CONVERGED STEPPED SPILLWAY MODELS IN OPENFOAM

by

BRIAN P. SWEENEY

B.S., Kansas State University, 2012
B.S., Kansas State University, 2012

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2014

Approved by:

Major Professor
Mitchell L. Neilsen

.

Abstract

The United States Department of Agriculture (USDA) is currently researching the effectiveness of various earth dam designs and their ability to prevent erosion. This report utilizes experimental results from the USDA experimental hydraulic engineering research unit to develop computational fluid dynamics models using OpenFOAM. Several variations of smooth and stepped dam models are created and analyzed with OpenFOAM on multiple cores using Message Passing Interface. In this report, seven dam designs are analyzed to extract flow velocities and pressures and animations. This data and OpenFOAM models are helpful for determining potential erosion conditions.

Table of Contents

List of Figures iv
List of Tables viii
Acknowledgements ix
Chapter 1 - Introduction 1
Chapter 2 - Background 3
2.1 Earthfill Dams 3
2.2 OpenFOAM 7
2.3 hexBlocker 11
Chapter 3 - Models 17
3.1 Smooth Dam 17
3.2 Smooth Dam with Auxiliary Spillway 18
3.3 Smooth Dam with Principal and Auxiliary Spillway 20
3.4 Smooth Dam with Converging Wall 21
3.5 Stepped Dam 22
3.6 Stepped Dam with Auxiliary Spillway 24
3.7 Stepped Dam with Converging Wall 25
Chapter 4 - Results 27
4.1 Smooth Dam 27
4.2 Smooth Dam with Auxiliary Spillway 31
4.3 Smooth dam with Principal and Auxiliary Spillway 34
4.4 Smooth Dam with Converging Wall 39
4.5 Stepped Dam 43
4.6 Stepped Dam with Auxiliary Spillway 47
4.7 Stepped Dam with Converging Wall 51
4.8 Performance Analysis 55
Chapter 5 - Conclusions 57
References 58

List of Figures

Figure 1. Trapezoidal cross section 1
Figure 2. Homogeneous dam demonstrating seepage limit (Design of Small Dams, 1987, p. 191) 4
Figure 3. Modified homogeneous dam with a rockfill toe (Design of Small Dams, 1987, p. 191)4
Figure 4. Modified homogeneous dam with horizontal drainage blanket (Design of Small Dams, 1987, p. 191) 5
Figure 5. Modified homogeneous dam with inclined blanket and blanket drain (Design of Small Dams, 1987, p. 191) 5
Figure 6. Inverted trapezoidal channel (Design of Small Dams, 1987, p. 349)..... 6
Figure 7. Dam with two trapezoidal channels on the left (Design of Small Dams, 1987, p. 349) 6
Figure 8. Baffled dam utilizing upright baffles to slow flood waters (Design of Small Dams, 1987, p. 361) 7
Figure 9. OpenFOAM workflow (Introduction, 2014). 8
Figure 10. Post-processing in ParaView 11
Figure 11. hexBlocker opening screen..... 12
Figure 12. Result of adding a new hex block..... 12
Figure 13. Extruding a hex block..... 13
Figure 14. Result of editing existing vertices 14
Figure 15. Creating a boundary condition in hexBlocker 14
Figure 16. Setting edge properties 15
Figure 17. Smooth dam mesh created in hexBlocker 17
Figure 18. Smooth dam viewed in ParaView 18
Figure 19. Smooth dam with auxiliary spillway mesh created in hexBlocker 19
Figure 20. Smooth dam with auxiliary spillway viewed in ParaView 19
Figure 21. Smooth dam with principal and auxiliary spillway mesh created in hexBlocker 20
Figure 22. Smooth dam with principal and auxiliary spillway viewed in ParaView 21
Figure 23. Smooth dam with converging wall mesh created in hexBlocker. 21
Figure 24. Smooth dam with converging wall viewed in ParaView. 22
Figure 25. Stepped dam mesh created in hexBlocker..... 23

Figure 26. Stepped dam in ParaView 23
Figure 27. Stepped dam with auxiliary spillway mesh in hexBlocker 24
Figure 28. Stepped dam with auxiliary spillway in ParaView 25
Figure 29. Stepped dam with converging wall mesh created in hexBlocker 25
Figure 30. Stepped dam with converging wall viewed in ParaView 26
Figure 31. Initial conditions for the smooth dam 27
Figure 32. Smooth dam after 3.5 seconds 28
Figure 33. Smooth dam after 7 seconds 28
Figure 34. Smooth dam after 30 seconds 29
Figure 35. Velocity of the smooth dam over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s 30
Figure 36. Dynamic pressure of the smooth dam over time, where the x-axis represents the frame number and the y-axis represents maximum dynamic pressure in pascals at each of the four slices 31
Figure 37. Initial conditions for the smooth dam with auxiliary spillway 31
Figure 38. Smooth dam with auxiliary spillway after 3.5 seconds 32
Figure 39. Smooth dam with auxiliary spillway after 7.5 seconds 32
Figure 40. Smooth dam with auxiliary spillway after 30 seconds 33
Figure 41. Velocity of the smooth dam with auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s ... 33
Figure 42. Dynamic pressure of the smooth dam with auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents the maximum dynamic pressure in pascals at each of the four slices 34
Figure 43. Initial conditions for the smooth dam with principal and auxiliary spillway 35
Figure 44. Contour view of the smooth dam with principal and auxiliary spillway showing water flowing through the pipe 35
Figure 45. Smooth dam with principal and auxiliary spillway after 3.5 seconds 36
Figure 46. Smooth dam with principal and auxiliary spillway after 7.5 seconds 36
Figure 47. Smooth dam with principal and auxiliary spillway after 30 seconds 37

Figure 48. Velocity of the smooth dam with principal and auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s 38
Figure 49. Dynamic pressure of the smooth dam with principal and auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents maximum dynamic pressure in pascals at each of the four slices. 39
Figure 50. Initial conditions for the smooth dam with converging wall 39
Figure 51. Smooth dam with converging wall after 3.5 seconds 40
Figure 52. Smooth dam with converging wall after 8 second 40
Figure 53. Volume view of the smooth dam with converging wall after 30 seconds 41
Figure 54. Velocity of the smooth dam with converging wall over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s ... 42
Figure 55. Dynamic pressure of the smooth dam with converging wall over time, where the x-axis represents the frame number and the y-axis represents the max pressure in pascals at each of the four slices 43
Figure 56. Initial conditions for the stepped dam 43
Figure 57. Stepped dam after 3.5 seconds 44
Figure 58. Stepped dam after 7.5 seconds 44
Figure 59. Stepped dam after 30 seconds 45
Figure 60. Velocity of the stepped dam over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s 46
Figure 61. Dynamic pressure of the stepped dam over time, where the x-axis represents the frame number and the y-axis represents maximum dynamic pressure in pascals at each of the four slices 47
Figure 62. Initial conditions for the stepped dam with auxiliary spillway 47
Figure 63. Stepped dam with auxiliary spillway after 3.5 seconds 48
Figure 64. Stepped dam with auxiliary spillway after 7.5 seconds 48
Figure 65. Stepped dam with auxiliary spillway after 30 seconds 49
Figure 66. Velocity of the stepped dam with auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s ... 50

Figure 67. Dynamic pressure of the stepped dam with auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents maximum dynamic pressure in pascals at each of the four slices 51
Figure 68. Initial conditions for the stepped dam with converging wall 51
Figure 69. Stepped dam with converging wall after 3.5 seconds 52
Figure 70. Stepped dam with converging wall after 7.5 seconds 52
Figure 71. Volume view of the stepped dam with converging wall after 30 seconds 53
Figure 72. Velocity of the stepped dam with converging wall over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s ... 54
Figure 73. Dynamic pressure of the stepped dam with converging wall over time, where the x-axis represents the frame number and the y-axis represents maximum pressure in pascals at each of the four slices..... 55

List of Tables

Table 1. Comparison of how the number of threads on each dam affected the how fast the solver was. The number represents the time, in seconds, required to solve one second of the simulation. 56

Acknowledgements

I would like to thank my major professor, Dr. Mitchell L. Neilsen for all of his support and guidance over the past six years I have been here.

Chapter 1 - Introduction

Earth dams, the most common type of dams in the world, have many applications, including impounding and diverting water (Stephens, 2010, p. 20). According to Stephens, earth dams also are advantageous because they allow local material utilization in dam construction, contain a simple and straight forward design procedure, require comparatively small amounts of fabricated materials and equipment, have less stringent requirements compared to other types of dams, and are more resistant to settlement and movement than more rigid structures. However, earth dams also have disadvantages, such as susceptibility to damage and destruction when in contact with flowing water. Stephens goes on to explain that the design and construction of a dam spillway is the most difficult part of the project, especially for earth dams, and these dams "require continual maintenance to prevent erosion, tree growth, subsidence, animal and insect damage and seepage". Earth dams are typically built with a trapezoidal cross section, with a large base at the bottom and smaller top, as shown in Figure 1.

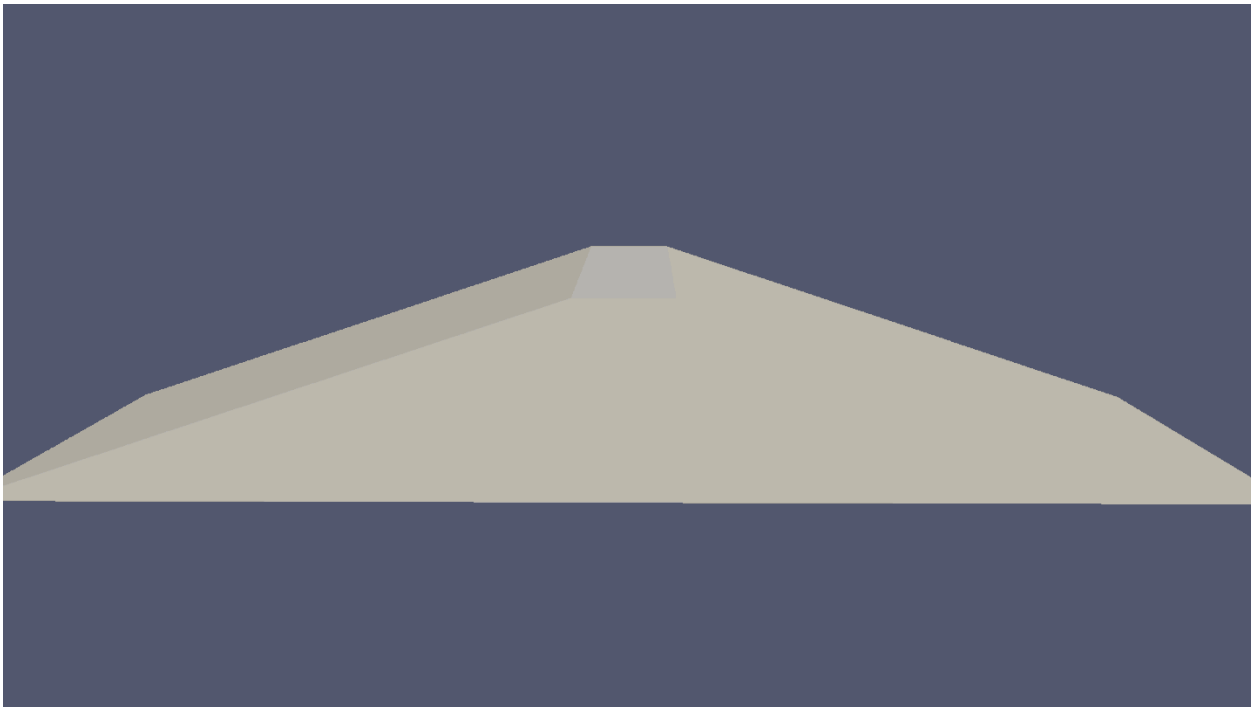

Figure 1. Trapezoidal cross section

During flooding conditions, water will slowly rise up the embankment of the dam. "Overtopping" occurs when "excessive flood flows pass over the embankment" (Stephens, 2010,

p. 94). Erosion generally follows overtopping, often leading to major damage to the dam. According to Foster, Fell, and Spangle, overtopping is one of the primary causes of dam failures (Visser, Tejral, & Neilsen, page 5). To help prevent overtopping from occurring, dams are often built with principal and auxiliary spillways. Stephens (2010) explains that the principal spillway is comprised of a pipe going through the dam, allowing water to flow through it when the reservoir water level is higher than the entrance of the pipe. The size and height of the principal spillway depends on factors such as runoff and streamflow characteristics and flooding frequency and magnitude (Design of Small Dams, 1987, p. 266). The auxiliary spillway is created for emergency situations, such as flood conditions, when the principal spillway cannot keep up with the increasing amount of water (Stephens, 2010, p. 74). The auxiliary spillway, a dam section, designed to allow water to go over it, is more resistant to erosion than the material around it. In order to lower the construction and maintenance costs, the auxiliary spillway typically is not built to the same degree of safety requirements as the principal spillway (Design of Small Dams, 1987, p. 345). An example of reduced safety requirements is that the auxiliary spillway may be designed around the premise that minor damage will occur should they need to be used. Because of their infrequent use, auxiliary spillways can be repaired immediately following a flooding event. The goal of this report is to create a model that can be used to simulate this overtopping. Erosion damage can be simulated with this model at a later point.

Chapter 2 - Background

2.1 Earthfill Dams

Earthfill dams have been in use since the earliest known civilizations (Design of Small Dams, 1987, p. 187). An earthfill dam built in 504 B.C. in Ceylon, Sri Lanka was 11 miles long and 70 feet high and contained 17 million cubic yards of embankments. Currently, earthfill dams are the most common dam type in the world because they can be built with materials in their natural state with little processing. While large earthfill dams have proven successful, small earthfill dams experience failure far more often.

Although some failures are a result of improper design, the majority of failures are caused by improper construction (Design of Small Dams, 1987, p. 187). To properly design an earthfill dam, an architect must account for "actual foundation conditions at the site and materials available for embankment construction". In addition, the dam must be built for specific site geology, instead of being modeled from another successful dam. The concept of a "Rolled-fill" dam has long been one of the most successful templates for a stable earth dam.

In a Rolled-fill dam, the "major portion of the embankment is constructed in successive, mechanically compacted layers" (Design of Small Dams, 1987, p. 188). The embankment material is spread by "motor graders or bulldozers and sprinkled, if necessary, to form lifts of limited thickness having proper moisture content" (Design of Small Dams, 1987, p. 189). Rolled-fill dams are categorized into three species: diaphragm, homogeneous, and zoned.

In diaphragm type dams, a majority of the embankment is built out of pervious (permeable) material, such as sand, gravel, or rock (Design of Small Dams, 1987, p. 189). A thin diaphragm of impermeable material is also "provided to form the water barrier". Diaphragm position varies from a "blanket of the upstream face to a central vertical core" based on needs of the dam and available materials. The diaphragm is generally made of earth, Portland cement concrete, bituminous concrete, and sometimes other impervious material. The diaphragm has a "horizontal distance of less than 10 feet at any elevation or its thickness at any elevation [is] less than the height of the embankment above that elevation". A dam of this type is considered a zoned dam if previous conditions are not met. While dams of these diaphragm and zoned dams have been successful, they are not recommended, because internal diaphragms can crack from

differential movement and are not easily available for inspection or repair. An impervious blanket on the upstream side is also extremely expensive and difficult to construct.

The third type of earthfill dam, a homogenous dam, is composed of only one type of material, such as sand, gravel, or rock (Design of Small Dams, 1987, p. 189). Seepage is inevitable for this type of dam and must be considered. The seepage limit is generally 1/3rd the height of the water level, as seen in Figure 2 (Design of Small Dams, 1987, p. 190).

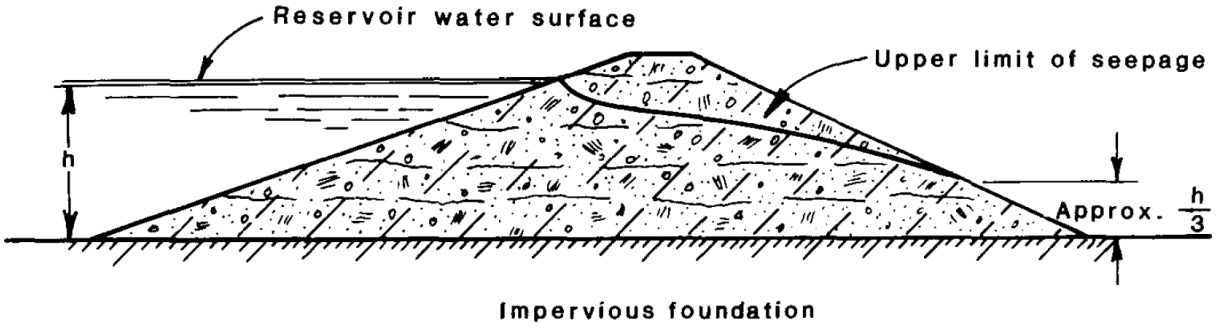

Figure 2. Homogeneous dam demonstrating seepage limit (Design of Small Dams, 1987, p. 191)

Homogeneous dam construction has been replaced by a type of modified homogeneous dam construction in which "small amounts of carefully placed pervious materials control the action of seepage to permit much steeper slopes" (Design of Small Dams, 1987, p. 190). Large rock toes are sometimes used to provide drainage for this seepage, as seen in Figure 3.

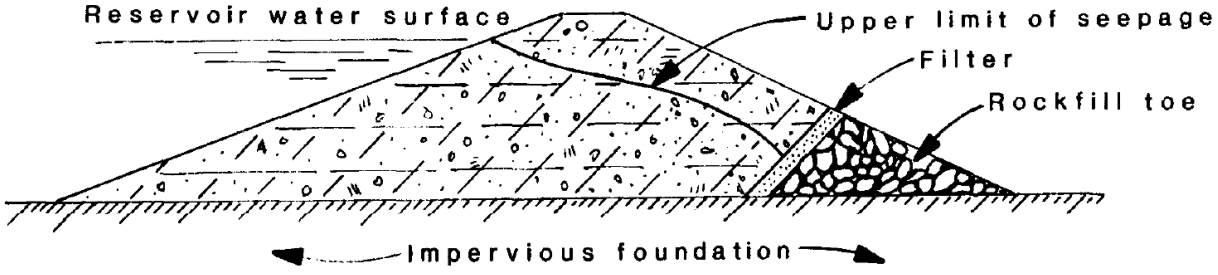

Figure 3. Modified homogeneous dam with a rockfill toe (Design of Small Dams, 1987, p. 191)

If suitable graded materials are available, a horizontal drainage blanket may be used instead of the rock toe, as demonstrated by Figure 4 (Design of Small Dams, 1987, p. 190).

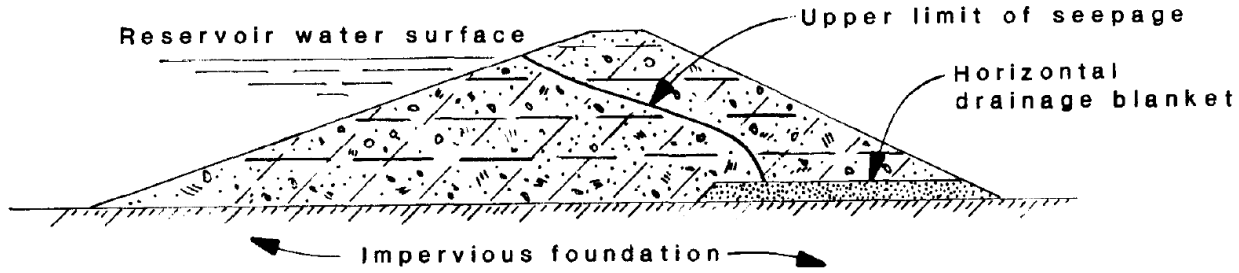

Figure 4. Modified homogeneous dam with horizontal drainage blanket (Design of Small Dams, 1987, p. 191)

To avoid construction defects, an inclined filter drain and horizontal blanket can be used in combination and have become almost standard. (Design of Small Dams, 1987, p. 190). The inclined filter drain can be seen in Figure 5.

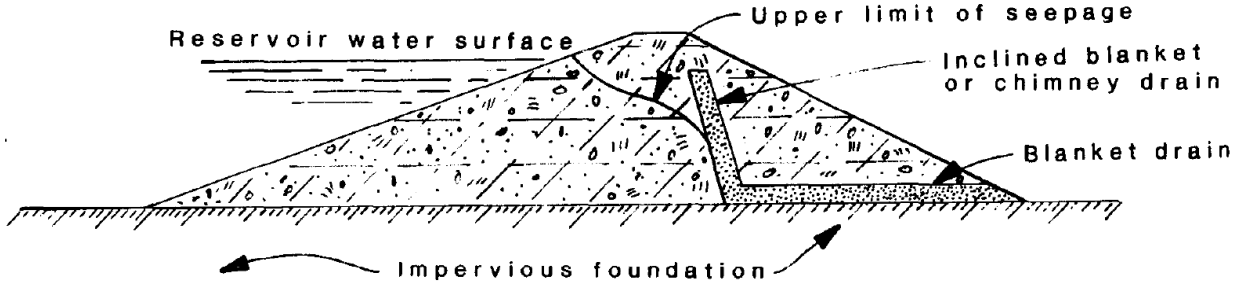

Figure 5. Modified homogeneous dam with inclined blanket and blanket drain (Design of Small Dams, 1987, p. 191)

Another approach to provide drainage in a dam involves placing pipe drains alongside the drainage blanket (Design of Small Dams, 1987, p. 190). However, pipe drains should not be solely relied upon because they clog easily (Design of Small Dams, 1987, p. 190).

Dams often include principal spillways to "release surplus water or floodwater that cannot be contained in the allotted storage space" (Design of Small Dams, 1987, p. 339). However, principal spillways of insufficient capacity can cause overtopping, potentially destroying the earthfill dam. The principal spillway must also be carefully constructed so it does not "erode or undermine the downstream toe of the dam". In addition to a principal spillway, many dams also include an auxiliary spillway to handle the occasional flooding period.

Auxiliary spillways are used infrequently and, therefore, do not require the same safety standards as the rest of the dam. However, auxiliary spillways must be able to forestall failure of the rest of the dam (Design of Small Dams, 1987, p. 345). "Minor damage by scour to an unlined channel, by erosion and undermining at the downstream end of the channel, and by

creation of an erosion pool downstream from the spillway" are often acceptable risks that can be repaired between floods. A typical auxiliary spillway often resembles an inverted trapezoid, as shown in Figure 6.

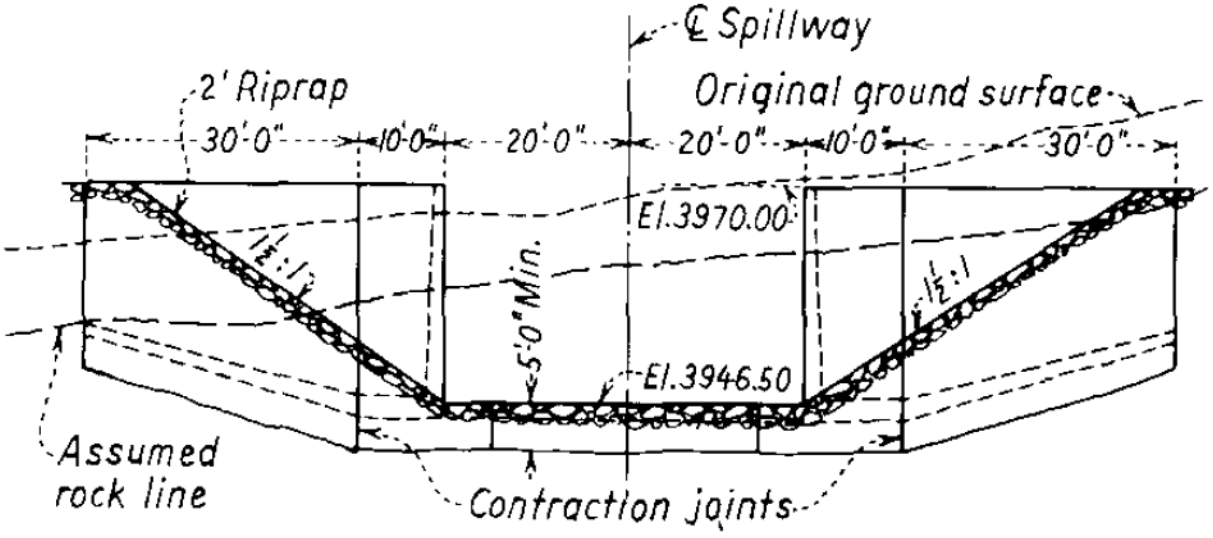

Figure 6. Inverted trapezoidal channel (Design of Small Dams, 1987, p. 349)

Figure 7 illustrates a dam with two channels located on the left side.

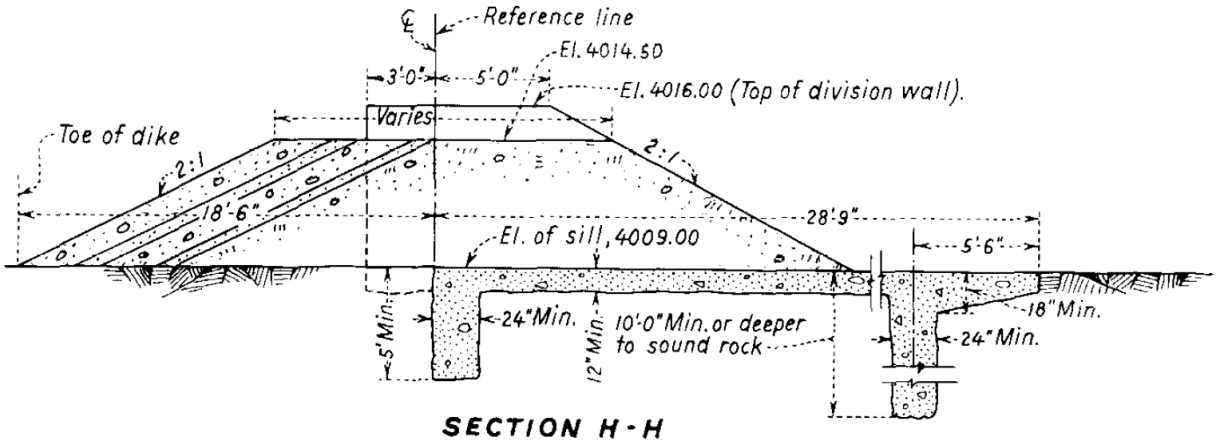

Figure 7. Dam with two trapezoidal channels on the left (Design of Small Dams, 1987, p. 349)

Additional spillway designs help to slow down flow of water and reduce the amount of erosion. Upturned deflectors, cantilevered extensions, flip buckets, baffles, steps, and converging walls can be used to impede the flow of water after it has overtopped a dam (Design

of Small Dams, 1987, p. 351). Often times, erosion can be minimized by fanning the water out into a thin sheet at the point it meets the streambed.

