INL/EXT-15-37091

Light Water Reactor Sustainability Program

Flooding Capability for River-based Scenarios

Curtis Smith, Steve Prescott, Emerald Ryan, Donna Calhoun,
Ramprasad Sampath, S. Danielle Anderson, Cody Casteneda

October 2015

DOE Office of Nuclear Energy

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/EXT-15-37091

Light Water Reactor Sustainability Program

Flooding Capability for River-based Scenarios

Curtis Smith – INL
Steven Prescott – INL
Emerald Ryan – Idaho State University
Donna Calhoun – Boise State University
Ramprasad Sampath – Centroid Labs
S. Danielle Anderson – INL
Cody Casteneda – Boise State University

October 2015

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov/lwrs

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

ABSTRACT

This report describes the initial investigation into modeling and simulation tools for application of riverine flooding representation as part of the Risk-Informed Safety Margin Characterization (RISMC) Pathway external hazards evaluations. The report provides examples of different flooding conditions and scenarios that could impact river and watershed systems. Both 2D and 3D modeling approaches are described.

.

CONTENTS

ABSTRACT iii

FIGURES v

ACRONYMS vii

1. INTRODUCTION 1
   1.1 Background 1
   1.2 Large Scale Flooding Methods 1

2. 2D GRID-BASED APPROACHES 2
   2.1 Watershed Codes 2
   2.2 Shallow Wave Codes 3
   2.3 Dam Break Example 4
   2.4 GeoClaw Setup 7
   2.5 Teton Dam Break GeoClaw Results 7

3. PARTICLE-BASED APPROACHES 10
   3.1 DualSPHysics 10
      3.1.1 Running the Code 10
      3.1.2 Code Output 12
   3.2 Flooding Model and Scenarios 13
      3.2.1 Creating the Simulated Flood 15
      3.2.2 Flooding Model 17
      3.2.3 Flooding Scenarios 17
   3.3 Simulation Results 19
      3.3.1 Case 1 19
      3.3.2 Case 2 25
      3.3.3 Case 3 31
      3.3.4 Case 4 37
   3.4 Large Scale SPH Summary 43

4. COUPLING 2D AND SPH METHODS 44
   4.1 Boundary Data 44
   4.2 SPH Coupling 48

5. CONCLUSIONS 50

6. REFERENCES 51

FIGURES

Figure 1. Representation of the SWMM RUNOFF algorithm [4] 2
Figure 2. Example of the HEC-HMS graphical user interface. 3
Figure 3. Conservative form of Navier-Stokes shallow-water equations. 4
Figure 4. GeoClaw Adaptive Mesh Refinement 4
Figure 5. Teton Dam site after completion and the beginning of failure. 5
Figure 6. Teton Dam site showing complete failure 6
Figure 7. Topographical area selected for initial dam break simulation. 6
Figure 8. Initial GeoClaw simulation setup. 7
Figure 9. GeoClaw dam break simulation at time = 1.2e3 8
Figure 10. GeoClaw dam brea simulation at time = 2.7e3 9
Figure 11. DualSPHysics DPI constants tab. 11
Figure 12. DualSPHysics DPI Geometry tab 11
Figure 13. DualSPHysics DPI parameters tab 12
Figure 14. Flooding scenario topography 14
Figure 15. Water container option 15
Figure 16. Horizontal water container option 16
Figure 17. Water slab option 16
Figure 18. Flooding scenarios 18
Figure 19. Case 1 beginning 19
Figure 20. Case 1 75 seconds 20
Figure 21. Case 1 150 seconds 20
Figure 22. Case 1 225 seconds 21
Figure 23. Case 1 300 seconds 21
Figure 24. Case 1 375 seconds 22
Figure 25. Case 1 450 seconds 22
Figure 26. Case 1 525 seconds 23
Figure 27. Case 1 600 seconds 23
Figure 28. Case 1 675 seconds 24
Figure 29. Case 1 750 seconds 24
Figure 30. Case 2 beginning 25
Figure 31. Case 2 75 seconds 26
Figure 32. Case 2 150 seconds 26
Figure 33. Case 2 225 seconds 27

Figure 34. Case 2 300 seconds 27
Figure 35. Case 2 375 seconds 28
Figure 36. Case 2 450 seconds 28
Figure 37. Case 2 525 seconds 29
Figure 38. Case 2 600 seconds 29
Figure 39. Case 2 675 seconds 30
Figure 40. Case 2 750 seconds 30
Figure 41. Case 3 beginning 31
Figure 42. Case 3 75 seconds 32
Figure 43. Case 3 150 seconds 32
Figure 44. Case 3 225 seconds 33
Figure 45. Case 3 300 seconds 33
Figure 46. Case 3 375 seconds 34
Figure 47. Case 3 450 seconds 34
Figure 48. Case 3 525 seconds 35
Figure 49. Case 3 600 seconds 35
Figure 50. Case 3 675 seconds 36
Figure 51. Case 3 750 seconds 36
Figure 52. Case 4 beginning 37
Figure 53. Case 4 75 seconds 38
Figure 54. Case 4 150 seconds 38
Figure 55. Case 4 225 seconds 39
Figure 56. Case 4 300 seconds 39
Figure 57. Case 4 375 seconds 40
Figure 58. Case 4 450 seconds 40
Figure 59. Case 4 525 seconds 41
Figure 60. Case 4 600 seconds 41
Figure 61. Case 4 675 seconds 42
Figure 62. Case 4 750 seconds 42
Figure 63. Map of a NPP added below the Teton dam. 44
Figure 64. GeoClaw gauge location and simulation data at time = 1.2e3 47
Figure 65. GeoClaw gauge location and simulation data at time = 2.7e3 47

ACRONYMS

AMR	Adaptive Mesh Refinement
ASCII	American Standard Code for Information Interchange
CFD	computational fluid dynamics
CPU	central processing unit
CSV	comma separated-values file
CUDA	Compute Unified Device Architecture
DOE	Department of Energy
DPI	DualSPHysics Pre-processing Interface
GPU	graphics processing unit
GUI	graphical user interface
INL	Idaho National Laboratory
ISU	Idaho State University
LWRS	Light Water Reactor Sustainability
NPP	nuclear power plant
OBJ	object file
PLY	polygon file format
PRA	probabilistic risk assessment
RISMC	Risk-Informed Safety Margin Characterization
SPH	Smoothed Particle Hydrodynamics
SSCs	structures, systems, and components
STL	stereolithography file
SW	Shallow Water
TXT	text file
VTK	visualization toolkit file
XML	Extensible Markup Language

Flooding Capability for River-based Scenarios

1. INTRODUCTION

1.1 Background

One of the primary avenues for collaboration with industry is through the Risk-Informed Safety Margin Characterization (RISMC) Pathway Industry Applications. The primary purpose of Industry Applications in the RISMC Pathway is to demonstrate advanced risk-informed decision making capabilities for relevant industry questions. The end goal of these activities is the full adoption of the RISMC tools and methods by industry applied to their decision making process. For the evaluation of external hazards within the RISMC Pathway [1], we have identified that hazards external to a nuclear facility may negatively impact a variety of structures, systems, and components (SSCs) from direct damage (e.g., failure during a fire) or indirect damage (e.g., consequential failure from a flood following a pipe break), there is a possibility that initiating events, reduced redundancy levels, reduced reliability, or degraded safety barrier may be realized, thereby increasing the likelihood or severity of potential accident scenarios.

External events are a class of hazards to nuclear facilities that originates external to the plant. These external events are initiating event that has the initial deviation caused by a hazard located outside the normal plant SSCs. Physical impacts such as fires, floods, and earthquakes are typically included in this group of initiating events. In this report, we focus on modeling and simulation of floods, specifically representation for river-caused floods – or riverine flooding. Riverine flood modeling brings up unique issues:

- The expanse of land area encompassed by the flooding realm can be very large compared to other initiating events. The water reflected in this large land area may need to be tracked, which tend to make detailed models a challenge due to the size of the analysis domain.
- Warning time. It is possible that some flooding events occur over days and weeks, these types of events, when modeled, should include the possibility for preemptive actions.
- Slow (relatively) change in time, but possible long duration events.

This report does not provide an investigation into the frequency of riverine-types of floods. In general, these types of floods may be caused by localized intense precipitation, seasonal flooding cycles, and dam failures. The quantification of the frequency for these events would be site specific, but could possibly be treated by Bayesian approaches.

1.2 Large Scale Flooding Methods

There are several methods or codes used for large scale flooding scenarios that have been used historically to determine the flood plain for heavy rainfall or dam breach events. Several of these watershed codes such as HEC-HMS and SWMM have been validated and approved for use by the FEMA [2]. These codes use various algorithms applied to a 2D grid to determine the fluid height for each grid cell over time.

While these methods are great and provide ample information for many scenarios, they are limited in scope. Water height and velocity for a given 2D area can be calculated, but more in-depth measurements and any 3D effects such as splashing or component contact cannot be performed. In this repot we investigate the capabilities of 2D codes, what advantages or limitations Smoothed Particle Hydrodynamics (SPH) may have, and a proposed method for coupling 2D methods and SPH to combine the benefits of both methods. Finally we demonstrate how to use the coupling approach on a dam break scenario flooding a nuclear facility along a river.

2. 2D GRID-BASED APPROACHES

Hydrology engineers have developed several models for predicting water movement of watershed or runoff from rainfall events and wave propagation from tsunamis or dam breaks. These methods use a topography or bathymetry model and input parameters to propagate the flow of water on the model over time.

2.1 Watershed Codes

One example of a watershed code developed by the EPA is SWMM [3]. It is a robust code with a several program modules such as the RUNOFF block. The RUNOFF block generates a runoff hydrograph from s sub-catchment and employs a surface water budget approach. (See Figure 1) The topography model is broken up into a grid with input values applied. Next, the SWMM module equations are solved for each cell of the grid and the results propagated to the neighboring cells as part of the input values for the next time step solve.

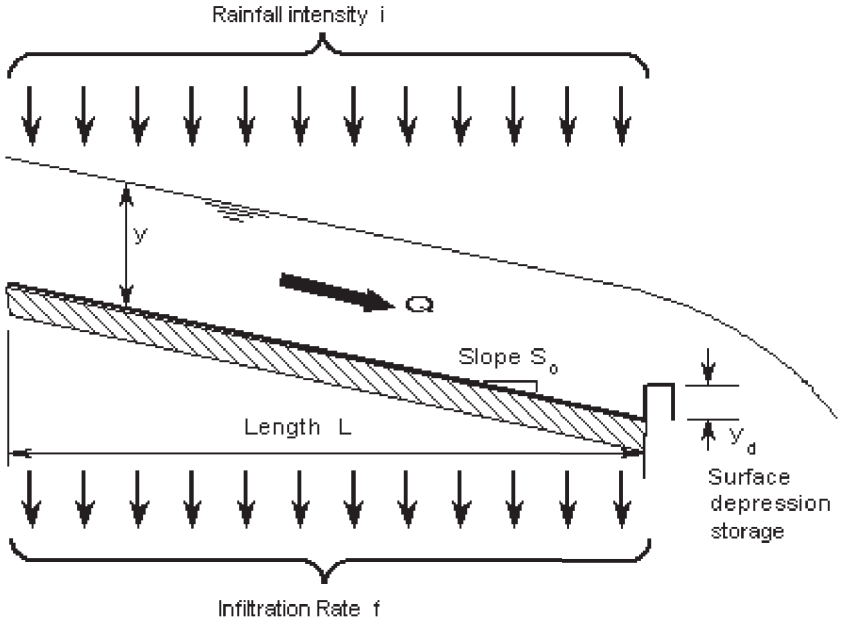


Figure 1. Representation of the SWMM RUNOFF algorithm [4]

The Army Corps of Engineers developed a Hydrologic Modeling System called HEC-HMS [5]. This system was designed to handle a large range of problems including flood hydrology, watershed runoff, and river basin water supplies. Like SWMM, it separates the water cycle into separate modules and breaks up the topography into sections for solving the modules. Recent versions have also included a graphical user interface for easy visualization. (See Figure 2)

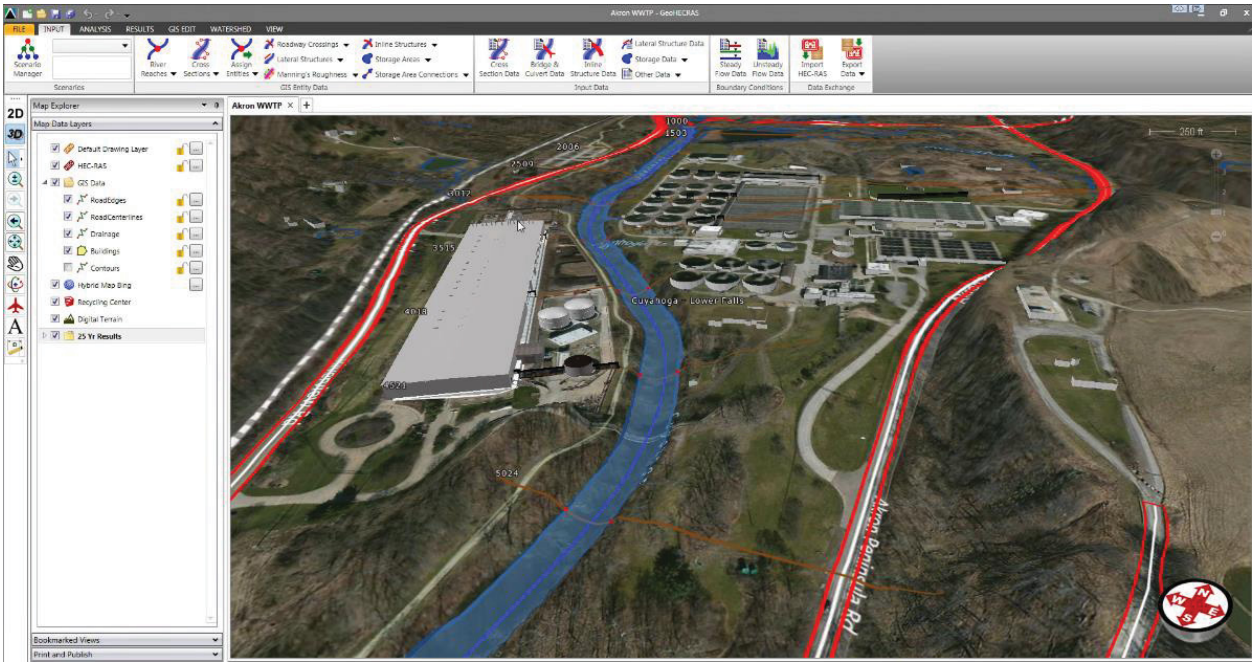


Figure 2. Example of the HEC-HMS graphical user interface.

2.2 Shallow Wave Codes

Other large scale simulation codes are designed based upon the shallow wave equations. (See Figure 3) GeoClaw is one such code. GeoClaw uses the depth-averaged shallow wave equations to compute water propagation over topography. These equations are derived from integrating 3-dimensional governing equations in the vertical direction, thus reducing the simulation to a time dependent, two-dimensional computation [6].

Adaptive mesh refinement (AMR), built into GeoClaw, is the method of refining the accuracy of the computed solution in areas where the solution is more sensitive. This method is dynamic, meaning that as the simulation progresses and turbulent or sensitive areas of the solution change, the level of grid refinement in these areas also changes. AMR in GeoClaw uses refined grids in areas where the water height is non-zero and coarser grids on dry land or over areas where flooding has not yet occurred. (See Figure 4)

Specialized routines built into GeoClaw allow the computation to accurately capture small perturbations between the water and given topography. Additional physical modeling parameters, such as a Manning friction coefficient and dry tolerances, stabilize the code in regions near edges of the flow. Conservation of total water mass is ensured through the use of a conservative, flux based (finite volume) scheme.

$$\frac{\partial \eta}{\partial t} + \frac{\partial (\eta u)}{\partial x} + \frac{\partial (\eta v)}{\partial y} = 0$$

$$\frac{\partial (\eta u)}{\partial t} + \frac{\partial}{\partial x}\left(\eta u^2 + \frac{1}{2} g \eta^2\right) + \frac{\partial (\eta u v)}{\partial y} = 0$$

$$\frac{\partial (\eta v)}{\partial t} + \frac{\partial (\eta u v)}{\partial x} + \frac{\partial}{\partial y}\left(\eta v^2 + \frac{1}{2} g \eta^2\right) = 0.$$

Figure 3. Conservative form of Navier-Stokes shallow-water equations.

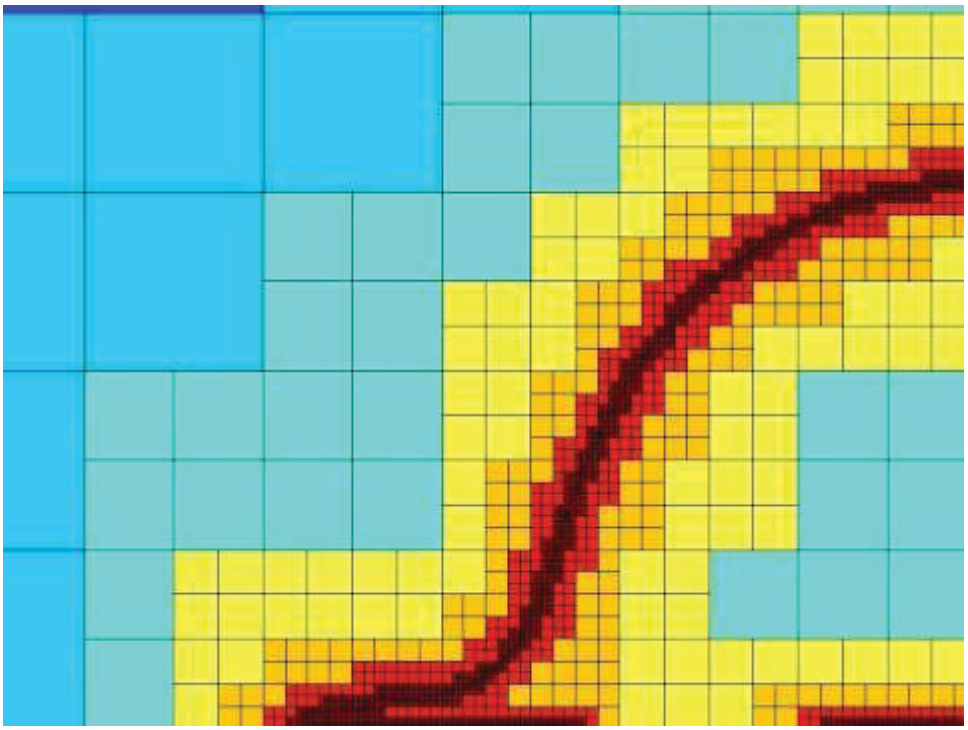


Figure 4. GeoClaw Adaptive Mesh Refinement

2.3 Dam Break Example

The GeoClaw package traditionally is used for tsunami simulations, but has been demonstrated as a method to simulate dam failure events. D.L. George used it to simulate the Malpasset dam-break flood in France, 1959 [7]. Although GeoClaw does not have all the calculation modules as some watershed codes, simple rain fall and infiltration could be done through source term methods. Because this one package could be used for multiple events we selected it for this dam break scenario and coupling with SPH as described later.

