
5. **Save** your work so far.

6. Create the second hole as a copy of the first one:
   a. *Right-click* the name **Subcomponent 2: Outfall 1** in the tree and select **Copy…**
   b. Name the new subcomponent **LH hole.**
   c. Add it to **component 1: weir**.
   d. Assign the **Position: As Last Subcomponent**.
   e. Make sure the **Subcomponent Type** is **Hole.**
   f. Select **OK** to create the subcomponent copy.

7. Practice finding the **Translation** branch of the new subcomponent. Change the **Y Translation** from negative to positive **1**. Check your work against the image below.

8. The holes will only be "drilled" through the weir plate (Component 1). Now re-do the holes to drill the rest of the way into the bed (Component 2).
   a. Practice: make a **Copy** of both holes by *right-clicking* either the tree or the shape.
   b. Set **Add to Component = 2: channel** for both copies.
   c. Set **Position = As Last Subcomponent** for both copies.
   d. Check that you have **9** subcomponents in **3** components, and that **#2**, **#3**, **#7**, and **#8** are the holes, as shown below. Use *right-click* **Reassign** if necessary to match the picture.

9. **Save** your work again.



- Geometry
  - Global
  - Component 1: weir
    - Name: weir
    - Material:
    - Component Type: Solid
    - Type of Simple Deforming Object: Non-Deforming
    - Type of Moving Object: Non-Moving
    - Heat Transfer Mode: Full Heat Transfer
    - Electric Potential Mode: Fixed Potential
    - Enabled
    - Subcomponents
      - Subcomponent 1: weir plate
      - Subcomponent 2: RH Hole
      - Subcomponent 3: LH Hole
  - Component 2: channel
    - Name: channel
    - Material:
    - Component Type: Solid
    - Type of Simple Deforming Object: Non-Deforming
    - Type of Moving Object: Non-Moving
    - Heat Transfer Mode: Full Heat Transfer
    - Electric Potential Mode: Fixed Potential
    - Enabled
    - Subcomponents
      - Subcomponent 4: channel bed
      - Subcomponent 5
      - Subcomponent 6
      - Subcomponent 7: RH Hole
      - Subcomponent 8: LH Hole
  - Component 3: topo
    - Name: topo
    - Material:
    - Component Type: Solid
    - Type of Simple Deforming Object: Non-Deforming
    - Type of Moving Object: Non-Moving
    - Heat Transfer Mode: Full Heat Transfer
    - Electric Potential Mode: Fixed Potential
    - Enabled
    - Subcomponents
      - Subcomponent 9: topo
  - Solidified Fluid Region

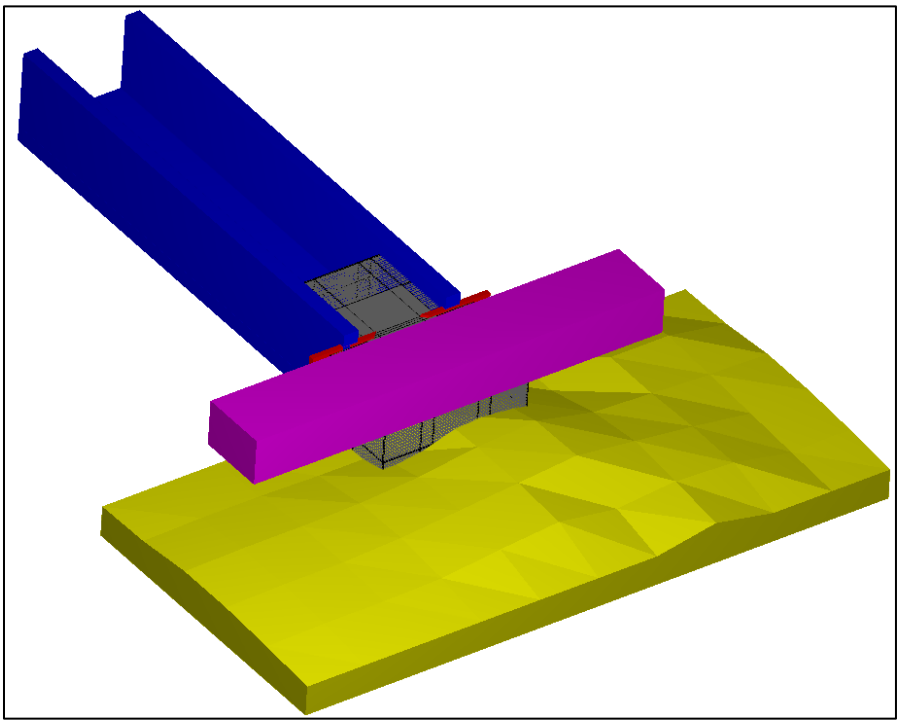
Exercise 3: Meshing & Boundary Conditions

Exercise 3: Meshing & Boundary Conditions

Problem Description & Objectives

In this exercise a computational grid will be created with boundary conditions. The grid (mesh) defines the domain: the region in which fluid calculations take place. The goal of meshing is to resolve the flow features and geometry adequately to get good results. Finer meshes (smaller cell sizes) give more precise results but require longer run-time and more memory. There is a limit to how fine the mesh should be; beyond this limit (called the *converged* mesh) reducing the cell size does not improve results. An efficient mesh may be developed by some or all of the following methods:

- Picking cell sizes to resolve the geometry and flow features according to rules of thumb,
- Specifying grid lines to resolve the surfaces of flat geometry,
- Adding domain-removing components to eliminate unnecessary cells,
- Using symmetry boundaries at the centerline to halve the domain size,
- Using nested, linking, and conforming mesh blocks to selectively resolve flow regions,
- Checking the geometric resolution of the mesh with multiple methods,
- Mesh dependency studies: reducing cell sizes in multiples of 2 until the results converge.



FLOW-3D Learning Objectives

In this exercise you will learn techniques to:

1. Import an existing simulation,
2. Make *linked*, *nested*, and *conforming meshes*,
3. Check mesh quality with ***FAVORize***, *mesh block summary tool*, *mesh information tool* and *preprocessor output files*,
4. Add *mesh planes* to resolve flat geometry,
5. Check the number of computational cells in the mesh,
6. Deactivate empty cells with *domain removing* components to improve simulation speed, and
7. Add appropriate boundary conditions.

Import an Existing Simulation File

The starting point for this exercise will be the simulation created in Exercise 2. You could just continue working on the same *prepin* file, but for practice you will copy the completed project and add it to the **Portfolio**. This method is useful when receiving a simulation via email.

1. Launch ***FLOW-3D***®, go to the **Simulation Manager** tab, and select the **Hydraulics Examples** workspace.

2. Make a new project folder:
   a. Click the **Workspace File** path in the upper right of the screen to open the workspace folder or navigate to it manually in the operating system. Note that there are two sub-folders, one for each simulation you've run so far.
   b. *Right-click* in the folder and select **New > Folder** to create a third folder. Name it **Weir_Mesh.**

3. Copy the completed simulation into the new folder:
   a. Open the *Weir_Geometry* folder.
   b. Hold *Ctrl+left-click* to select ***prepin.Weir_Geometry***, ***weir1_repair2.stl***, and ***Topo_Example_04_fixed.stl***.
   c. Press *Ctrl+C* to copy them to the clipboard.
   d. Open the new *Weir_Mesh* folder and use *Ctrl+V* to paste the three files into the new simulation folder. Make sure you moved the *repaired .stl* files.

4. Rename the prepin file:
   a. At the top of the folder window select **Organize > Folder and Search Options > View** tab. U**ncheck** the **Hide Extensions for Known File Types** option. This allows you to edit the simulation name, which is the extension to the *prepin.xxxx* file.
   b. *Right-click* the *prepin.Weir_Geometry* file and **Rename** it to ***prepin.Weir_Mesh***. Click **Yes** when prompted to change the extension.

5. Load the file on the **Simulation Manager** tab:
   a. Select the **Hydraulic Examples** workspace and select **File > Add Existing Simulation…** from the menu.

b. Browse to the **Weir_Mesh** folder and open ***prepin.Weir_Mesh***. The simulation will appear with the old name (which is stored as text in the prepin file itself).
c. **Right-click** on the new (highlighted) **Weir Geometry** and select **Rename**. Name it **Weir Mesh**.
d. Select **File > Save All** from the menu at top to save the new name in the prepin text and in the *Workspace file* that specifies the project paths.

Create Upstream, Downstream, and Nested Overlapping Mesh Blocks

The region near the weir will be characterized by sharp changes in momentum (especially at the overflow nappe) and thin geometry (the weir is only about a foot thick). This will require a fine mesh to resolve. On the other hand, the approach channel will be characterized by relatively straight, steady flow, and can tolerate a coarse mesh without too much loss of detail. Similarly the downstream flow over the topography doesn't require high resolution for this case. You will create four mesh blocks to resolve the flow and conform one of them to the weir. The first two blocks will be the upstream and downstream coarse grids. The third will overlap and bridge the two coarse blocks, and the fourth will be conforming and will be added later.

1. Make sure that **Weir Mesh** is the highlighted project on the **Simulation Manager** tab.

2. Select the **Model Setup > Meshing & Geometry** tab and show the **Mesh Window**  .

3. **Right-click** on **Mesh – Cartesian** in the tree and select **Add a Mesh Block**. New blocks are created at the origin and are 1 m x 1 m x 1m. Find the block at the bottom of the weir.

4. Expand the **Mesh - Cartesian > Mesh Block 1** tree and expand Block 1's **X Direction**, **Y Direction**, and **Z Direction** subtrees.

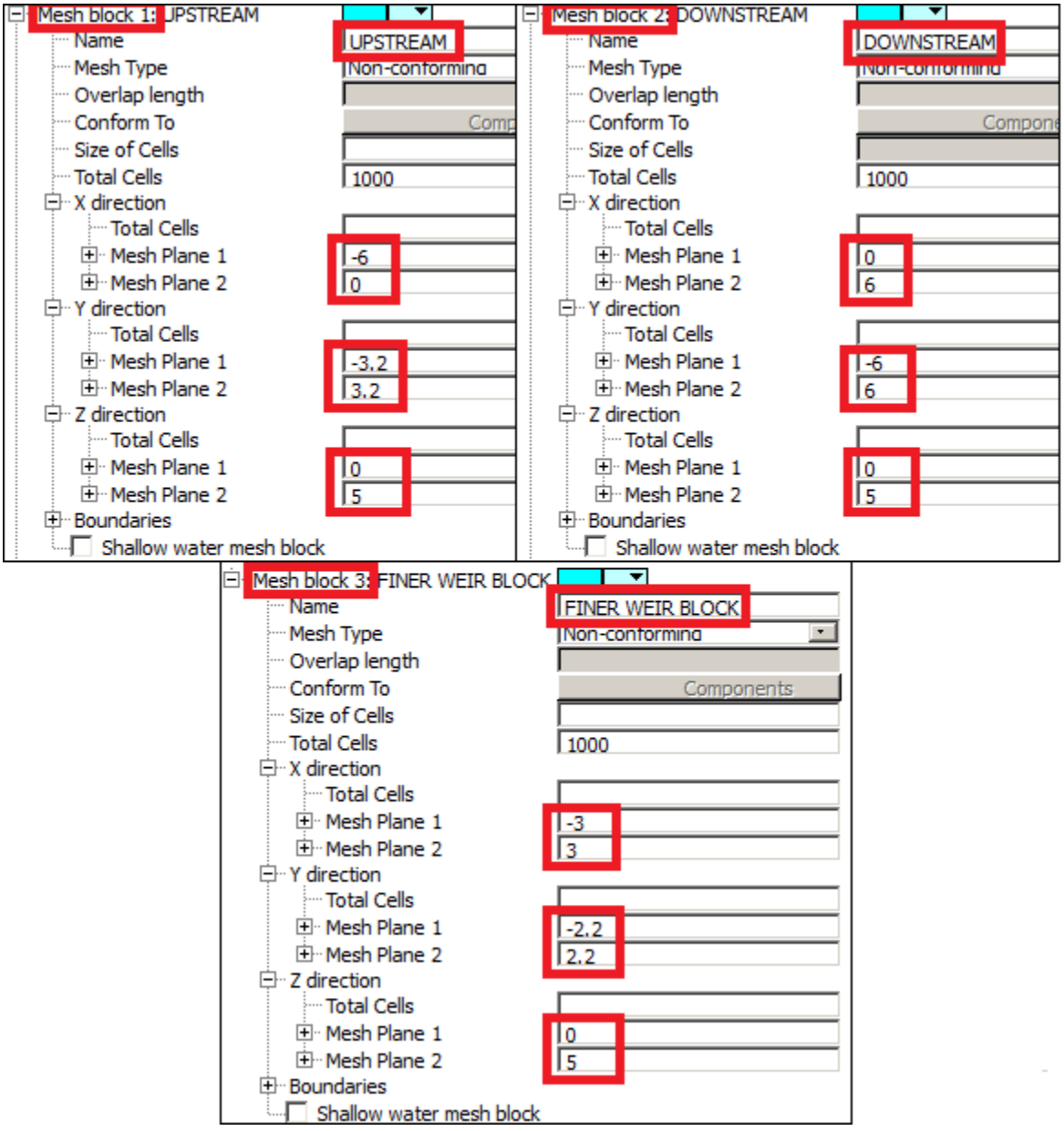
5. The first and last **Mesh Plane** coordinates for each direction are the block edges. Enter values for both **Mesh Planes** in the **X Direction**, **Y Direction**, and **Z Direction** as shown on the *left side* of the image below.

6. Repeat steps 3 – 5 using the values on the *right side* of the image below to add the DOWNSTEAM block 2.

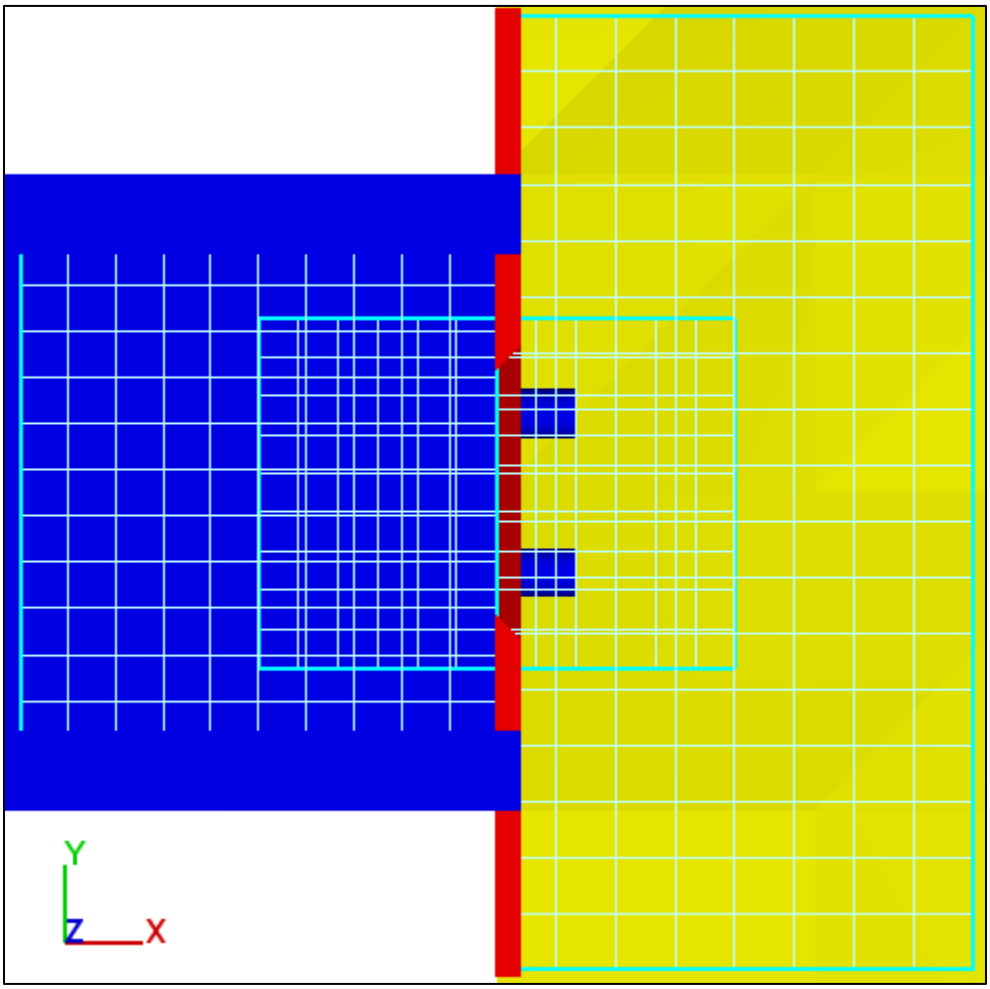
7. Repeat steps 3 – 5 using the values on the bottom side of the image below to add the FINER WEIR BLOCK block 3.

8. Check that the mesh tree matches the next page, and the mesh seen from above matches the page after that.
   a. Use the **X-Y View** button  to view the mesh from above.
   b. Turn on **Mesh > View Mode > Grid Lines** if necessary.
   c. Experiment with different levels of **Transparency**.

9. **Save** your work!



The settings shown above define the three mesh blocks that appear on the next page. Notice that the grid lines don't match at all. It's important to match grid lines between blocks or risk increasing the interpolation error. Now you'll work on making the cell sizes accurate.



Set Uniform Cell Dimensions

The domain now contains the region we are interested in. The resolution of both blocks is too coarse, so now set uniform cell sizes in each block. Later you will add more **Mesh Planes** (user-specified grid lines) to resolve tricky parts of the geometry and improve interpolation at the inter-block boundaries.

Rules of Thumb

1) The ideal cell shape is a cube.
2) Non-cubic cells should be shortest in the direction of the greatest gradient.
3) The long:short aspect ratio of any single cell should be no greater than 3:1.
4) Cells in regions of high gradients should be as close to cubes as possible.
5) Nested/linked blocks should have cell sizes that are even multiples of their neighbors.
6) The ideal cell size ratio across an inter-block interface is 2:1.

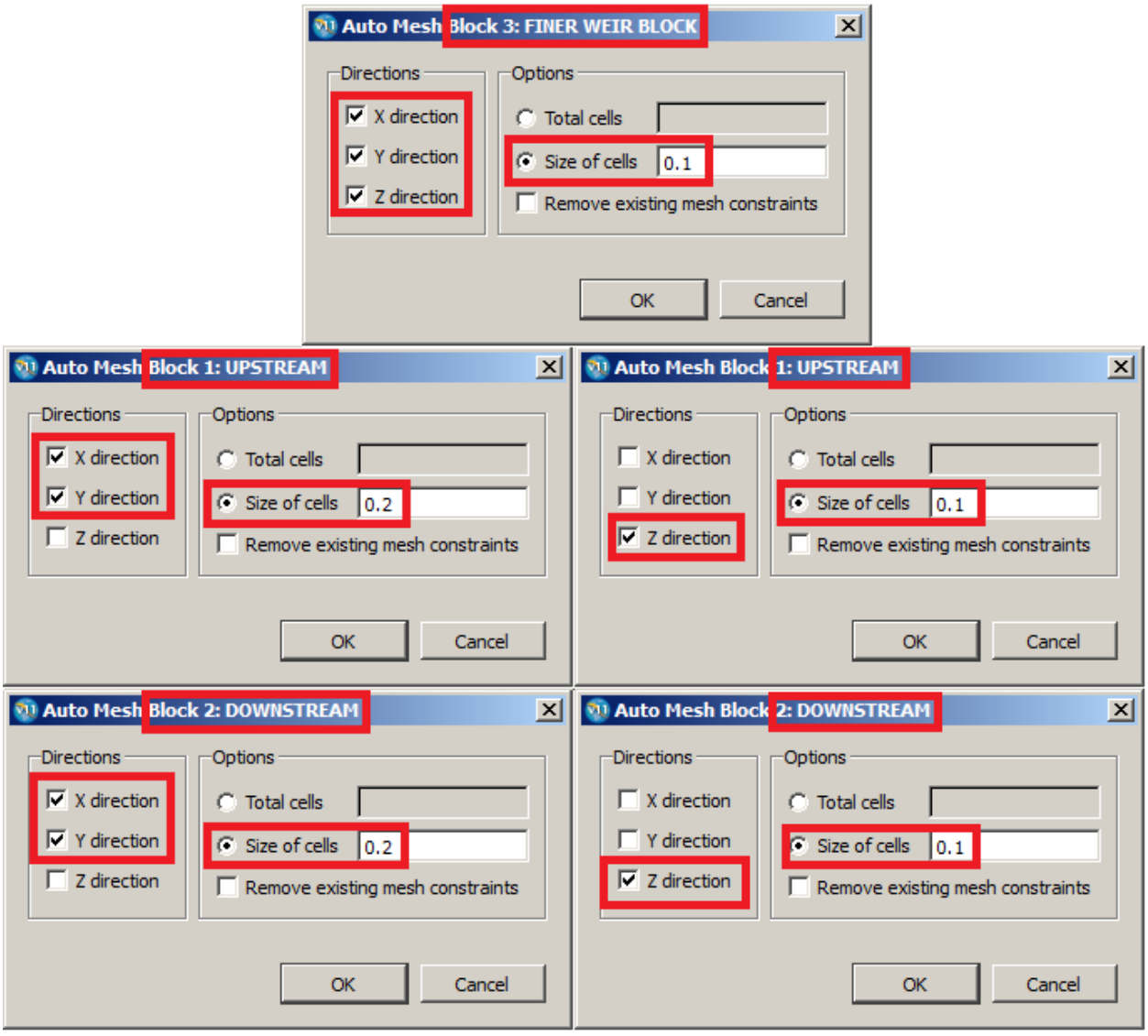
The nappe will have strong gradients (direction changes) in the X- and Z-directions, and some contraction in the Y-direction. Gradients exist in all directions for the nappe. Per the fourth rule of thumb above, *cubic cells should be used in the overlapping nested block*. Unmatched grid lines at mesh block interfaces are sources of convective volume error and artificial momentum diffusion, so the cell

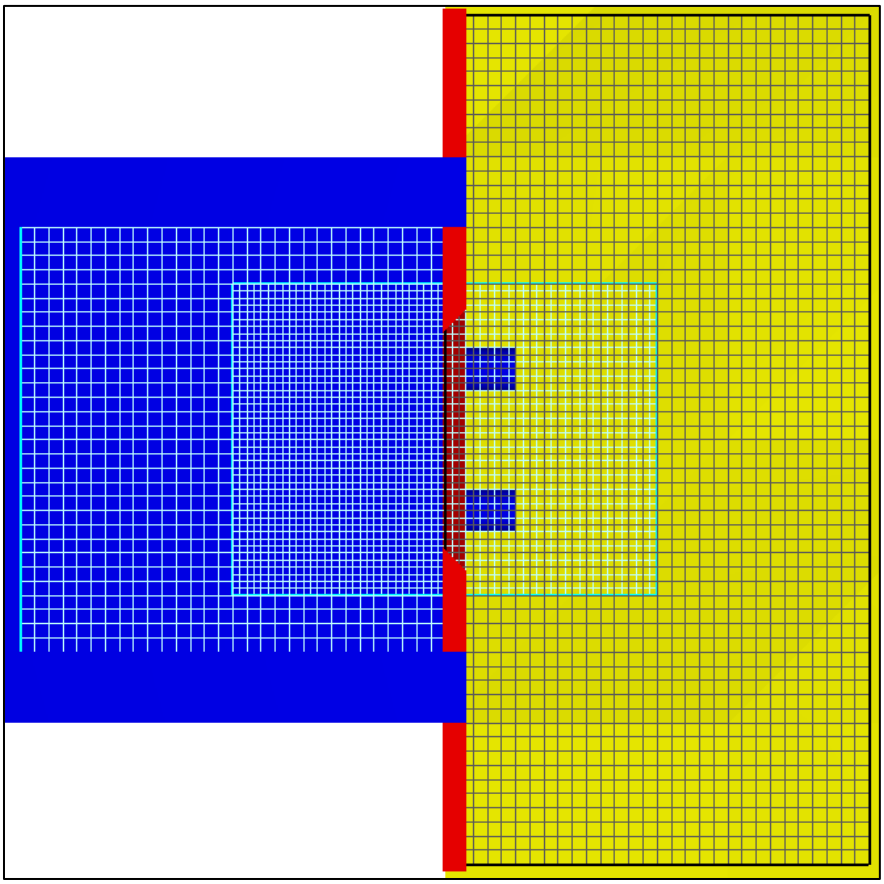
size of the containing block should be an even multiple of the smaller mesh block cell size to match the grid lines.

1. *Right-click* on the title **Mesh Block 3: FINER WEIR BLOCK** and select **Auto Mesh…**

2. You will activate directions and fill in values as shown below. Blocks 1 and 2 will get **Auto Mesh…** twice each for different vertical and horizontal cell sizes. For each step:
   a. Delete the value in **Total Cells** (not required, but makes things more clear)
   b. Select the **Size of Cells** radio button
   c. Enter **Size of Cells** as shown below.
   d. Select **OK** to accept the new criteria.
   e. Repeat for *each block* and *each direction* as shown below, clicking **OK** after each step.
   f. Check visually that *all grid lines match*. Toggle between *views* and *zoom in* as needed.

3. **Save** your work.





Check Model Resolution Using FAVOR

At this point in the geometry building process it will be useful to **FAVORize**  the geometry to determine if all the specified geometry is adequately resolved.

1. Practice what you remember from the first exercise.
2. Toggle **Mesh > Hide All/Show All** to make the geometry easier to see.
3. *Rotate* and *zoom* the view to look at the ***FAVORized*** geometry closely.
4. Check to see if there are any ***FAVOR*** issues; there should be none.

The geometry looks decently resolved. Remember that ***FAVORize*** is only an approximation because it has to fit an iso-surface through the cells and ignores the area fractions at the cell faces. It's a good first test, though, and so far the mesh resolution looks adequate.

Optional: Tweak the Mesh to Minimize Inter-block error

The simulation will run with the mesh you've already added. However, it's always a good idea to eliminate inter-block grid line mismatch. This section will sharpen your meshing skill.

1. The meshes have perfect gridline matching right now. This is because the distance between all mesh planes (within and between blocks) is an even multiple of the largest cell size, and the largest cell size is an even multiple (x2) of the smaller cell size. This is already ideal, and fine for all kinds of modeling.

2. An experienced CFD modeler who is used to body-fitted meshes might have a slight concern because the weir plate faces and wall surfaces are in the middle of cells, and not exactly on a cell *face*. This isn't a problem for ***FLOW-3D***; the partially blocked cells will get area fractions and volume fractions that correctly locate the solid surfaces inside them.

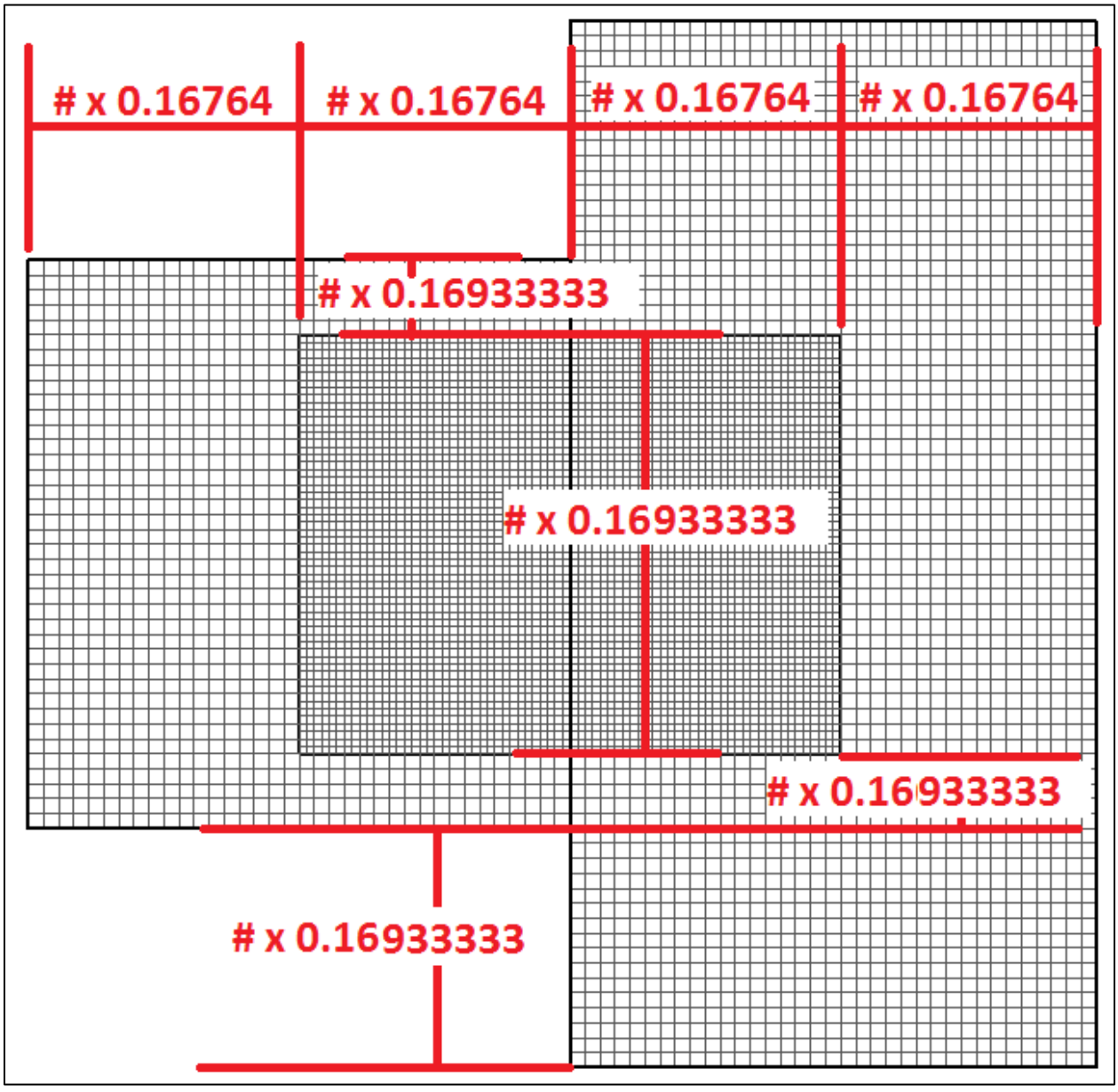
3. An experienced ***FLOW-3D*** modeler will be concerned instead with the resolution of the weir crest. The rule of thumb is that solids need at least 2 full cells and liquids need at least 4 full cells across their narrowest thickness. If you look carefully at the mesh, you'll see that the coarse blocks do not satisfy this (they have ~ 1.5 cells across the weir), and the fine block has only 3 cells across the weir. This might affect the accuracy of the flow, though you'll find with time that ***FLOW-3D*** is both robust and forgiving. Still, it would be better to make everything perfect.

4. The way to do this will be to make the cell sizes such that the distances between all mesh planes and solid surfaces in each direction (X, Y, Z) are even multiples of the largest cell size in that direction, and then to make the smaller cell sizes in that direction exactly ½ of the larger cell sizes.

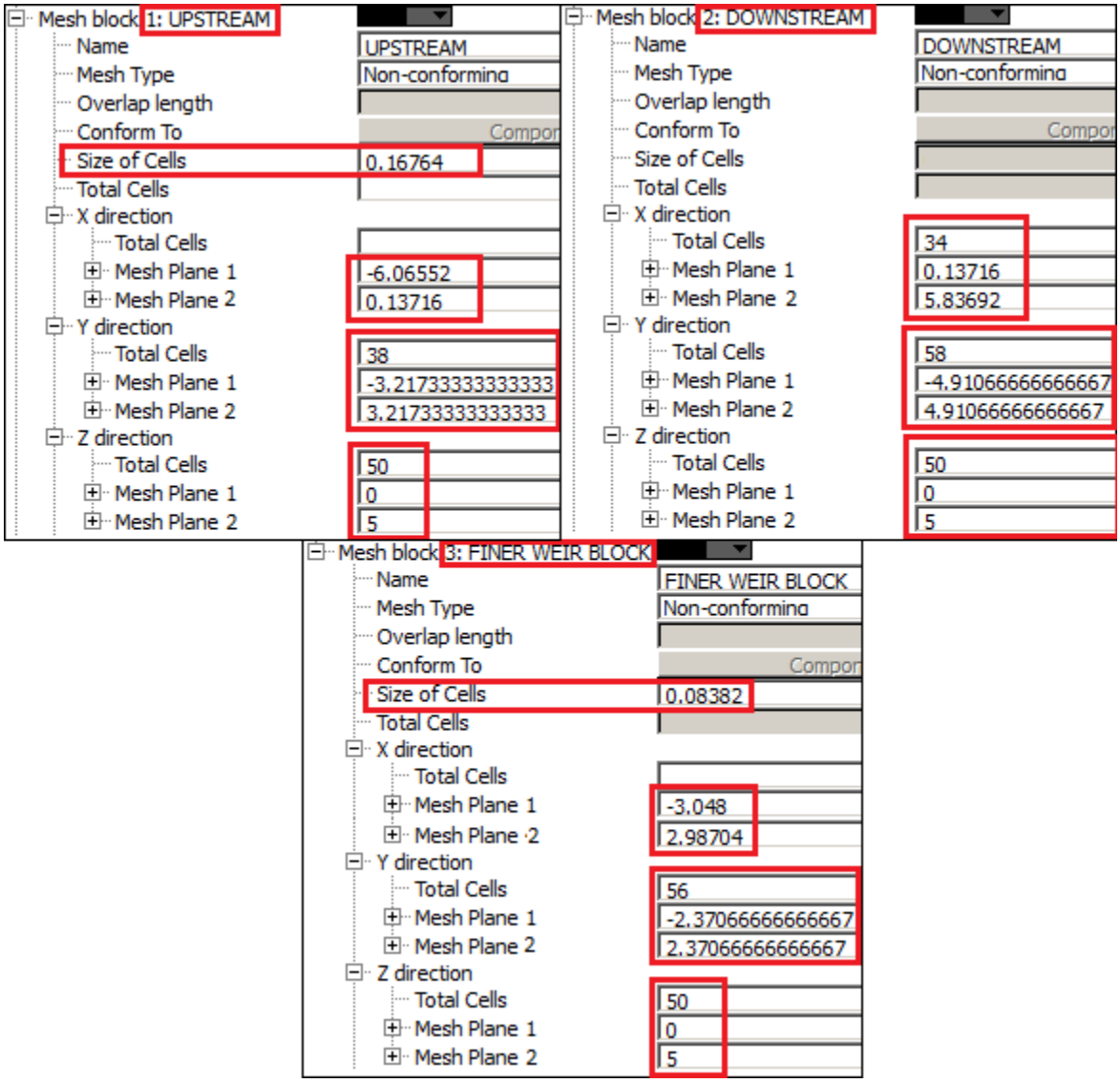
5. Don't worry about entering the values right now, just try to follow the ideas:
    a. The weir plate is **0.33528** m thick and has surfaces at **X = -0.03048** and **X = 0.3048**.
    b. Therefore the large cell size should be ½ of the weir: **0.16764 in X**, and the small cell size should be ½ of that: **0.08382 in X**.
    c. The interface line between the upstream and downstream coarse blocks should be in the middle of the weir at **X = (-0.03048 + 0.16764)**. This is the x-min *mesh plane* of the downstream block and the x-max *mesh plane* of the upstream block.
    d. The remaining edges of all three blocks must be shifted slightly now to make all distances even multiples of the cell size. This will make it so there's no stretching and grid lines match exactly in X.
    e. The same idea applies in Y, but here the important distances are between the walls (6 m) and across the weir opening (3.048 m). The cell size that goes evenly into both distances is 0.0024 m, which is far too small. Remember that *cells should ideally be as close to cubes as possible*.
    f. One solution for the Y-direction would be to move the walls, but that's a trick that's only appropriate for testing CFD code. The real world doesn't move so it's numerically agreeable. Instead, you'll have to decide which is more important: (1) resolving the corners of the weir opening blade exactly at cell corners, or (2) resolving the channel wall edges exactly at cell faces.

g. For this exercise, let's assume it's (2). The cell size that gives most closely approximates a square is then **dx = 0.16764** and **dy = 0.16933333**. You can find this by dividing the weir opening 3.048 by the already-determined X size 0.1674 to find *nearly* 18 cells across it. Divide 3.048 by 18 to find the exact cell size.

h. The Y mesh planes now need to be shifted slightly so all distances are even multiples of 0.16933333 m. The image below illustrates the idea. Don't worry about inputting numbers yet.



6. This is a time-consuming algebraic exercise, which is why it's often skipped. It is perfectly OK to skip entering the values, which are given on the next page. However, the rest of the exercise will assume that you've entered the values as shown below.



Now the grid lines in X and Y match *exactly* and align to the most important geometry *exactly*. Check this by making the geometry transparent and viewing the grid lines in the Cartesian directions. The same process can be applied to the Z direction, where the important geometry is the bed and the weir crest. Instead of using algebra to find the cell size, you'll just place a mesh plane on the crest and the floor instead.

Add Mesh Planes to Resolve Flat Geometry

Now you'll add **Mesh Planes**, which are user-specified grid lines. These will be placed along geometry surfaces to better resolve them. For this step, you will need to know the location of the edges and interfaces. You determined these in the previous exercise; to save time the values will be provided in SI units (remember scaling the geometry?)

Rules of Thumb

1) Use **Mesh Planes** *as sparingly as possible* because they're likely to cause stretching.
2) **Mesh Planes** should only be used to define *flat* solids that are *parallel to an axis*.
3) **Mesh Planes** must *never* be used to try to resolve *fluid free surfaces*.
4) **Mesh Planes** should *always* be used to match nested and overlapping block extents.

1. Place the new mesh planes in the upstream block:
   a. Under **Mesh block 1: UPSTREAM**, *right-click* **X Direction** and select **Add**.
   b. Set the **Mesh Block = 1** and the **Direction = X**.
   c. Enter **New Mesh Plane = -3.048**. This is the upstream edge of the nested block.
   d. Select **Add**. Note that the new **Mesh Plane** is highlighted in the display.
   e. Repeat *b* through *d* to put a second plane at **X = -0.03048** (the weir) in **Block 1**.
   f. Set **Direction = Y** and add two planes: **Y = -1.524** and **Y = 1.524** for the blade corners.
   g. Add two more planes in Y by copying the value from the tree (under the fine block) and pasting it into the box: **Y = -2.37066666666667** and **Y = 2.37066666666667**.
   h. Finally, add two planes in **Direction = Z**: **Z = 1** to match the channel bed, and **Z = 3.6576** to match the weir crest.

2. Repeat the process with **Mesh Block = 2**:
   a. **X = 0.3048** for the downstream face of the weir,
   b. **X = 2.98704** for the edge of the fine block,
   c. **Y = -2.37066666666667** and **Y = 2.37066666666667** for the edges of the fine block, and
   d. **Y = -1.524** and **Y = 1.524** for the weir corners.
   e. **Z = 1** and **Z = 3.6576** for the bed and crest elevations.

3. Finally add matching mesh planes for **Mesh Block = 3**:
   a. **X = -0.03048** and **X = 0.3048** to resolve the weir faces,
   b. **Y = -1.524** and **Y = 1.524** to resolve the crest edges, and
   c. **Z = 1** and **Z = 3.6576** for the bed and crest elevations.

4. Check your work and correct mistakes with the following methods:

   a. Expand the Mesh Blocks in the tree and check against the image on the next page.
   b. Select **Mesh > View Mode > Mesh Planes** to view only the specified planes.
   c. Click on **Mesh Planes** in the tree to highlight them.
   d. If there is an error, edit the **Mesh Plane** coordinate manually in the **Mesh Block** tree,
   e. or *right-click* the incorrect **Mesh Plane** and select **Delete**, then **Add** it again.

5. **Save** your work and check it again with ***FAVORize***.

Mesh block 1: UPSTREAM	
Name	UPSTREAM
Mesh Type	Non-conforming
Overlap length	
Conform To	Compor
Size of Cells	0.16764
Total Cells	
X direction	
Total Cells	
Mesh Plane 1	-6.06552
Mesh Plane 2	-3.048
Mesh Plane 3	-0.03048
Mesh Plane 4	0.13716
Y direction	
Total Cells	38
Mesh Plane 1	-3.21733333333333
Mesh Plane 2	-2.37066666666667
Mesh Plane 3	-1.524
Mesh Plane 4	1.524
Mesh Plane 5	2.37066666666667
Mesh Plane 6	3.21733333333333
Z direction	
Total Cells	50
Mesh Plane 1	0
Mesh Plane 2	1
Mesh Plane 3	3.6576
Mesh Plane 4	5

Mesh block 2: DOWNSTREAM	
Name	DOWNSTREAM
Mesh Type	Non-conforming
Overlap length	
Conform To	Compor
Size of Cells	0.166666666666667
Total Cells	
X direction	
Total Cells	34
Mesh Plane 1	0.13716
Mesh Plane 2	0.3048
Mesh Plane 3	2.98704
Mesh Plane 4	5.83692
Y direction	
Total Cells	58
Mesh Plane 1	-4.91066666666667
Mesh Plane 2	-2.37066666666667
Mesh Plane 3	-1.524
Mesh Plane 4	1.524
Mesh Plane 5	2.37066666666667
Mesh Plane 6	4.91066666666667
Z direction	
Total Cells	50
Mesh Plane 1	0
Mesh Plane 2	1
Mesh Plane 3	3.6576
Mesh Plane 4	5

Mesh block 3: FINER WEIR BLOCK	
Name	FINER WEIR BLOCK
Mesh Type	Non-conforming
Overlap length	
Conform To	Compor
Size of Cells	0.08382
Total Cells	
X direction	
Total Cells	
Mesh Plane 1	-3.048
Mesh Plane 2	-0.03048
Mesh Plane 3	0.3048
Mesh Plane 4	2.98704
Y direction	
Total Cells	56
Mesh Plane 1	-2.37066666666667
Mesh Plane 2	-1.524
Mesh Plane 3	1.524
Mesh Plane 4	2.37066666666667
Z direction	
Total Cells	50
Mesh Plane 1	0
Mesh Plane 2	1
Mesh Plane 3	3.6576
Mesh Plane 4	5

The generalized way to determine the optimal location of grid lines (for your later reference):

- Place internal mesh planes (internal means they are not the ends of the blocks) only where absolutely necessary to resolve geometry. In this exercise, mesh planes could be omitted altogether, *except that it's important you learn how to use them*.

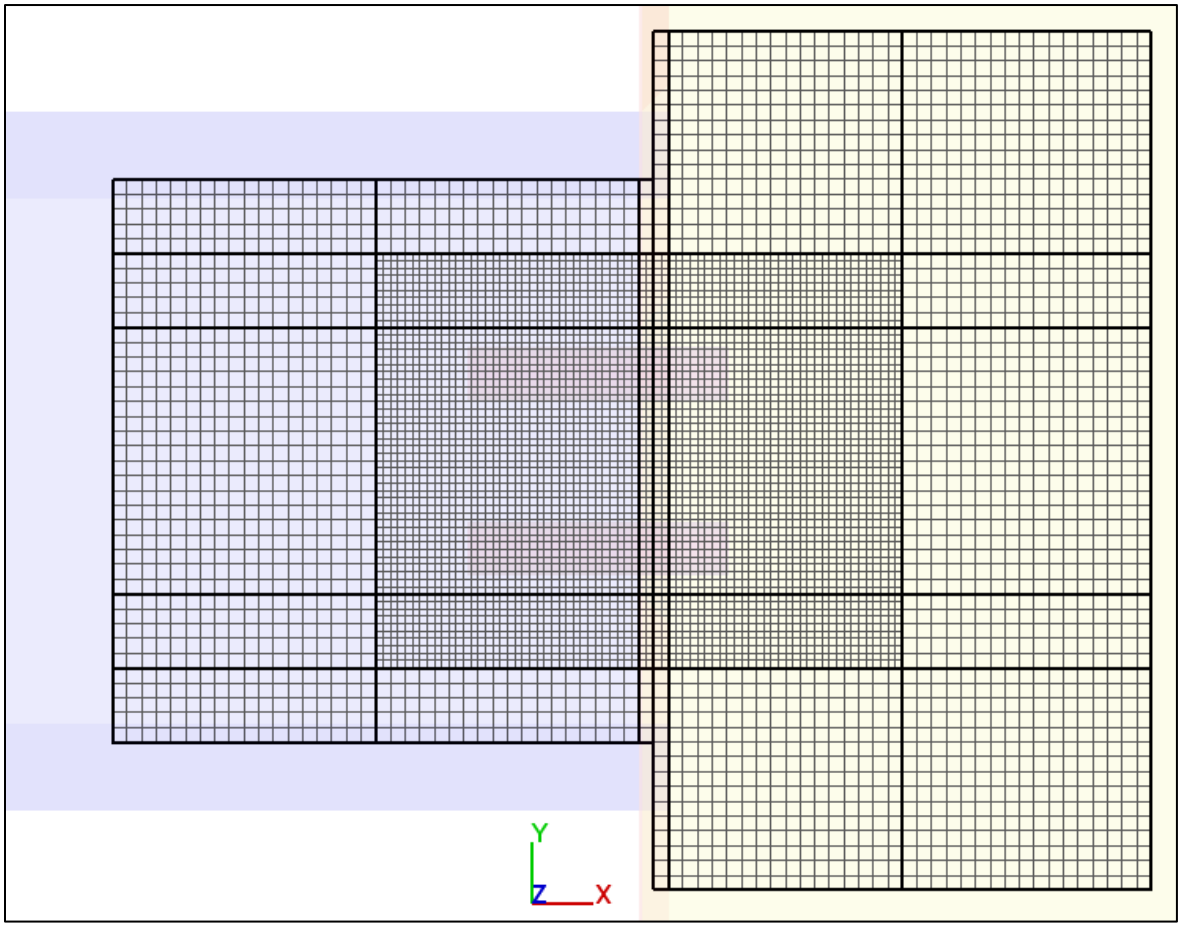
- Start by selecting the largest cell sizes in X, Y, Z so that there are even multiples of the largest cells between every set of adjacent mesh planes and important flat surfaces. Surfaces that are *not* flat are well-resolved in ***FLOW-3D*** without any effort on your part!

- Move block boundaries by starting at the required mesh plane location and adding or subtracting the largest cell size repeatedly until the value is close to the desired boundary location. Place the boundary there. If the boundary is outside the geometry, consider extending the geometry to the boundary.

- Continue the process in all directions until all mesh planes and important *flat surfaces* are separated by even multiples of the largest cell size.

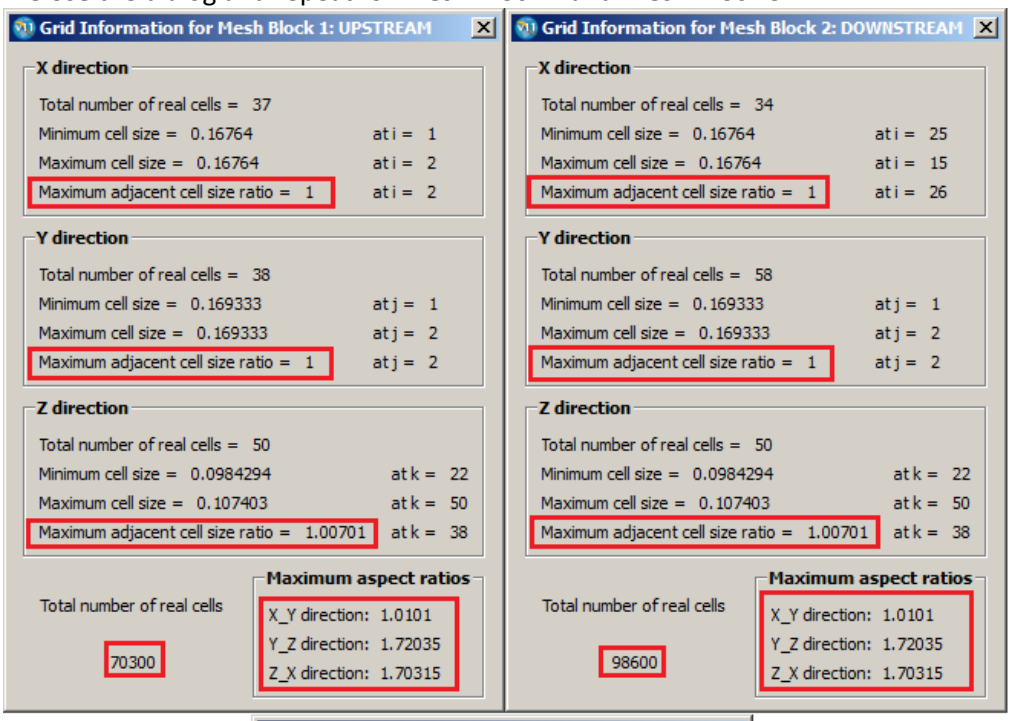
- Make cells in finer blocks exactly ½, ¼, etc. of the largest cell size.

- Repeat mesh planes in all adjacent blocks to tie the grids together precisely.



Check the Mesh Aspect Ratios with the Mesh Information Tool

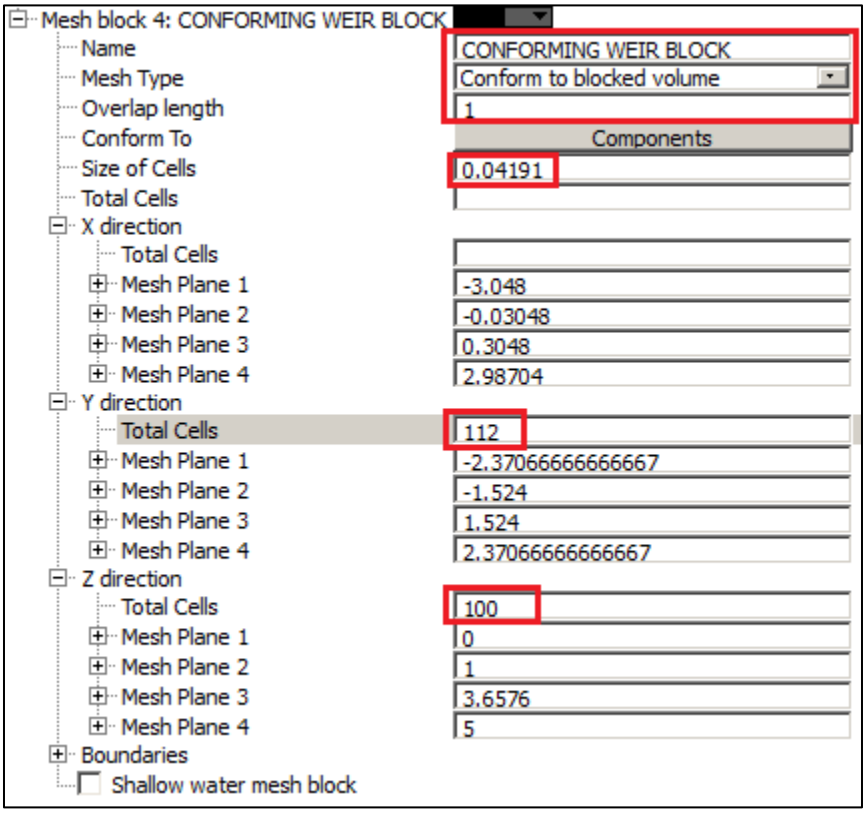
1. This is a very important tool for checking your mesh quality:
   a. *Right-click* **Mesh Block 1** and select **Mesh Information…**
   b. Check the **Maximum Adjacent Cell Size Ratios** (X, Y, and Z): they are close to 1 and **< 1.25**.
   c. Check the **Maximum Aspect Ratios** (ZY, YZ, and ZX): they are **< 3**. *1 would be optimal*.
   d. Check the output against that shown below.
   e. **Close** the dialog and repeat for **Mesh Block 2** and **Mesh Block 3**.



Make a Conforming Mesh Block

This exercise includes a lot of steps that might not be necessary to run the case, but are meant to show you your way around the software. The conforming mesh block you're about to create, however, is a powerful capability introduced in ***FLOW-3D*** v11.

1. *Right-click* **Mesh Block 3: FINER WEIR BLOCK** and select **Copy…**
2. Keep **Translations = 0** and click **OK**. Now there is a second, identical block on top of the fine one.
3. Open the tree for **Mesh Block 4** and rename it **CONFORMING MESH BLOCK**.
4. Set **Mesh Type = Conform to Blocked Volume**.
5. Set **Overlap Length = 1**. The length is in meters since SI units are being used.
6. Click the **Conform To Components** button and uncheck boxes so only **Component 1** is active.
7. Halve the cell sizes or double the cell counts to refine by a factor of 2. Type **/2** or ***2** after each value in the highlighted inputs below. ***FLOW-3D*** will do the math.
8. Check that the final values for **Mesh Block 4** are as shown below.
9. **Save** your work!



Now the fourth mesh block will include even finer cells, but only within a distance of one meter of the weir plate. The conforming mesh will only appear *as defined* in *2-D plots*, like those found from pre-processing the simulation. Note also that when the simulation is post-processed, even the unused parts of Block 4 will contribute to the memory usage of the post-processor. When simulating, the unused parts do not contribute to the resource demand.

Check the Total Cell Count with the Mesh Block Summary Information Tool

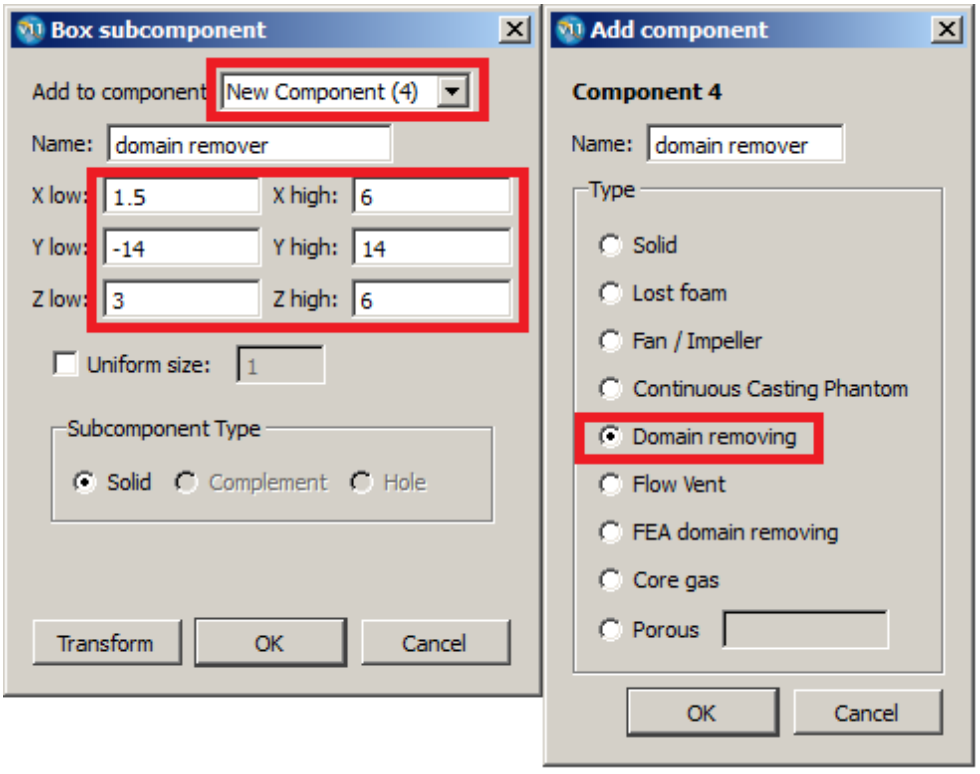
1. *Right-click* **Mesh - Cartesian** and select **Mesh Block Summary Information…**
2. Check the text display that appears. The total number of *real cells* defined in all the blocks is **1,983,000** cells. More "ghost cells" that are associated with boundary conditions will be added when you preprocess the simulation, and cells in solids and unused parts of the conforming block will be removed, so this is a preliminary estimate.
3. Two million cells is a lot for a classroom simulation. Removing some cells is advised.