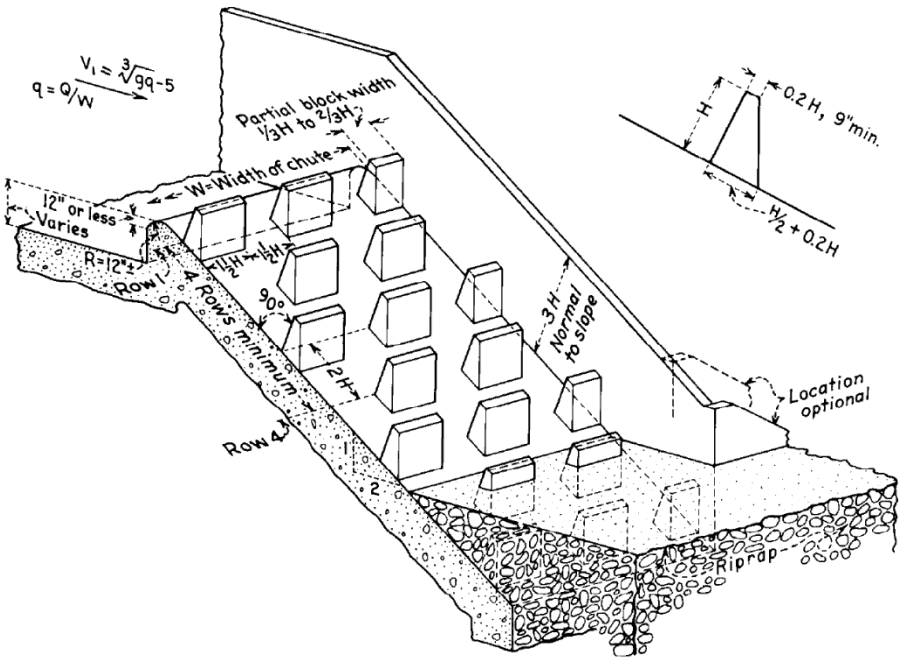

Figure 8. Baffled dam utilizing upright baffles to slow flood waters (Design of Small Dams, 1987, p. 361)

2.2 OpenFOAM

In order to perform fluid dynamics calculations on a dam, a computational fluid dynamics program is required. Open Source Field Operation and Manipulation (OpenFOAM) was chosen for this task (Features of OpenFOAM, 2014). OpenFOAM is a "free, open source CFD software package which has a large user base across most areas of engineering and science, from both commercial and academic organizations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetics". The OpenFOAM package contains 14 solvers for incompressible flows, 15 solvers for multiphase flows, 10 solvers for combustion, seven solvers for buoyancy-driven and heat-driven flows, 11 solvers for compressible flows, and several other solvers for particles and electromagnetics (Standard Solvers, 2014). Since OpenFOAM is open sourced, users are offered complete freedom to modify these solvers to fit their needs and create new solvers (Features of OpenFOAM, 2014). Complementing the solvers, OpenFOAM contains

over 170 utility applications to perform pre- and post-processing tasks. Collaboration of solvers and utilities is illustrated in Figure 9.

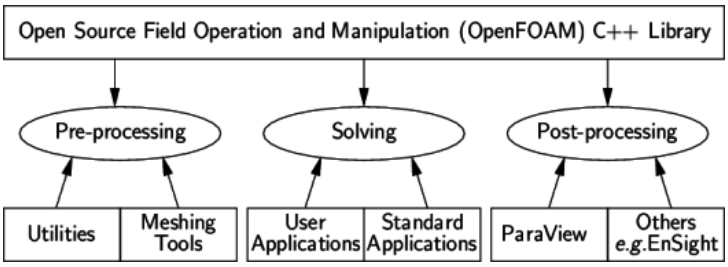

Figure 9. OpenFOAM workflow (Introduction, 2014).

The most important parts of pre-processing include creating a geometry to represent the simulation and setting physical parameters. OpenFOAM uses simple text files to allow a user to input all this information. These text files contain assignments or dictionaries of assignments and dictionaries, as demonstrated in the following:

```
dimensions      [1 -1 -2 0 0 0 0];
internalField   uniform 0;

boundaryField
{
    leftWall
    {
        type            fixedFluxPressure;
        value           uniform 0;
    }

    rightWall
    {
        type            totalPressure;
        p0              uniform 0;
        U               U;
        phi             phi;
        rho             rho;
        psi             none;
        gamma           1;
        value           uniform 0;
    }
}
```

To define the geometry of the situation a user must create a blockMeshDict file containing one dictionary and six assignments. The dictionary in the file is the FoamFile dictionary. All input files for OpenFOAM contain a FoamFile dictionary, explaining what that file does. This specific FoamFile dictionary indicates to the program that it is a blockMeshDict file (BlockMesh, 2014). The first assignment in the blockMeshDict file is called "convertToMeters" which allows a user to apply a multiplier to each point in their mesh to quickly scale the size of it.

The next assignment in this blockMeshDict file is "vertices." This assignment is a list containing all coordinates of every vertex or corner in the geometry. Each vertex contains three numbers separated by spaces and surrounded by a pair of parentheses. For example, "vertices" may give the value (9 1.6 3) representing the x, y, and z positions in a three-dimensional Cartesian coordinate system (BlockMesh, 2014).

The third assignment in this file is "edges" which is a list containing edges that are not blocks in the geometry, such as arcs and splines. Each edge consists of an edge type, two vertices, and interpolation points. For example, "arc 1 5 (1.1 0.0 0.5)" (BlockMesh, 2014).

The fourth assignment is "blocks" which is a list of all hexahedrons, wedges, prisms, pyramids, tetrahedrons, and tetrahedral wedges in the geometry (Mesh Description, 2014). Each block consists of a block type, list of vertices, list of number of cells, and cell expansion ratios. For example, a hexahedron may be defined as "hex (0 1 4 3 6 7 10 9) (20 30 10) simpleGrading (1 1 1)" where "(0 1 4 3 6 7 10 9)" refers to the list of vertices used by that particular block. "(20, 30, 10)" represents the number of cells in x, y, z directions respectively. "SimpleGrading (1 1 1)" indicates that all cells are the same size in each direction (BlockMesh, 2014).

"Boundary" follows the fourth assignment and is a list of dictionaries for all boundaries in the geometry, and each dictionary is a patch for the boundary of the entire geometry. The geometry is broken into patches so that each patch may serve a different function. For example, the user may want to let water flow in or out of one side of the geometry. The dictionaries contain a name, a type, and a list of vertices that comprise the patch (BlockMesh, 2014). For example,

```
boundary
(
    frontWall
    {
    type   wall;
    faces
    (
        (36 37 38 39)
        (37 46 47 38)
        (54 62 63 55)
        (46 54 55 47)
        (62 70 71 63)
        );
    }
)
```

The final assignment is "mergePatchPairs" which is a list of tuples that allows the user to merge two patches together in order to create an internal wall. Each tuple contains a so-called "master" and a "slave" patch. For example,

```
mergePatchPairs
(
  ( lowerWall pipeLeft ) // merge patch pair 0
  ( lowerWall pipeRight ) // merge patch pair 1
);
```

Manual creation of this file is possible, but extremely cumbersome to for any geometry beyond a few simple shapes. Three hexahedrons would force a user to attempt to remember which of the 16 to 24 vertices correspond with each corner and specifying the correct order to avoid the creation of negative volumes. The user also must visualize the desired shapes because the text editor does not illustrate the geometry. For users with access to CAD programs, a utility called "snappyHexMesh" automatically generates a blockMesh (Mesh Generation, 2014).

Next, the user must set the physical parameters similarly to the blockMesh file. OpenFOAM comes with at least one example for each solver, so a user can easily use these as a starting point. The OpenFOAM website also contains three tutorials, including a tutorial about a simple, two-dimensional dam, which was the starting point for this project.

Once the simulation is created the user can run the solver and begin post-processing. OpenFOAM comes with a program called "ParaView" as a companion simulation visualization tool, as demonstrated in figure 10 (The ParaView Post-Processor, 2014). However, OpenFOAM

also supports several other third-party post-processing products, such as “Fluent” and “Ensight,” and provides tools to convert to their data formats.

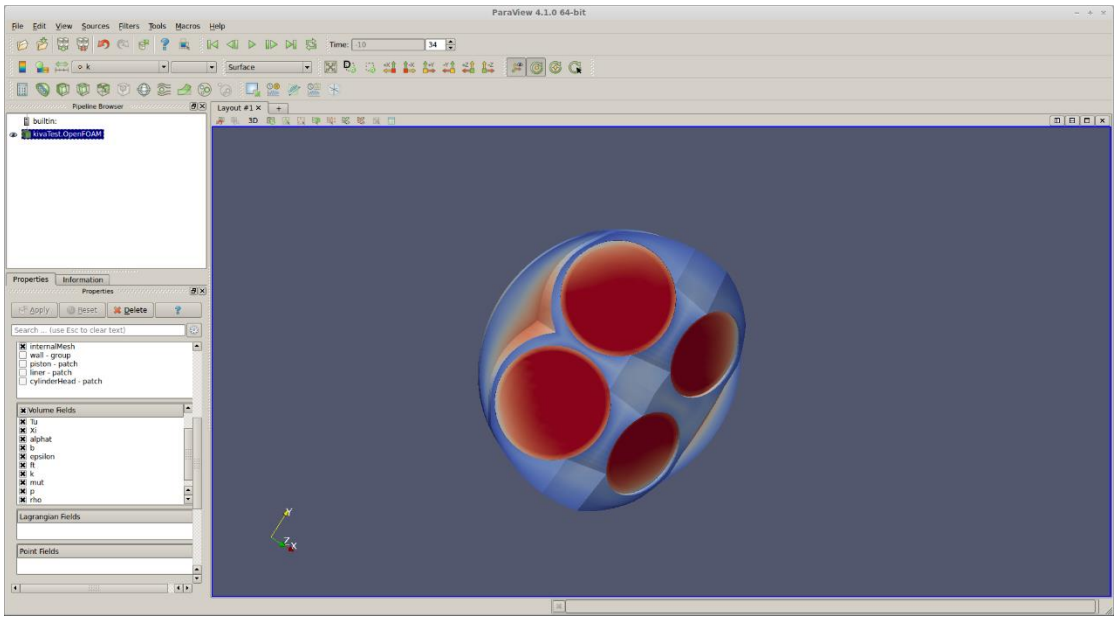

Figure 10. Post-processing in ParaView

2.3 hexBlocker

For users without access to CAD programs, “hexBlocker” can be used to create blockMeshDict files required for this project. HexBlocker, is an unofficial tool created by Nicholas Edh and shown in Figure 11, is released with a GPL license (HexBlocker, 2014). While it does not support all features contained in a blockMeshDict file, hexBlocker allows a user to visually create the file one hex block at a time. HexBlocker tracks all vertices in the mesh, which hex blocks use certain vertices and maintaining correct order. HexBlocker also keeps track of which faces belong to a particular boundary.

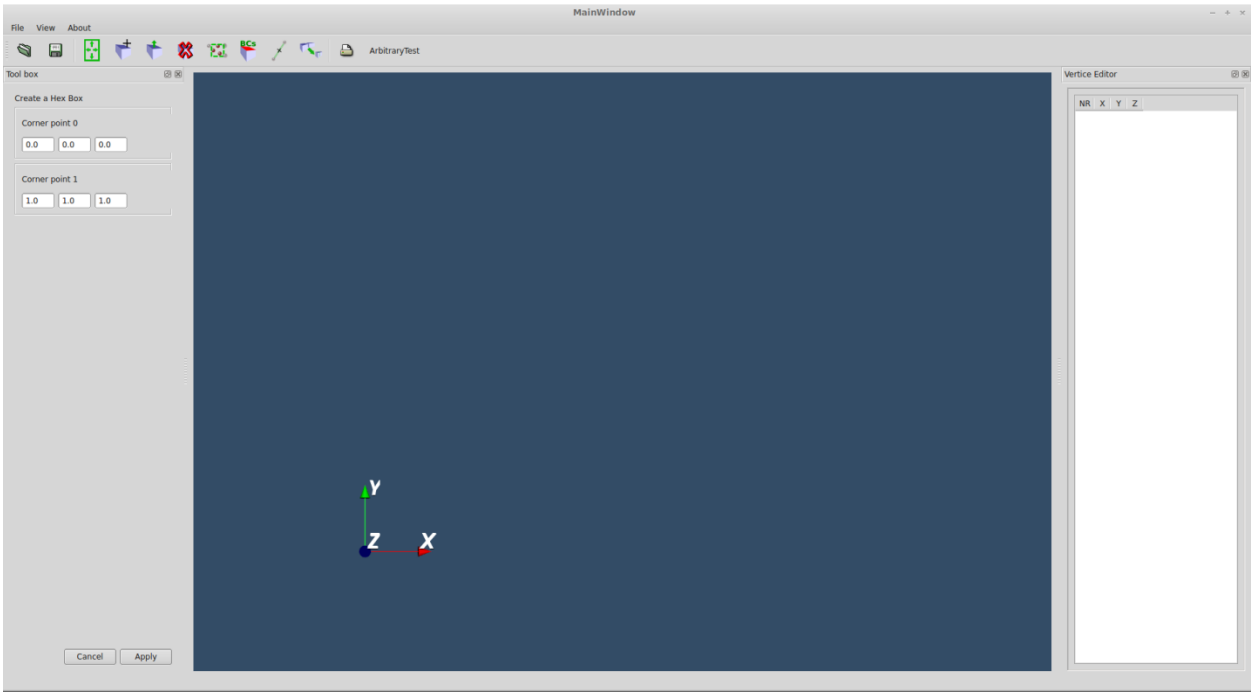

Figure 11. hexBlocker opening screen

HexBlocker is also able to add new hexahedron blocks to the mesh. After clicking the "add hex block" button, a dialog appears asking for the coordinates of two opposing corners of the block. After inputting and applying the coordinates, the new block is rendered in the display area and the vertices are added to the list in the vertex editor on the right, as shown in Figure 12.

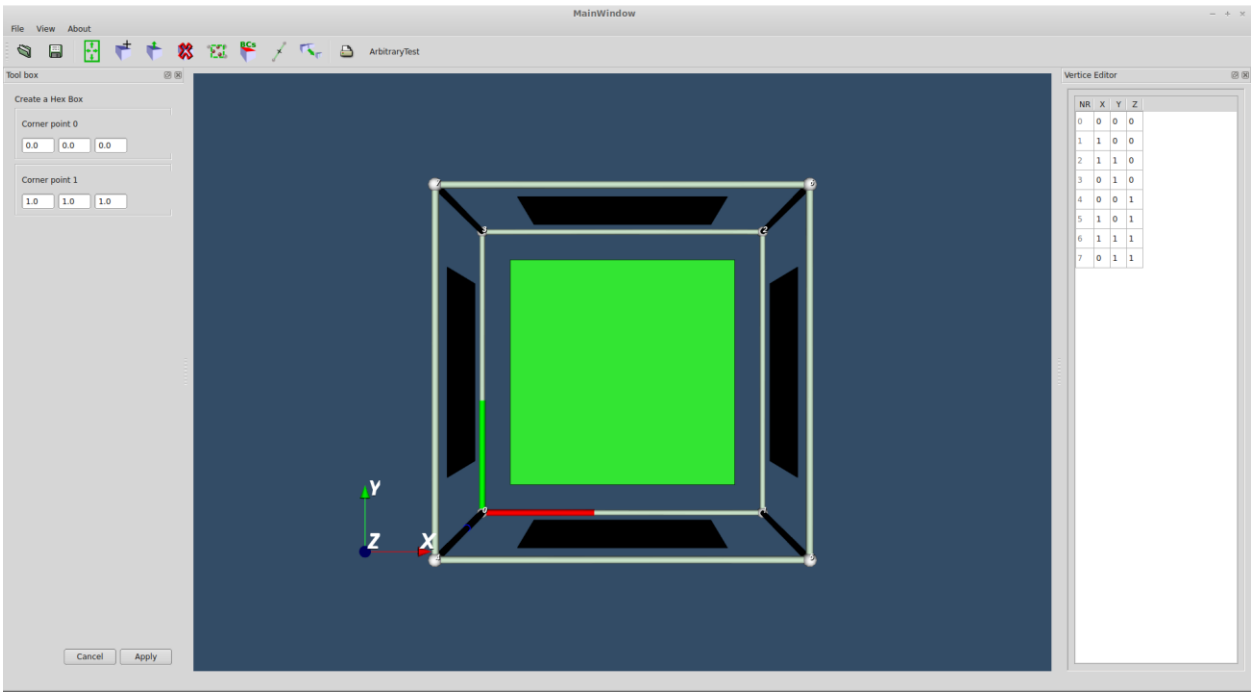

Figure 12. Result of adding a new hex block

When a block is added in this method, all faces on the block are marked as external. If a user adds another block next to the original block, two external faces will exist between them, inhibiting flow between the two blocks. The user can manually merge them into an internal face by using the "merge face button" in hexBlocker or by editing the blockMeshDict file and creating mergePatchPairs. However, the "extrude tool" in hexBlocker allows for quicker merging because a user can click the "extrude face" button and choose the face they would like to extend. A dialog box prompts the user to specify how far the new block should protrude. The program then updates the display area with the new block and automatically connects it to the original block, as shown in Figure 13.

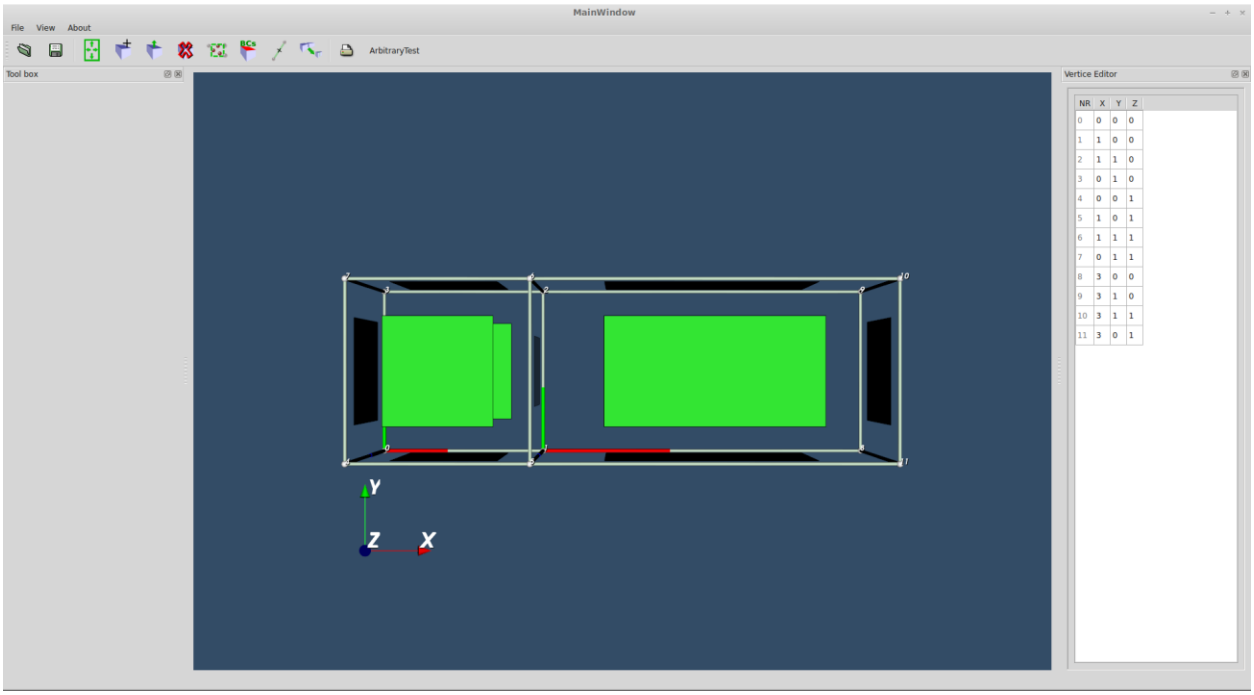

Figure 13. Extruding a hex block

HexBlocker can do more than make regular hexahedrons, however. Once a block is created, the vertices can be edited in the vertex editor by inputting a new coordinate for the vertex of interest, as shown in Figure 14.

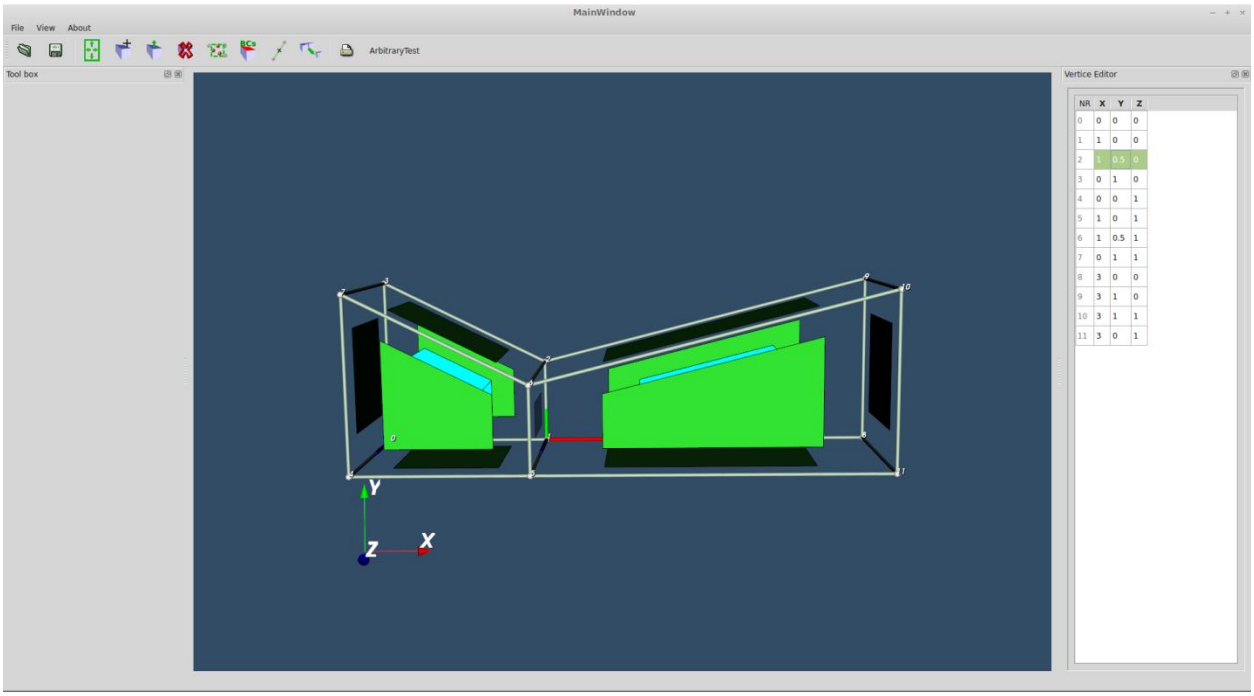

Figure 14. Result of editing existing vertices

Once the shape of the mesh is created, a user can use the "boundary conditions" tool to create and set the type of new boundaries in the geometry. The user can select which faces belong to each boundary and observe which faces belong to the boundary, as seen in Figure 15.

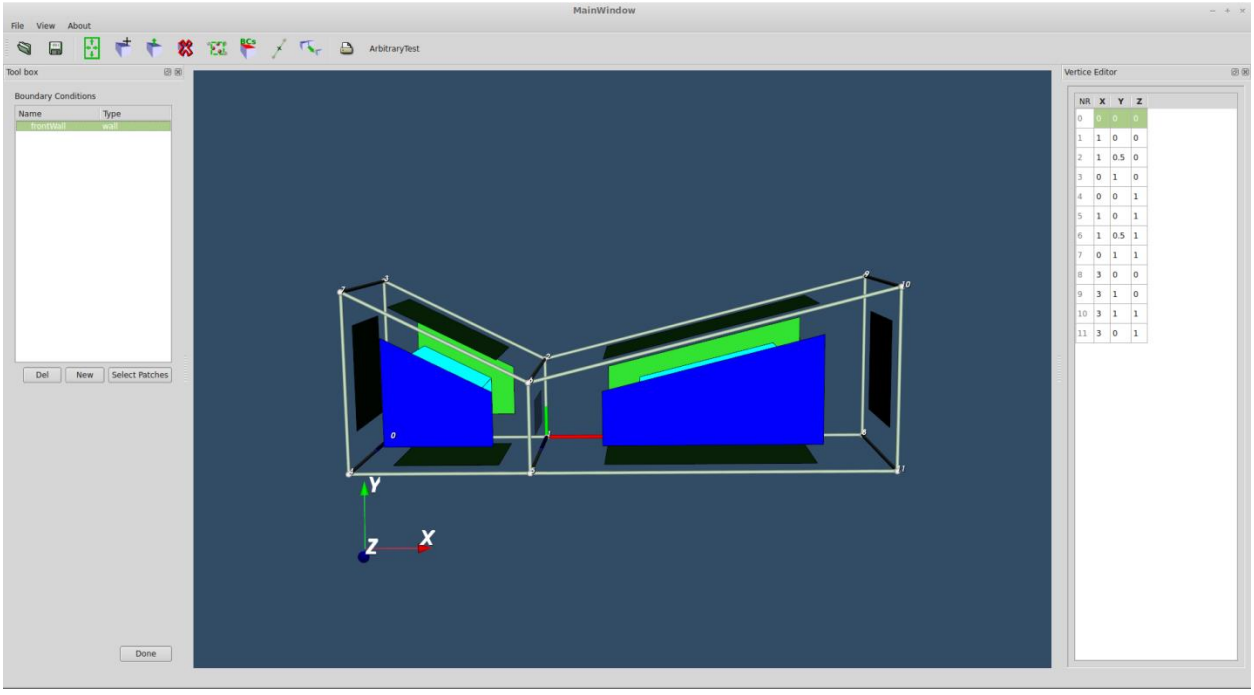

Figure 15. Creating a boundary condition in hexBlocker

HexBlocker also allows a user to set the level of detail on the mesh by using a "set edge properties" button. After selecting the edge, the user is prompted to set the number of cells. This value is recursively propagated to all neighboring cells throughout the mesh, as shown in Figure 16.

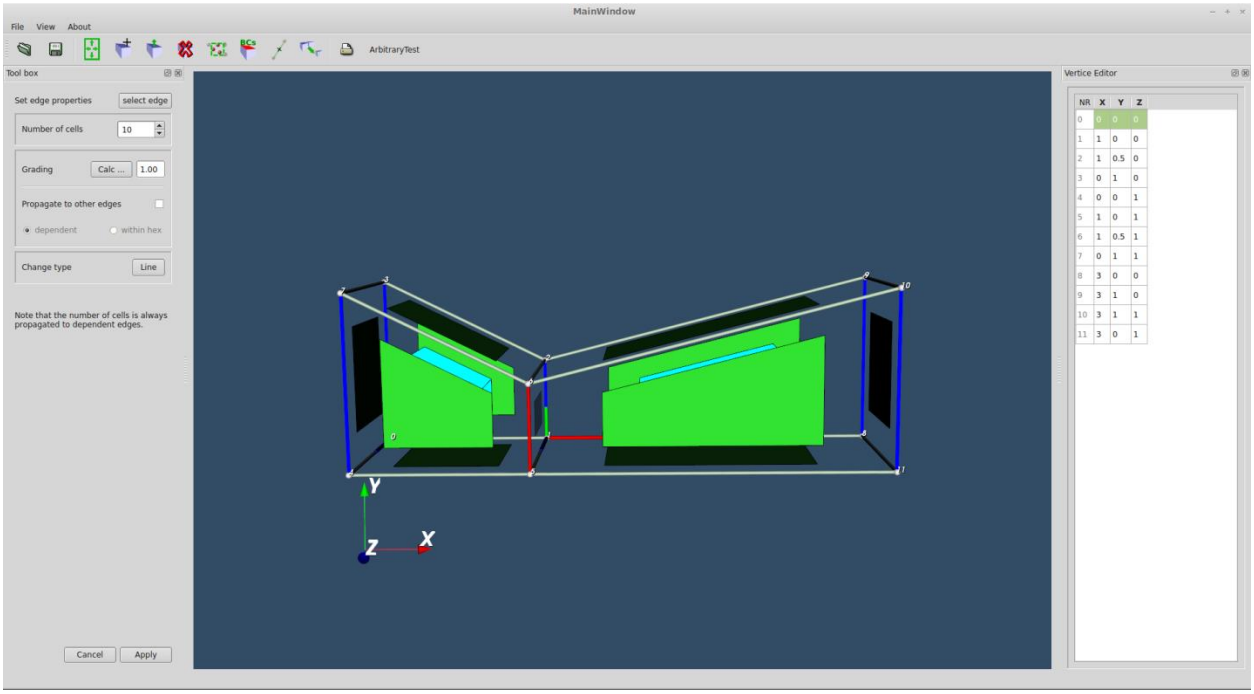

Figure 16. Setting edge properties

Despite its usefulness, hexBlocker also has its limitations. For example, it is unable to create mergePatchPairs, which were used several times during this project. A user must take great care to store mergePatchPairs somewhere else when using hexBlocker to create a blockMeshDict file, since the mergePatchPairs will be removed from the file when the user clicks save. The mergePatchPairs can then be added in later using a text editor. Another limitation of hexBlocker is that it automatically duplicates the list of patches on boundaries if a user attempts to add a patch to an already defined boundary. For example, if a boundary already has patches a, b, and c defined when the blockMeshDict file is opened and a user tries to add patch d, the blockMeshDict file will contain a, b, c, a, b, c, and d for that boundary, causing the blockMesh utility to throw a "FOAM FATAL ERROR". The duplication can be corrected by opening up the blockMeshDict file in a text editor and manually removing the duplicates by hand. Another solution is to remove all patches from the boundary and re-add them all. Another limitation of hexBlocker is that it is unable to create any objects beside hexahedrons. While arcs

and splines are out of the question, shapes that can be categorized as blocks can be faked by collapsing vertices on themselves. For example, in order to create a prism, two pairs of vertices can be placed at identical coordinates or to create a pyramid, four vertices of a block can be placed at the identical coordinates.