The dam break simulation is designed after an actual event, the Teton Idaho Dam failure of June 5 1976. (See Figure 5) This event allowed for topography data to be used and allows for validation of continued work against actual data. The initial test case was set up for an area several miles above and below the dam. (See Figures 6 and 7) Continued work could expand the simulation to the flood plane from Sugar City to Rexburg.

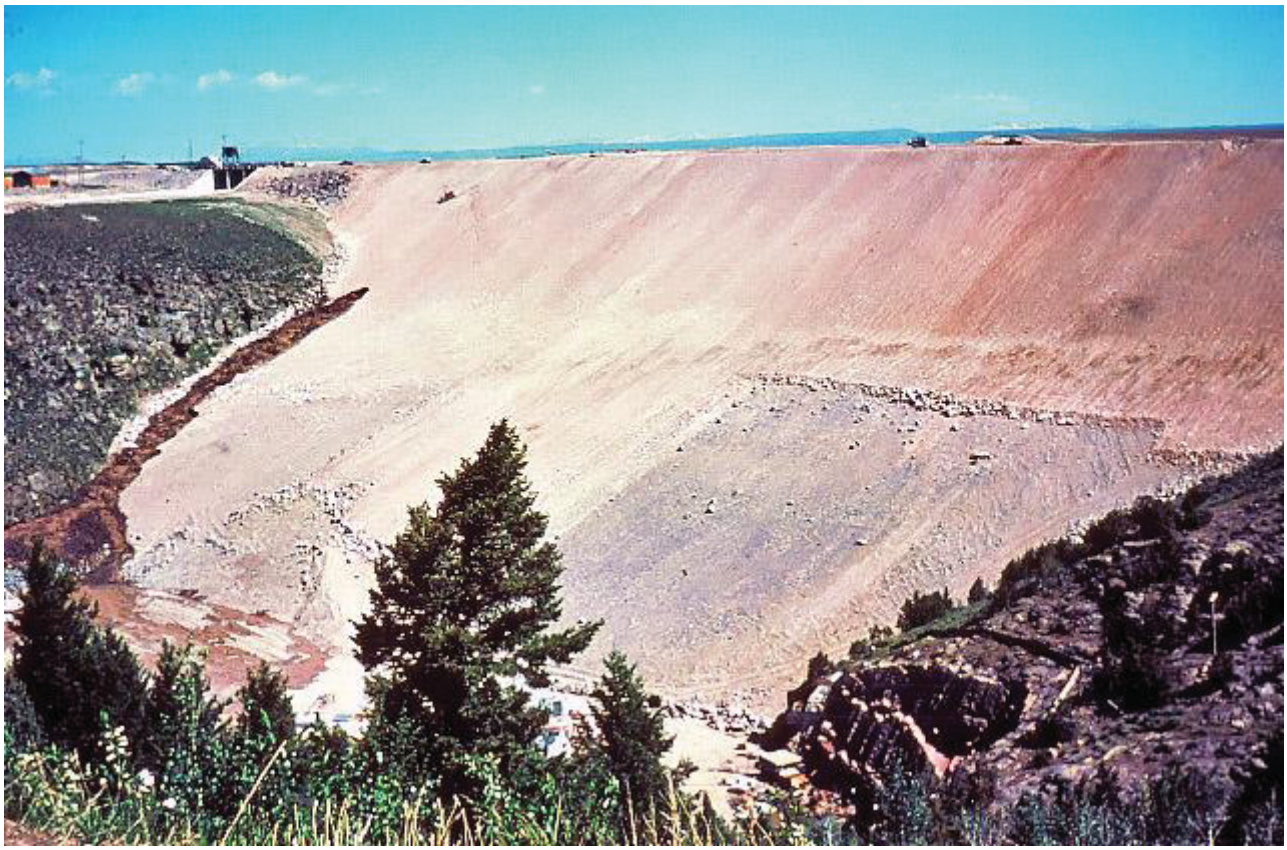


Figure 5. Teton Dam site after completion and the beginning of failure.

Figure 6. Teton Dam site showing complete failure

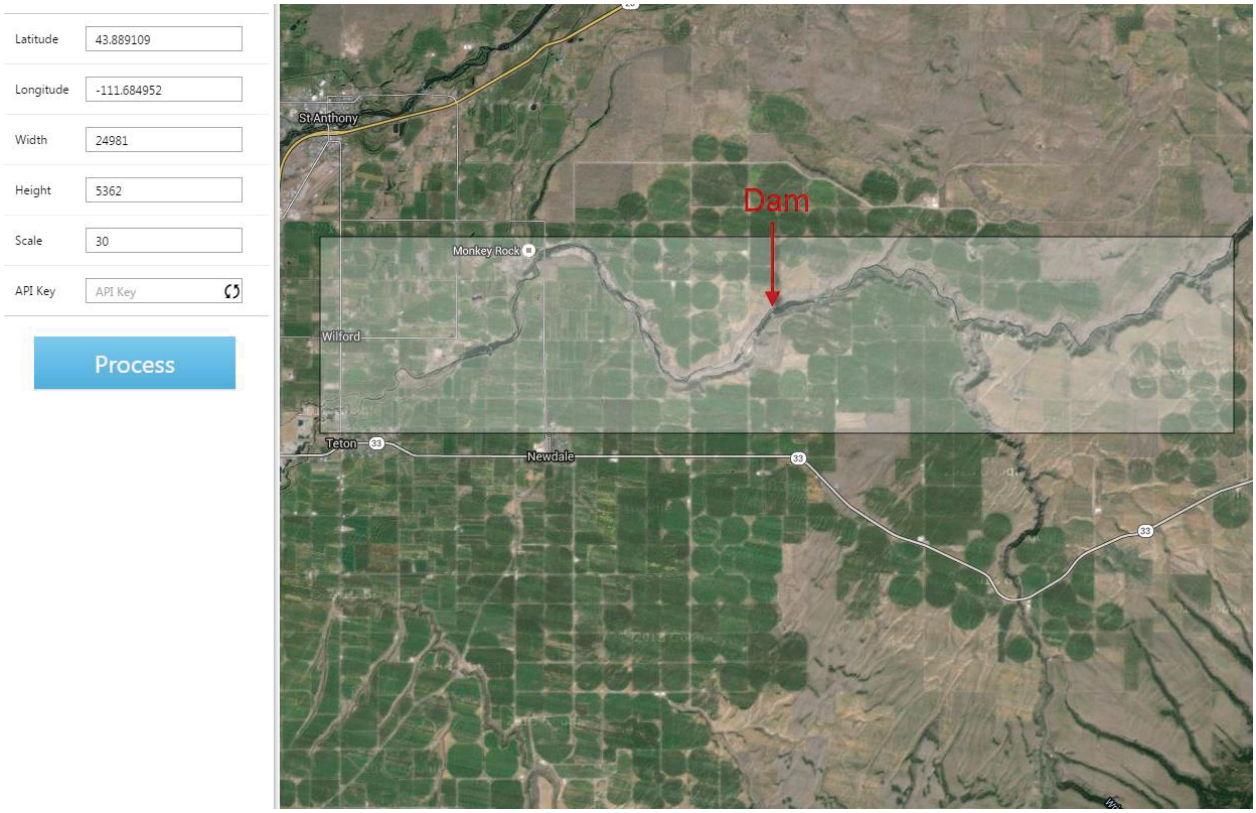


Figure 7. Topographical area selected for initial dam break simulation.

2.4 GeoClaw Setup

The Teton Dam failure problem is set up as an initial value problem in GeoClaw. GeoClaw requires one or more overlapping topography maps describing the terrain over which to carry out the simulation. This data can be obtained from a variety of sources, but is most useful in standard Digital Elevation Map (DEM) format file. The region to be simulated must be a rectangular region fully contained within the topography data. The river upstream of the dam is initially filled with water, creating a vertical wall of water where the dam was originally located. At the beginning of the simulation the vertical wall of water is released as if the dam was instantaneously breached. Because this is just initial testing and a proposed design for a coupling method, the erosion process was not considered in the simulation. Geoclaw's shallow water wave equations compute the propagation of the flow as it moves through the remaining riverbed and onto the flood plain. (See Figure 8) For more detail on GeoClaw methods refer to the following references [8], [9].

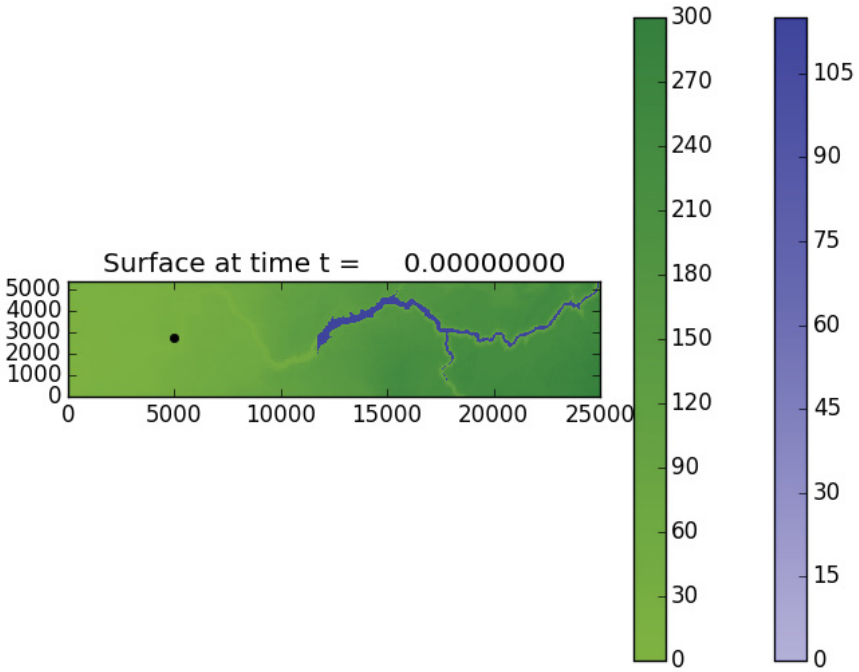


Figure 8. Initial GeoClaw simulation setup.

2.5 Teton Dam Break GeoClaw Results

Initial water height is set to 115 meters behind the dam. The large black dot shows the approximate location of a power plant. Figure 9 and Figure 10 show the progress of the flood waters area and depth over the simulation space. Gauges can be added to the simulation to get precise fluid data for the defined location over the simulation time. Videos of the results can also be produced to get a sense of the time-dependent flow of water from this riverine flood).

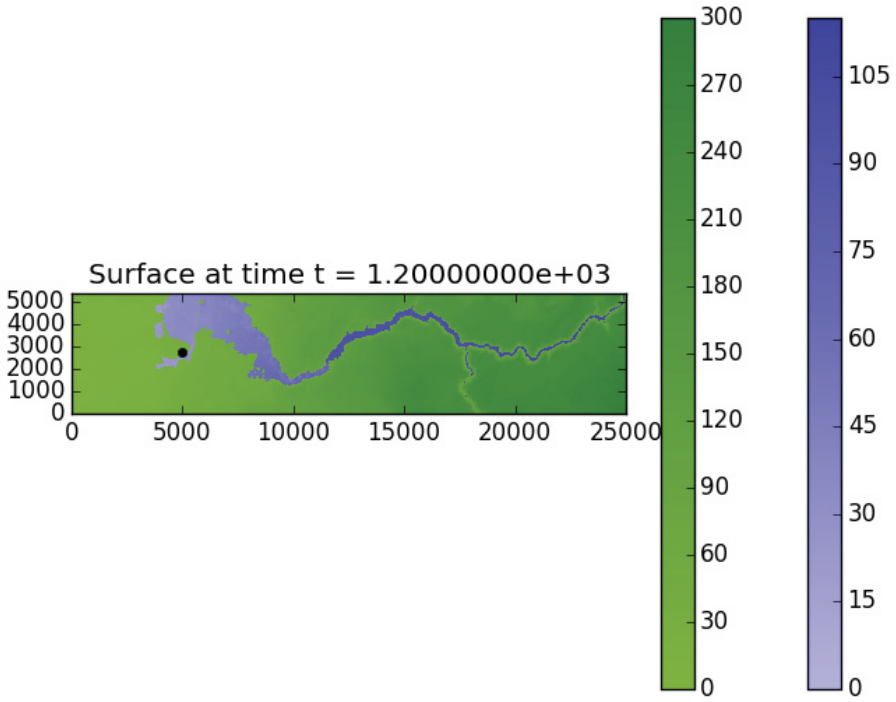


Figure 9. GeoClaw dam break simulation at time = 1.2e3

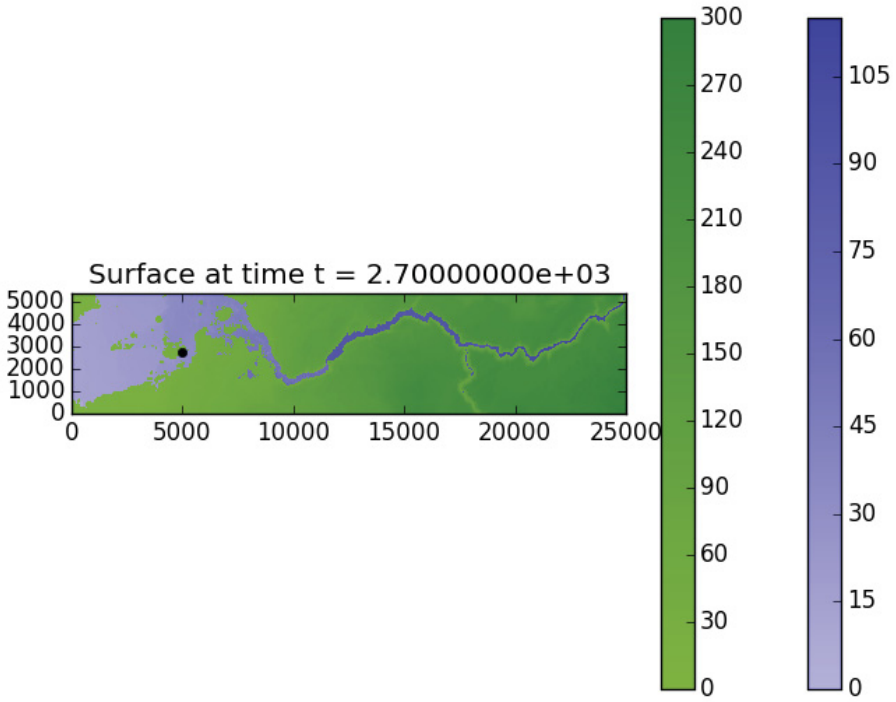


Figure 10. GeoClaw dam brea simulation at time = 2.7e3

3. PARTICLE-BASED APPROACHES

Two approaches are generally used for solving the Navier Stokes equations in Computational Fluid Dynamics (CFD). Grid based eulerian approach and particle-based lagrangian approach. Grid based Eulerian approaches are more common due to the fact that the computational domain becomes more manageable. However, Smoothed Particle Hydrodynamics (SPH) is a powerful lagrangian approach for simulating of fluid flows. This does not require the fluid to be placed in a grid and uses the integral formulation of the governing equations to solve and provides needed methods for fluid and rigid body interactions and measurements. However this approach requires a significant number of particles in the compact support domain therefore requiring large particle counts. Therefore an issue can arise in simulations where the area to be covered by the particles becomes very large, sometimes resulting in a sparseness of particles in the area of interest. In this section, we will investigate the use of SPH codes to evaluate the details of the flooding scenarios and the simulation results.

3.1 DualSPHysics

The selected SPH code used for simulating this flooding scenario was DualSPHysics. DualSPHysics is an open source, SPH code based on the SPHysics model [8]. It was developed by the University of Vigo, University of Manchester, University of Parma, Science & Technology Facilities Council, and CEHIDRO, Instituto Superior Tecnico. The code is implemented in C++ and CUDA and has the ability to run using Central Processing Units (CPUs) or Graphics Processing Units (GPUs).

DualSPHysics provides documentation as well as a support team that will answer questions as needed. Additionally, it provides multiple examples that allow a first time user to become familiar with the code. The provided examples include floating objects, solid objects, and importing in 3D geometries, such as SolidWork files. Lastly, the supporting documentation includes results of validation studies performed on DualSPHysics.

3.1.1 Running the Code

To run DualSPHysics, the computer operating system must either be Windows or Linux based. To begin, DualSPHysics requires an XML input file. The XML input file can be created two ways. Manually or by using the DualSPHysics Pre-processing Interface (DPI). The DPI is a GUI that allows the user to create their model. The DPI will then create the XML file based on what the user entered in the GUI. Figure 11 through Figure 13 show the different DPI GUI tabs. .

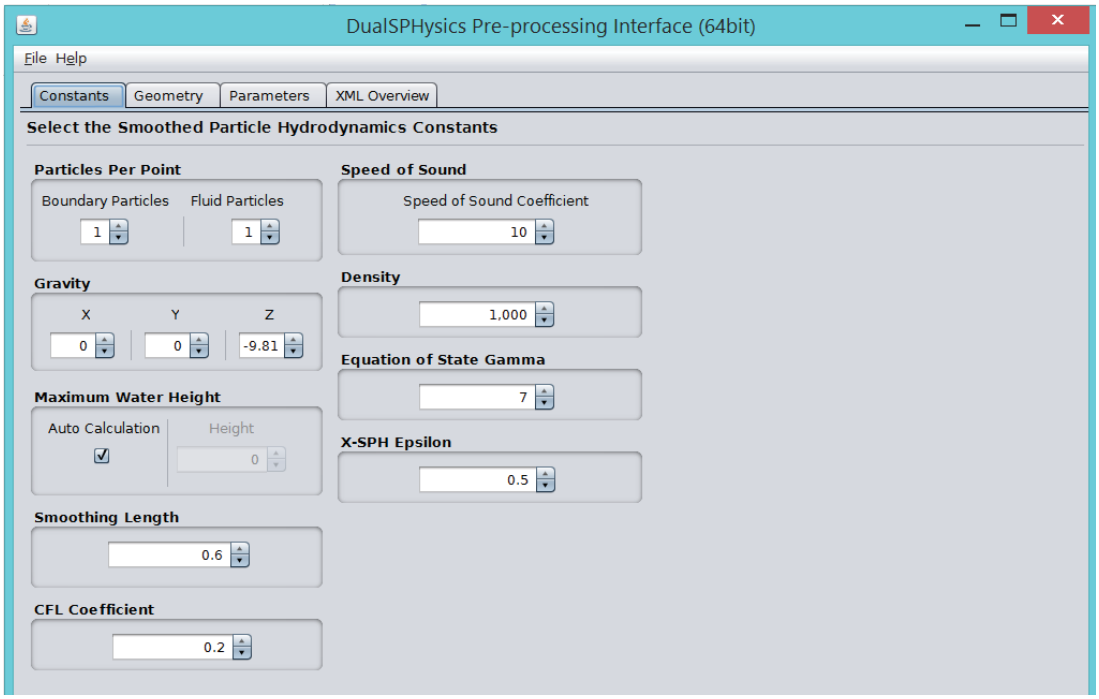


Figure 11. DualSPHysics DPI constants tab.