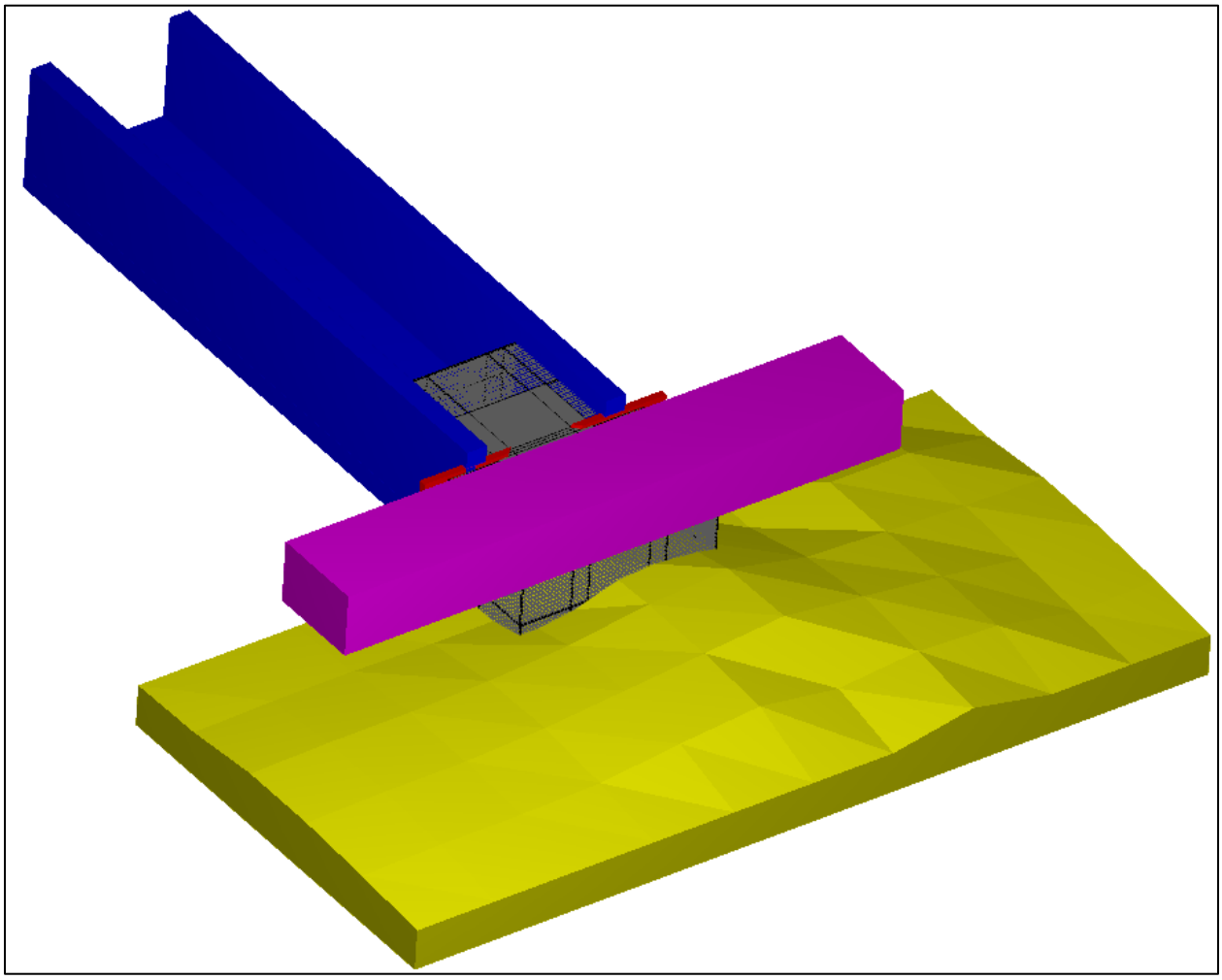
Deactivate Cells with a Domain Removing Component

Unnecessary cells can be deactivated to save memory and time, but only when they will *definitely never contain flow*. It is apparent that the downstream region is not likely to be filled to the top with running water. You can safely deactivate the cells above a reasonable Z-elevation. You could do this in two ways:

- Deactivate cells by adding a *solid component*. Solids will deactivate cells unless *solid heat transfer* physics are active. The solids will appear in post-processing.
- Deactivate cells by adding a *domain-removing component*. These will deactivate cells even if heat transfer is on. They are also transparent when viewing results. All subcomponents of domain-removing components deactivate cells regardless of whether they are solids or holes.

1. Start to make a new primitive box (**Subcomponent** menu **> Box**).
2. Enter the values shown below and click **OK**.
3. Select **Domain Removing** as the **Type**.
4. Check the component placement against the image below, and **Save** your work.





Add Boundary Conditions

Upstream boundaries are typically one of the following:

- **Volumetric Flow Rate** or **Velocity** type boundaries specify a *free surface elevation* and flow *rate* or *velocity*. They represent a design storm or are derived from field measurements.

- **Stagnation Pressure** type boundaries specify a *free surface elevation* and represent a fluid source with *zero velocity* but known depth (i.e., a reservoir). Water levels above the specified elevation exit the domain, and levels below the specified elevation (or pressure) draws fluid in.

The downstream boundary condition depends in large part on the *Froude number* of the downstream flow. There are two boundaries that are appropriate for downstream in channels.

- **Pressure** type boundaries are appropriate when the flow is *subcritical*. They represent tailwater, and require an estimate of the downstream *free surface elevation*.

- **Outflow** type boundaries are appropriate when the flow is *supercritical*. They do not transmit information back into the domain, so they are appropriate when the flow is faster than gravity waves.

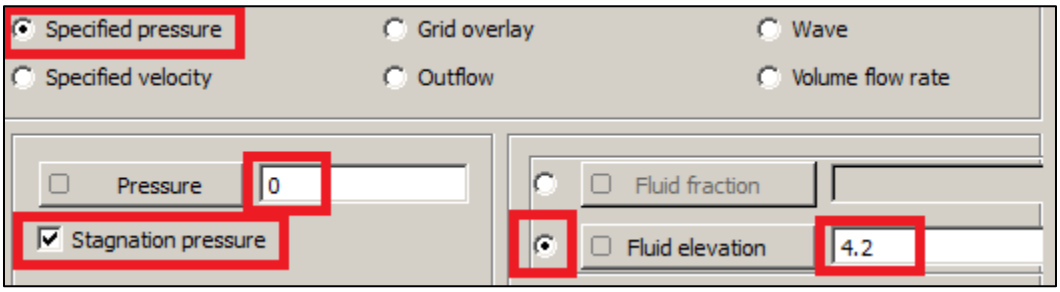
The upstream boundary condition here will be specified as pressure-type using total head H. The downstream boundary flow is expected to be *sub-critical* and you will use *Pressure boundaries* with *tailwater depths*. The top of the domain is open to the atmosphere (*Pressure-type* with *Fluid Fraction = 0*), and the bottom of the domain is a solid surface (*Wall*).

The pressure in the simulation can either be gauge (atmospheric pressure = 0) or absolute (atmospheric pressure ~ 101 kPa). The atmospheric pressure (gauge or absolute) must be specified at *all Pressure-type boundary conditions*, and as the *initial void pressure*.

Specify the boundaries as follows:

10. Open **Mesh block 1: UPSTREAM > Boundaries**. Each boundary has a one- or two-character identifier that can be clicked on to change it. The default is 'S' for **Symmetry** boundary.

11. Set the upstream boundary:
    a. Click on the 'S' next to **X Min** to open the boundary dialog.
    b. Select the **Specified Pressure** button.
    c. Select **Fluid Elevation** and set the value = **4.2** (measured from Cartesian z = 0).
    d. Specify **Pressure = 0**. This means that the pressure at the free surface is local "gauge" atmospheric pressure.
    e. Leave **Stagnation Pressure** checked. The stagnation option means that the water velocity outside the mesh (in the ghost cells) is set to *zero*. This corresponds to the idea that the free-surface elevation specified is the total head (static + velocity). **Stagnation pressure** is *required* for upstream pressure boundaries, because it is stable. When stagnation pressure is turned off, the boundary condition is called **static pressure**. Static pressure allows the solver to calculate a velocity in the boundary ghost cells, and this can lead to run-away flow unless there is significant back-pressure on the boundary.
    f. Select **OK** to close the dialog.
    g. Leave the rest of the boundaries in Block 1 as **Symmetry (S)** for now.
    h. **Save** your work.



12. Set the downstream boundaries:
    a. Repeat the process for **Block 2**: set the **X Max**, **Y Min**, and **Y Max** boundaries to:
    b. **Pressure**-type (with **stagnation pressure** option activated),
    c. **Pressure = 0** (at the free surface),
    d. **Fluid Elevation = 1.5** m.

13. Set the ceilings:
    a. Go through all four blocks.
    b. set **Z Max = Pressure (P)** with **stagnation pressure** option activated),
    c. Keep **Fluid Fraction** activated instead of the elevation.
    d. Set **Fluid Fraction = 0** (meaning the boundary is void/air, and will remove water).

The Z-Max boundaries could be **Symmetry** or **Wall**; it doesn't matter as long as fluid never touches them. Sometimes a model will have splashing, and when drops hit a wall or symmetry boundary they can stick and drip and make the slow down the solver. The **F = 0 Pressure boundary** approach eliminates that possibility by removing any fluid that splashes the top of the domain. *Results should always be checked to make sure that large quantities of fluid aren't accidentally removed this way*!

14. Set all **Z Min** boundaries as **Wall** with no tangential velocities.

15. Repeat steps 4 and 5 for **Mesh Block 2: FINE**, which shares Z-boundaries with the containing block.

16. Mesh Block 2 is completely enclosed by Block 1 on the X and Y sides. These boundaries will be reset to 'inter-block' type by the preprocessor. Leave them as **Symmetry**-type.

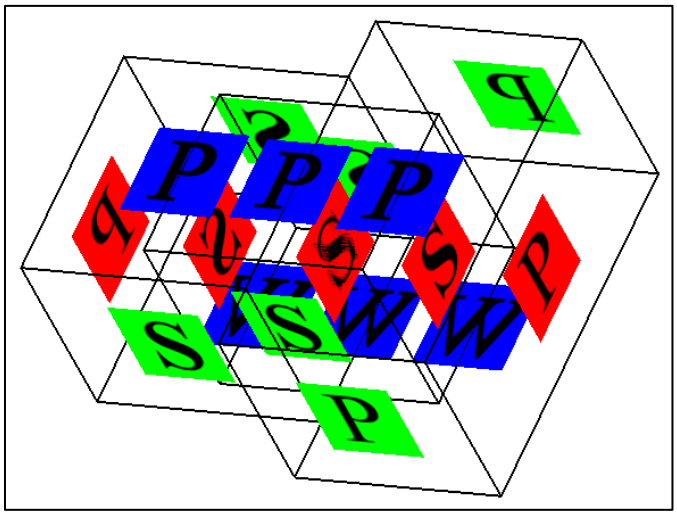
17. **Save** your work.

18. Select the **Boundary Condition Display button** 

(or **Mesh > View Mode > Boundaries**) in the display pane to show the boundary types in the display and check your work against the image below. Make any necessary fixes.

19. Toggle the **Boundary Condition Display button** 

to return to the normal mesh block view.



Set Fluid Density

This step is required to be able to preprocess the model.

1. Go to the **Model Setup > Fluids** tab.
2. Open **Properties > Fluid 1 > Density Properties**.
3. Set **Density = 1000**.
4. **Save**.

Preprocess the Mesh

Preprocessing the mesh is an important final step. From this you can find many summaries, including:

- The final number of cells including ghost cells and cell deactivation effects,
- The exact location of all grid lines and cell centers,
- Adjustments made to the geometry to eliminate "sliver cells" (described in more detail in the lectures on *Numeric Options* and *Troubleshooting*), and
- Disagreement between geometry resolution at inter-block boundaries.

1. Select **Simulate > Preprocess Simulation > Local** from the menu at the top. The view will change to the **Simulation Manager** tab, and the **solver text** will show the steps as they are completed. When it shows "Preprocessor Done", the results are ready to view.

2. Select **Diagnostics > Preprocessor Errors** to view the ***prperr.Weir_Mesh*** file. This file contains important messages when inconsistencies are detected by the preprocessor. It will warn you that different resolutions are causing different open areas between blocks. This is common when cross-boundary aspect ratios are greater than 2:1. All the reported warnings involve Block 4, which has the same extents as Block 3 but only half the resolution. This does not matter because the actual extents of Block 4 after it conforms to the weir will be inside Block 3, and the ratio will be 2:1 there.

3. Select **Diagnostics > Preprocessor Summary** to view the ***prpout.Weir_Mesh*** file generated by the preprocessor. You can also read the diagnostics files in any text editor.
   a. There is an enormous amount of information in this preprocessor summary, which documents every step taken by the preprocessor.
   b. Click on **Search** at the top of the dialog and search for **mesh**. The first instance of the term is at the beginning of the mesh block namelist. **Namelists** are sections of the ***prepin*** file; the preprocessor goes through the namelists in order and interprets them. Press **Cancel** to close the search box and scroll down to find the section labeled **mesh generated in x direction**:
   c. Click on **Search** at the top of the text dialog and search for **cell summary**. Review the information, which should appear as it does below. There are **144,385 active cells** out of **202,498 total cells**; both counts include boundary cells. The solid and domain-removing components deactivated **29%** of the cells. This correlates directly to a reduction in memory use and a substantial runtime efficiency improvement. **Fully blocked real cells** are cells that are filled with solid geometry but are still active for heat transfer: there are none here.

cell summary:

total number of cells (active and passive)	=	2221792
fluid sub-domain cells	=	783824
solid sub-domain cells	=	66570
cells common to both solid & fluid sub-domains	=	52888
active cells include:		
real cells (used for solving flow equations)	=	652642
open real cells	=	619659
fully blocked real cells	=	32983
external boundary cells	=	33336
inter-block boundary cells	=	111528

d. Search again for **areas at inter-block**. Review the information. The inter-block area fraction differences seems high, but most of them misleadingly apply to a conforming block (#4), while Block 1 and Block 2 only interface directly with each other across the solid sides of the weir plate. For *this case*, the inter-block area warnings can be ignored.

e. Search again for **area/vol**. Cells that have a large open facial area to volume ratio are likely to have high fluxes which can limit the time step (slow down runtime). ***FLOW-3D*** normally adjusts these by adding 'virtual' volume to cells with A/V ratio greater than a specified threshold called **AVRCK** (by default 3.1). This speeds up the solution and makes it more stable at the cost of a (usually) slight degree of accuracy.

   In this simulation, ***FLOW-3D*** added **0.21** m^3 of volume and the total open volume in the mesh is **383** m^3. *The total adjustment is less than **0.06%** of the total volume*: an insignificant loss of accuracy for a probable efficiency gain.

4. Select **OK** to close the text file.

Exercise 4: Fluid, Physics, & Initial Conditions

Hydraulics Exercise 4: Fluid, Physics, Initialization

Problem Description & Objectives

Continuing from the previous exercise, a fluid (water) will be specified for the simulation and fluid regions will be initialized within the domain. Gravity and turbulence models will be activated, and mass/momentum sources will be added to the simulation to create discharge from the pipe culverts. Final steps before running the model will include adding some sampling locations and specifying output.

FLOW-3D Learning Objectives

In this exercise you will learn techniques to:

1. Copy an existing simulation to a new folder in ***FLOW-3D***,
2. Activate and specify parameters for the **gravity** and **viscosity & turbulence** physics models,
3. Initialize **fluid regions** and boundaries with **velocity** and **hydrostatic pressure**,
4. Load fluid properties from the **materials database**,
5. Manually edit fluid properties, including **limited compressibility**,
6. Set the simulation **finish time**,
7. Add **mass/momentum fluid sources**,
8. Add **history probes** for point measurements, and
9. Add **flux surface baffles** for flow rate measurement.

The starting point for this exercise is the simulation created in Exercise 3. If you have successfully completed Exercise 3 you may choose to simply make a new copy of that simulation as described below. If you have not successfully completed Exercise 3, you may download or copy the file ***prepin.Weir_Mesh_Completed*** and the geometry files to a new simulation folder and import it as an existing simulation.

Make a Simulation Copy

1. Launch ***FLOW-3D®***, go to the **Simulation Manager** tab, and *right-click* the **Hydraulics Examples > Weir Mesh** simulation that you just finished. Select **Add Simulation Copy…**

2. Enter **Weir Finish** as the simulation name.

3. Leave the **Create Subdirectory Using Simulation Name** checked to keep files organized.

4. Check the boxes for ***weir1_repaired2.stl*** and ***Topo_Example_04_fixed.stl*** to copy the geometry.

5. Select **OK**.

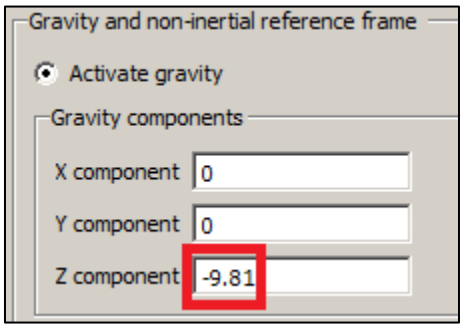
Activate the Gravity Physics Model

1. Navigate to the **Model Setup > Physics** tab.

2. Click on **Gravity and Non-Inertial Reference Frame**. The physics models are in alphabetical order from top to bottom, left to right.

3. Enter **Z Component = -9.81**. The negative sign points the gravity vector downward.

4. Click **OK** to exit the dialog. A green check mark now appears next to **Gravity**.



Activate the RNG Turbulence Physics Model

The presence or absence of turbulence can be determined using Reynolds number relations; in our case it can be assumed that the flow is turbulent. There are a number of choices for turbulence models. The most robust and widely applicable turbulence model is the **k- ε RNG (Renormalized Group)** model. In the next few steps you will activate the RNG turbulence model and specify **dynamically-computed maximum turbulent mixing lengths**. The actual turbulent mixing length is the characteristic turbulent length scale, a mathematical factor that roughly corresponds to the size of the smallest turbulent eddy that needs to be resolved in terms of turbulent transport; eddies that are smaller than the length scale are approximated as an average and dissipate to heat. The maximum turbulent mixing length is an upper stability bound to prevent the turbulence model from over-estimating the length scale. It can be global and constant (which requires some understanding of the limiting mixing length), or calculated in each time step (dynamically) for each cell based on empirical formulations.

Rule of Thumb

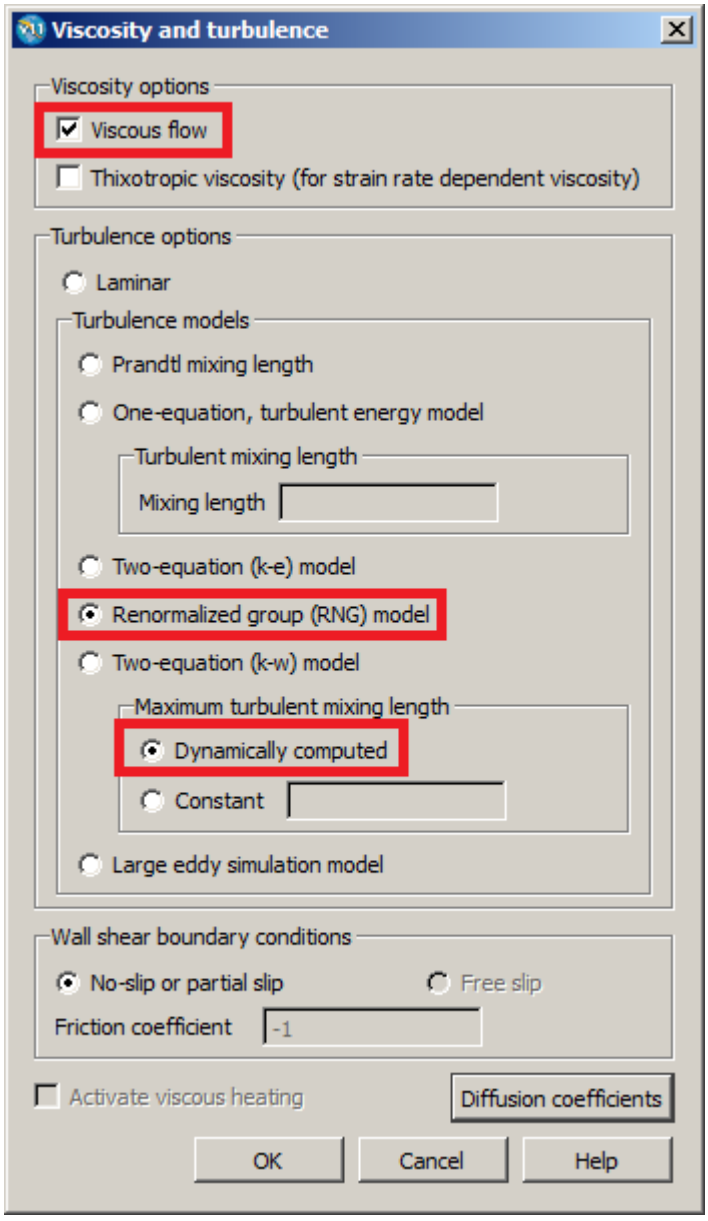
A typical estimate of the actual turbulent length scale is in the range of 7% to 18% of either the flow depth, wake width (abutments), or wake half-width (piers/jets).

We do not know the location of the greatest turbulence: is it the pipe outfall or the weir jet nappe impingement? We can guess that the impingement point is the most turbulent region in the domain and also controls significant flow features. Therefore, the thickness of the nappe could be used to estimate the maximum turbulent length scale, but perhaps the most turbulent region is the pipe outfall after all! Since we do not know enough to confidently specify the maximum mixing length, we will ask ***FLOW-3D*** to determine the maximum turbulent length scale for us.

1. On the **Model Setup > Physics** tab, select the **Viscosity and Turbulence** button.

2. Select the options shown below.

3. Click **OK** to accept the changes and **Save** your work.



Load Fluid Properties from the Fluids Database

Thus far, we have not specified which fluid we want to model. In order to run the simulation and check our boundary conditions for accuracy, we will need to set fluid properties. ***FLOW-3D*** includes databases of solid and fluid properties for a number of materials.

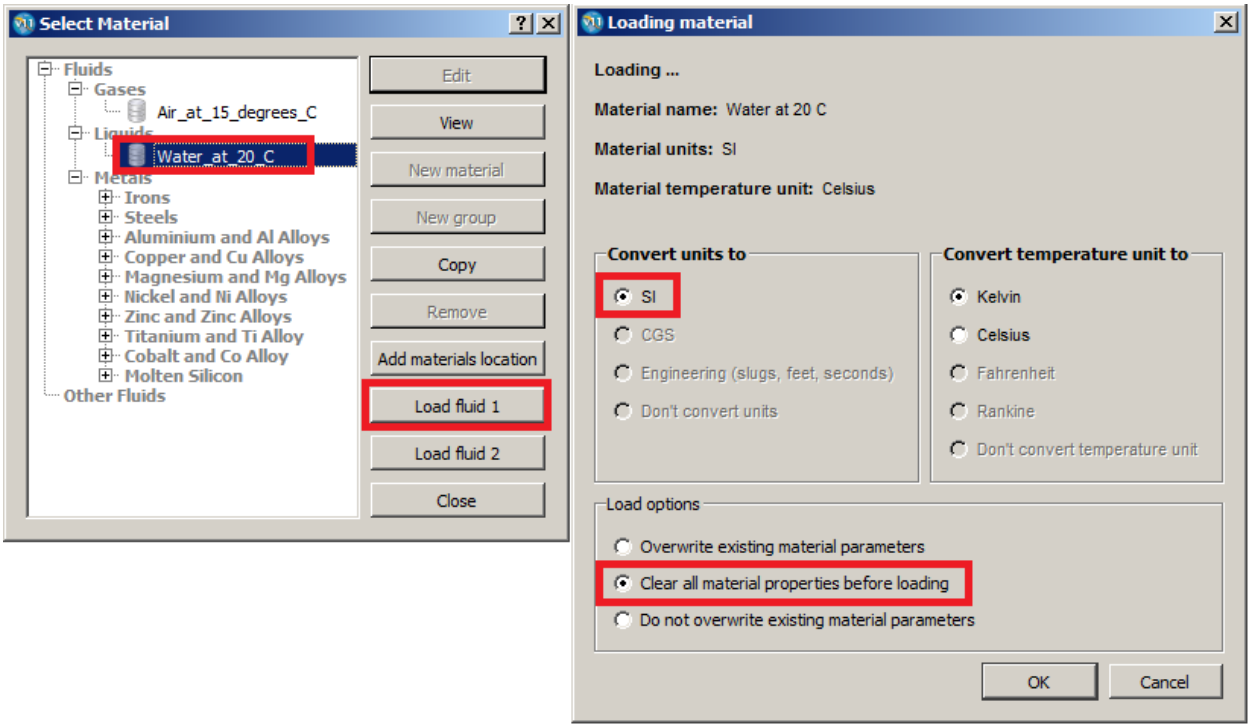
1. Select the **Model Setup > Fluids** tab.

2. Select **Materials > Fluids Database...** from the menus at the very top of the interface.

3. Choose **Water_at_20_C**, and select **Load Fluid 1**.

4. You have already set the unit system, so the only units available are **SI**.

5. Select **OK**, then **OK** again, then **Close** the database. **Save** again.



Edit Fluid Parameters and Add Limited Compressibility

1. Expand the **Properties > Fluid 1** tree and the sub-branches.

2. Note that fluid properties are grayed out if they require inactive physics models (e.g., thermal properties).

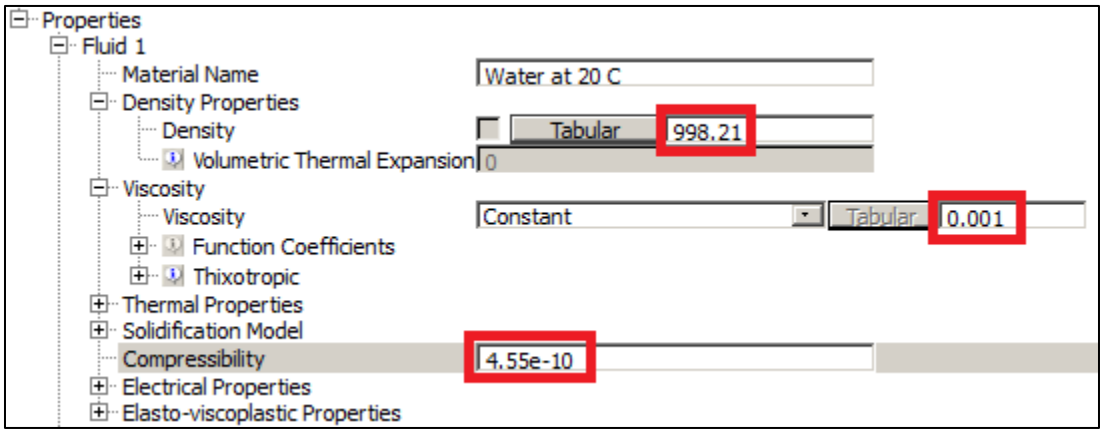
3. Check that the values for fluid **Density** ρ_f = **1000** kg/m^3 and dynamic **Viscosity** μ_f = **0.001** kg/m/s.

4. The values are approximate. Change **Density = 998.21** kg/m^3. This is true for clear water at 20°C and standard pressure (101,325 Pa).

5. Locate **Compressibility** and set it to **4.55e-10** Pa^{-1}. This is so-called *limited compressibility*. It is used in **Incompressible** flow simulations (like this one) to allow the cell pressure to vary as if density was changing due to compression. Volume stays constant. This makes it easier for the solver to converge (faster runtime), and allows pressure waves (e.g., water hammer) to propagate. The value entered is correct for clear water at 20°C and standard pressure. For your

future reference, the equation is: **Limited Compressibility RCSQL = 1/K = 1/(ρ_f W^2)**. **K** is the fluid bulk modulus, **ρ_f** is the fluid density, and **W** is the adiabatic speed of sound.

6. **Save** your simulation!



Initialize Fluid Regions & Hydrostatic Pressure

If you run your simulation right now it will take a long time to reach steady-state because it will fill the empty domain before topping the weir. You should add initial fluid to the simulation to speed up the time to steady state. The upstream and downstream initial depths should match the *free surface elevation* at the boundaries. You will also specify a hydrostatic pressure distribution for fluid in the domain at t = 0 and at the pressure boundaries. Hydrostatic pressure is specified as an initial condition, but it *also applies at all vertical pressure boundaries for the entire run*.

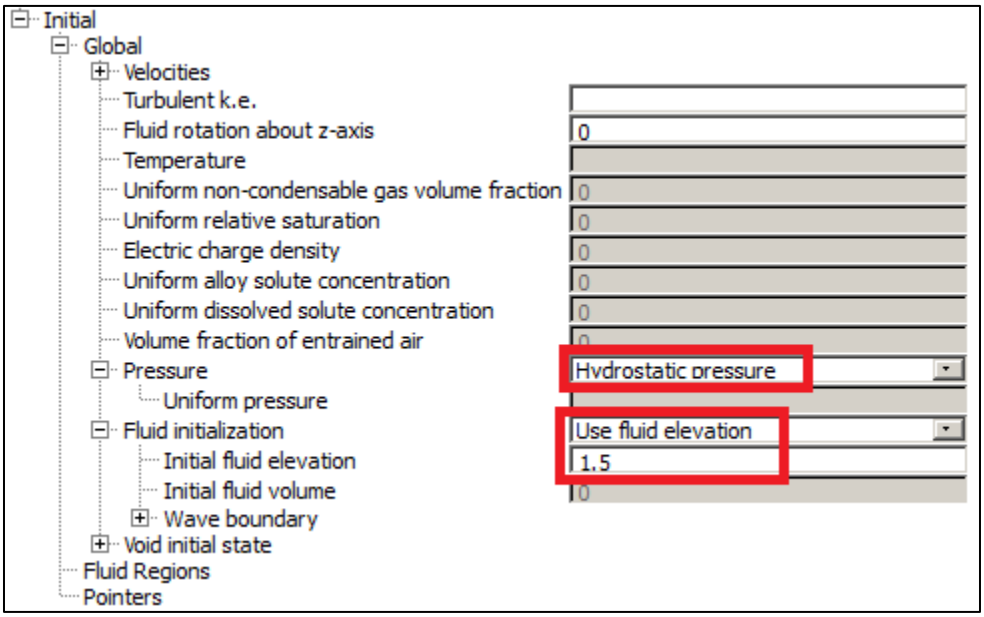
1. Navigate to the **Model Setup > Meshing & Geometry** tab.

2. Open the **Initial Conditions Window** 

and then the **Initial > Global > Pressure** branches.

3. Change the **Pressure** from **Uniform Pressure** to **Hydrostatic Pressure**. The surface pressure will be that of the initial void pressure, which defaults to zero, and will be maintained by the Z-max pressure boundary (where F = 0). The **boundary pressures** and **initial void pressures** match at 0 Pa. Matching the void, initial, and boundary pressures can be very important when the void pressure varies via bubble physics.

4. Open the **Global > Fluid Initialization** branch and set **Fluid Initialization = Use Fluid Elevation** and **Initial Fluid Elevation = 1.5** m. This will fill the entire domain to the level of the downstream boundary condition.



5. Close the **Global** tree branch, then *right-click* **Fluid Regions** and select **Add a Fluid Region**.

6. Name the new fluid region **Upstream**. Fluid regions overwrite global areas that they overlap.

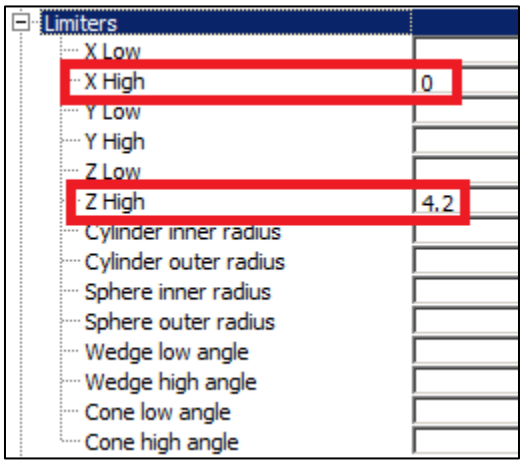
7. Note that the region will **Add Fluid**, and that the **Fluid Fraction = 1** (100% Fluid 1 = water).

8. Open the **Limiters** branch of the tree fluid region tree.

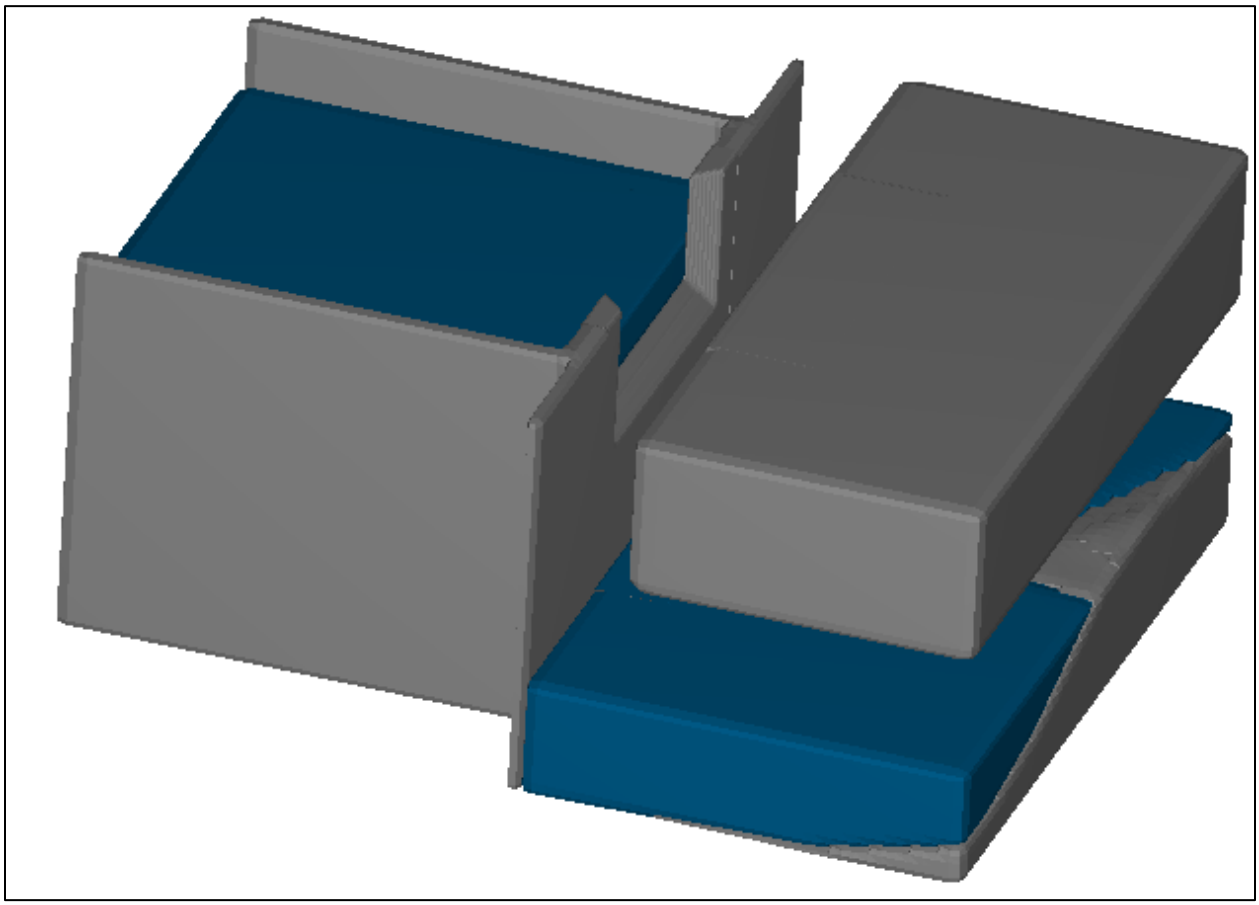
   a. Specify **X High = 0** m to fill the upstream channel to the weir.
   b. Specify **Z High = 4.2** (to match the upstream boundary condition free surface elevation).
   c. The *unspecified* limits default to the *domain extents*.

9. Note that initial **Velocities**, **Pressure**, and **Turbulent Kinetic Energy** can also be specified, but do not need to be here.

10. **Save** your work.



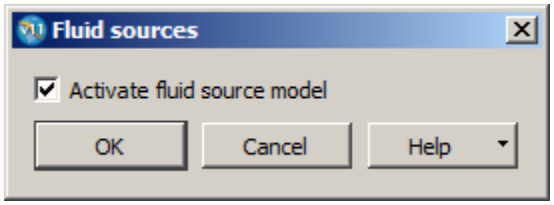
11. View the **FAVORized** geometry using the computational mesh. You should already know how to do this from previous exercises. This time select **Fluid Surface = Fluid 1**.



Add Mass/Momentum Sources to the Pipes

Mass/momentum sources and sinks are flat planes that add or remove fluid. They are completely porous to flow approaching from behind them, and are used to represent nozzles, pipes, intakes, vents, and other devices. Mass/momentum sources are ideal when fluid must be added somewhere inside the domain. They can also be used to remove fluid if they are placed so they are always underwater.

1. Navigate to the **Model Setup > Physics** tab, select **Fluid Sources**, and check the box to activate the physics. Select **OK**.

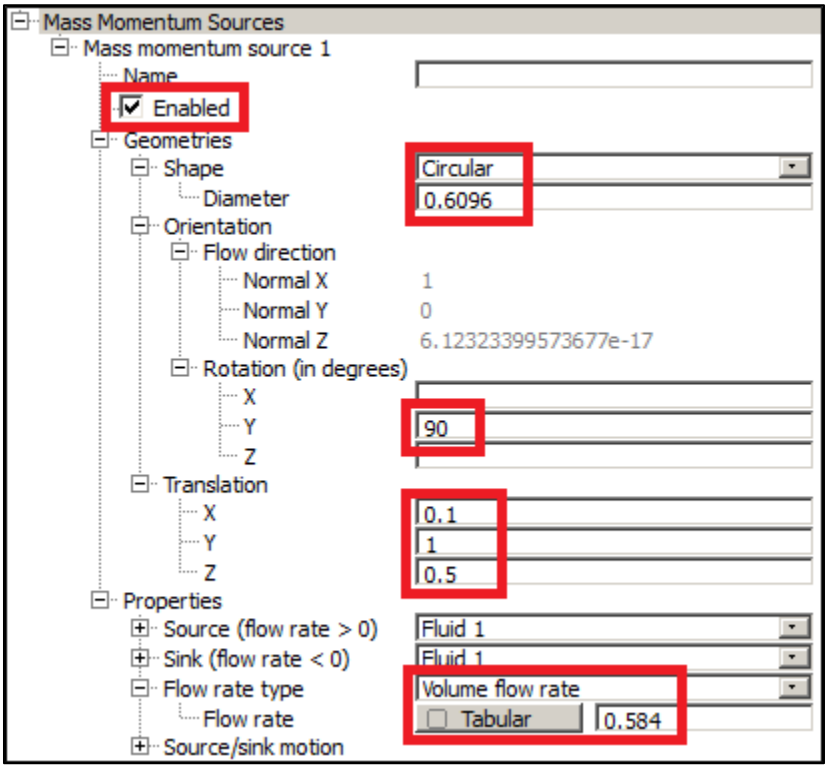


2. Go back to the **Model Setup > Meshing & Geometry** tab and open the **Mass/Momentum Source Window**  .

3. Right-click **Mass Momentum Sources** and select **Add a Mass Momentum Source**.

4. Click to check the **Enabled** box to activate the first source.

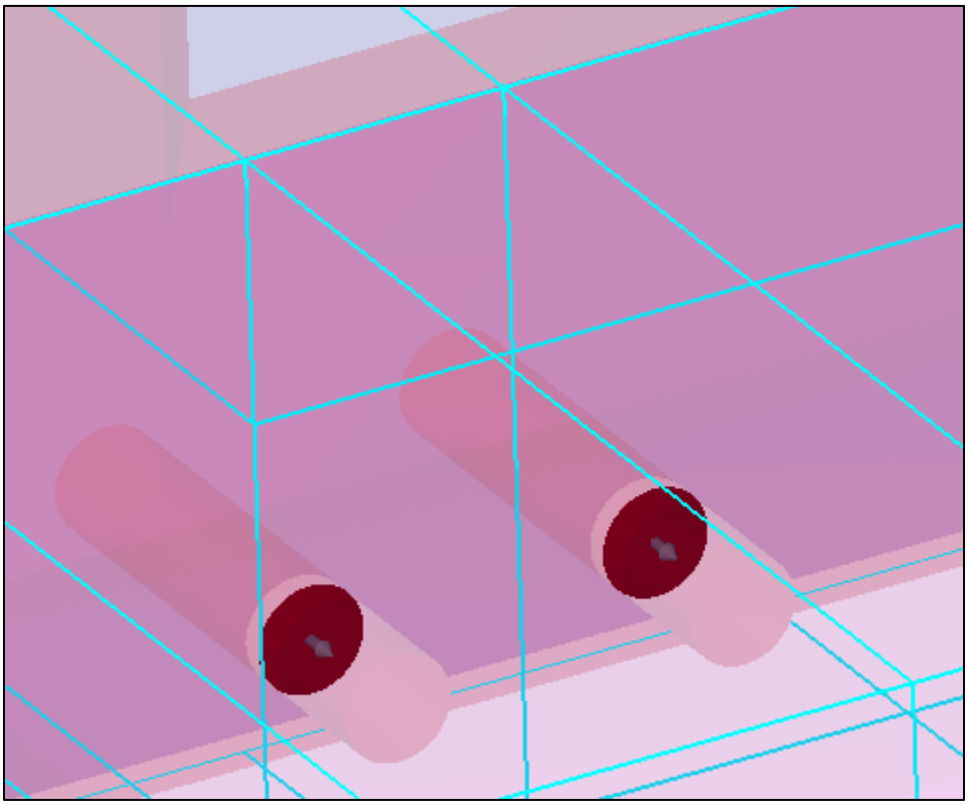
5. Enter the values shown below. The **Volumetric Flow Rate** gives 2 m/s velocity: $Q_V = U \times A$.



1. Adjust **Transparency** so you can see the circular source at the origin.

2. *Right-click* **Mass Momentum Source 1** and select **Copy**. In the new source, change only **Geometries > Translation > Y = -1**.

3. Check that your model appears as shown, with flow vector pointing outward, and **Save**.



Add a Flux Surface Baffle to Measure Weir Flow

Baffles are flat planes along grid lines that block or measure flow. They can be solid or porous. **Flux Surfaces** are a special kind of baffle which measure flow parameters and can be completely open to flow (100% porous).

1. Still on the **Meshing & Geometry** tab, open the **Baffles window** and select **Add**.

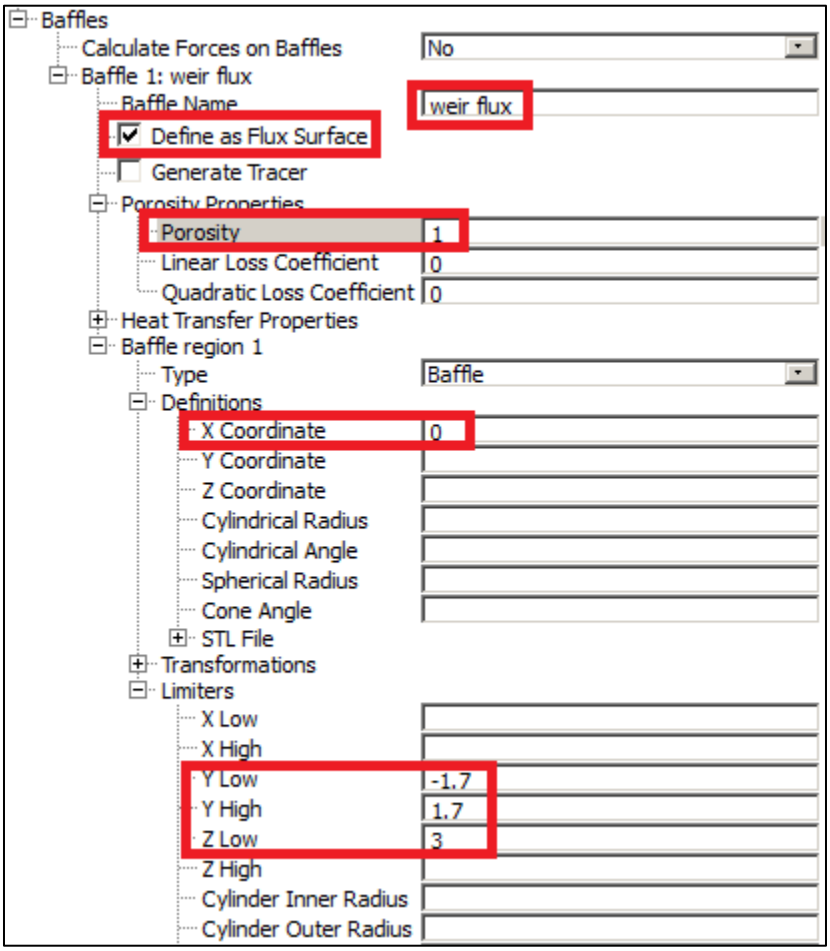
2. Name the new baffle **weir flux**. Output will be titled with this name.

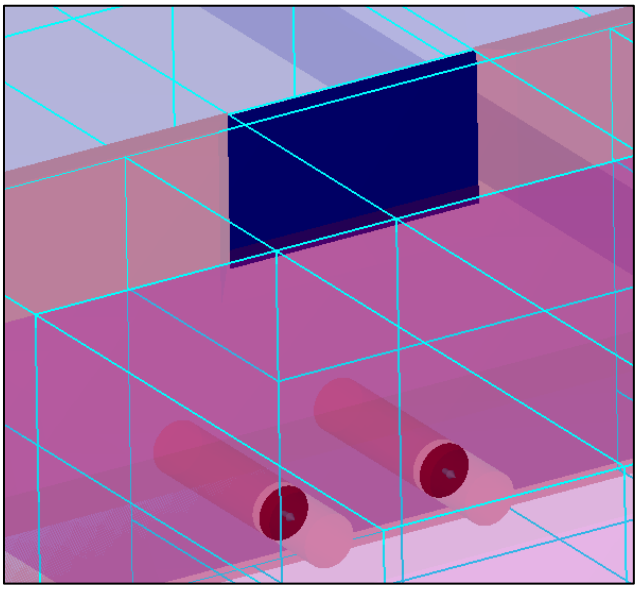
3. Check the **Define as Flux Surface** option. Now the baffle will measure flow rates, depth, and hydraulic energy and head.

4. Define the baffle location and parameters:

   a. Open **Baffle Region 1 > Definitions**. Only one definition should be specified per baffle.
   b. Specify **X Coordinate = 0**. The baffle now defines the plane at x = 0.
   c. Open **Baffle Region 1 > Limiters**. These are used to bound the defined plane.
   d. Specify **Y Low = -1.7**, **Y High = 1.7**, and **Z Low = 3**. This makes the baffle slightly larger than the opening, which may be important when the baffle moves to the nearest grid lines.
   e. Open **Porosity Properties** and set **Porosity = 1**. This is not strictly required, since flux surfaces cannot be completely solid, and will default to porosity = 1 if the default 0 is left.

5. **Save** your work.

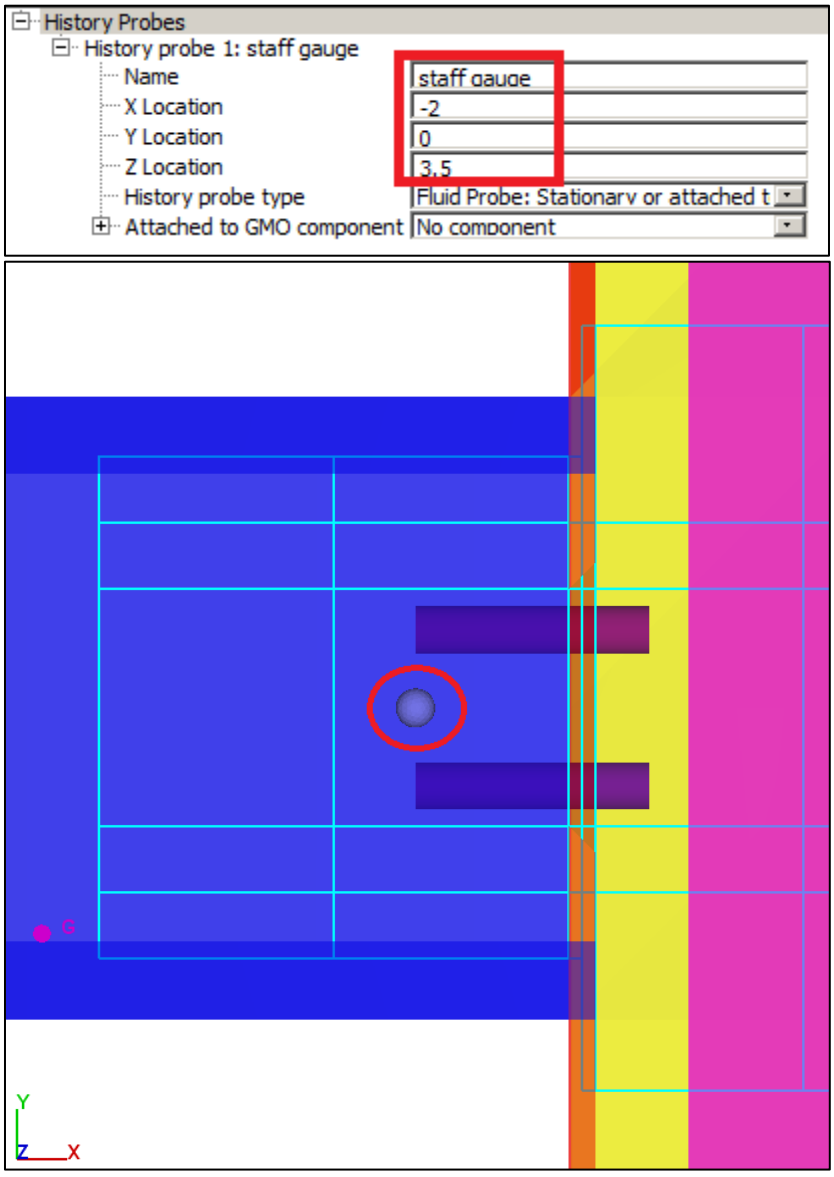

Baffles
Calculate Forces on Baffles
No
Baffle 1: weir flux
Baffle Name
weir flux
Define as Flux Surface
Generate Tracer
Porosity Properties
Porosity
1
Linear Loss Coefficient
0
Quadratic Loss Coefficient
0
Heat Transfer Properties
Baffle region 1
Type
Baffle
Definitions
X Coordinate
0
Y Coordinate
Z Coordinate
Cylindrical Radius
Cylindrical Angle
Spherical Radius
Cone Angle
STL File
Transformations
Limiters
X Low
X High
Y Low
-1.7
Y High
1.7
Z Low
3
Z High
Cylinder Inner Radius
Cylinder Outer Radius



Add a History Probe to Measure the Approach Depth

History Probes are single points which measure the cell-centered flow parameters. They can be made to move with solid objects or float freely in the flow. Here you will create a fixed point probe that will act as a staff gauge for measuring the flow depth. Like **flux surfaces**, the **history probe** output is labeled with it's name.

1. Still on the **Meshing & Geometry** tab, open the **History Probes Window**  .
2. Select **Add**  and enter the values shown below.
3. **Save** again.



Set the Finish Time

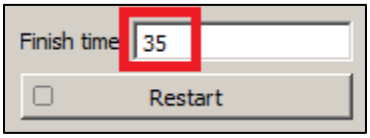
The default runtime for a new simulation is 1 second, and this is not long enough to reach steady state.

Rule of Thumb
A conservative measure of the time for a free-surface fluid system to reach steady state is 10 to 20 times that required for a fluid packet to pass through the system. A characteristic velocity in the domain needs to be estimated to determine the characteristic (packet-transit) time scale.

Without experimental or simulation data we can only estimate the characteristic velocity. The velocity of the water falling under gravity though a distance H of 2.7 m (the drop height) is $V = (2\ g\ H)^{1/2}$ = (2*9.81 m/s^2*2.7 $m)^{1/2}$ = 7.3 m/s. The total length of the mesh domain is about 12 m, so the packet-transit time is approximately 12/7.3 ≈ 1.6 seconds. Real-world time to steady state is then about 16 to 32 seconds. Model time to steady state also depends on the mesh resolution: for example, meshes that are too coarse or have very poor aspect ratios may not converge to steady state at all.

1. Navigate to the **Model Setup > General** tab.

2. In the upper left corner, enter **Finish Time = 35** seconds.



Set the Number of Processors

It's good practice to leave at least one processor available for the OS and other programs to use.

1. Specify **Number of Processors = 4**.

2. **Save** your work!

Select Output Options

The solver will automatically write 10 data sets (e.g., every 10 seconds) that can be used to generate 2-D and 3-D plots. These data sets are called **Restart data** because they include all of the information necessary to describe the flow field and can be used as the initial and/or boundary condition configuration of a new simulation called a **Restart simulation**. More frequent output can be added as **Selected data**; the default frequency adds 100 data sets (e.g., every 1 seconds) containing only the variables you select. These **Selected Data** sets are written in addition to the **Restart data**.

1. Navigate to the **Model Setup > Output** tab.

2. Find **Additional Output** on the left. Selections made here will be added to *Restart Data* and *History Data* (time-dependent data recorded for the *flux surface* and *history probe*).

3. Check the box for **Hydraulic Data** under **Additional Output**. The parameter is also added to the **Selected Data** list in the center of the screen.

4. Under **Selected Data**, check the boxes for:

    a. **Fluid fraction**,
    b. **Fluid velocities**,
    c. **Hydraulic Data**, and
    d. **Pressure**

5. Note that the **Restart Data Interval**, **Selected Data Interval**, and **History Data Interval** can be forced to other than the default $1/10^{th}$, $1/100^{th}$, amd $1/100^{th}$ of the finish time, respectively. Intervals are specified in seconds and can usually be left to the default.

6. **Save** your work!

Set Higher-Order Momentum Advection

The default momentum advection is quite robust and usually very accurate for linear flow. It tends to diffuse momentum (and kinetic energy) when the flow is curving. There will likely be some flow curvature near the weir corners, so although it may not help (and will definitely slow down the run) we will add a higher order momentum advection method.

1. Go to the **Model Setup > Numerics** tab. Locate the Momentum Advection options on the right-hand side.

2. Select **Second Order Monotonicity-Preserving** momentum advection. This option is 2^{nd}-order accurate in space and 1^{st}-order accurate in time, and is stable in cells that have a free surface.

3. **Save** your work! Have you figured out that this should happen with every change?

Run the Simulation

You already pre-processed the simulation in Exercise 3. A careful modeler would preprocess the simulation again now, but throw caution to the wind and run the model anyway!

1. Select **Simulate > Run Simulation > Local** from the menu at the top.

2. The run will usually take between 8 and 10 hours, depending on your computer's power.

Your instructor is available to answer any questions you'd like to discuss!

Exercise 5: Post-Processing & Analysis

Hydraulics Exercise 5: Analysis

Problem Description & Objectives

The results from the simulation you built in the previous four exercises will be analyzed. This exercise applies and expands the techniques introduced in Exercise 1.

FLOW-3D Learning Objectives

In this exercise you will learn techniques to:

1. Use **3-D** plots to check the results qualitatively.
2. Review the **Runtime Plots** to check for steady state.
3. Review **Solver Output text** and **files** to check for convective flux errors.
4. Review the meaning and significance of convective flux errors.
5. Use **2-D** plots to visualize the location of the minimum stability criteria.
6. Use **Probe** plots to quantify flow rate and depths.
7. Discuss model comparisons and possible sources of error.

Load the Results File

The starting point for this exercise is the output generated at the end of Exercise 4. The *flsgrf* file is 4.9 GB, so it is not offered for download or copy. If you have not completed Exercise 4 yet, please ask your instructor for a copy of the results file.

1. Launch ***FLOW-3D®***.
2. Select the **Simulation Manager** tab.
3. Click to highlight the **Hydraulics Examples > Weir Finish** simulation that you ran.
4. Select the **Analyze** tab.
5. If prompted to select a file, continue with step 7 below.
6. If not, select the **Open Results File** button at the bottom left of the **Analyze** tab.
7. Select **Custom** to display *flsgrf* and *prpgrf* files.
8. Select ***flsgrf.Weir_Finish***.
9. Select **OK**. The file loads and the GUI switches to the **Analyze > 3-D** tab with all options reset.