Chapter 3 - Models

Before any computational fluid dynamics calculations could be conducted for this study, meshes had to be created for various types of dams. The interFoam solver from OpenFOAM requires a mesh that represents the atmosphere around the model rather than the model itself. Therefore, each of the following meshes were created with this in mind.

3.1 Smooth Dam

In order to test the effectiveness of principal spillways, auxiliary channels, converging walls, and steps, a simple dam was created without them as a control. As a basis for all the dam types tested, each dam had a horizontal distance of six meters from the heel to the crest. Each crest was one meter in depth with a horizontal distance of six meters from crest to toe, a total span of five meters and a maximum height of two meters. For modeling the atmosphere around the dam, each mesh had a two meter section in front of the dam to allow water to pool on the stream side, a three meter section after the dam to allow water to flow past the dam, and a height of five meters. The wall upstream from the dam was always marked as an inlet, forcing water to continuously flow into the mesh. The opposite wall was marked as an outlet to allow the water to continue moving after overflowing the dam. The ensuing mesh was created in hexBlocker using five blocks, as shown in Figure 17.

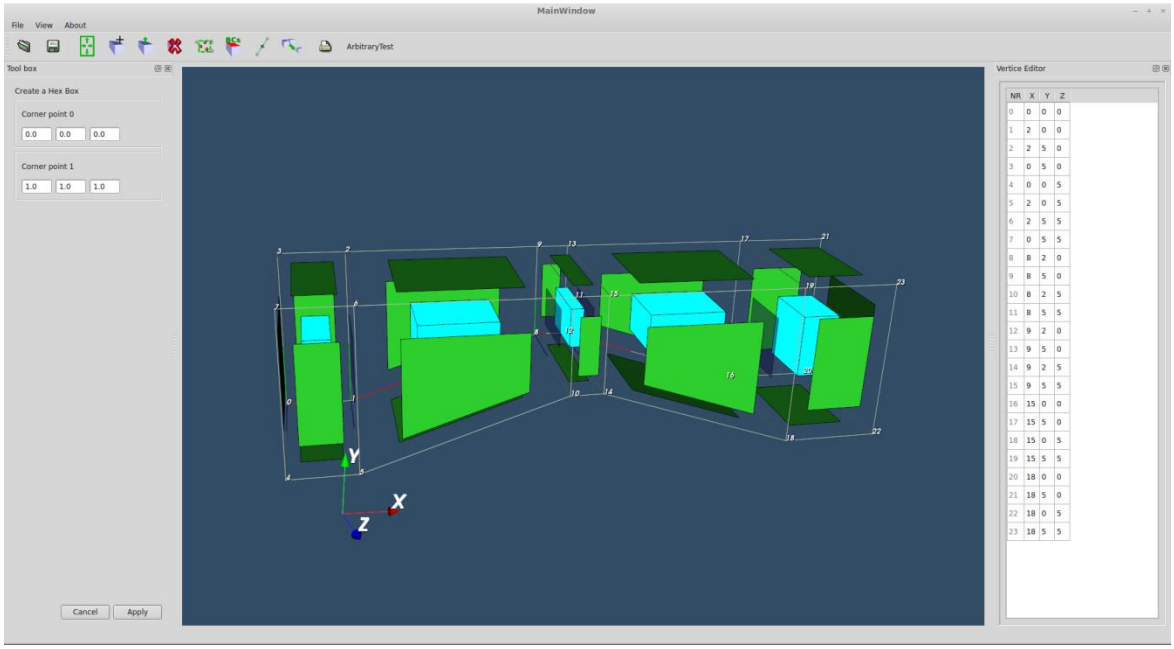

Figure 17. Smooth dam mesh created in hexBlocker

The blockMesh utility was then run on the created blockMeshDict file that and the dam was ready to be used by OpenFOAM and ParaView, as shown in Figure 18.

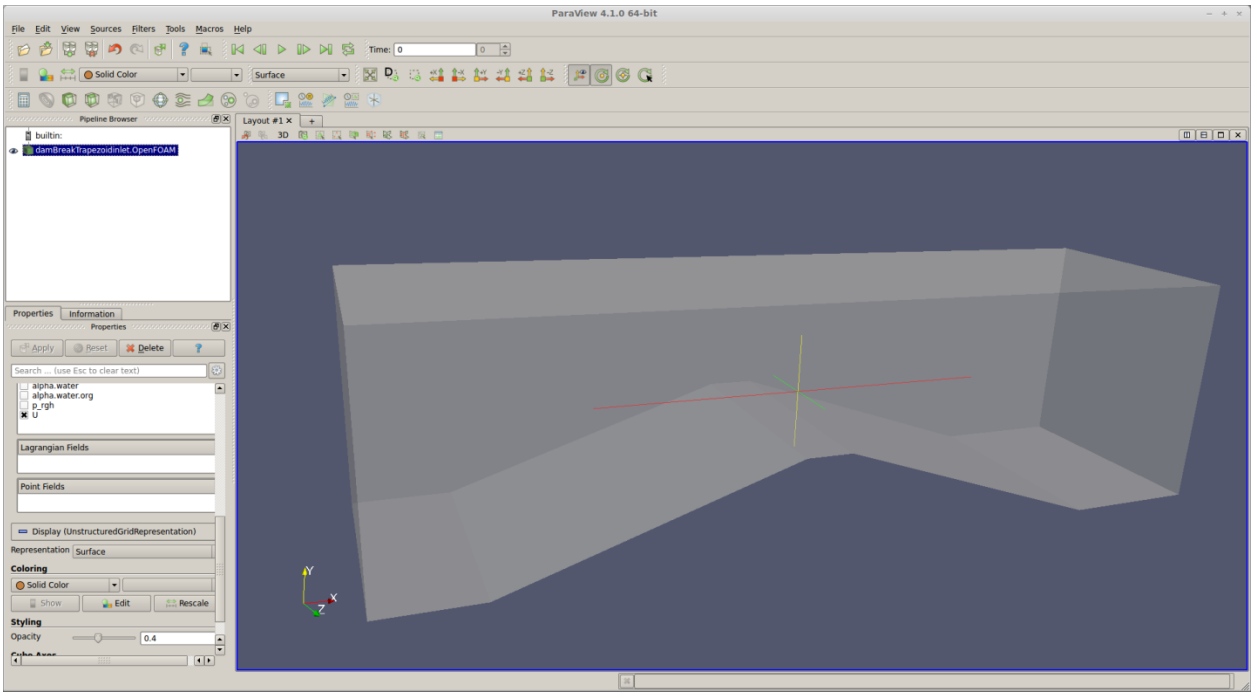

Figure 18. Smooth dam viewed in ParaView

3.2 Smooth Dam with Auxiliary Spillway

In order to test the effectiveness of an auxiliary spillway and provide a model on which to apply erosion in the future, a trapezoidal channel was added to the middle of the dam created in Chapter 3.1. This channel lowered the height of the middle section of the crest from two meters to 1.6 meters. The lower base of this trapezoidal channel spanned one meter on the bottom and three meters on the top. The mesh was created in hexBlocker using twenty-five blocks, as demonstrated in Figure 19.

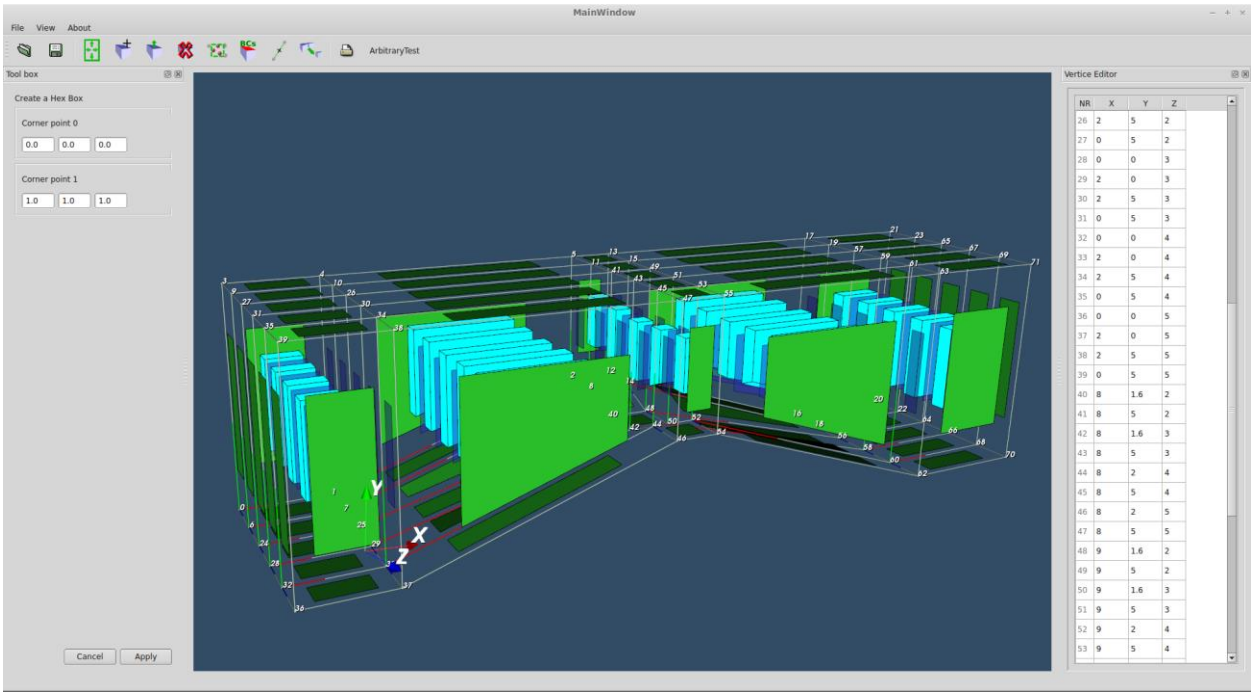

Figure 19. Smooth dam with auxiliary spillway mesh created in hexBlocker

The blockMesh was run and the mesh was ready to be used, as illustrated in Figure 20.

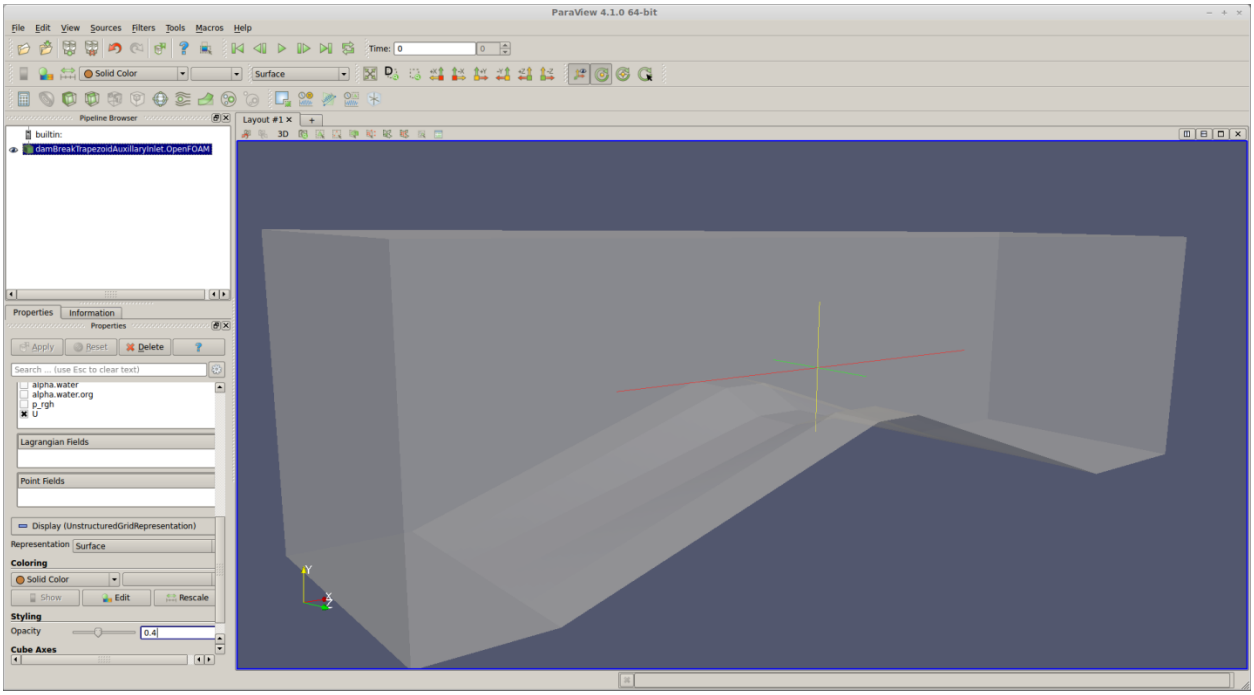

Figure 20. Smooth dam with auxiliary spillway viewed in ParaView

3.3 Smooth Dam with Principal and Auxiliary Spillway

The next test required a principal and auxiliary spillway to work together since they are commonly paired together in the real world. A principal spillway in the form of a pipe had to be added to the dam in Chapter 3.2. A pipe .01 meters in diameter was added to the trapezoidal channel at a height of .8 meters on the upstream side and a height of .5333333 meters on the downstream side in hexBlocker, using twenty-six blocks, as shown in Figure 21.

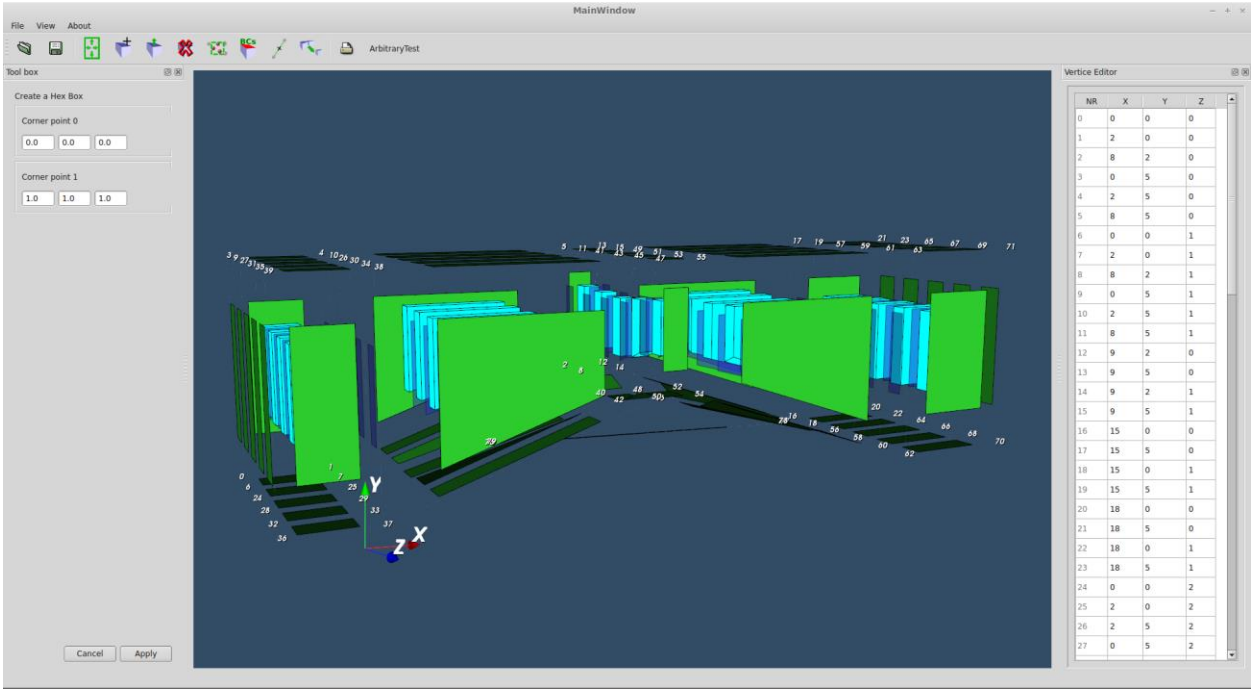

Figure 21. Smooth dam with principal and auxiliary spillway mesh created in hexBlocker

Then the blockMeshDict file had to be edited to allow water to pass from the ramp to the inside of the pipe. This was done by creating a mergePatchPair with each end of the pipe and the ramp to which it was attached, as can be seen in Figure 22.

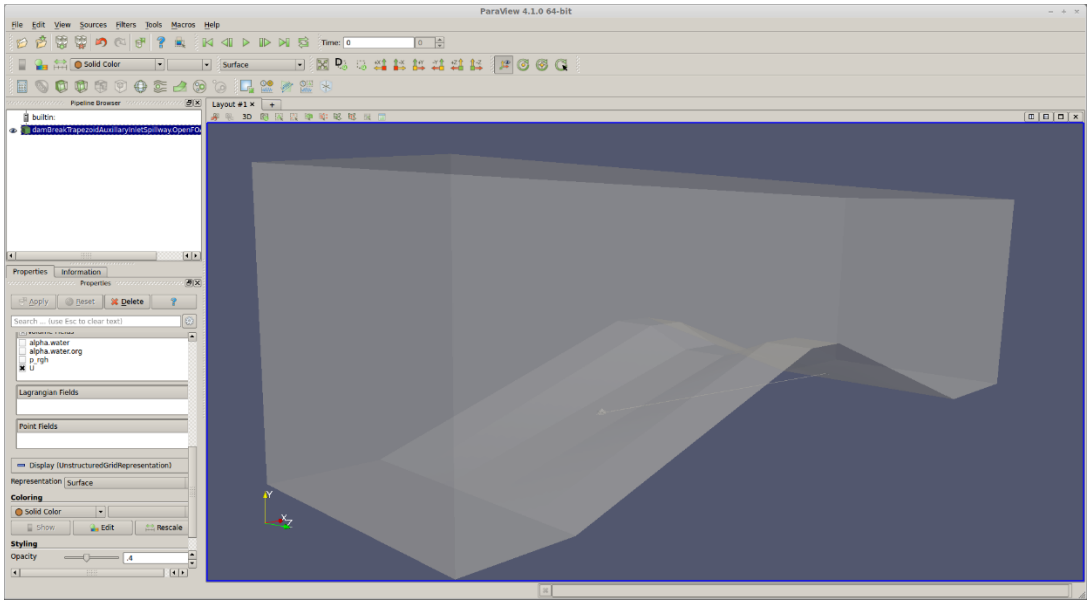

Figure 22. Smooth dam with principal and auxiliary spillway viewed in ParaView

3.4 Smooth Dam with Converging Wall

For the last smooth dam, a converging wall was added to the dam in Chapter 3.1 at an angle of 52 degrees. This dam had a span of five meters at the top of the crest and a span of 1.09357 meters at the toe. The mesh was created in hexBlocker using ten blocks, as shown in Figure 23.

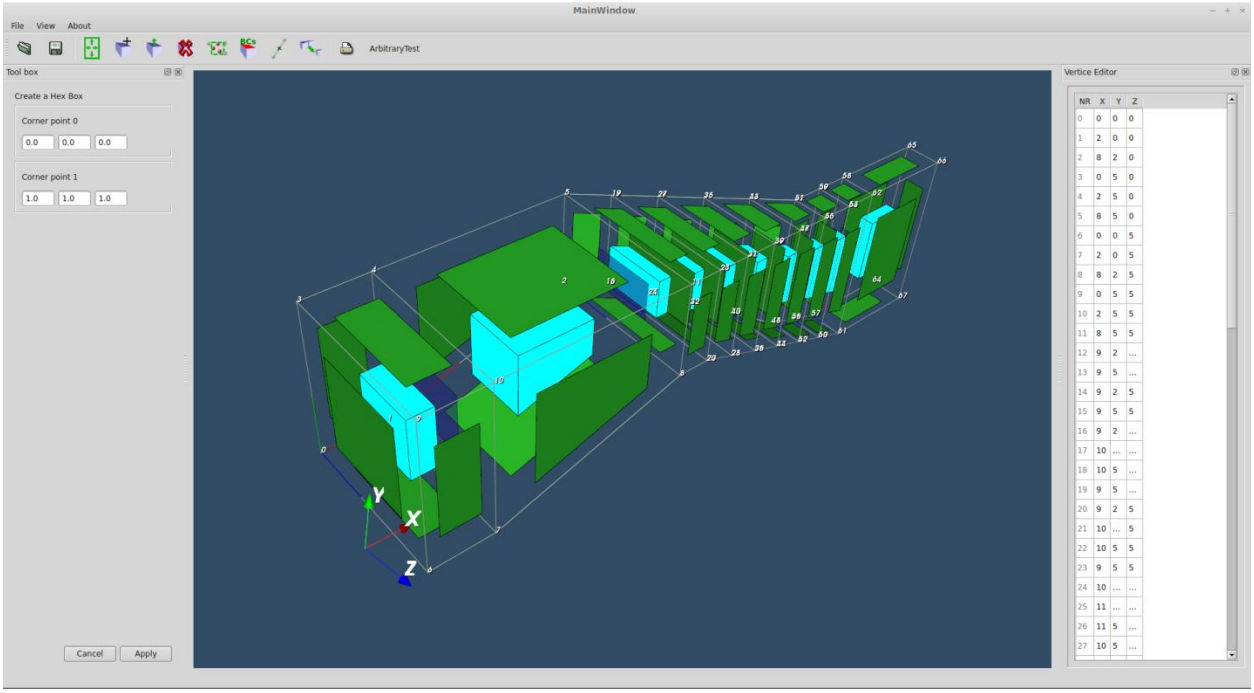

Figure 23. Smooth dam with converging wall mesh created in hexBlocker.

After creating a mergePatchPair for each section of the downstream ramp and running blockMesh, the mesh was ready for use, as shown in Figure 24.

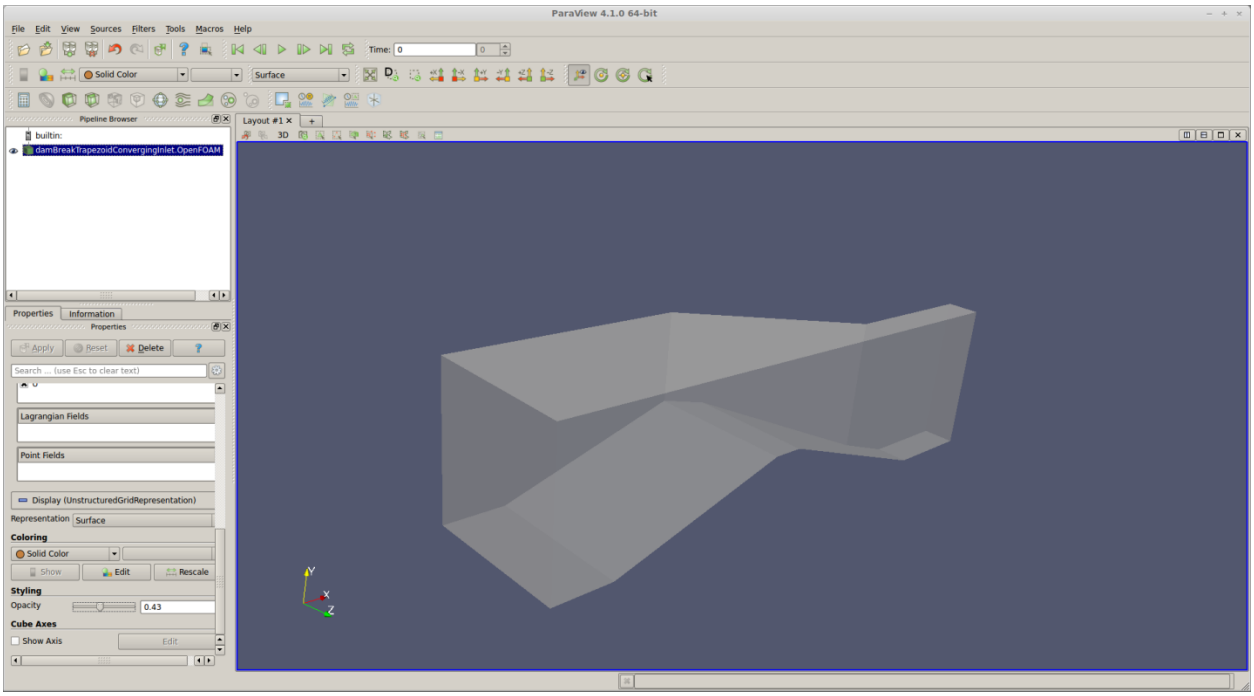

Figure 24. Smooth dam with converging wall viewed in ParaView.

3.5 Stepped Dam

On some dams with auxiliary spillways, steps are used to slow water flowing over them. To test their effectiveness, a stepped dam without an auxiliary spillway was created. Using the dam created in Chapter 3.1 as a starting point, the downstream ramp was removed and replaced by stairs that were one meter in length and ⅓ meters in height, meaning that the horizontal distance from the crest to the toe remained only six meters long. This mesh was implemented in hexBlocker using thirty blocks, as can be seen in Figure 25.