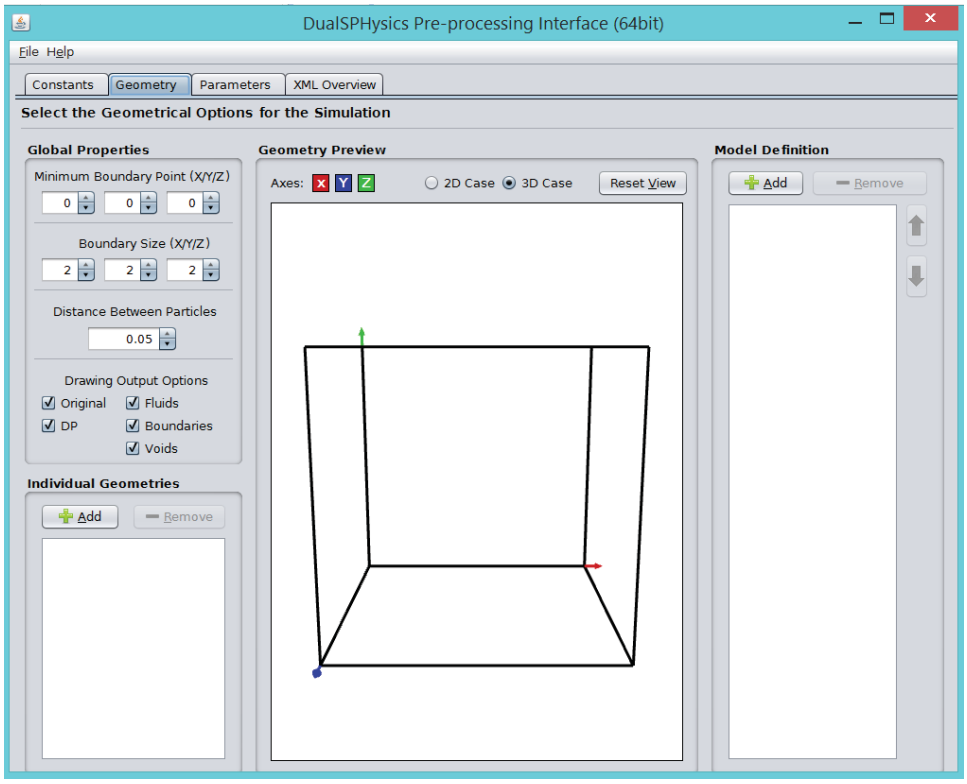


Figure 12. DualSPHysics DPI Geometry tab

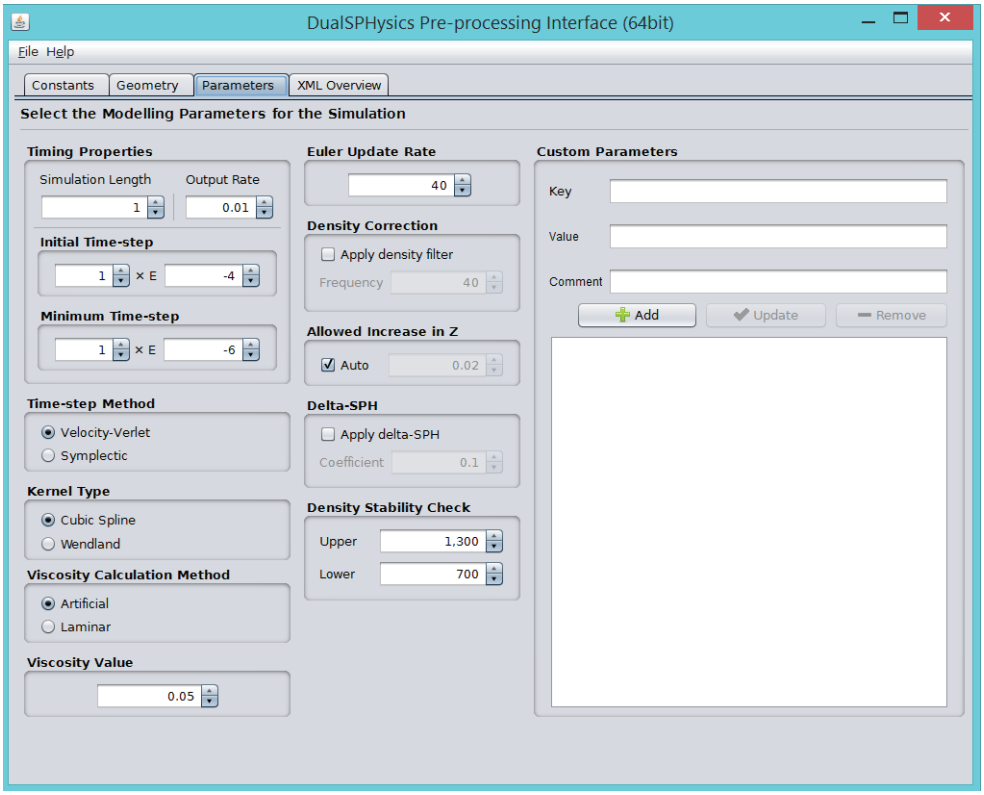


Figure 13. DualSPHysics DPI parameters tab

Once the user has created the input XML file, a pre-processing program must be executed. The GenCase executable creates the required input to run the DualSPHysics code. The DualSPHysics executable then runs the actual SPH code. Following the SPH calculations, a series of post-processing steps are available. The post-processing executables are as follows:

- PartVTK.exe – Used to visualize particle data output
- MeasureTool.exe – Used for comparing experimental and numerical values
- BoundaryVTK.exe – Allows for the boundary shapes formed by the boundary particles to be visualized
- IsoSurface.exe – Allows for the simulation to be represented by the surface rather than individual particles

All of the executables have different options associated with them.

3.1.2 Code Output

The DualSPHysics output will depend on what executables the user runs as well as what options the user specifies. The following is a brief outline of what each executable will output:

- GenCase.exe
  - XML file (input for DualSPHysics.exe)
  - Binary file (input for DualSPHysics.exe)
  - VTK files of geometry (used to visualize the initial setup)
- DualSPHysics.exe
  - Binary files (includes particle information at different times during the simulation)
  - Run.out file (brief summary of the simulation)
  - Optional output files with particle information (CSV, VTK, ASCII)

- PartVTK.exe
    - VTK files (used for plotting in ParaView)
- MeasureTool.exe
    - VTK, CSV, or ASCII (includes the desired information at the specified point)
- BoundaryVTK.exe
    - VTK, STL, or PLY (includes loaded information and boundary information)
- IsoSurface.exe
    - VTK (used for plotting in ParaView)

After the user has run the code, another code is provided by DualSPHysics to extract more information from the binary files that are produced by the DualSPHysics executable. The provided code is called ToVTK. When ToVTK is run, it will extract information from the binary files, such as position, velocity, pressure, density, etc…, and then create CSV files that the user can then open in Excel. ParaView was used in order to visualize the simulation, it is an open source data analysis and visualization application [9]. Since DualSPHysics is the SPH code that was used, the .VTK files outputted from the GenCase.exe, PartVTK.exe, and IsoSurface.exe were opened in ParaView in order to visualize the flooding simulation.

3.2 Flooding Model and Scenarios

A topography file was created so that a flooding scenario could be simulated using an actual location. For this example, we chose to use a portion of the INL DOE site as a test case, however any location on the Earth could have been chosen. Figure 14 shows an aerial view of the topography where flooding was simulated – note the topography of this site was chosen to represent the possibility of river-type of flow to enter into the area used for the simulation. The goal of this simulation was to see how an SPH-based approach would work when particles are distributed over a very large area.

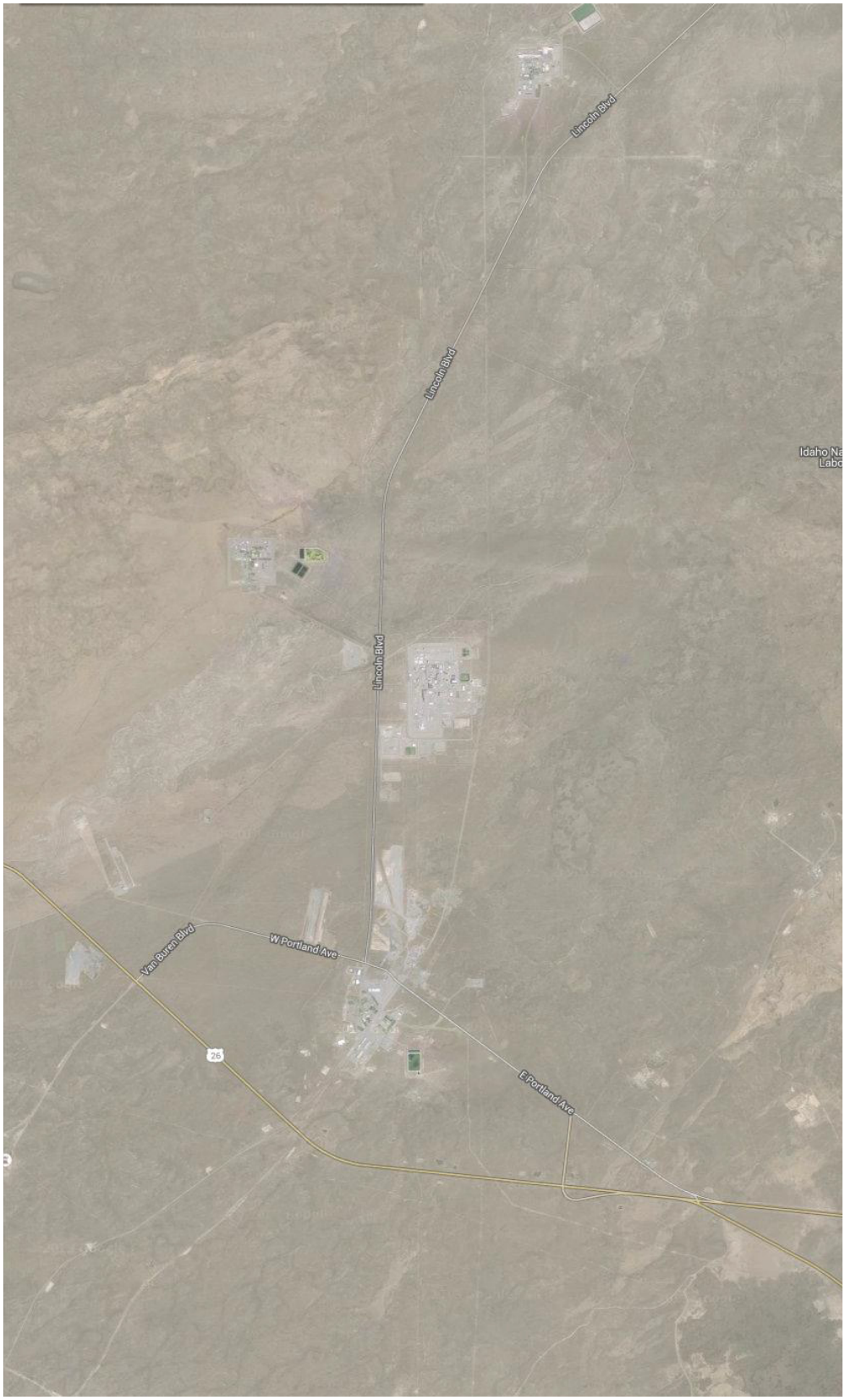


Figure 14. Flooding scenario topography

3.2.1 Creating the Simulated Flood

Since riverine flooding was the flood type of interest, a model needed to be created so that water would flow like a riverine flood into the area of interest. For example, it may be possible that a dam up-stream of the simulated area might be breached or a large rain-on-snowpack event might be experienced – for these cases, a large volume of water would enter the region. In order to accomplish the SPH simulation for the scenarios, several options were investigated. One proposed approach was to model a container that had an opening at the bottom that could be opened programmatically, providing a representative inundation of water. Water would then be filled in the container and would flow out of the opening like a riverine flood. Figure 15 shows the container which was created using SolidWorks.

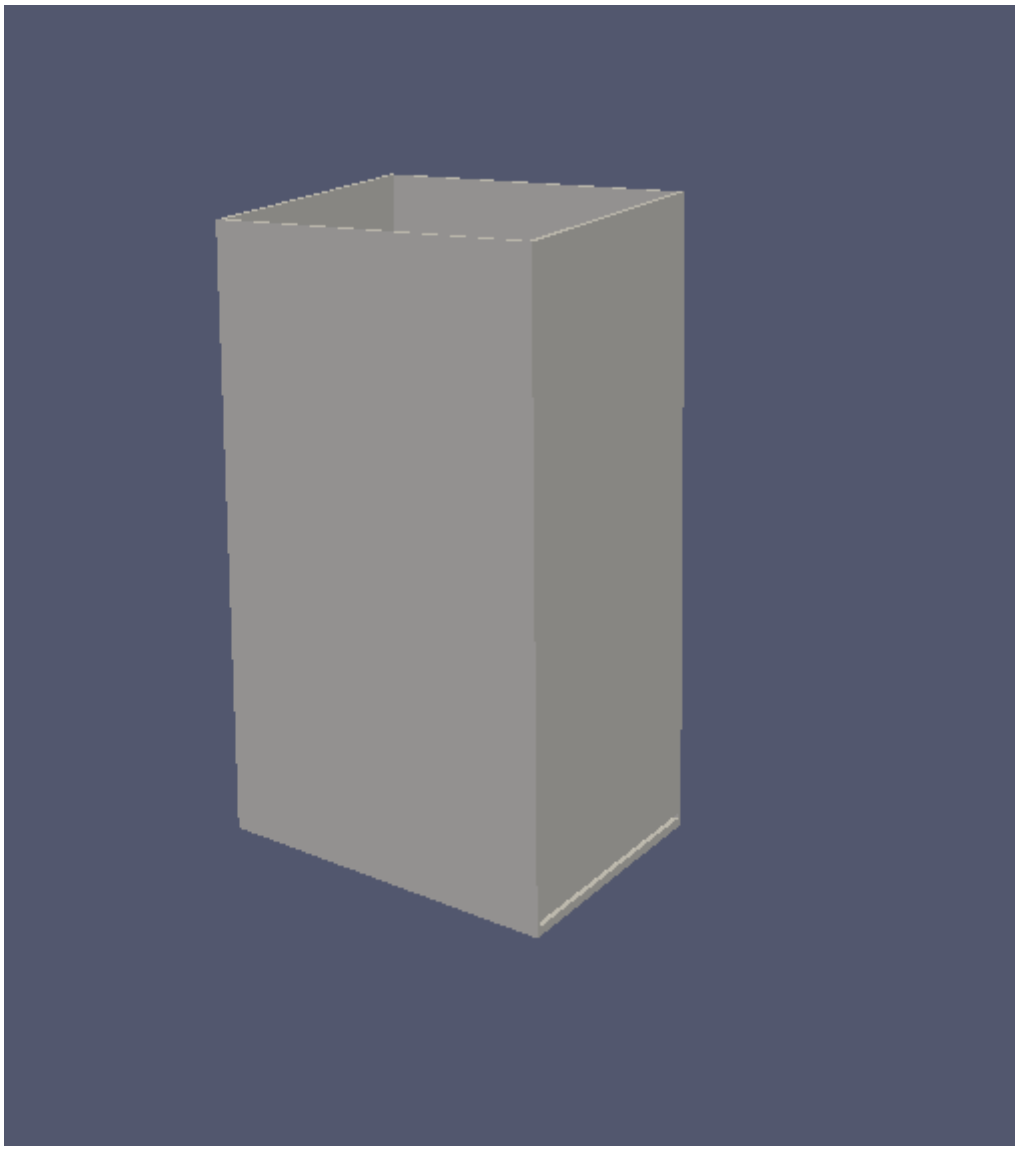


Figure 15. Water container option

The first container option did not produce desired results. The water would either come out of the opening too fast, 300mph, or would not come out of the opening at all. In order to get a lower velocity, the container was adjusted so that it was horizontal instead of vertical. Figure 16 shows the horizontal container which was created using SolidWorks.

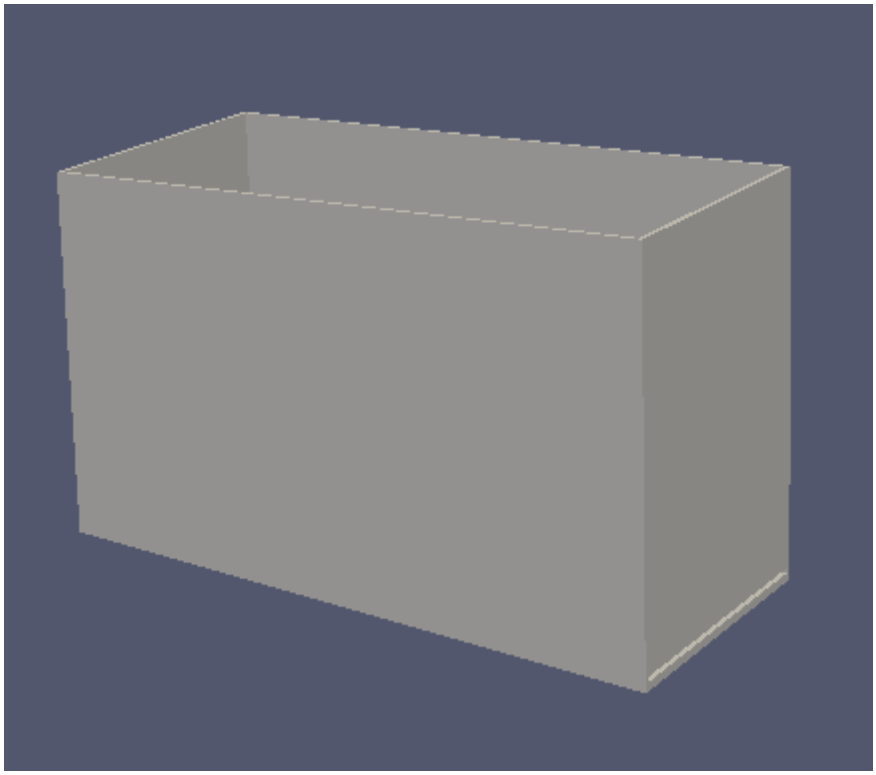


Figure 16. Horizontal water container option

The results from the horizontal container were similar to the vertical container which was again undesirable. The next option was to model a river similar to a slab of water that would start away from the topography and then flow onto the land like a riverine flood would in reality. Figure 17 shows the water slab option which was created using the DualSPHysics Pre-Processing Interface. The red exterior is an open channel so that the water has a boundary to stay within before it reaches the topography area.