Select 3-D Plotting Options

1. Go to the **3-D** sub-tab.

2. Select **Data Source = Selected**.

3. Check that **Iso-Surface = Fraction of Fluid** so you plot the water/air and water/solid interfaces.

4. Select **Color Variable = Velocity Magnitude**.

5. Click **Additional Variables** below the **Color Variable** selection.

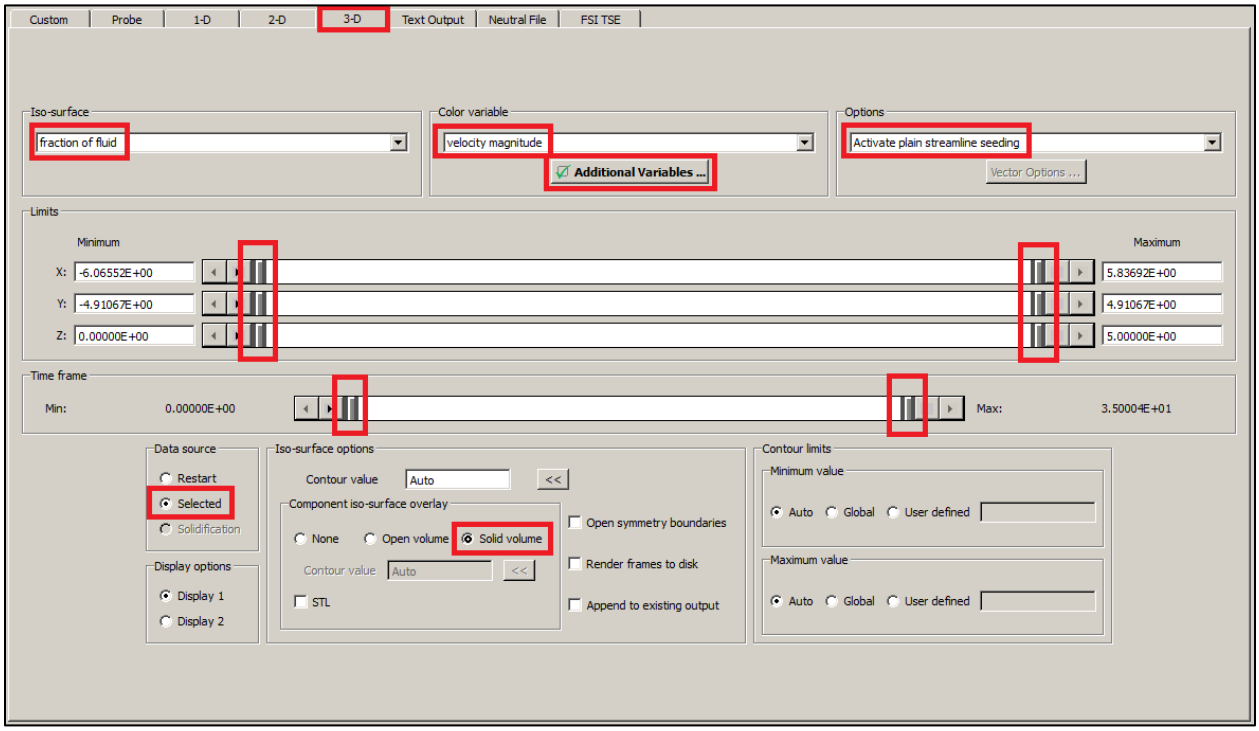
6. Select **Flow Depth, Free Surface Elevation**, **Depth-Averaged Velocity**, and **Froude Number**, and click **OK** to accept the variables for plotting.

7. Open the **Time Frame sliders** all the way: **Min = 0** and **Max = 35** seconds.

8. Open the **X, Y, and Z Limits sliders** all the way.

9. Select **Component Iso-Surface Overlay = Solid Volume**. This plots the solids as well.

10. Select **Options = Activate Plain Streamline Seeding**.



View and Interpret the 3-D Results

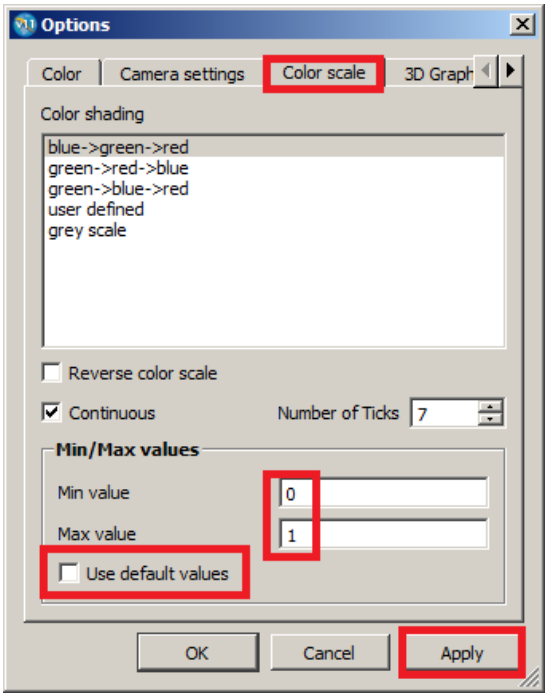
1. Click the **Render** button at the lower right. The display switches automatically to the **Display** tab, and a new *Streamline Edit* dialog appears. Find a good setting for the **Transparency** slider.

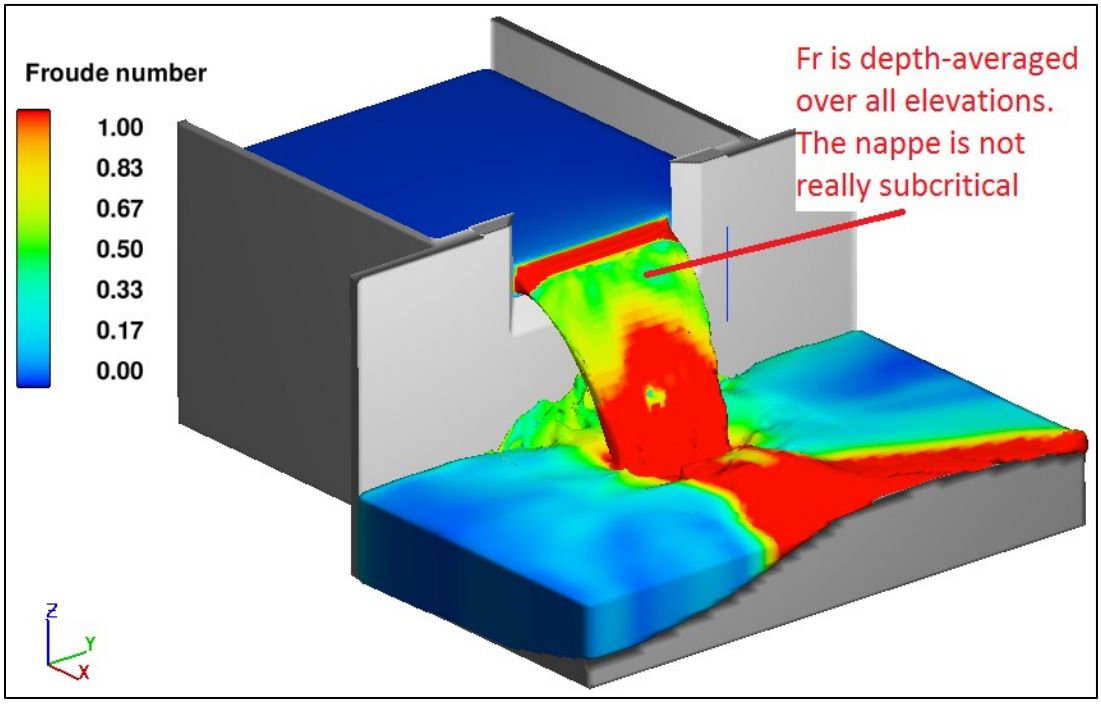
2. Use the **Next** and **Previous** buttons, and/or *double-click* on the **Available Time Frames** listed, to see the results at different times. Note that the baffle is displayed as a *gray solid plane*, even though it's completely porous. **Hide** it on the **View menu > Baffle Options** dialog.

3. Adjust the **Transparency** slider. Observe that the colored results are only the values at the surface; changing transparency does not allow you to see the velocity profile inside the flow. **2-D** plots or 3-D **Mesh Slices** can be used to visualize internal (submerged) flow properties.

4. Select the other **Color Variables** from the dropdown in the upper left corner of the interface. Do the results make sense?

5. Set **Color Variable = Froude Number**, and open Tools > Options from the menu above the display window. Uncheck the default values option and enter the values shown below. Note that the nappe appears subcritical. This is because *Hydraulic Data* is *depth-averaged along z*, so the Froude number includes the depth and velocity of the pool beneath the nappe.





6. Re-activate the **Default Values** option and click **Apply** to return to the normal color scale.

Note how a supercritical jet expands away from the wake, while the rest of the pool is subcritical. This makes the appropriateness of the downstream boundary unclear. Pressure boundaries are best for subcritical tailwater, while Outflow boundaries are best for supercritical flow. The X-max and Y-max boundaries should be placed farther from the nappe to avoid this difficulty.

Also note the contraction of the jet as it falls; the degree of contraction in the CFD model depends on the momentum advection method, volume-of-fluid (VOF) method for free-surface tracking, and the geometric resolution of the weir crest.

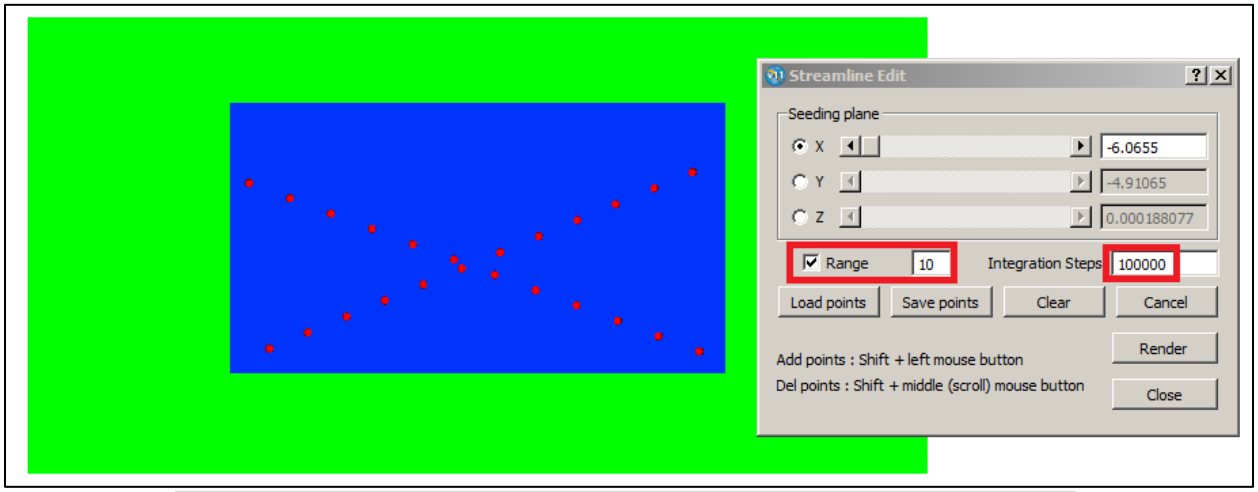
Finally, note that the upstream depth (as shown for **Fluid Depth** and **Free Surface Elevation**) does not vary much. This indicates very minor head loss due to friction. The degree of friction appearing in the model depends on the water velocity, the roughness height of the component, and the mesh resolution near the bed.

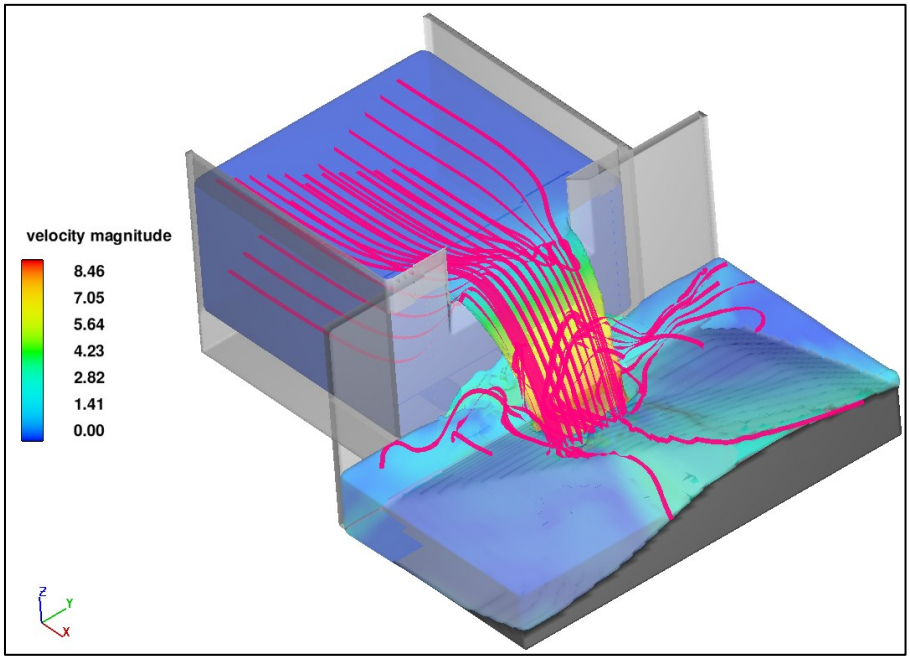
Add Streamlines

1. Access the **Streamline Edit** dialog  or select **Tools > Streamline Edit**. If the dialog doesn't appear, return to the **Analyze** tab, activate **Plain Streamline Seeding**, and **Render** again.

2. Set a single streamline starting point and edit it's appearance:
   a. Select the **X** direction radio button for the **Seeding Plane**.
   b. Slide the **Seeding Plane** slider back and forth and watch the green plane move in the display.
   c. Adjust **Transparency** so you can see the entire plane.
   d. Hold **Shift** and *left-click* on the plane at a point near the bed and approximately on the centerline of the flow.
   e. Select **Render**.
   f. A line should appear showing the streamline from t = 0 to the selected time in the **Available Time Frames** list.
   g. If no line appears, troubleshoot by:
      i. adjusting transparency,
      ii. selecting a later time frame (for a time when fluid is moving),
      iii. making sure that the point you added isn't inside the bed (no flow there).
      iv. Use **Clear** and try again, if necessary. Remember to **Render** again after adding a point.
   h. Increase the number of **Integration Steps** by a factor of 10 (add a zero) and **Render** again. The number of steps controls the resolution of the line, as well as the length. Usually 10,000 or 100,000 integration steps are sufficient to complete the line. Repeat the increase until the line reaches the end or no longer changes. If the line terminates at a free surface, try a different starting location.
   i. Locate the **Object List** pane in the lower left. *Right-click* the **Streamlines** entity there to get display options.
   j. Try changing the display shape to **Ribbons**, adjusting the **transparency**, and changing the **color** and weight (**ribbon scale percentage**) of the streamline. Also try adjusting the transparency of the other entities (solid and fluid surfaces),

3. Make *many* streamlines:
   a. Select **Clear**. The seeding points are erased, but the lines remain.
   b. Move the slider so that it is located at the minimum **X** value **-6.0655** m.
   c. Activate the **Range** checkbox and set the number of points to **10**.
   d. Hold *Shift* and *click* on the seeding plane in a corner near the bed, and then in the opposite corner near the surface. A line of 10 points appears between the points you clicked, for a total of 12.
   e. Repeat the *Shift + click* process for the other two corners, to make an X with 24 points.
   f. **Render** the streamlines and adjust the display (global and streamline-entity) as before. You may need to add more integration steps (100,000 shown below) to complete the lines.
   g. Select **Clear** to remove the seed points and **Close** to remove the plane. The lines remain.
   h. Step through the **Available Time Frames** and observe the early oscillations of the streamlines and how they eventually become steady.
4. If you want to change the streamlines, select **Tools > Streamline Edit** from the menu above the display to reopen the dialog and seeding plane.





Review the Runtime Plots

1. Select the **Simulation Manager** tab.

2. Select the following plots from the list on the left of the screen:

   a. **Stability Limit & Time Step Size**: the time step size is very close to the stability limit, matching it in most cases. There are two types of solvers: implicit and explicit. Implicit solvers are iterative, while explicit solvers are direct (one step). The stability limit is the smallest time step required for any explicit solver (e.g., convection). The time step *dt* may be less than the stability limit when an implicit solver (like pressure) does not converge to the error tolerance: then it will restart the calculation with a smaller *dt*. The only implicit solver in our model is the pressure solver, so this suggests that the pressure solver is working well.
   b. **Time Step Size**: Repeats part of the information from the last plot. Move the mouse over the plot to find that the time step size oscillates about **0.005** seconds per step.
   c. **Epsi & Maximum Pressure Residual**: The maximum residual is always below *epsi*, so the pressure solver converged to a solution in every time step. See Exercise 1 for more explanation.
   d. **Pressure Iteration Count**: the default pressure solver is the Generalized Minimum Residual method (GMRES), which should usually operate between 1 and 10 iterations per cycle.
   e. **Fill Fraction**: the simulation is about **53.5%** full of fluid for the second half of the simulation. The relatively flat shape of this plot suggests that fluid volume in the simulation is near-constant, and therefore outflow = inflow. This is one criterion for steady-state.
   f. **Conv**. **Volume Error (% Lost)**: the total volume lost due to numerical mass calculation error is less than **0.1%**. This is acceptable. Typically less than 1% to 3% loss is acceptable.
   g. **Interblock Boundary Volume Error (% Lost)**: the total volume lost due to interpolation at inter-block boundaries is about **2.5%**, and follows the same criteria above. The two values should be considered additive. 2.5% is on the high side. It suggests that one of the nested blocks has its boundary in a region of high gradients. *Later you will locate where this occurs*.
   h. **Volume of Fluid 1**: the flow volume appears to reach quasi-steady state by about 60 seconds, with an average near **163** m^3 in the simulation.
   i. **Fluid 1 Surface Area**: the water surface area comes to quasi-steady state around 10 seconds.
   j. **Mass-Averaged Kinetic Energy**: MKE here is the sum of total kinetic energy in all cells divided by the mass of fluid in the simulation. It reaches quasi-steady state around 20 seconds.
   k. **Mass-Averaged Turbulent Energy**: This integral is the sum of time-mean TKE in all cells, weighted in each cell by the fluid mass. TKE appears to be quasi-steady by about 10 seconds, but there are too many oscillations to be certain. You could launch a Restart Simulation from the end of this one to see if TKE levels off later, or maybe it's just because of the splashing on the topography under the nappe.
   l. **Particle Count**: The number of particles in the simulation stays constant at 1,330 particles. These particles are used to define the **Mass/Momentum** sources in the pipes (as point fluid sources) and do not move with the flow.

Use Solver Text Output to Diagnose Errors

1. Locate the **Warnings & Errors** button  above the runtime plots. It means warnings are present.

2. After the meshblock mismatch errors discussed in There is one warning, which is displayed below.

   Solver Messages:
   convective flux exceeded stability limit
   at t= 3.0009E-02 cycle= 1 iter= 51 delt= 3.0009E-02 mesh block 4
   restarting cycle with smaller time step
   maximum failure ratio = 2.26944E+00 is in x-coordinate direction
   at cell (82, 34, 17) mesh block 4

3. The meaning is as follows:
   a. At **t = 0.03** seconds, a packet of fluid tried to move more than one cell at a time. The packet probably accelerated unexpectedly, which is why the solver didn't predict the velocity of the packet. The *stability limit* is the requirement that fluid packets only move one cell per time step. The solver takes care of the problem by restarting the time step with a smaller *dt*.
   b. The error occurred in **Mesh Block 4**, the conforming block.
   c. The offending cell number is **i, j, k = 82, 34, 17**. Remember that **i,j,k** values are block-specific, while **x,y,z** values are global.
   d. The problematic fluid packet was moving in the x-direction. In the next step you will find the problem cell and identify why the problem occured.

4. Use the small **X** in the upper right corner of the dialog to close the **Warnings & Errors**.

5. Scroll up through the text output displayed below the plots and find the same warning. Notice that the cell location and mesh block are not given in this summary.

6. Select **Diagnostics > Solver Errors** from the menu at the top of the screen: the same information is displayed. Some errors report *additional* information to the **Solver Errors** file, so it should always be checked.

The only run-time error reported in this simulation is *a single convective flux error* which does not affect solver accuracy. Except for the purposes of this exercise, you can safely ignore isolated errors. If convective flux errors recur frequently, they will at best slow down the simulation, and at worst cause it to terminate before completion. Errors should be addressed when they are repetitive.

Use 2-D Plots to Diagnose Errors

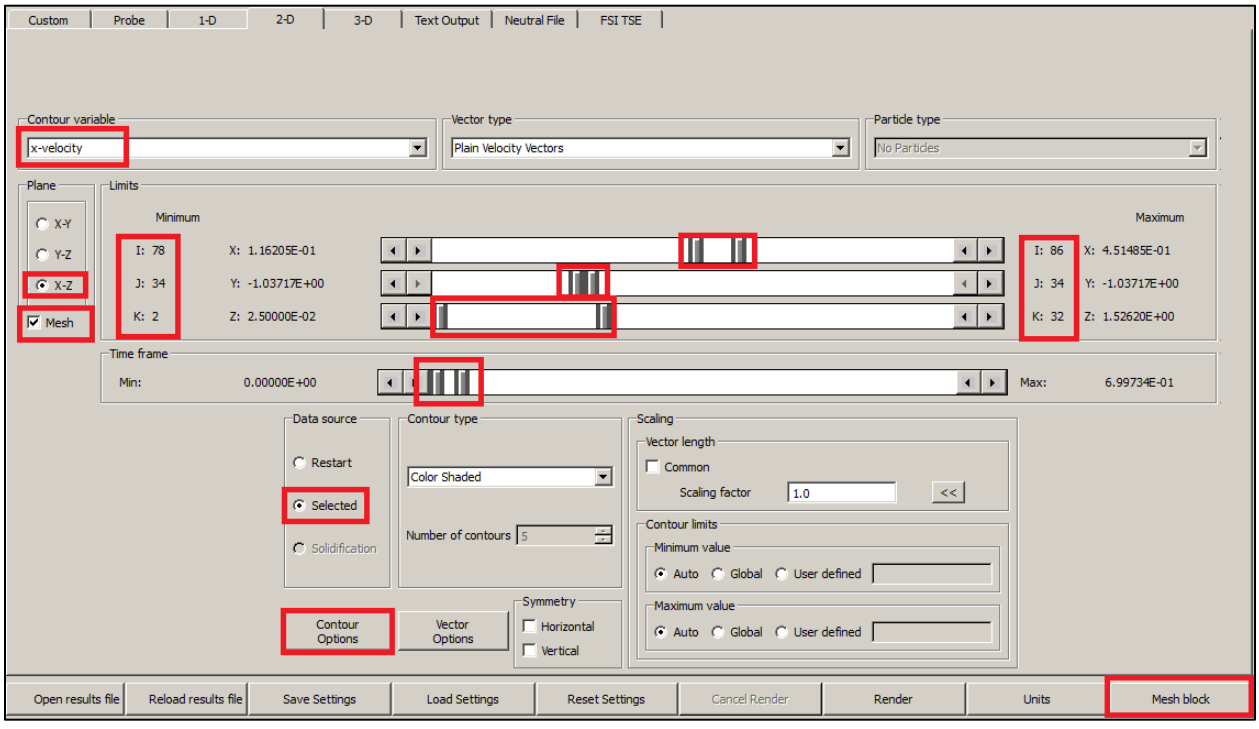
1. Go to the **Analyze > 2-D** tab.

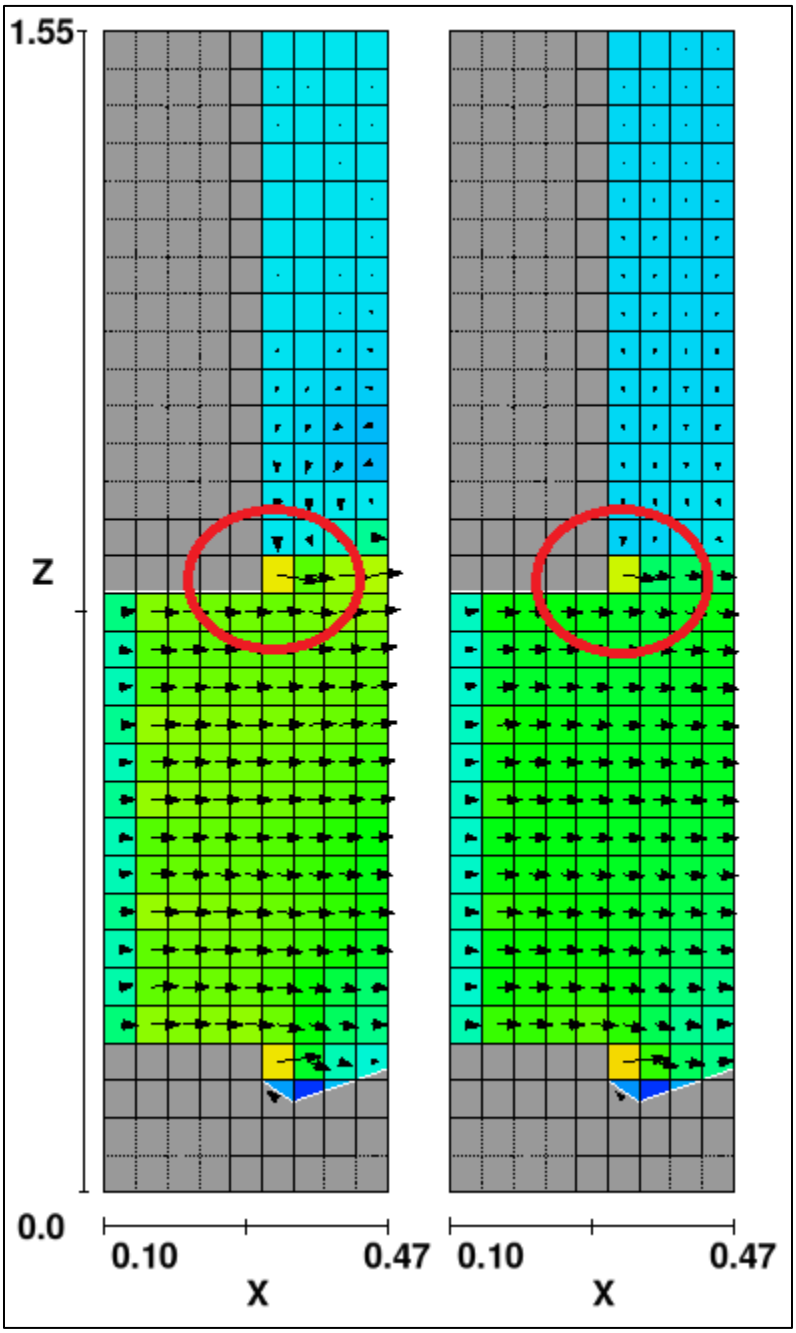
2. Click the **Mesh Block** button at the lower right and select **Mesh Block 4** only. When you click **OK**, the **Limits** sliders will reset to show the **I J K** cell number in the selected block at their sides.

3. Set plotting extents for Block 4:
   a. Set **Contour Variable = X-Velocity** and **Vector Type = Plain Velocity Vectors**.
   b. Select **Plane = XZ**.
   c. Select the **Mesh** checkbox to plot the grid lines.
   d. Select **Data Source = Selected**.
   e. Slide the **Max Time Frame** slider (or use the associated arrow) so that **t = 0 to 0.7** seconds.
   f. Slide the **X Limits** sliders so that **I = 78 to 86**. The problem cell (I = 61) will be in the center.
   g. Move the **Y Limits** sliders so that both **Min** and **Max J = 34**. This will display one planar slice along the cell centers of Block 4. The slice will contain the problem cell.
   h. Slide the **Z Limits** sliders so that **K = 2 to 32**.

4. Select contour plotting options:
   a. Click the **Contour Options** button in the bottom central portion of the tab.
   b. Activate the **No Contour Smoothing** checkbox to turn off the aesthetic blurring of cell-centered values.
   c. Select **Blanking Variable = None**. This clears the **Blanking Variable Value** entry. All cell values will be displayed, whether they contain fluid or not.
   d. Select **OK** to close the dialog.

5. Check that you've made the selections above (marked below for reference) and click **Render**. The two plots closest in time to the error should be similar to the ones on the next page.





6. Locate the cell at I = 82, J = 34, K = 17:
   a. Select **multi** as the plot type and then select the time frames for **t = 0.35** and **t = 0.73** seconds from the list at the right. The problem occurred before these plots, but they still show why.
   b. Check that the text below the plot shows the range **ix = 78 to 86** and **kz = 2 to 32**.
   c. Start with the bottom-right cell at K = 2 and count up to 17. From the left, start at I = 78 and count up to 82. This is the cell specified by the warning message.
   d. Induce an explanation (or see the next page).

The problem cell is just inside the top of the pipe. Notice the recirculation zone to the right above the pipe discharge jet. Convective flux errors are associated with sudden accelerations. The flow is making a corner here, so it can be induced that the discharge, forming in quiet water, interacted with the pool and experienced an unexpected acceleration at t = 0.03 seconds. Even at t = 0.35 and 0.7 seconds you can still see where the recirculating pool flow accelerates from the downward direction to the horizontal direction as it's caught by the pipe discharge. By the visible time steps, however, the flow is predictable enough that the dynamically chosen time step accounts for the change in direction.

Identifying the problem cell is useful when there are many errors in a row. In *this case* the failure is of a type that can usually be ignored, especially since it occurs only once. Common causes of serious (that is, numerous) errors include:

- bad geometry files: fix with ***pyAdmesh*** and ***netfabb*** or re-export from CAD,
- unresolved or poorly resolved gaps between geometry: use finer cells or different geometry,
- incorrect boundaries or sudden changes in boundary conditions or mass/momentum sources,
- incorrect initial conditions, and/or
- flying droplets/splashing of fluid.

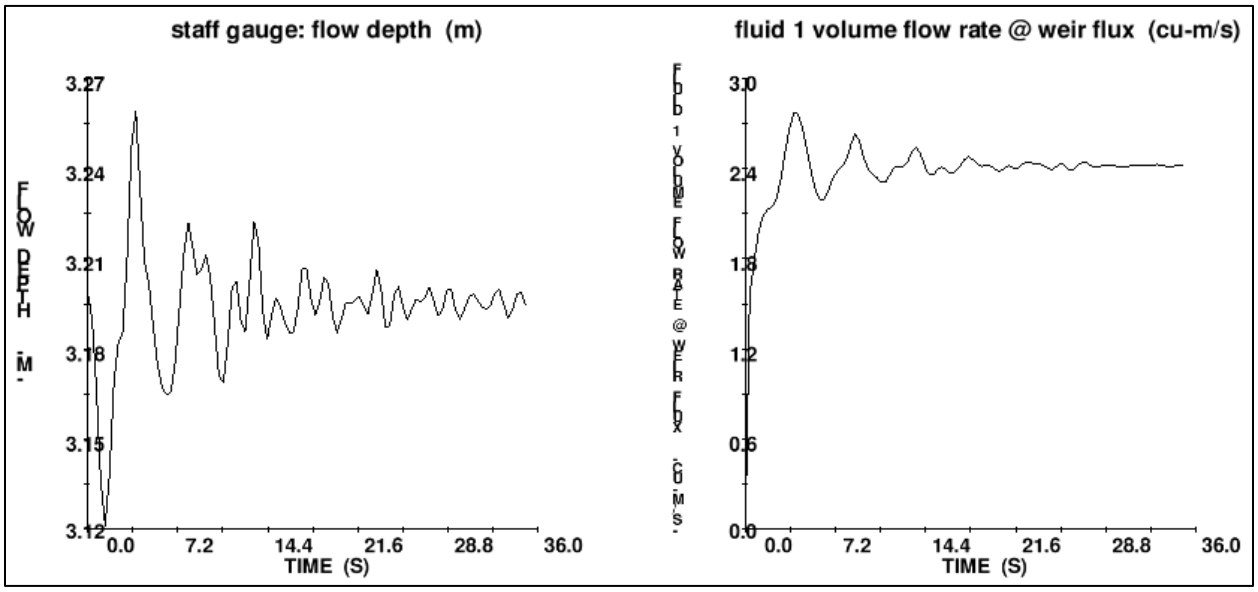
Use Probe > General History Data for Measurement Location Output

1. Go to the **Analyze > Probe** tab and select **Data Source = General History**.

2. Display units:
   a. Click **Units** at the bottom right of the screen.
   b. Activate **Show Units on Plots**.
   c. Select **Units System = SI**.
   d. Select **OK** to close the dialog.

3. Scroll down in the **Data Variables** list and select the following:
   a. **staff gauge: flow depth** (around #14). This is output from the **History Probe** you placed upstream.
   b. **fluid 1 volume flow rate @ weir flux** (around #59). This is output from the **Flux Surface** you put at the weir.

4. **Render** the plots graphically, using **multi** plots to look at the data side-by-side.



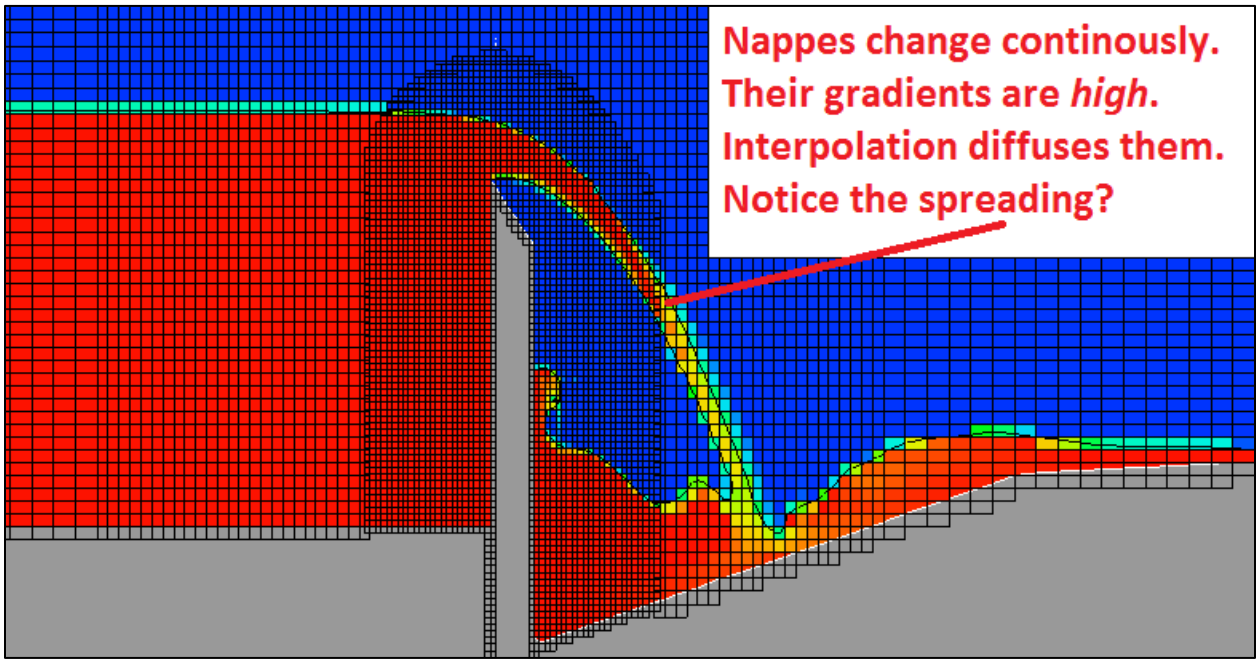
5. Get measurement station output as text:
   a. Return to the **Analyze > Probe** tab and select **Output Form = Text** (in the lower left).
   b. The options for **Integrate**, **Differentiate**, and **Moving Average** become inactive, as they are only available for graphical output.
   c. Check that the **Time Frame Min** and **Max** cover the entire duration of the run.
   d. Click **Render**.
   e. The text output is given with the variable titles: **TIME (s)**, **HYFD (m)**, and **FFLUX1 (m^3/s)**. It appears that the flow rate reaches approximate steady state around 20 seconds.
   f. The time-averaged flow rate can be found by copying and pasting the data into a spreadsheet. If you were to do this now, you would use a text import function to separate the space-delimited values and some method of averaging the later quasi-steady results.
   g. The average flow rate *Q* over the weir after 20 seconds is approximately **2.42 m^3/s**.
   h. The average upstream centerline depth h_1+P after 50 seconds is **4.20 m**.

6. Select **Continue** to close the text output.

View the Mesh and Find The Multi-Block Volume Error

Now you will practice what you know.

1. Make a 2-D plot with the following settings:
   a. Plot all 4 **mesh blocks**.
   b. Plot the flow's center line as an **XZ** plane.
   c. Plot **Fraction of Fluid**.
   d. Turn off **contour smoothing**.
   e. Turn off cell **blanking**. All cells should show values with no white space.
   f. **No Velocity Vectors**.
   g. Plot only the last **output time**, which is when the multi block error was at maximum.
   h. Plot the **Mesh**.
2. Do you see any multi-block problems? Answers on the next page…



Luckily for you, the *diffusion* and *volume loss* appear to occur after the *critical control* at the weir crest. If you are interested in flow rate over the weir, the multi-block error won't affect your results. If, however, you're interested in the nappe impingement and its effect on the downstream topography, then the model has a problem and should be re-meshed to enclose the nappe in a single block.

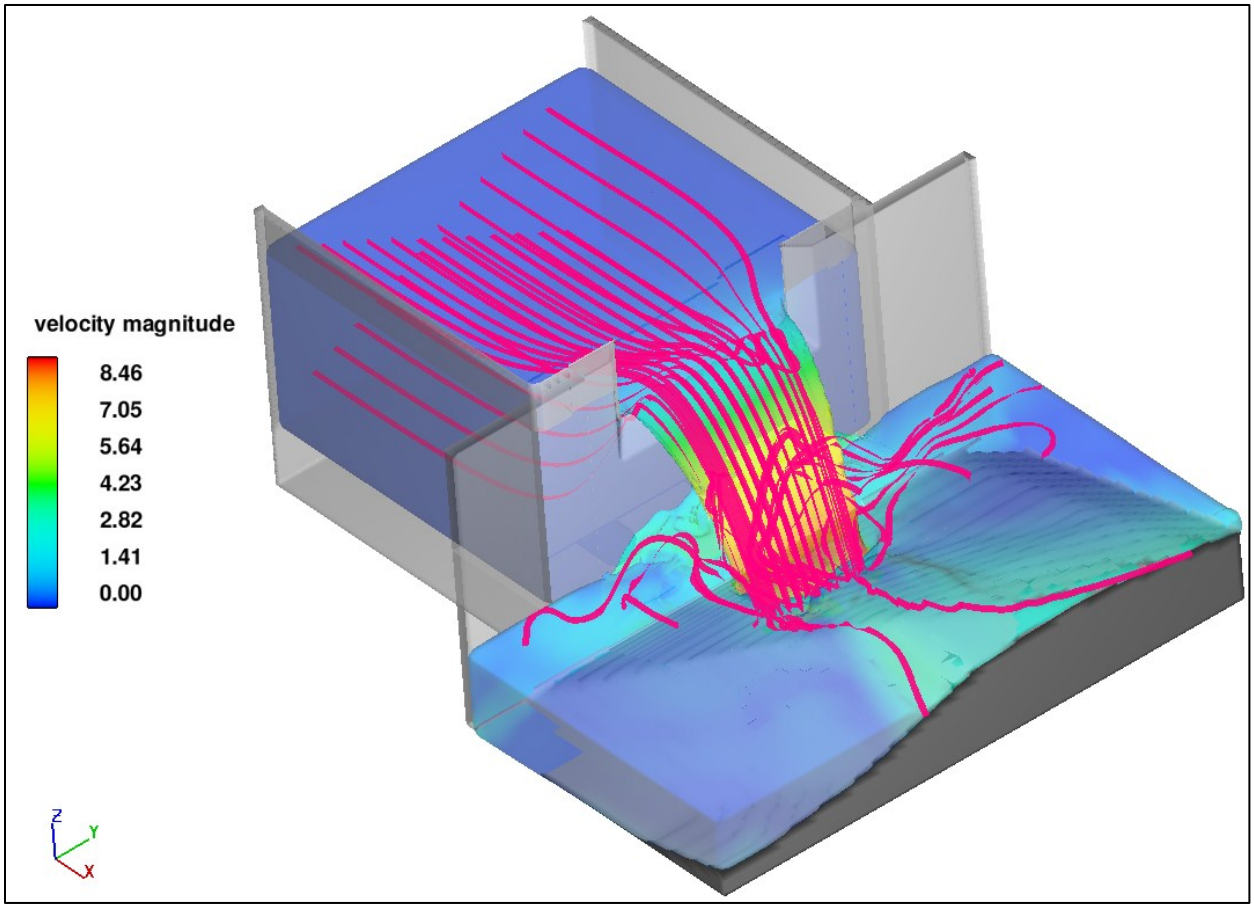
Compare Results to Physical or Empirical Data

At this point, you should:

1. Compare the model results to the physical or empirical results.
2. Identify the relative error as $\boldsymbol{e = (Q_{CFD} - Q_{physical})/Q_{physical}}$.
3. Is the relative error < 5%? If not, try to explain why or make the model better. Sources of error may include:
   a. *Inter-block gridline mis-match* may cause momentum diffusion before or at the weir.
   b. *Non-cubic cell aspect ratios* may diffuse momentum.
   c. *Poor resolution of the crest* may cause too little contraction or nappe spring.
   d. *Mesh-resolution studies* address many potential sources of error. Perform one by reducing all cell sizes by 2x, running, and repeating until *Q* varies less than 2 – 3%.
   e. *Momentum advection* and/or *VOF methods* may diffuse momentum and/or free surfaces.
   f. *Turbulence model assumptions* may affect results.
   g. *Empirical equations have limits of validity* that affect their accuracy: does the modeled situation match the experiments that provided the data?
   h. *Physical test data may be scaled incorrectly*, especially for turbulent effects.
   i. The CFD model may not match the physical model.
   j. The physical model data includes measurement uncertainty. Are error brackets known?

This model predicts a flow rate of about 2.4 m^3/s, while the Kindsvater and Carter (1957) equations predict about 2.1 m^3/s. The disagreement is about 13%. A careful check of the limits of applicability of their equation (found in Bos 1989) show that the model is within the region of applicability for approach head depth, crest height above the channel, channel size, and drop length of the nappe. Feel free to suggest how this model could be improved to your instructor. For one successful example, see Abd El-Hady Rady (2011) "2D-3D Modeling of Flow Over Sharp Crested Weirs", Journal of Applied Sciences Research.

Congratulations on completing your first ***FLOW-3D*** simulation from start to finish!



Exercise 6: Intro to Air Entrainment

Hydraulics Exercise: Air Entrainment

Problem Description & Objectives

Air entrainment downstream of a hydraulic jump will be simulated in three ways. Entrained air will be modeled as a *massless scalar*, as a *miscible scalar with mass*, and as a *drifting dispersed phase* with both density and bubble-drag effects. The student will set up the first model from a set of specifications as an exercise in critical thinking. Hints and suggestions will be included after the specifications.

The experiment models a bent pipe that discharges below the surface of a standing pool. A hydraulic jump forms in the downward bend of the pipe, as shown in **Figure 1**, below. To reduce the simulation time, only the downstream portion of the pipe will be modeled, using 2-D CFD, and this domain is illustrated in Figure 1. The 2-D assumption will limit the validity of the results; the exercise is for illustration of the various approaches rather than a formal validation.

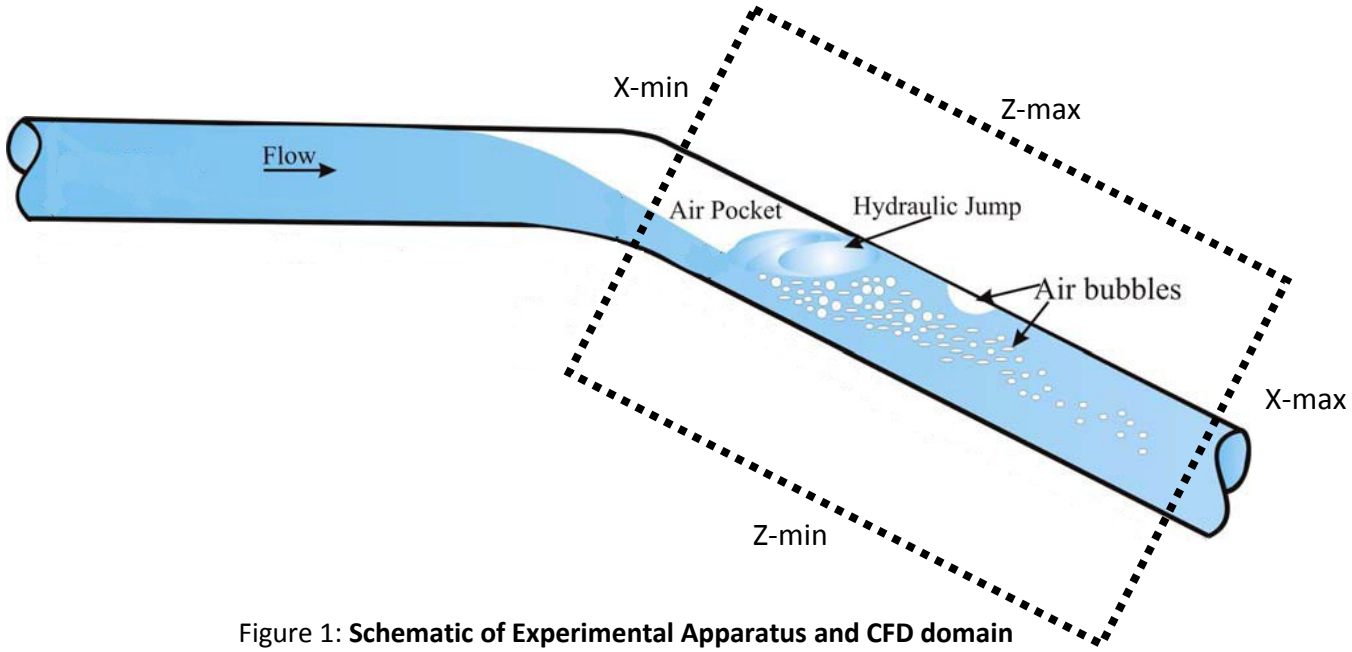


Figure 1: **Schematic of Experimental Apparatus and CFD domain**

FLOW-3D Learning Objectives

In this exercise you will practice techniques to:

1. Create **2-D** geometry and mesh,
2. Use **component gravity** to create an angled (with respect to gravity) domain,
3. Set boundary conditions (including turbulent energy) based on Froude number and depth,
4. Use the **Air Entrainment** model, and
5. Use the **Density Evaluation** and **Drift-flux** models.

Determine Component Gravity Parameters

We want to model the downstream portion of the pipe in such a way that the x-axis is parallel to the pipe walls, and the z-axis is normal to the flow direction, as illustrated in **Figure 2**. The downstream leg of the pipe is 15 cm diameter and angled at a 10% slope (i.e., the slope of the pipe is 10 cm/100 cm). We can use this to determine component gravity:

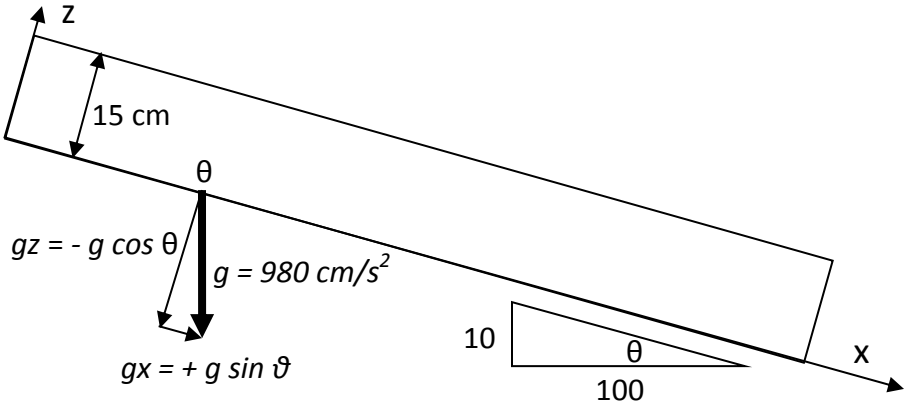


Figure 2: **Determining component gravity from pipe slope**

The angle of the pipe θ as measured from horizontal is $\tan^{-1}(10/100) = 5.71059°$. It is easy to show that θ is also the angle between gravity and the vector normal to the flow direction. Therefore:

gx = 980 sin 5.71059° = **+ 97.5136 cm/s²**
gz = - 980 cos 5.71059° = **- 975.136 cm/s²**

Another way to solve for *gz* would be to start with slope: $gx = gz/10$, so $[gz^2 + (gz/10)^2] = 980^2$.

1. Add a new simulation called **Air Entrainment Scalar**.
2. Set the **Simulation Units = CGS**.
3. Set **Gravity** component **gx** and **gz** given above on the **Physics** panel.

Determine Upstream Velocity from Froude Number and Depth

The flow accelerates and things as it crests the bend of the pipe. Most air entrainment experiments characterize the flow by a depth and Froude number at the control point at the pipe bend. In this case, let the Froude number be 3.0 and the depth be 5 cm.

$$Fr = \frac{U}{\sqrt{gh}}$$

where:

Fr = Froude number, given as 3.0,
U = average velocity of the fluid,
g = gravity, 980 cm/s^2,
h = mean depth of the flow, given as 5 cm.

Technically, *h* is the *hydraulic mean depth*, equal to the wetted area A_w divided by top surface width *T*. Because this is a 2-D model that ignores the curvature of the pipe, the true meaning of *h* will also be ignored (again contributing to the results being "for illustration only").

1. Add a new mesh block, ignoring it's dimensions for now.
2. Rearrange the Froude equation and solve to find the mean upstream velocity **U = 210** cm/s.
3. Make the upstream (x-min) boundary **velocity-type**:
   a. Set **X-Velocity = 210** cm/s.
   b. Set **Free Surface Elevation = 5** cm.

Compare Estimates of Upstream Turbulent Kinetic Energy and Dissipation

Boundary conditions have zero turbulent kinetic energy (TKE) by default. This is often correct if the fluid enters from a resting state like a reservoir. An easy but time-consuming way to model upstream turbulence is to model the upstream pipe flow *explicitly*, and then make a restart simulation with an upstream *grid-overlay (GO) boundary* to copy the calculated TKE to a new boundary in a smaller domain.

A less certain but much faster approach is to use empirical estimates of turbulent parameters. There are several ways to approach this. Ideally the approach is consistent in all respects with the turbulence model and flow geometry. In this case, although you will use the RNG turbulence model and a partially full pipe modeled as a 2-D plane, the empirical estimates below are derived from the *standard k-e* model and empirical approximations for *full round pipes*. The discrepancy between approaches is a source of uncertainty.

$$Tu = 0.16\, Re^{-1/8}$$

$$k_T = 1.5\, (U\, Tu)^2$$

$$L_T = 0.07\, D_H \cong 0.07h$$

$$\varepsilon_T = 0.09^{3/4}\, k_T^{3/2} L_T^{-1}$$

$$Re = \frac{\rho\, U\, D_H}{\mu} = \frac{4\, A_w \rho\, U}{P_w\, \mu} \cong \frac{4\, \rho\, U\, h}{\mu}$$

where:

Tu = initial intensity of turbulence (dimensionless, equation is empiric for fully-developed pipe flow),
Re = pipe Reynolds number (dimensionless),
A_w = wetted area (cm^2),
D_H = characteristic hydraulic length, taken to be the pipe hydraulic diameter (cm),
h = depth of 2-D flow (cm),
k_T = mean turbulent kinetic energy per unit mass (cm^2/s^2),
ε_T = rate of dissipation of turbulent kinetic energy per unit mass (cm^2/s^3),
L_T = turbulent length scale, aka turbulent mixing length, about 7% of pipe diameter (cm),
P_w = wetted perimeter (cm),
U = mean fluid velocity (cm/s),
μ = fluid dynamic viscosity (g/cm/s),
ρ = fluid density (g/cm^3),

The upstream depth of flow is 5 cm, velocity is 210 cm/s, density is 1 g/cm^3, and viscosity is 0.01 g/cm/s. The Reynolds number is therefore 4.2E5, so the flow is *fully turbulent*. Assume the pipe is *hydraulically smooth*. The following empirical upstream boundary turbulence values apply:

- ***Tu* ≈ 0.0156 (1.56%)**
- **k_T ≈ 16 cm^2/s^2**
- **L_T ≈ 0.35 cm (potential range 0.25 – 0.65 cm)**
- **ε_T ≈ 30 cm^2/s^3.**

A second way to approach the problem is to put brackets on the initial turbulent intensity. Most pipe flows exhibit *Tu* between 1% and 5%. In rivers *Tu* often exceeds 10%! The brackets give us two more possible values:

Estimation Method	k_T (cm^2/s^2)	ε_T (cm^2/s^3)
Full-pipe *Tu* ≈ **1%**	7	8
Full-pipe *Tu* ≈ 0.16$Re^{-1/8}$ ≈ **1.6%**	17	33
Full-pipe *Tu* ≈ **5%**	165	998

***** Note that there is no widely agreed-on method of deducing upstream turbulent parameters, and, as shown in **Figure 3** below, the effect of an arbitrary value can be significant! For this reason, some CFD modelers prefer to *model upstream turbulence explicitly*, even if it requires more than one simulation.

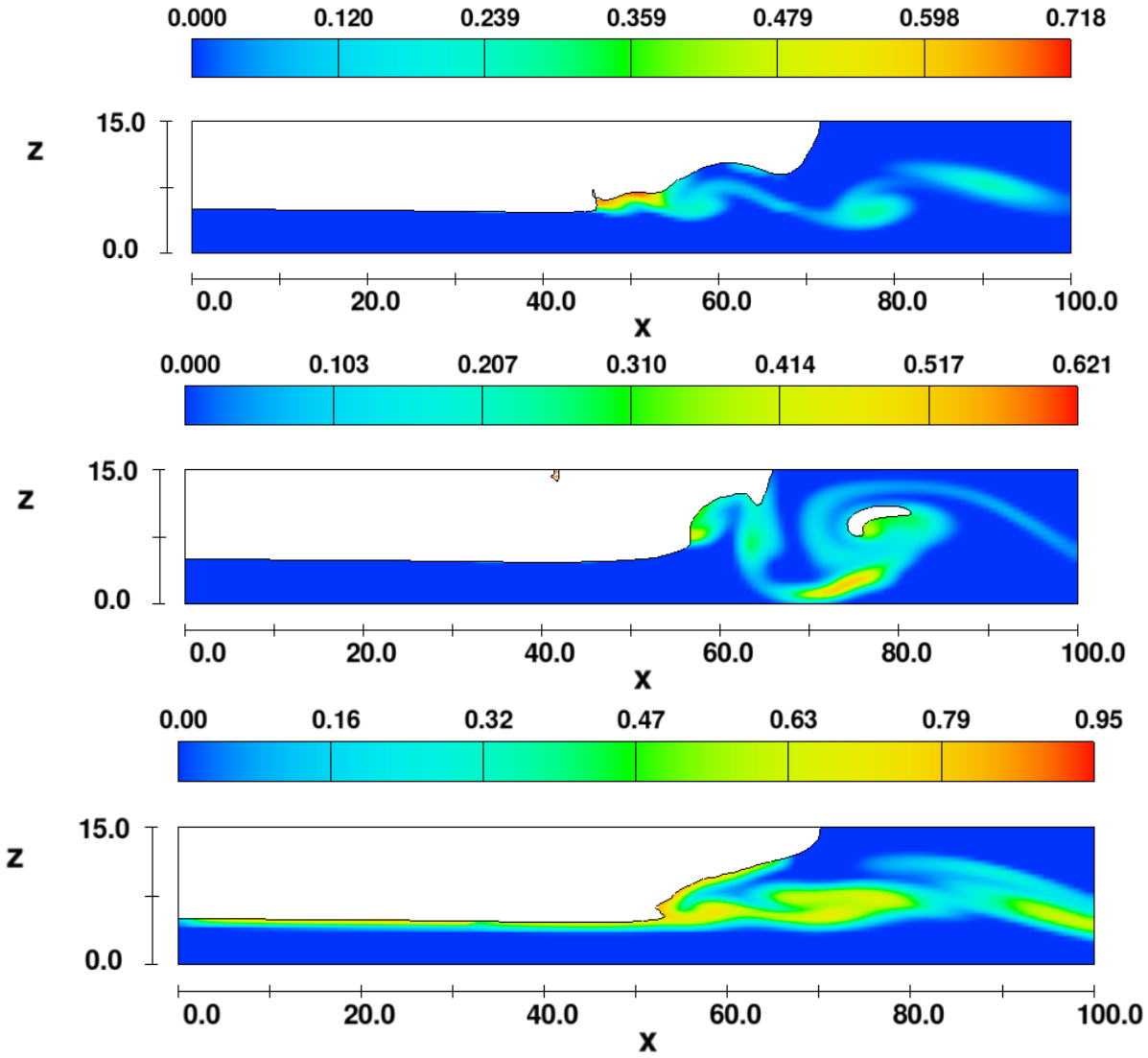


Figure 3: **Volumetric fraction of entrained air and jump shape at 3 seconds as a function of upstream boundary turbulence (top to bottom: k_T = 80, 660, 4400 cm^2/s^2, ε_T = 140, 3600, 61500 cm^2/s^3)**

Problem Specification (Run 1: Scalar Air Entrainment)

General, Physical, and Numeric Specifications

- **Simulation name: Air Entrainment Scalar**
- **Simulation units = CGS**
- **Simulation time = 5.0 seconds**
- **Number of processors = 2**
- **Initial time step = none needed**

Physics

- **Air Entrainment**
  - **rate coefficient = 0.5,**
  - **surface tension coefficient = 73,**
  - **air density = 1.2E-3 g/cm^3**,
- **Gravity**
  - **gx = 97.5136 cm/s^2,**
  - **gz = - 975.136 cm/s^2,**
- **Viscosity & Turbulence**
  - **viscous flow,**
  - **renormalized group (RNG) turbulence model,**
  - **dynamically-computed TLEN**

Fluid Properties (generic properties for water)

- **Density ρ = 1.0 g/cm^3**
- **Viscosity μ = 0.01 poise (i.e. g/cm/s)**
- **Compressibility = 1/(ρ w^2) = 4.653E-11 cm-s^2/gm**

Output

- **Additional Output = Hydraulic Data**
- **Selected data at 1% time intervals**
  - **dynamic viscosity**
  - **hydraulic data (add to Additional Output)**
  - **volume fraction of entrained air**

Geometry, Initial, and Boundary Specifications

- **No geometry. Walls will be defined as boundary conditions.**
- **Boundaries and Initial Regions as shown in Figure 4, below.**
- **Initial Pressure = Hydrostatic**

Measurement Stations

- ***Flux Surface* Baffle w/ Porosity = 1 at X = 95 cm**
- **History Probe at X = 60 cm, Y = 0.5 cm, Z = 7.5 cm**
- **Sampling Volume from X = 60 to X = 95, Y = 0 to 1, Z = 0 to 15 cm**

Set up the model and check each bullet off as you go.