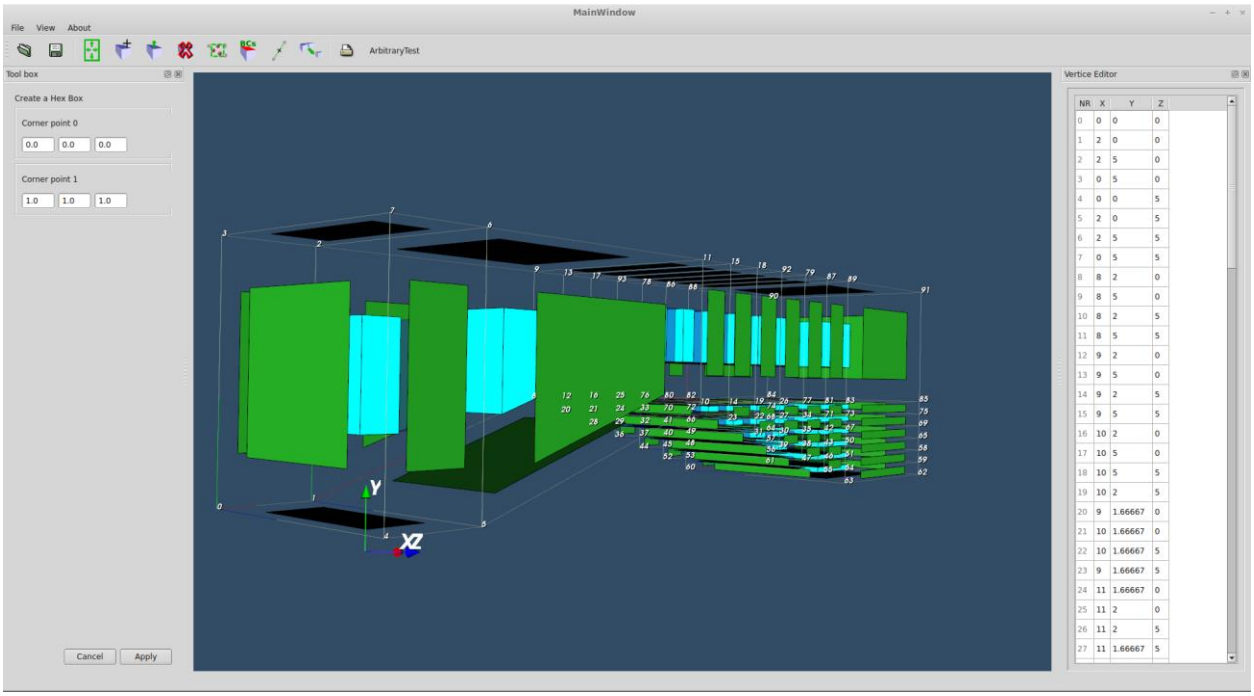

Figure 25. Stepped dam mesh created in hexBlocker

Similar to the previous dams, the blockMesh utility was run on the mesh and the dam was ready for use, as shown in Figure 26.

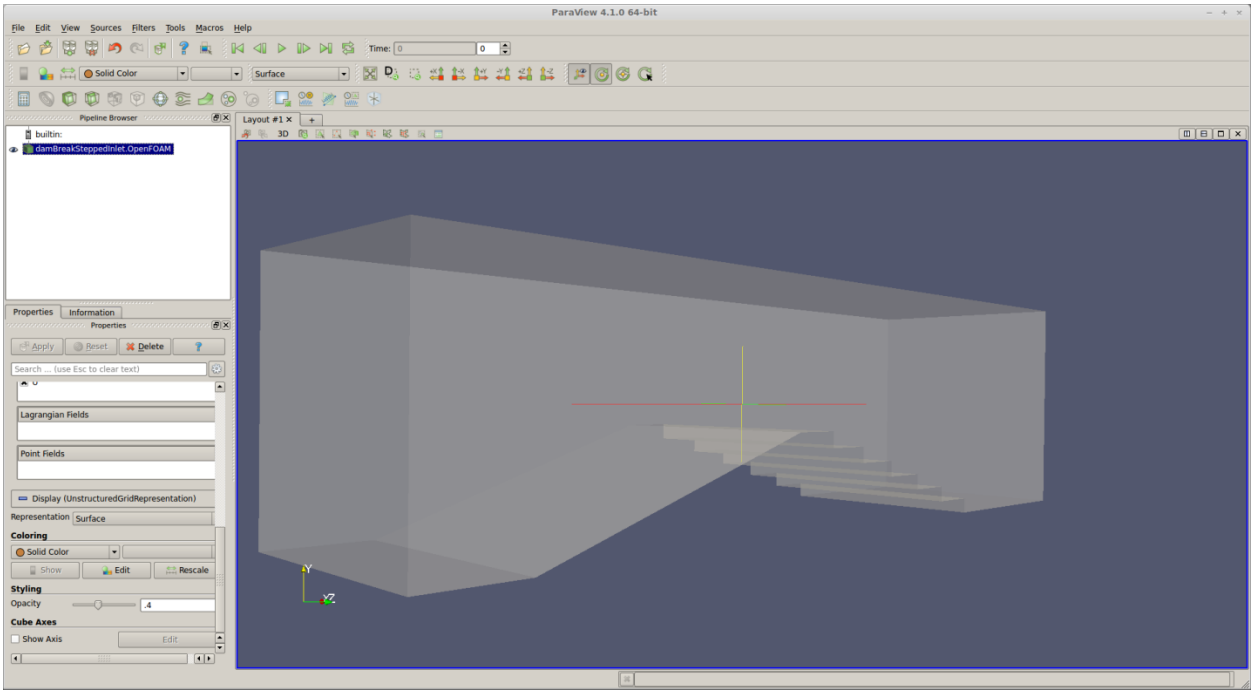

Figure 26. Stepped dam in ParaView

3.6 Stepped Dam with Auxiliary Spillway

For the next dam, the trapezoidal channel shape from the dam in Chapter 3.2 was added to the dam from Chapter 3.5. The dam was created with the stairs as separate sections in hexBlocker to minimize the number of blocks. This mesh required forty-five blocks, as can be seen in Figure 27.

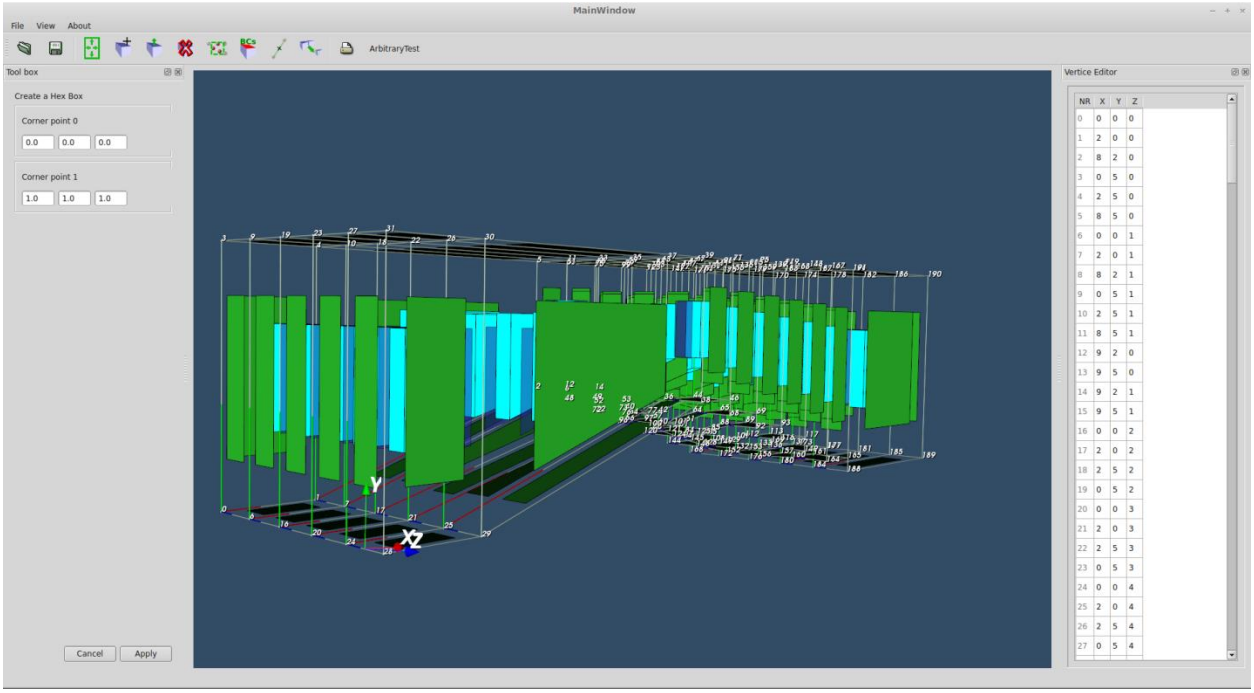

Figure 27. Stepped dam with auxiliary spillway mesh in hexBlocker

Because the stairs were created in sections, the blockMeshDict had to be edited to create a mergePatchPair for all stairs to join the separate sections together. The blockMesh utility was run and the mesh was then ready, as shown in Figure 28.

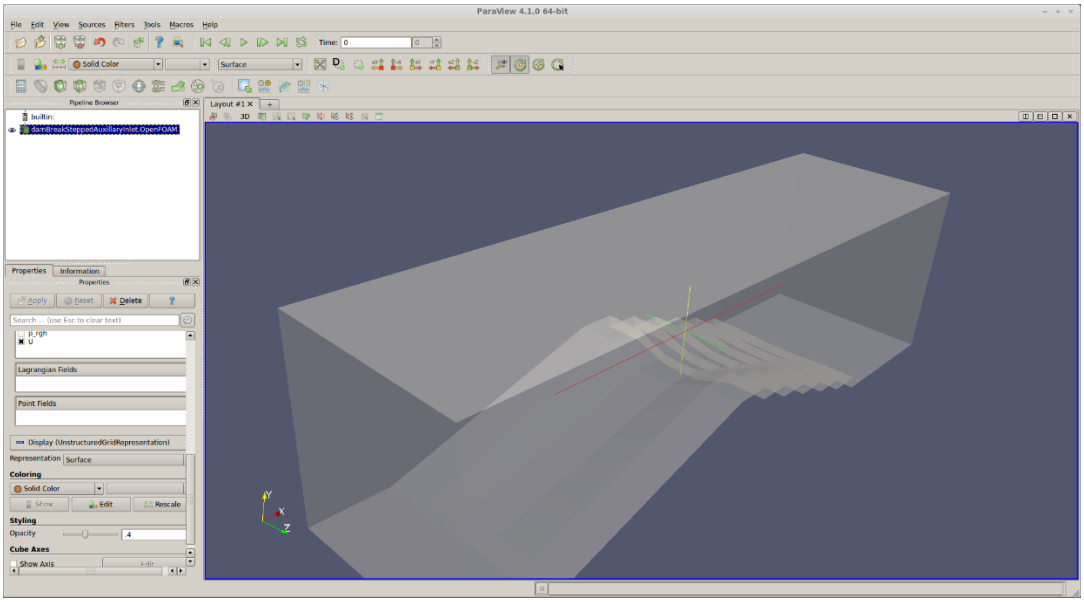

Figure 28. Stepped dam with auxiliary spillway in ParaView

3.7 Stepped Dam with Converging Wall

For the final dam, a converging wall was added to the dam in Chapter 3.5 at an angle of 52 degrees, identical to the dam in Chapter 3.4. This dam had a span of five meters at the top of the crest and a span of 1.09357 meters at the toe. The mesh was created in hexBlocker using nine blocks, as shown in Figure 28.

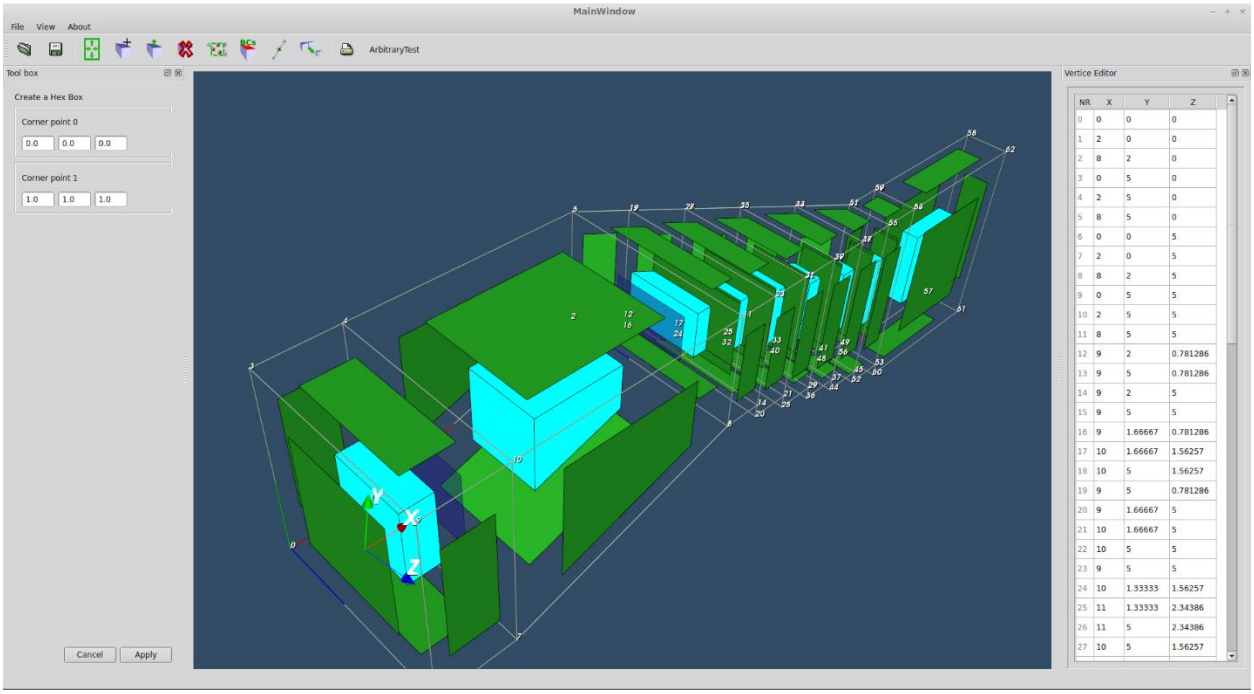

Figure 29. Stepped dam with converging wall mesh created in hexBlocker

After creating a mergePatchPair for each step and running blockMesh, the mesh was ready for use, as illustrated by Figure 30.

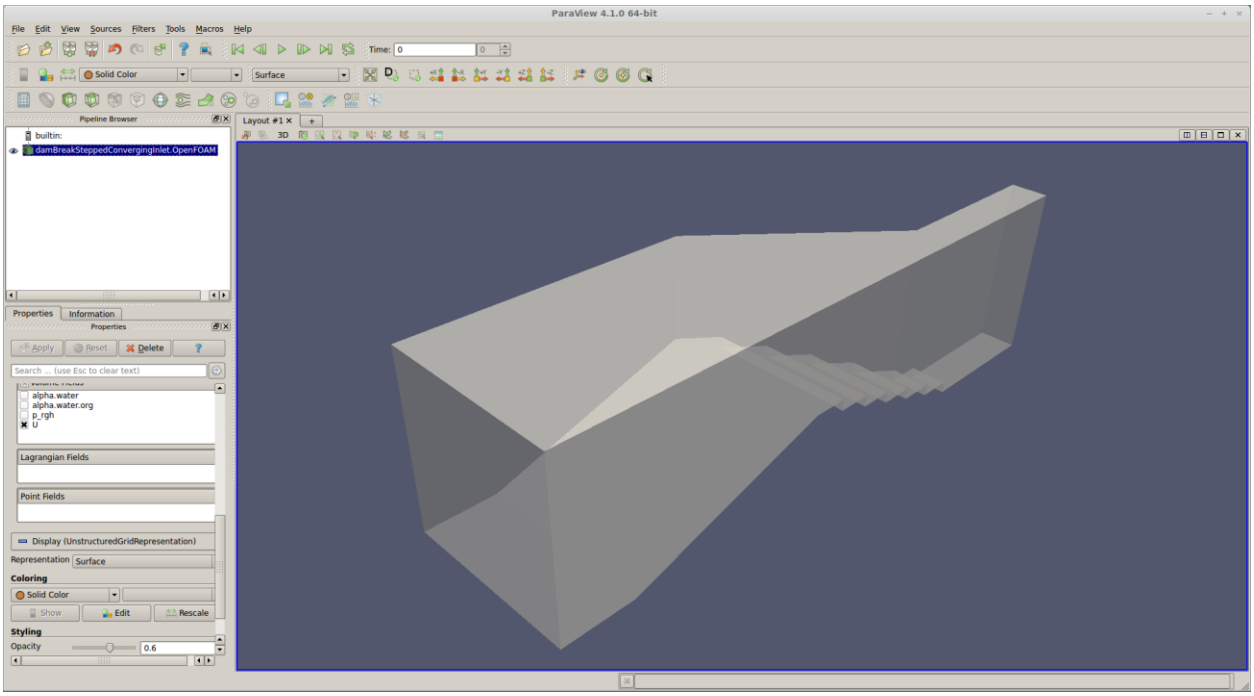

Figure 30. Stepped dam with converging wall viewed in ParaView

Chapter 4 - Results

All dams were processed on a 3.4 GHz, six core, hyper-threaded processor using MPI. OpenFOAM allows users to multithread the processing using the "Decompose Par" utility (DecomposePar, 2009). Each model was broken into six sub-domains so that six threads could be utilized. DecomposePar then required specification of the number of sub-domains in each direction, with the requirement that the numbers in each direction multiply to six. A decomposition of x=3, y=2, and z=1 was chosen because the x axis had the longest length. InterFoam was then run on each dam to produce thirty seconds of data, producing the following results. The decision to use six threads is explained in Chapter 4.8.

4.1 Smooth Dam

The smooth dam took 2127 seconds (35 minutes) to create thirty seconds of data with the interFoam solver. The water line was placed 0.2 meters below the top of the dam at 1.8 meters, as shown in Figure 31.

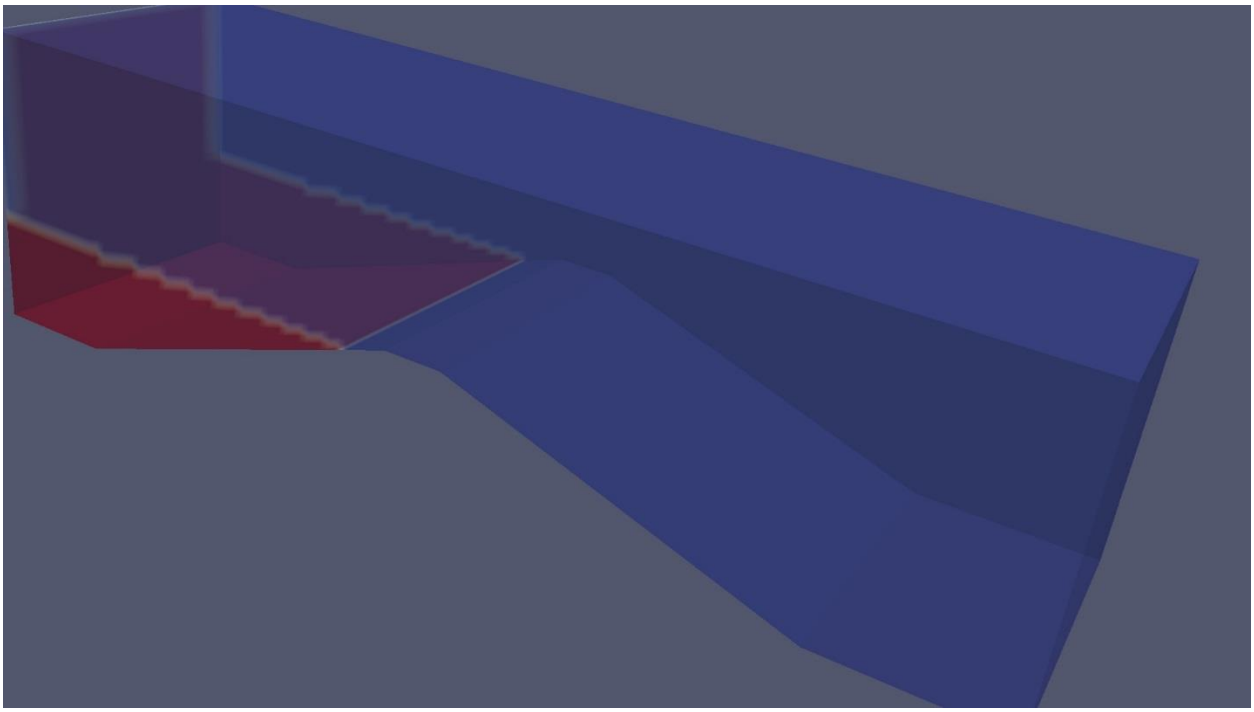

Figure 31. Initial conditions for the smooth dam

After 3.5 seconds, the first wave temporarily pushed water over the crest of the dam and water began spilling over to the downstream side, as shown in Figure 32.

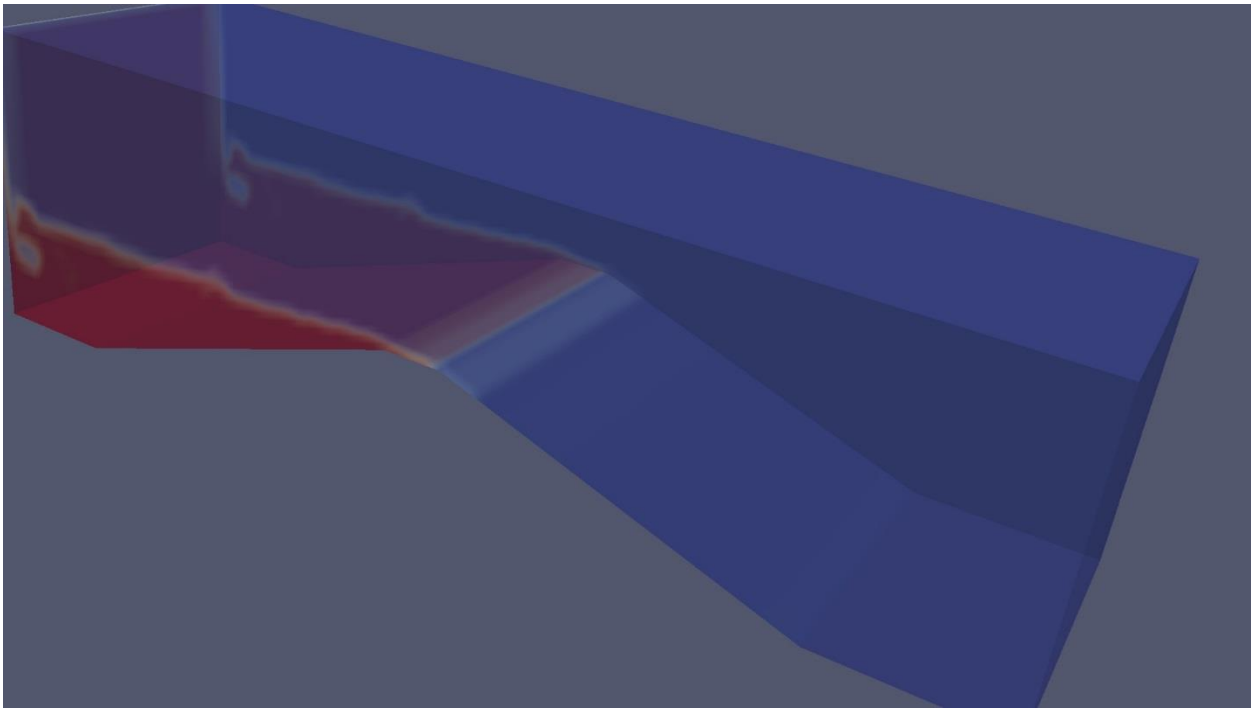

Figure 32. Smooth dam after 3.5 seconds

After approximately 7 seconds, the water level was above the crest of the dam and continuously poured over the crest of the dam, as shown in Figure 33.

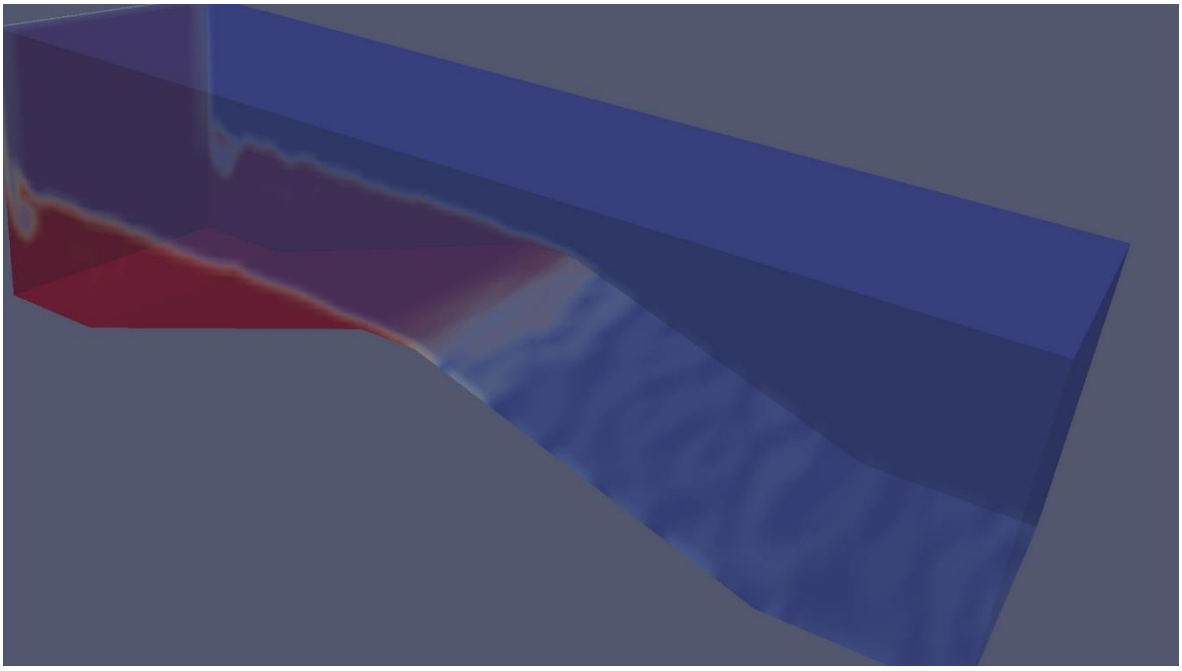

Figure 33. Smooth dam after 7 seconds

The water continued in this fashion for the remainder of the thirty seconds. The water moved quite fast and unhindered as it flowed down the downstream side of the dam. Gaps in the water show that the water is drained faster than it was being sent over the dam, as shown in Figure 34.

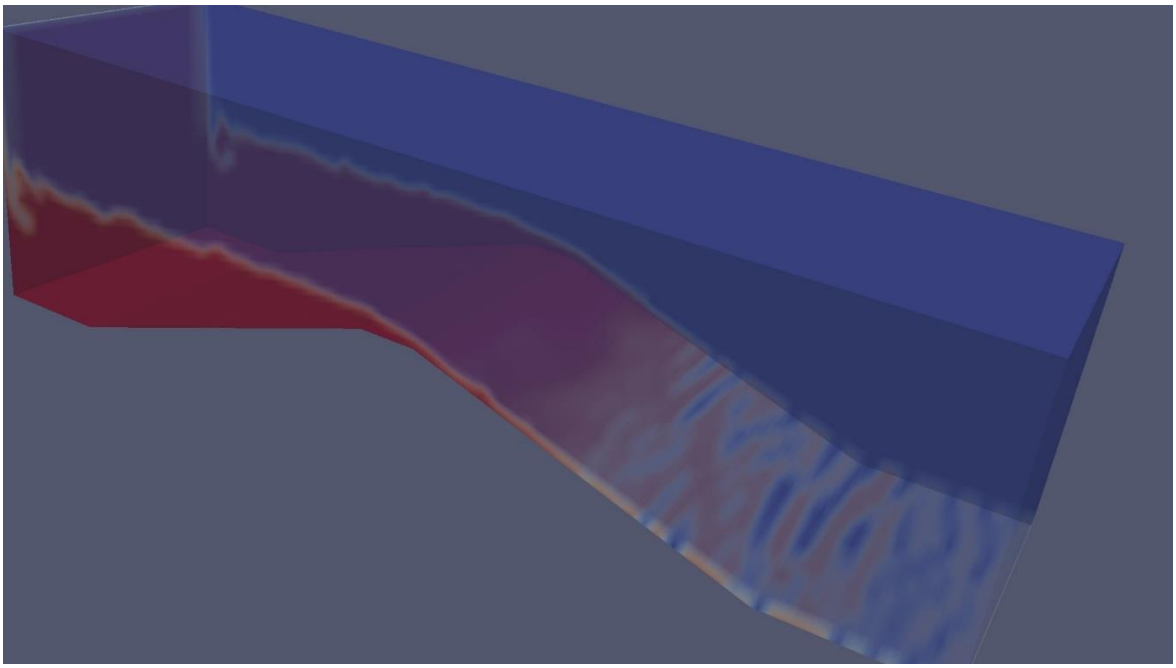

Figure 34. Smooth dam after 30 seconds

Velocity of the water, as illustrated in Figure 35, displays the velocity of the water at four points along the ramp during the last twenty seconds. The points “max U9.5,” “max U11.5,” “max U13.5,” and “max U15.5” show the maximum velocity at .5 meters, 2.5 meters, 4.5 meters, and 6.5 meters from the crest of the dam, respectively.