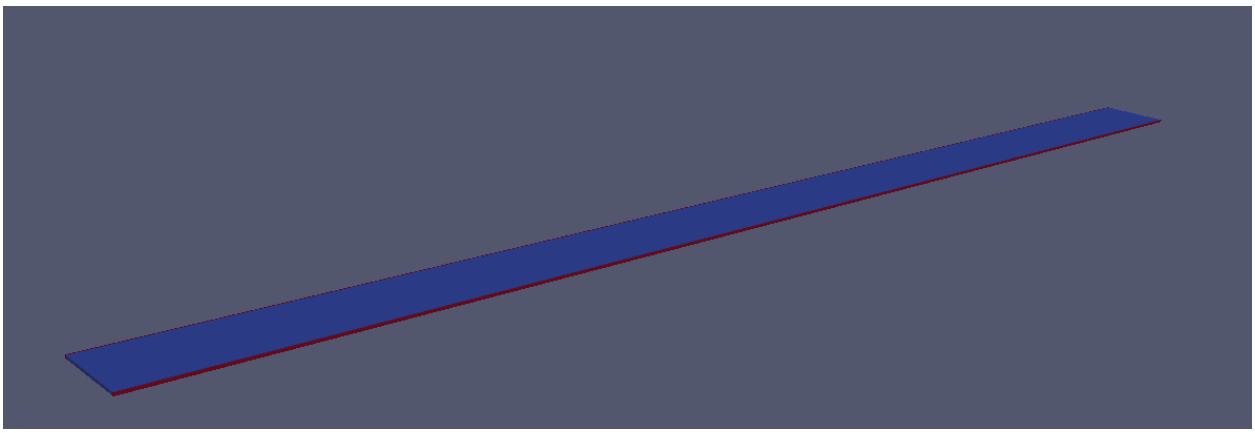


Figure 17. Water slab option

The results from the water slab provided results that were much better and seemed similar to a riverine flood. Therefore, the water slab option was used as the method for creating the simulated flooding scenarios.

3.2.2 Flooding Model

The flood was created using the "water slab" technique. Since the water must move like a riverine flood, it was given an initial velocity of 30 m/s, which is comparable to a flash flood. The dimensions of the water storage container were:

- Height. 100 m
- Width. 2,900 m
- Length. 25,100 m

The dimensions of the topography file were:

- Elevation (maximum). 78.2 m (zero represents the lowest point in the area)
- Width. 11,560 m
- Length. 20,080 m

Since flooding in the simulation area was the only concern, water that traveled off of the topography was removed from the simulation.

Once the model is created, the open channel and water needed to be placed with respect to the topography. The water and open channel was placed 100m past the edge of the topography so that the water would make it to the topography and not be eliminated from the simulation. The water and open channel was also placed 121.8 m above the highest point of the topography, which is 78.2m, so that particles would be guaranteed to flow on top of the topography (in case a high point was near an edge).

The DualSPHysics code does not specify the size of the particle, but rather the distance between particles. To determine the particles size, the density and mass of each particle can be used. However in DualSPHysics, the density of the particle changes as the particle moves throughout the simulation. The changing density then results in the particle size changing since the total water mass is constant.

After the water properties, open channel information, and topography file was imported into DualSPHysics, the simulation was ran using a distance of 10m between particles. Each simulation was run for 750 seconds at an output rate of 0.5 seconds so that the flow of the water could be tracked flowing through the simulated area.

3.2.3 Flooding Scenarios

Four scenarios were simulated for flooding, each one representing a different location for entry of water into the simulated area. Figure 18 shows the four scenarios that were simulated. A flooding event was simulated at each of the four corners, called Case 1 through Case 4. This was done to see how water will move throughout the topography when water is coming from different locations. The initial flooding model for each case was the same as described above except starting the water at different locations.

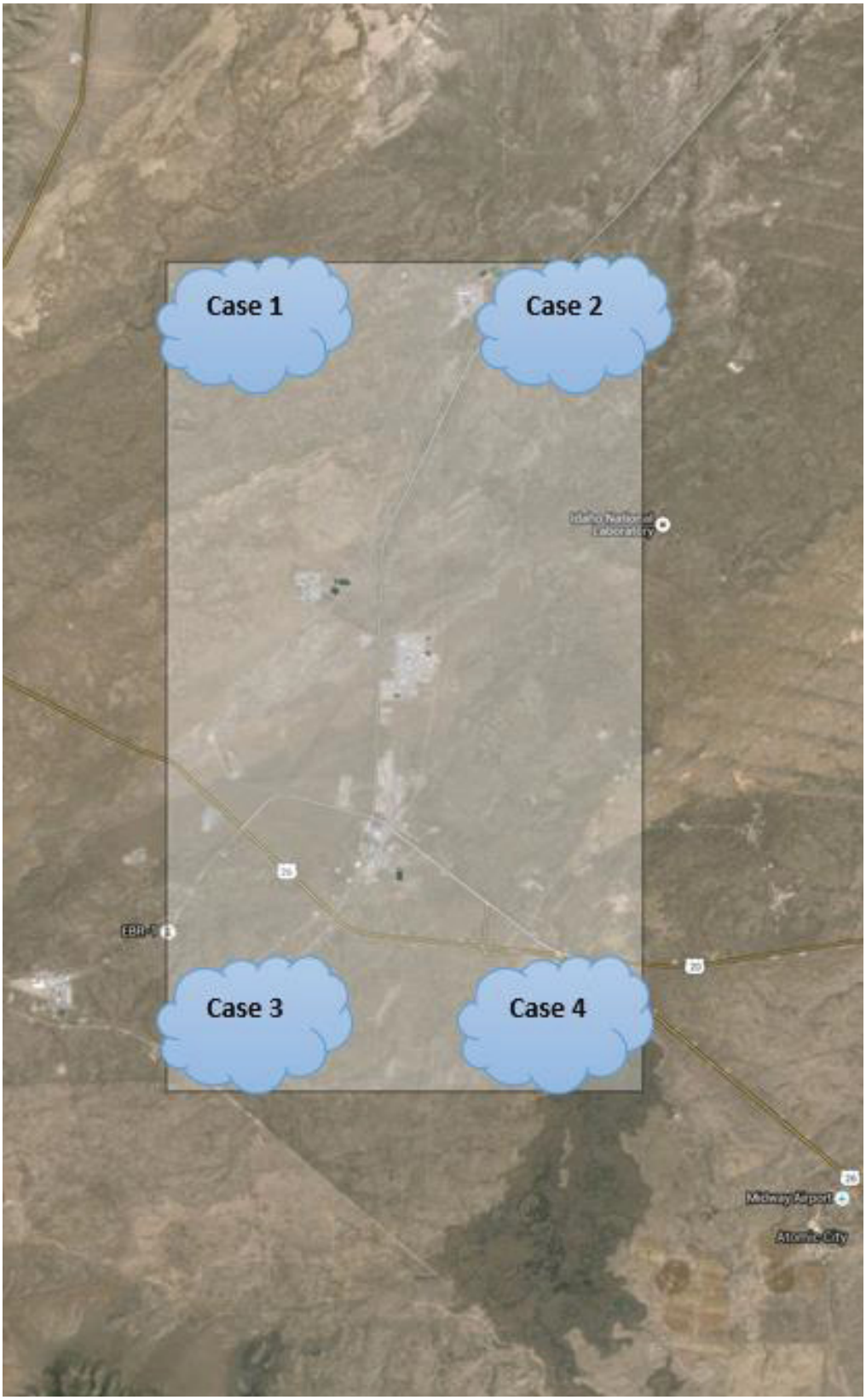


Figure 18. Flooding scenarios

3.3 Simulation Results

Because DualSPHysics was used, the flooding simulations were computed using GPUs (NVidia GeForce GTX Titans). The results of each case are provided in this subsection.

3.3.1 Case 1

The results for Case 1 were as follows.

- Distance between particles. 10m
- Simulated time. 750 seconds
- Computational runtime. 11.45 hours
- Time per second of simulation. 54.95 seconds
- Number of fluid particles. 7,304,100
- Total number of particles. 10,463,749
- Excluded number of particles. 5,827,051
- GPU memory usage. 7647.84 MB
- Output space usage. 659 GB

Figure 19 through Figure 29 show screen shots of the simulation at different points in time. The left side is the PartVTK output which shows the individual particles (numbered and then identified by color) and the right side is the IsoSurface output which shows the surface of where the water particles reside during the scenario.