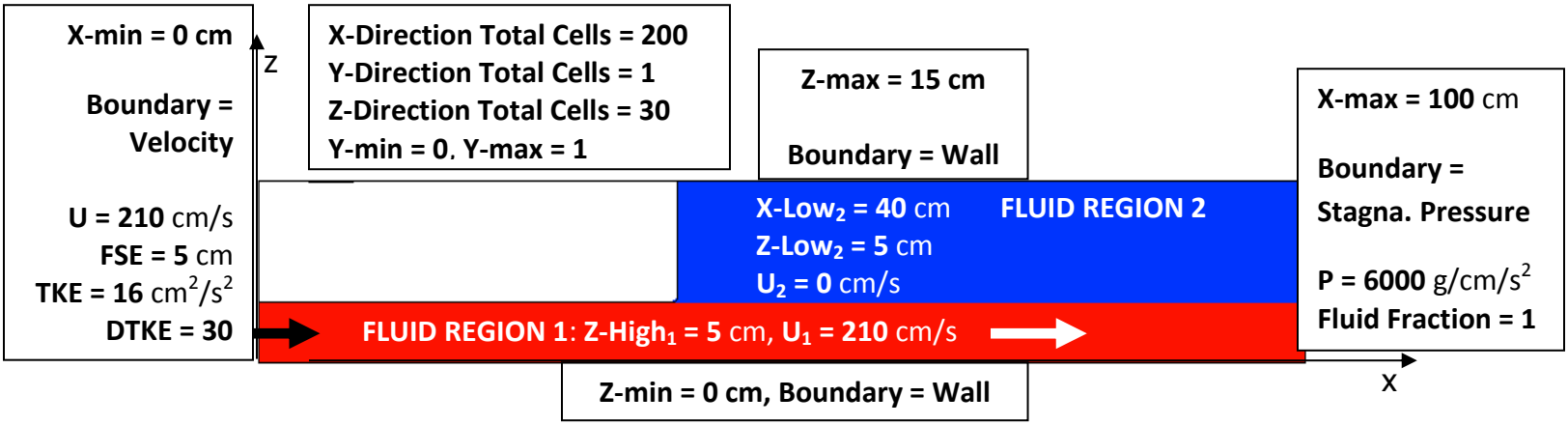


Figure 4: **2-D definition sketch w/ initial fluid regions. Measurement locations not shown.**

Check the Set Up

Use the problem specification statement and the sketches above and below to set up the simulation. If you are unsure of a value or selection, ask the instructor. Check that you've added all the definitions. You may need to *right-click* the **History Probe** and **Baffle** to adjust their transparency and the probe's size. Note that there is no solid geometry in this simulation.

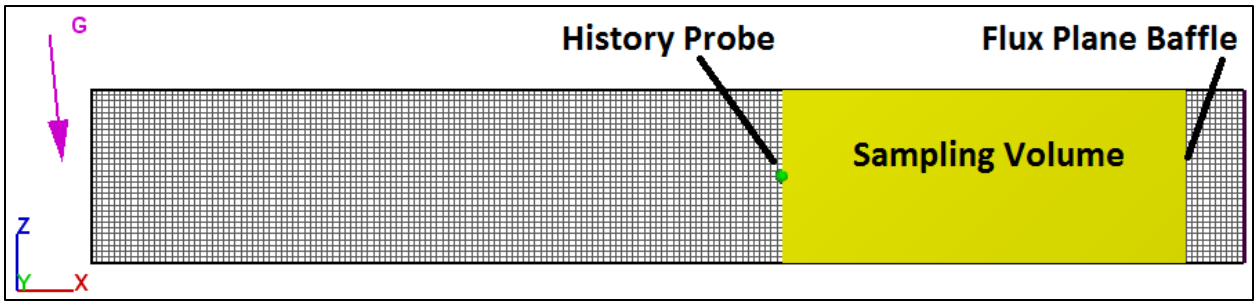


Figure 5: **Finished mesh**

Run #1: Scalar Air Entrainment

1. Run the simulation: it should take minutes or less.
2. Check if the flow has reached steady state by looking at the *runtime plots*. It would be difficult to claim that flow is at steady state.
3. Check your work by **2-D** plotting **Dynamic Viscosity** and **Volume Fraction of Entrained Air** as contour variables using **Selected** data. See that the boundary layer develops rapidly and observe the pattern and magnitude of scalar air entrainment.
4. Check the final time step output against **Figure 6** below before proceeding. If your results don't match, try to determine why, and then ask an instructor for help.

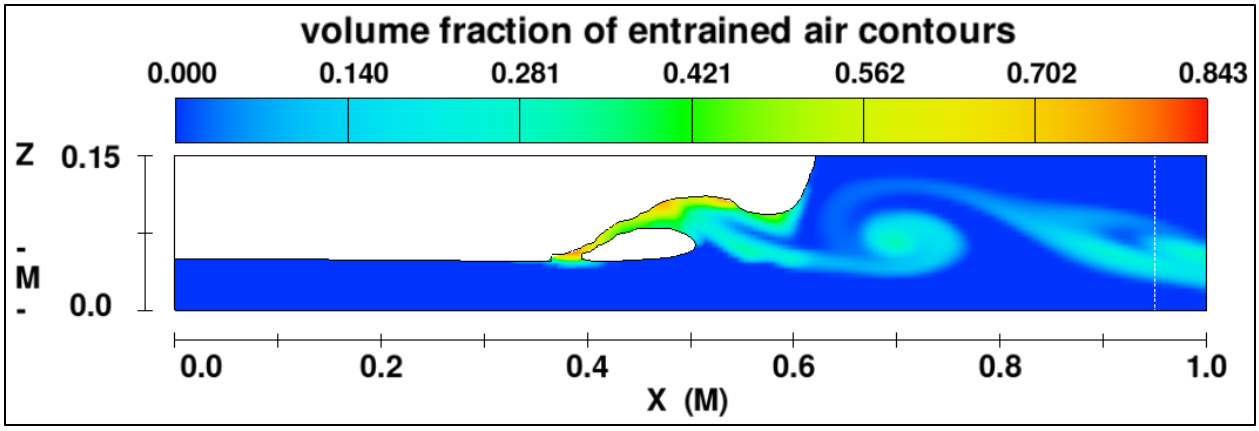


Figure 6: **Check your scalar results visually at the final time-step**

Use **Analyze > Probe tab > General History data** to check the results for the **History Probe**, **Flux Surface**, **Sampling Volume**, and **Total Domain**. These three measuring techniques are the built-in methods of tracking entrained air. Activate the following check-boxes (units are given following the variable):

- **fluid probe #1: volume fraction of entrained air:** cm^3 air/cm^3 air-water mix
- **fluid 1 volume flow rate @ flux surface 1**: cm^3/s air-water mix x +/- unit vector
- **entrained air flux @ flux surface 1**: cm^3/s air, may be negative x +/- unit vector
- **sampling volume 1: entrained air volume**: cm^3 air at time t
- **entrained air volume**: cm^3 air at time t in the *entire domain* including *ghost cells*

Note that the outputs above count only *entrained air scalar* and do not include *resolved void bubbles*. Do not use **Mesh Dependent History** data, as it will only report the *specified* air volume fraction in the ghost cells (zero), and not the *calculated* volume crossing it.

Example Calculation for Flux Plane:

1. Select **Units > CGS** to avoid converting the output units.
2. **Render** the **Probe > General History** data variables above as **Text**. The variables are written left-to-right in the same order that they are listed (top-to-bottom) in the selection list.
3. Find **TIME = 5** seconds (the last output time).
4. Find **FFLUX1** (the Fluid 1 Flow Rate at the **Flux Plane**) = **1050** (cm^3 air-water mix)/s. Since **Density Evaluation** physics are not activated, the density of the *mixture* is the density of *pure water*. No density variation due to entrained air is included. Therefore the liquid flow rate through the plane $\mathbf{Q_{V,WATER} = 1055\ cm^3\ water/s}$.
5. Find **FPSCL1** (the volumetric air concentration at the plane) = **96.1** "amount"/s. The units for FPSCL1 are tricky: they represent a scalar, so ***FLOW-3D*** reports them as g/s. In actuality, scalars can be defined with any consistent measure of amount unless **Density Evaluation** physics are active: then density scalars must be defined as mass/volume. So what is the measure of "amount" for the scalar **Air Entrainment** model? It's a dimensionless volumetric fraction: the volume of air per volume of liquid. When the fraction and velocity are integrated over the area of the flux plane, the result is a *volumetric* flow rate (despite what ***FLOW-3D*** reports). So **FPSCL1** $\mathbf{= Q_{V,AIR} = 96.1\ cm^3\ air/s}$.
6. **% concentration of air in the mixture** $\mathbf{= Q_{V,A}/Q_{V,MIX}}$ = 96.1/(1055 + 96.1) = **8.3%** by volume crossing the flux plane at t = 5 seconds. This is an *instantaneous* value in an oscillating flow; the *time-average* is better for reporting and is **6.6%** by volume over the final 2.5 seconds.

The **Scalar** model should only be used without **Density Evaluation** physics when the volume of air is less than 1% or 2%. When the fraction is higher the mixture should include bulking and density effects.

Run #2: Add Density Evaluation Physics

When the entrained air exceeds some volumetric concentration, bulking begins to increase the mixture volume and decrease the mixture density. Experiments suggest that bulking begins to affect the flow when the entrained air is between 1% and 3% by volume. Modeling the bulking mixture effects requires adding **Density Evaluation** physics.

1. Select **Simulation Manager > File > Add Simulation Copy…**
2. Name the copy **Air Entrainment Density**.
3. Activate **Model Setup > Physics > Density Evaluation > First Order Approximation to Density Transport Equation**. Entrained air transport models *must* use the transport equation option.
4. Open the **Air Entrainment** physics dialog. Notice how **Activate Bulking** is now active.
5. No other changes are required. **Save** and **Run Simulation**.

The simulation will run more slowly with the added physics, but will still finish quickly due to the small number of cells (6000 total).

1. **Open** the results file. Make sure you're not using the old results from the last simulation!
2. **Check Solver Warnings:** there are many *F-packing errors* at the toe of the jump. Ignore them.
3. **2-D** plot **Volume Fraction of Entrained Air**, using **Contour Limits = Global**, and **Vector Type = No Velocity Vectors**.
4. Look for the extent and characteristics of the entrained air. The air now rises to the surface, but cannot leave the water, even at free surfaces (a limitation that will be addressed in the next run).
5. Compare your last output to **Figure 7**, below. Did you set up the model correctly?

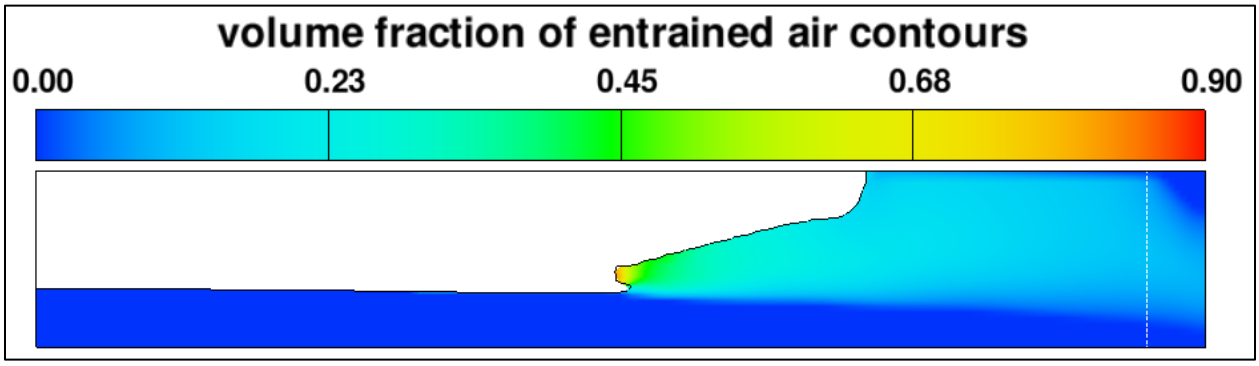


Figure 7: **Check your density-evaluation results visually**

The **Flux Plane** is visible in the plot as a dotted line. Calculating as before, we find at t = 5 seconds:

1. $\mathbf{Q_{V,MIX}}$ **= FFLUX1 = 1236** cm^3 air-water mixture/s: 25% higher than in the scalar run! Explain why.
2. $\mathbf{Q_{V,AIR}}$ **= FPSCL1 = 78** cm^3 air/s. Note that the units are still *incorrectly* reported as g/s: when baffle scalar flux **FPSCL1** is *entrained air* its units are cm^3/s.

3. **% concentration of air in the mixture = $Q_{V,AIR}/Q_{V,MIX}$ = 6.3%** by volume at t = 5 seconds. The *time average* is **6.7%** by volume over the final 2.5 seconds, similar to the scalar-only physics results. The *magnitude of oscillation* is much smaller with density physics than with scalars only.

Run #3: Add Drift-Flux Physics (Two-Phase Flow)

The results of the model with **Density Evaluation** physics look more physical than with scalar-only entrainment. This indicates that the bubble concentration cannot be modeled as if it were molecularly dissolved in the water. Now consider the possibility that the air is entrained as a *cloud* of tiny bubbles, which not only *bulk* the fluid (volume and buoyancy effects) but interact with the flow in three additional ways: by *drifting* at a different velocity than the water, by exerting form and wake *drag* on the water, and by *escaping* to the atmosphere through the free surface. These extra three effects are modeled by **Drift-Flux** physics. With the drift-flux model, you define the properties both of a *continuous* **Phase #1** (in this case, water), and a *dispersed* **Phase #2** (air bubbles).

1. Make a copy of **Air Entrainment Density** and name it **Air Entrainment DriftFlux**.
2. Activate **Drift-flux** on the **Physics** tab and apply water air bubble values per **Figure 8**, below.
3. **Save** and **Run Simulation**.

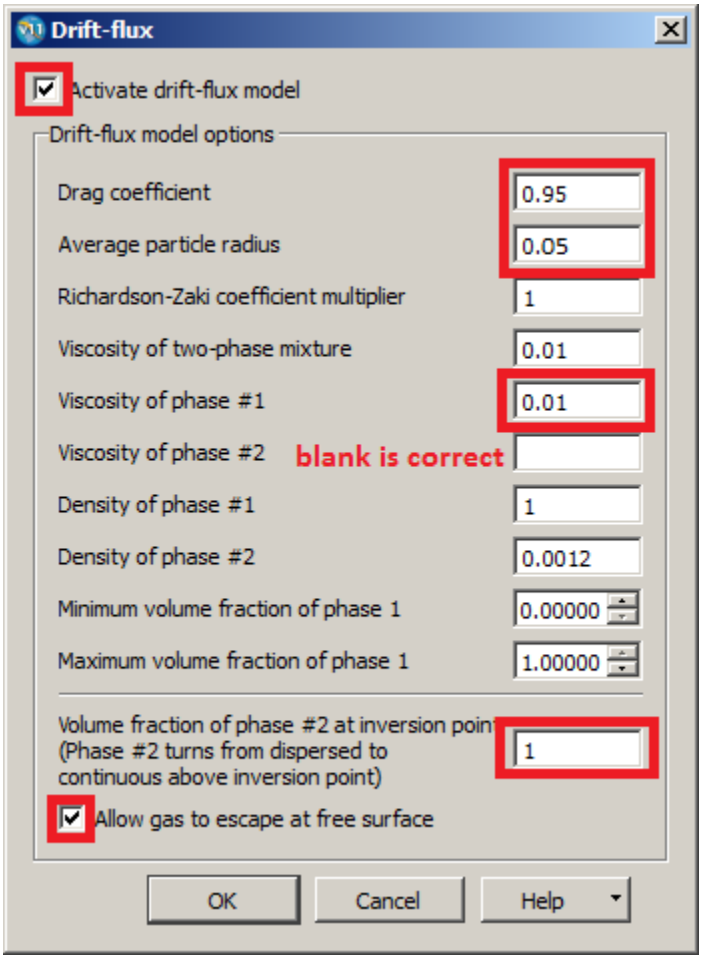


Figure 8: **Parameters for air bubbles and water for drift-flux model**

A brief description of the parameters in **Figure 8** follows. Values for air bubbles can be found in the literature, and a description of the drift-flux model equations that can be used to fit parameters to experimental data can be found in the User Manual and in Technical Note 83 at users.flow3d.com. Note that the **Drift Flux** model *requires* **1st or 2nd order Density Evaluation** physics.

- **Drag Coefficient** is the fully-turbulent limit of the drag coefficient C_D. ***FLOW-3D*** uses a smooth function that gives Stokes settling in the laminar regime and drag defined by C_D in the fully turbulent regime. The default value of C_D is 0.5, which is correct for dense spheres. Values of 0.75 to 1.5 are typical for sand and gravel. **0.95** has been demonstrated to be accurate for buoyant spheres ($\rho_S < 900$ kg/m^3), where the rising sphere exhibits a corkscrew motion due to pressure/buoyancy effects that causes more drag; see Karamenev (1992).

- **Average Particle Radius**: typically taken to be between 0.5 mm and 1 mm for air bubbles, but the data scatter is significant. Impinging jets often exhibit bidisperse-normal distribution of bubble radius, which complicates the estimate of a single mean radius.

- **Richardson-Zaki Coefficient Multiplier**: this should be **1.0** for the standard Richardson-Zaki correlation, which describes the additional effect of particle-particle (bubble-bubble) wake interaction in a concentrated cloud of the dispersed phase. The multiplier can be used to curve-fit existing data, if needed. See the Tech Note for the full equation.

- **Viscosity of Two-Phase Mixture**: the viscosity used in the Navier-Stokes cell-to-cell momentum equations, regardless of phase concentrations. Typically assumed to be the viscosity of the continuous phase, so here it is the viscosity of water. Values entered here will change the value of **Fluid 1 Viscosity** on the **Fluids** tab. If the cell-to-cell viscosity varies, consider borrowing the solidification variables as discussed in the lecture on **Multiphase Flow**. An example of this situation would be two liquids (water and oil, e.g.) that flow as unmixed phases in different parts of the domain. Here, the continuous phase will always be water.

- **Viscosity of Phase #1** is the viscosity used in the in-cell evaluation of *relative (separation) velocity* between the continuous and dispersed phases. Set this to the viscosity of water also.

- **Viscosity of Phase #2** is the viscosity of the dispersed phase if it were to become so concentrated as to be continuous. This only has meaning for dispersed phases that can flow, such as oil droplets. Leave it blank, and it will default to **Fluid 1 Viscosity**, that of water. This is the *best representation* of the system, because no matter how many air bubbles are present, they are still separated by thin liquid films.

- **Density of Phase #1** is the density of the continuous phase, pure water. Values entered here will change **Fluid 1 Density** on the **Fluids** tab.

- **Density of Phase #2** is the density of the dispersed phase, pure air. This value was entered on the **Air Entrainment** physics dialog, and changes made *here* will change the value *there*.

- **Minimum Volume Fraction of Phase 1** is the irreducible fraction of water allowed in a cell. Unity minus this value is the maximum fraction of air allowed in a cell. The correct value should be

some small number that reflects the thin film of water between bubbles, but this value is not known, so use **0** as a reasonable approximation.

- **Maximum Volume Fraction of Phase 1** is the upper limit fraction of water permitted in a cell. 1– this value is the irreducible minimum air permitted in a cell. Water can be present without air in it, so **1** is appropriate.

- **Volume Fraction of Phase #2 at Inversion Point** is the maximum volumetric fraction of air in a cell before the continuous and dispersed phases switch so that air is considered continuous. If modeling another liquid, 0.5 would make sense. Because we assume air bubbles will always be discontinuous (separated by films), **1** is appropriate. The equations of the **Drift Flux** model do not make physical sense for any other value when the two phases are different states (liquid vs. solid or gas).

- **Allow Gas to Escape at Free Surface** allows phase #2 to disappear at free surfaces and should be activated for the best physical representation of bubbles.

The model will take even longer to run than the last one, due again to the added physics. Review the results as before, but use **Auto** contour limits this time. Compare your results to **Figure 9**.

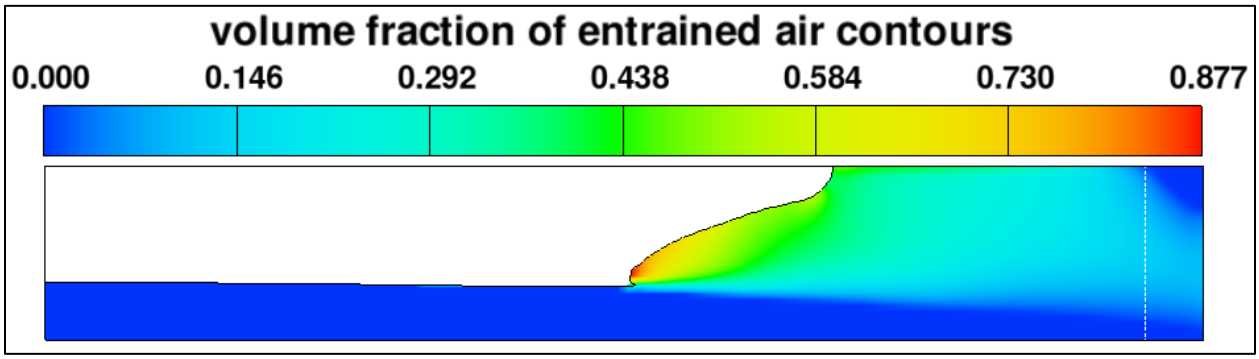


Figure 9: **Final time step (t = 5.0 seconds) for drift-flux model version of problem**

Calculate as before to find the instant and time-averaged air flow:

1. $\mathbf{Q_{V,MIX}}$ **= FFLUX1 = 1141** cm^3 mixture/second at t = 5 seconds.
2. $\mathbf{Q_{V,AIR}}$ **= FPSCL1 = 82** cm^3 air/s, 5% greater than in the density-evaluation only model.
3. **% concentration of air in the mixture =** $\mathbf{Q_{V,AIR}/Q_{V,MIX}}$ **= 7.2% by volume** at t = 5 and **6.6%** by volume averaged over the final 2.5 seconds, about the same as the density-only model.

One last calculation:

1. View the results in **2-D** using **Restart Data** and **Contour Variable = Macroscopic Density**.
2. Now use **Probe > General History Data** to plot text for **fluid 1 volume flow rate @ flux surface 1 (FFLUX1)** and **mass flow rate @ flux surface 1 (MFLUX)**.
3. Divide the mass flow rate (g/s) by the volume flow rate (cm^3/s) at the last time step to find the *spatially-averaged mixture density* ρ_{MIX} at the flux surface. Use the equation:

$$\%\ concentration\ of\ air\ in\ mixture = \frac{V_{AIR}}{V_{MIX}} = \frac{\rho_{MIX} - \rho_{WATER}}{\rho_{AIR} - \rho_{WATER}}$$

The percentage of air calculated from the density is the same as was determined by dividing **entrained air flux @ flux surface 1 (FPSCL1)** by **fluid 1 volume flow rate @ flux surface 1 (FFLUX1)**. This verifies that FPSCL1 is the *volume fraction* of air in the mixture, not g/s. You can use this approach for any density scalar to determine if the reported units are correct.

Intro to Testing for Uncertainty

One interesting test is to examine the *model sensitivity* to the air bubble size. If the mean radius is increased from 0.5 mm to 1 mm, the entrained air flow rate increases by a factor of 3.75! The sensitivity of this input increases *uncertainty* in the model. It's an example of *input uncertainty*.

Another important test is the *mesh dependency* of the results. Turbulence is known to be quite mesh dependent, and since the physics model uses TKE to calculate air entrainment, the volume of entrained air is also likely to be mesh dependent. In this case, halving the cell sizes in each direction twice for a total of three runs on three meshes changes the time-average air concentration from **6.6%** to **9.8%** and then to **4.0%**. Note that the variation is mostly due to the effect of the cell size on the hydraulic jump itself: the jump location experiences significant boundary effects from the nearby downstream pressure boundary, and so it's a poorly-posed problem to begin with.

One way to use the air entrainment model is to find a mesh that you can live with and then use it to model a known air entrainment experiment with similar Reynolds and Froude numbers as the target model. Calibrate the air entrainment rate coefficient and/or bubble size until you match the experiment, and then model the target case.

If no experimental data is available with similar *Fr* and *Re*, then successively vary the mesh resolution, air entrainment rate coefficient, and bubble size. Use probability distributions that describe the likely possibilities of the "real" value of the input variables to reduce the number of models, if possible. Find the range of CFD output and develop uncertainty bands around the mean results.

The above two options are time-consuming and are appropriate when *quantitative* results are needed, like the exact volume of entrained air. It's much easier to use air entrainment physics when all you need are *qualitative* results. For example, if one design option causes much less air entrainment than the other, you can pick the better design without worrying much about the input and numeric uncertainty!

References

Brethour, J.M. and C.W. Hirt (2009). *Drift Model for Two-Component Flows*. FSI-09-TN83Rev. Flow Science Inc.: Santa Fe, NM.

Hirt, C.W. (2003, 2011). *Modeling Turbulent Entrainment of Air at a Free Surface*. FSI-03-TN61R. Flow Science Inc.: Santa Fe, NM.

Exercise 7: Intro to General Moving Objects

Hydraulics Exercise: General Moving Objects

This is a simple example to illustrate the general moving object model in ***FLOW-3D***. Surface tension and viscosity are also used. By now you should be familiar with the basic set up methods in ***FLOW-3D***.

Problem Description and Objectives

The model represents a simple piston-driven inkjet print head with a contaminant (particle of dirt) in the ink. There is air above the print head, liquid ink inside it, and a solid object in the ink.

Select General Settings

Begin on the **Model Setup > General** tab. The motion and compressibility of the air can be neglected in this case, so use the one fluid method. The remaining parameters are selected based on knowledge of the physical case.

1. **Number of Fluids = One**
2. **Interface Tracking = Free surface or sharp interface**
3. **Flow Mode = Incompressible**
4. **Units = CGS / Kelvin**
5. **Finish time = 1E-4** seconds
6. **Number of Processors = 2**

Select Physics Options

1. **Moving and Simple Deforming Objects**
   a. **Activate general moving objects (GMO) model**
   b. Select **Implicit** for the Moving object / fluid coupling. The **Implicit** option is usually better when a moving object will have **coupled** (as opposed to **prescribed**) motion. The dirt particle will move freely with the fluid.

2. **Viscosity and turbulence**
   a. Activate **viscous flow**
   b. Select **Laminar** as the turbulence option because the flow is expected to have a low Reynolds number.

3. **Surface tension**
   a. **Activate surface tension model**
   b. Choose **Explicit** approximation
   c. The remaining parameters will be entered on the **Model Setup > Fluids** tab.

Define Fluid Parameters

Only the fluid density, viscosity, and surface tension properties are needed for this simulation.

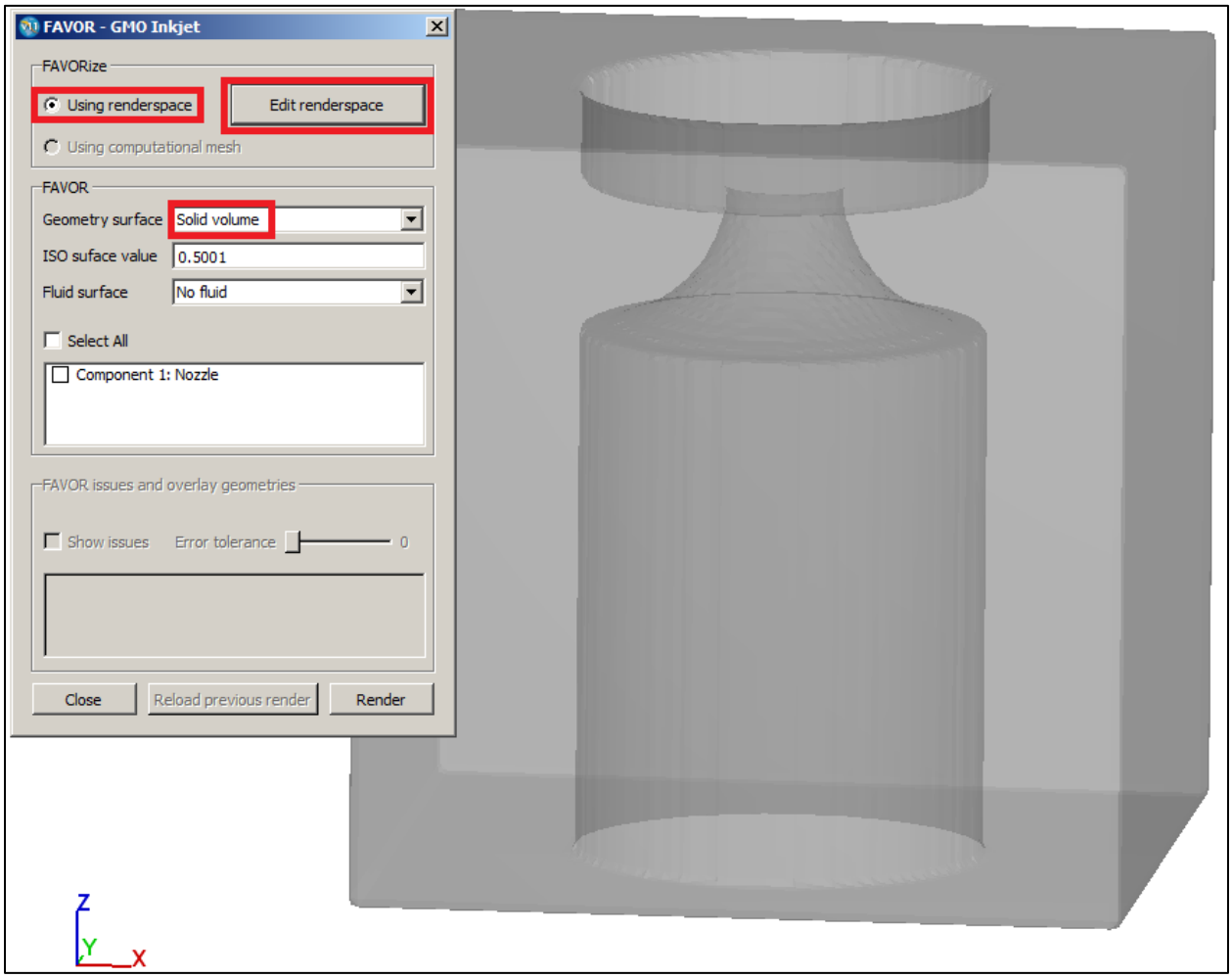
1. Fluid 1
   a. Name = Ink
   b. **Density = 1** g/cm^3
   c. **Viscosity = 0.06** Poise (g/cm/s)
   d. Surface tension:
      i. **Surface tension coefficient = 73** g/s^2
      ii. **Contact angle = 15** degrees. This is a global parameter, and applies to wall boundaries and subcomponents that do not have a different contact angle specified as a surface property.

Define Geometry

The general process for the **Model Setup > Meshing & Geometry** tab will be to *define the geometry*, *assign properties* to the components, *define the mesh* and *boundary conditions*, and then set the *initial state* of the fluid and void. Start by defining the components. The tank will primarily be defined using the mesh boundaries, but the filling tube and valve will be components.

1. Geometry: defined in the **Geometry Window**  .
   a. Nozzle
      i. Create a **solid cylinder subcomponent** with the following dimensions. Name the new **Component 1: Nozzle**.
         1. **Z low = undefined** (blank, not zero), **Z high = 0.15** cm
         2. **Cylinder outer radius = undefined** (make it blank after you build the cylinder)
         3. **Cylinder inner radius = 0.0105** cm. The inner radius is like pipe inner diameter; it must set it in the tree under **Limiters** and with the undefined outer radius will make everything outside the inner radius solid.
      ii. Create a **solid torus subcomponent** with the following dimensions:
         1. **Component = Nozzle**
         2. Torus **Minor radius = 0.0075** cm
         3. Torus **Major radius = 0.0105** cm
         4. **Z low = undefined**, **Z high = 0** cm. This must be set in the subcomponent tree **Limiters**. It cuts the torus across the centerline so only the lower half remains.
         5. Cylinder **outer radius = 0.0105** cm. Also set in the subcomponent **Limiters**; this bevels the outer surface of the torus so it's straight perpendicular in z.
      iii. Move the **torus subcomponent** into position by translating it **0.015** cm in **Z**.

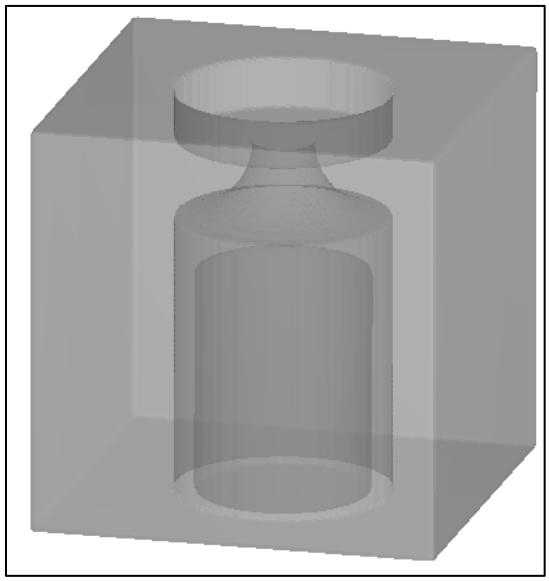
iv. Check the effect of the additional limiters by ***FAVORizing*** the geometry: use a **Renderspace** with extents -**0.02** to **0.02** in all directions and **0.0005**-cm "cells". Because the part is small relative to the unit system, the camera may not be zoomed in enough when you **Render**. Use  to reset the view.



b. <u>Piston</u>

i. Create a **cylinder subcomponent** and assign it to a **new Component 2: Piston**.

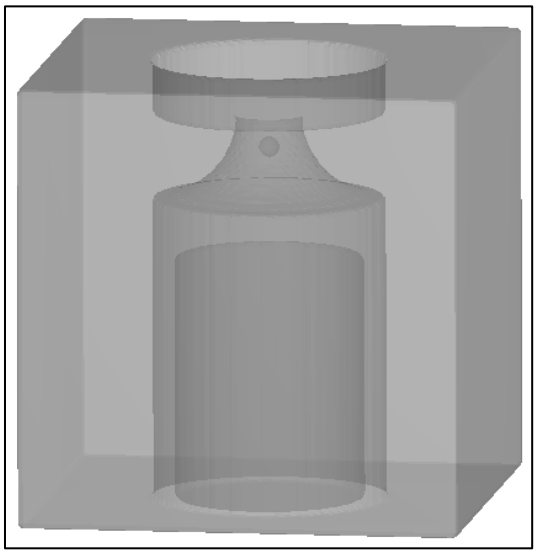
1. **Z low = undefined**, **Z high = 0.002** cm
2. **Cylinder inner radius = undefined**, **Cylinder outer radius = 0.0085** cm. This makes the piston solid.
3. ***FAVORize*** again and check your work against the picture below. You may need to adjust transparency and/or rotate the view to see the piston inside the cavity.



c. Dirt Particle

i. Create a **sphere subcomponent** and assign it to a new **Component 3: Dirt**.

1. **Sphere outer radius = 0.001**
2. Translate it **0.012** cm in **Z**.

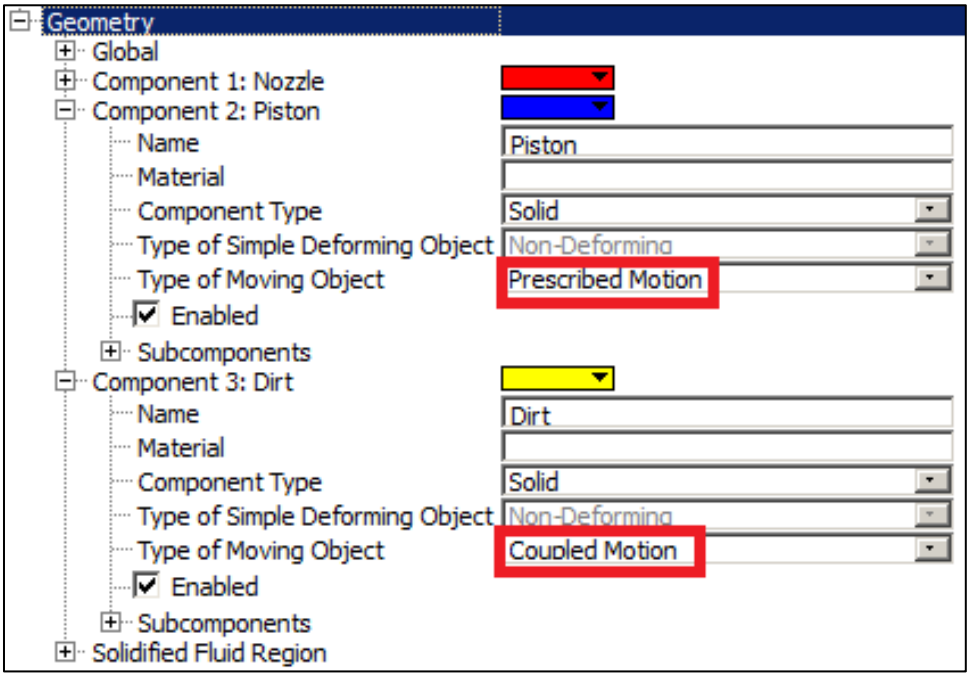


Now add **Component Properties** to the *Piston* and *Dirt* components:

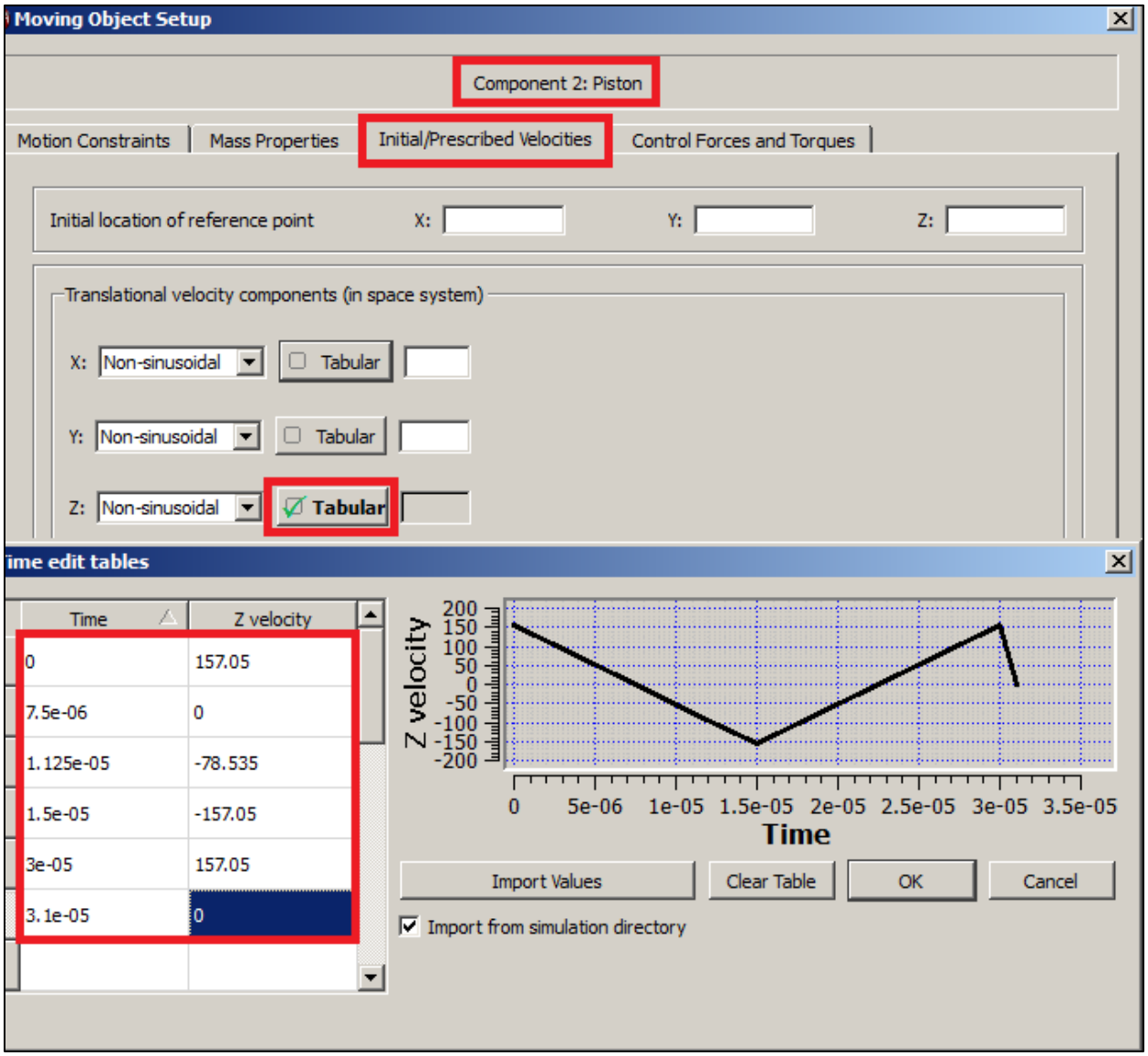
2. Use the tabs at the bottom of the **Component Properties Window** to toggle between the components. The **Type of Moving Object** setting must be set in the **Geometry Window** under the **Component** itself.

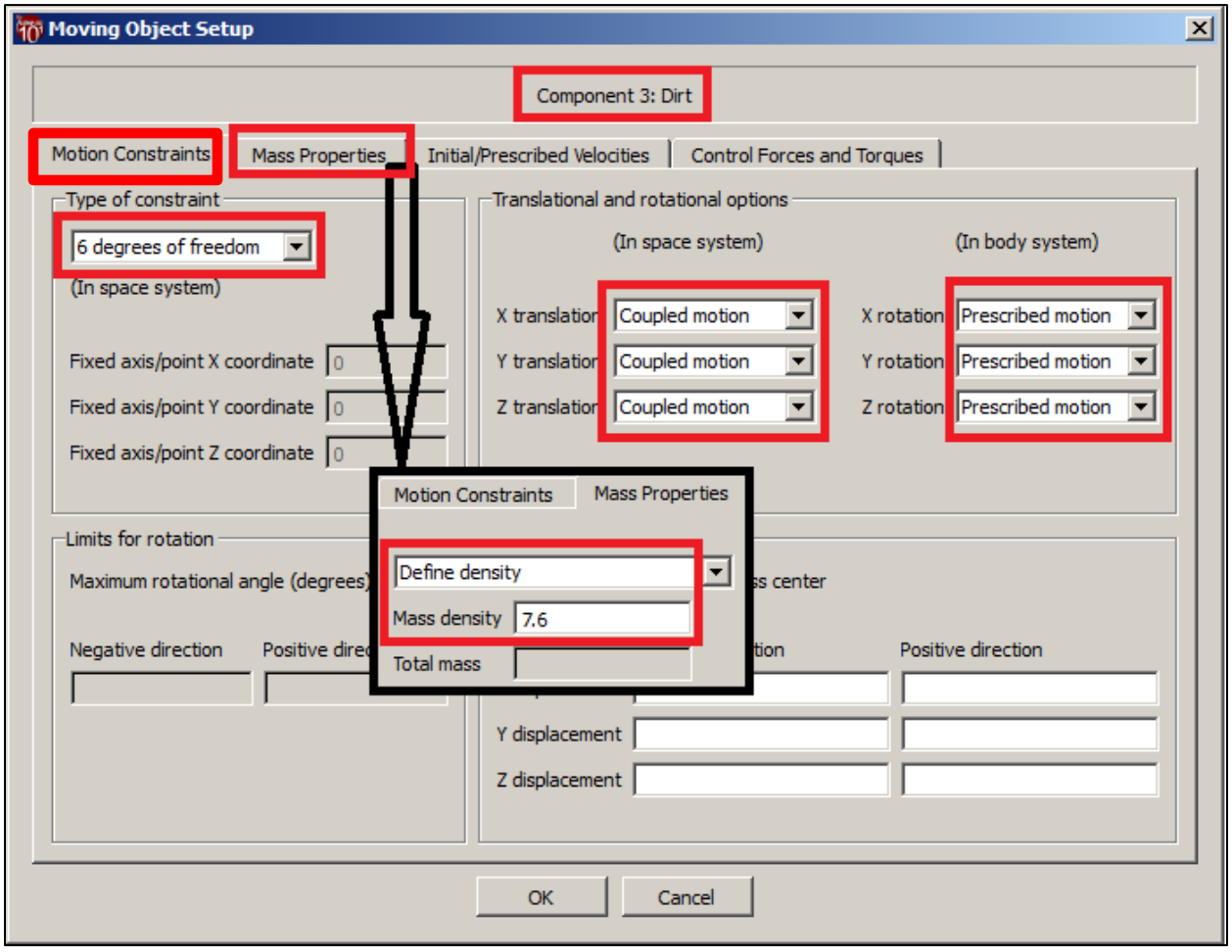
   a. **Component 2: Piston**: the piston will move in the z-direction to eject an ink droplet.
      i. **Component 2 > Type of moving object = Prescribed**
      ii. **Component 2 > Component Properties > Type of Moving Object > Edit Moving Object Properties** > **Initial/Prescribed Velocities tab** > **Tabular Z Translational Velocity Components** per the table below (see next page).

   b. **Component 3: Dirt**: the particle will move with the fluid.
      i. **Component 3 > Type of moving object = Coupled**
      ii. **Component 2 > Component Properties > Type of Moving Object > Edit Moving Object Properties > Motion Constraints tab**: all translations **coupled**, all rotations **prescribed** (defaults to zero, no rotation permitted).
      iii. **Same dialog > Mass Properties tab**: **Mass Density = 7.6** g/cm^3



Check your *Moving Object Setup* dialogs against the images on the next pages.

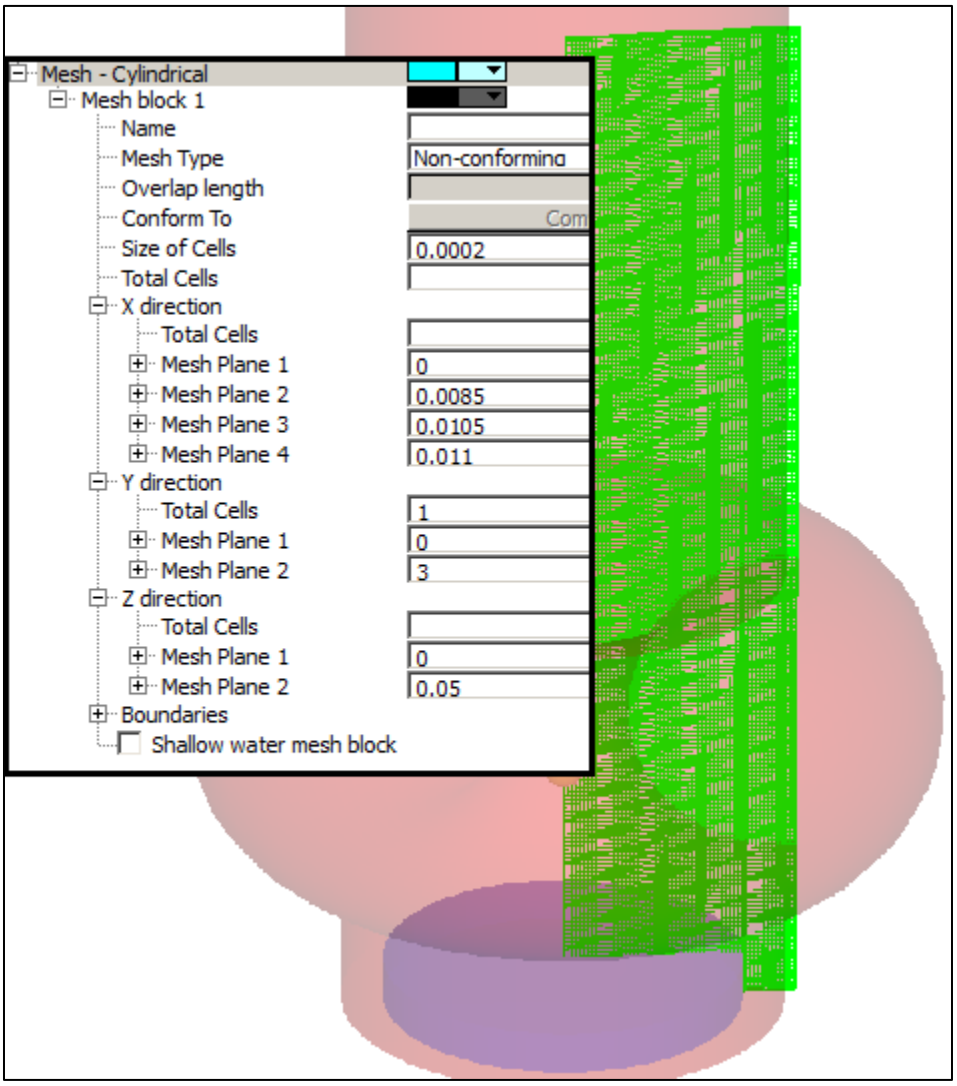

Moving Object Setup
Component 2: Piston
Motion Constraints
Mass Properties
Initial/Prescribed Velocities
Control Forces and Torques
Initial location of reference point
X:
Y:
Z:
Translational velocity components (in space system)
X: Non-sinusoidal
Tabular
Y: Non-sinusoidal
Tabular
Z: Non-sinusoidal
Tabular
ime edit tables
Time
Z velocity
0
157.05
7.5e-06
0
1.125e-05
-78.535
1.5e-05
-157.05
3e-05
157.05
3.1e-05
0
Z velocity
200
150
100
50
0
-50
-100
-150
-200
0
5e-06
1e-05
1.5e-05
2e-05
2.5e-05
3e-05
3.5e-05
Time
Import Values
Clear Table
OK
Cancel
Import from simulation directory



Define Mesh, Boundary, and Initial Conditions

Now add a 2-D cylindrical mesh around the geometry.

1. Mesh: defined in the **Mesh Window**  . Right-click Mesh – Cartesian and select **Change to Cylindrical Coordinates**. **Add** a mesh block and define the 2-D cylindrical mesh to enclose the geometry and align the grid lines (using **X Mesh Planes**) with the outside edges of the piston and nozzle as shown on the next page.



2. Open the **Boundaries** branch of the **Mesh – Cylindrical** tree.
   a. **Z-min** boundary is **Wall**-type.
   b. **Z-Max** boundary is an **Outflow** boundary to allow fluid to exit the domain smoothly.
   c. All other boundaries are or **Symmetry**-type.

The initial conditions for this simulation are an initial depth and pressures such that output will be in gauge pressure (local ambient atmospheric pressure = 0).

3. Open the **Initial Conditions Window** 

.
   a. **Global > Pressure = Uniform Pressure**
   b. **Uniform Pressure = 0** Barye (g/cm/s^2)
   c. **Global > Fluid Initialization = Use Initial Fluid Elevation**
   d. **Initial Fluid Elevation = 0.0149** cm above Z = 0.
   e. **Global > Void Initial State > Pressure = 0** g/cm/s^2

Set Output Options

The default output rates are acceptable. Add **Fluid Velocities** and **Pressure** as **Selected Data**. The GMO location is always output in any time step, so adding any **Selected Data** will give 101 GMO visualizations. If you wanted more frequent numeric GMO output (e.g., mass-center coordinates, forces acting on the object, etc.), you could set **History Data Interval < 1E-6 seconds** (the default 1/100th of the simulation).

Set Numerics Options

There are some numerical options that will help make this simulation run well.

1. **Initial Time Step DELT = 1E-9** seconds
2. **Momentum advection = Second order monotonicity preserving**
3. **Viscous stress solver = Successive under-relaxation**
4. **Multiplier for Convergence Criteria EPSVIS = 0.9**
5. **Over-Relaxation Factor OMEGVS = 0.8**
6. M**aximum Number of Iterations ITVSMX = 50**

Run and Troubleshoot the Simulation

1. **Run** the simulation. If it's set up as defined above, there will be dozens of *consecutive* **pressure iteration failures**.
2. **Terminate** the simulation: select it in the **Queue** on the **Simulation Manager** tab and click  .
3. Look at **Diagnostics > Solver Errors** to find the IJK locations of the failures. At first most of them occur at i = 46, j = 2, k = 3. This is just above the bottom of the domain. Later iteration failures occur at different locations. This doesn't look like a problem with a boundary condition, because the Z-min boundary is Wall type.
4. Try giving the solver more overhead to converge: go to **Numerics > Pressure Solver Convergence Controls** and set **Minimum Number of Iterations ITMIN = 2**, **Maximum Number of Iterations ITMAX = 500**, and **GMRES Subspace Size MRSTRT = 50**. These options will only slow down the solver, and will not relax the convergence criteria. If they work, the solver will run more slowly, but accuracy will either be improved or will not be affected.
5. **Run** the simulation again. This time there should be a few (around 3) pressure iteration failures at the very start of the simulation, and after that there should be none.
6. Check the **Solver Text** for the implicit solver convergence performance: **pressure res/epsi** should be small (around 1E-7 to 1E-2) and **viscous stress res/epsi** should be on the order of **E-1**.

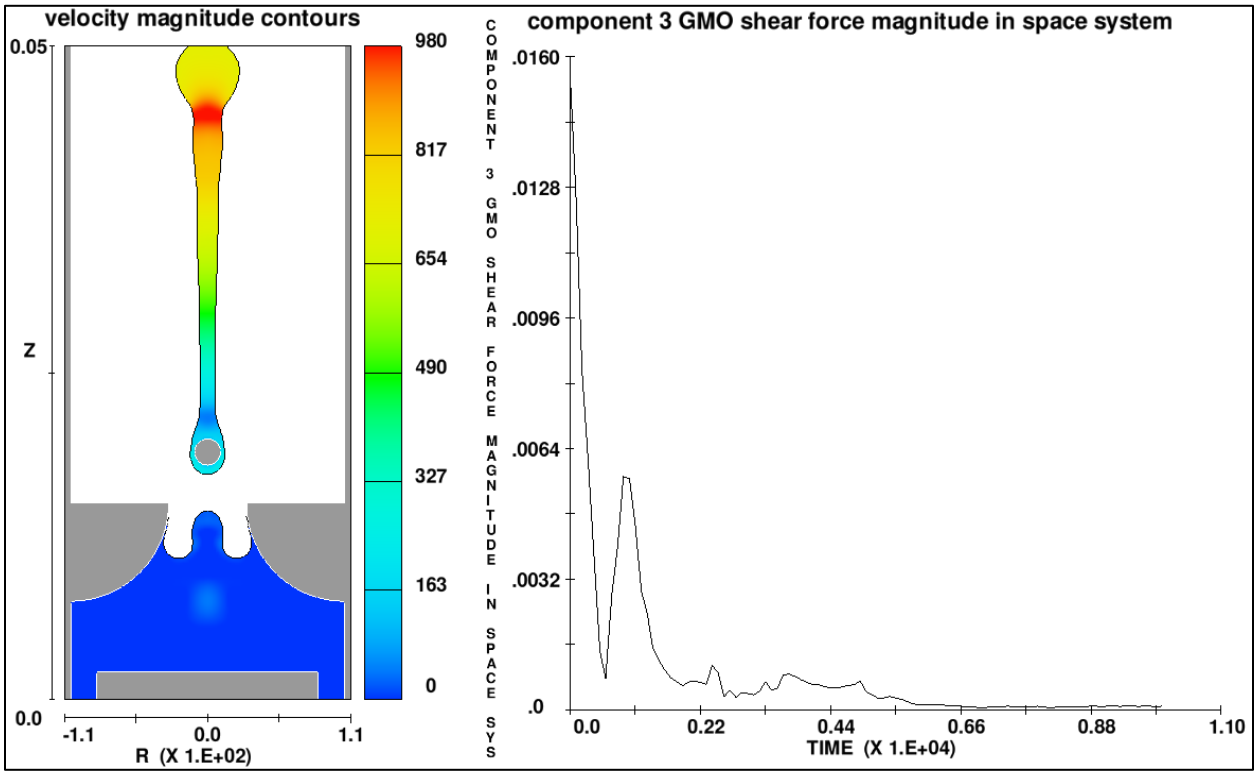
Check for instances where the viscous solver did not converge. The default max iterations for the **Successive Under-Relaxation Viscous Solver** is **iter = 20**, which you raised to 50. Are there any lines where **res/epsi > 1**, and an asterisk ***** is written next to **res/epsi** to mark a failure? Why not?