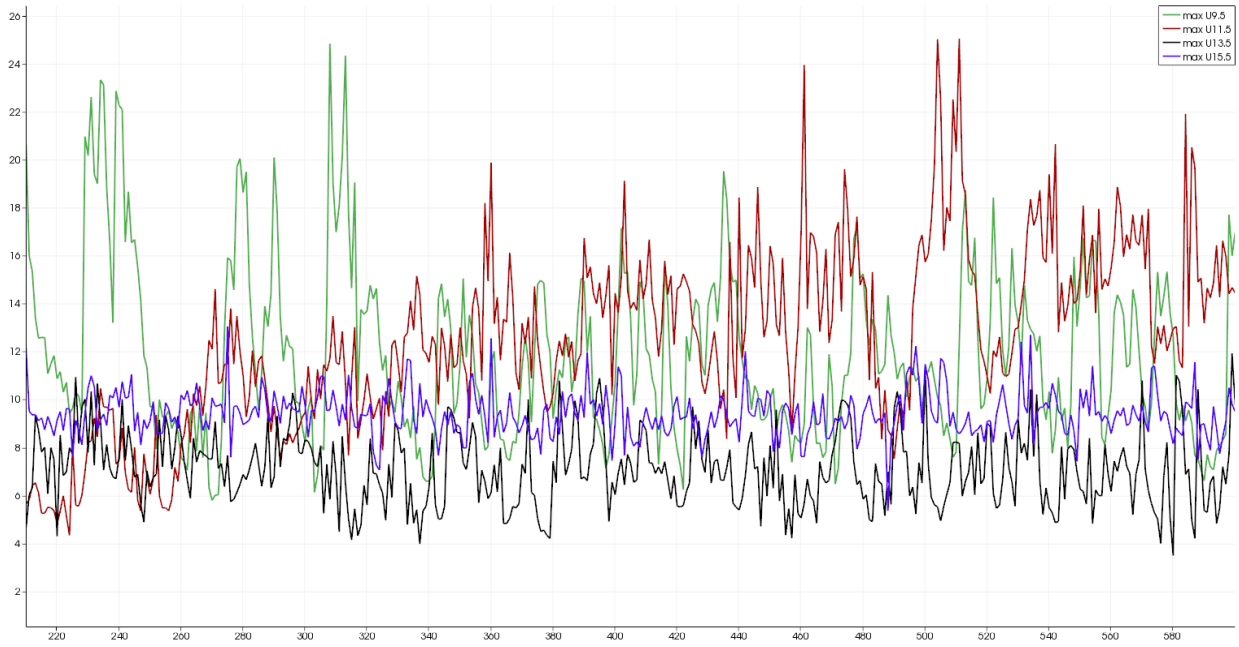

Figure 35. Velocity of the smooth dam over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s

The highest velocity tended to vacillate between the top two points of the ramp, but the speed dropped in half by the time water reached the bottom of the ramp, before speeding up after leveling out.

As shown in Figure 36, dynamic pressure at the top of the dam averaged almost twenty times more than at the bottom of the ramp. Dynamic pressure appears to remain fairly constant throughout the last twenty seconds of the simulation.

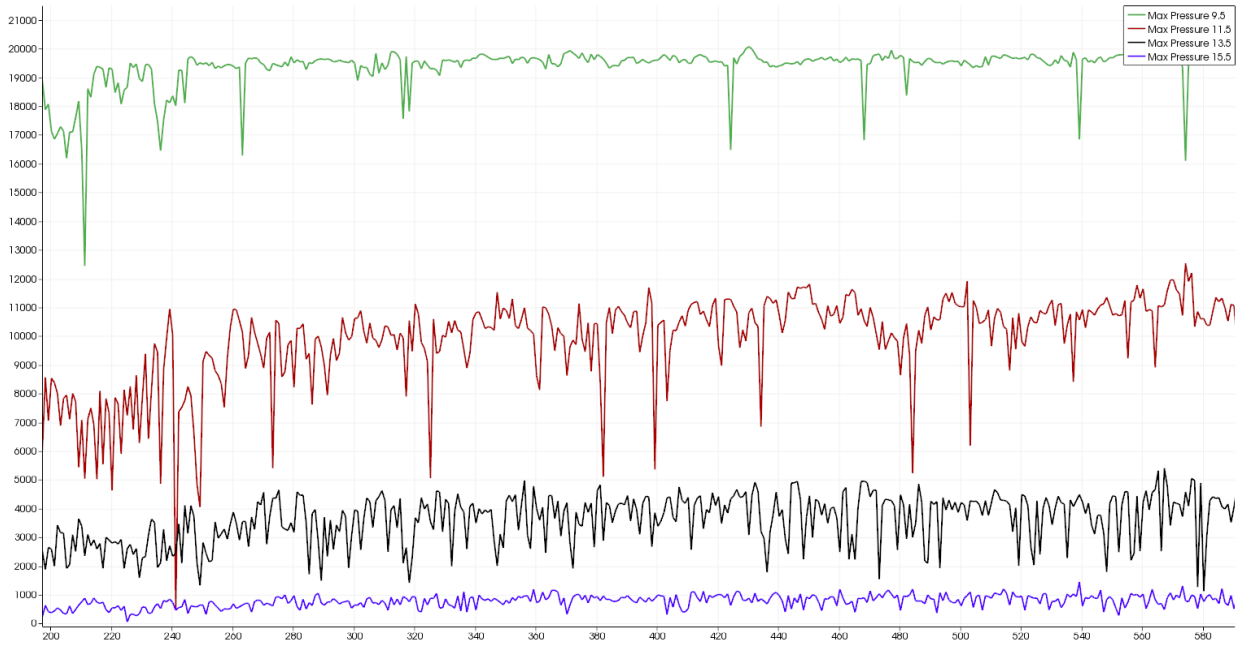

Figure 36. Dynamic pressure of the smooth dam over time, where the x-axis represents the frame number and the y-axis represents maximum dynamic pressure in pascals at each of the four slices

4.2 Smooth Dam with Auxiliary Spillway

The smooth dam with auxiliary spillway took 9449 seconds (2 hours and 37 minutes) to create thirty seconds of data with the interFoam solver. The water line was placed .2 meters below the lowest portion of the dam at 1.4 meters, as shown in Figure 37.

Figure 37. Initial conditions for the smooth dam with auxiliary spillway

Similar to the dam in Chapter 4.1, a wave pushed some water over the crest of the dam 3.5 seconds into the simulation, as shown in Figure 38.

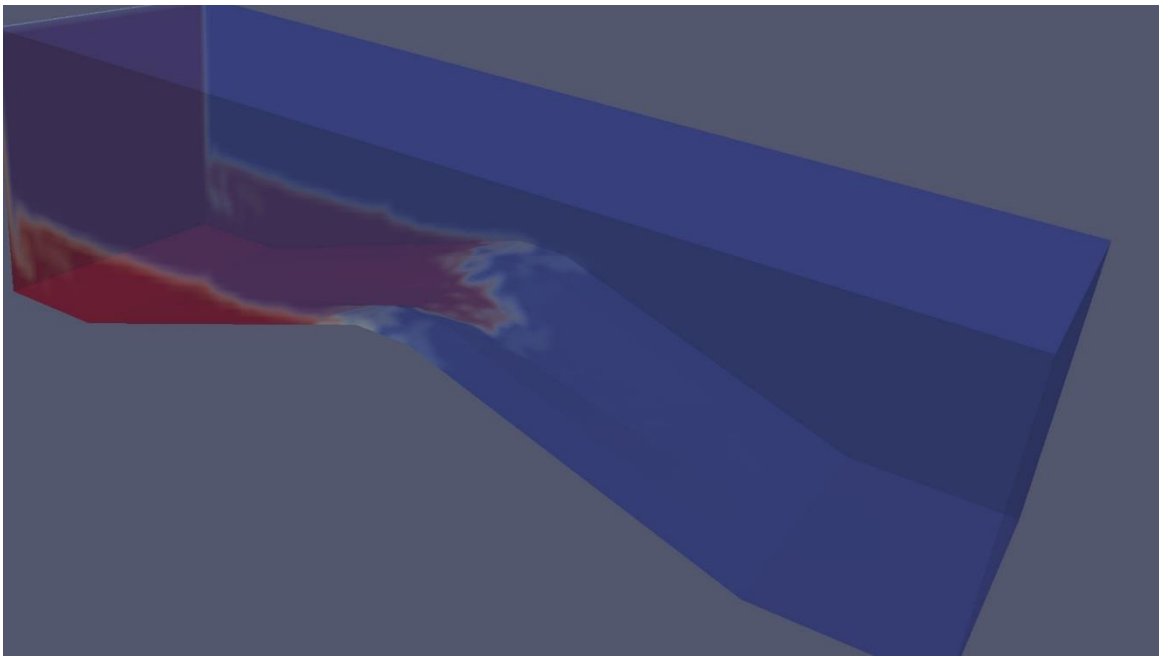

Figure 38. Smooth dam with auxiliary spillway after 3.5 seconds

After approximately 7.5 seconds, water began overflowing the trapezoidal channel, which is the desired effect, because this auxiliary spillway was designed to have water flow over it before it flows over the crest of the rest of the dam, as shown in Figure 39.

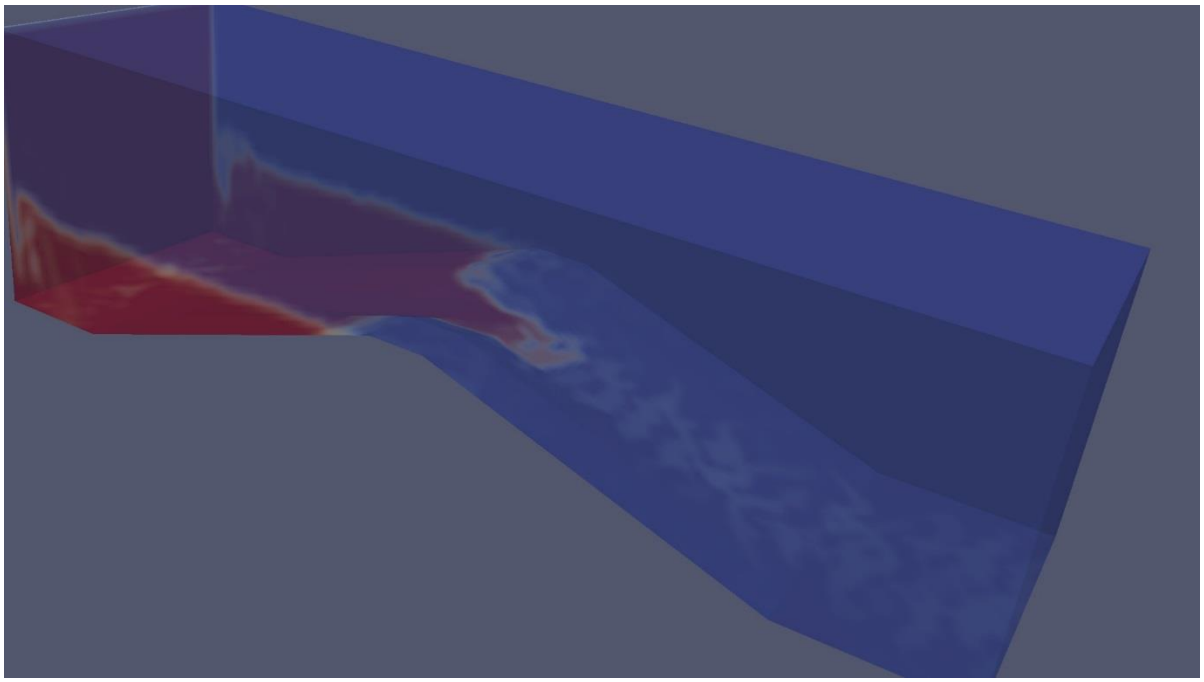

Figure 39. Smooth dam with auxiliary spillway after 7.5 seconds

For the remainder of the thirty seconds, water continued to flow down the trapezoidal channel, with waves periodically pushing water over the outer parts of the dam, as shown in Figure 40.

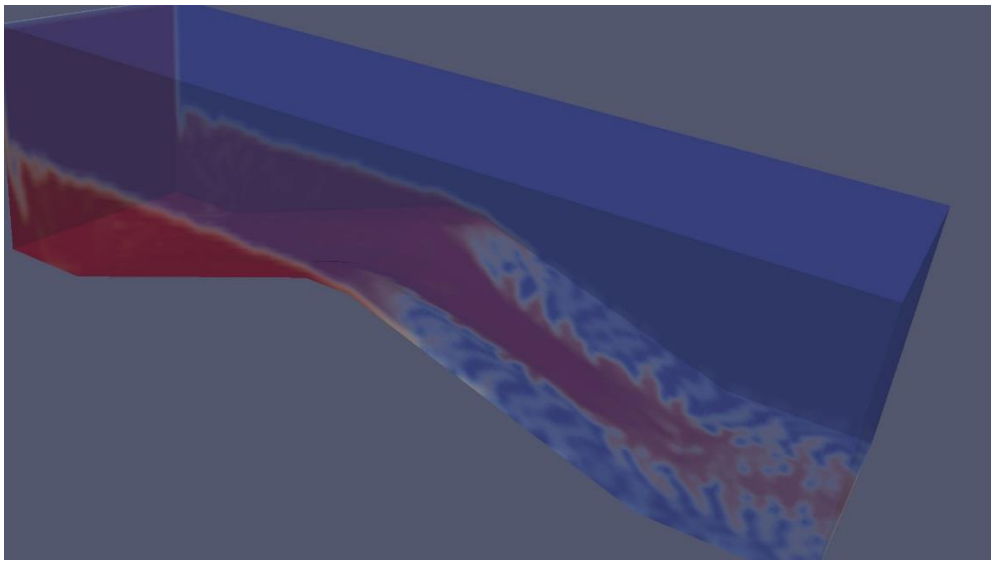

Figure 40. Smooth dam with auxiliary spillway after 30 seconds

As a contrast to the smooth dam in Chapter 4.1, the water at the top of the dam tended to be the slowest, with the fastest flows near the bottom of the dam, as illustrated in Figure 41. The water slowed after it was off the ramp because it was no longer trapped in the three meter channel and it was able to fan out.

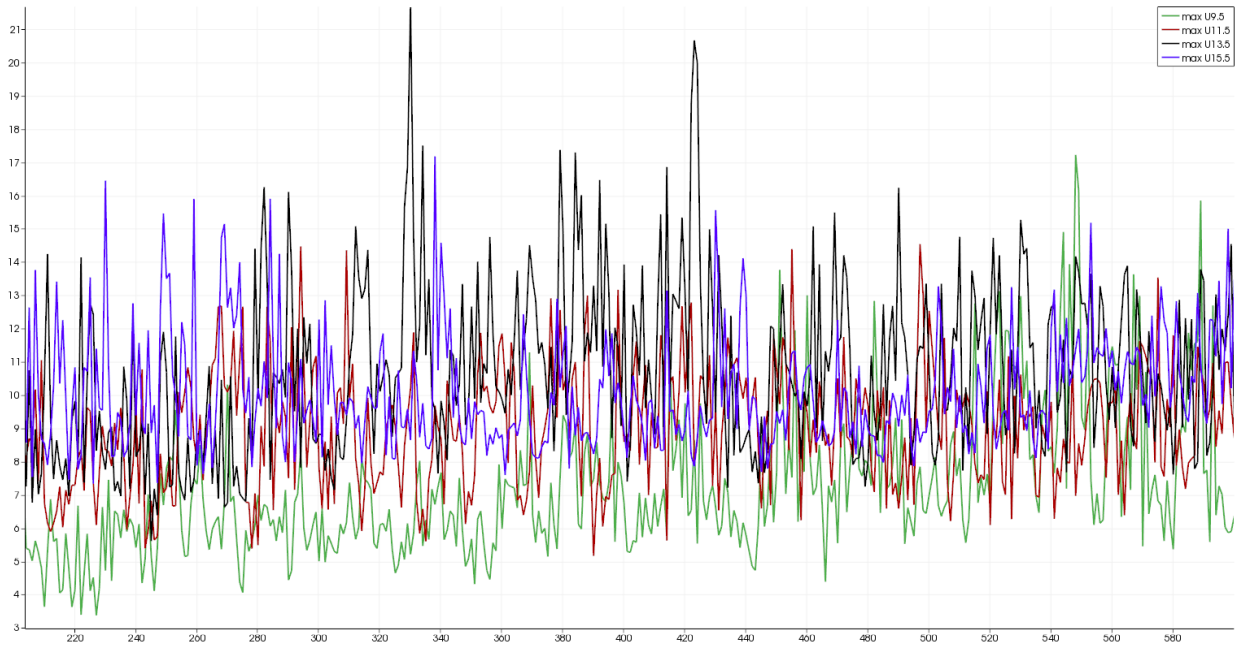

Figure 41. Velocity of the smooth dam with auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s

Similar to the smooth dam, dynamic pressure at the top of the dam was higher than at the bottom. Unlike the smooth dam, dynamic pressure was rose slowly during the last twenty seconds of the simulation, as illustrated in Figure 42.

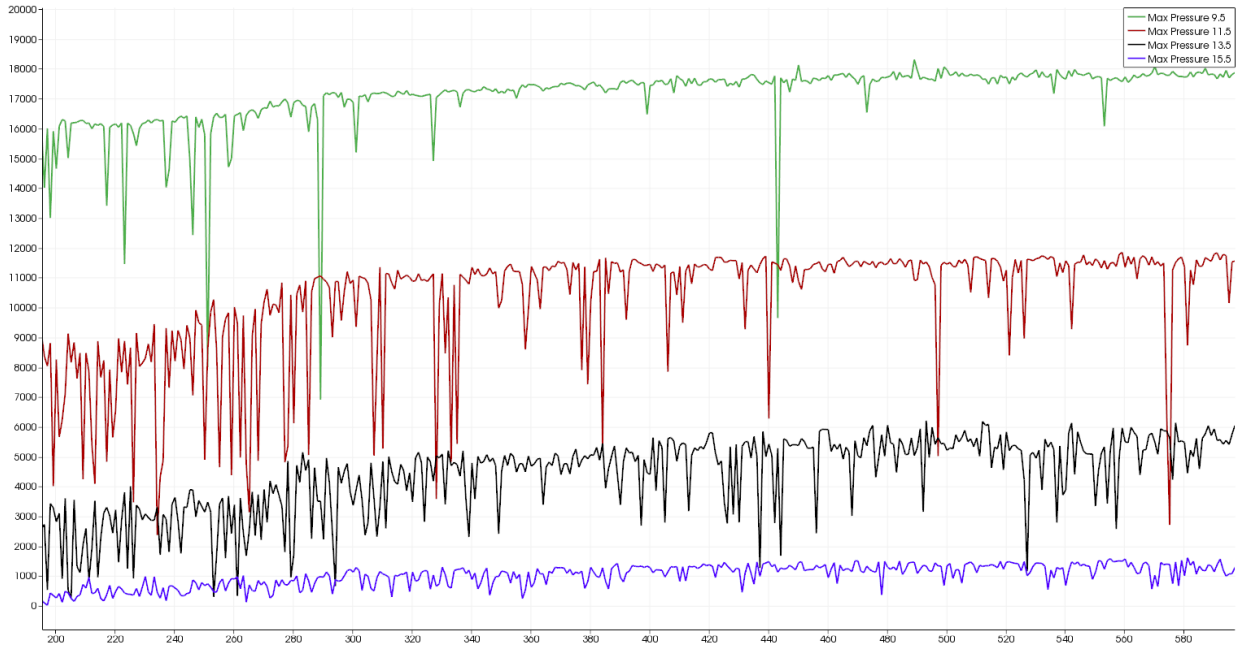

Figure 42. Dynamic pressure of the smooth dam with auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents the maximum dynamic pressure in pascals at each of the four slices

4.3 Smooth dam with Principal and Auxiliary Spillway

The smooth dam with principal and auxiliary spillway took 9609 seconds (2 hours and 40 minutes) to create thirty seconds of data with the interFoam solver. The water line was placed .2 meters below the lowest portion of the dam at 1.4 meters, well above the principal spillway, as shown in Figure 43.

Figure 43. Initial conditions for the smooth dam with principal and auxiliary spillway

A few frames later, water can be seen coming out of the principal spillway using the contour view, as shown in Figure 44.

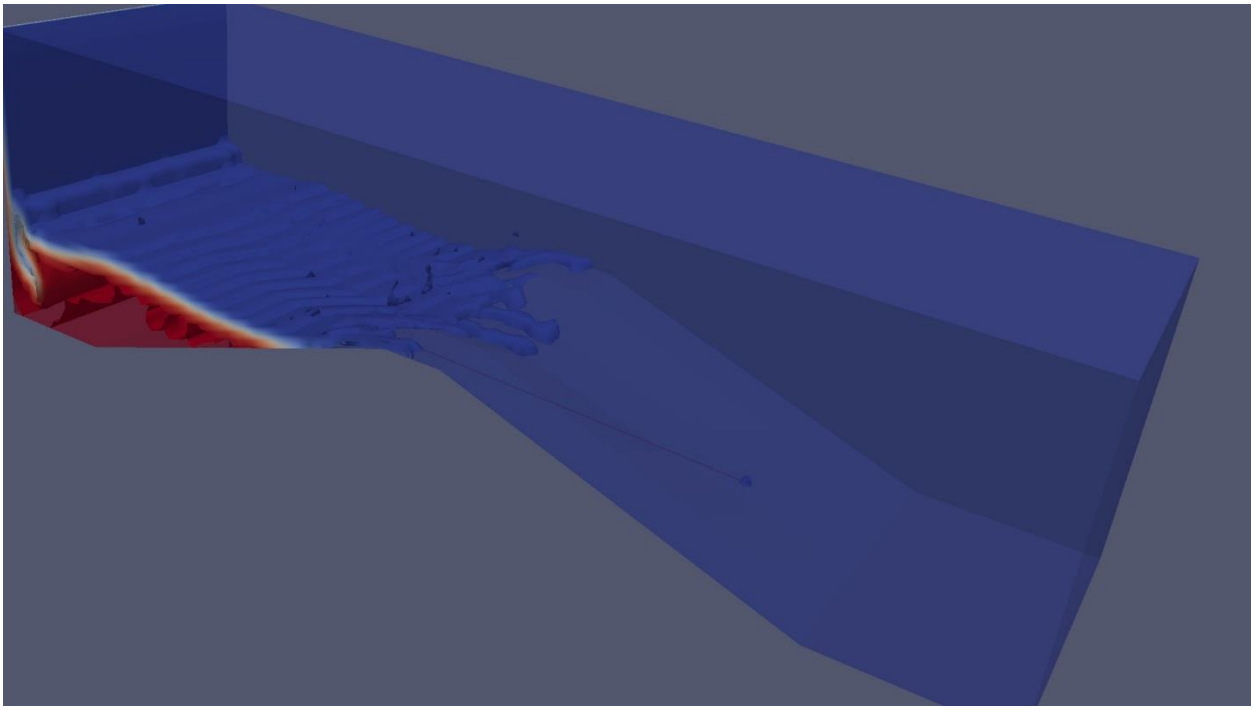

Figure 44. Contour view of the smooth dam with principal and auxiliary spillway showing water flowing through the pipe

As with the previous dam, waves temporarily pushed some water over the top of the crest of the auxiliary spillway 3.5 seconds later, as shown in Figure 45.

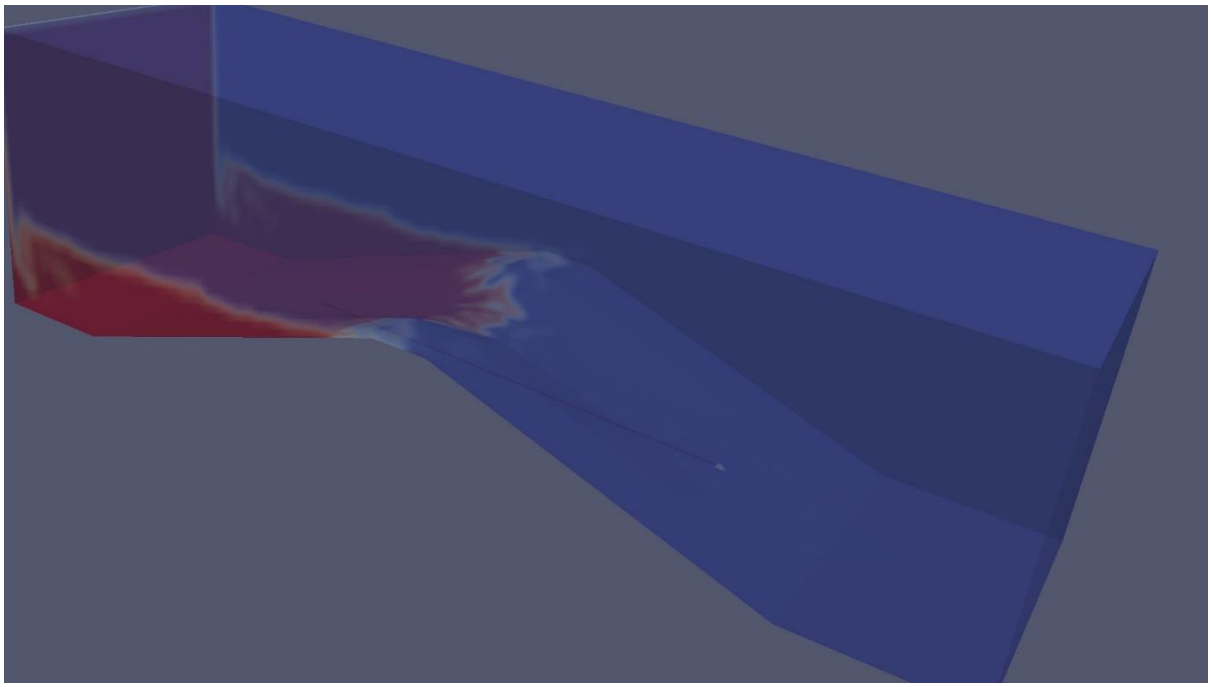

Figure 45. Smooth dam with principal and auxiliary spillway after 3.5 seconds

In the same fashion as the last dam, water began overflowing the trapezoidal channel 7.5 seconds into the simulation, as seen in Figure 46.