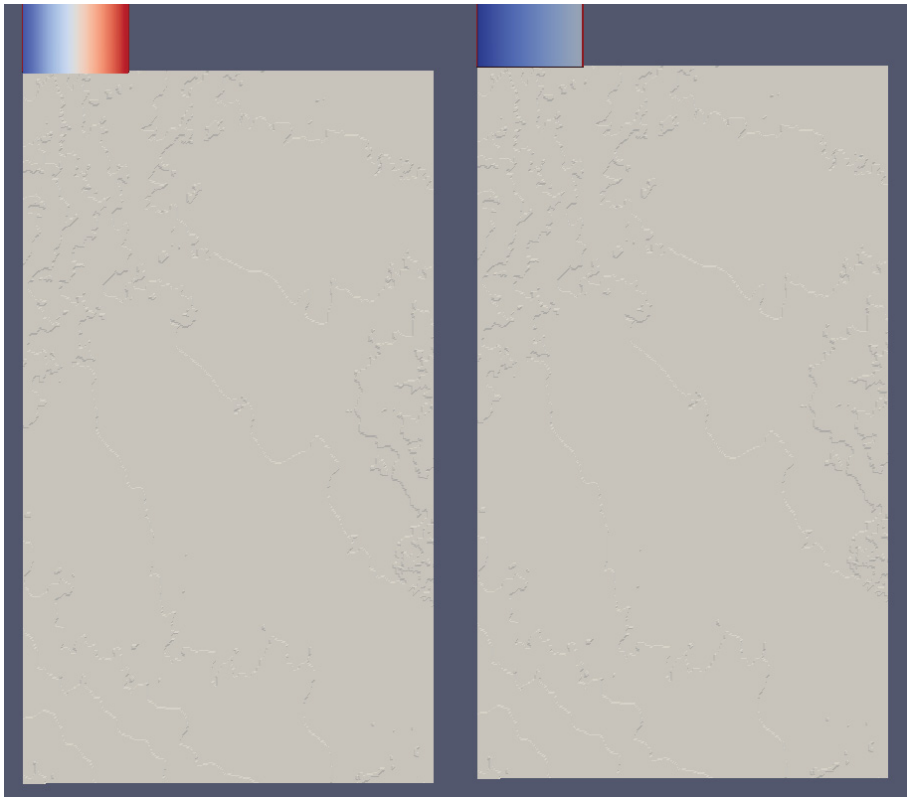


Figure 19. Case 1 beginning

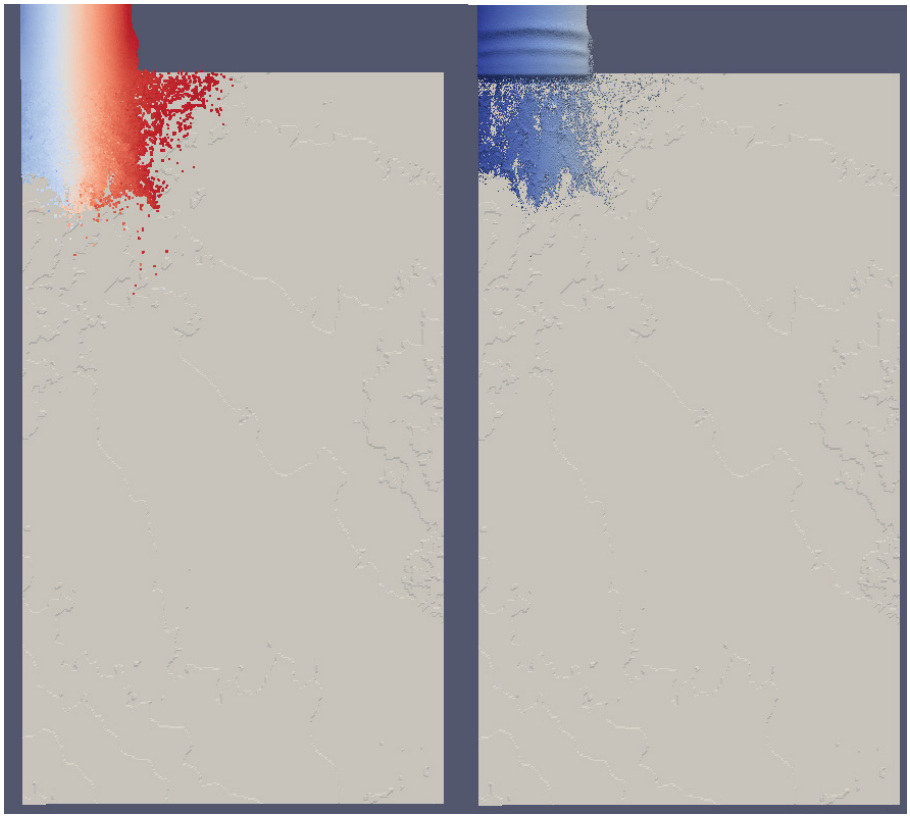


Figure 20. Case 1 75 seconds

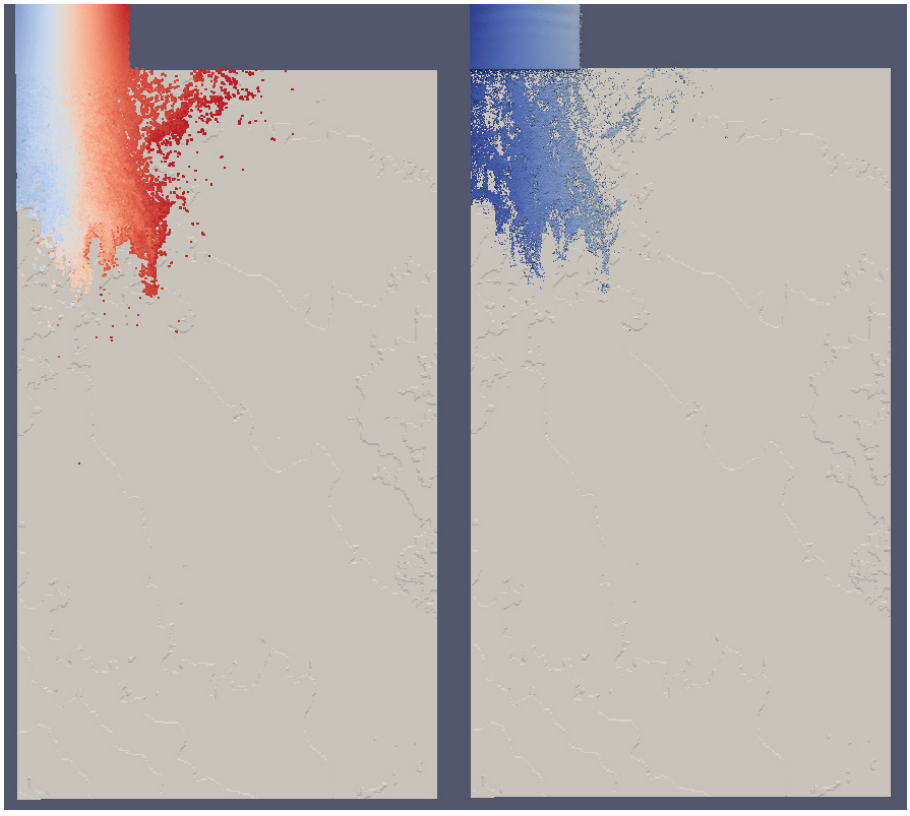


Figure 21. Case 1 150 seconds

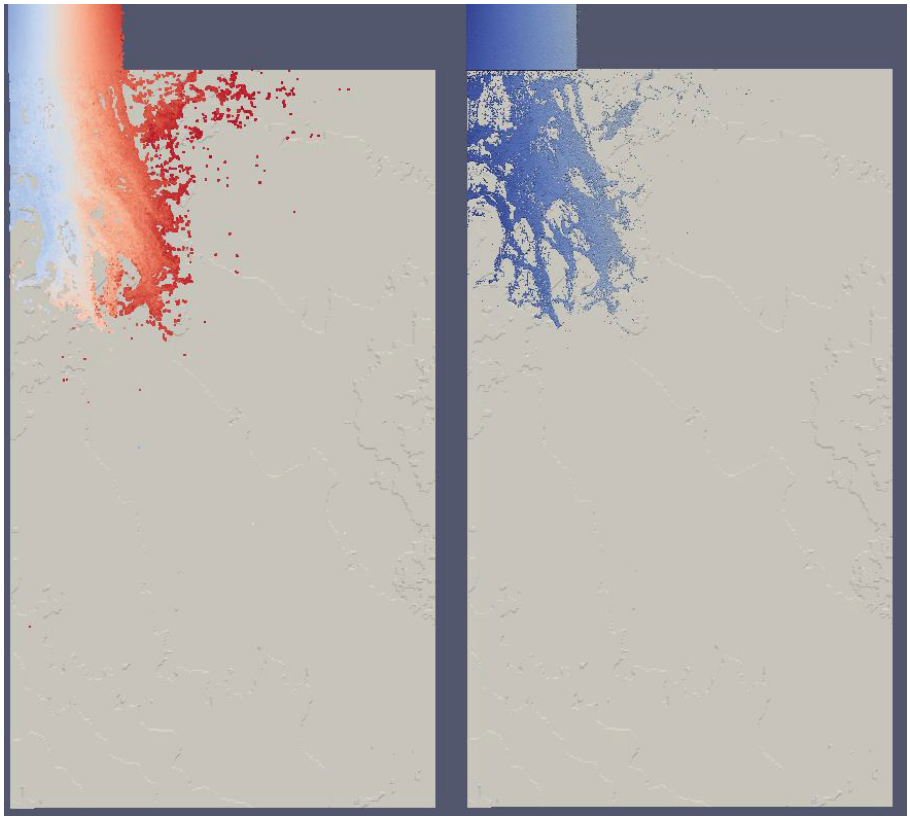


Figure 22. Case 1 225 seconds

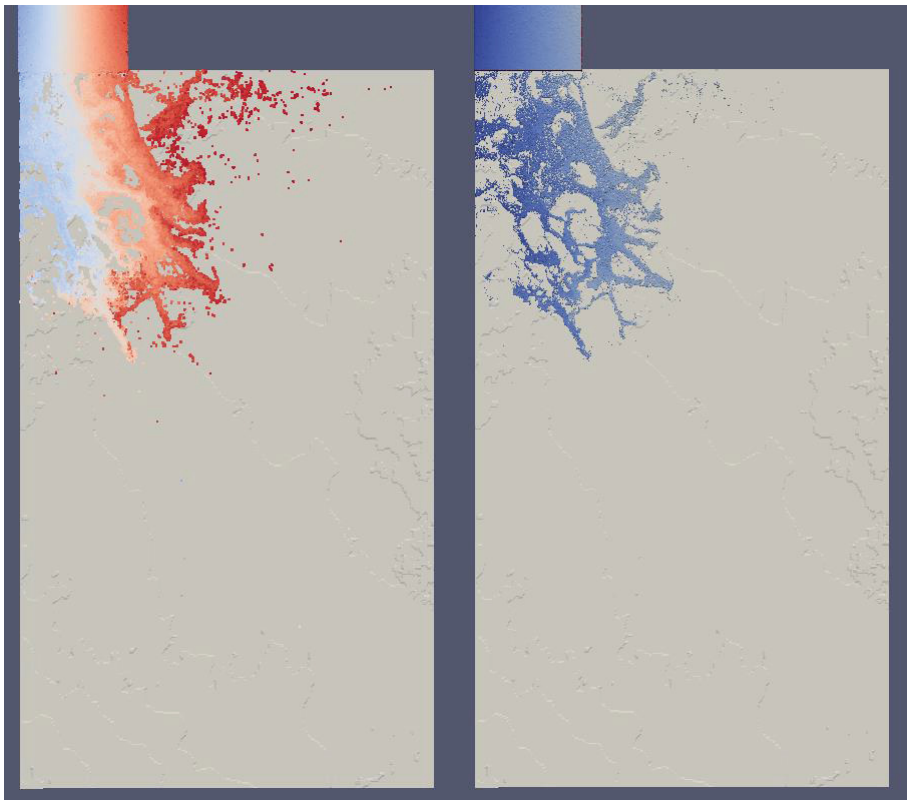


Figure 23. Case 1 300 seconds

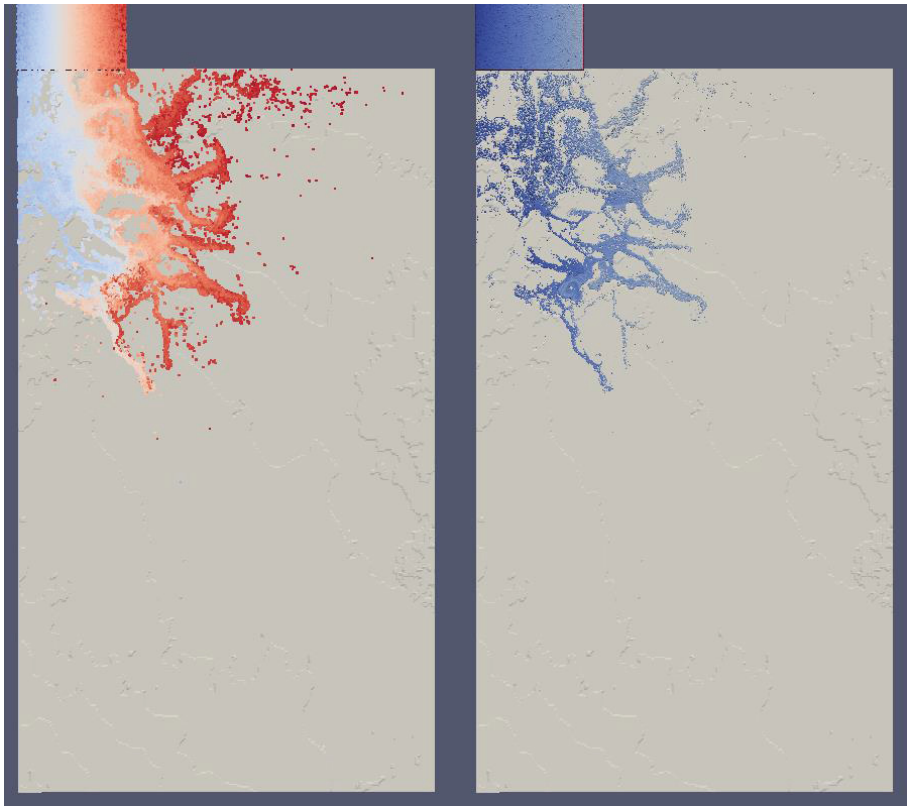


Figure 24. Case 1 375 seconds

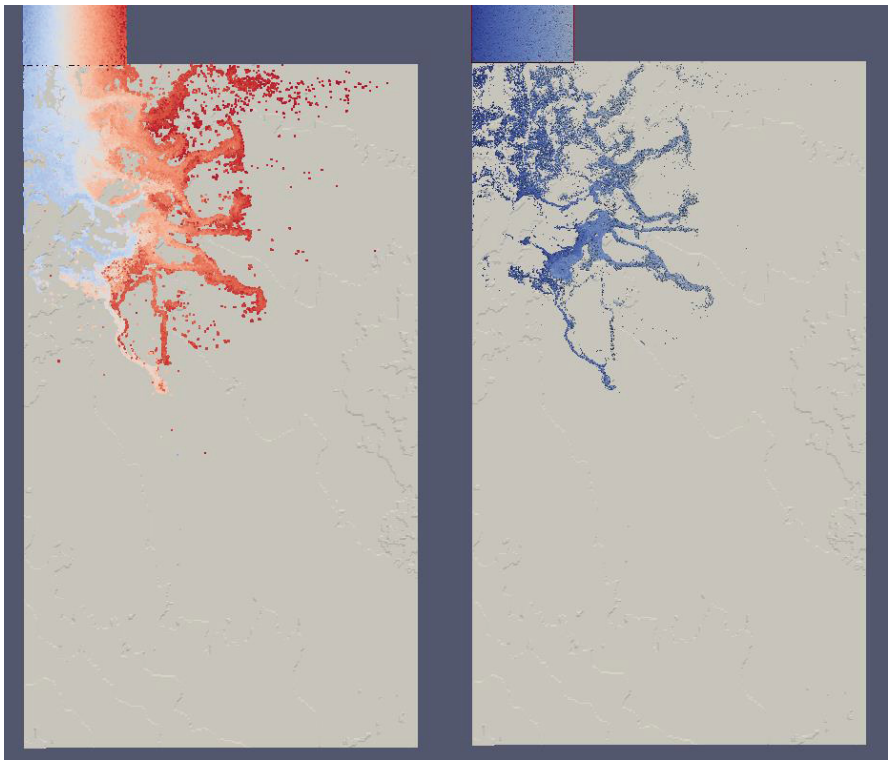


Figure 25. Case 1 450 seconds

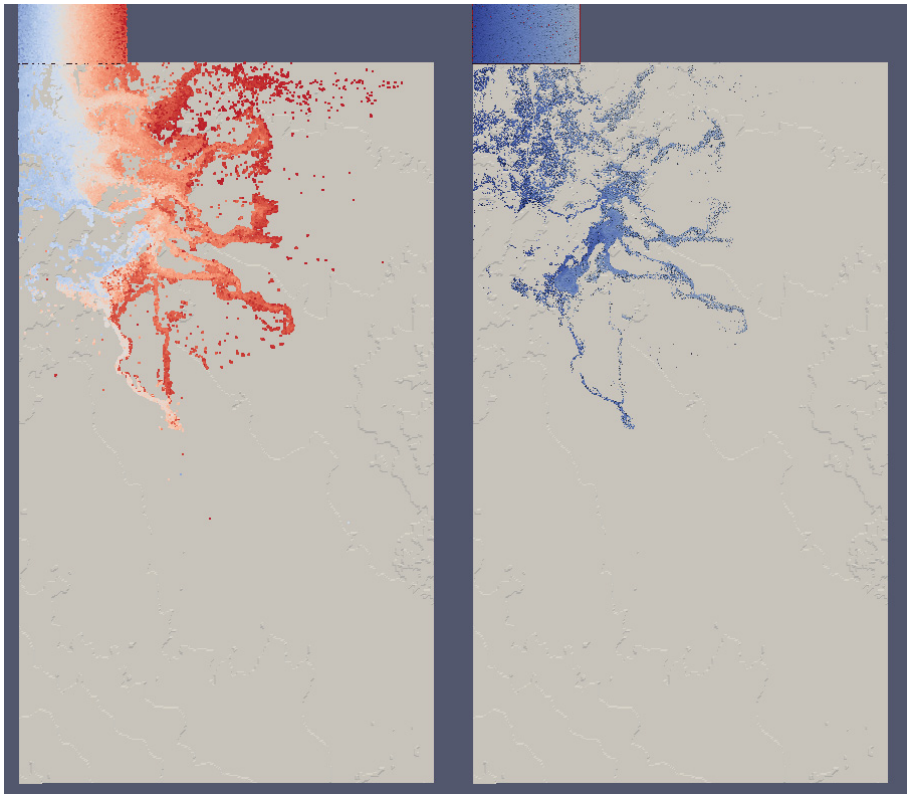


Figure 26. Case 1 525 seconds

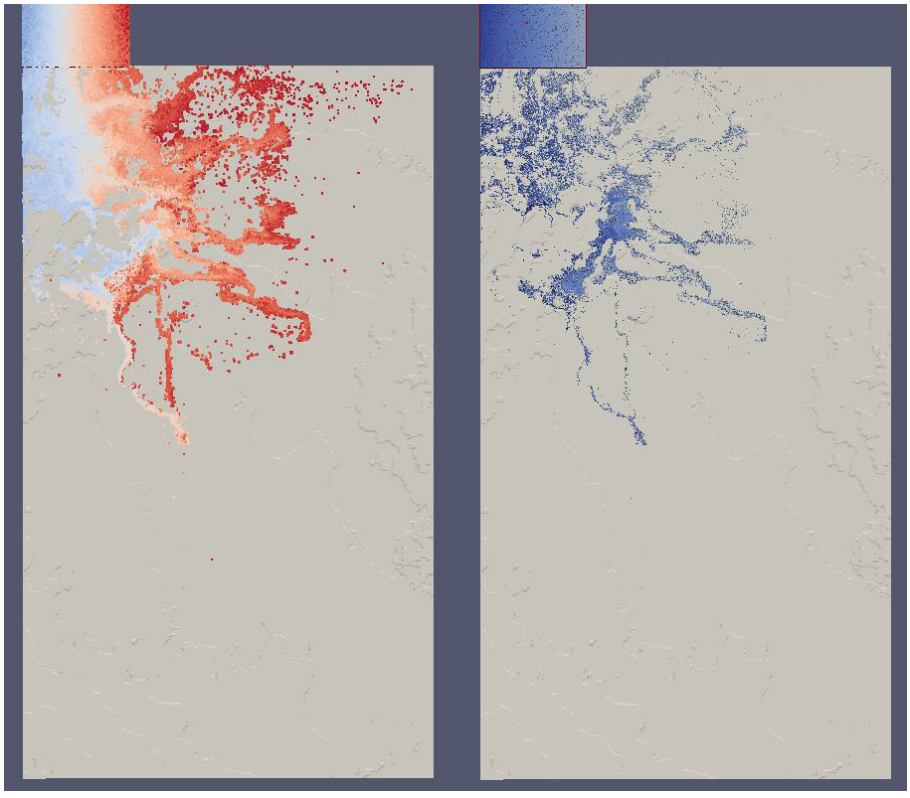


Figure 27. Case 1 600 seconds

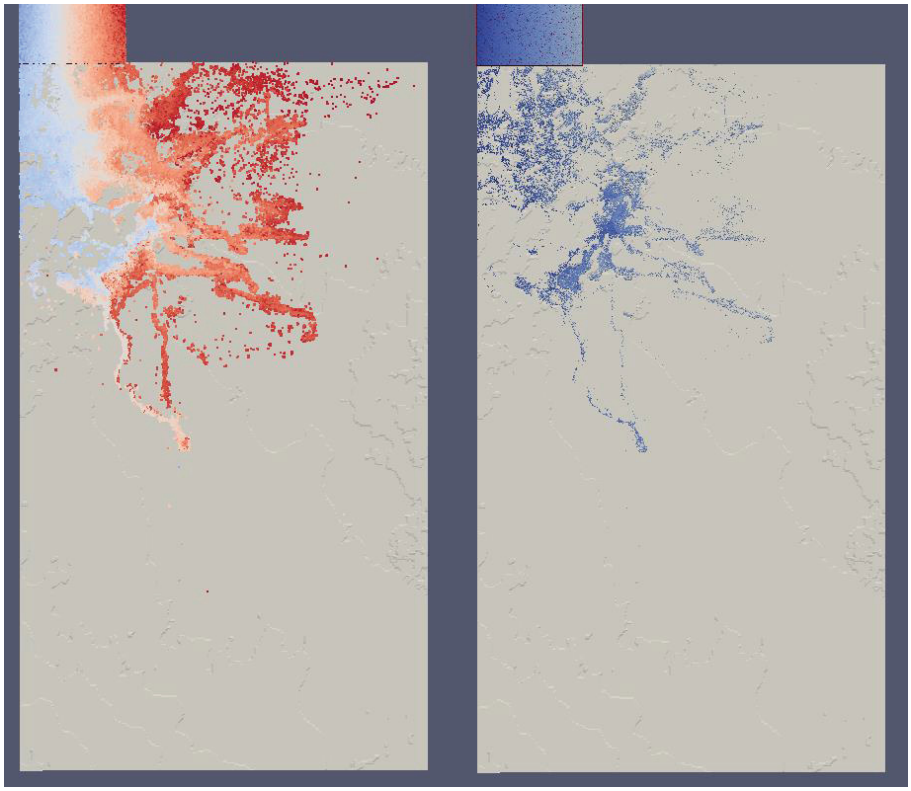


Figure 28. Case 1 675 seconds

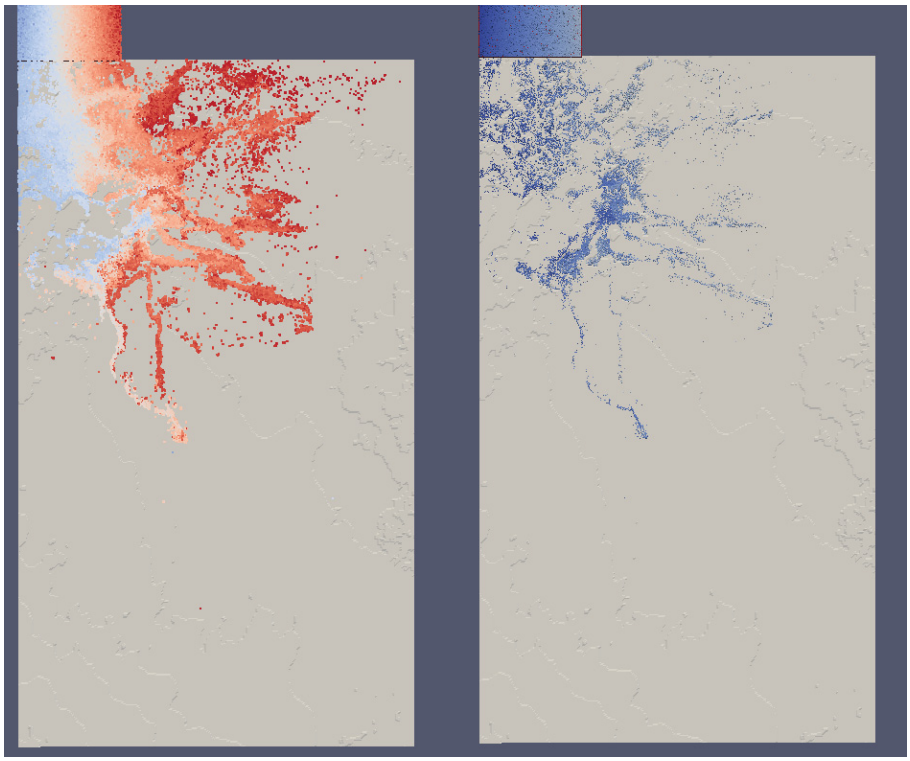


Figure 29. Case 1 750 seconds

3.3.2 Case 2

The results for Case 2 were as follows.

- Distance between particles. 10m
- Simulated time. 750 seconds
- Computational runtime. 11.19 hours
- Time per second of simulation. 53.69 seconds
- Number of fluid particles. 7,304,100
- Total number of particles. 10,463,749
- Excluded number of particles. 5,860,416
- GPU memory usage. 7647.84 MB
- Output space usage. 663 GB

Figure 30 through Figure 40 show screen shots of the simulation at different points in time. The left side is the PartVTK output which shows the individual particles and the right side is the IsoSurface output which shows the surface.