There are a few viscous stress iteration failures, but they are not consecutive and, unlike pressure iteration failures, they do not affect the time step *dt*.

Postprocessing Visualization

1. Plot **Selected Data** for **Pressure**, and then **Velocity Magnitude**, in **2-D** with **No Velocity Vectors**.
2. Activate the **Horizontal Symmetry** checkbox and **Render**.
3. Compare your output to that shown below at **t = 3.9E-5** seconds.
4. Plot **Probe > General History** data **> Component 3 GMO Shear Force Magnitude in Space System** to see the particle drag magnitude.

The dirt gets trapped in the ink droplet and drag makes the droplet move more slowly and elongate the droplet. This would cause poor printer performance.



Exercise 8: Intro to Saturated Porous Media

Hydraulics Exercise: Saturated Porous Media

Problem Description & Objectives

The flow of water through a crushed coal media filter will be simulated using the **Saturated Porous Media** physics model. Drag coefficients will be determined from experimental data given by Abood (2009). The model results will be compared to the experimental results. The purpose of this exercise is to practice setting up and parameterizing a saturated porous media problem.

FLOW-3D Learning Objectives

In this exercise you will practice techniques to:

1. Select a saturated porous drag model,
2. Compare empirical coefficients to project-specific coefficients,
3. Calculate ***FLOW-3D*** drag coefficients from more common coefficients,
4. Set up a simulation from a list of specifications,
5. Specify a boundary hydrograph so boundary velocity varies with time, and
6. Analyze simulation results and compare them to experimental data.

Review the Physical Experiment

The physical experiment is described in *An Experimental Model for Flow Through Porous Media Using Water Filter*, by D. W. Abbood of Al-mustansiryiah University, Baghdad, Iraq (published in the proceedings of the Thirteenth International Water Technology Conference, 2009, Hurghada, Egypt). The experiment consisted of recirculating clean water through parallel columns of packed media and measuring head loss vs. velocity. Multiple experiments were conducted with varying flow rates (0.001 to 0.020 m/s), varying temperatures (9 °C to 57 °C), varying porosities (0.36 – 0.65), and varying characteristic grain size (0.5 mm to 2.0 mm). Five different media types were studied: crushed sand, crushed coal, glass beads, crushed garnet, and crushed porcilinaite. Measurements were taken from three flow meters and four differential pressure gauges. The media was packed as homogenously as possible using a constant-height-of-fall pluviation method. Porosity for each media was determined from the packed column volume and mass and the known microscopic media density. Several tests were performed to ensure the packed bulk density of the media was repeatable. The experimental setup is shown in Figure 1, below.

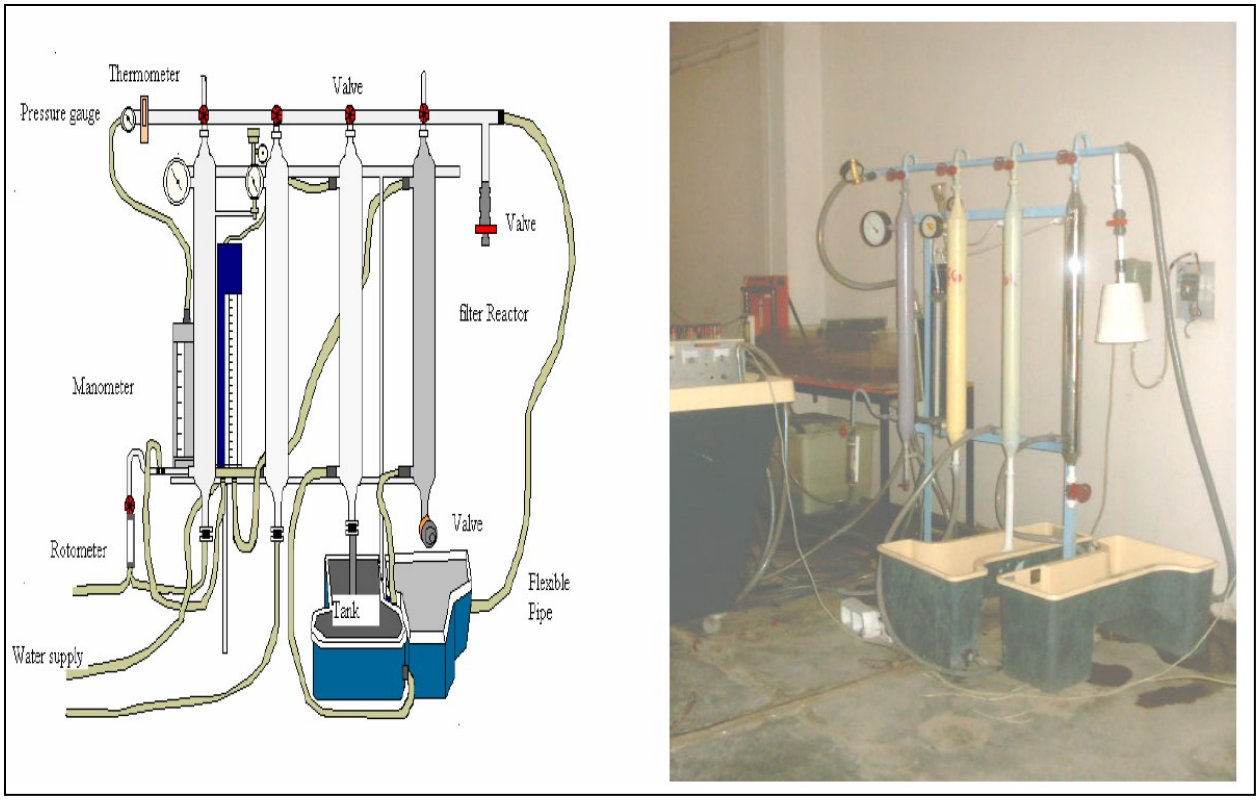


Figure 1: Experimental apparatus from Abood (2009)

Select Model Approach

The media tubes are straight cylinders without bends, so the **Cylindrical coordinate system** can be used. Select the number of dimensions of the model:

- **1-D**
  - Pros: fastest run-time, easiest analysis.
  - Cons: velocity and pressure profile due to inlet/outlet effects and wall effects not modeled.
- **2-D**
  - Pros: fast run-time, easy analysis, inlet/outlet/wall effects can be approximated.
  - Cons: Must assume axially and radially symmetric flow about z-axis.
- **3-D**
  - Pros: inlet/outlet/wall effects and non-axial/radial symmetric flow can be modeled.
  - Cons: slowest run-time, most difficult analysis (requires spatial and temporal averaging).

Based on the above, a **2-D** run in **cylindrical coordinates** is suggested to model axial and radial flow symmetrically around the bed tube axis.

Select Physics Models

Physics models should be considered carefully for all simulations. Here we will check whether gravity, surface tension, viscous flow, turbulence, and porous media physics are needed.

GRAVITY

Because the media vessel is arranged vertically, we might assume that gravity must be included. However, looking at the results indicates the opposite: even at very low flows, there is still a head loss across the media. We can demonstrate this using the *Extended Bernoulli Equation*: if the fluid is incompressible, this relation holds true for any two points 1 and 2 along a flow path:

$$(E_Z + E_V + E_P)_1 + E_A = (E_Z + E_V + E_P)_2 + E_E + E_f$$

Here *E* represents energy. Subscript *Z* is the potential (elevation) energy, *P* is the pressure energy, and *V* is the kinetic (velocity) energy. Subscripts *A* and *E* are energy added or removed by hydraulic machinery, *f* is the energy lost due to friction, and *1* and *2* are two points along a streamline. In this case because the tubes are uniform diameter, the velocity is constant (conservation of mass), and the velocity and hydraulic machinery terms can be omitted. Converting the remaining terms from energy to hydraulic head (in units of feet) gives:

$$\frac{P_1}{\rho g} + z_1 = \frac{P_2}{\rho g} + z_2 + h_f$$

where *P* and *z* are pressure and elevation at point *1* or *2*, ρ is the fluid density, *g* is gravitational acceleration, and h_f is head loss due to friction. Let z_2 be the outlet reference elevation and z_1 be the inlet elevation: they will be 0.4 m apart. Frictional losses will be zero when there is no flow, and the above equation then indicates that the static pressure head at the outlet will be greater than the static pressure head at the inlet by 0.4 m. This implies that at flow rates near zero, the pressure will increase from the inlet to the outlet.

The graphical results reported in the paper show a head *loss* at *all* flow rates. We can (correctly) infer that the researcher has already accounted for static head before reporting the results. The experimental results reported are purely a function of the inlet and outlet pressures in the absence of gravity. They would be equally valid for horizontal flow through the same media. Therefore do not use **Gravity and Non-Inertial Reference Frame physics**.

SURFACE TENSION

The **Surface Tension** physics model is *never active inside* porous media components, since surface tension effects are included empirically in the **Porous Media** physics, including capillary action. **Surface Tension** physics can be used outside of porous media where there are free-surfaces. Since there is no free surface in the model, do not use **Surface Tension** physics.

VISCOSITY AND TURBULENCE

Walls are free-slip surfaces when flow is modeled as inviscid. We should include viscosity at the least so that wall effects are not neglected. **Viscous Flow** physics should be used. ***FLOW-3D*** always assumes that flow in porous media is **Laminar**, but turbulence may be included for free-water regions above and below the media.

Check the assumption that flow in the media will be laminar: as will be calculated below, the pore Reynolds numbers (Reynolds number within a pore of the media) will be less than 40. True turbulence (eddies) in porous media begins at approximately Re_P = 100 to 500. This is a different threshold than *form drag*, the effects of which begin to occur around Re_p = 1. ***FLOW-3D***'s assumption of laminar flow in porous media is valid for this case.

Check whether turbulence occurs in free-water regions: the media-filled pipe is 0.24 m in diameter, and the maximum free-stream velocity is 0.02 m/s. Hand calculation shows that Re_D outside the media = 4,800. Turbulent flow in circular pipes begins at approximately Re_D = 4,000. The flow may be turbulent at the maximum velocity, but will be laminar at lower velocities. Since the flow is delivered to the tubes via small-diameter pipes, it is probably entering the tubes with some turbulence for most flow rates. Activate **viscous flow** and **RNG Turbulence** physics.

Select a Porosity Model

Porous media flow exhibits viscous and inertial drag and may be enhanced or retarded by capillary effects. There are many empirical approaches that predict the velocity, pressure, and fluid fraction of flow within the media. Several of these approaches are incorporated into ***FLOW-3D*** as **Porous Media** physics options. Each approach uses different coefficients and is valid for a specific set of conditions. Because fitting experimental data to empirical equations is a detailed process, it is important to first identify the model which best represents the physical case.

Given that the media is saturated at all times during the experiment, we will disregard the **Unsaturated Flow** model, and the three drag models associated with it.

There are three choices left: the **D'Arcian Saturated Drag** model, **Permeability Dependent Saturated Drag**, or the **Forcheimer Saturated Drag** model. In general, the D'Arcian model applies when pore size is small and flow is relatively slow: Re_P < 1. The permeability and Forcheimer models are different formulations of the same quadratic drag curve. They are valid for all Re_P up to the onset of true turbulence, and fit many experimental data sets. You should decide D'Arcian model and the two quadratic models on the basis of the pore Reynolds number, Re_P:

$$Re_p = \frac{\rho\, U_p\, D_p}{\mu}$$

where ρ is the fluid density, μ is the fluid dynamic viscosity, D_p is the characteristic pore diameter, and U_p is the mean *microscopic (or pore)* velocity of the fluid. Microscopic velocity is almost never measured

directly due to the difficulty of inserting probes between grains. Instead, *macroscopic (or superficial)* velocity U_{macro} is measured as the transit time across the media, neglecting tortuosity. Macroscopic and microscopic velocities are theoretically related by the media *effective porosity*:

$$U_{macro} = U_p \eta_{eff}$$

where η_{eff} is the *average effective porosity*. Effective porosity is the porosity that is available for flow, and excludes pores that cannot be filled or drained. In a completely saturated loose media η_{eff} can be assumed to be the same as *bulk porosity* η (0.50 in this case).

The experimental velocity given in the paper is *superficial velocity* U_{macro} (range 0.001 – 0.020 m/s). Use porosity to convert to macroscopic velocity to microscopic velocity, and then find the pore Reynolds number for each velocity assuming the pore diameter D_p is equal to the average grain diameter (1 mm):

EXPERIMENTAL SUPERFICIAL VELOCITY U_{macro} (m/s)	CALCULATED MICROSCOPIC VELOCITY $U_p = U_{macro}/\eta_{eff}$ (m/s)	CALCULATED PORE REYNOLDS # Re_P (dimless)
0.001	0.002	2.0
0.002	0.004	4.0
0.003	0.006	5.9
0.004	0.008	7.9
0.005	0.010	9.9
0.006	0.012	11.9
0.007	0.014	13.9
0.008	0.016	15.8
0.009	0.018	17.8
0.010	0.020	19.8
0.011	0.022	21.8
0.012	0.024	23.8
0.013	0.026	25.7
0.014	0.028	27.7
0.015	0.030	29.7
0.016	0.032	31.7
0.017	0.034	33.7
0.018	0.036	35.7
0.019	0.038	37.6
0.020	0.040	39.6

Table 1: Calculating microscopic velocity and pore Reynolds number

The pore Reynolds number ranges from 2.0 to 39.6. **The D'Arcian flow regime is strictly speaking only valid to Re_P = 1, but is generally assumed accurate up to Re_P = 10**. Therefore select the **Permeability Dependent Saturated Drag** model, which is accurate for the D'Arcian region and outside of it. The particle size (1 mm) and porosity (0.50) are larger than in the soils D'Arcy studied, so it makes sense that his model does not apply here.

Review the Permeablity-Dependent Saturated Drag Equation

You need to determine two coefficients: **D'Arcian Permeability (*PERM*)** and **Non-D'Arcian Permeability (*PERM2*)** for the porous media component. The equations to derive these values from known data are given below as a formal exercise, based on the equations in the **User Manual**.

$$\frac{\Delta P}{\Delta x} = \frac{\mu}{K_1} U_{macro} + \frac{\rho}{K_2} U_{macro}^2$$

where K_1 is the media's experimental *D'Arcian permeability* and K_2 is the coefficient of *non-D'Arcian permeability*. $\Delta P/\Delta x$ is the pressure loss per unit distance traveled superficially, ρ is fluid density, and μ is the fluid dynamic viscosity. Of the three saturated porous drag models, only **Permeability-Dependent Saturated Drag** is written in terms of *macroscopic* velocity U_{macro}; the rest use pore velocity U_P.

To use the relationship above, the experimental *head loss ΔH* needs to be converted to *pressure loss ΔP*. They are related by density ρ and gravity g:

$$\Delta P = \Delta H \rho g$$

Figure 2 and Table 2, below, show the experimental head loss data ($\Delta H/\Delta x$), digitized using ***EnGauge*** freeware, and the calculation of unit pressure loss $(\Delta P/\Delta x)$ for use in the coefficient fitting equations.

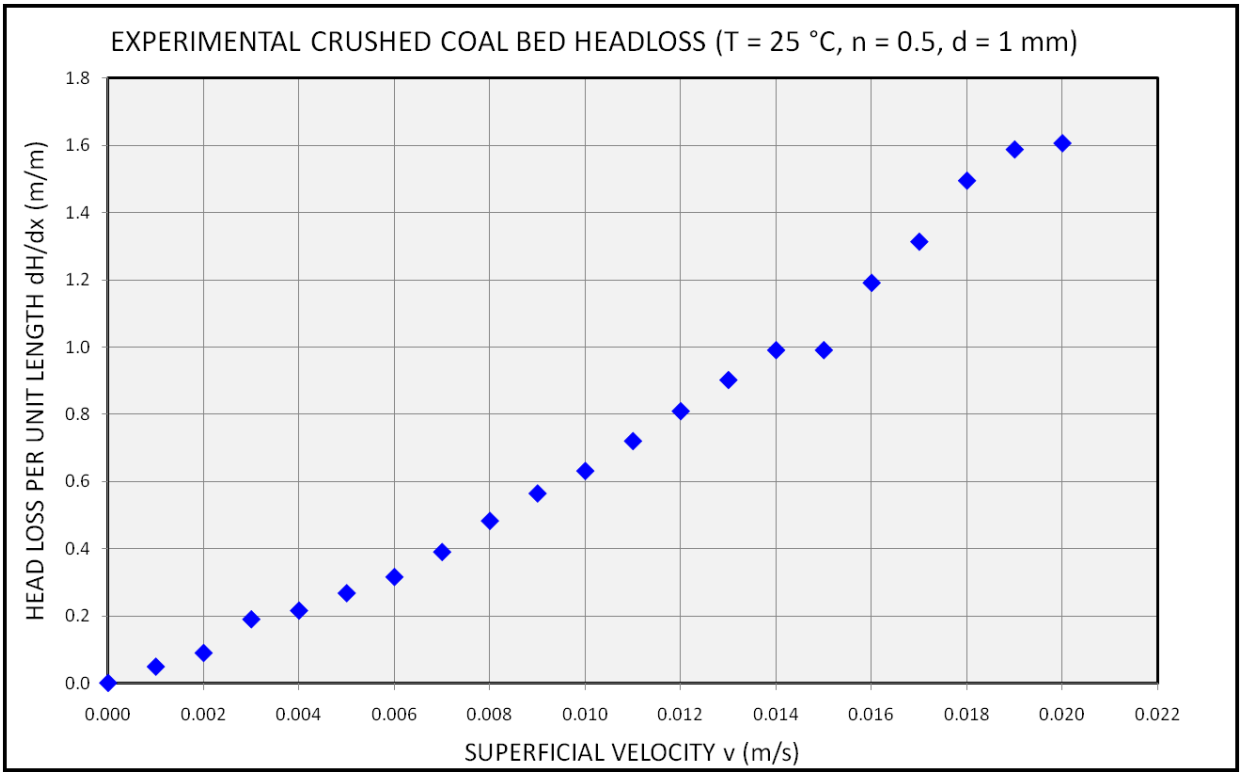


Figure 2: Experimental head loss vs. superficial velocity

SUPER. VELOCITY U_{macro} (m/s)	MICRO. VELOCITY $U_p = U_{macro}/\eta_{eff}$ (m/s)	PORE REYNOLDS Re_P (dimless)	HEAD LOSS $(\Delta H/\Delta x)$ (m/m)	PRESS. LOSS $(\Delta P/\Delta x)$ (Pa/m)
0.000	0.000	0.0	0.000	0
0.001	0.002	2.0	0.048	472
0.002	0.004	4.0	0.089	872
0.003	0.006	5.9	0.189	1,853
0.004	0.008	7.9	0.215	2,108
0.005	0.010	9.9	0.267	2,617
0.006	0.012	11.9	0.315	3,089
0.007	0.014	13.9	0.390	3,816
0.008	0.016	15.8	0.482	4,725
0.009	0.018	17.8	0.564	5,524
0.010	0.020	19.8	0.631	6,178
0.011	0.022	21.8	0.720	7,050
0.012	0.024	23.8	0.809	7,923
0.013	0.026	25.7	0.902	8,831
0.014	0.028	27.7	0.991	9,704
0.015	0.030	29.7	0.991	9,704
0.016	0.032	31.7	1.191	11,666
0.017	0.034	33.7	1.314	12,865
0.018	0.036	35.7	1.496	14,646
0.019	0.038	37.6	1.588	15,555
0.020	0.040	39.6	1.607	15,736

Table 2: Experimental Head Loss Converted to Pressure Loss

The pore velocity U_P and the experimental pressure loss $(\Delta P/\Delta x)$ is now known and the permeability coefficients can be found.

Find Drag Coefficients

With the correlated variables known, there are two ways to find the drag coefficients:

- Fit literature drag coefficients from empirical relations to the superficial equation, or
- Fit case-specific experimental data to the superficial equation.

The second option is always a better choice when case-specific data exists. Compare the coefficients from both options:

Empirical Correlations Without Experimental Data

The Ergun (1952) equation for packed spheres and a relation for crushed angular materials (sand) by Idelchik (1986) are plotted below along with the experimental data.

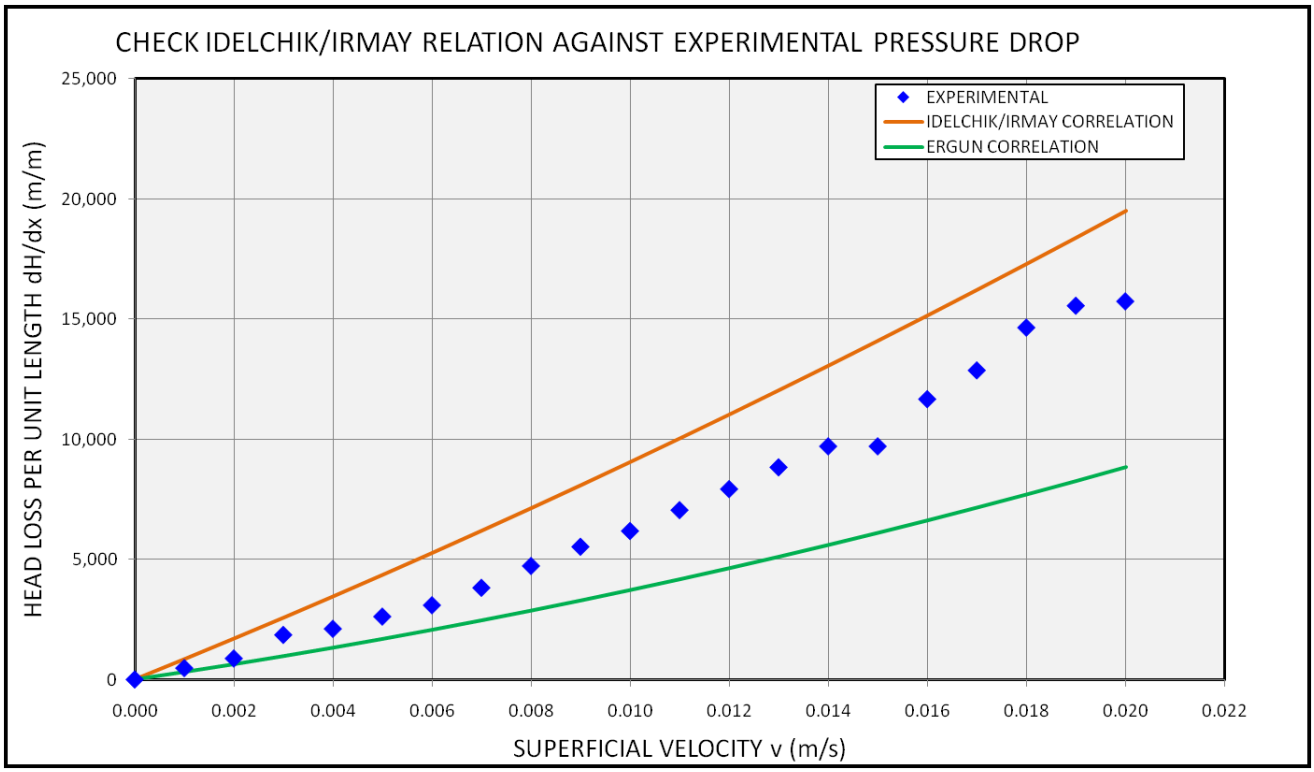


Figure 3: Empirical correlations are not recommended

The chart shows neither the Idelchik correlation nor the Ergun equation is a good fit. An historic overview of empirical correlations with examples are given in the source paper (Abood 2009). The purpose of this step was to demonstrate the poor fit of most empirical correlations. Now you will fit the experimental data directly.

Curve-Fitting Experimental Data

The drag model equation is given below in simplified form for fitting experimental data:

$$\left(\frac{\Delta P}{\Delta x}\right) = A\, U_{macro} + B\, U_{macro}^2$$

A and *B* are coefficients fitted to the experimental data. Use a statistics package or a spreadsheet program to fit a 2nd order polynomial 'trend line' to the experimental data. This gives a well-fit correlation as shown below.

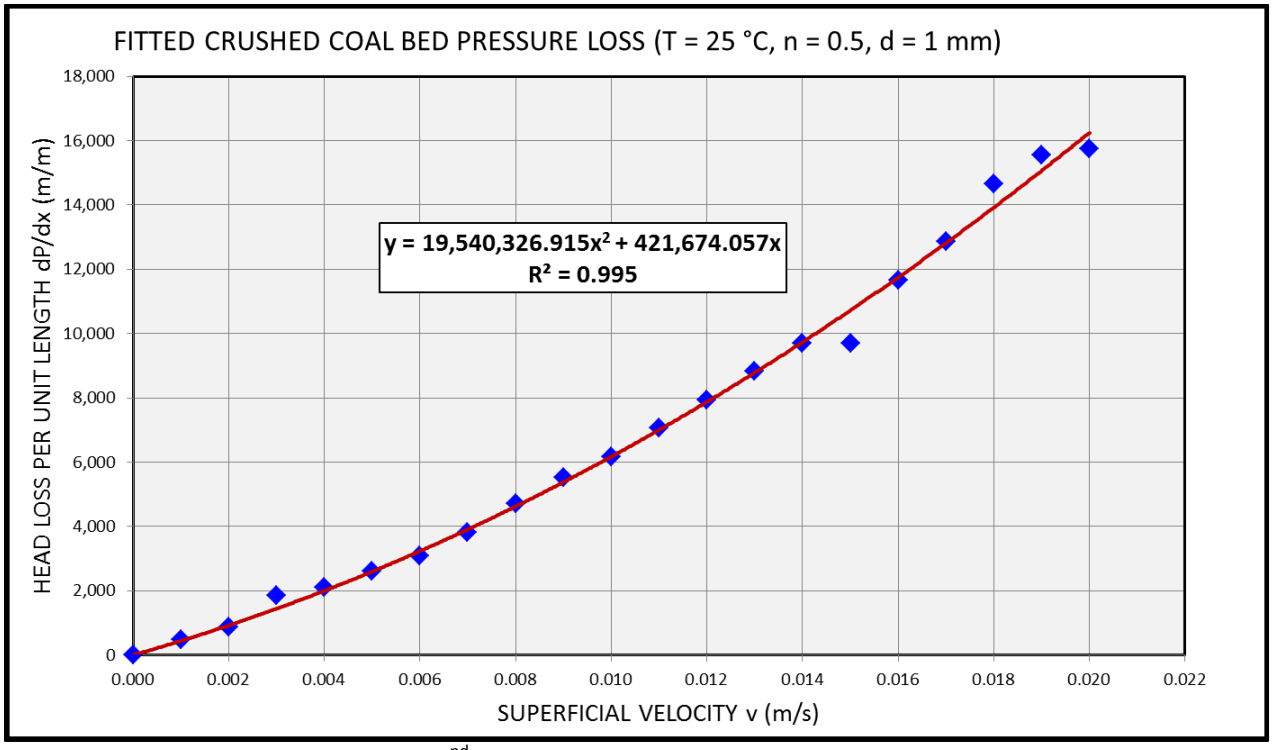


Figure 4: Fit a 2nd-order polynomial to the experimental data

This is obviously a better match than generalized correlations. From the fitted equation, we see that

***A* = 421,674**

***B* = 19,540,327**

Compare the fitting equation to the full equation with **K_1 (*PERM*)** and **K_2 (*PERM2*)**:

$$\left(\frac{\Delta P}{\Delta x}\right) = A\, U_{macro} + B\, U_{macro}^2 = \frac{\mu}{K_1} U_{macro} + \frac{\rho}{K_2} U_{macro}^2$$

And then solve for K_1 and K_2 using the known fluid density and viscosity:

$K_1 = \mu/A$ = 2.3715E-9

$K_2 = \rho/B$ = 5.1084E-5

Problem Specification

Set up the problem following the model specifications below:

General and Numeric Tab Options

- **Simulation name: Saturated Porous**
- **Simulation units UNITS = SI**
- **Simulation time TWFIN = 3598 seconds**
- **Number of processors NPROC = 2**
- **Initial time step DELT = 0.001 sec**
- **Minimum number of pressure iterations ITMIN = 2**
- **Maximum number of pressure iterations ITMAX = 500**
- **GMRES subspace size MRSTRT = 25**

Physics

- **Porous Media - Permeability Dependent Saturated Drag**
- **Viscosity & Turbulence**
  - **Viscous Flow**
  - **RNG Turbulence with Dynamically-Computed TLEN**

Fluid Properties (Water, measured experimentally @ 24 – 26 °C)

- **Density ρ = 998.2 kg/m^3**
- **Viscosity μ = 0.001008 Pa-s (i.e. kg/m/s)**
- **Compressibility = 1/(ρ W^2) = 4.55E-10** (assume speed of sound W ≈ 1483 m/s)

Output

- **Selected data at default 1% time intervals**
  - **Drag Function**
  - **Fluid Velocities**
  - **Pressure**

Geometry

- **1 Subcomponent**
  - **Shape = Cylinder**
  - **Name = Packed Bed**
  - **Radius = 0.12 m**
  - **Z Low = 0.0 m**
  - **Z High = 0.3 m**
  - **Axis along origin**
- **1 Component**
  - **Name = Packed Bed**
  - **Component Type = Porous**
- **Properties (Component Properties > Porous Properties)**
  - **Porosity = 0.5**
  - **Darcian Permeability = 2.3715E-9**
  - **Non-Darcian Permeability = 5.1084E-5**

Problem specification is continued on the next page:

Mesh Specifications

- **Cylindrical** (right-click on **Mesh-Cartesian** and select **Change to Cylindrical Coordinates**)
- **2-D Mesh Block**
- **Domain**
  - **X (r) = 0 m** to **0.12 m ---------- # Total Cells = 5**
  - **Y(θ) = 0 degrees** to **30 degrees --- # Total Cells = 1**
  - **Z (z) = -0.012 m** to **0.312 m -------- # Total Cells = 20**

Boundary Conditions

- **X Min: Symmetry (S)** (r = 0 is a *discontinuity*, and should always be symmetry)
- **X Max: Wall (W)**
- **Y Min & Y Max: Periodic (Pd)** (flow can rotate out one side and back in the other)
- **Z Min: Gauge Stagnation Pressure (P)**
  - **F = 1**
  - **P = 0 Pa**
- **Z Max: Velocity (V)**
  - **varies with time: import velocities & times from a *.csv* file** (described below)

Initial Conditions

- **Global > Fluid Initialization = Use Fluid Elevation**.
- **Initial Fluid Elevation = 0.312** m.

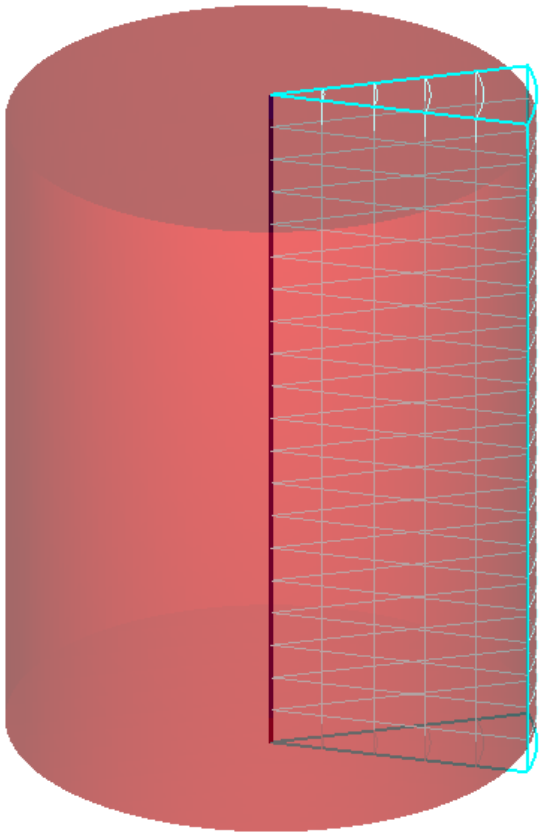


Figure 5: Check the mesh, geometry, and boundaries

Specify a Time-Dependent Velocity Boundary

The time for a fluid packet to pass through the simulation can be found from the domain length and the superficial velocity of the fluid ($t = L/U_{macro}$). This is the *front time* of the velocity profile. The pressure field also needs time to come to a steady-state solution, so the front time should be multiplied by a safety factor. Assuming a 2x safety factor for pressure, the table of time and velocity values looks like this:

POROUS MEDIA LENGTH:	0.300	m
MODEL DOMAIN LENGTH:	0.324	m
STEADY STATE TIME MULTIPLE:	2	

SUPER. VELOCITY U_{macro} (m/s)	FRONT TRANSIT t_{front} (s)	EST. STDY STATE $t_{steady,est.}$ (s)	TOTAL TIME t_{total} (s)
0.020			0
0.020	25	50	50
0.019	26	53	103
0.018	28	56	158
0.017	29	59	217
0.016	31	63	280
0.015	33	67	346
0.014	36	71	418
0.013	38	77	495
0.012	42	83	578
0.011	45	91	669
0.010	50	100	769
0.009	56	111	880
0.008	63	125	1,005
0.007	71	143	1,148
0.006	83	167	1,314
0.005	100	200	1,514
0.004	125	250	1,764
0.003	167	333	2,098
0.002	250	500	2,598
0.001	500	1,000	3,598

Table 3: Determining runtime of each velocity for the boundary profile

This is the basis for the **Finish Time** that was given above. ***FLOW-3D*** interpolates linearly between tabular data points. Given the above table, ***FLOW-3D*** will generate an inflow velocity profile that looks like Figure 6 below:

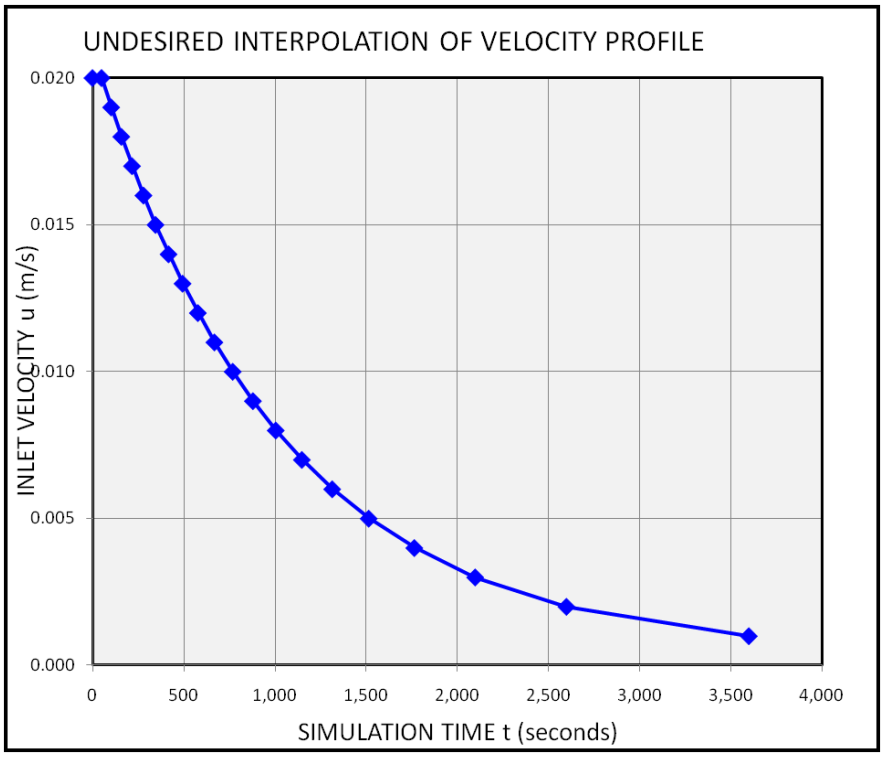


Figure 6: Incorrect interpolation of velocity profile

This is not desired! *The model should match the experiment*. To do this, each velocity must be held constant long enough for the results to become steady. Adding two points per velocity gives us the desired profile, which looks like this:

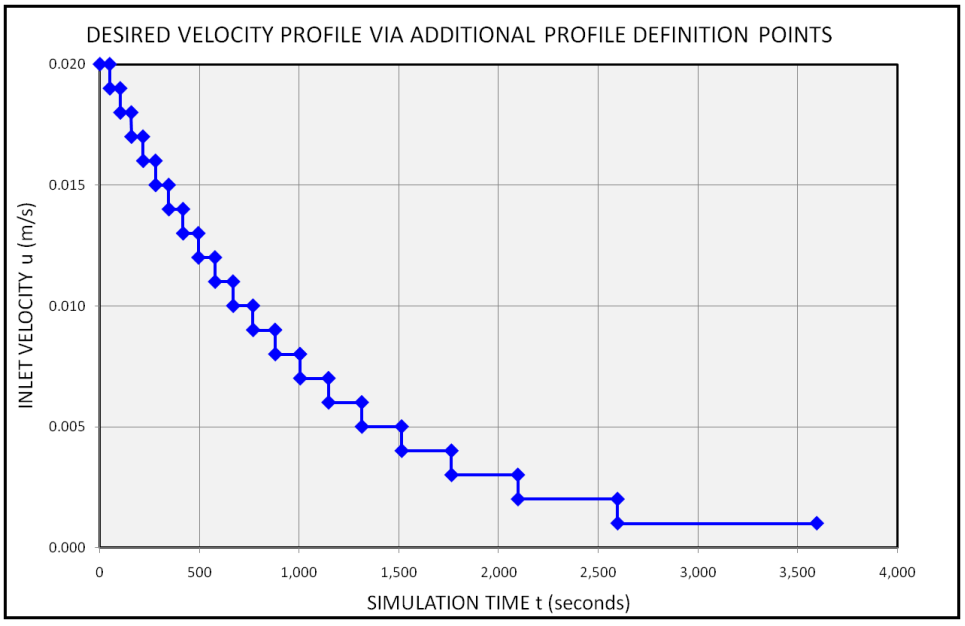


Figure 7: Desired velocity profile

At each step in the velocity profile, the elapsed time between the lower step and the upper step is arbitrarily selected to be one second, since ***FLOW-3D*** requires that each boundary condition time step

have only one associated value. The steps will cause *rapid change* at the inlet *boundary condition*, a *known cause* of pressure problems! This is why increased *pressure solver overhead* was specified above.

Generate a table in a spreadsheet program, starting with the fastest velocity and working down, with a header row, and save it as a *.csv* file. Note the negative values mean the velocity vector is pointing toward the z-min boundary.

TIME	W
0	-0.020
49	-0.020
50	-0.019
102	-0.019
103	-0.018
157	-0.018
158	-0.017
216	-0.017
217	-0.016
279	-0.016
280	-0.015
345	-0.015
346	-0.014
417	-0.014
418	-0.013
494	-0.013
495	-0.012
577	-0.012
578	-0.011
668	-0.011
669	-0.010
768	-0.010
769	-0.009
879	-0.009
880	-0.008
1,004	-0.008
1,005	-0.007
1,147	-0.007
1,148	-0.006
1,313	-0.006
1,314	-0.005
1,513	-0.005
1,514	-0.004
1,763	-0.004
1,764	-0.003
2,097	-0.003
2,098	-0.002
2,597	-0.002
2,598	-0.001
3,597	-0.001

Table 4: Time-step/velocity entries saved as *.csv* file

Import the table as follows:

1. Copy ***BoundaryVelocity.csv*** to the project folder (**Saturated_Porous**).
2. Open the **Z-Max** boundary condition.
3. Select **Specified Velocity** as the **Boundary Type**.
4. Specify **Fluid Fraction = 1**.
5. Click on the **Velocity** button above the X, Y, and Z velocity components.
6. Select **Import to Column = W** in the lower right of the dialog.
7. Check **Import from Simulation Directory** to tell the interface where to start browsing.
8. Click **Import Values**.
9. Browse if necessary and open the ***BoundaryVelocities.csv*** file.
10. Click the header **W** to select the row to import (there is only one row in this file), and then **OK**.
11. You have the options of **Import Tables to Prepin** or **Link External CSV Data to Prepin**. The first option will copy the data into the prepin file so the .csv file is no longer needed, but *limits the number of points per table to 500*. The second option adds a link path to the prepin file to tell it where to look: the .csv file is required to *stay with* the prepin file, but can have *more than 500* points. For this case, select **Import Tables to Prepin**.
12. Check that the **W** column is populated, and that the chart appears as shown below.
13. Click **OK** to get out of the Boundary Dialog and **Save** your work.

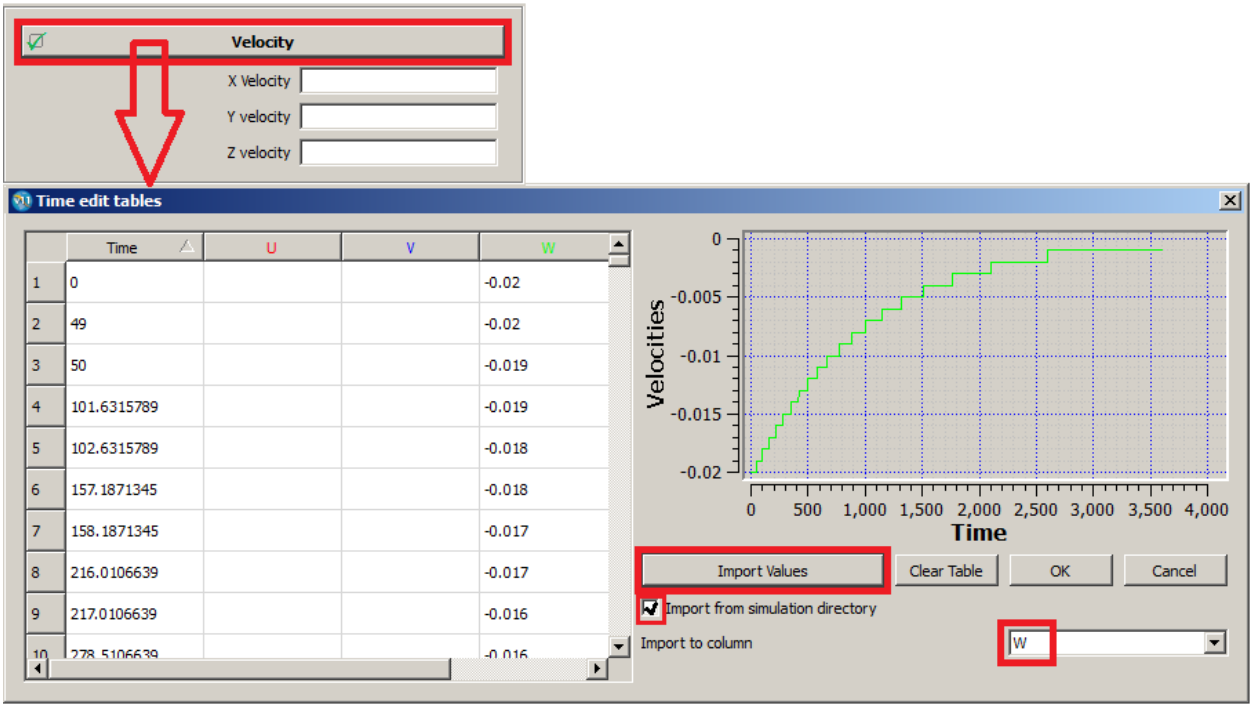


Figure 8: Importing velocities from a *.csv* file

You could also enter all these values manually in the table, but importing from a *.csv* file is much easier.

Run the Simulation and Review Runtime Output

The model is completely set up, so go ahead and run it.

1. Review **Solver Warnings**  . There are a few messages that the simulation is nearly steady. They occur between velocity changes at the boundary, implying that the duration of the steps is sufficient to get steady flow between velocity changes.

2. Select the **Mass-Averaged Mean Kinetic Energy** runtime plot and note how the kinetic energy relates to the magnitude of the velocity profile.

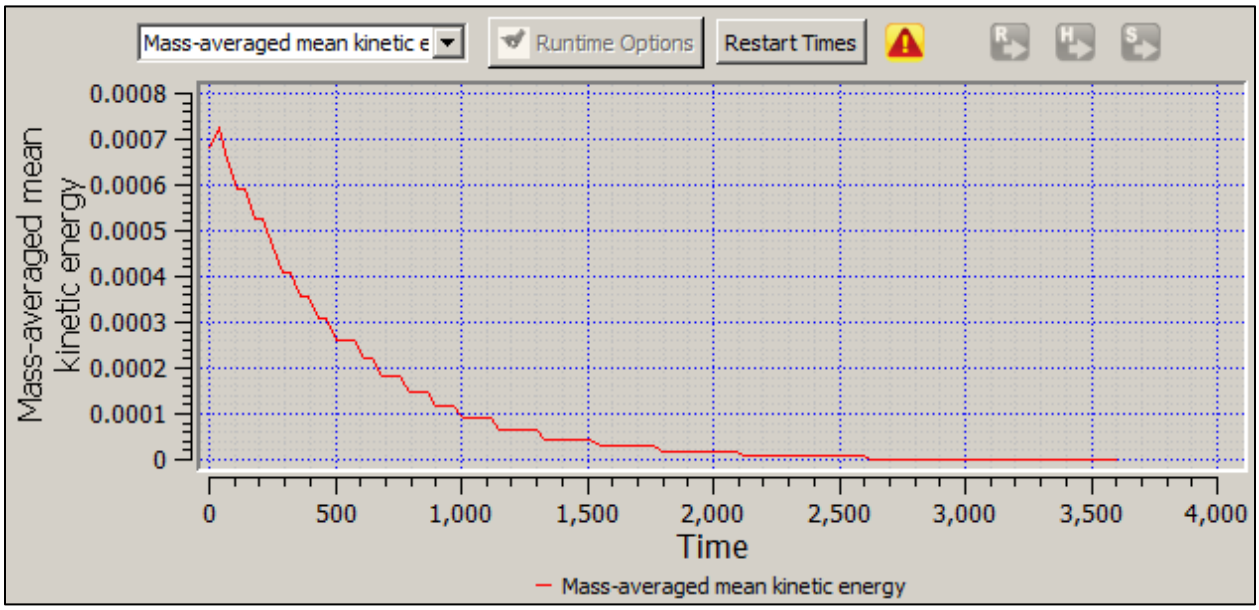


Figure 9: MKE relates to velocity

3. Now take a look at the **Pressure Iteration Count** plot. The most iterations reported is 2 per cycle: increasing the maximum number of pressure iterations permitted **ITMAX = 500** was unnecessary. Notice that even though the minimum number of iterations was specified **ITMIN = 2**, the solver does not always iterate twice.

Analyze 2-D Results

1. Using what you already know, **Open** the results file and analyze the flow in **2-D**.
2. Plot **Pressure** and **Plain Velocity Vectors**.
3. Turn off **Contour Smoothing**.
4. Make sure you're plotting with **SI units**.
5. Check your work against the final time step shown below.

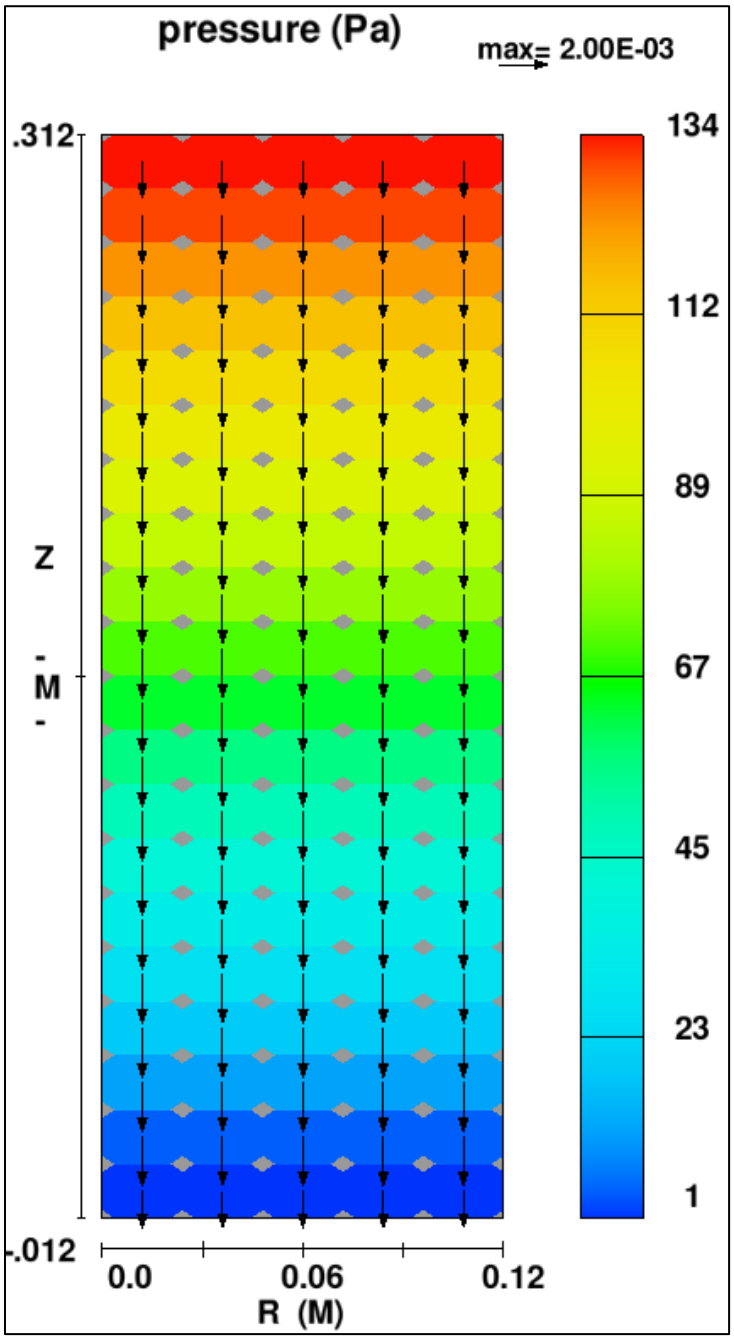


Figure 10: Pressure drop (Pa) across porous media at t = 3598 sec

Analyze Results using Probe

1. Using what you know, plot **Probe > Selected** data for **Pressure** and **Z-Velocity**.
2. Plot results at the center of the top (inlet) of the cylinder (**I = 2**, **K = 22**).

The rest of the analysis is illustrated below. The steps are:

1. The *pressure loss ΔP* across the media is determined by subtracting the outlet pressure (**I = 2, K = 2**) from the inlet pressure (**I = 2, K = 22**) at each output time. A more thorough approach would spatially average the inlet and output pressures, but this step is skipped here.
2. Divide *ΔP* by the length of the porous media (0.3 m) to find the *unit loss ΔP/Δx*.
3. Let U_{macro} = the *average* of the inlet and outlet velocities at each output time.
4. Convert *pressure loss ΔP/Δx* to *head loss ΔH/Δx*.
5. Plot the unit head losses against their matching average velocities.
6. Calculate and plot 2nd-order polynomial fitting equations for both sets of data.
7. Compare the results to the physical experiment, as shown below.

The model output matches the experimental data with a high degree of accuracy. Further statistical error correlations can be developed from the point data and from the polynomial fitting curves. The exercise is now complete. If you have any questions, please ask one of your instructors.

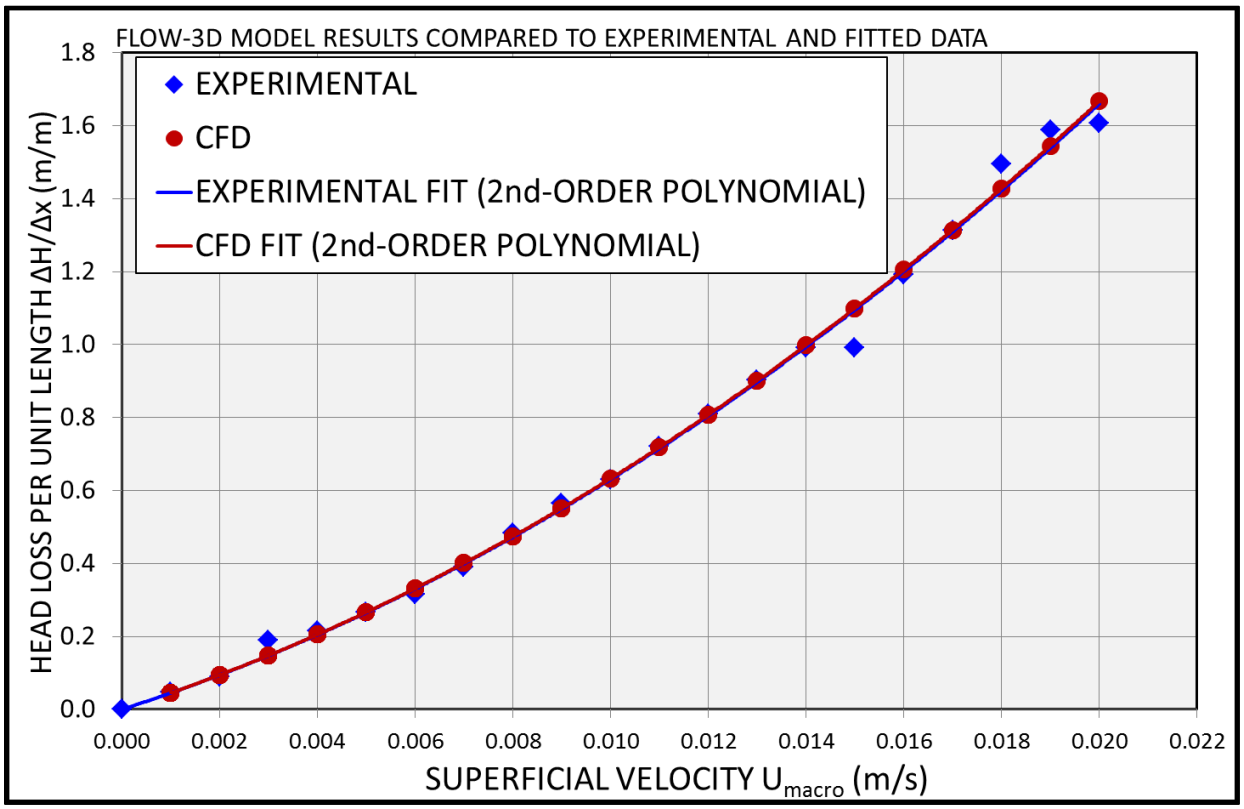


Figure 11: Agreement between experiment and CFD

References

- Abbood, D.W. 2009. *An Experimental Model for Flow Through Porous Media Using Water Filter*. Proc. 13th International Water Technology Conference: Hurghada, Egypt.
- Bejan, A. & D. Nield. 2006. *Convection in Porous Media*, 3rd Ed. New York, NY: Springer Science+Business Media.
- Ergun, S. 1952. *Fluid flow through packed columns*. Chem Engrg. Prog. 48, 89-94.
- Flow Science. 2009. FLOW-3D v9.4 User Manual. Santa Fe, NM: Flow Science Inc.
- Freeze, R.A. & J.A. Cherry. 1979. *Groundwater*. Englewood Cliffs, NJ: Prentice-Hall.
- Idelchik, I.E. 1986. *Handbook of Hydraulic Resistance*. New York, NY: Hemisphere Publishing Corp.
- Irmay, S. 1958. *On the theoretical derivation of Darcy and Forchheimer formulas*. Eos, Trans. AGU 39, 702-707.
- Masch, F.D. & K.J. Denny. 1966. *Grain size distribution and its effects on the permeability of unconsolidated sands*. Water Resour. Res. 2, 665-677.

Exercise 9: STL Topography from LiDAR

CREATING STL FILES FROM TOPOGRAPHIC POINT CLOUDS USING TOPO2STL AND FREEWARE

INTRODUCTION

Topography can be defined as "the configuration of a surface including its relief and the position of its natural and man-made features". It can also be defined as "the art or practice of graphic delineation in detail … of natural and man-made features of a place" (Merriam-Webster 2013). Digitizing topographic data for use in ***FLOW-3D*** combines both definitions: the process of recreating accurate, detailed surface features requires both art and practice.