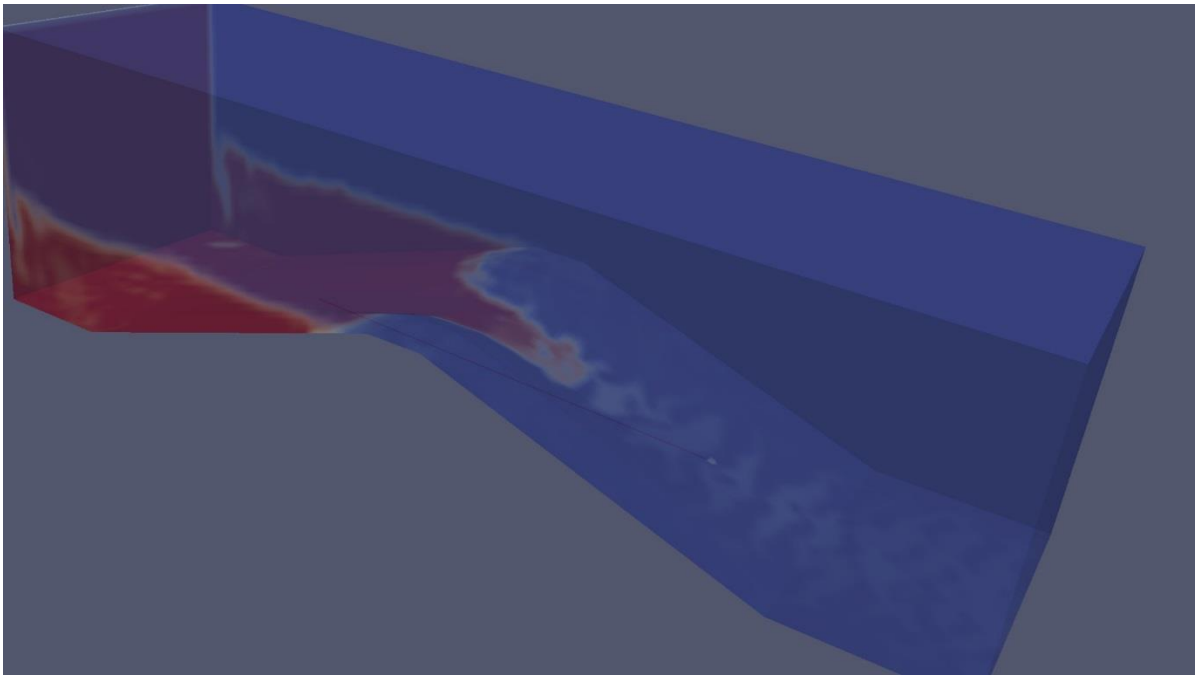

Figure 46. Smooth dam with principal and auxiliary spillway after 7.5 seconds

For the rest of the thirty seconds, waves periodically pushed water over the sides of the dam, while a steady stream of water flowed down the auxiliary channel, as shown in Figure 47.

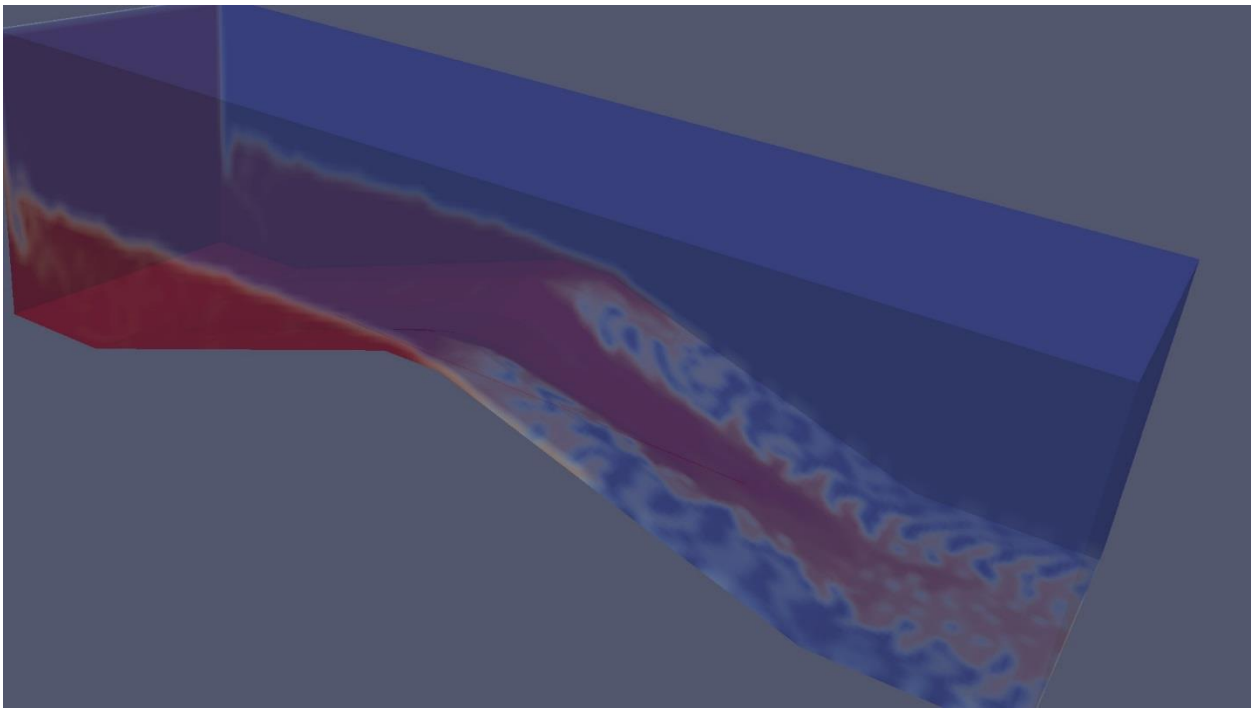

Figure 47. Smooth dam with principal and auxiliary spillway after 30 seconds

The velocity graph in Figure 48 produces similar results to the dam in Chapter 4.2 as the dams were identical except for the tiny spillway going through the dam. This result was expected because the principal spillway was designed to mitigate water before it gets to the point at which the simulation began. The small pipe was unable to expel the volume of water required to be helpful.

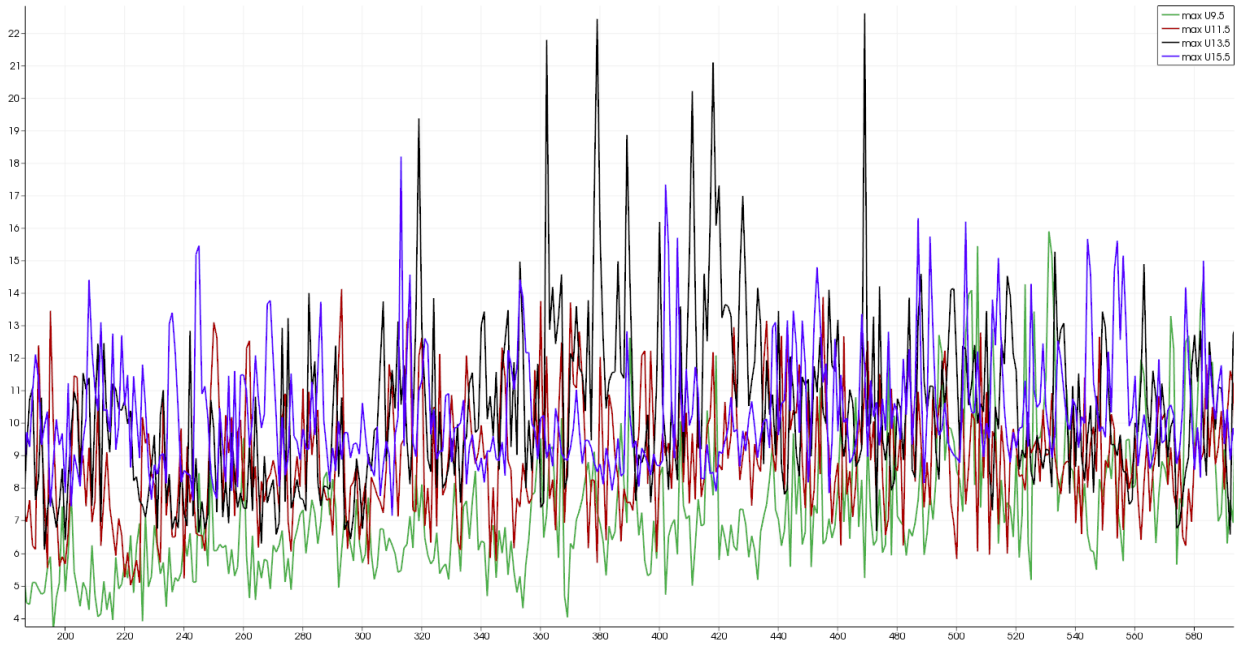

Figure 48. Velocity of the smooth dam with principal and auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s

Since this dam is almost identical to the smooth dam with auxiliary spillway, the pressure graphs would expected to be similar, as was the case, as shown by Figure 49.

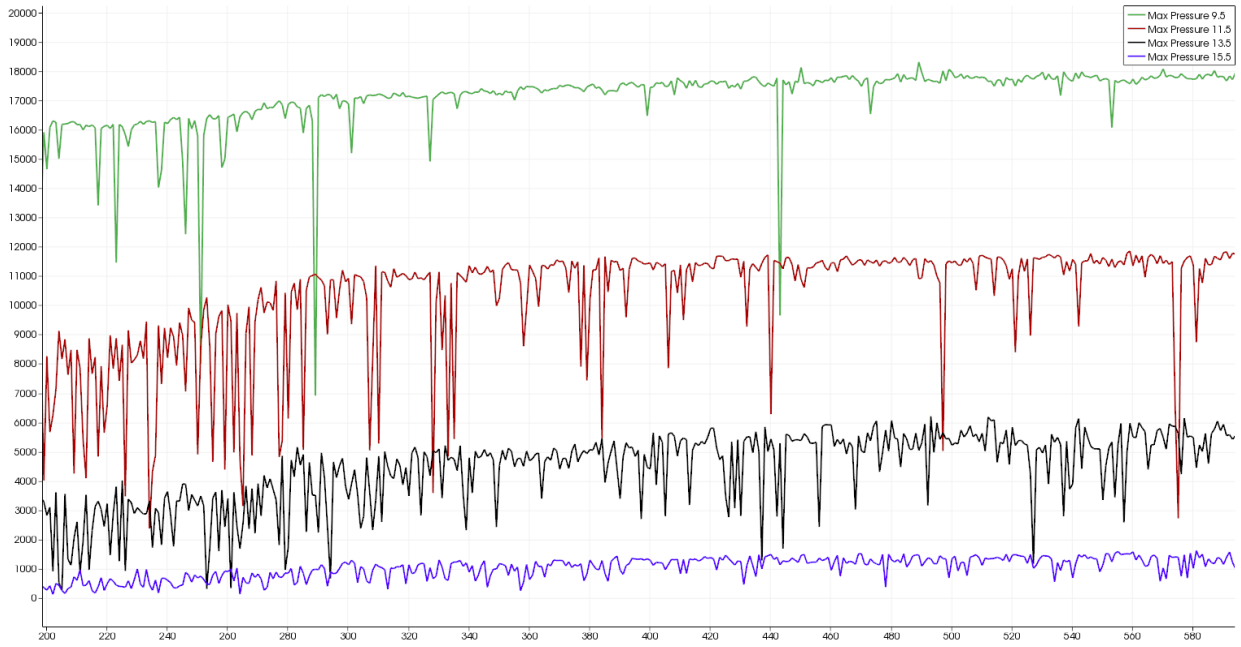

Figure 49. Dynamic pressure of the smooth dam with principal and auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents maximum dynamic pressure in pascals at each of the four slices.

4.4 Smooth Dam with Converging Wall

The stepped dam took 7191 seconds (2 hours) to create thirty seconds of data with the interFoam solver. The water line was placed .2 meters below the top of the dam at 1.8 meters, as shown in Figure 50.

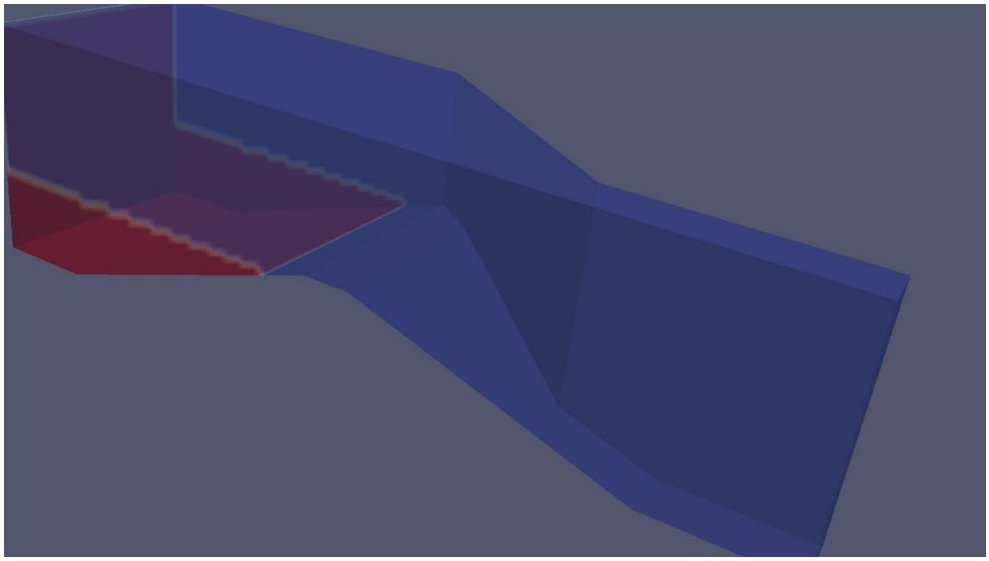

Figure 50. Initial conditions for the smooth dam with converging wall

Since the upstream portion of this dam's mesh was identical to the dam in Chapter 4.1, the first wave pushed some water over at 3.5 seconds, as shown in figure 51.

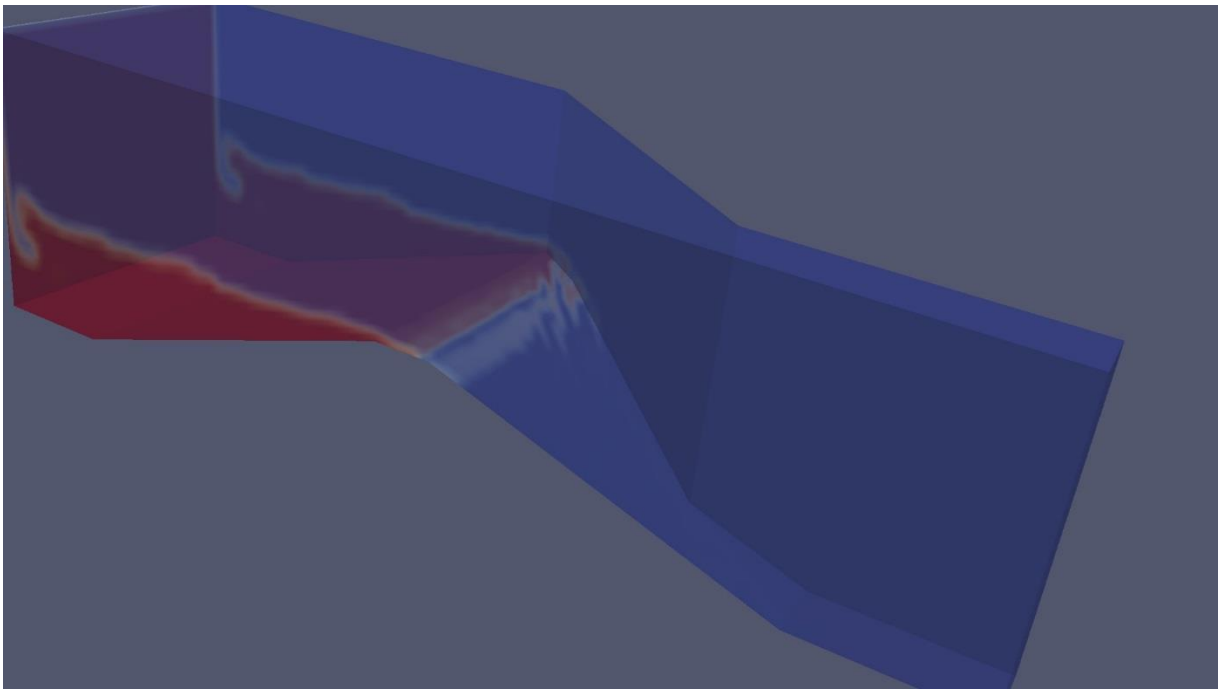

Figure 51. Smooth dam with converging wall after 3.5 seconds

After 8 seconds, the water line was above the crest of the dam and it flowed freely over the crest, as shown in Figure 52.

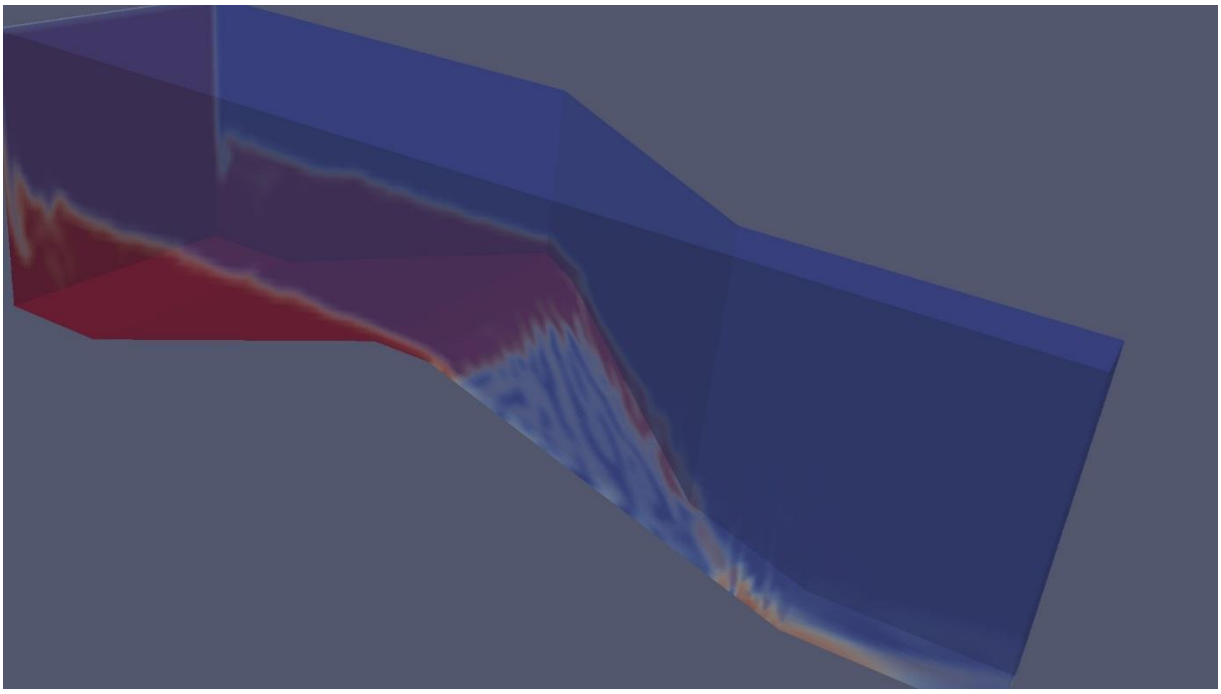

Figure 52. Smooth dam with converging wall after 8 second

Switching to the volume view, the water was observed to consistently run along the converging wall into the smaller opening at the bottom and then smash into the other wall, making a giant splash for the remainder of the thirty seconds.

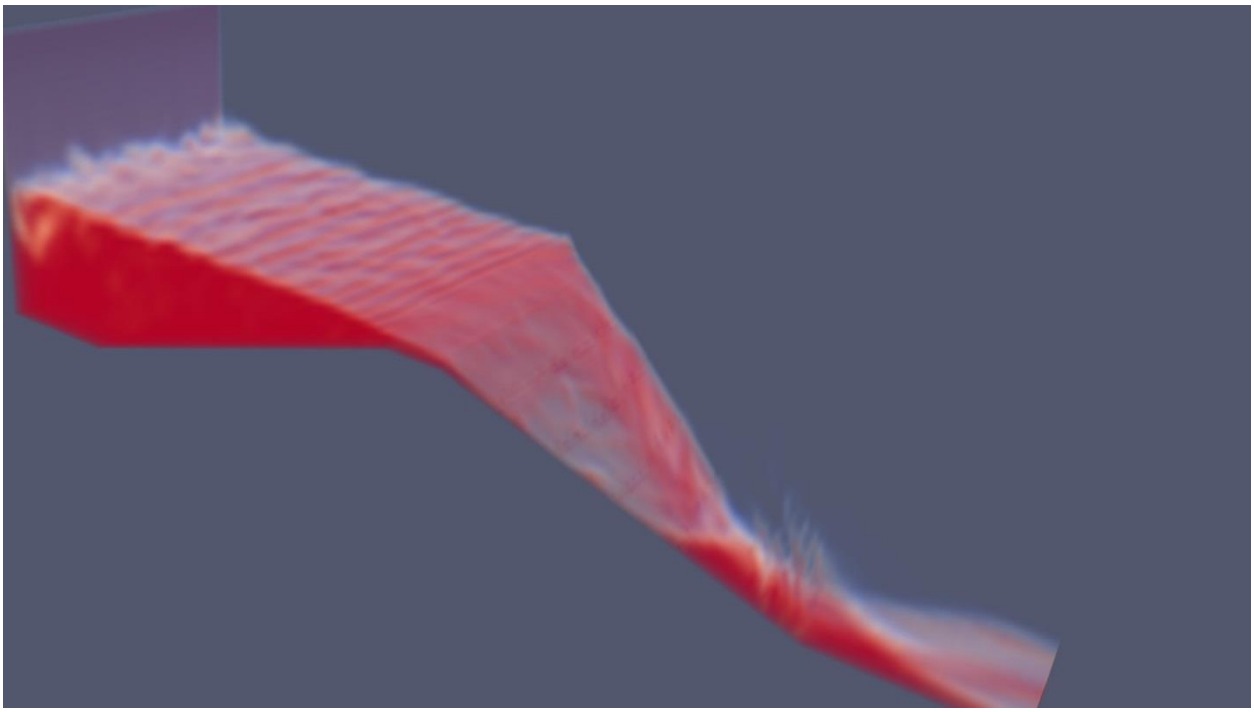

Figure 53. Volume view of the smooth dam with converging wall after 30 seconds

The velocity graph, demonstrated in Figure 54, produced in this example shows more consistent velocities and lower overall velocity than the dam in Chapter 4.1. The water at the top of the dam flowed the slowest. Maximum speed is vacillated between the middle two points on the dam before decreasing to slightly higher than the velocity at the top of the dam. Two factors produced this slow down: The cross stream bouncing off of the wall and the fast-moving water colliding with the slow moving water.

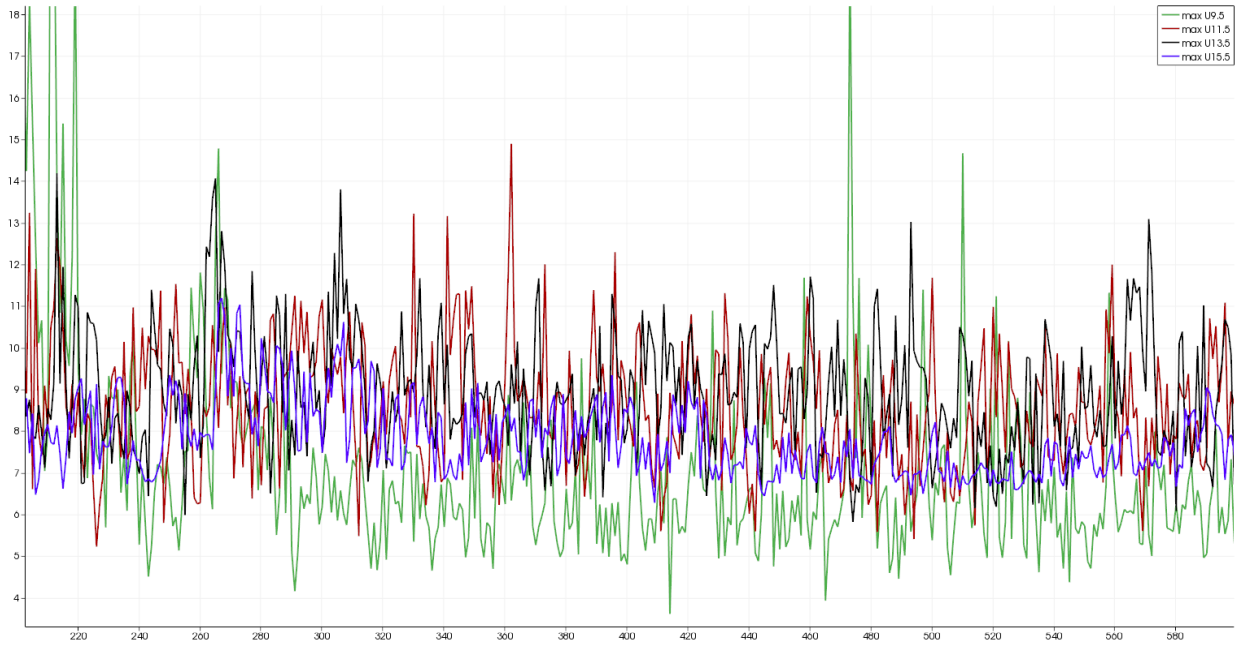

Figure 54. Velocity of the smooth dam with converging wall over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s

As with previous dams, the smooth dam with converging wall had higher dynamic pressure at the top of the dam and lower dynamic pressure at the bottom of the dam, as shown in Figure 55. The pressure at each slice was higher than previous dams, and the pressure increased as time went on.

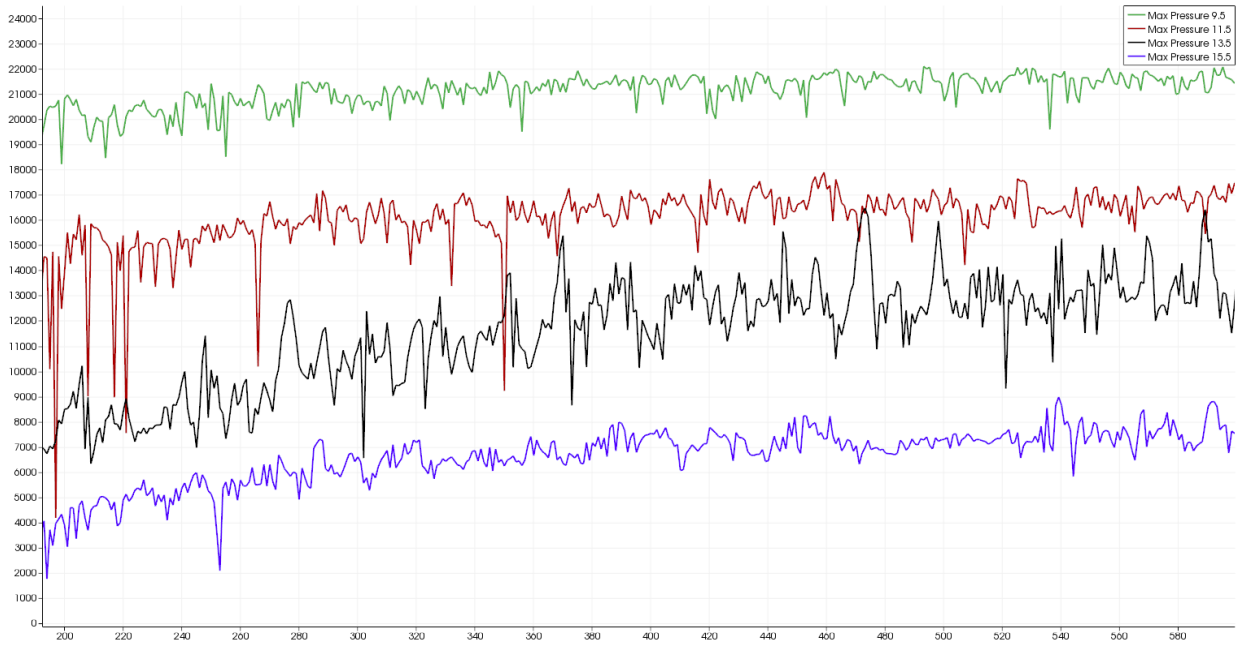

Figure 55. Dynamic pressure of the smooth dam with converging wall over time, where the x-axis represents the frame number and the y-axis represents the max pressure in pascals at each of the four slices

4.5 Stepped Dam

The stepped dam took 2245 seconds (37 minutes) to create thirty seconds of data with the interFoam solver. The water line was placed .2 meters below the top of the dam at 1.8 meters, as shown in Figure 56.