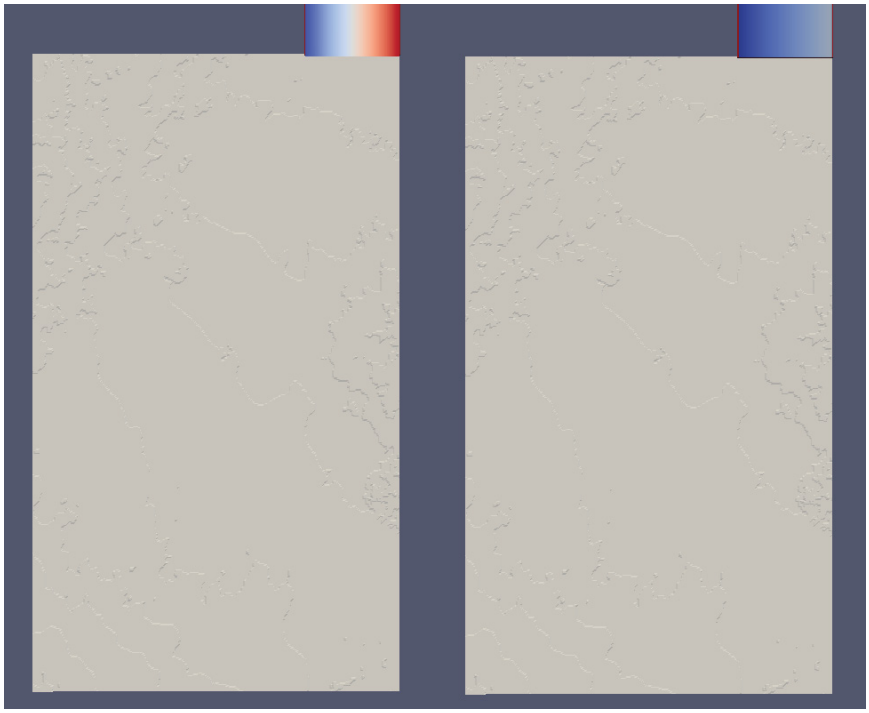


Figure 30. Case 2 beginning

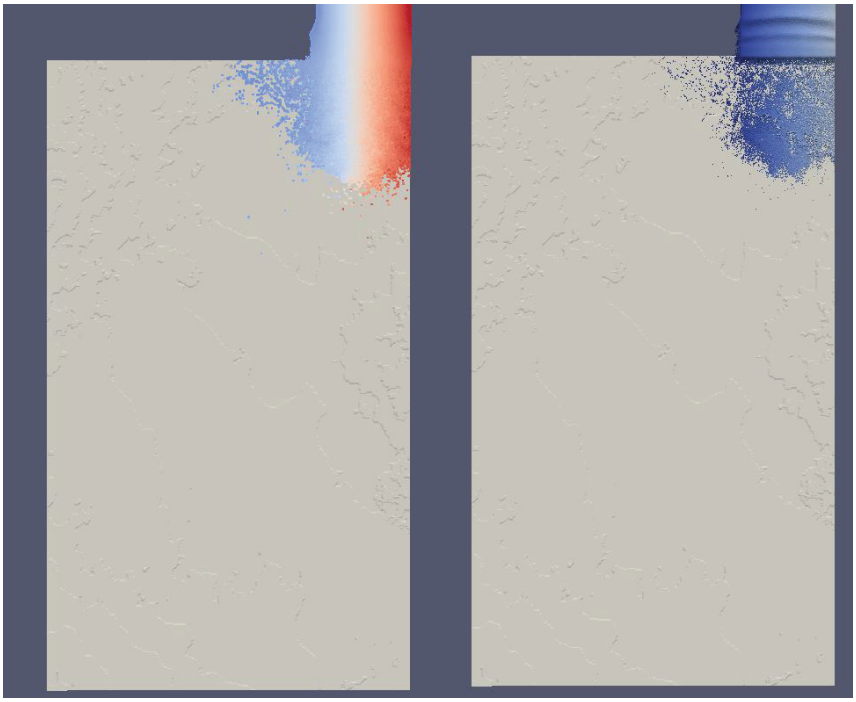


Figure 31. Case 2 75 seconds

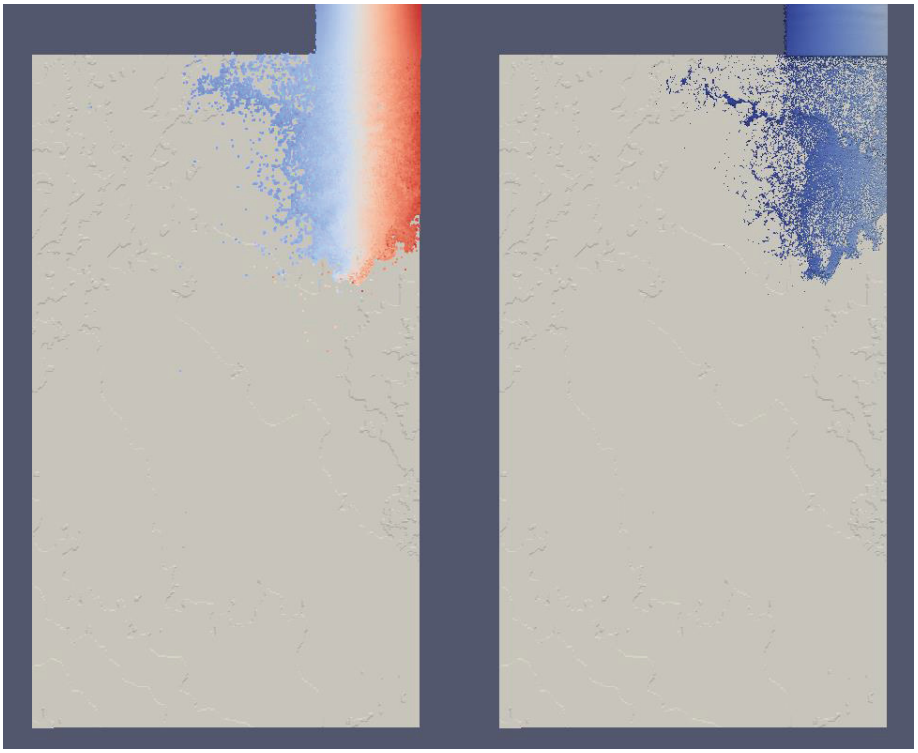


Figure 32. Case 2 150 seconds

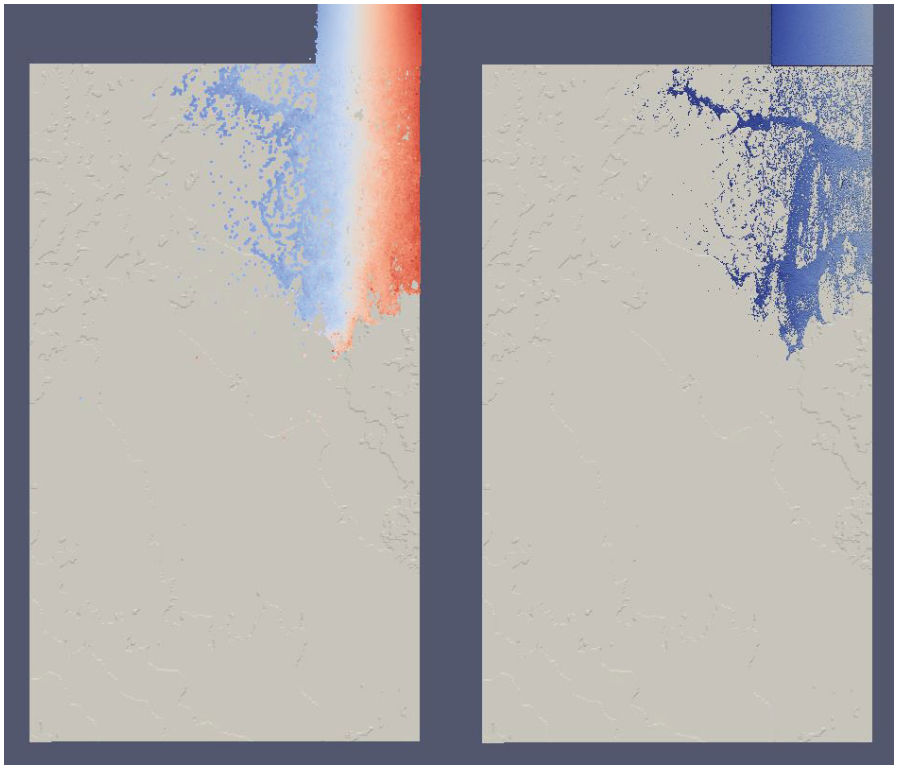


Figure 33. Case 2 225 seconds

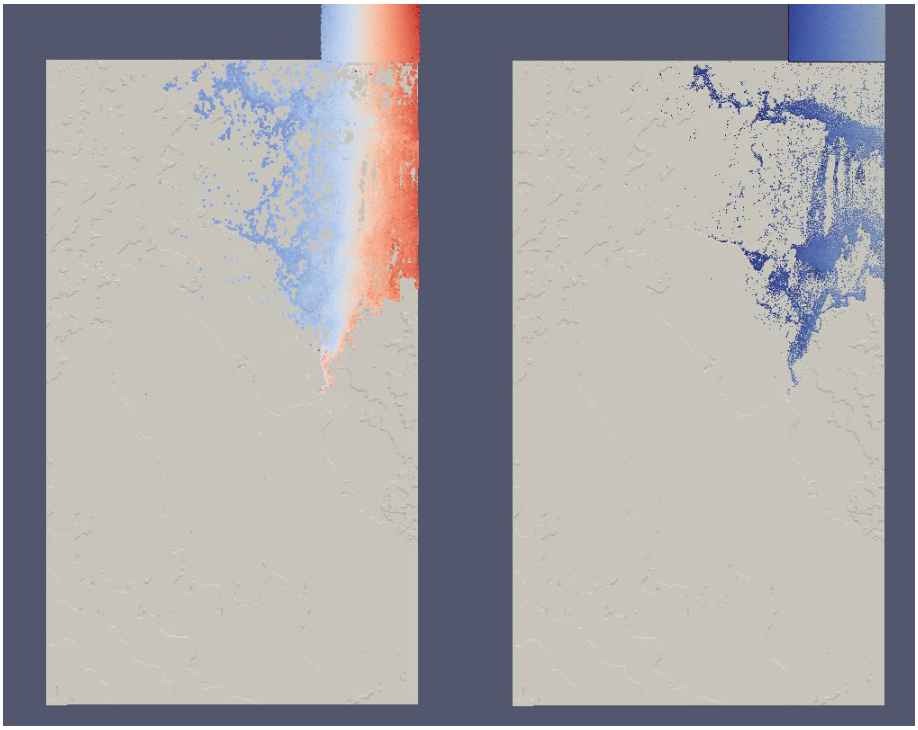


Figure 34. Case 2 300 seconds

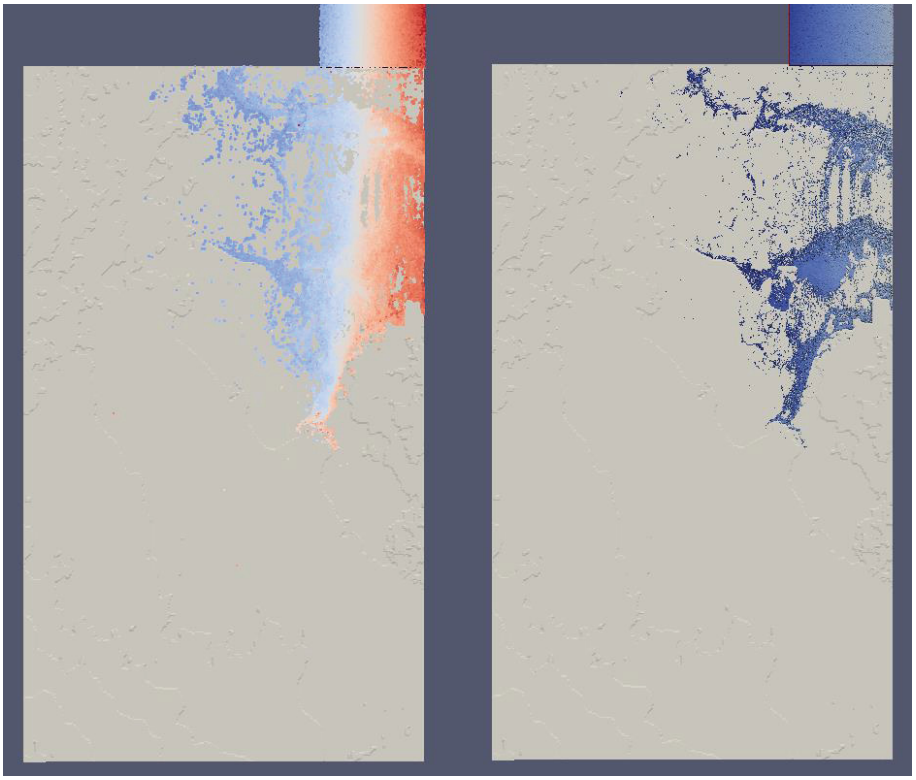


Figure 35. Case 2 375 seconds

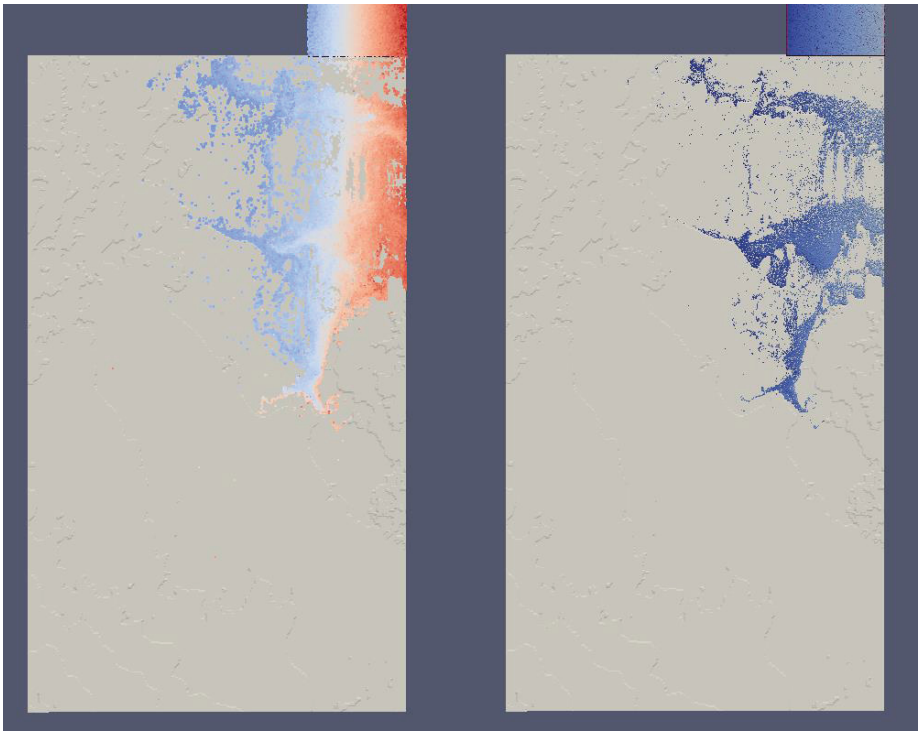


Figure 36. Case 2 450 seconds

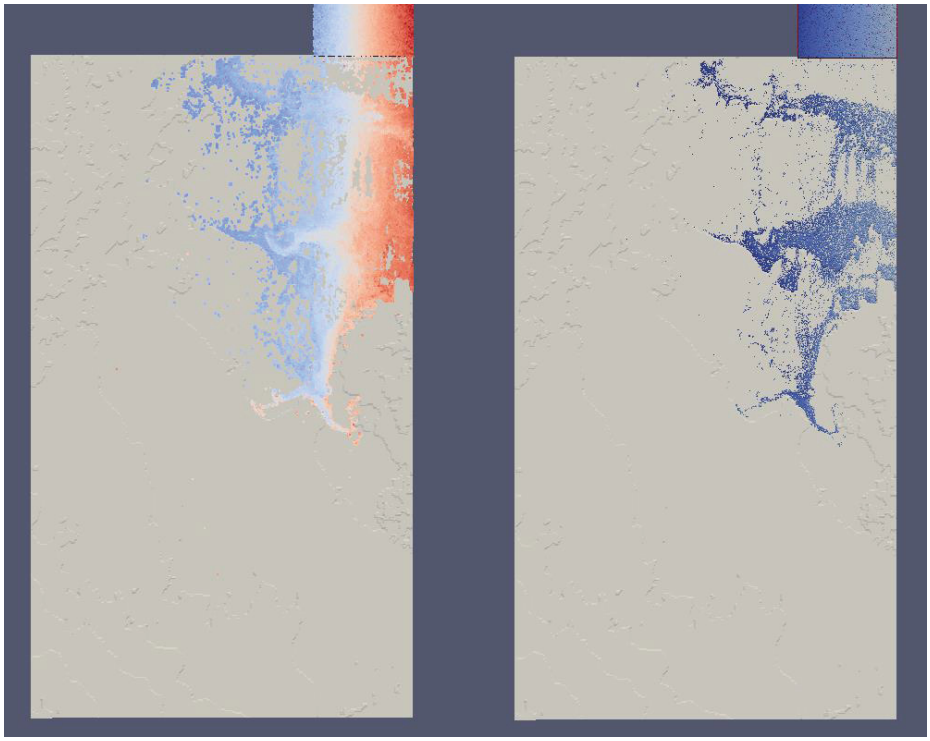


Figure 37. Case 2 525 seconds

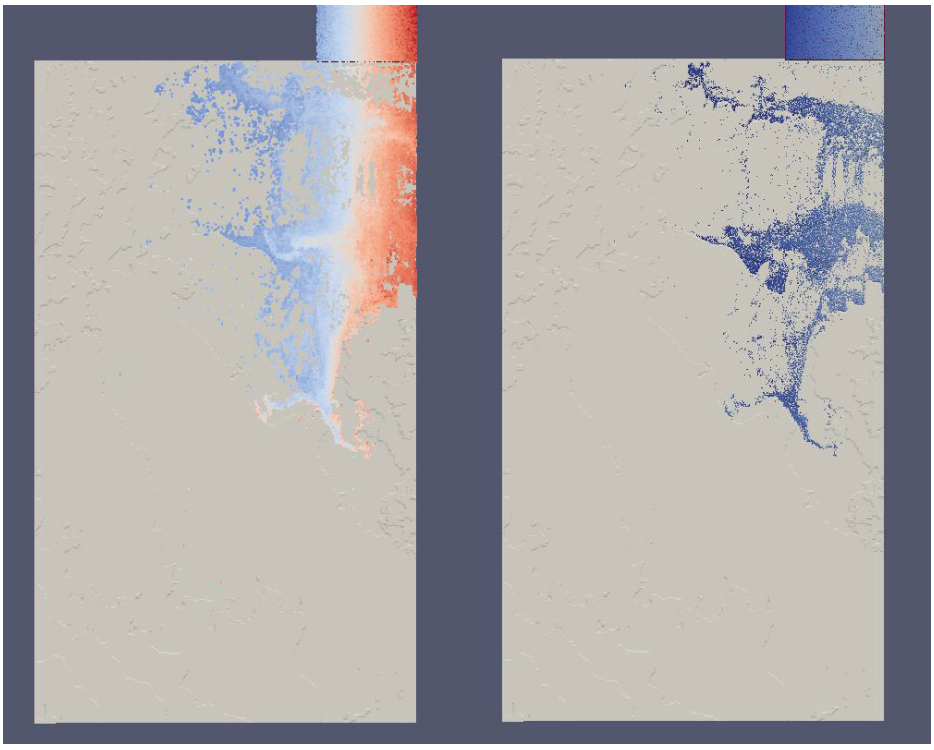


Figure 38. Case 2 600 seconds

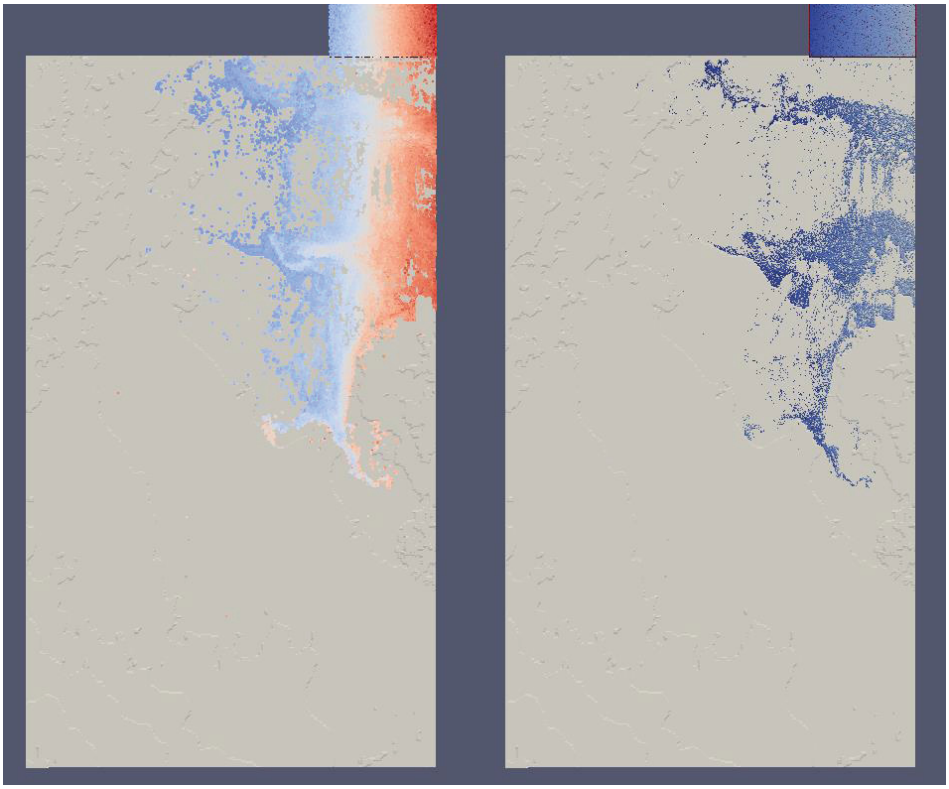


Figure 39. Case 2 675 seconds

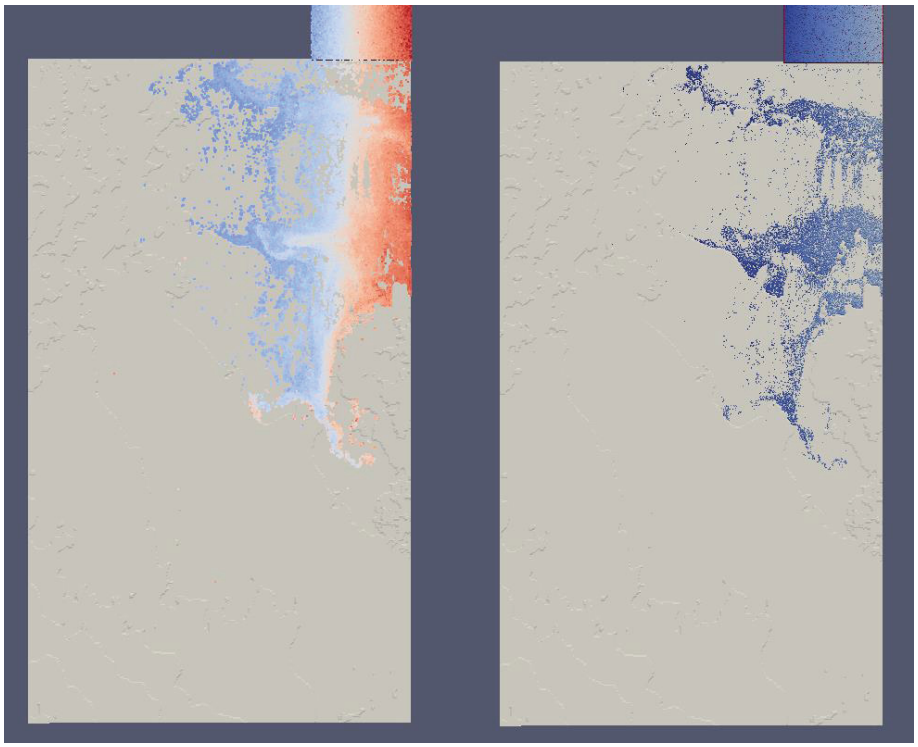


Figure 40. Case 2 750 seconds

3.3.3 Case 3

The results for case 3 were as follows.