Point clouds are one of the most common ways that topographic data is recorded. Each point is a measurement of surface location, defined by three coordinates in some reference system: latitude/longitude/elevation, for example. The raw point data may be digitized from contour maps, exported from GIS (geographical information system) software, or may be the output of LiDAR (Light Detection and Ranging), TLS (terrestrial laser scanning), or satellite profiling equipment.

Stereolithography file format (*.stl*) is a non-proprietary system for recording 3-D surfaces. The format is raw and unstructured, meaning that the elements of the surface can be recorded in any order. The surface elements are triangles (called facets) composed of three points and a normal vector pointing away from the surface. The points and vectors are recorded in Cartesian coordinates. The *.stl* file format may be either binary (smaller file size) or ASCII (readable in text editors); see Appendix A for the format definitions. ***FLOW-3D*** uses *.stl* files as inputs to define complex geometry and as outputs to describe resolved fluid and solid surfaces.

This exercise introduces ***FLOW-3D***'s ***TOPO2STL*** utility for creating *.stl* parts from large point clouds. No single program is ideal for converting large surface point sets to 3-D geometry, so a method is presented here that combines freeware utilities with ***TOPO2STL*** to get excellent results. Freeware tools introduced in this exercise include ***VIM***, ***MeshLab***, ***CloudCompare***, ***Admesh***, and ***netfabb Basic***.

OVERVIEW OF THE BASIC PROCESS

The steps to convert a point cloud of XYZ data to *.stl* are as follows:

1) Precondition the data for use with ***TOPO2STL***:

a. Create a space-delimited ASCII text file with one coordinate per line and no headers. Each line must contain three numbers, separated by spaces, so that the first number is the X coordinate, the second number is the Y coordinate, and the third number is the Z (elevation) coordinate. The coordinate units must all be the same (e.g., feet or meters).

b. Visualize and check the data. The first test of the raw data is visual examination. Look for outlying points, strange stretching (indicating that units are not the same), noise, and other problems. Locate the minimum and maximum extents of the raw data.

c. Clean the data by manually removing artifacts. The most common artifacts are people and equipment that are picked up by the recorder and outlying points that are added inadvertently (usually when filtering the data with software).

d. Fill in surface elevations for absent data. Many point sets only include data in a region of interest (like a river bed). The areas outside of this region are "blank": there is no data there. These outlying regions should be filled in with a uniform surface elevation, usually one that is higher than the any of the known data. This optional step is highly recommended because it speeds up the *.stl* conversion by a factor of 100-plus, and because it produces better output with fewer artifacts (Figure 1 below).

e. Pre-decimate (subsample) the point data. This is an optional step that reduces the number of points in the data set. The benefits of pre-decimating include more accurate *.stl* surfaces, faster conversion times, and more reliable results when using software post-condition the data. ***TOPO2STL*** also subsamples the data during the conversion, so the purpose of this step is to prepare the data by making it more uniform. It's important to use subsampling algorithms that preserve the data integrity (i.e., that don't create new points) and that evenly distribute points (e.g., that don't cause higher point density near contour lines).

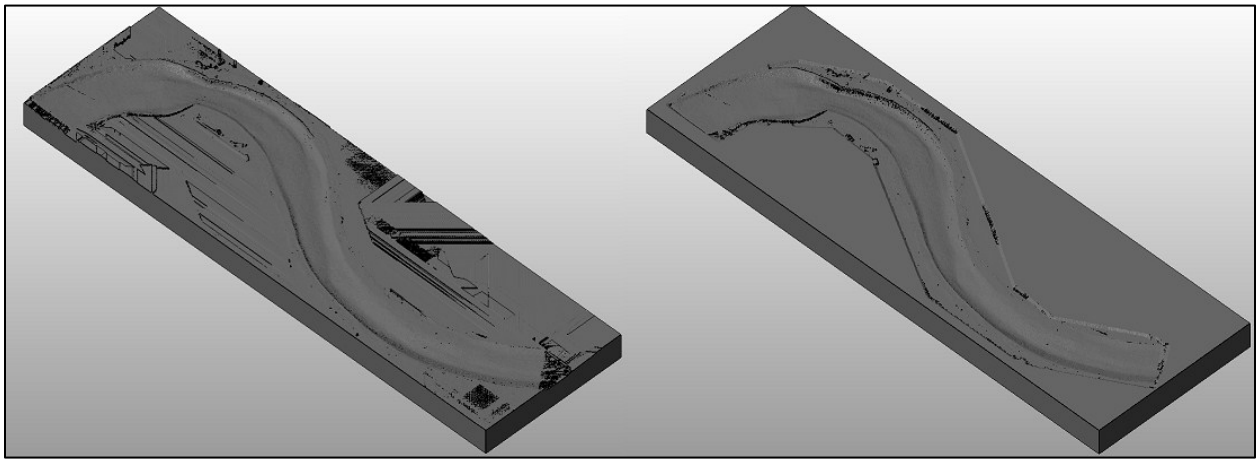


Figure 1. 1-inch resolution *.stls* with (right) and without (left) data pre-conditioning

2) Use ***TOPO2STL*** to convert the point surface to a 3-D *.stl* file:

    a. Select spatial resolution. ***TOPO2STL*** generates a rectangular sampling grid around the minimum and maximum horizontal extents of the point cloud. The grid is made of square stencils of the specified side length. Each grid cell retains a single coordinate point out of all the original points in that cell; the points that are not retained are thrown away and do not appear in the *.stl* output. This is one method of sampling (decimation). Cells that do not have any original points in them are assigned a coordinate based on the elevations of the points in nearby cells. The end result is a more-or-less rectangular grid of coordinate points that are called vertices.

    b. Select z-minimum elevation. ***TOPO2STL*** generates a network of triangular facets using the vertices as corners. This creates the upper surface of the *.stl* part. The lower surface of the *.stl* part is created as a flat plane at the specified z-minimum elevation. The *.stl* is closed by stitching the upper and lower surfaces together with more triangular facets.

3) Post-condition the *.stl* file:

    a. Simplify unnecessary regions to reduce file size. ***TOPO2STL*** creates *.stl* facets that are about the same size, regardless of location. In regions where there wasn't any data in the original point cloud, this adds unnecessary size to the file. Recombining facets into larger ones only where no data originally existed is the recommended way to reduce the final *.stl* size.

    b. Convert the *.stl* to binary format. Binary format *.stl* file size is about 20% of ASCII format. Converting the ASCII ***TOPO2STL*** output to binary format is recommended.

    c. Repair the part. ***TOPO2STL*** and other converters create artifacts in the output. Some of the most common artifacts are small holes, disconnected edges, triangles with bad aspect ratios, and inverted facets. These must be fixed before using the *.stl* file in ***FLOW-3D*** or unrealistic results might occur near the bad surfaces.

4) Quality control: Compare the *.stl* file surface to the original point cloud data to verify the accuracy and precision of the conversion. Statistical measures that will be demonstrated in this exercise include the mean error (ME), mean absolute error (MAE), signed and unsigned minimum and maximum errors (E_{MIN}, E_{MAX}, $|E_{MIN}|$, $|E_{MAX}|$), standard deviations (σ_E and $\sigma_{|E|}$), and chi-squared tests of continuous probability functions (χ^2_{NORMAL} and $\chi^2_{WEIBULL}$).

EXAMPLE DATA: RIO GRANDE RIVER SCALE MODEL

The example point cloud file (*MiddleRioGrande.txt*, see Figure 2 below) can be downloaded from the ***FLOW-3D*** Users Site at users.flow3d.com. The data is from a joint project of the U.S. Bureau of Reclamation and Colorado State University, and is provided as a courtesy by Dr. Amanda Cox. Measurements were taken via ground-based LiDAR of a scale model of the Middle Rio Grande River in New Mexico. The physical model was built at the Colorado State University Engineering Research Center. The measurement units are feet.

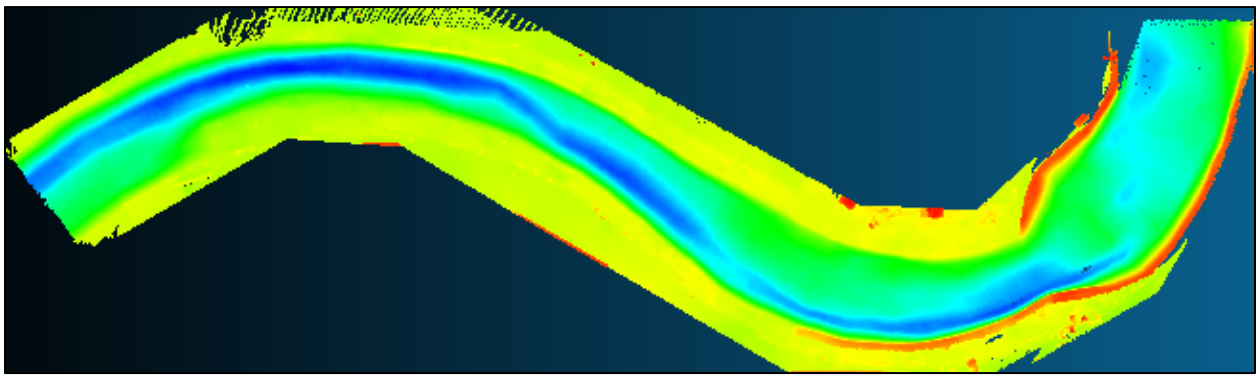


Figure 2. Scale model LiDAR data

DETAILED EXAMPLE OF THE BASIC PROCESS

We'll follow the basic process outlined above. Several ways of accomplishing each step will be given. The workflows given in this note are only one way of approaching the solution. At a number of points in the process, you'll be asked to save your work with a different file name. Saving your work with different file names is important. It helps you keep track of your progress, revert to earlier data states, and compare different stages and approaches.

1) **Format the text file**. The original file won't load in ***TOPO2STL*** because it is not formatted correctly. The file size is 445 MB: too large to work with in most text editors. Two ways to clean and check the data are given here.

    a. Use ***CloudCompare*** to import and re-export the data: ***CloudCompare*** is a useful free tool for working with large point cloud files. It does not yet include as many features as ***MeshLab***, but the features it includes are well-implemented and easy to use. It can be downloaded from www.danielgm.net/cc/.

    Open ***CloudCompare*** and select **File > Open**. Set **File Type = ASCII** and select *MiddleRioGrande.txt*. Set **Skip Lines = 2** to ignore the headers, and then set the three column labels at the top to **coord. X**, **coord. Y**, and **coord. Z**. The column labels are a useful feature when the raw data is not in the order ***TOPO2STL*** expects. Select **OK** to begin the import. When the file is done loading, select **MiddleRioGrande – Cloud** in the *DB Tree* pane at the upper left: this displays some data about the cloud in the *Properties*

panel. Note there that the file has 15,010,488 points. **Save** the highlighted cloud as *MiddleRioGrande_CCtxt.xyz*.

b. Use ***MeshLab*** to import and re-export the data: ***MeshLab*** is a powerful and free open-source tool for working with meshes and point clouds. There are many ways to use ***MeshLab*** with point clouds and *.stl* files, a few of which will be detailed below. ***MeshLab*** can be downloaded from meshlab.sourceforge.net. It is not as user-friendly as ***CloudCompare***, and "Mr. P's" tutorials (www.youtube.com/user/mrpmeshlabtutorials) are highly recommended when you are ready to learn more.

First make a copy of *MiddleRioGrande.txt* and rename it *MiddleRioGrande.asc*. Open ***MeshLab*** and select **File > Import Mesh**. Set the file format type to **ASC** *(ascii triplets of points)* and open *MiddleRioGrande.asc*. Set **Header Row to be Skipped = 2** (per what you saw in ***CloudCompare***) and *deselect* **Grid Triangulation**, because this import filter won't work with non-uniform point distributions. Select **OK** to begin importing the data. The progress bar at the bottom will fill several times as various importation steps are completed. Note at the bottom of the screen that the number of vertices (points) is 15,010,488, as before.

Importing the data as file type *.asc* works correctly but importing as *.xyz* type produces an incorrect extra point at coordinates 0, 0, 0 at the beginning of the file. This is because the *.xyz* format cannot contain headers and there is no option to skip initial lines. ***MeshLab*** has a lot of little quirks like this.

Select **File > Export Mesh As**, set **Files of Type = XYZ Point Cloud**, and name the file *MiddleRioGrande_MLASCtxt.xyz*. Click **Save** and on the next dialog *deselect* **Normal** so that the file saves with only three coordinate values per line (instead of six).

All algorithms have the potential to produce noise or error in the results. Converting the data with several methods and comparing the results is recommended. Comparing the files:

CLEANING METHOD (OUTPUT IS *.XYZ* FORMAT)	NUMBER OF POINTS (15,010,488 IN ORIGINAL)	DIFFERENCE IN COORDINATE VALUES FROM ORIGINALS
CloudCompare *.txt* import	15,010,488	+/- 3E-6 feet (< 1 micrometer, µm)
MeshLab *.asc* import	15,010,488	+/- 2E-6 feet (< 1 micrometer, µm) (excluding the extra point)
MeshLab *.xyz* import	15,010,48**9** **(adds point at 0,0,0)**	

Table 1. Comparison of cleaning methods for *MiddleRioGrande.txt*

The precision loss is due to rounding of large numbers. Few data sets require micrometer precision, so the formatting method will usually depend on factors like file size, data column order, and type(s) of bad values that must be removed. This particular data set contains a highly precise measurement record. Though it will have negligible effects on modeling results, the

cleaned file with the least introduced error should be used. The remainder of the exercise will use *MiddleRioGrande_MLASCtxt.xyz*.

2) **Visualize and check the data**. The next step is checking that the topography looks right, locating any bad data points, and making note of the minimum and maximum data extents. Two methods are introduced.

   a. Open and view the point cloud in ***MeshLab***. Select **File > Import Mesh** and load *MiddleRioGrande_MLASCtxt.xyz*. Select **Render > Render Mode > Points** from the menus or  on the *Render* toolbar to display the points.

   b. Color the points by elevation. Select **Filters > Quality Measure and Computations > Per Vertex Quality Function**. Set **func q = -z** (you may have to use the *Tab* key to gain access to the input box), and activate **map into color**. Click **Apply**. Select **Render > Color > Per Vertex** from the menu bar to display the colored results.

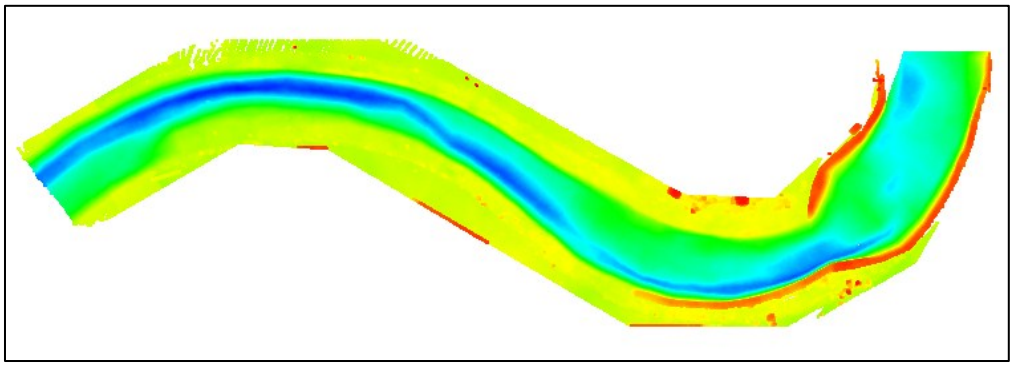


Figure 3. Points colored by elevation from blue (low) to red (high)

   c. Find approximate min/max coordinates. Select **Filters > Color Creation and Processing > Colorize by Vertex Quality**. The min and max values reported are the inverse of the min and max z-elevations (97.2 and 99.3 feet) because the quality was set to the negative of z-elevation. Changing the min and max values manually equalizes the color, which will be useful later. Click **Cancel**.

   d. Find precise min/max coordinates. Open the data in ***CloudCompare***. Select **Tools > Projection > Export Coordinate(s) to Scalar Field(s)** and export **X**, **Y**, and **Z**. Select the coordinate to view under *Properties > Scalar Fields* in the lower left: the numeric coordinate values are given as the slider limits of the *displayed SF Scale* (see Figure 4 below). Check again for outlying points and record the precise min/max values (Table 2).

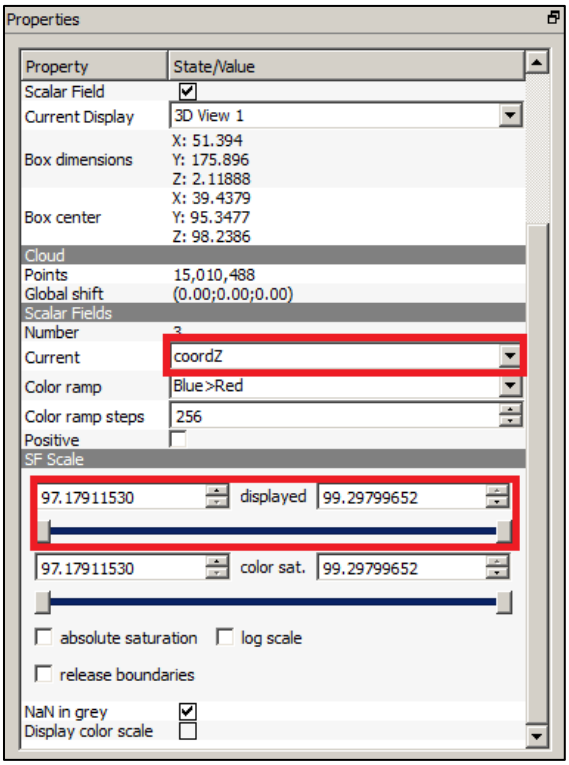


Figure 4. Min/max coordinates (mapped into scalar fields in ***CloudCompare***)

LIMIT	X (feet)	Y (feet)	Z (feet)
Minimum coord.	13.74089909	7.39987087	97.17911530
Maximum coord.	65.13487244	183.29547119	99.29799652

Table 2. Precise minimum and maximum coordinates of original data from ***CloudCompare***

3) **Clean the data by manually removing artifacts**. In the image below, you can see that there are some obvious artifacts: many look like people or equipment. They should be removed before triangulation, and since they are clearly visible at this stage now is the best time to remove them. The next exercise in this series describes how to remove large sections at a time with ***CloudCompare***, which is more useful for records from large digital elevation databases.

   a. Turn on point selection. Back in ***MeshLab***, select the **Edit > Select Vertexes** tool from the menu or  from the *Edit toolbar* above the display. Hold the left mouse button and drag around the points to select them. Appendix C introduces useful ***MeshLab*** keyboard and mouse commands for making additive and subtractive selections.

   b. Delete a set of points. Press *Ctrl+Del* to delete vertexes (or select **Filters > Selection > Delete Selected Vertices** from the menu or  from the *Action toolbar*).

   c. Save your work after each operation: there is no *Undo* command in ***MeshLab***! Select **File > Export Mesh** and save the cleaned data in **.xyz** format with **no normals** as *MiddleRioGrande_MLASCtxt_clean.xyz*. For this exercise, just remove a few artifacts.

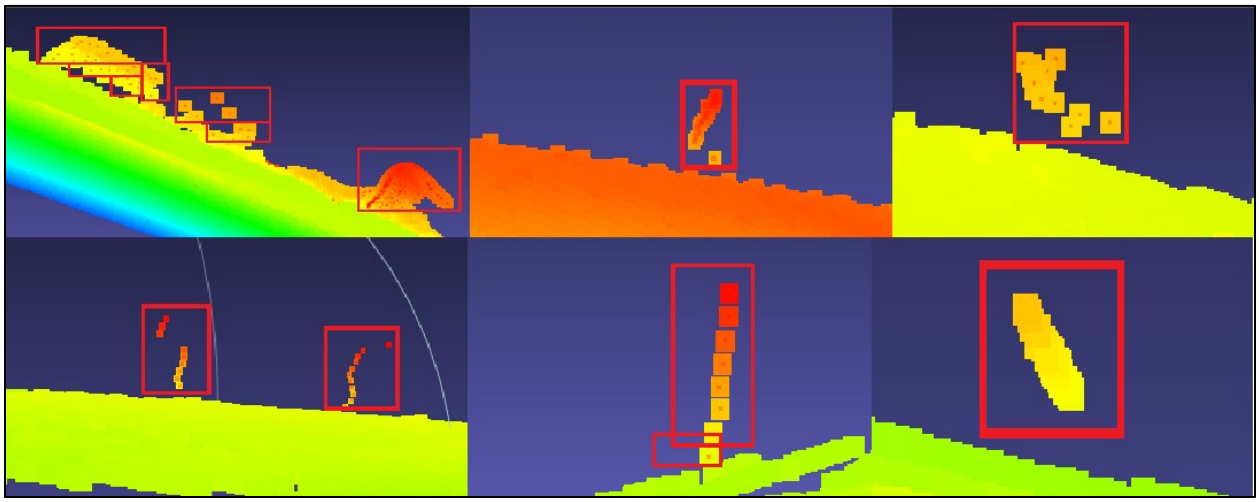


Figure 5. Removing some point artifacts

4) **Fill in empty regions with a flat plane**. This step produces more accurate and attractive *.stl* files. Pre-filling also speeds up the *.stl* conversion time by a factor of 100 or more!

   a. Create the plane. Select **Filters > Create New Mesh Layer > Mesh Generator**. Set **X scale = 60** and **Y scale = 180** (the absolute length of the grid sides) per Table 2 above. Set **Num vertices = 1443** and **4327** points in **X** and **Y**, respectively, for ½-inch spacing. Activate **Centered On Origin**, and click **Apply**. The plane will be built with both points and facets on a new layer named *Grid Generator* (see Figure 6 below).

   b. View layers. Select **View > Show Layer Dialog** from the menus or  on the *Standard* toolbar. Click on the  icons to hide and compare the original point cloud and new plane layers. If you need to reset the view, you can select **Windows > View From > Front**.

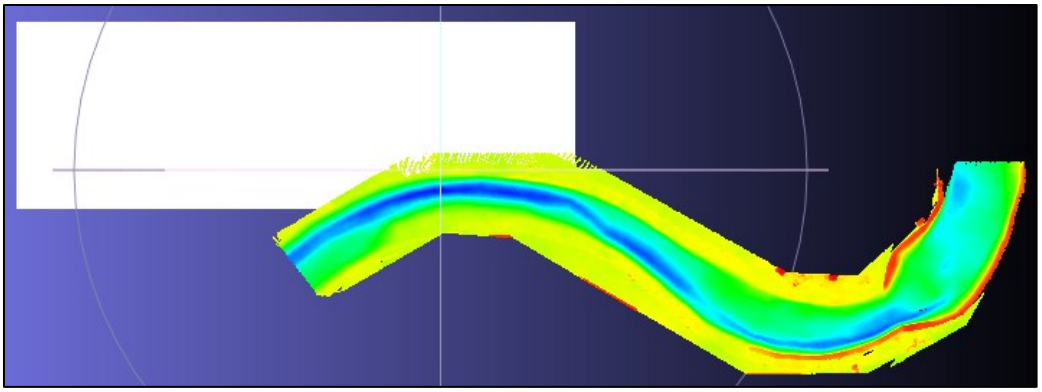


Figure 6. New plane added

   c. Translate the grid to position. Select the new layer, then **Filters > Normals, Curvatures, and Orientation > Transform: Move, Translate, Center**. Set **Z-Axis = 99.3** and **Freeze**

Matrix to actually change the point coordinates. **Apply** the filter, and then set the **Z-Axis = 0** again so the grid doesn't rise another 99.3 feet next time you click **Apply**. Enter **X-axis = 1** for translation and repeatedly click **Apply** until the grid has shifted so that it just encloses the point cloud extents. Set **X-axis = 0** and repeat the process for **Y-axis**. The plane should overlap the original data with a small margin on all sides.

d. Color the layer. Assign a **per-vertex attribute = *–z*** on **Filters > Quality Measure and Computations > Per Vertex Quality Function** and manually equalize the colors with **Filters > Color Creation and Processing > Colorize by Vertex Quality** to **Min = -99.3** and **Max = -97.2**. Check that the new grid color is red, meaning it is higher than the maximum elevation of the original data.

e. Remove the plane vertices that overlap the point cloud edges. Select the original point cloud topography layer and apply **Filters > Selection > Select All**. Then select **Filters > Sampling > Vertex Attribute Transfer**. Make sure the **source mesh** is the point cloud and the **target mesh** is the new plane. **Deselect** all default attributes and select **Transfer Selection**. Set **Max Search Distance = 0.5 (absolute world units)**. **Apply** then **Close** the filter. **Hide** the point cloud layer, select the plane layer, and delete the vertices that have been selected with the **Delete Current Set of Selected Vertices**  tool (see Figure 7 below).

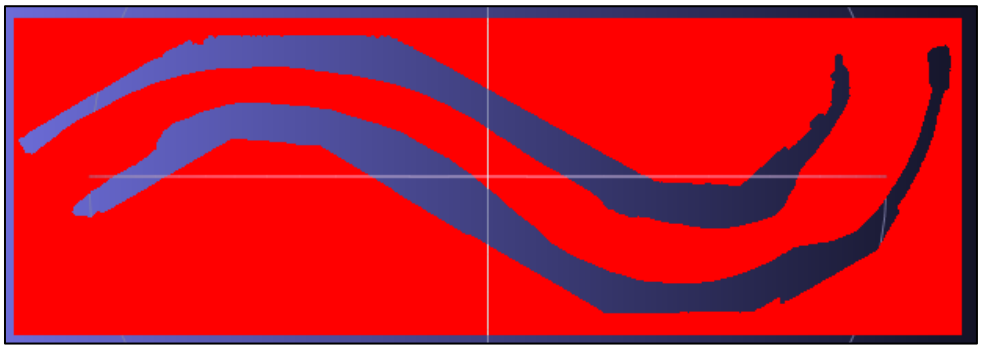


Figure 7. Delete points in the new plane that are within 0.5 feet of the river data

f. Remove the plane vertices that overlap the point cloud center. Keep the original data hidden and turn off the coloring by selecting **Render > Color > None**. Activate the **Select Vertexes on a Plane** tool  and roll the *mouse-wheel* to set the **HopThru** size to less than **0.5**. Hold *Control* and *click-drag* the mouse to add and size selections. Select and delete all of the points inside the originally-scanned data area. Use the **Select Vertexes** tool  where necessary to make sharp edges in the selection and manually clean up any points that escape the first attempt (see Figure 8 below). Check your work by showing both layers together and individually (see Figure 9 below). **Export** the final plane in **.xyz** format with **no normals** in case ***MeshLab*** crashes during the next step.

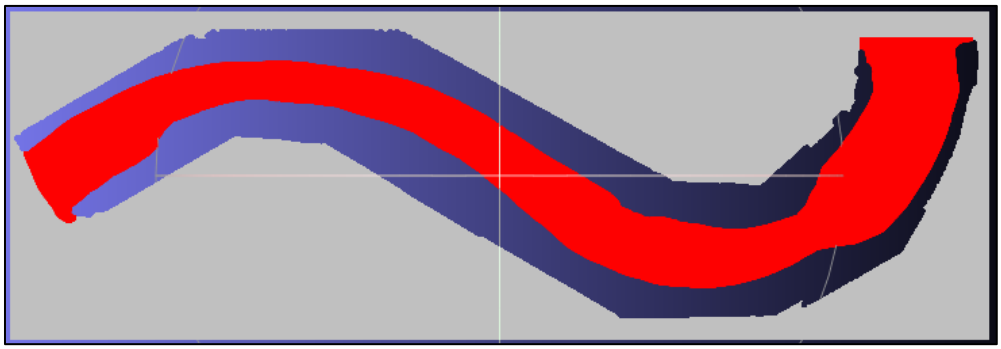


Figure 8. Select and delete remaining plane points that overlap river data

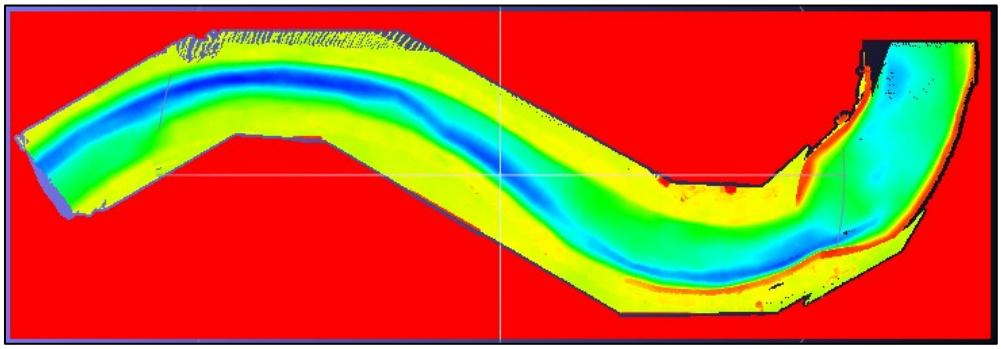


Figure 9. Final configuration of plane and river layers, colored by elevation

g. Merge the plane and river data. **Show** both layers, then *right-click* one of the layers in the *Layer Dialog* and select **Flatten Visible Layers**. Select **Keep Unreferenced Vertices** or all original points will be deleted! Don't delete the existing layers yet. **Export** the flattened layer with **no normals** as *MiddleRioGrande_MLASCtxt_clean_fill.xyz*.

h. Reload the data as points only. Now **Delete** all the layers (on the *Layer Dialog*) and **Import** the point cloud you just saved (which has been stripped of all facet data and will now be easier to work with). Remember how to colorize? Repeat the colorization process (**per vertex quality function = -z**). Your point data should appear similar to Figure 9 above.

5) **Pre-subsample the data**. There are many reasons for decimating a large point cloud (stability and run-time of various meshing algorithms, for example), and there are usually ways to do it that retain a high degree of fidelity. Two common methods are *Clustered Vertex* subsampling, which reduces the data via a one-per-gridded-cell approach (either averaging the point coordinates in a cell or taking the single point closest to the center), and *Poisson-disk* subsampling, which uses a stochastic method to retain points. The *Clustered Vertex* cubic grid method tends to concentrate points near contour lines (see Figure 10 below). *Poisson-disk* is therefore recommended for topographic applications.

a. Apply **Filters > Sampling > Poisson-disk Sampling**. Activate **Base Mesh Subsampling** (very important) and select **Explicit Radius (absolute world unit) = 0.0833** for 1-inch samples. **Export** the subsampled mesh with **no normals** as *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833.xyz*.

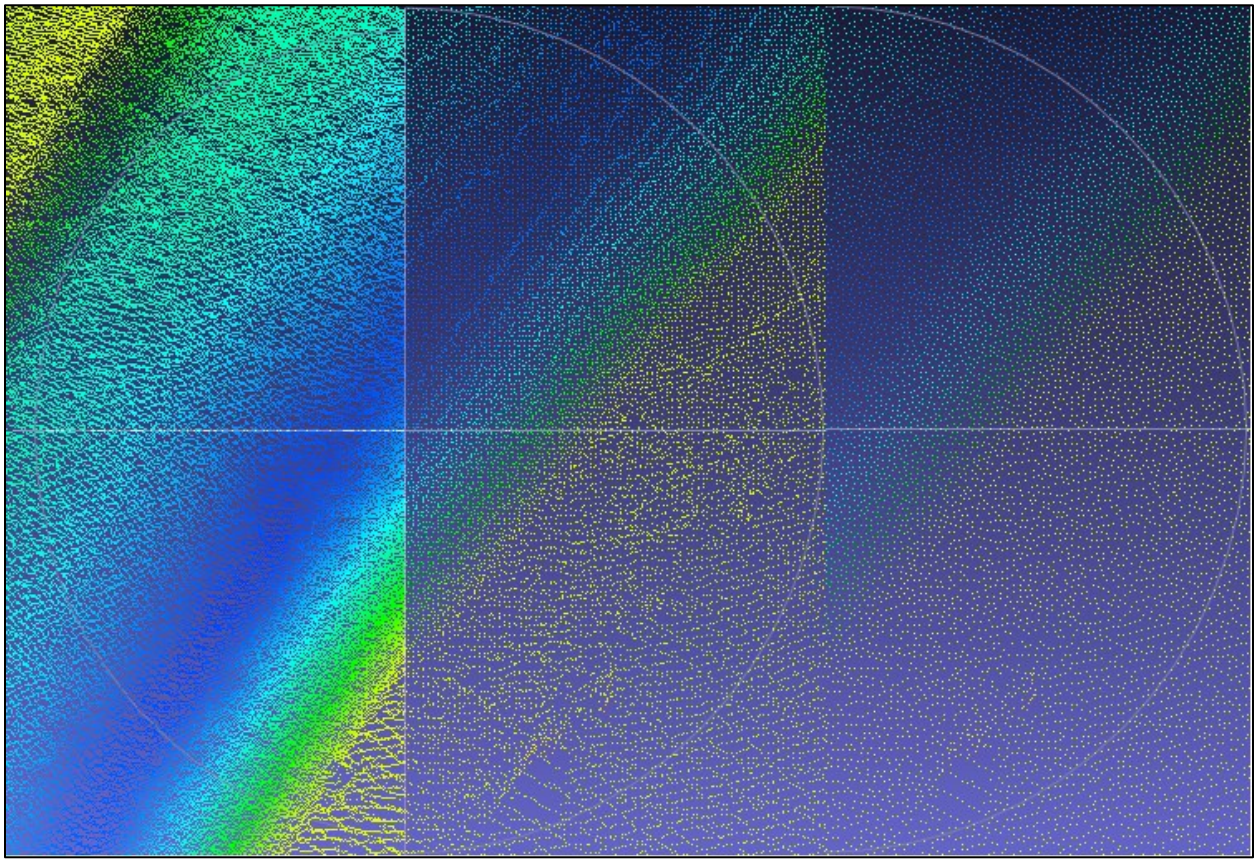


Figure 10. Subsampling. *Left*: original point cloud. *Middle*: 1-inch *clustered vertex* subsampling (emphasizes 1-inch contour lines). *Right*: recommended 1-inch *Poisson-disk* subsampling

6) **Use the *TOPO2STL* utility to convert to *.stl***. Download ***TOPO2STL*** from the *Utilities* section at users.flow3d.com. Uncompress it and run **Topo2STL.exe** to launch the file selector. Browse and pick *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833.xyz*. The file will take several minutes to load. Check that the **Data Size** (point count) reported is about 870,000 points.

a. Set point-sampling stencil. Set **Spatial Resolution = 0.0833** feet. Very large *.stl* files that exceed the available graphics card memory will not load in the ***FLOW-3D*** interface, although they can be added manually to *prepin* files that are run from the command line. The spatial resolution selection directly affects the file size. Post-conditioning can reduce the output file size.

b. Estimate the total number of surface points. Find the number of points ***TOPO2STL*** will either retain or create by multiplying the data extents in X and Y (60 x 180 = 1800 ft^2)

and dividing by the spatial resolution squared (0.0833^2) to get the approximate total number of points (1.6 million). A good rule of thumb is that *.stl* output files are about 1 GB for every 11 million points (including converting the output to binary *.stl* format).

c. Other conversion settings. Set **STL Minimum Z-Coordinate = 90** ft. Set the name of the output file as *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833_T2STL0833.stl*.

d. Generate the *.stl*. Click **Convert** to begin. Table 3 shows the conversion time for various spatial resolutions, with and without pre-conditioning the data set. Adding planar data to blank regions speeds up the conversion by a factor of 50 to 100. Pre-decimating the data set after filling in the empty regions gives additional speed improvements.

POINT SPACING FOR ***MESHLAB*** PRE-DECIMATION & ***TOPO2STL*** SAMPLING	PRE-ESTIMATE OF TOTAL SURFACE POINTS TO BE GENERATED BY ***TOPO2STL***	***TOPO2STL*** CONVERSION STATISTICS		
		RAW DATA SET: NOT CLEANED, NOT PRE-FILLED, NOT DECIMATED	CLEANED DATA SET WITH EMPTY REGIONS PRE-FILLED, NOT PRE-DECIMATED	CLEANED DATA SET WITH EMPTY REGIONS PRE-FILLED & POISSON-DISK PRE-DECIMATION TO TARGET ***TOPO2STL*** SPACING
¼-inch (0.0208 feet)	24.96 million points	***TOPO2STL*** did not complete the conversion within 4 days. Process was manually terminated.	18.6 million pts in/ **53 minutes**/ 21.40 million pts out/ 10.7 GB ASCII *.stl*	6.94 million pts in/ **45 minutes**/ 21.36 million pts out/ 10.7 GB ASCII *.stl*
½-inch (0.0416 feet)	6.24 million points	15.0 million pts in/ **18 hours**/ 4.88 million pts out/ 2.45 GB ASCII *.stl*	18.6 million pts in/ **20 minutes**/ 5.83 million pts out/ 2.93 GB ASCII *.stl*	4.43 million pts in/ **10 minutes**/ 5.85 million pts out/ 2.95 GB ASCII *.stl*
1-inch (0.0833 feet)	1.56 million points	15.0 million pts in/ **3.5 hours**/ 1.22 million pts out/ 629 MB ASCII *.stl*	18.6 million pts in/ **10 minutes**/ 1.39 million pts out/ 714 MB ASCII *.stl*	0.87 million pts in/ **2 minutes**/ 1.37 million pts out/ 707 MB ASCII *.stl*
2-inch (0.1666 feet)	0.389 million points	15.0 million pts in/ **10 minutes**/ 0.308 million pts out/ 158 MB ASCII *.stl*	18.6 million pts in/ **8 minutes**/ 0.356 million pts out/ 183 MB ASCII *.stl*	0.26 million pts in/ **<1 minute**/ 0.349 million pts out/ 179 MB ASCII *.stl*

Table 3. Conversion times and other data for four ***TOPO2STL*** resolutions
(tested on Intel Core-i7 3930K CPU with 16.0 GB of RAM & Windows 7 Professional 64-bit)

7) **Fix the output and convert it to binary format**. All automated triangulation algorithms create minor errors: disconnected edges between triangles, missing triangles (holes), and reversed or incorrectly-computed facet normal vectors. These errors can cause bad pressure and velocity solutions in ***FLOW-3D***. Additionally, ***TOPO2STL***'s ASCII output is five times larger than binary format. Fixing and converting the format of the *.stl* is *always required*. Three methods of fixing

and converting the part are demonstrated here; the third also introduces post-conversion simplification of unnecessary surfaces. The three methods use the free programs ***qAdmesh***, ***netfabb Basic***, and ***MeshLab***.

a. Use ***qAdmesh*** for repair & format conversion. This software is a user-interface for the free command-line program ***Admesh***. It comes bundled with ***FLOW-3D***. Source code and theory for ***Admesh*** are available at https://sites.google.com/a/varlog.com/www/admesh-htm. Launch ***qAdmesh*** from ***FLOW-3D*** by navigating to the **Model Setup > Meshing & Geometry** tab and selecting **Tools > qAdmesh** from the menu above the viewing pane. Select **Browse** under *Input* and load *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833_T2STL0833.stl*. Set the **Output type = Binary STL**, and name the output file *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833_T2STL0833_admesh.stl*. Select **Default check**, and click **Apply**.

   The results are given as text output: ***qAdmesh*** identifies about 5,400 facets with disconnected edges, about 2.73 million incorrect normal vectors, and two identified surface shells (which are called *parts* in ***qAdmesh***).

   Run ***qAdmesh*** again on the fixed output until no more changes are made to make sure that the part is repairable. Notice that the repaired part still has 2 shells. This is not ideal, but since the parts are fully closed and have no inverted normals it should not affect the ***FLOW-3D*** model.

b. Use ***netfabb Basic*** for repair & format conversion. This free software can be downloaded from http://www.netfabb.com/basic.php. It includes visualization, manual and automatic analysis and repair, and other options.

   Select **Project > Add Part** and load the ASCII ***TOPO2STL*** output file *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833_T2STL0833.stl*. The entire part is selected so all faces are green. Clicking anywhere except on the part will deselect it. Re-select the part by clicking on it or selecting it in the *Parts* tree at upper right, which works much like the layers in ***MeshLab***. Use the eye logo  to toggle part visibility. Notice the warning label  in the lower-right corner. It indicates that a loaded part has detectable problems.

   Select **Extras > New Analysis > Standard Analysis**  . A new *Part Analysis* branch is added to the tree: selecting the *Standard Analysis* sub-branch displays data about the part: it has two shells, two holes, an unclosed surface, and about 5,400 unjoined edges (called *boundary edges* in ***netfabb***): very nearly the same as the number of *facets with*

unclosed edges reported by ***qAdmesh***. Note that ***netfabb*** does not report incorrect normal vectors like ***qAdmesh***, but it does correct them automatically.

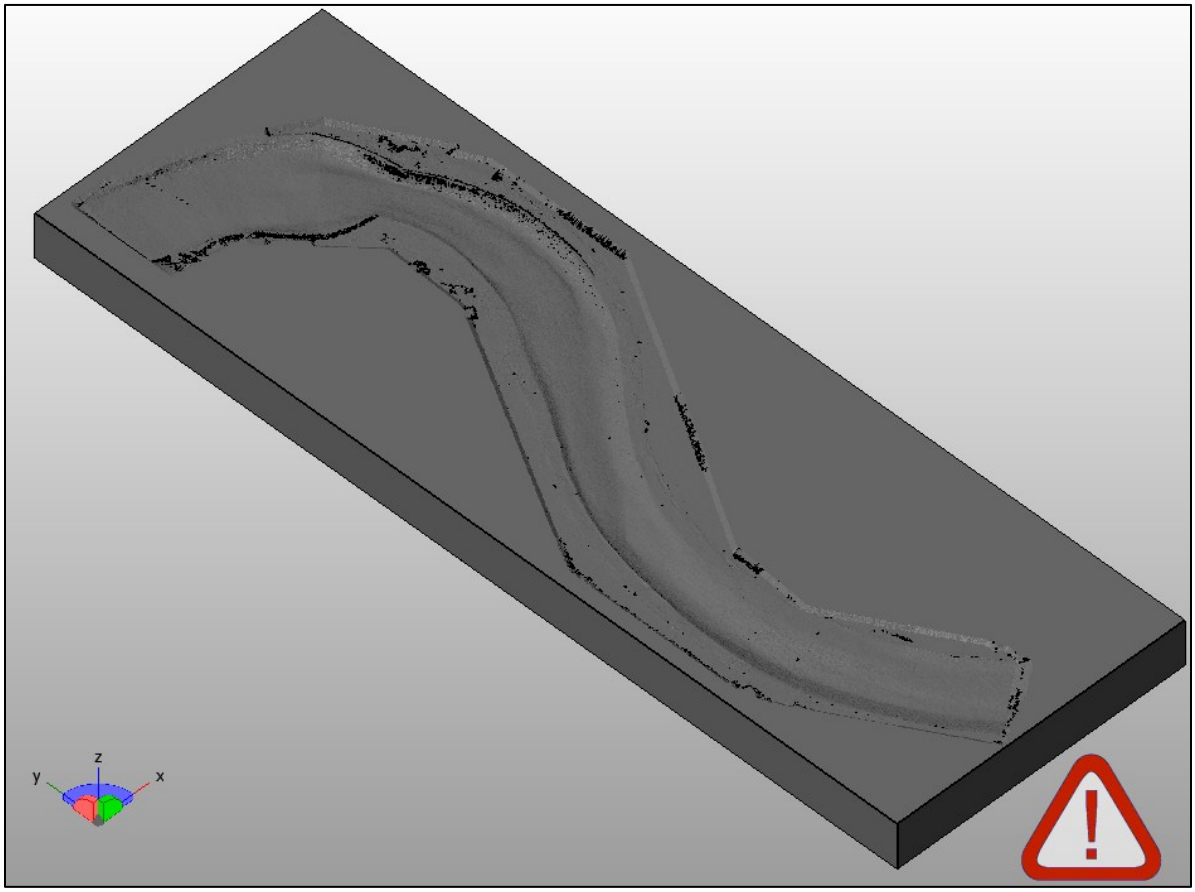


Figure 11. ***TOPO2STL*** output (1-inch-spacing, pre-decimated-and-filled) in ***netfabb Basic***

Select **Extras > Repair Part** to create a new *Part Repair* branch in the tree. When this branch is selected, individual facets and points can be added and deleted and automatic repairs can be applied. Select the *Part Repair* branch and click **Automatic Repair** at the bottom of the information display. Choose **Default Repair** and **Execute**. Accept the changes by clicking **Apply Repair** to generate a new fixed part. You can choose to leave the original part in the tree or delete it.

Select the newly repaired part in the tree and perform a new **Standard Analysis** on it. Two holes have been filled, all facets have been connected, and unlike the ***qAdmesh*** results, there is only one shell. Repair algorithms are implemented differently in software packages, and almost always produce slightly different results. In this case the ***Netfabb Basic*** output is slightly preferable to that of ***qAdmesh***.

Choose **Part > Export Part > As STL** to save the repaired geometry in binary format. Name the part *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833_T2STL0833_netfabb.stl*. The binary file size will always be about 81.5% smaller than the ASCII file size. Re-check it with ***qAdmesh*** and verify there are no artifacts.

c. Use ***MeshLab*** for simplification, repair, and format conversion. ***MeshLab*** can be used to reduce the size of *.stl* files, which is particularly important for those created with a fine resolution. The goal is to reduce the file size without simplifying the data in the region of interest. The way to do this is to recombine and simplify only the facets of the surrounding box.

Import *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833_T2STL0833.stl* into ***MeshLab***. Choose **Unify Duplicated Vertices** during import.

Select only the non-river plane faces using **Filters > Selection > Conditional Vertex Selection**, with **Boolean function = (z > 99.298) or (z < 97)** (see Table 2). Note that **or** *must* be in lowercase. Activate **Strict Face Selection** to avoid picking up any river data.

Once the non-river surfaces are selected, apply **Filters > Remeshing, Simplification and Reconstruction > Quadric Edge Collapse Decimation** with the following options: **Target Number of Faces = 100000**, **Quality Threshold = 0.3**, **Preserve Boundaries of the Mesh (Boundary Preserving Weight = 10)**, **Preserve Normal**, **Preserve Topology**, **Planar Simplification**, **Post-Simplification Cleaning**, and **Simplify Only Selected Faces** which is very important to avoid simplifying the river! Check that the non-river facets have been simplified as in Figure 12 below. **Export** the simplified part in **Binary Format** as *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833_T2STL0833_MLquad.stl*.

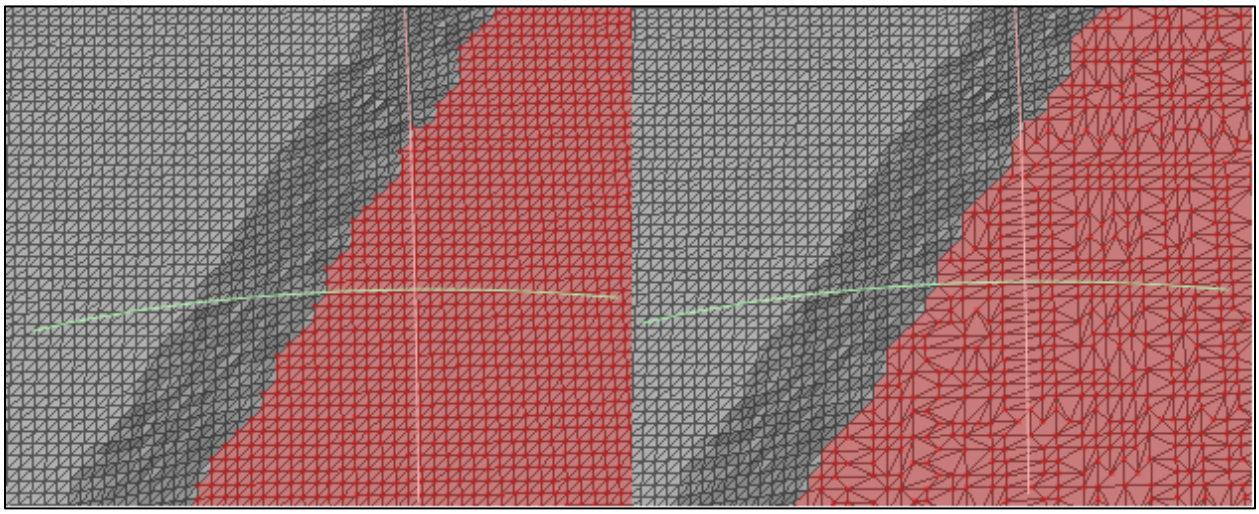


Figure 12. Simplification of the non-river surrounding box in ***MeshLab***

Apply **Filters > Selection > Select None** to deselect the faces. Select **Filters > Quality Measure and Computations > Compute Topological Measures**. The *Layers dialog* lists the text results. There are a number of boundary edges (meaning the part is not closed), non-manifold vertices, and incident faces.

Activate **Render > Show Non Manif Edges** and **Render > Show Non Manif Vertices** to visualize where the problems are. They are at the four lower corners of the box.

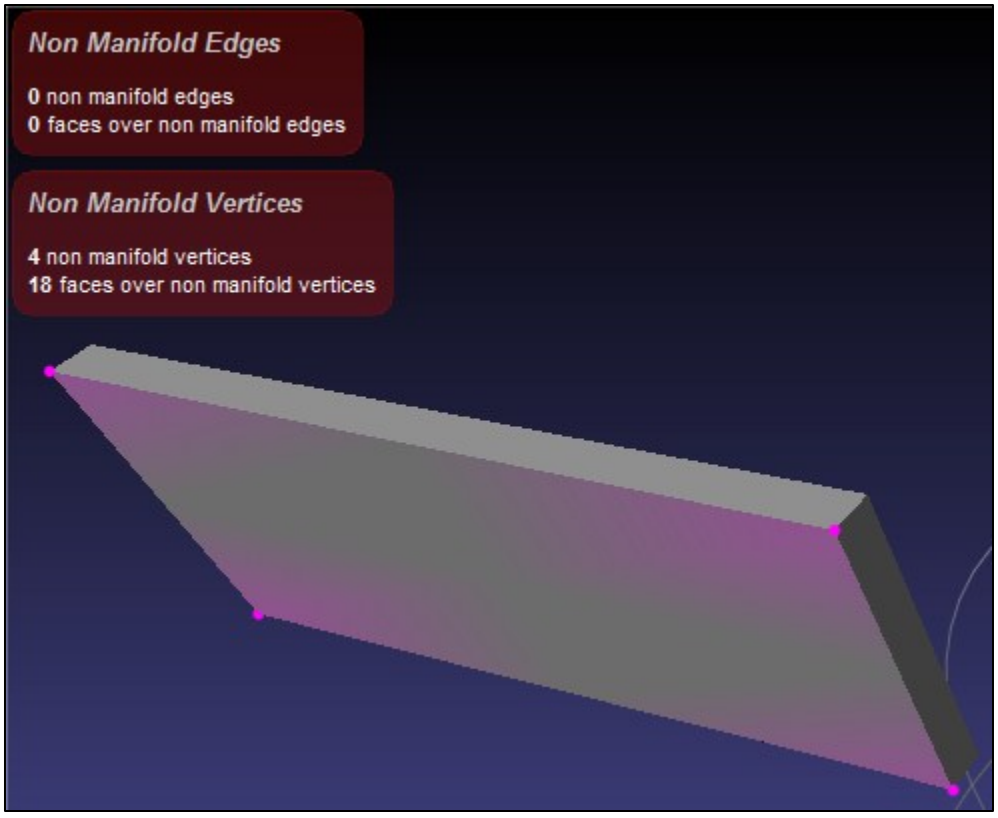


Figure 13. Locating non-manifold vertices with ***MeshLab***

Apply **Filters > Selection > Select non Manifold Vertices** and then delete them with the  tool. The bottom of the box will be deleted as well, since all facets in it share the same four corner vertices.

Start the **Edit > Fill Holes** tool  . No holes should be marked as non-manifold (thanks to the previous step). If there are non-manifold holes, first close and re-launch the **Fill Holes** tool: sometimes it's wrong. Mark the **Select** check-boxes for holes with *only three edges* (if any exist) and activate the **Trivial Holes** fill option. Select **Fill** and wait for the **Accept** check-boxes to appear: mark all the boxes and click **Accept** at the bottom of the

tool dialog. If there were many trivial holes, re-export the mesh in case ***MeshLab*** crashes later.

Now select the *big* hole, this time using the **Self Intersection** repair option. Click **Fill**, wait, and then click **Accept**. ***MeshLab*** will start to close the hole and incorrectly report that there are no more holes to fill. Ignore this message (click **OK** to close it) and repeat the process by re-launching the  tool. Keep **selecting**, **filling**, **accepting** and re-launching the tool until no more holes are reported when the tool starts. The bottom of the part should now be closed (Figure 14 below). **Export** the part, and check **Compute Topological Measures** again: there should be no boundary or non-manifold edges. The part should be two-manifold and composed of a single component.

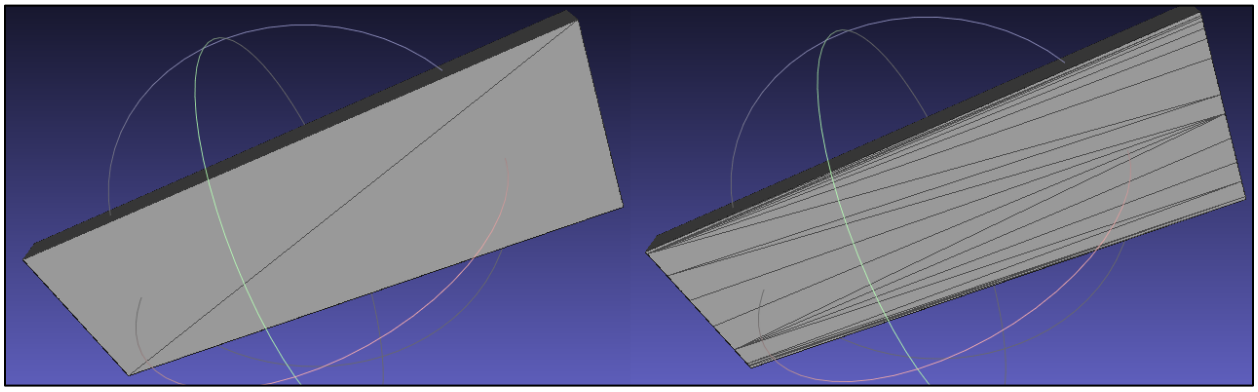


Figure 14. *.stl* bottom before (left) and after (right) removing non-manifold vertices and filling holes

Finally, apply **Filters > Normals, Curvatures and Orientation > Recompute Face Normals** and then **Filters > Normals, Curvatures and Orientation > Normalize Face Normals**. Export the part and check it with ***netfabb Basic*** (no errors detected) and ***qAdmesh*** (about 5,400 bad normals detected). Fix the normals with ***qAdmesh*** and save the output. Re-check the output with ***netfabb Basic*** and ***qAdmesh***: both programs should agree that the part is fixed.

The three repair methods introduced above are compared in Tables 4 and 5 below. The ***MeshLab*** simplification and repair procedure with ***qAdmesh*** finishing is recommended for general use because it can be performed on parts that are too detailed to load with the other methods, and it further reduces the final *.stl* file size.