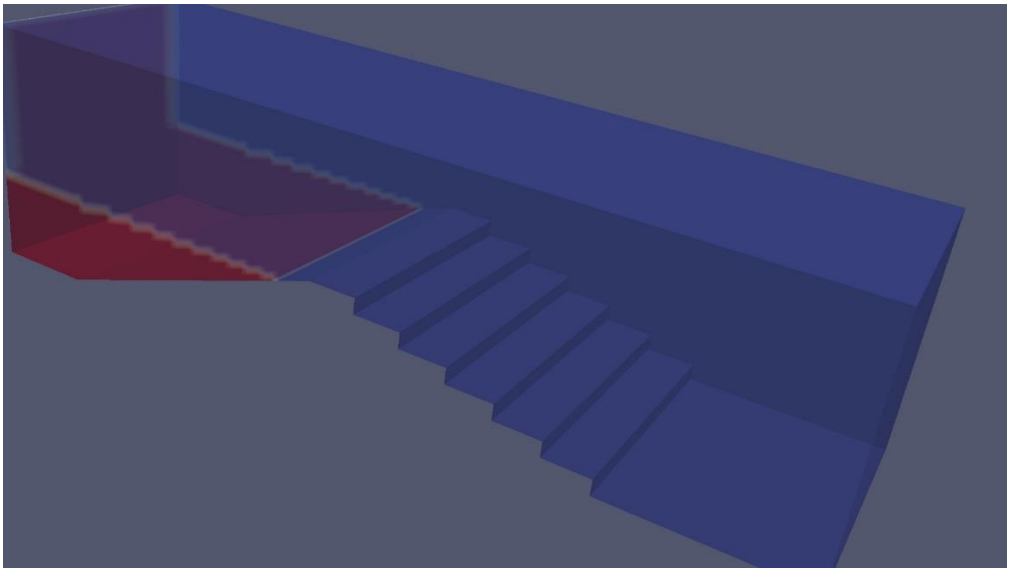

Figure 56. Initial conditions for the stepped dam

The stream side of this mesh was identical to the smooth dam mesh in Chapter 4.1, so the simulation initially behaved identically, with the first wave pushing water over the crest of the dam at 3.5 seconds, as shown in Figure 57.

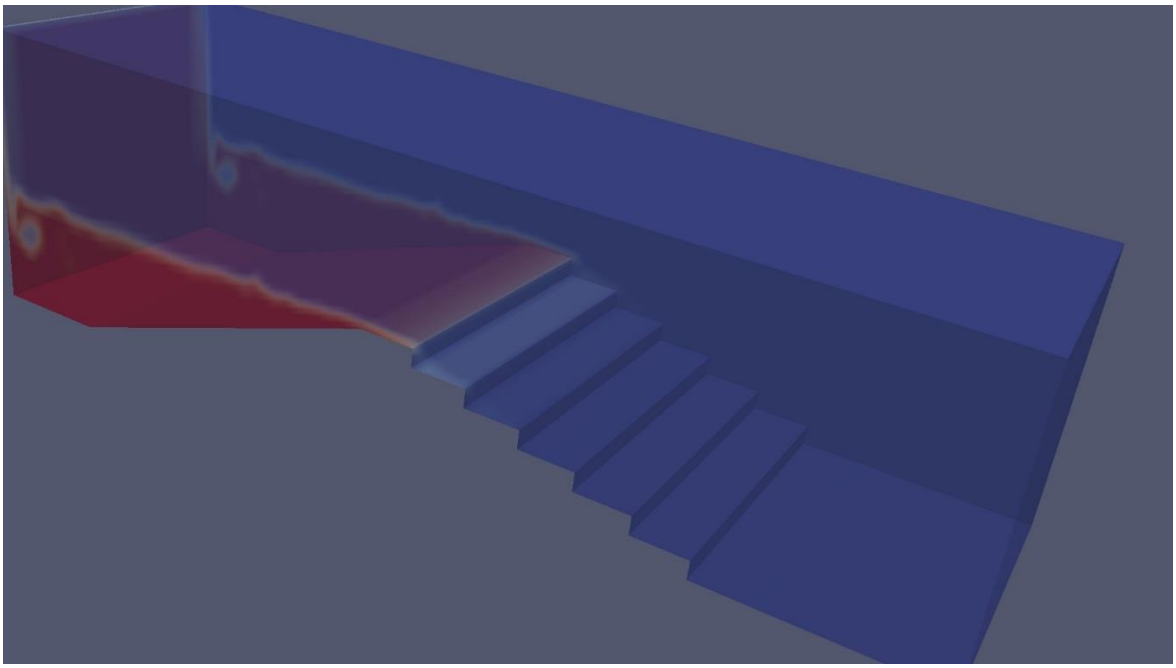

Figure 57. Stepped dam after 3.5 seconds

After 7.5 seconds, the water streamed over the top of the dam and water from the initial wave reached the end of the simulation space, as seen in Figure 58.

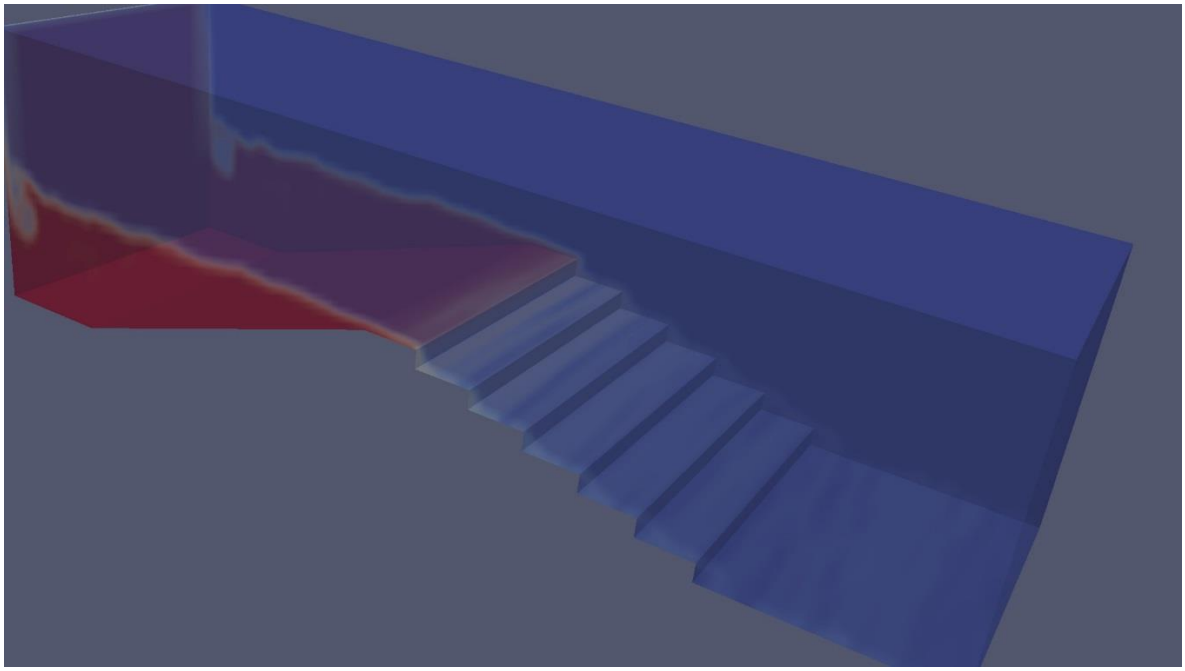

Figure 58. Stepped dam after 7.5 seconds

For the remainder of the simulation, the water appeared to be sitting on the entirety of the stairs, as shown in Figure 59. No gaps were present as in the dam in Chapter 4.1, meaning that the water moved slowly enough to bunch up.

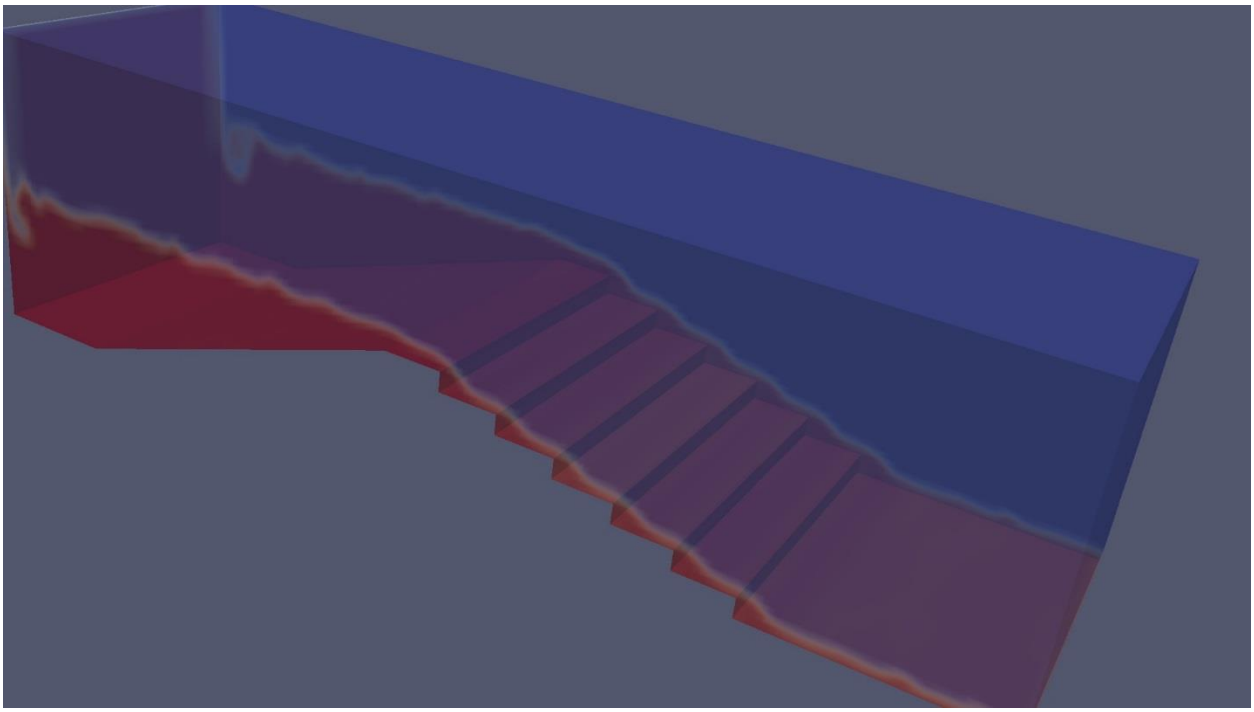

Figure 59. Stepped dam after 30 seconds

The velocity graph in Figure 60 confirms that the water was moving substantially slower than the smooth dam in Chapter 4.1. The maximum speed was 4.4 meters per second versus the 25 meters per second on the smooth dam. Water on this dam increased in speed as it flowed down the stairs, but only at a rate of approximately one meter per second in speed.

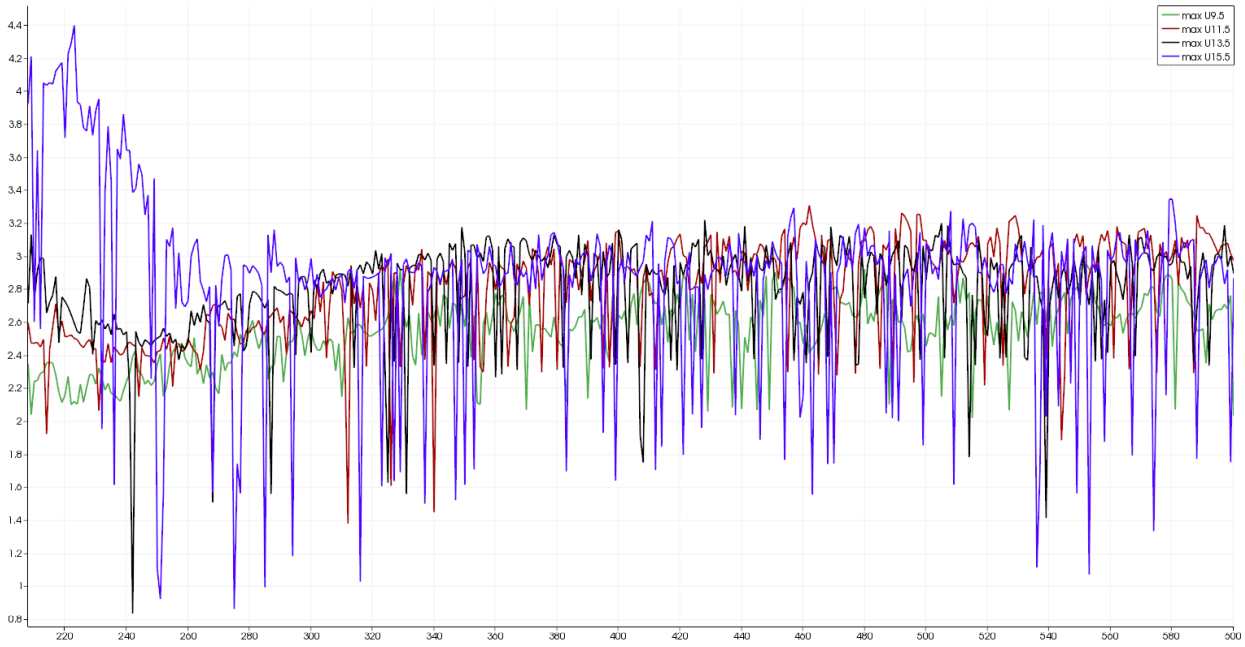

Figure 60. Velocity of the stepped dam over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s

As with the other dams, dynamic pressure at the top was higher than at the bottom. Compared to the smooth dam, the pressure tended to be higher after the water overtopped the dam and slowly increased during the last twenty seconds, as can be seen in Figure 61.

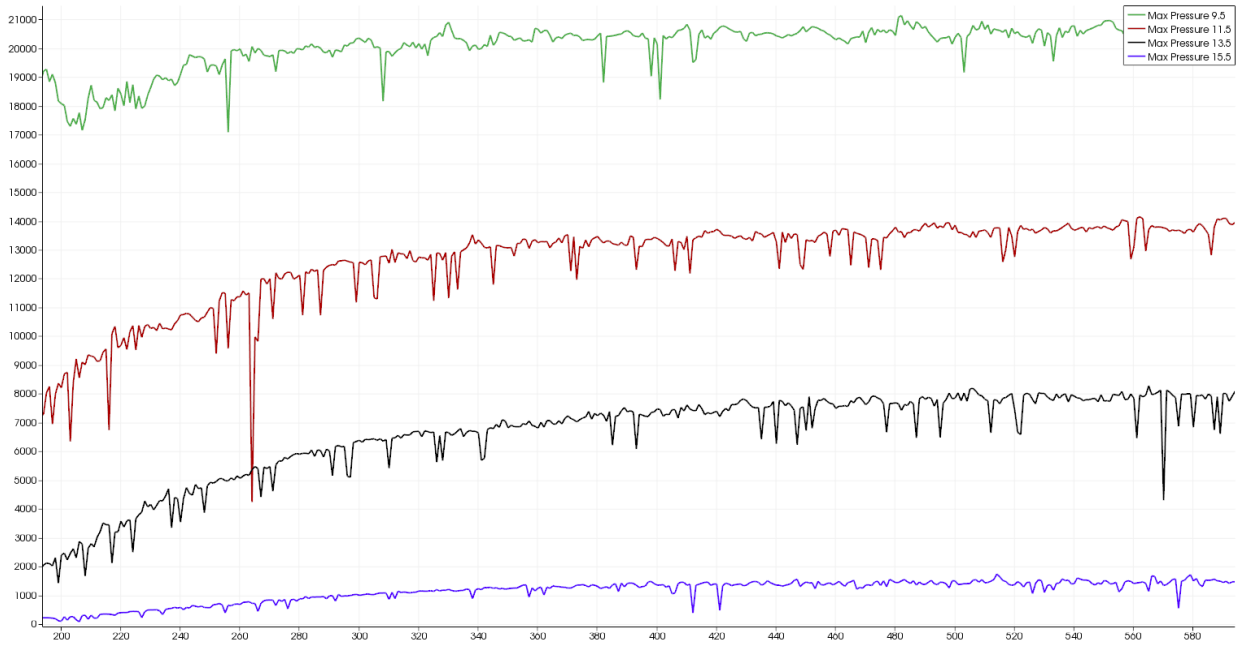

Figure 61. Dynamic pressure of the stepped dam over time, where the x-axis represents the frame number and the y-axis represents maximum dynamic pressure in pascals at each of the four slices

4.6 Stepped Dam with Auxiliary Spillway

The stepped dam with auxiliary spillway took 8776 seconds (2 hours and 26 minutes) to create thirty seconds of data with the interFoam solver. The water line was placed .2 meters below the lowest portion of the dam at 1.4 meters, as shown in Figure 62.

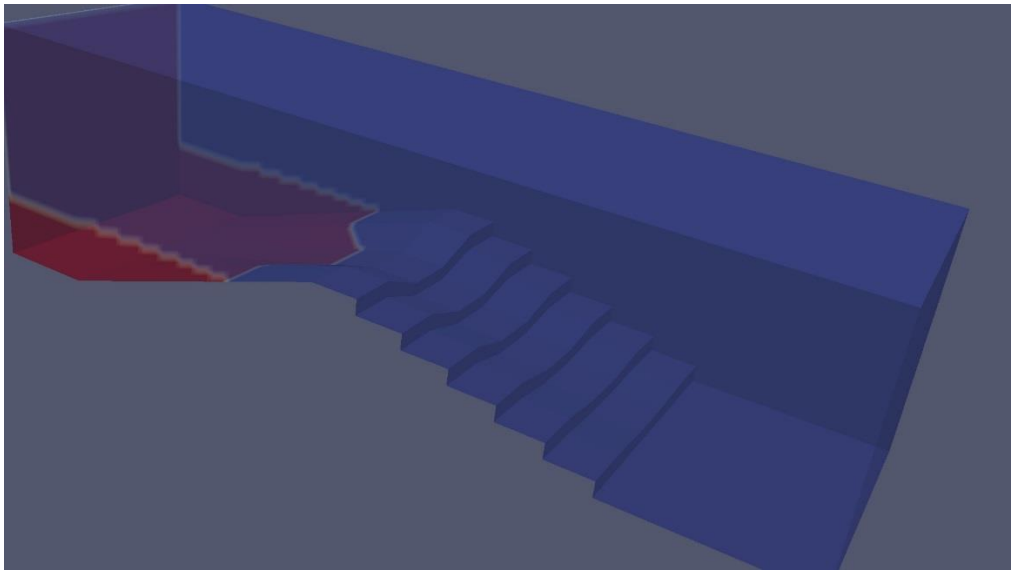

Figure 62. Initial conditions for the stepped dam with auxiliary spillway

The upstream part of this mesh is identical to the basic dam mesh with the auxiliary spillway, so it initially behaved identically, with the first wave pushing water over the crest 3.5 seconds after the start of the simulation, as shown in Figure 63.

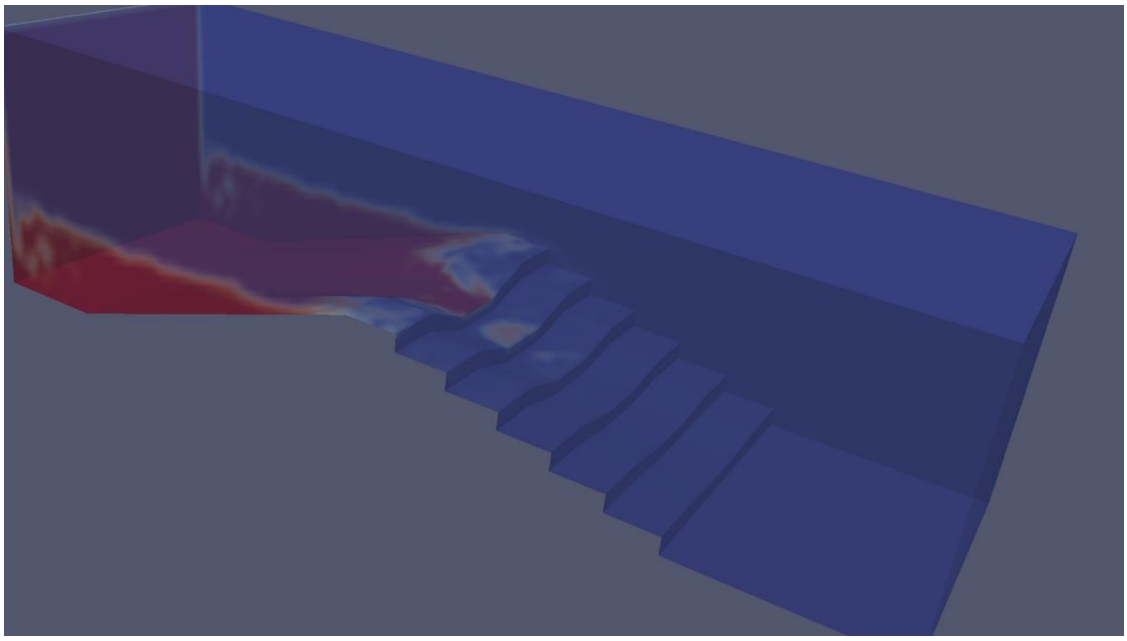

Figure 63. Stepped dam with auxiliary spillway after 3.5 seconds

After 7.5 seconds, the water reached the crest of the auxiliary spillway and flowed over the crest of the dam, as shown in Figure 64.

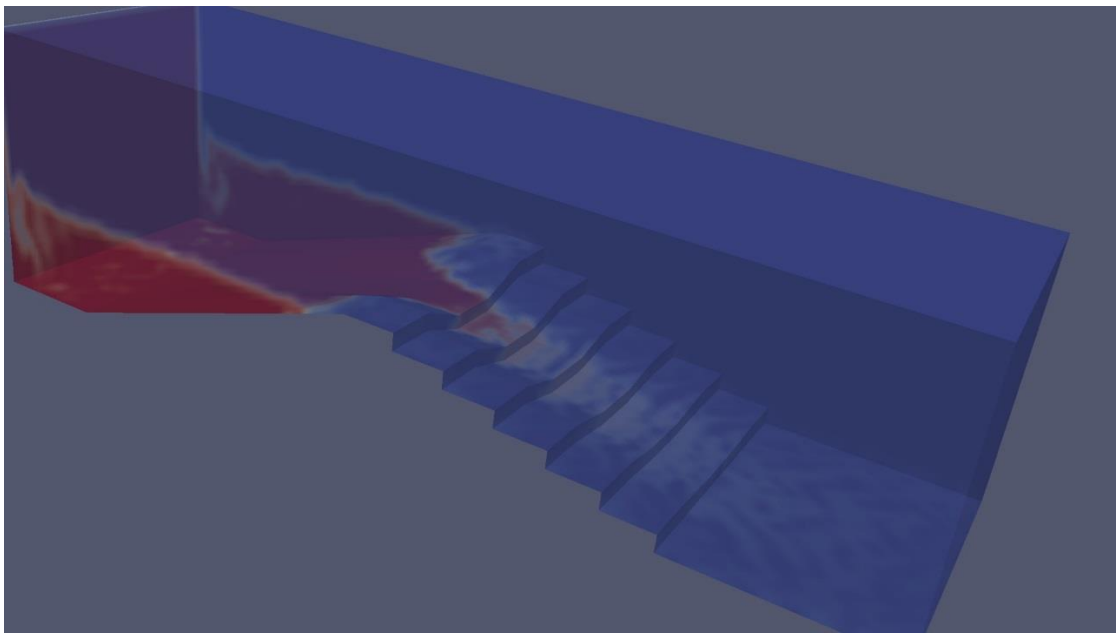

Figure 64. Stepped dam with auxiliary spillway after 7.5 seconds

During the remainder of the thirty seconds, water spread across the entire span of the last step due to the slow speed of the water, causing it to build up on each stair, as shown in Figure 65. The waves caused extra water to go over the sides.

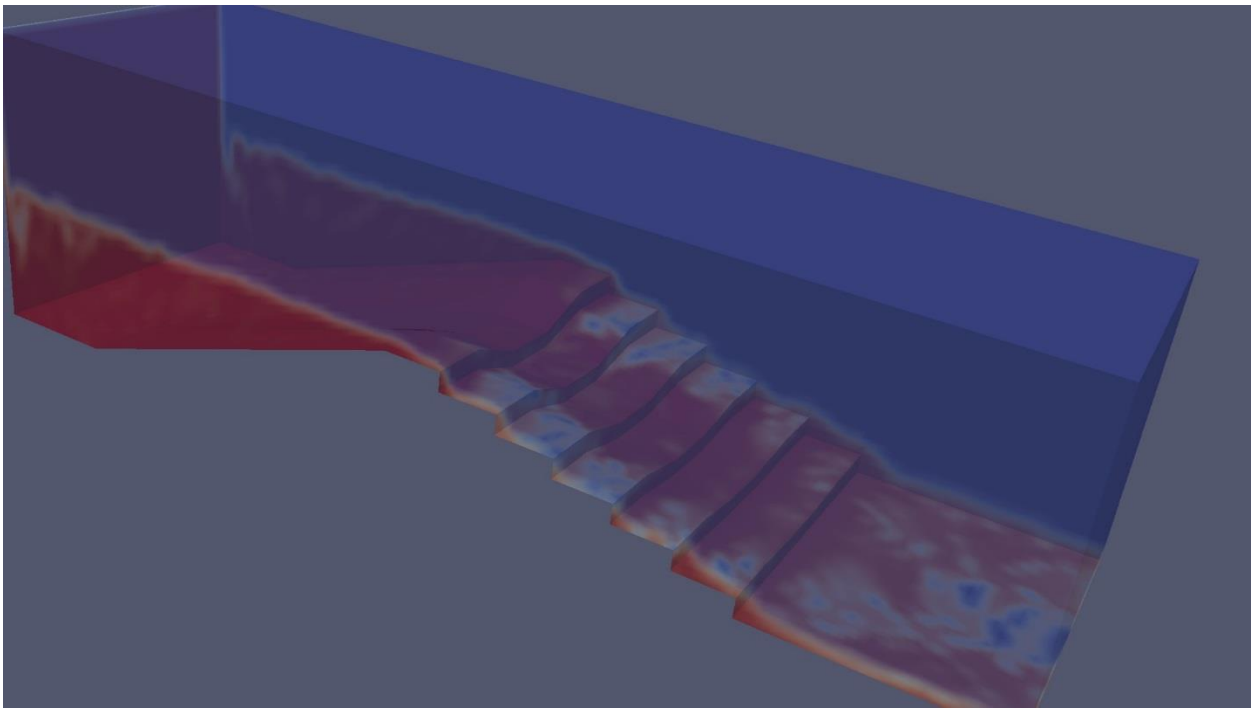

Figure 65. Stepped dam with auxiliary spillway after 30 seconds

Unlike the smooth dam with auxiliary channel in Chapter 4.2, velocity at the bottom was the slowest instead of the fastest. The auxiliary spillway and the stairs slowing down the water demonstrated how fanning the water out can slow the flow of water described in Chapter 2.3. The overall velocity also tended to be about half the velocity of the dam in Chapter 4.2, as illustrated in Figure 66.