- Distance between particles. 10m
- Simulated time. 750 seconds
- Computational runtime. 11.71 hours
- Time per second of simulation. 56.22 seconds
- Number of fluid particles. 7,304,100
- Total number of particles. 10,463,749
- Excluded number of particles. 4,836,849
- GPU memory usage. 7647.84 MB
- Output space usage. 827 GB

Figure 41 through Figure 51 show screen shots of the simulation at different points in time. The left side is the PartVTK output which shows the individual particles and the right side is the IsoSurface output which shows the surface.

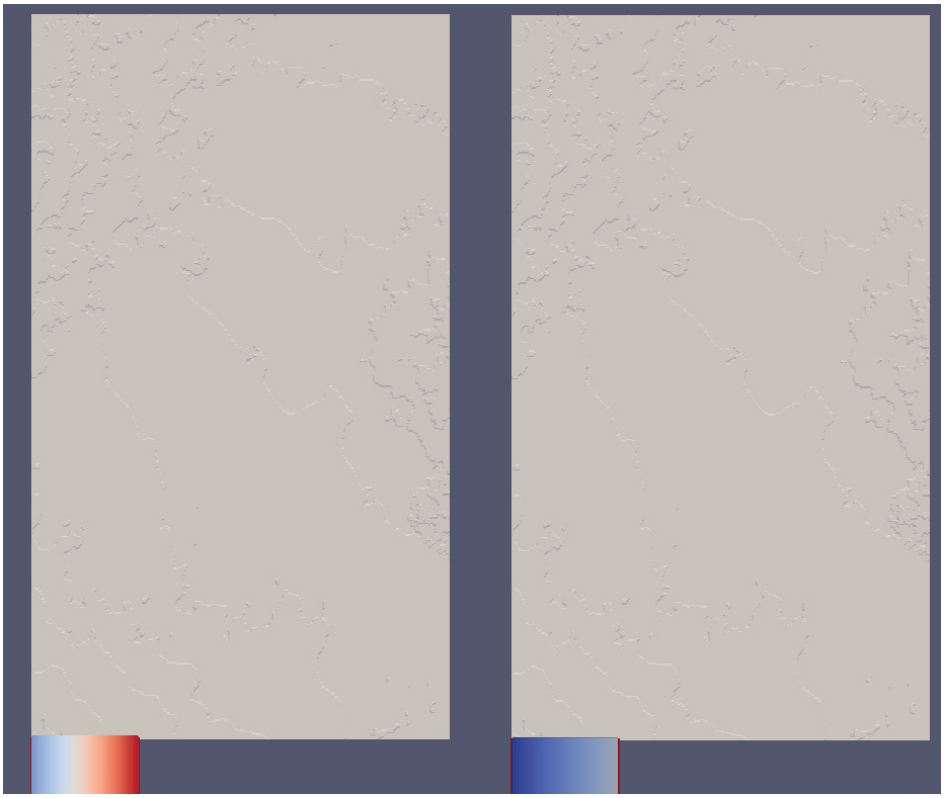


Figure 41. Case 3 beginning

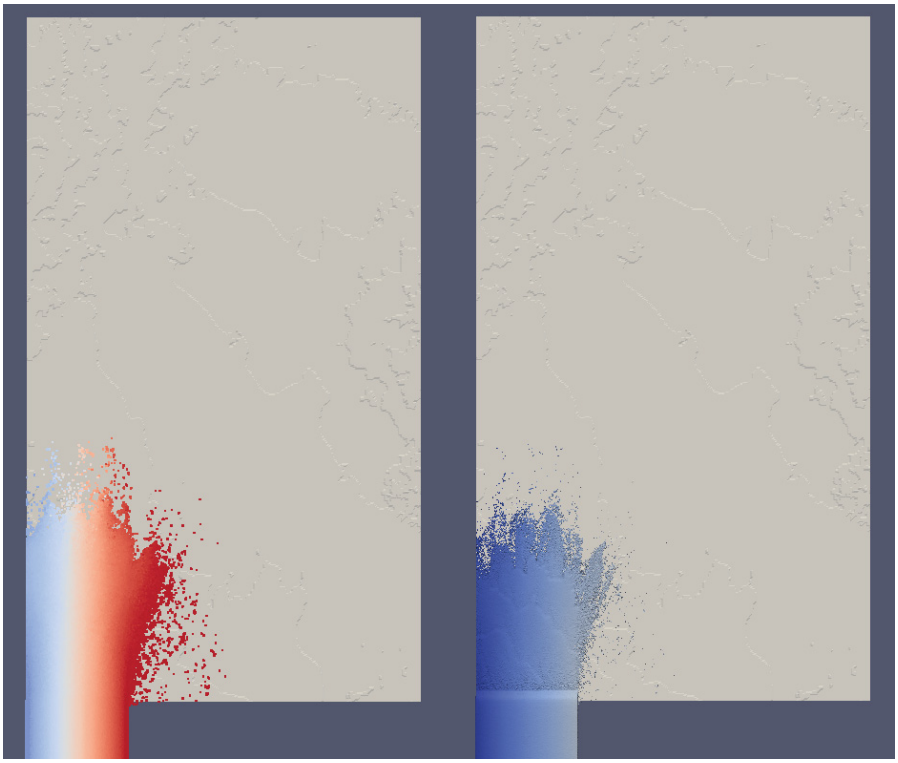


Figure 42. Case 3 75 seconds

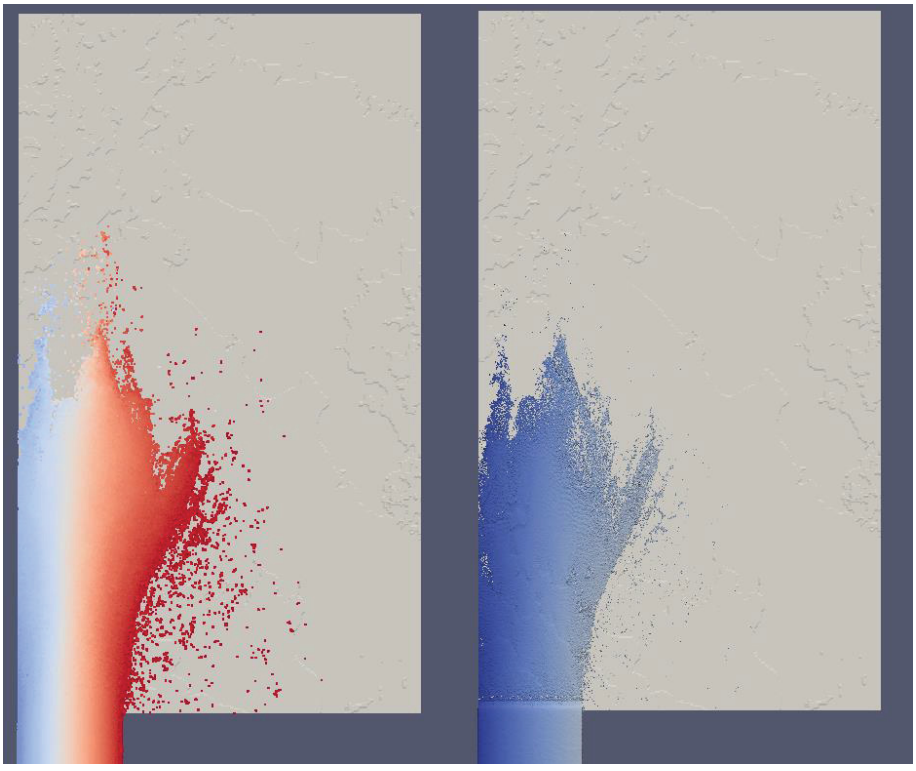


Figure 43. Case 3 150 seconds

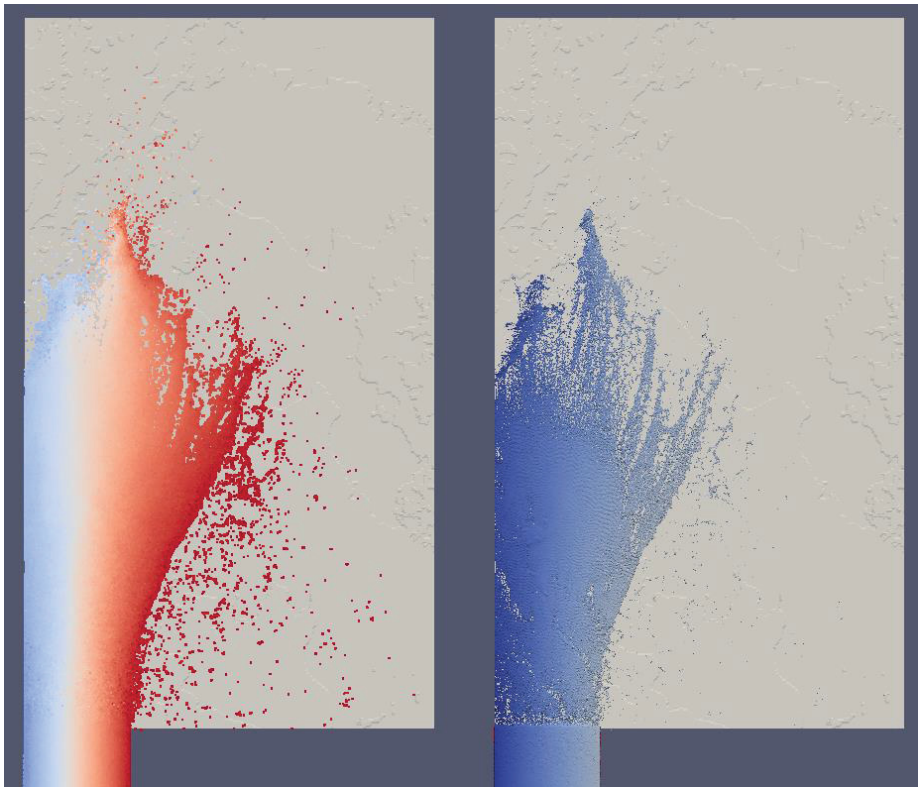


Figure 44. Case 3 225 seconds

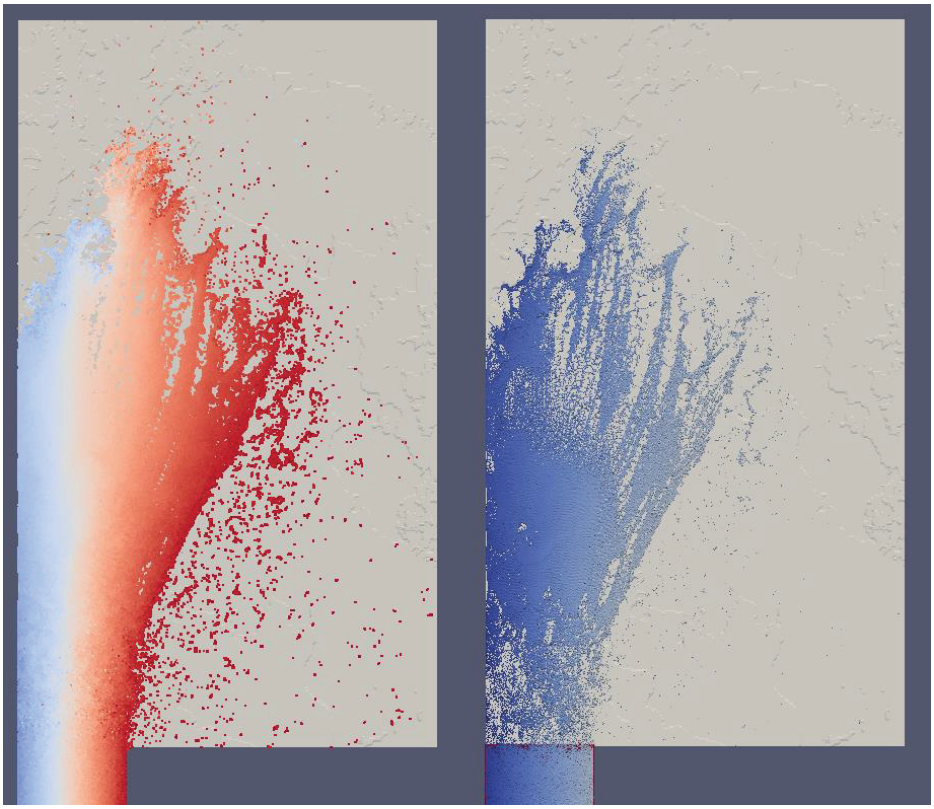


Figure 45. Case 3 300 seconds

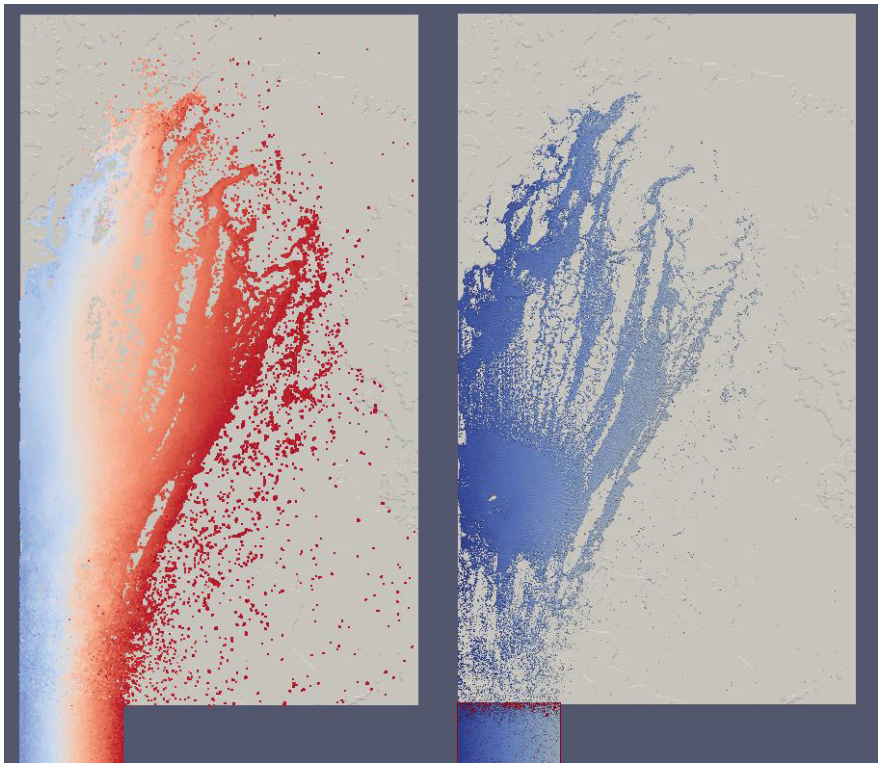


Figure 46. Case 3 375 seconds

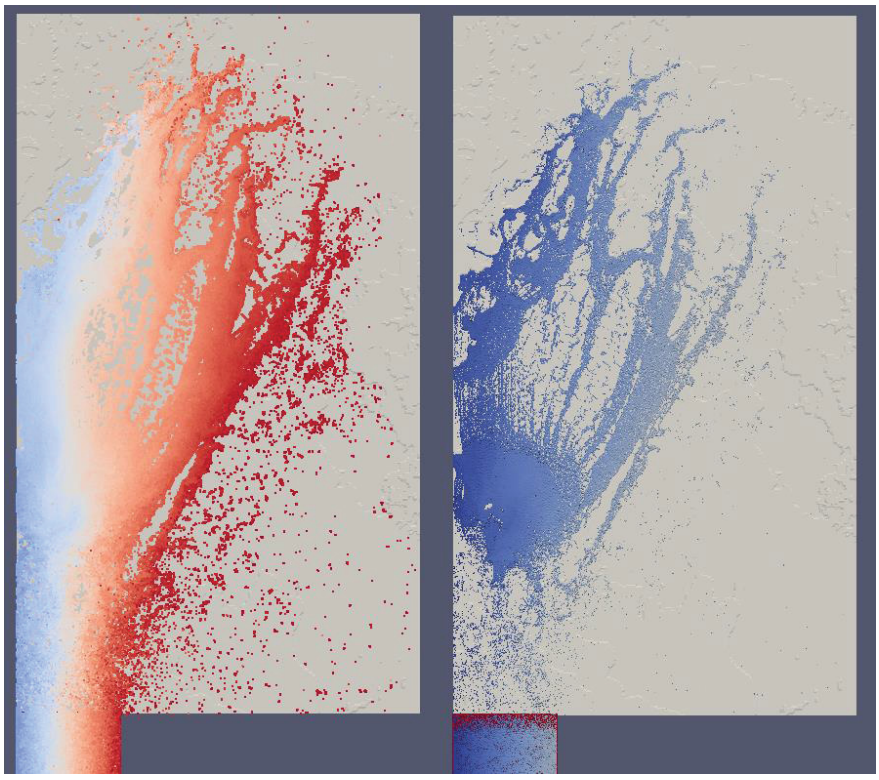


Figure 47. Case 3 450 seconds

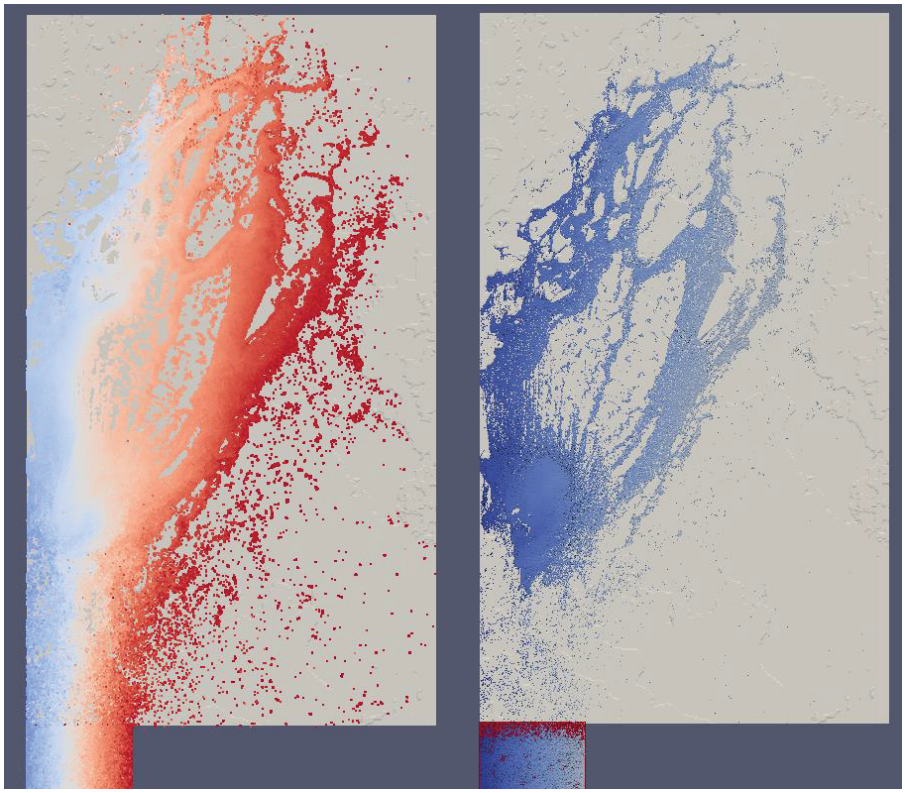


Figure 48. Case 3 525 seconds