SPATIAL RESOLUTION	***qAdmesh*** repair	***netfabb Basic*** repair	***MeshLab*** simplification & repair w/ ***qAdmesh*** finish
¼-inch (0.0208 feet)	Too many facets to fix in ***qAdmesh*** and ***netfabb Basic*** without prior simplification.		1E5 target facets/ Fixed: 4 vertices, 2 holes, 21599 normals/ **1 shell output/ 949 MB binary *.stl***
½-inch (0.0416 feet)			1E5 target facets/ Fixed: 4 vertices, 4 holes, 11079 normals/ **1 shell output/ 243 MB binary *.stl***
1-inch (0.0833 feet)	Fixed: 9509 facets, 2733148 normals/ **2 shells output/ 130 MB binary *.stl***	Fixed: 2 holes, 5440 borders, 2 shells/ **1 shell output/ 130 MB binary *.stl***	1E5 target facets/ Fixed: 4 vertices, 8 holes, 5400 normals/ **1 shell output/ 64 MB binary *.stl***
2-inch (0.1666 feet)	Fixed: 4765 facets, 692898 normals/ **2 shells output/ 33 MB binary *.stl***	Fixed: 2 holes, 2726 borders, 2 shells/ **1 shell output/ 33 MB binary *.stl***	1E5 target box facets/ Fixed: 4 vertices, 6 holes, 2722 normals/ **1 shell output/ 20 MB binary *.stl***

Table 4. Repair data for ***TOPO2STL*** output of pre-cleaned, pre-filled, and pre-decimated point sets

SPATIAL RESOLUTION	***qAdmesh*** repair	***netfabb Basic*** repair	***MeshLab*** simplification & repair w/ ***qAdmesh*** finish
¼-inch (0.0208 feet)	Too many facets to fix in ***qAdmesh*** and ***netfabb Basic*** without prior simplification.		1E5 target facets/ Fixed: 4 vertices, 1 hole, 21618 normals/ 1 shell output/ 951 MB binary *.stl*
½-inch (0.0416 feet)			1E5 target facets/ Fixed: 4 vertices, 3 holes, 11054 normals/ 1 shell output/ 242 MB binary *.stl*
1-inch (0.0833 feet)	Fixed: 9587 facets, 2759748 normals/ 2 shells/ 132 MB binary *.stl*	Fixed: 2 holes, 5480 borders, 2 shells/ 1 shell output/ 132 MB binary *.stl*	1E5 target facets/ Fixed: 4 vertices, 14 holes, 5446 normals/ 1 shell output/ 64 MB binary *.stl*
2-inch (0.1666 feet)	Fixed: 4824 facets, 707198 normals/ 2 shells/ 34 MB binary *.stl*	Fixed: 2 holes, 2760 borders, 2 shells/ 1 shell output/ 34 MB binary *.stl*	1E5 target facets/ Fixed: 4 vertices, 11 holes, 2722 normals/ 1 shell output/ 20 MB binary *.stl*

Table 5. Repair data for ***TOPO2STL*** output of pre-cleaned & pre-filled point sets without pre-decimation

8) Check the output (quality control). The final step is to visually and numerically compare the repaired *.stl* to the original artifact-clean point cloud to determine if the conversion process was accurate.

   a. Load the comparison data. Open ***CloudCompare*** and load *MiddleRioGrande_MLASCtxt_clean.xyz* (set file type to *ASCII*) and *MiddleRioGrande_MLASCtxt_clean_fill_PDSub0833_T2STL0833_netfabb.stl* (file type *STL*).

   b. Get preliminary comparison statistics. Hold *Control + left-click* to select both *Cloud* and *Mesh* entities in the *DB Tree*. Select **Tools > Distances > Cloud/Mesh Dist**  . The dialog that opens lists *Approximate Results* for the distance magnitudes between original points and the nearest *.stl* facets. It will shortly be demonstrated that *these values are incorrect*. The estimate of *Max Relative Error* gives the likely error of the distance results themselves as a function of the distance *d* between any point and the *.stl*. The *Approximate Results* are chamfer distances and not reliable for small *d* (Girardeau-Montaut *et al*. 2012).

   c. Get precise magnitude statistics and map them into color. Locate the *Precise Results* section. For best accuracy, set **Octree Depth = 9** or **10**. Deactivate **Signed Distances** and click **Compute**. After the calculation is complete click **OK** to exit the dialog and map the point results into a scalar field.

      Find the text *Console* panel at the bottom of the screen (press *F8* to toggle it). The text output gives a new estimate of the mean and standard deviation σ: about 0.01 feet (3 mm) for both. *The mean unsigned error is referred to as the Mean Absolute Error (MAE).*

   d. Find the maximum error magnitude. The *min and max SF scale slider values* are the *minimum and maximum error magnitudes*. In this case $|E|_{MIN} = 0$ and $|E|_{MAX} = 0.242$ feet ≈ 7 cm. The real maximum error is much smaller than was reported by the *Approximate Results*.

      In the *DB Tree*, un-check the *Mesh* part to hide it and then select the *Cloud* entity. Check that **Current Scalar Field = C2M Distances** in the *Properties* panel. Activate **Display Color Scale** under *SF Scale*. Under *SF Scale*, slide the *displayed minimum slider* all the way to the right (or set the minimum value box to the same as the maximum). This makes all values less than the maximum render as gray, and values equal to the maximum will be displayed as red. Deselect **NaN in grey** to hide the non-maximum points. Move the mouse near the upper-left of the display window to display *Point Size* options. Toggle the *Mesh* display under *DB Tree*, rotate, pan, and zoom the view, and increase **Point**

Size (*+*) until you locate the point of greatest error. Reactivate the *Mesh* entity to see the *.stl*.

Figure 15. Maximum error between *.stl* and point cloud is high on a steep bank slope

e. Visualize the error magnitude. Reset the *displayed* sliders to the minimum and maximum permissible (check that they match the *color saturation slider* values). Set the *color saturation maximum slider value box* **= 0.06** feet. Now all error magnitudes larger than about 1.8 cm appear in red, illustrating the regions of greater error. As visible in Figure 16 below, there are also some red areas that appear to be electrical cable for the ground-based LiDAR camera and that were not included in the point cloud. From this visualization it is clear that the majority of the error magnitudes are much less than 0.03 feet (less than 1 cm).

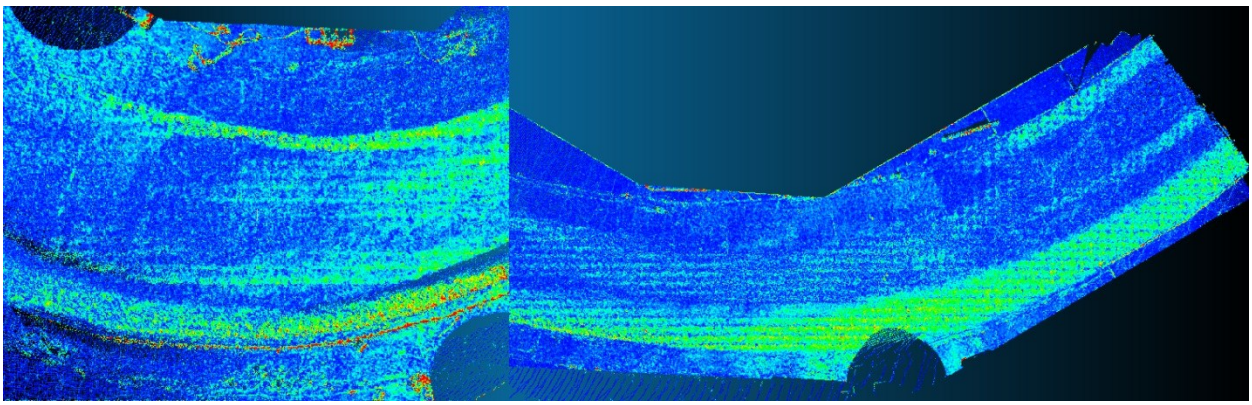


Figure 16. Visualization of error magnitude with blue = 0 and red = 0.06 feet.

f. Visualize the error magnitude distribution. Select **Tools > Statistics > Compute Stat. Params**, and set **Distribution = Gauss** to compare the actual error magnitudes to a normal distribution. The resulting histogram contains all the information that has already been determined: the maximum and minimum errors are the ends of the abscissa, and the mean and standard deviation are summarized (and rounded) above the chart. More significant figures can be displayed in the *Settings* options.

The white line represents a normal distribution (based on the mean and standard deviation). The normal distribution is not a good fit! Repeat the process and select **Weibull** as the distribution type. *The Weibull distribution is a good fit to the error magnitudes*. The curve parameters are given above the chart. Results should look like Figure 17. *The error mean and deviation should be reported in terms of these unsigned values*. The *Console* (F8 to toggle) gives a *Chi-squared (χ^2) test* for each fit.

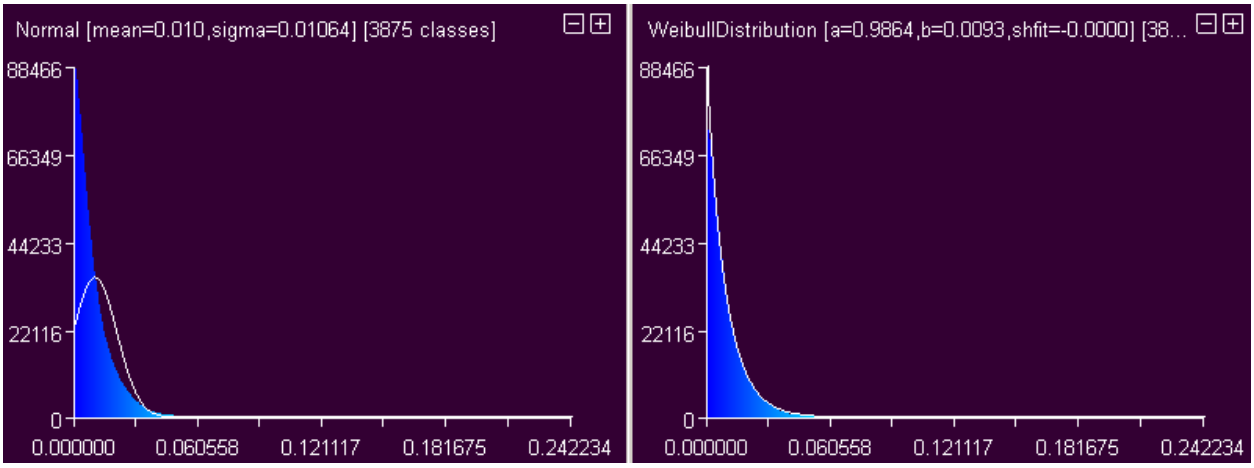


Figure 17. Statistical fitting of the error magnitude distribution: normal (left) and Weibull (right)

g. Visualize the signed error distribution. Select both the cloud and *.stl* layers, and re-run the **Tools > Distances > Cloud/Mesh Dist** tool. Activate **Signed Distances** under *Precise Results* and **Compute** the error to map it into the scalar field of the point cloud. Select the point cloud layer and run the **Tools > Statistics > Compute Stat. Params** tool first for **Distribution = Gauss** and then for **Weibull**. The results are shown in Figure 18. It's clear from these and the χ^2 tests that the signed error is uniformly distributed about the mean and that the *.stl* surface error (relative to the point cloud) is nearly unbiased (neither higher nor lower on average).

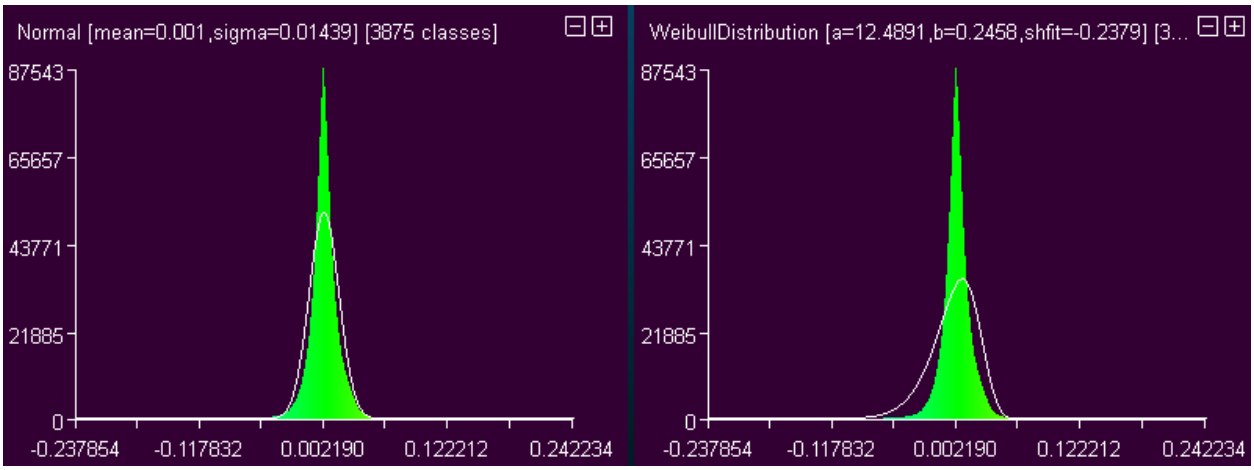


Figure 18. Statistical fitting of the signed error distribution: normal (left) and Weibull (right)

Table 6 below shows comparative mean error magnitude (MAE) results for various sampling spacing and pre-conditioning approaches. The 3 repair methods described in Tables 4 & 5 gave identical statistics (within 0.3 µm) for each method and spacing, so the results shown in Table 6 are true for all repair methods, regardless of how many shells the method produced.

POINT SPACING FOR ***MESHLAB*** PRE-DECIMATION & ***TOPO2STL*** SAMPLING	POINT CLOUD TO *.STL* DISTANCE MAGNITUDE (FEET)		
	RAW DATA SET: NOT CLEANED, NOT PRE-FILLED, NOT DECIMATED	CLEANED DATA SET WITH EMPTY REGIONS PRE-FILLED, NOT PRE-DECIMATED	CLEANED DATA SET WITH EMPTY REGIONS PRE-FILLED & POISSON-DISK PRE-DECIMATION TO TARGET ***TOPO2STL*** SPACING
¼-inch (0.0208 feet)	TOPO2STL could not convert the raw data within 4 days	Max: 0.247521 Mean: 0.004533 σ: 0.004472	Max: 0.146637 Mean: 0.004532 σ: 0.004175
½-inch (0.0416 feet)	MeshLab could not simplify non-river facets in 12 hours	Max: 0.307642 Mean: 0.006653 σ: 0.006960	Max: 0.200105 Mean: 0.006409 σ: 0.006309
1-inch (0.0833 feet)	Max: 0.33902451 Mean: 0.010307 σ: 0.012193	Max: 0.324913 Mean: 0.010395 σ: 0.012463	Max: 0.242234 Mean: 0.009717 σ: 0.010644
2-inch (0.1666 feet)	Max: 0.383123 Mean: 0.016889 σ: 0.023922	Max: 0.379844 Mean: 0.017011 σ: 0.023929	Max: 0.316623 Mean: 0.016472 σ: 0.019153

Table 6. Quality Control statistics for different conversion methods and spacing

CONCLUSIONS AND SUMMARY

An approach for converting large topographic point clouds to 3-D *.stl* parts has been recommended. Tables 3 and 6 indicate that *better conversion speed* and *better accuracy* can be achieved by *pre-filling empty regions* and *pre-decimating* the data set. Tables 4 and 5 illustrate the file-size reduction that can be achieved through *post-conditioning* the *.stl* part: *simplifying* the non-important surfaces and converting it to *binary format*. Recommended filters include *Poisson-disk subsampling* for pre-decimation and *quadric edge collapse decimation* for post-conditioning simplification.

The method outlined in this exercise has been tested on a 15-million point river (Middle Rio Grande scale model), a 61-million point watershed (Dalalven river), and several other sets. Comments and experiences should be addressed to the author (jeff.burnham@flow3d.com) or Flow Science support staff (support@flow3d.com).

REFERENCES

1. Girardeau-Montaut, D., Bey, A., and Marc, R. (2012). "CloudCompare User's Manual for Version 2.1." Le Faive, R., translator. http://www.danielgm.net/cc/, retrieved 3/7/2013.
2. Merriam-Webster (2013). "topography." http://www.merriam-webster.com, retrieved 3/7/2013.
3. Willmott, C.J., Ackleson, S.G., Davis, R.E., Feddema, J.J., Klink, K.M., Legates, D.R., O'Donnell, J., and Rowe, C.M. (1985). "Statistics for the Evaluation and Comparison of Models." *Journal of Geophysical Research* 90(C5): 8995-9005.

APPENDIX A – STL FORMAT

.stl ASCII format

solid <*partname*>
 facet normal <n_i> <n_j> <n_k>
  outer loop
   vertex <$v1_X$> <$v1_Y$> <$v1_Z$>
   vertex <$v2_X$> <$v2_Y$> <$v2_Z$>
   vertex <$v3_X$> <$v3_Y$> <$v3_Z$>
  endloop
 endfacet
……..
<*repeat beginning with "facet normal" through "endfacet" for every facet in the part*>
……..
endsolid <*partname*>

indentation must be with spaces (no tabs)
<n_I> <n_J> <n_K> are facet normal vector components in single precision float and may be negative.
<$v1_X$> <$v1_Y$> <$v1_Z$> are the vertex Cartesian coordinates (single precision float) and may not be negative.

.stl binary format

bytes	data type	data content
80-byte	ASCII	<*partname*>
4-byte	unsigned long integer	<*total number of facets*>
4-byte	floating point	<n_i>
4-byte	floating point	<n_j>
4-byte	floating point	<n_k>
4-byte	floating point	< $v1_X$ >
4-byte	floating point	< $v1_Y$ >
4-byte	floating point	< $v1_Z$ >
4-byte	floating point	< $v2_X$ >
4-byte	floating point	< $v2_Y$ >
4-byte	floating point	< $v2_Z$ >
4-byte	floating point	< $v3_X$ >
4-byte	floating point	< $v3_Y$ >
4-byte	floating point	< $v3_Z$ >
2-byte	unsigned integer	<*attribute byte count*>

……..
<*repeat beginning with the "n_i" line through the "attribute byte count" line for every facet in the part*>
……..

APPENDIX B - COMPARISON OF TEXT EDITORS FOR LARGE POINT CLOUDS

It is helpful to have a text editor which includes the following capabilities:

- Opens large (up to 2 GB or larger) ASCII files
- Finds and converts tabs, commas, and spaces using regular expressions,
- Finds and converts DOS, Unix/Linux, Macintosh, and Netscape line-end characters,
- Includes a column-edit method for block editing.

PROGRAM	CAN OPEN & VIEW 445 MB POINT CLOUD[1,2]	CAN OPEN & VIEW 953 MB TEXT FILE[1,3]	CONVERTS TAB TO SPACE IN 445 MB FILE[1,2]	INCLUDES COLUMN EDIT MODE	FREEWARE ?
Sublime Text 2	YES. Loads entire file into memory.		NO. Uses 4 GB memory. Does not complete within 1 hour.	LIMITED. Large column operations fail.	NO.
TextPad 6	YES. Loads whole file into memory.	YES. Converts some characters.	NO. Unreliable. Requires multiple tries.	YES.	NO.
gVIM 7.3 32-bit with ***LargeFile*** plugin	YES. Becomes briefly unresponsive during scrolling operations due to piecewise loading method.		YES. Total time about 3 minutes.	LIMITED. Requires text commands.	YES.
Crimson Editor 3.70	NO. Out-of-memory error.			YES.	YES.
EditPad Lite 7	YES.	NO. Location is loaded unreliably: final line # changes on separate tests.	NO. Uses all available memory. Does not complete within ½ hour.	NO. Available in EditPad Pro only.	MAYBE. Free for non-commercial use only.
ConTEXT 0.98.6	NO. Out-of-memory error.			YES.	YES.
PilotEdit Lite	YES. Loads entire file into memory. YES (loads into memory first).		NO. Does not complete within ½ hour.	YES.	YES.
Notepad++ 5.9.8	YES. Loads entire file into memory.	NO. Uses 1 GB of memory, then crashes.		YES.	YES.

Table B-1. Comparison of eight text editors

Notes:

1) Tested on Intel Core i7-3930K 3.20 GHz CPU w/ 16.0 GB RAM, Windows 7 Professional 64-bit OS
2) *MiddleRioGrande.txt*, courtesy of Colorado State University and US Bureau of Reclamation.
3) "Enwiki9.txt", courtesy of Matt Mahoney and the Large Text Compression Benchmark project (http://mattmahoney.net/dc/text.html, accessed Nov 19, 2012).

APPENDIX C – A BRIEF GUIDE TO KEYBOARD AND MOUSE COMMANDS IN MESHLAB

Navigation commands

- *Left click and drag* to **rotate** the part and coordinate system around the trackball center.
- *Mouse wheel* to **move** the part and coordinate system along line of sight relative to the trackball.
- *Center mouse button and drag* to **pan** the part and coordinate system.
- *Shift+mouse wheel* to change **camera** field-of-view.
- *Control + mouse wheel* to move **"near" clipping** plane.
- *Control + Shift + mouse wheel* to move **"far" clipping** plane.
- *Double-left-click* to place the **trackball center** at the clicked-on point.
- *Alt + Enter* to enter **full screen** mode.
- *Control+Shift+left mouse button and drag* to change **lighting** direction (only if there are normals).

Selection commands (when a selection tool is active)

- *Left-click and drag* to draw a box and **select all** faces or vertices in it (even those behind surfaces).
- *Alt+left click and drag* to select **only visible** faces or vertices in the box.
- *Ctrl+left click and drag* to **add** the faces or vertices in the box to the already-selected group.
- *Shift+left click and drag* to **remove** the faces or vertices in the box from the already-selected group.
- *Ctrl+Alt+left click and drag* to **add only visible faces** in the box to the group.
- *Shift+Alt+left click and drag* to **remove only visible** faces in the box from the group.
- *Esc* to **toggle** between navigation mode and the selection tool.

Important considerations

- The trackball is the point about which the coordinate system (and part) rotates and moves.
- There is no undo function: after every major processing step, export the mesh with a new filename.
- Some filters are less stable: most do better with a few million points or less. Consider subsampling.
- The MeshLab project file (*.mlp*) file records mesh file path, transformation, & camera/lighting data.
- The *.mlp* file stores no information about the points or faces. That data is stored in the mesh file.
- The transformation matrix is a 4x4 record of movement (scaling, rotation, translation) of the mesh.
- The Freeze Matrix option means that when the mesh is exported it will include the transformations.

Exercise 10: STL Topography from Google Earth

MAKING STL FILES FROM GOOGLE EARTH USING SKETCHUP AND CAD2STL

INSTALL NECESSARY SOFTWARE

You will need to install the following programs:

- ***SketchUp Make 2014*** (http://www.sketchup.com/)
- ***TIG Extrude Tool 2.5*** (http://sketchucation.com/resources/pluginstore?pln=ExtrudeTools)
- ***Cad2Stl v1.4*** (users.flow3d.com)

The *TIG Extrude Tool* plugin will come in handy later so you won't have to manually draw each *.stl* face. Follow the steps below to install it.

1. Launch ***SketchUp Make 2014***.
2. Select **Window > Preferences** from the top menu.
3. Choose **Extensions** from the list on the left.
4. Choose the **Install Extension...** button at the bottom.
5. Specify the **ExtrudeTools_v2.5.rbz** file.
6. Make sure the **Extrude Tools** plugin is active in **System Preferences** and click **OK**.
7. If the **Extrude Tools toolbar** doesn't appear, right-click on the toolbar and select **Extrusion Tools** from the list.

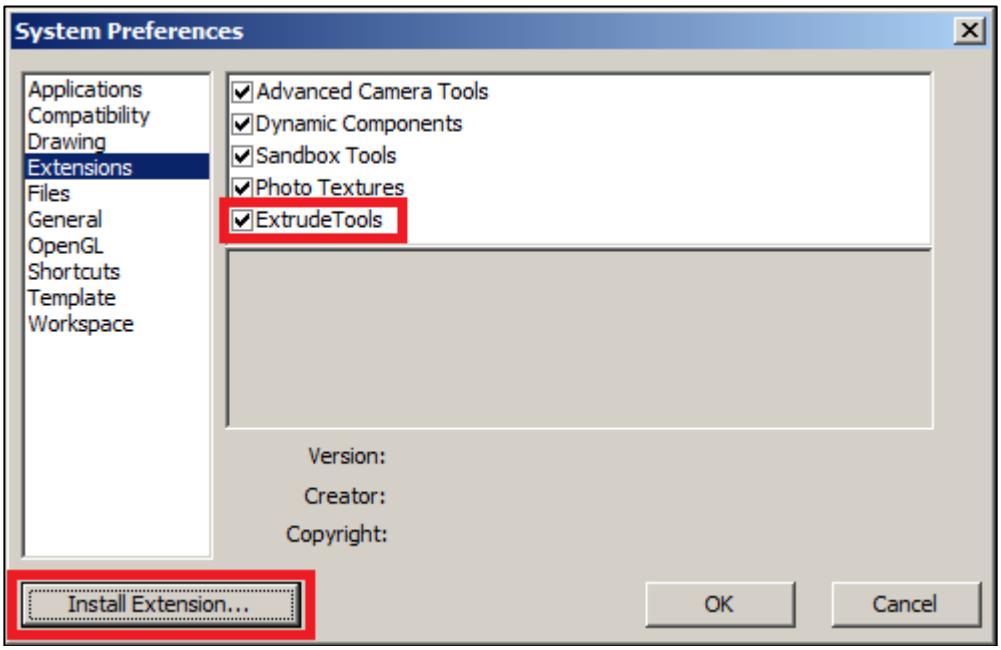


Figure 1. Install the ***TIG Extrude Tools v2.5*** extension in ***SketchUp Make 2014***

GET TOPOGRAPHY FROM GOOGLE EARTH USING SKETCHUP

Getting topography is simple in ***SketchUp***. The resolution may not be as fine as that from national or regional DEM databases, but it is often adequate for general modeling.

1. Select **File > Geo-Location > Add Location...** or select the icon in the toolbar.
2. **Search** and/or **Navigate** to find your location of interest.
3. Click the **Select Region** button and drag the corner pins to enclose the area of interest.
4. Click the **Grab** button to lay the selected topography image on the ***SketchUp*** map.

Figure 2. Locate your region of interest in ***SketchUp***.

MAKE A WATERTIGHT SURFACE FROM THE TOPOGRAPHY USING SKETCHUP

1. Select **File > Geo-location > Show Terrain** to three-dimensionalize the terrain map.
2. Select the model by clicking it, then *right-click* and select **Unlock** from the context menu.
3. *Right-click* the model again and select **Explode**.

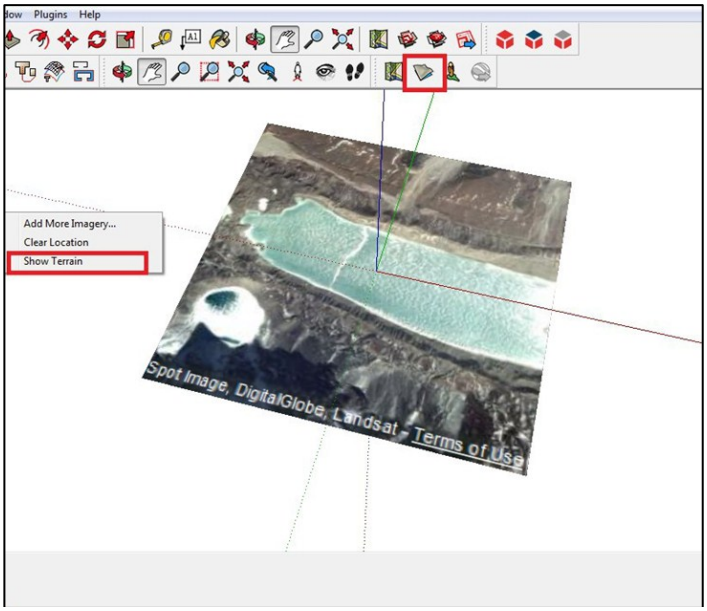


Figure 3. Three-dimensionalize the terrain.

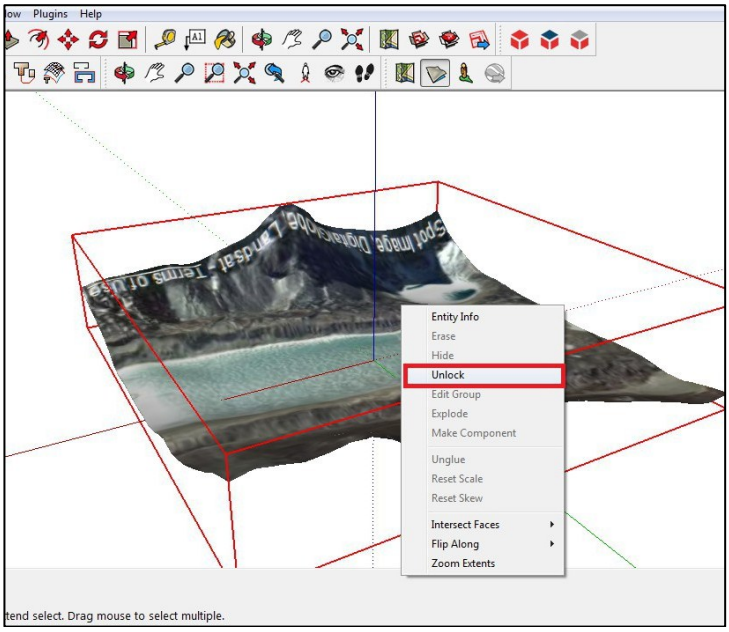


Figure 4. Unlock the terrain model in ***SketchUp***.

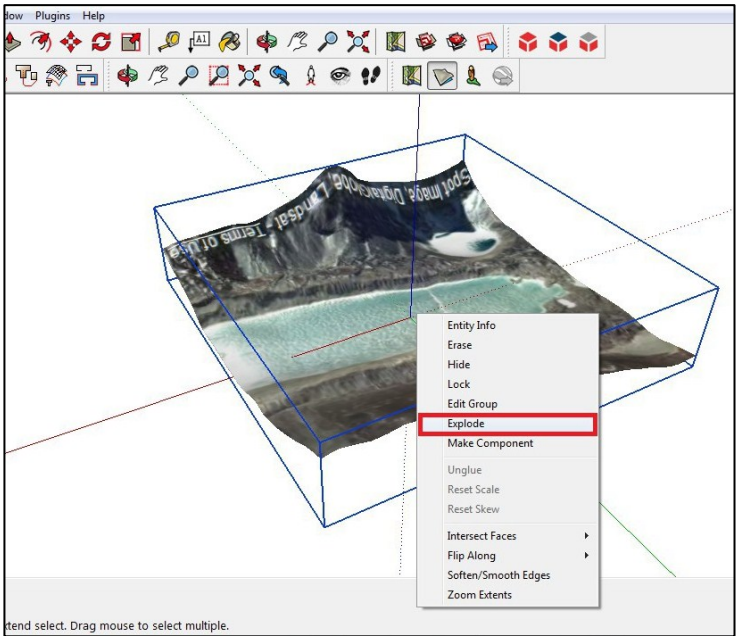


Figure 5. Explode the terrain to create a triangulated surface from the terrain model in ***SketchUp***.

4. Click the background to deselect the triangulated surface. Then *double-click* the object to select it *with edges*.
5. Select **Extrude Edges by Vector** from the **Extrusion Tools toolbar**.
6. Extrude in -z axis. You may need to move the mouse somewhat until it extrudes by the **blue axis**.
7. Click the **Select** arrow icon from the main toolbar and a dialog will appear.
8. The tool sometimes extrudes reversed faces. If the faces are reversed they will appear gray, as in Figure 7. If they are reversed, select **Yes**. Otherwise select **No**.
9. Another dialog will appear asking whether to explode extruded faces. Select **Yes**.

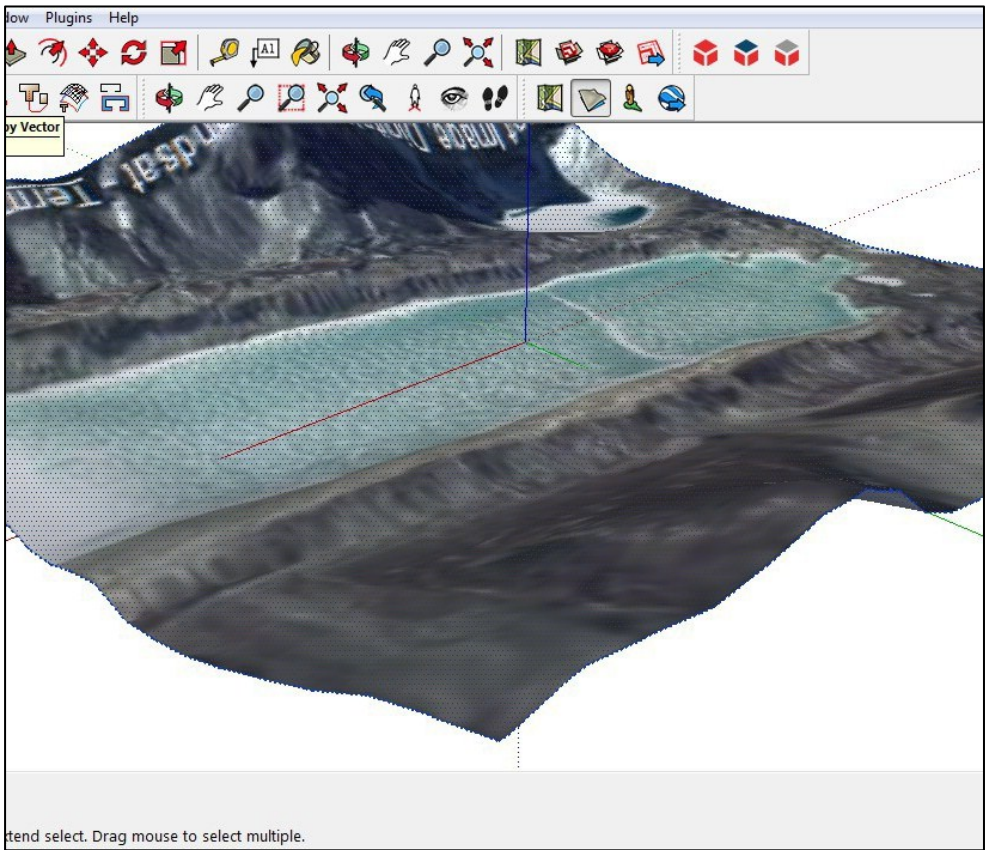


Figure 6. Reselect the triangulated surface as edges in ***SketchUp***.

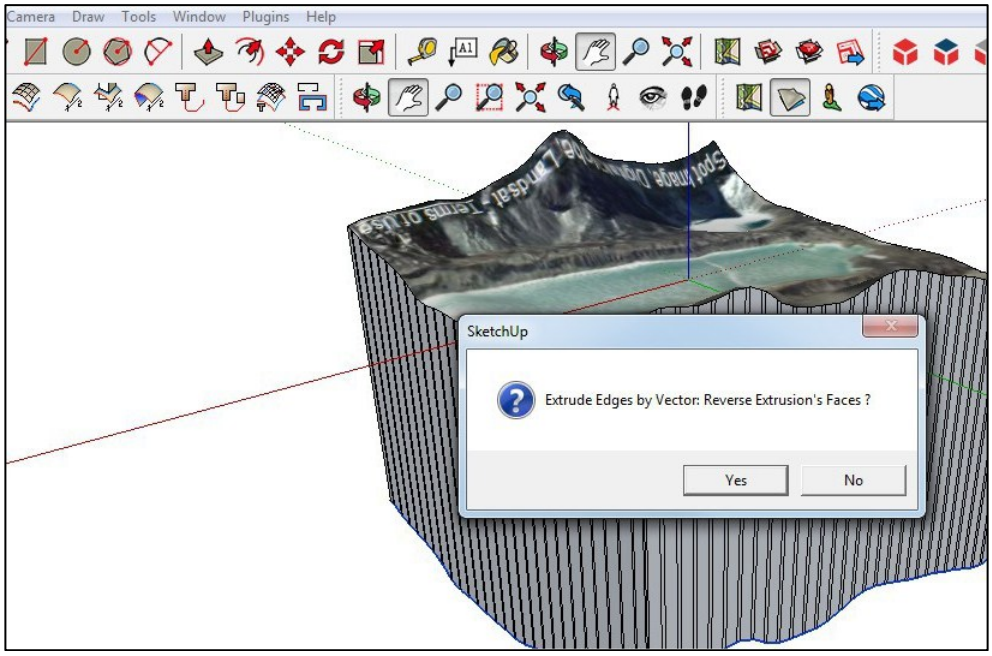


Figure 7. Extrude the surface downward following the blue (z) axis and check and fix reversed faces.

10. Select **Shapes > Rectangle** and draw a large rectangle on the canvas to completely surround the topography.
11. Click the **Move** tool icon and lower the plane so it cuts the extruded faces without cutting the surface.

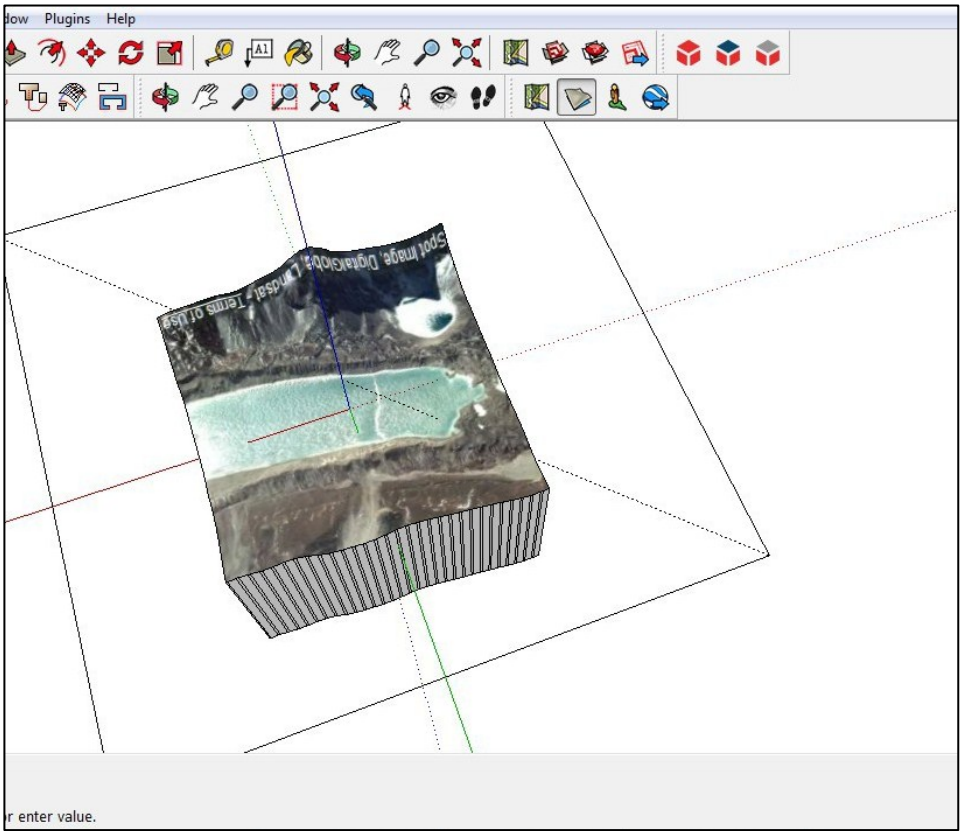


Figure 8. Draw a rectangle to surround the topgraphy.

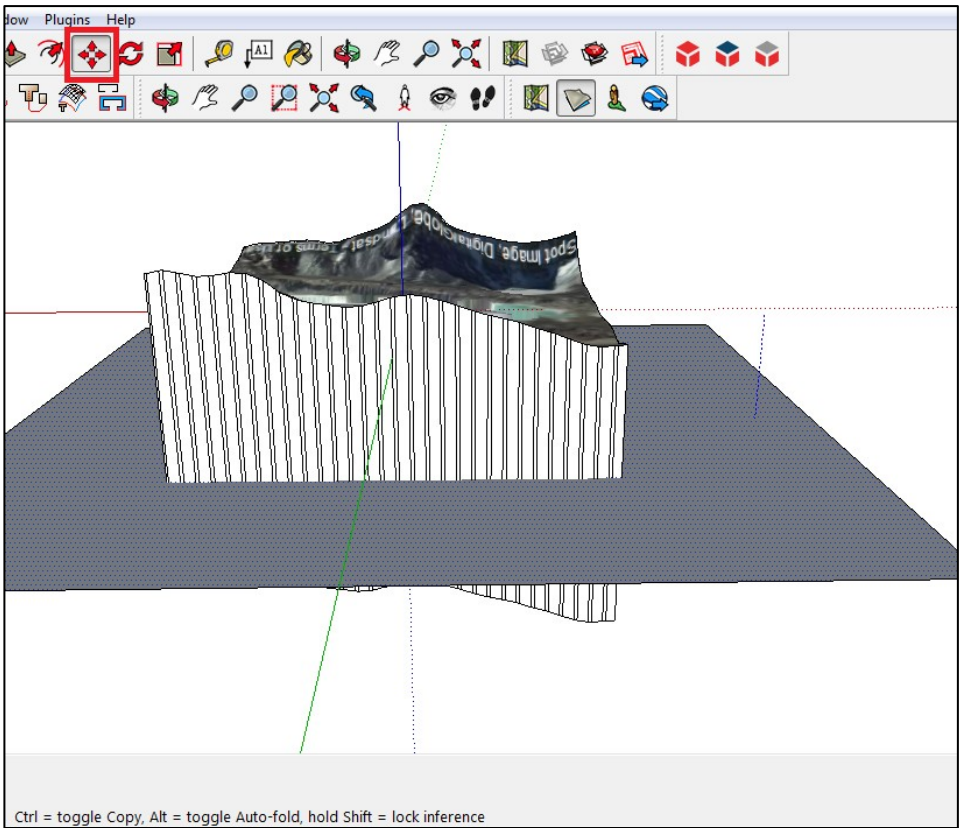


Figure 9. Place the plane so it cuts through the extruded faces.

12. Adjust the camera and make a selection that includes all the extruded faces *and* the rectangle.
13. Right-click the selection and select **Intersect Faces > With Selection**.
14. Draw a selection box to select the surfaces below the rectangle and **Delete** the extended surfaces, including the rest of the rectangle that extends past the topography.
15. **Select All** (*Ctrl+A*) and then *right-click* the selection and select **Make Group** from the context menu.
16. Select **File > Export > 3D Model...** to export the geometry file. Save the file as *.obj* or *.3ds* format.

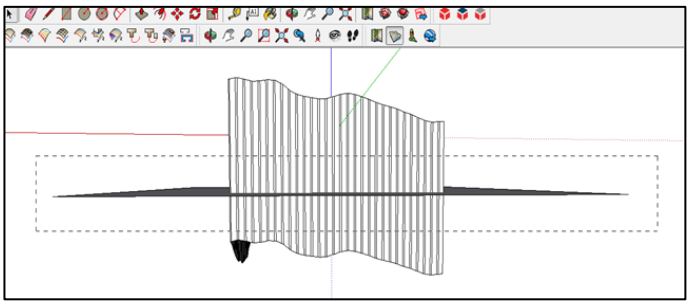


Figure 10. Select extruded faces and the rectangular plane together.

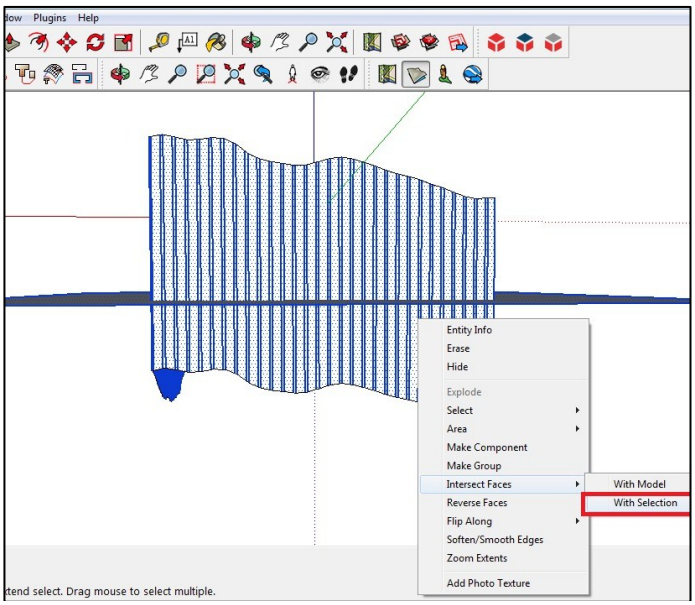


Figure 11. Intersect the faces and then delete the unneeded bottom faces and rectangle sides.

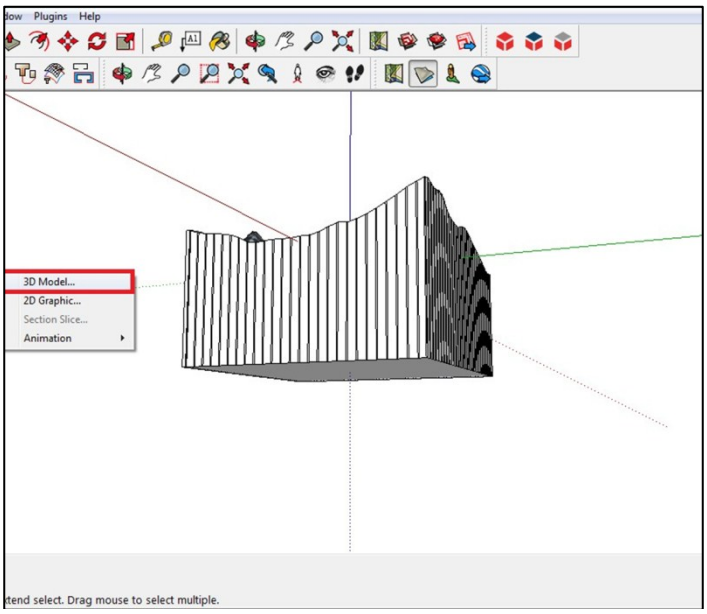


Figure 12. Export the completed model.

CONVERT THE SURFACE USING CAD2STL

1. Open ***Cad2Stl*** by running **run_cad2stl.bat**.
2. Select **Add** and browse to select your *.3ds* or *.obj* file.
3. Select the file from the list and click convert.
4. ***Cad2Stl*** will save the *.stl* file in same folder and same name as the original file by default.
5. Check and fix any errors in the *.stl* using tools like ***pyAdmesh*** or ***netfabb Studio***.

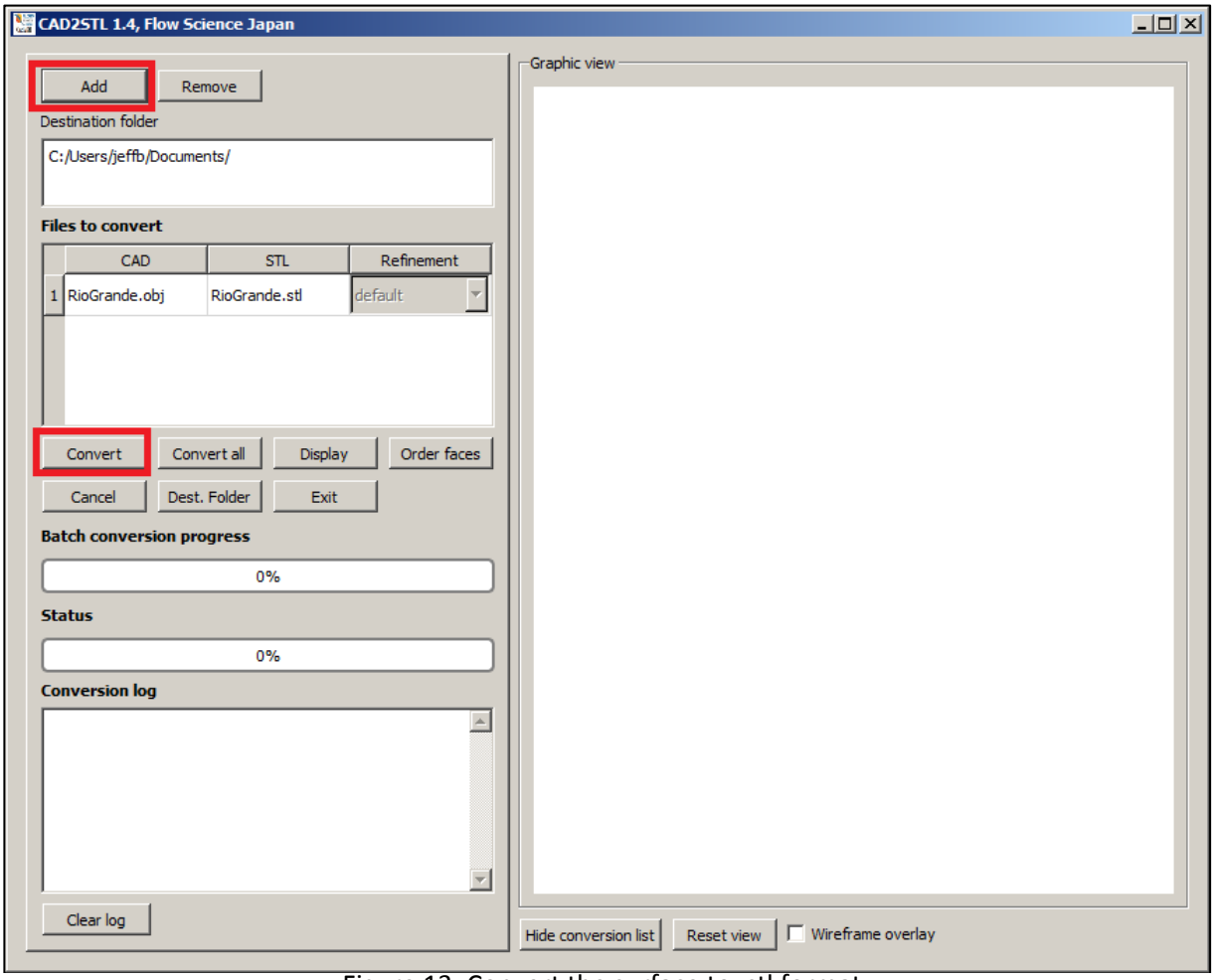


Figure 13. Convert the surface to .stl format.

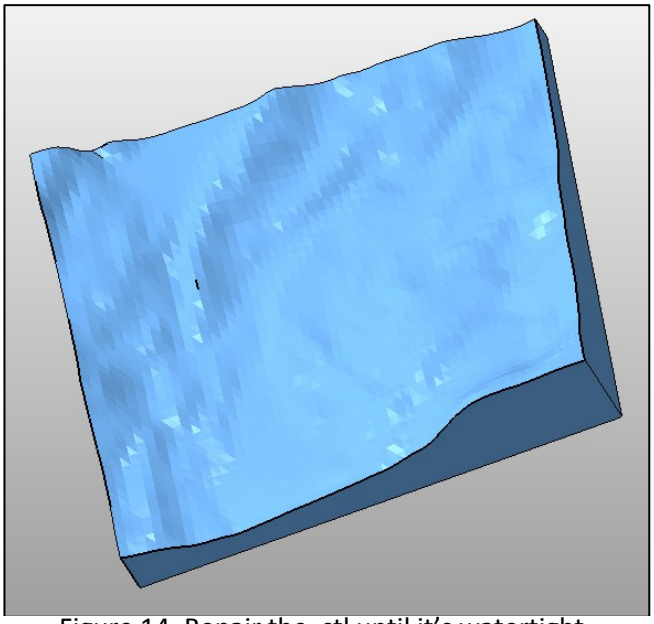


Figure 14. Repair the .stl until it's watertight.

Exercise 11: STL Topography from USGS Database

Making .STL Files from USGS Data

TABLE OF CONTENTS

INTRODUCTION

This note outlines a method for converting large topographic data sets to *.stl* format for use with ***FLOW-3D***. This note builds on the exercise ***Creating .STL Files from Topography***, available on the ***FLOW-3D Users Site***. Many of the methods in this note are described in more detail in that exercise. This note adds instructions for data sets from the U.S. Geological Survey (USGS) National Elevation Dataset (NED). The example describes converting an area of interest from 1/3-arc-second resolution USGS data in *ArcGrid* format to *DEM*, point cloud, and finally *.stl* formats. The method can also be used to convert *GeoTIFF* formats and *floating point* formats to *.stl*. This exercise uses open source freeware and ***FLOW-3D***'s ***TOPO2STL*** utility. There are many other possible workflows that involve 3rd-party commercial software that may be faster or more reliable. The basic processes and principles will be the same regardless of software.

STEP 1 – DOWNLOAD USGS DATA FOR THE REGION OF INTEREST

The region of interest is the inundation map of the St. Francis dam break flood of 1928, from the California coastline near Ventura up to and including the St. Francis dam and reservoir (Figure 1). Topographic data is downloaded for free from the USGS National Elevation Dataset (NED) at http://viewer.nationalmap.gov/viewer/. The download will contain *all* 1-degree x 1-degree tiles that

contain the region of interest. For this case, the region of interest results in two compressed files, each a 1 degree-square (35 N and 119W through 120W), in 1/3 arc-second *ArcGrid* format. Unzip each file into a new temporary working folder located on the root. Refer to http://ned.usgs.gov/Ned/faq.asp for details on NED data, including formats, accuracy, and methods of collection.

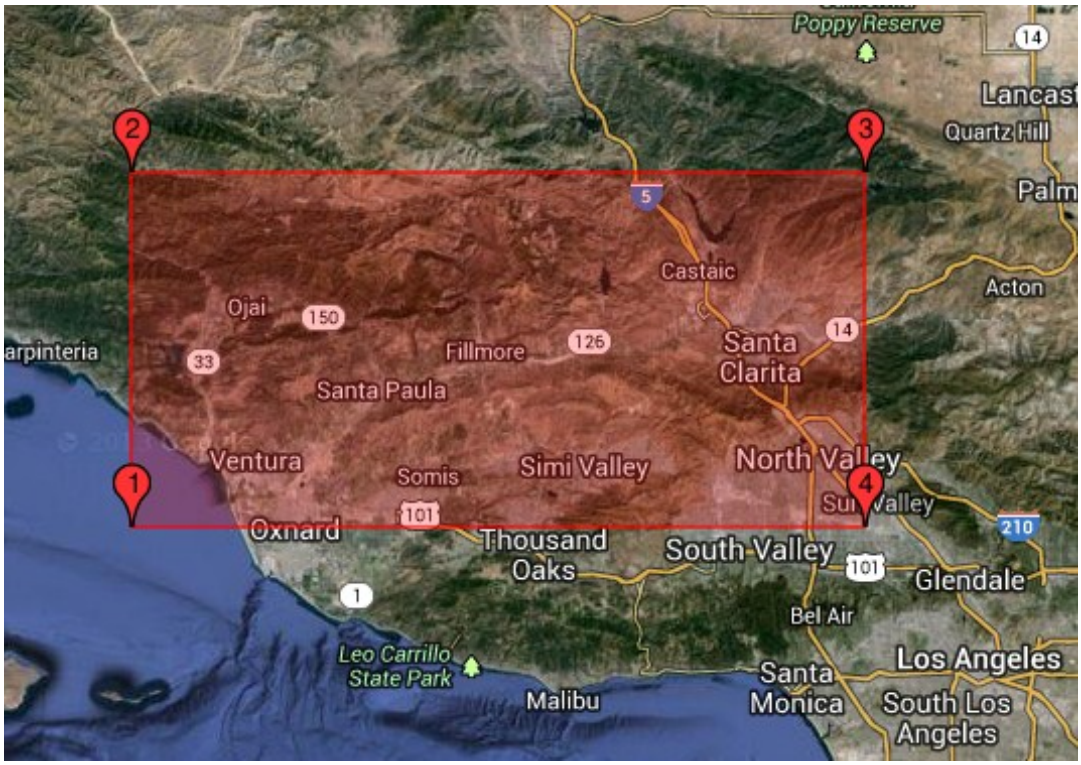


Figure 1. Approximate region of interest, viewed at http://earthexplorer.usgs.gov/

STEP 2 – USE *MicroDEM* TO CONVERT DATA TO POINT-CLOUD FORMAT

Download ***MicroDEM*** from Professor Peter Guth's page at the U.S. Naval Academy website: http://www.usna.edu/Users/oceano/pguth/website/microdem/microdem.htm. Install the software with full installation and default directory options. Update the program (*.exe*) and help file (*.chm*) builds from **File > Tools**, or download them from the ***MicroDEM*** website. *Right-click* the *.chm* file and select **Properties > Unblock** to allow it to work in ***Windows 7***.

FWTools provides the functionality to read the USGS formats. Download and install the latest version from http://fwtools.maptools.org/. In ***MicroDEM***, Set **Options > Directories > FW Tools** to the *bin* folder of the installation (*e.g.*, *C:\Program Files (x86)\FWTools2.4.7\bin*). Note that ***MicroDEM*** is under heavy development as of Fall 2013, so refer to its *Help* file for more current instructions.

Select **File > Data Manipulation > Resample > DEM formats > ADF Directories**. Select the folder that contains the *ArcGrid* data (*e.g.*, *n35w119_13sec_ArcGrid_a*). When prompted if the output should be *GeoTiff* format, select **No**. A ***MicroDEM***-format *DEM* file (*.dem*) will be created in the data subdirectory of the *ArcGrid* folder. Repeat for all *ArcGrid* sets. Select **File > Open > Open DEM** to open and view the new files. Arrange them as shown in Figure 2.