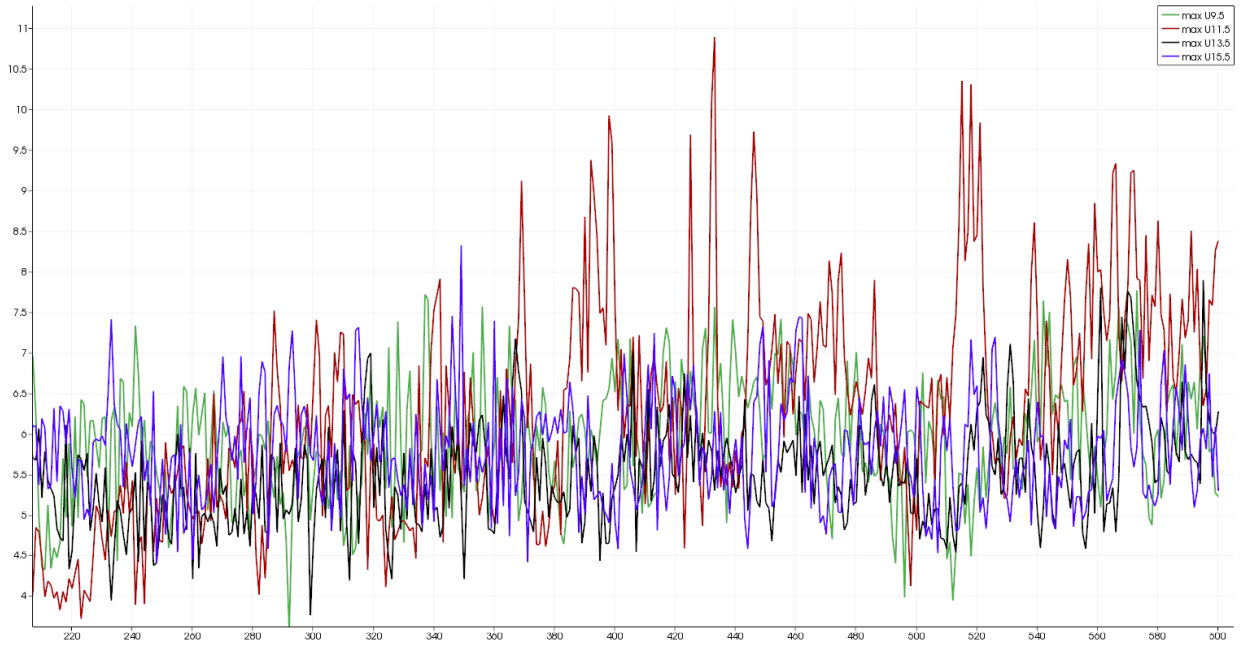

Figure 66. Velocity of the stepped dam with auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s

This dam produced a similar graph to the smooth dam with auxiliary spillway; however, compared to the smooth dam with auxiliary spillway, dynamic pressure increased at a higher rate throughout the simulation, as shown in Figure 67.

Figure 67. Dynamic pressure of the stepped dam with auxiliary spillway over time, where the x-axis represents the frame number and the y-axis represents maximum dynamic pressure in pascals at each of the four slices

4.7 Stepped Dam with Converging Wall

The stepped dam with converging wall took 3818 seconds (1 hour and 4 minutes) to create thirty seconds of data with the interFoam solver. The water line was placed .2 meters below the top of the dam at 1.8 meters, as shown in Figure 68.

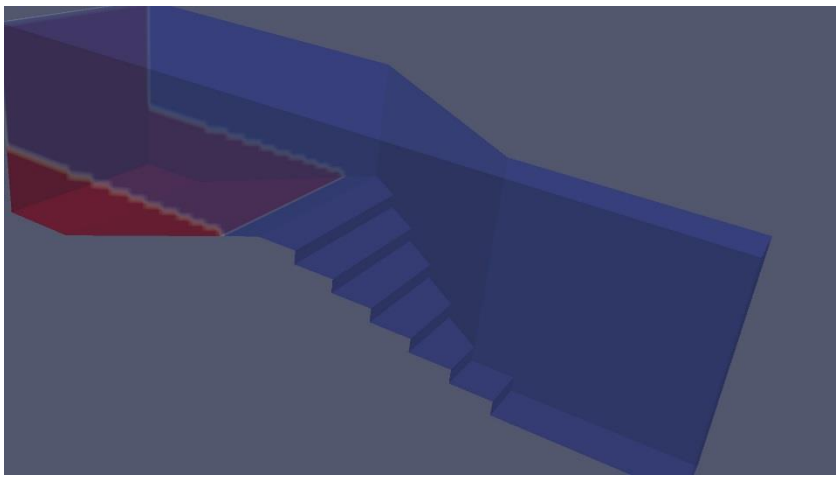

Figure 68. Initial conditions for the stepped dam with converging wall

Since the upstream side of this dam was identical to the dam in Chapter 4.1, initial results were the identical. The first wave pushed water over 3.5 seconds into the simulation, as shown in Figure 69.

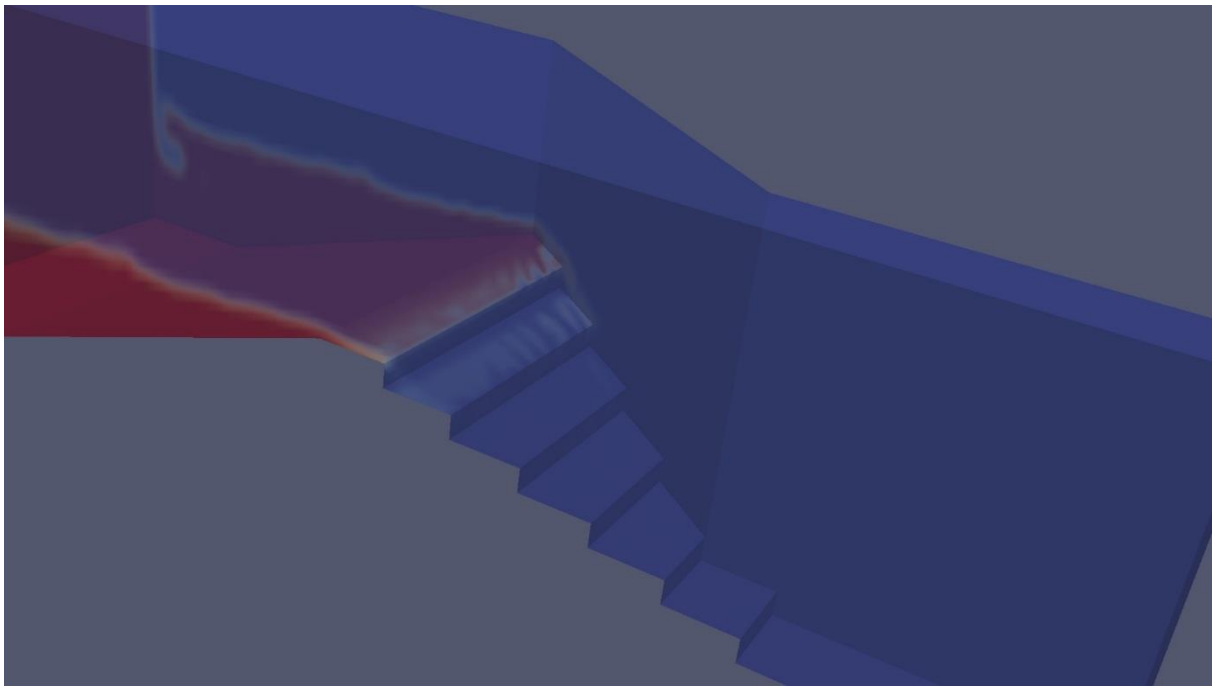

Figure 69. Stepped dam with converging wall after 3.5 seconds

After 7.5 seconds, the water was flowing over the crest, as shown in Figure 70.

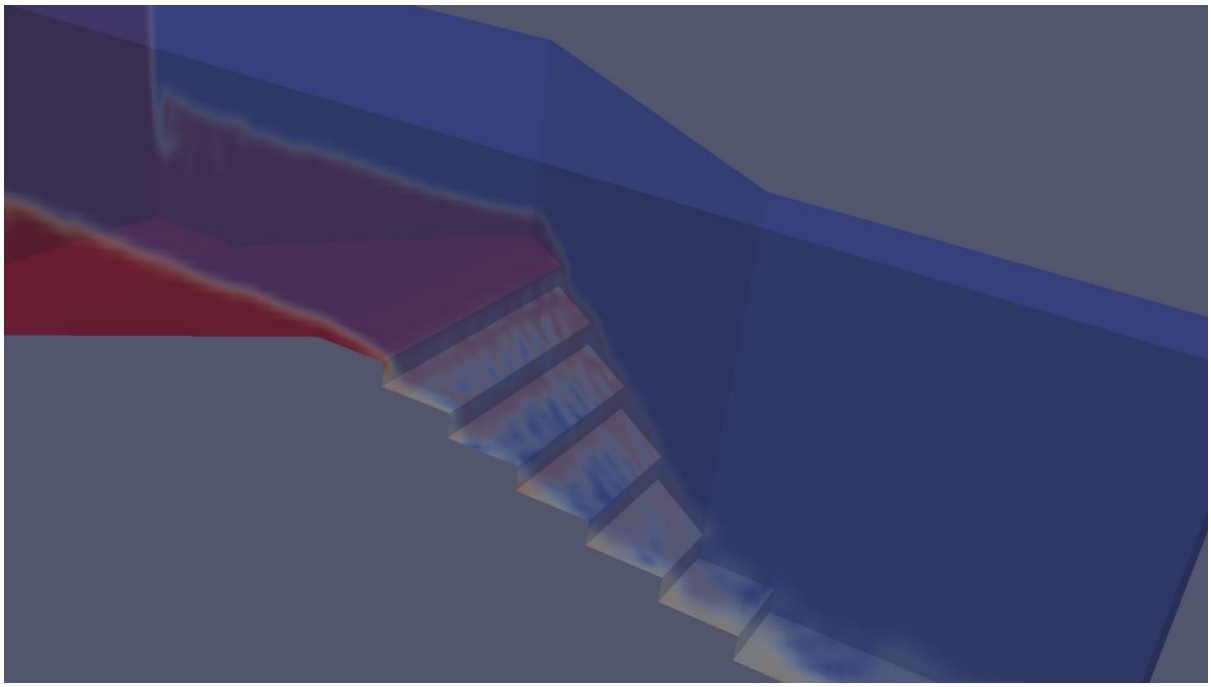

Figure 70. Stepped dam with converging wall after 7.5 seconds

For the remainder of the thirty second simulation, the water traveled along the converging wall and stayed spread out along the span of each step, as shown in Figure 71. This is in contrast to the dam in Chapter 4.4, where the water stayed along the converging wall. This much water on the step also indicatives that it is flowing very slowly.

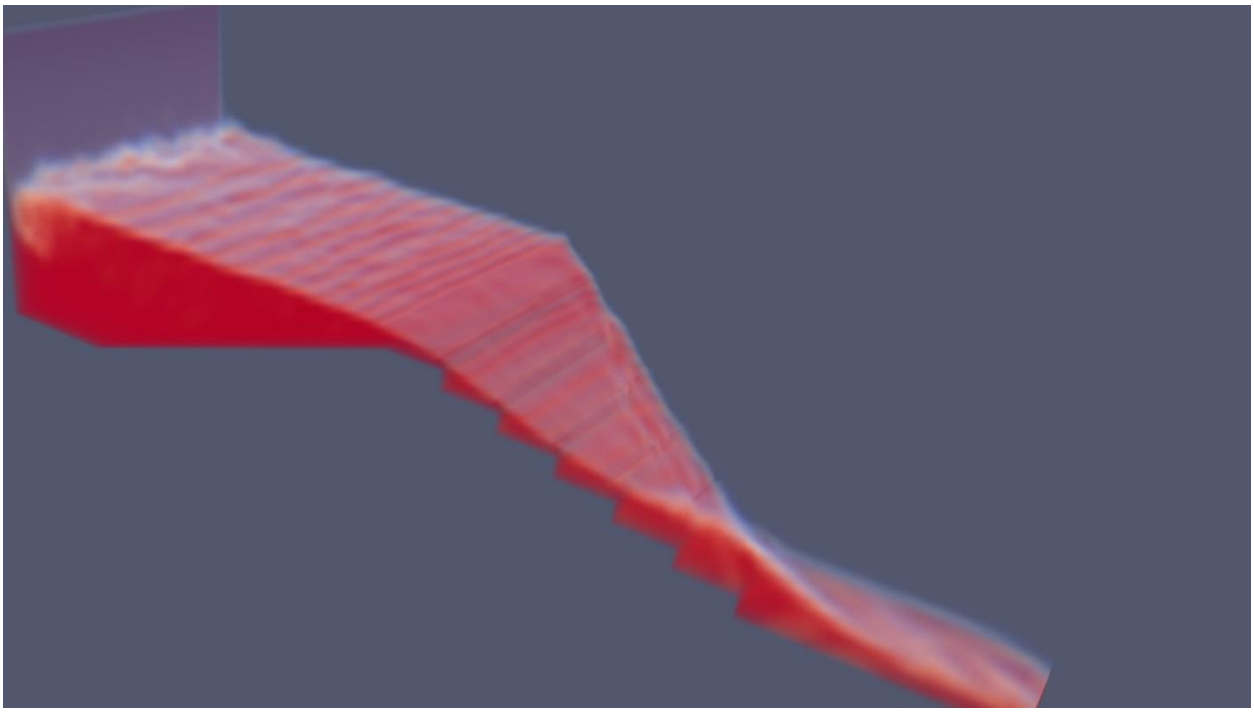

Figure 71. Volume view of the stepped dam with converging wall after 30 seconds

Comparison of the velocity graph in Figure 72 to the converging dam in Chapter 4.4 indicates that the water in this dam was traveling approximately 2/3rds the speed of smooth dam. Unlike the dam in Chapter 4.4, the water at the bottom of this dam is moving the fastest.

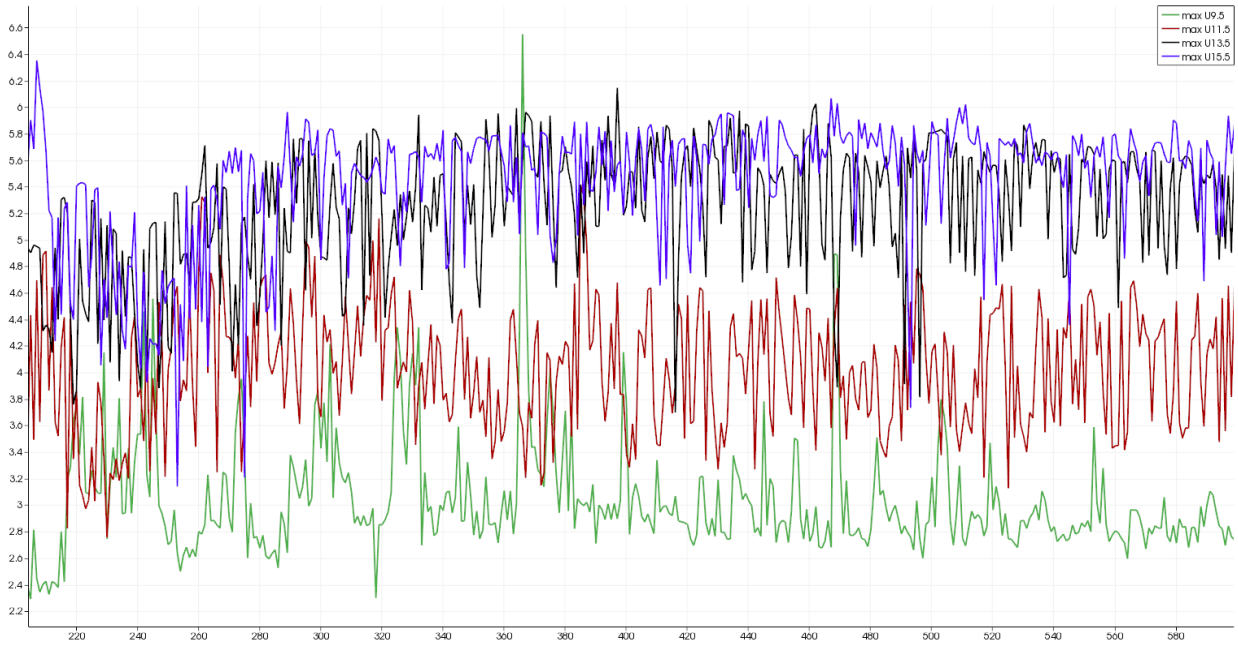

Figure 72. Velocity of the stepped dam with converging wall over time, where the x-axis represents the frame number and the y-axis represents the magnitude of velocity in m/s

The stepped dam with converging wall produced the highest dynamic pressure. As shown in figure 73, the graph is similar to the one produced by the smooth dam with converging wall, but does not fluctuate as rapidly.

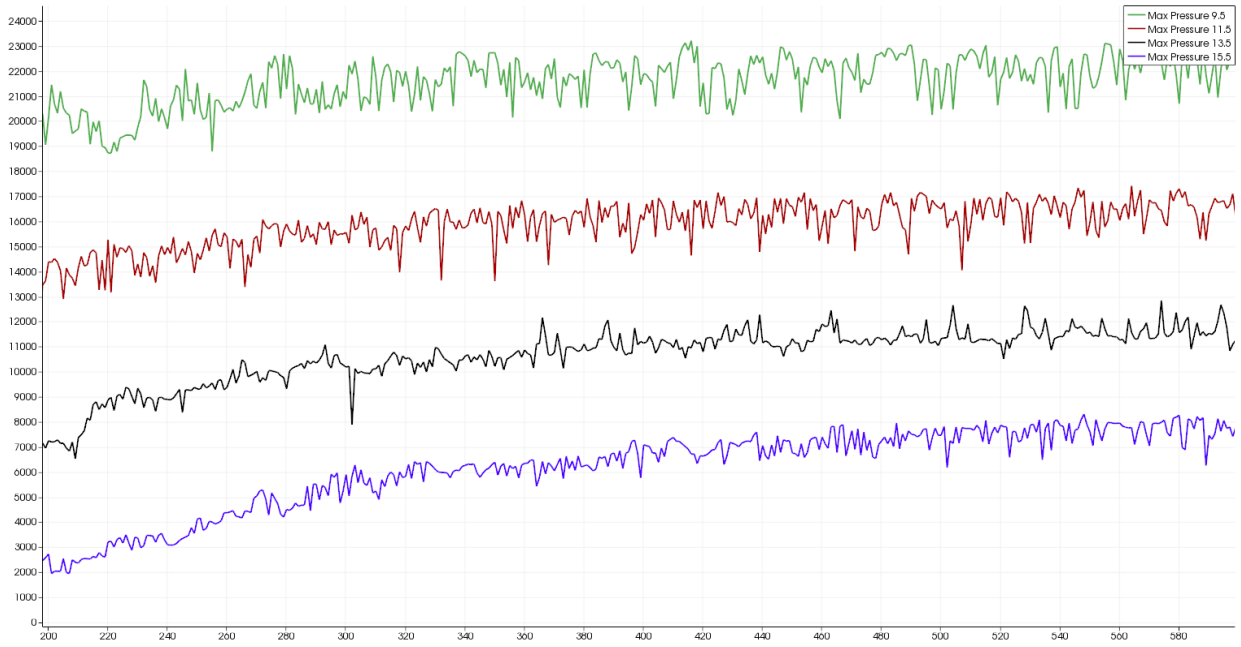

Figure 73. Dynamic pressure of the stepped dam with converging wall over time, where the x-axis represents the frame number and the y-axis represents maximum pressure in pascals at each of the four slices

4.8 Performance Analysis

OpenFOAM utilizes the Message Passing Interface (MPI) standard to allow its solvers to run on multiple threads (Parallel Computing, 2014). According to the standard, "MPI addresses primarily the message-passing parallel programming model" (MPI: A Message-Passing Interface Standard, 2012, p. 33). The standard also describes the model as the act of moving data from one process's address space to another process's address space through "cooperative operations on each process". OpenFOAM is able to work with this model by dividing sections of the mesh into sub-domains and passing information to neighboring sections when flow runs between them (Parallel Computing, 2014). As a user adds additional threads, the mesh is divided into smaller pieces, meaning that each thread has less to calculate for its individual sub-domain. As a result, less work is required for each thread, but it requires the threads to "talk" to each more as the flow moves throughout the mesh and each sub-domain. If a user continues to increase the number of threads used, the threads will eventually spend more time communicating with one another than actually working on the problem, consequently slowing down the solver. The interFoam solver was run on each dam to create one second of data with 1, 2, 4, 6, 8, 10, and 12 threads to

determine which number should be used to create thirty second simulations. By examining Table 1, six threads were determined to be the critical threshold for the computer running the simulation, because the time required to solve one second of the simulation increased if additional threads were added.

Table 1. Comparison of how the number of threads on each dam affected the how fast the solver was. The number represents the time, in seconds, required to solve one second of the simulation.

Time (in seconds) to solve one second	1 Thread	2 Threads	4 Threads	6 Threads	8 Threads	10 Threads	12 Threads
Smooth Dam	704	376	220	204	216	239	306
Smooth dam with Auxiliary Spillway	780	458	286	261	289	296	505
Smooth Dam with Prinipal and Auxiliary Spillway	843	512	269	243	279	289	491
Smooth Dam with Converging Wall	532	311	164	159	172	180	301
Stepped Dam	822	332	249	243	271	283	360
Stepped Dam with Auxiliary Spillway	647	391	259	261	236	267	482
Stepped Dam with Converging Wall	435	261	159	155	167	180	353

Chapter 5 - Conclusions

As the United States Department of Agriculture (USDA) continues to research the effectiveness of various earth dam designs, OpenFOAM exhibits capabilities that make it a viable option for conducting computational fluid dynamics research. OpenFOAM allows the creation of new dam simulations and the ability to modify existing dams by editing the text files described in Chapter 2.2. As new discoveries are made in the science of dam construction, they can be implemented and tested in OpenFOAM.

This report looked into implementing and simulating various dam design features in OpenFOAM, based on experimental results from the USDA experimental hydraulic engineering research unit. Among the models created, features such as principal spillways, auxiliary spillways, converging walls, and steps were included. Each of these features changed the flow of water down the downstream side of the dam. These results can be confirmed in figures in Chapter 4. These models can serve as a basis for future, more complex dam analysis. Employing these research techniques would also likely save time and money in designing and testing new dams.

With its ability to use Message Passing Interface, OpenFOAM can also be scaled up to run faster on more powerful computers. This allows for more complex simulations and algorithms to be processed at a quicker rate. As a result of these findings, it is indeed possible to construct converged stepped spillway models in OpenFOAM.

References

Basic Input-Output File Format. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/docs/user/basic-file-format.php

BlockMesh. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/docs/user/blockMesh.php

DecomposePar. (2009, 11 25). Retrieved 3 2014, from The OpenFOAM Foundation: http://openfoamwiki.net/index.php/DecomposePar

Design of Small Dams. (1987). Washington, DC: U.S. Government Printing.

D.M. Temple, G.J. Hanson, and M.L. Neilsen, "WinDAM -- Analysis of overtopped earth embankment dams", In Proc. of the ASABE Annual Conference, Paper Number 062105, 2006.

Features of OpenFOAM. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/features/

HexBlocker. (2014, 2 9). Retrieved 3 2014, from Github: https://github.com/nicolasedh/hexBlocker

Hunt, S. L., & Kadavy, K. C. (2013). Inception Point for Embankment Dam Stepped Spillways. *Journal of Hydraulic Engineering* .

Hunt, S. L., Kadavy, K. C., Abt, S. R., & Temple, D. M. (n.d.). Impact of Convering Chute Walls for Roller Compacted Concrete Stepped Spillways. *Journal of Hydraulic Engineering* , 2008.

Hunt, S. L., Temple, D. M., Abt, S. R., Kadavy, K. C., & Hanson, G. (2012). Converging Stepped Spillways: Simplified Momentum Analysis Approach. *Journal of Hydraulic Engineering* .

InterFoam. (2013, 1 31). Retrieved 3 2014, from The OpenFOAM Foundation: http://openfoamwiki.net/index.php/InterFoam

Introduction. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/docs/user/userch1.php#x3-20001

K. Visser, R.D. Tejral, and M. Neilsen, "WinDAM C Earthen Embankment Internal Erosion Analysis Software", in Proceedings of the 30th Annual Conference of the Association of State Dam Safety Officials, Providence, RI, Sept. 8-12, 2013.

M.L. Neilsen, "Global sensitivity analysis of dam erosion models", in Proceedings of the 10th International Conference on Scientific Computing, Paper No. CSC-3502, July 22-25, 2013.

Mesh Description. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/docs/user/mesh-description.php

Mesh Generation. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/features/mesh-generation.php

MPI: A Message-Passing Interface Standard. (2012). Knoxville: University of Tennessee.

Parallel Computing. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/features/parallel-computing.php

Standard Solvers. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/features/standard-solvers.php

Stephens, T. (2010). *Manual on small earth dams.* Rome: FAO.

Taghizadeh, H., Neyshabour, S. A., & Ghasemzadeh, F. (2012). Dynamic Pressure Fluctuations in Stepped Three-Side Spillway. *Iranica Journal of Energy & Environment* .

Temple, D. M., Hanson, G. J., & Hunt, S. L. (2010). Observations on dam overtopping breach processes and prediction. *The Journal of Dam Safety* , 28-33.

Temple, D. M., Hanson, G. J., & Neilsen, M. L. (2006). *WINDAM - Analysis of overtopped earth embankment dams.*

Temple, D. M., Wibowo, J. L., & Neilsen, M. L. Erosion of earth spillways. *United States Society on Dams. Reducing Risk from Extreme Events.*, (pp. 331-339).

The ParaView Post-Processor. (2014). Retrieved 3 2014, from The OpenFOAM Foundation: http://www.openfoam.org/features/paraview.php

United States Army Corps of Engineers. (2005). *Hydrologic Modeling System HEC-HMS User's Manual.* Davis, CA.: USACE Hydrologic Engineering Center.

United States Department of Agriculture, Natural Resources Conservation Service. (1997). Earth spillway erosion model. In *Dams, National Engineering Handbook.*

United States Department of Agriculture, Soil Conservation Service. (1956). *Hood inlets for culvert spillways.*

United States Society on Dams. (2006). *Erosion of unlined spillways.*

WinDAM C Dam Overtopping and Internal Erosion Model. (2012).