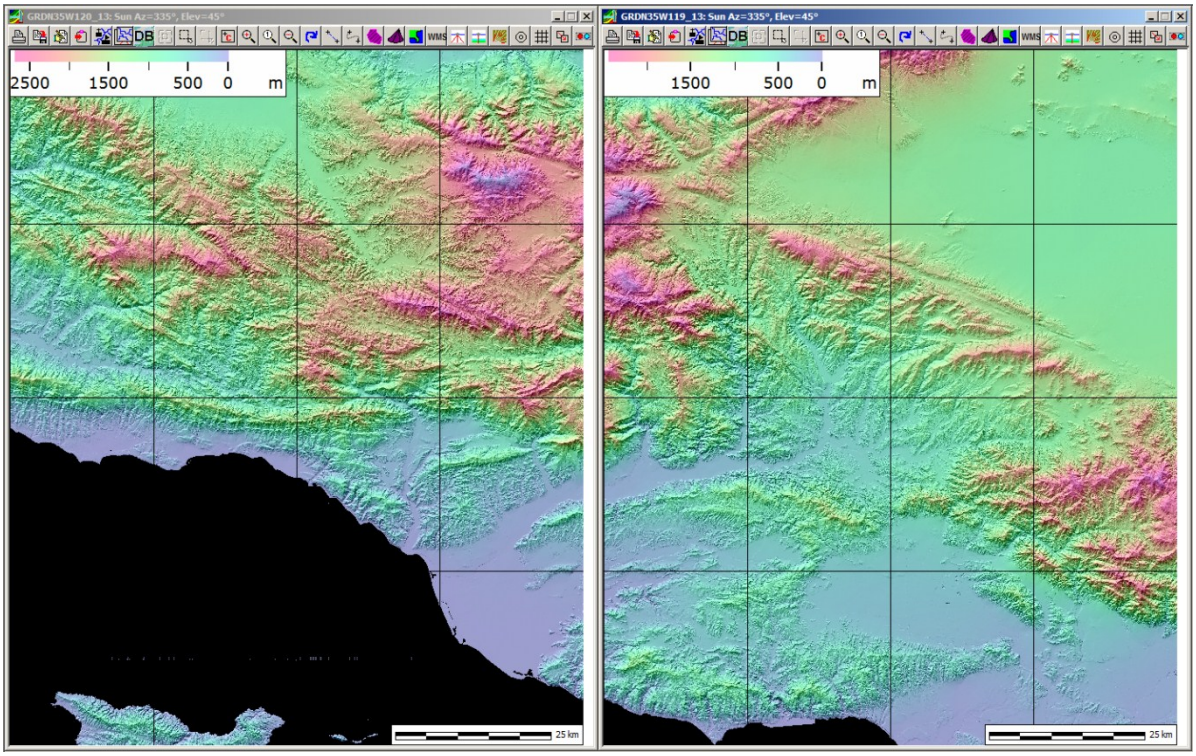


Figure 2. Two USGS 1-degree tiles, converted to *DEM* with ***MicroDEM*** and ***FWTOOLS***

MicroDEM has the option to simultaneously open & merge, but memory limitations in the executable prevent this from being useful for large data sets. Select **File > Save DEM > ASCII > ASCII XYZ (.xyz)** and the following options for each of the *DEM* files.

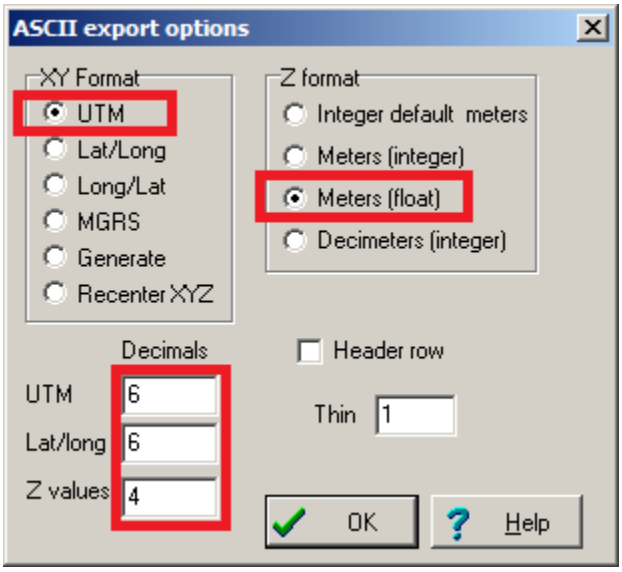


Figure 3. Recommended ***MicroDEM*** Save options

If you are computer-savvy you can choose to download *.bat* scripts from USGS instead of using ***MicroDEM***. The scripts use the ***GDAL*** library to convert the data without a software user-interface. The scripts are available as of November 2013 at

http://imsdemo.cr.usgs.gov/Website/Ned_Conversion_Scripts.zip. The ***GDAL*** library can be downloaded from http://www.gdal.org/, and must be installed and included in the *PATH* environment variable before running the scripts. The same ***GDAL*** library is included in ***FWTOOLS*** for ***MicroDEM***.

STEP 3 – USE *CloudCompare* TO COMBINE AND TRIM TOPOGRAPHY

Open the new files in ***CloudCompare***. You'll be shown a column list for each cloud where you can check the precision of the ***MicroDEM*** output. When prompted, shift each cloud by **-300,000.00 m in x**, **-3,000,000.00 m in y**, and **0.00 m in z**. This prevents loss of precision due to *machine rounding*. *Select* both clouds in the *DB Tree* pane and use the **Fuse** tool on the menu bar to combine them into a single object. **Save** the fused cloud as **ASCII .xyz** format with the options shown below. The new file will be about 197 million points (6.3 GB) that will be saved with the *original* coordinates. Color the points by elevation: **Tools > Projection > Export Coordinate(s) to Scalar Field(s) > Z**.

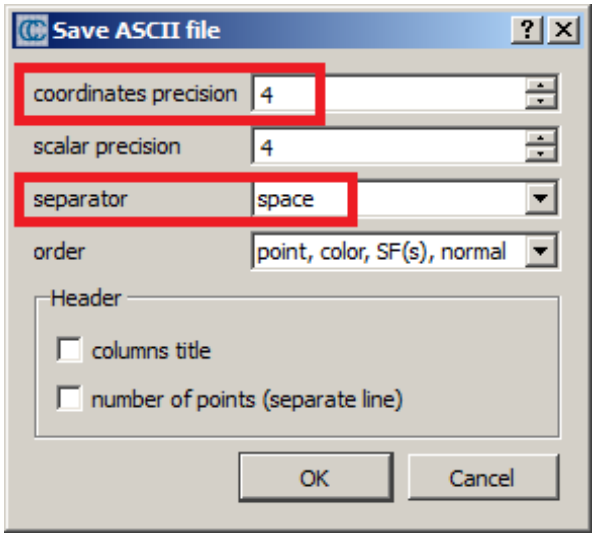


Figure 4. Recommended ***CloudCompare*** Save options

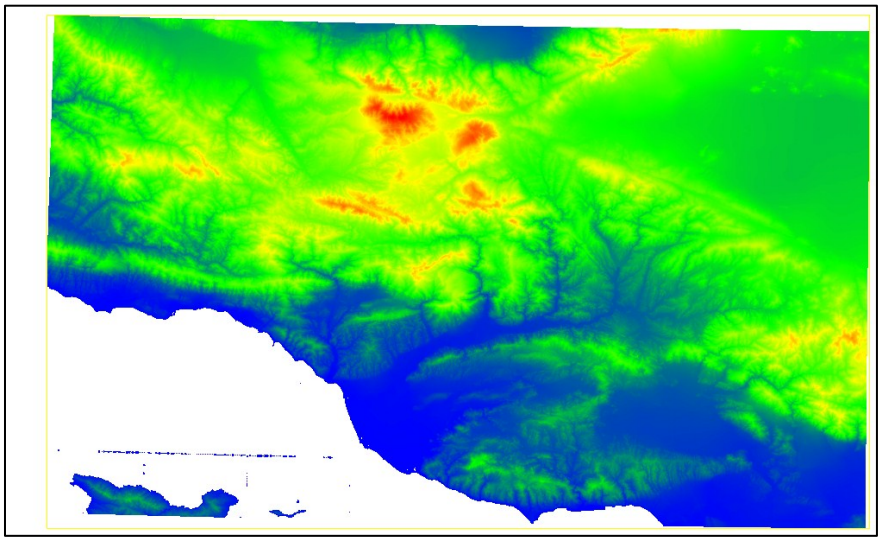


Figure 5. Fused UTM projections in ***CloudCompare***

Use **Edit > Scalar Fields > Filter by Value** to create a new cloud with elevations above **570** m removed. This is because the initial water level behind the St. Francis Dam was 559.4 m (1,835 ft) per Outland (1963), as cited in Sanders (2007). Select the **Segment** tool from the menu bar to cut out the region of interest. The segmentation tool opens with a new toolbar. Use *left-click* to add a point to the selection polygon and *right-click* to close the polygon. Select the action to make the contents of the selection polygon a separate point cloud. Edit the cutoff values in the *Properties pane > SF Display Params* section to help visualize the elevation gradients. Select to accept the new point cloud. Repeat as necessary. The goal is to minimize the X and Y *extents* of the topography. **Save** the final point cloud.

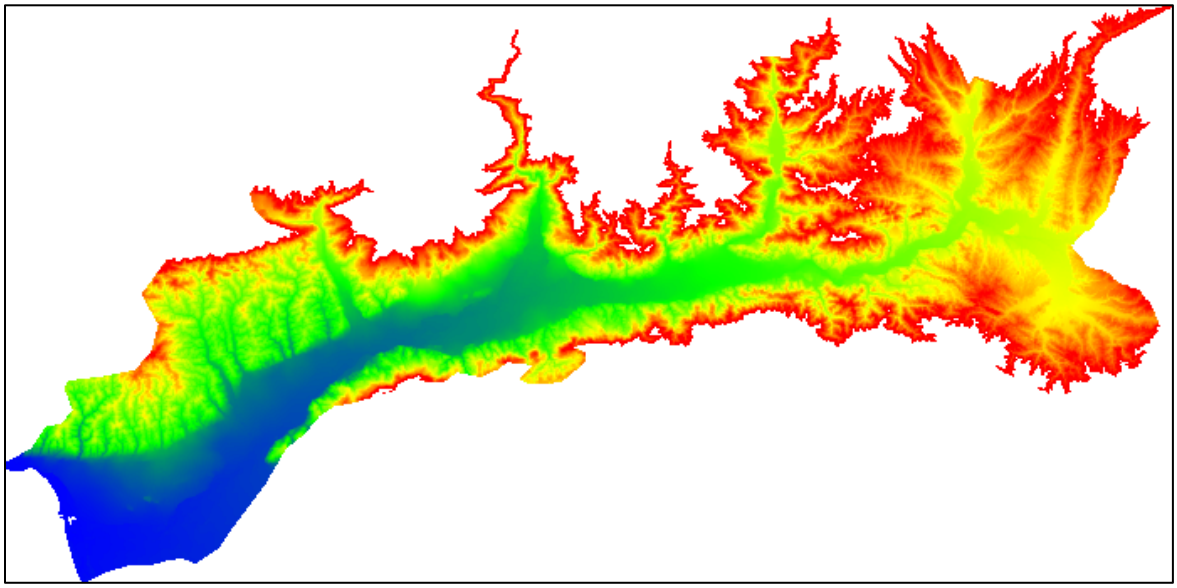


Fig 6. Final point cloud, filtered to below 570 m and segmented

Reload the point cloud with the new recommended shift, rounded by order of magnitude. For example:

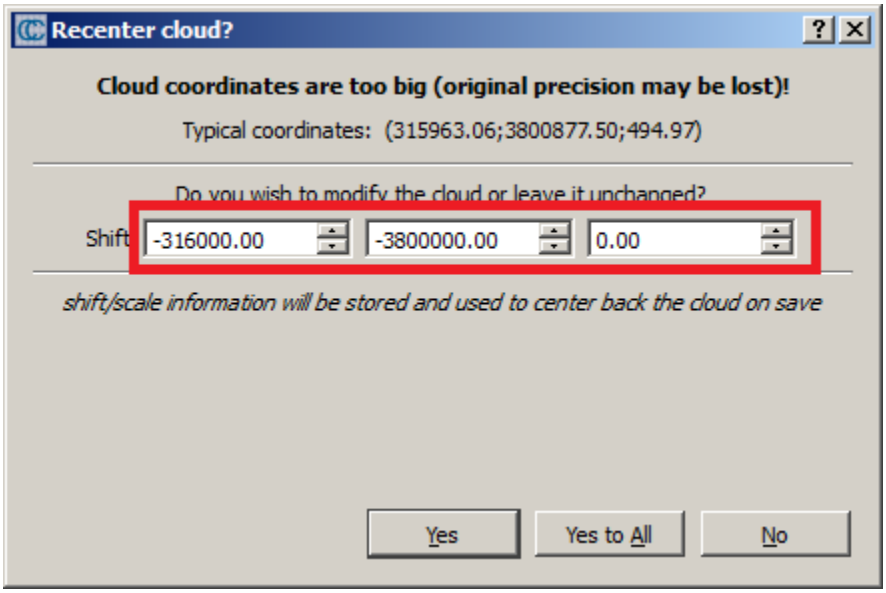


Figure 7. ***CloudCompare*** temporary shift options for segmented topography

Use **Tools > Projection > Export Coordinate(s) to Scalar Field(s) > X,Y,Z** to export the coordinates. Check the extents in the *Properties pane > SF Display Params* section. Toggle between coordinates by changing the **Active** dropdown selection in the *Properties pane > Scalar Fields* section. Calculate the required shift values so that the number of significant digits is minimized, and all coordinates are positive.

Use **Edit > Apply Transformation** and enter the calculated values for the *Tx* and *Ty* entries in the transformation matrix. This step permanently shifts the *saved* cloud coordinates. The ***FLOW-3D*** output will need to be shifted back by the opposite sign of these values to get UTM coordinates. This approach minimizes the effect of rounding error in the CFD calculations.

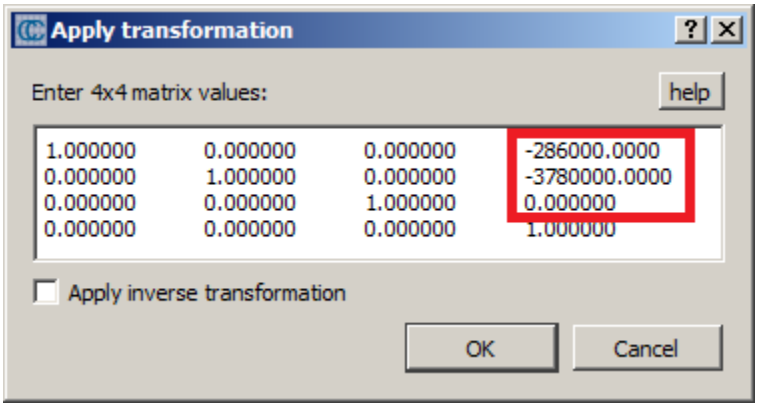


Figure 8. ***CloudCompare*** permanent shift options for shortest positive coordinates

Select **Tools > Scalar Fields > Delete All(!)** to eliminate the scalar fields from the saved output so that it can be opened in ***MeshLab***. **Save** the shifted cloud (now with no scalar fields) and re-open it. Repeat the coordinate checking process to verify and record the new extents. For example:

LIMIT	X (m)	Y (m)	Z (m)
Original max/min coordinates	288,889.7 to 365,560.8	3787191.3 to 3826153.5	-0.167 to 570.000
Translate	-286,000.00	-3,780,000.00	0.00
New max/min coordinates for modeling	2889.7 to 79560.8	7191.3 to 46153.5	-0.167 to 570.000
Coordinate Extents	76,671.1	38,962.2	570.167

Table 1. Translated Area of Interest

STEP 4 – USE *MeshLab* TO PREPARE THE TOP SURFACE

Skip the pre-decimation step described in the ***Creating .STL Files from LiDAR*** exercise. The goal here is to maintain the 1/3-arcsec resolution of the source data. Use ***MeshLab*'s Grid Generator** filter to create a plane with extents slightly larger than those in Table 1: use 78,240 m x 39,600 m with 15-m spacing.

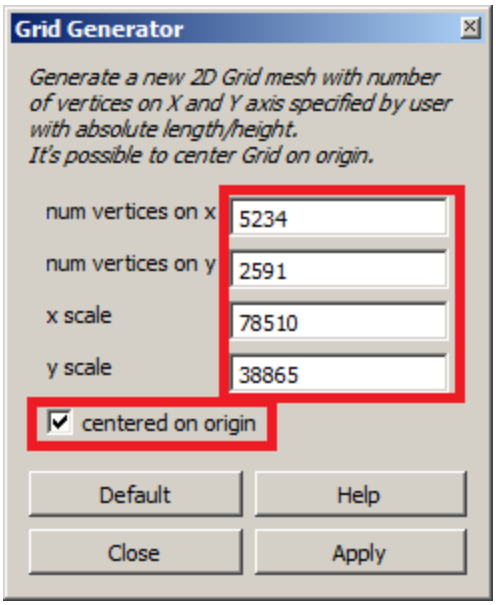


Figure 9. ***MeshLab*** grid generation options

Use the **Translate** filter to move the grid so it overlaps the topography extents and is at elevation **284.915** m, midway between the maximum 570 and minimum -0.17 m. **Select All** the points in the topography layer and use the **Vertex Attribute Transfer** filter, with **Transfer Radius** as close to **290** m as possible. **Delete** the overlapping points on the plane grid, and then raise the plane back to **571** m.

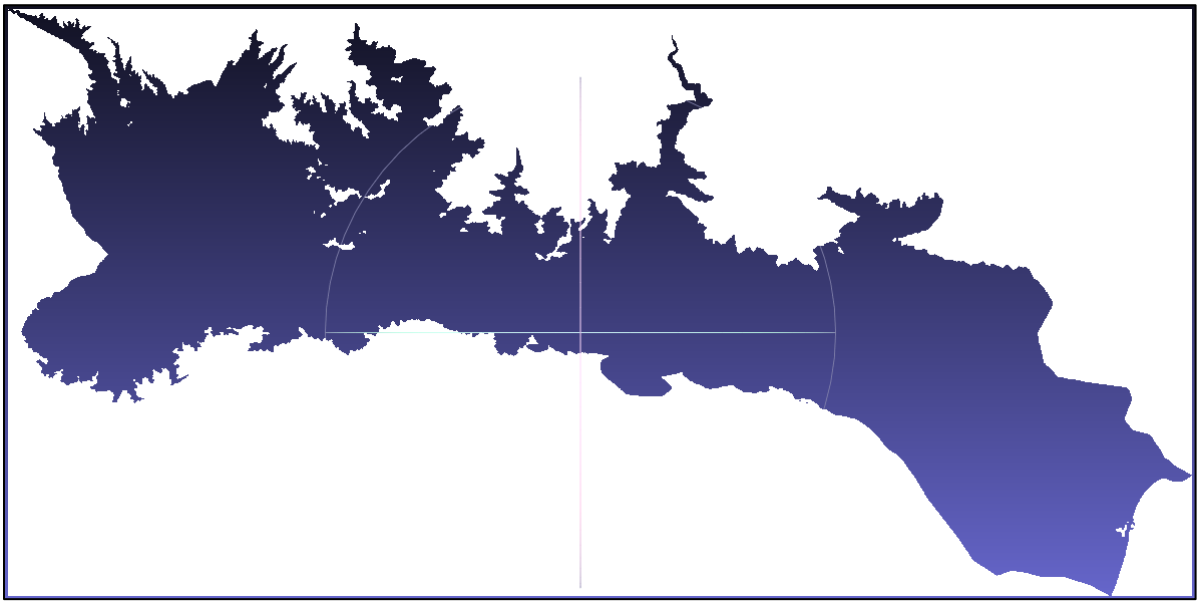


Figure 10. Delete vertices that overlap the topography

Duplicate the plane layer. **Select Vertices** with the  and  tools to select those points that comprise the ocean. **Invert** the selection and **Delete** the land points. Use the **Geometry Function** filter to set the elevation of all points in the layer to **Z = -2** m.

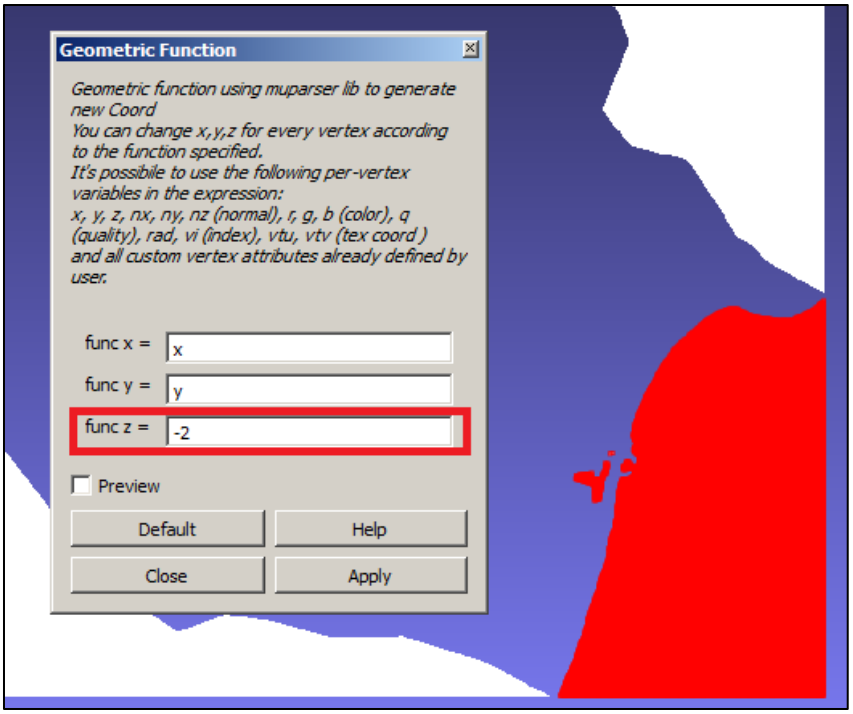


Figure 11. Lower the sea floor to -2 m in ***MeshLab***

On the *original* plane layer, **Select** and **Delete** *only* the ocean region, and set the elevations of all remaining points in that layer to **Z = 571** m (above the flow region). There are now three layers: two horizontal planes (one at -2 m and one at 571 m), and one layer of USGS topography. **Flatten** all three layers in ***MeshLab*** to combine them. **Save** the final combined layer with **no normals**.

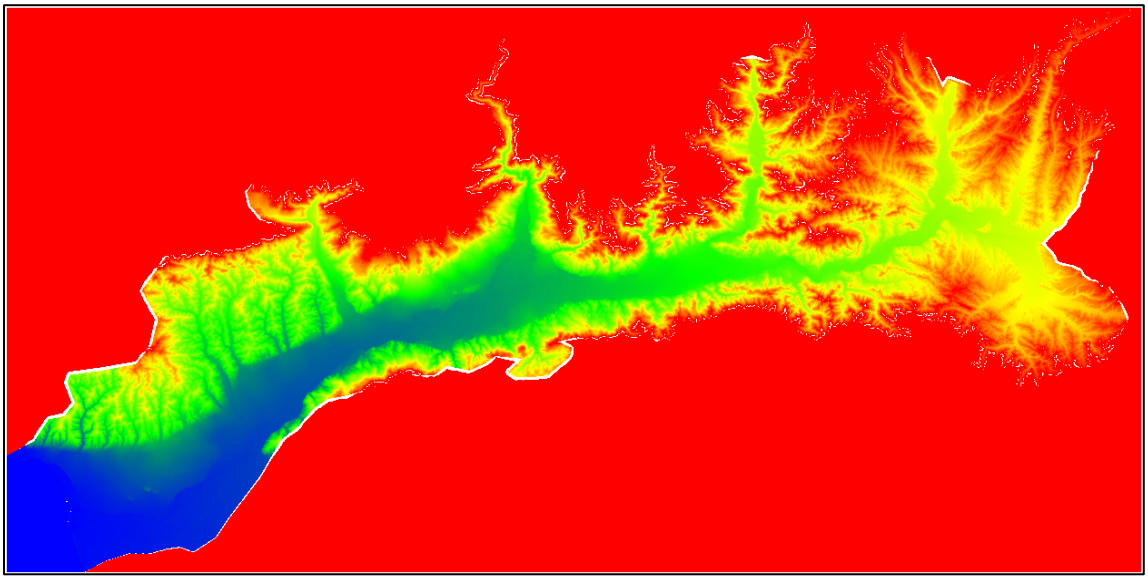


Figure 12. The final surface combines ocean, bounding box, and topography layers

STEP 5 – ESTIMATE THE MINIMUM STL RESOLUTION

Use rules of thumb to find the minimum resolution. The number of surface points in ***TOPO2STL*** output is a function of the area of the rectangular bounding plane and the square resolution stencil. On average, the size of the ***TOPO2STL*** *ASCII*-format *.stl* output in megabytes (MB) can be estimated as:

$$MB_{ASCII} = 4.5 \times 10^{-4} \times \frac{EXTENT_X \times EXTENT_Y}{RESOLUTION^2}$$

Binary-format *.stl* files have the identical data but are 5.42 times smaller than ASCII format:

$$MB_{binary} \approx \frac{MB_{ASCII}}{5.4}$$

The above estimates are accurate within 15% of the actual size. The file can be made smaller by recombining facets to simplify the unused (planar) surfaces. The ratio between high-resolution topography and the entire surface can be estimated by checking the number of points in each layer during Step 4 above:

$$\frac{A_{TOPO}}{A_{TOTAL}} \cong \frac{n_{TOPO\ AFTER\ CLIPPING}}{n_{FINAL\ POINT\ CLOUD}}$$

The file size reduction is a function of the area of the upper surface, the area of the bounding plane, and the aggressiveness of the simplification algorithm. The simplified file size can be estimated from the averages of past projects:

$$MB_{simplified} \cong \frac{MB_{binary}}{2}$$

The estimate is usually within 20% of the actual size of the final, simplified, binary-format file.

Using these rules of thumb, the simplified size of the output file will be 141 MB with 30-m resolution and 565 MB with 15-m resolution. 420 MB is the approximate maximum binary file size that can be repaired with 32-bit STL tools like ***qAdmesh*** and ***netfabb Basic***, so the 15-m resolution file will need to be repaired with a 64-bit tool like ***MeshLab*** or a professional version of ***netfabb***.

STEP 6 – CONVERT THE POINT CLOUD TO .STL WITH *TOPO2STL*

Launch ***TOPO2STL*** from ***FLOW-3D*** > *Utilities menu* > *Topo2STL*. Set **Minimum Z= -250 m** and **Spatial Resolution = 30 m**. Repeat the process with **15 m** resolution.

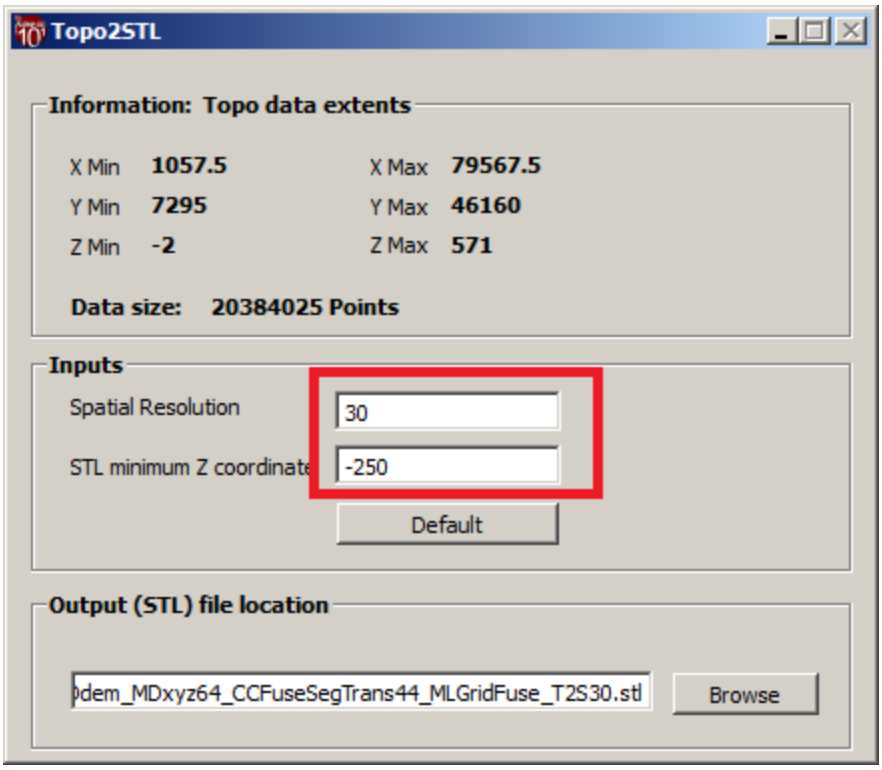


Figure 13. Convert the point cloud to *.stl* with ***TOPO2STL***

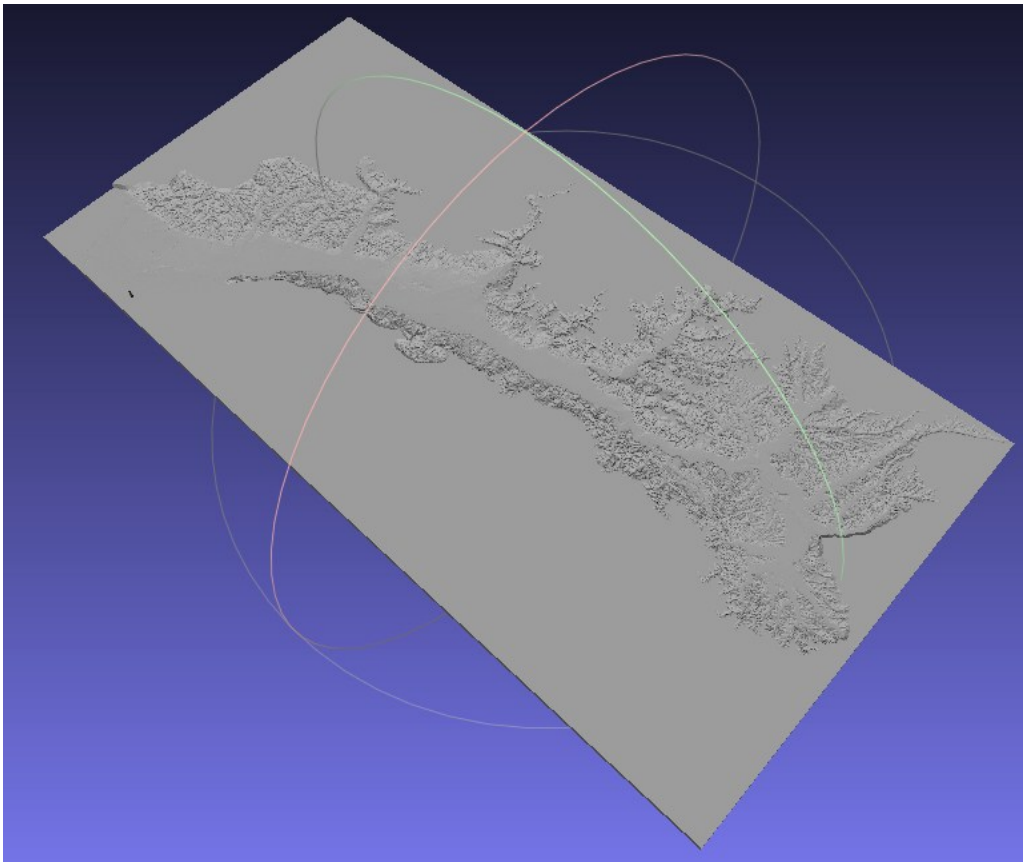


Figure 14. ***TOPO2STL*** output (30-m spacing), viewed in ***MeshLab***

STEP 7 – SIMPLIFY/RECOMBINE FACETS

Import the *ASCII .stl* output into ***MeshLab***. Use the **Conditional Vertex Selection** filter to pick the facets that have all three vertices above **570.9** m. Use the **Quadric Edge Collapse** filter to reduce the bounding box facet count with a target of 90% fewer facets and the other options shown in Figure 16 below.

Figure 15. Select the bounding *.stl* surface in ***MeshLab***

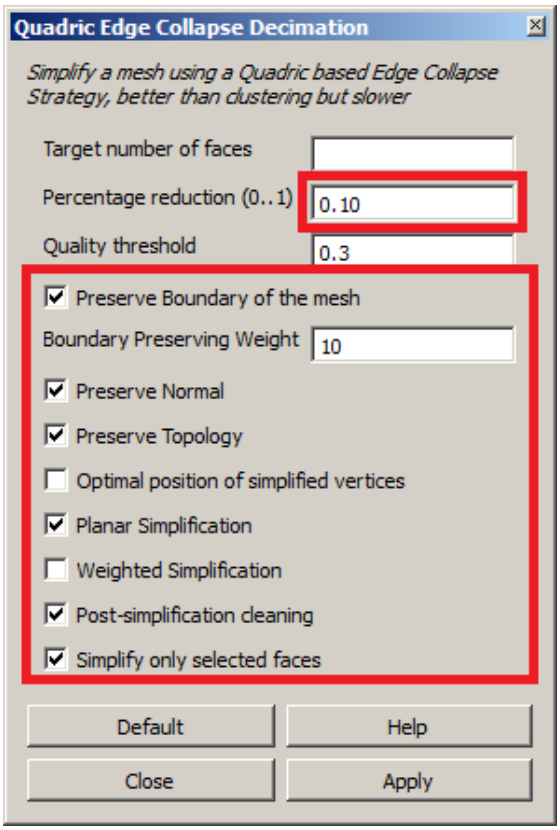


Figure 16. Recombine the bounding surface facets to reduce them by 90%

STEP 8 – DELETE NON-MANIFOLD VERTICES

Select **Render > Show Non-Manifold Vertices** to mark vertices with connecting facets that do not adjoin perfectly. Use the **Select Non-Manifold Vertices** filter to select and then delete these. Repeat as necessary to eliminate all problem vertices, which should all be in the simplified area as shown below.

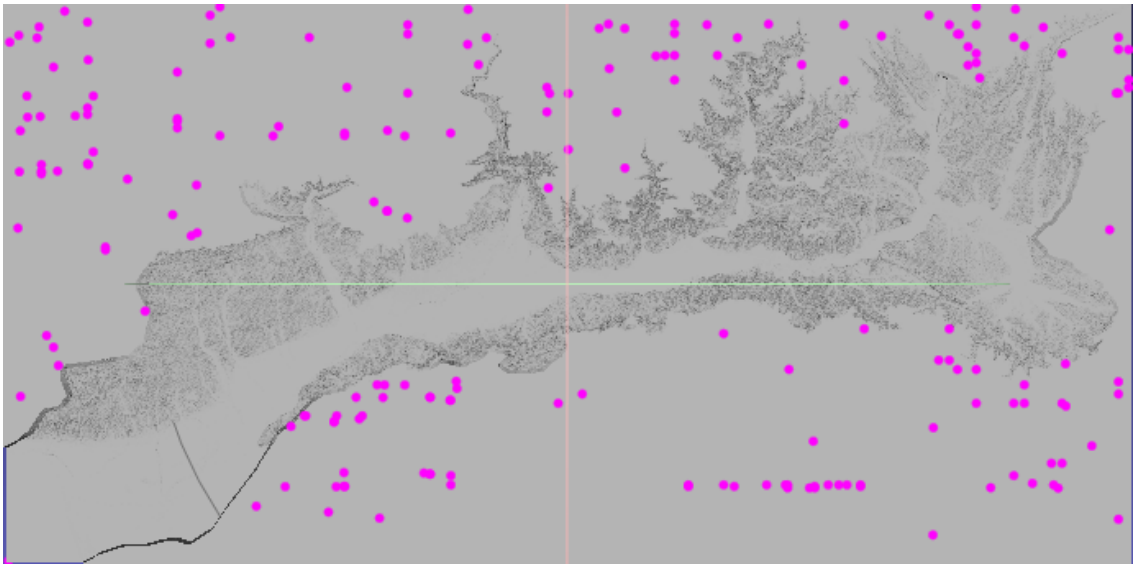


Figure 17. Render, select, and delete non-manifold vertices

STEP 9 – CLOSE TRIVIAL HOLES

Use the **Select Borders** filter to identify holes generated from the first simplification. Use the **Close Holes** filter to auto-close all holes with *fewer than 10* edge facets.

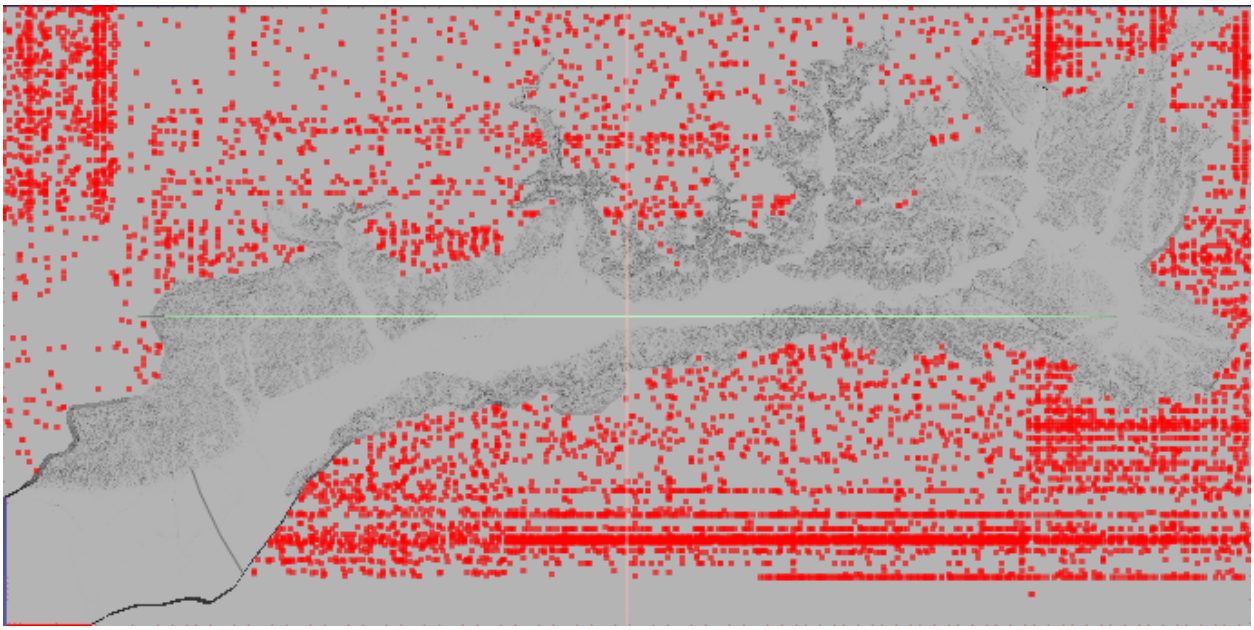


Figure 18. Select and close holes with fewer than 10 edge facets

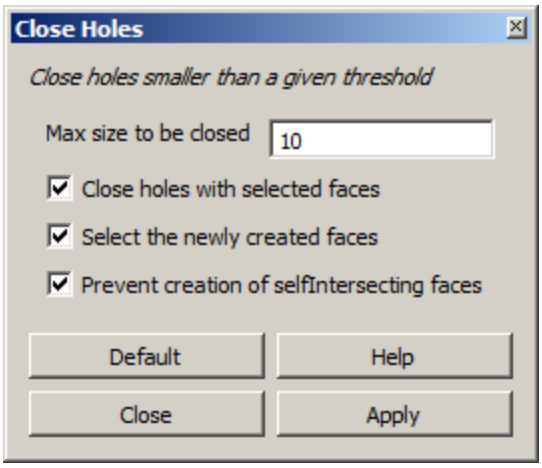


Figure 19. Close small holes

STEP 10 – REPAIR THE BACK FACE

Delete the four non-manifold vertices on the back corners and the facets that share them, if you didn't already in Step 8. Use ***MeshLab***'s **Fill Hole** tool to repair any trivial holes that remain from simplification, and then *iteratively* use the **Self-Intersection** repair option on the single large hole on the back. You will have to re-launch the **Fill Hole** tool after each operation to slowly fill in the back side.

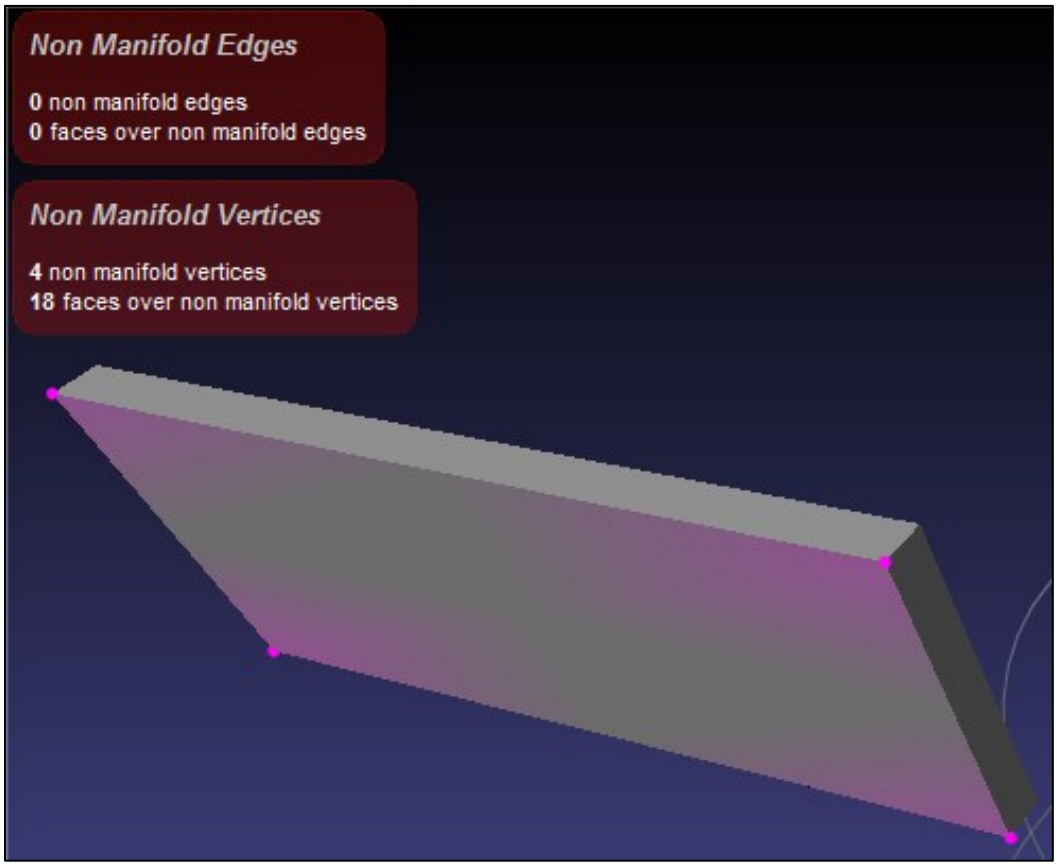


Figure 20. Locate and delete non-manifold vertices with ***MeshLab***

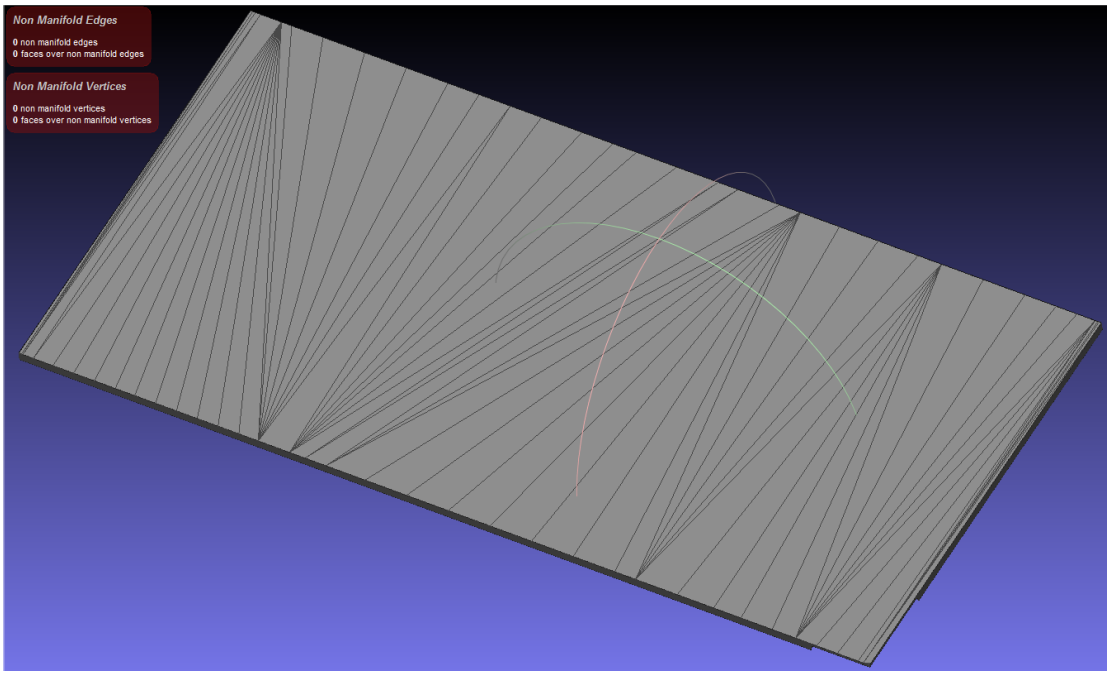


Figure 21. Fill holes and repair the *.stl* bottom with ***MeshLab***

If a smaller *.stl* file size is desired, repeat Steps 7 through 9 until the desired size is achieved. Note that too much simplification can make a watertight surface impossible.

STEP 11 – FIX FACET NORMALS

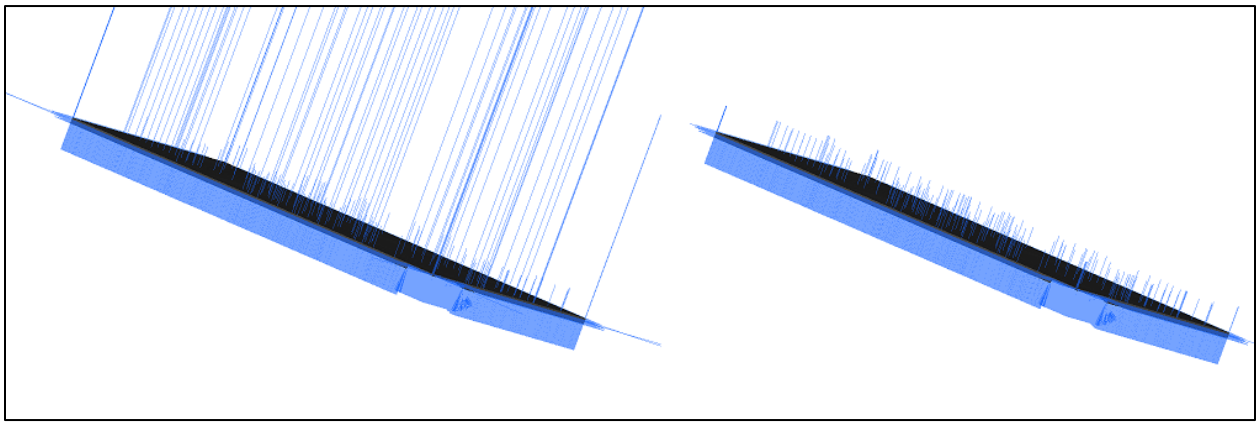


Figure 22. Normals before (left) and after (right)

Use the **Recompute Face Normals** and **Normalize Face Normals** filters. **Save** the output. Launch ***qAdmesh*** from the ***FLOW-3D*** user interface (**Model Setup > Meshing & Geometry > Tools** menu). Check and repair the *.stl*, including the normals, which will be slightly re-adjusted from ***MeshLab***'s orientation. If the file is small enough (≤ 420 MB), re-check the saved final *.stl* with the ***netfabb Basic*** **Standard Analysis**.

STEP 12 – CHECK ACCURACY

Use ***CloudCompare***'s **Compute Cloud/Mesh Distances** tool with **Octree Level = 10** and compare the *.stl* file to the original trimmed point cloud. Do not include the generated 2-D grid that was added to make the upper surface of the *.stl*.

	SIGNED ERROR		ERROR MAGNITUDE	
	MEAN	STANDARD DEVIATION	MEAN ±	STANDARD DEVIATION
NED (USGS 2003) (NED vs. 13,305 geodetic controls)	-0.32 m	2.42 m	2.44 m (RMSE)	~2.3 m (1/2 95th percentile)
15-m ***TOPO2STL*** spacing (*.stl* surface vs. 20.3 x 10^6 points)	-0.10 m	3.6 m	2.4 m	2.6 m
30-m ***TOPO2STL*** spacing (*.stl* surface vs. 20.3 x 10^6 points)	-0.28 m	7.9 m	5.5 m	5.7 m

Table 2. Accuracy statistics for 1/3 arc-second data: original and converted to *.stl*

As of June 2003, the USGS NED data exhibits vertical uncertainty vs. the real world that is similar to the uncertainty generated by converting the NED data to *.stl* format with 15-m spacing. For details of the NED error estimation, see USGS (2006) "Vertical Accuracy of the National Elevation Dataset", online as of 04/29/2014 at http://ned.usgs.gov/documents/NED_Accuracy.pdf. The area of interest (the valleys where the St. Francis flood occurred) has a very low error, as shown in Figure 23. The error magnitude distribution is shown in Figure 24: it is best described as a *Weibull distribution*. The 15-m *.stl* file (Figure 25) shows the least error, and will be used in subsequent ***FLOW-3D*** simulations.

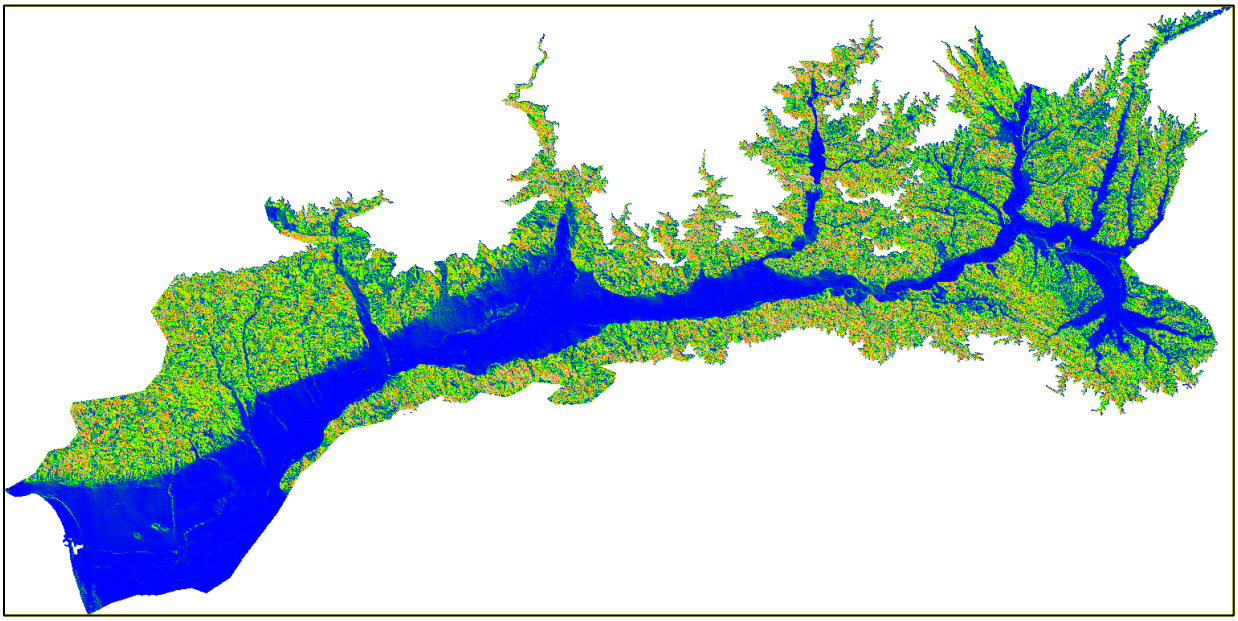


Figure 23. Original point cloud, colored by distance from 15-m resolution *.stl* surface (blue = 0 m, red = ±7.6 m *i.e.* mean + 2 standard deviations)

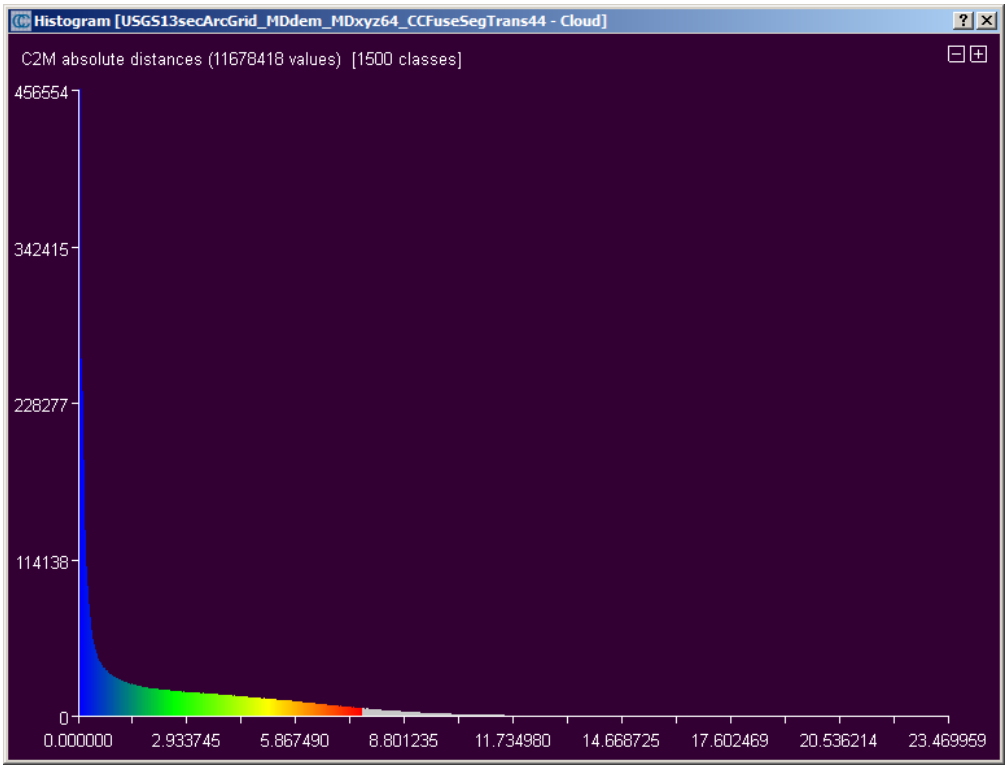


Figure 24. Histogram of point-cloud to 15-m *.stl* facet error. Gray region is outside 2 standard deviations.

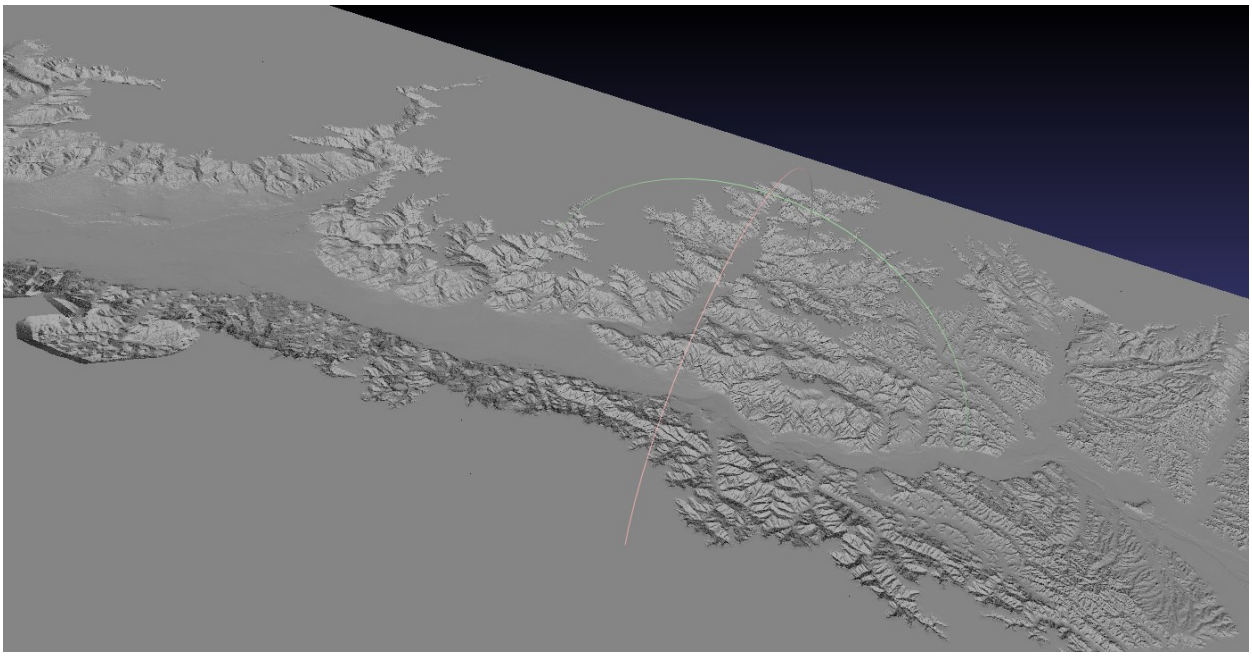


Figure 25. Close-up of the final 15-m *.stl*.