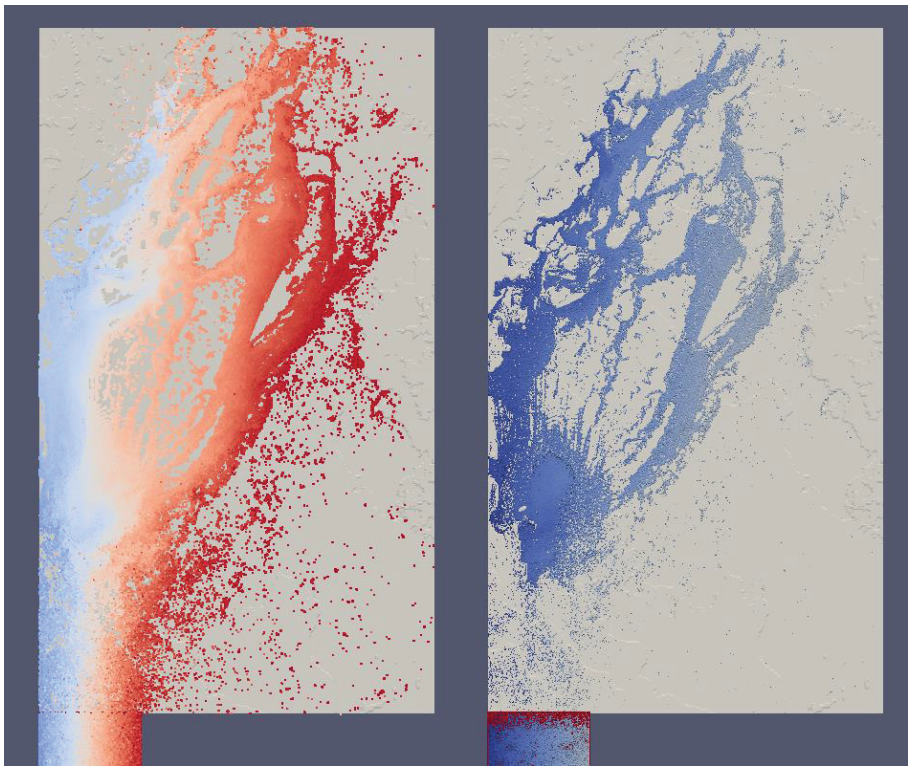


Figure 49. Case 3 600 seconds

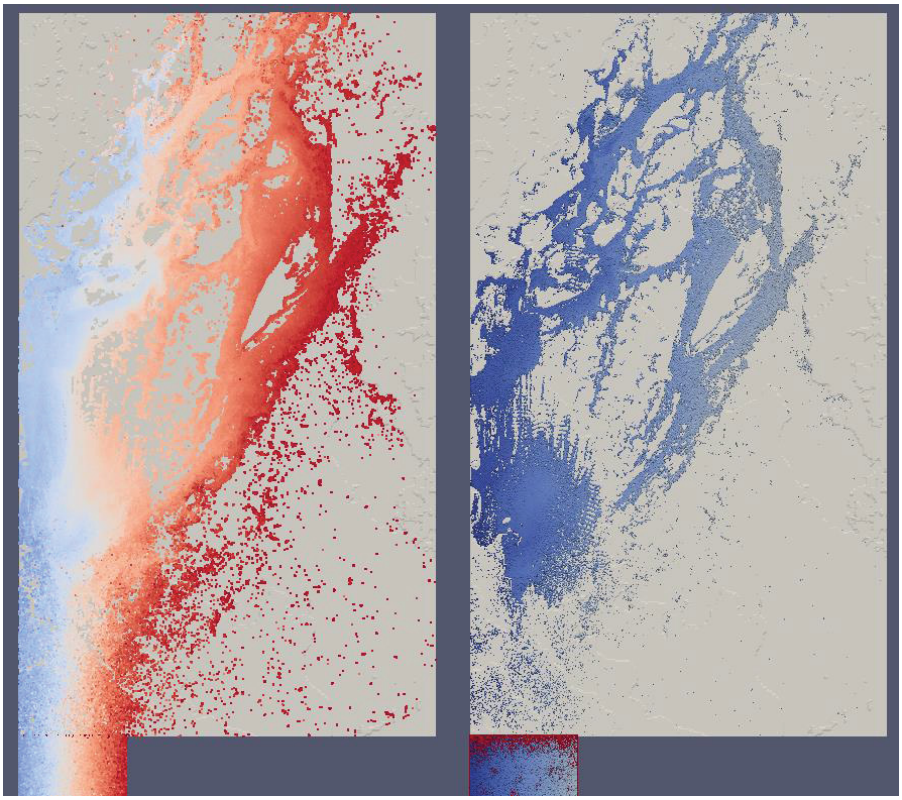


Figure 50. Case 3 675 seconds

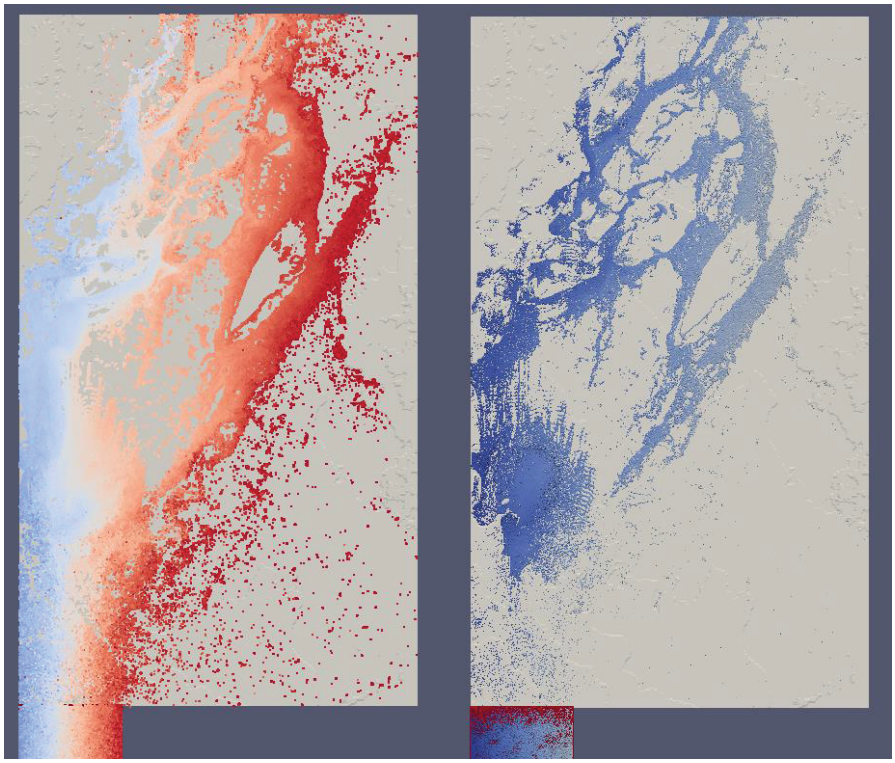


Figure 51. Case 3 750 seconds

3.3.4 Case 4

The results for case 4 were as follows.

- Distance between particles. 10m
- Simulated time. 750 seconds
- Computational runtime. 11.67 hours
- Time per second of simulation. 56.02 seconds
- Number of fluid particles. 7,304,100
- Total number of particles. 10,463,749
- Excluded number of particles. 5,002,284
- GPU memory usage. 7647.84 MB
- Output space usage. 799 GB

Figure 52 through Figure 62 show screen shots of the simulation at different points in time. The left side is the PartVTK output which shows the individual particles and the right side is the IsoSurface output which shows the surface.

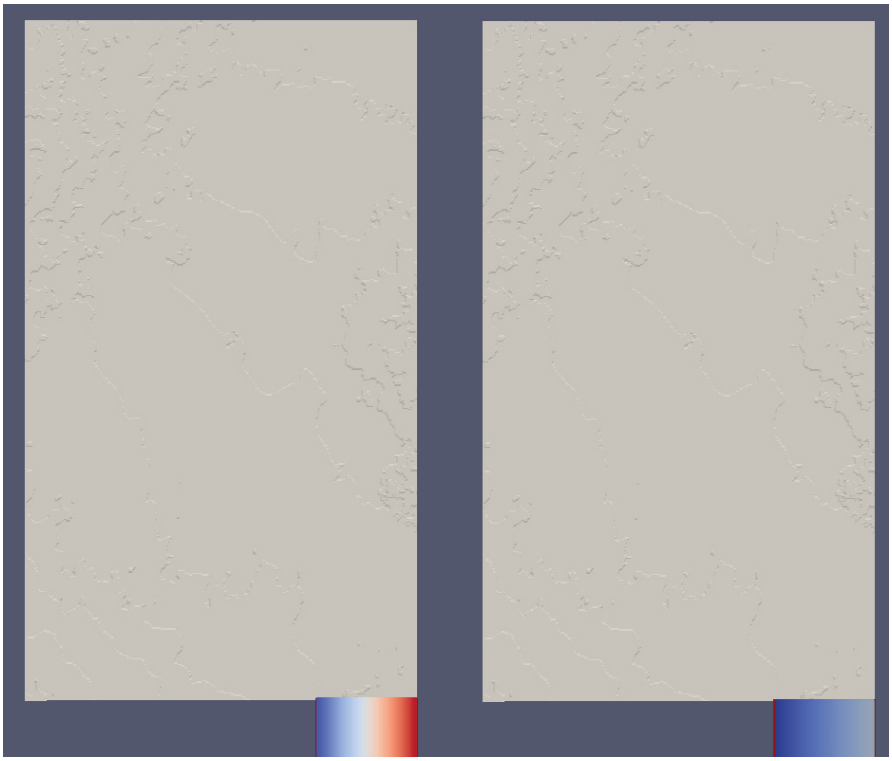


Figure 52. Case 4 beginning

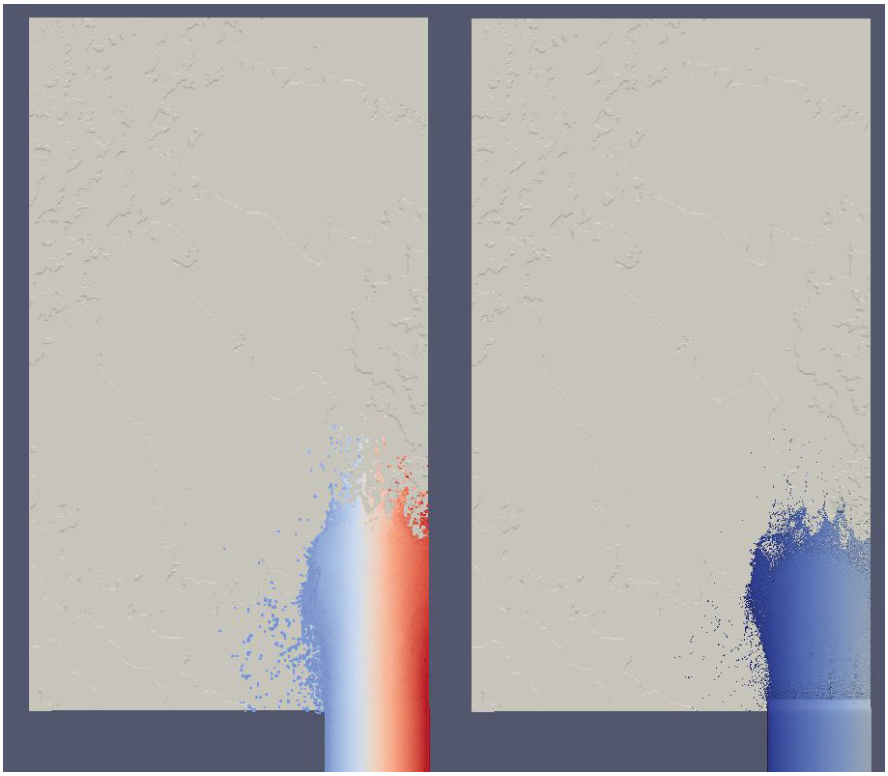


Figure 53. Case 4 75 seconds

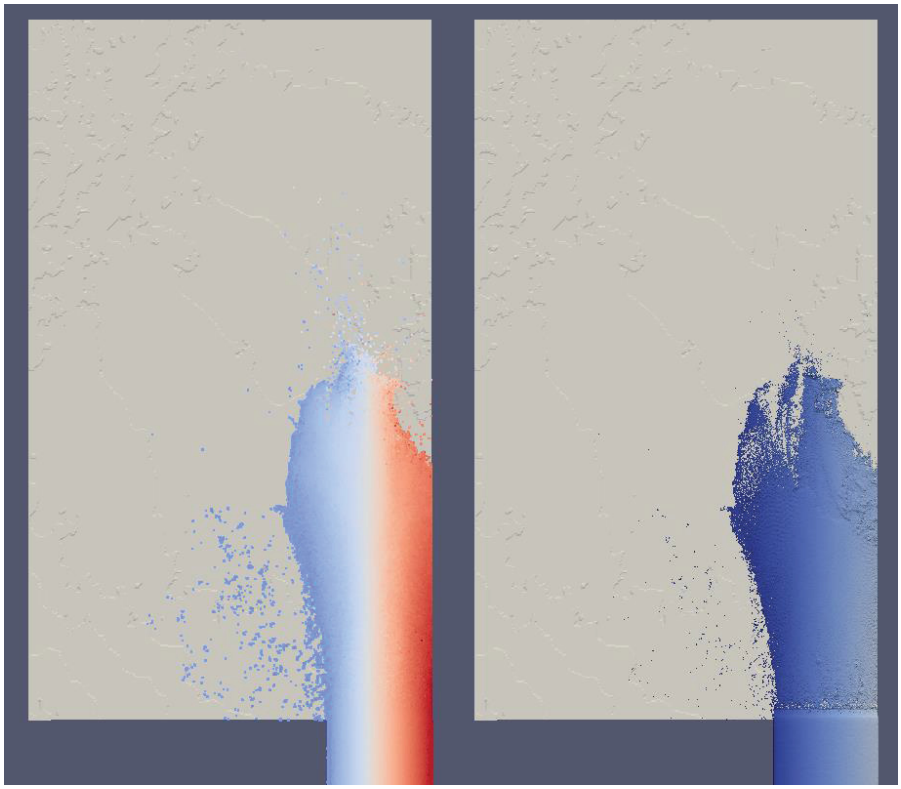


Figure 54. Case 4 150 seconds

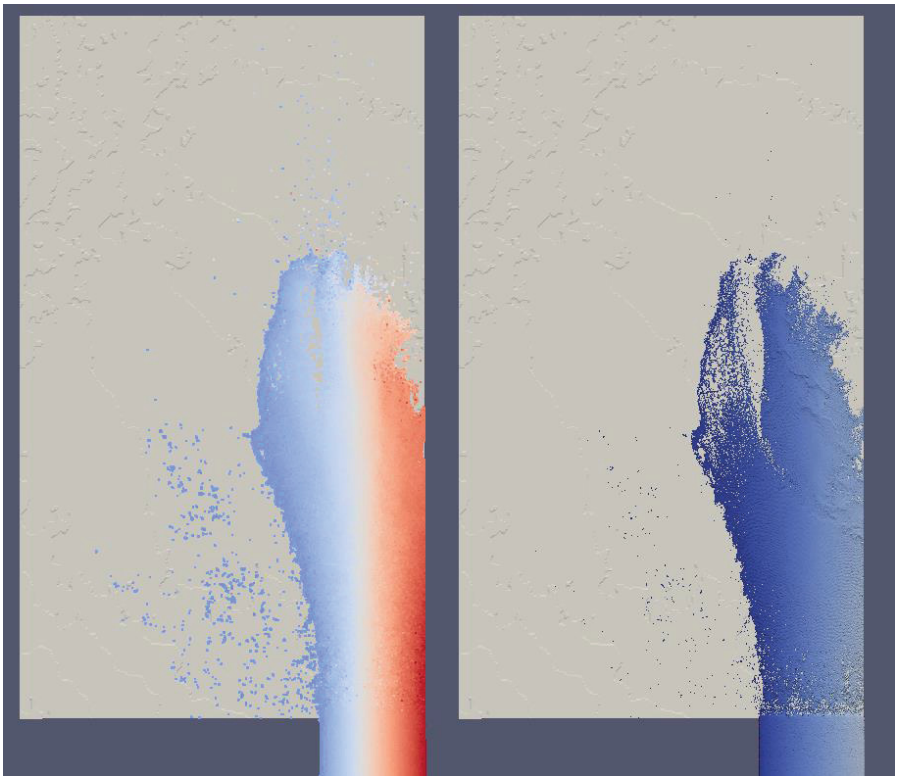


Figure 55. Case 4 225 seconds

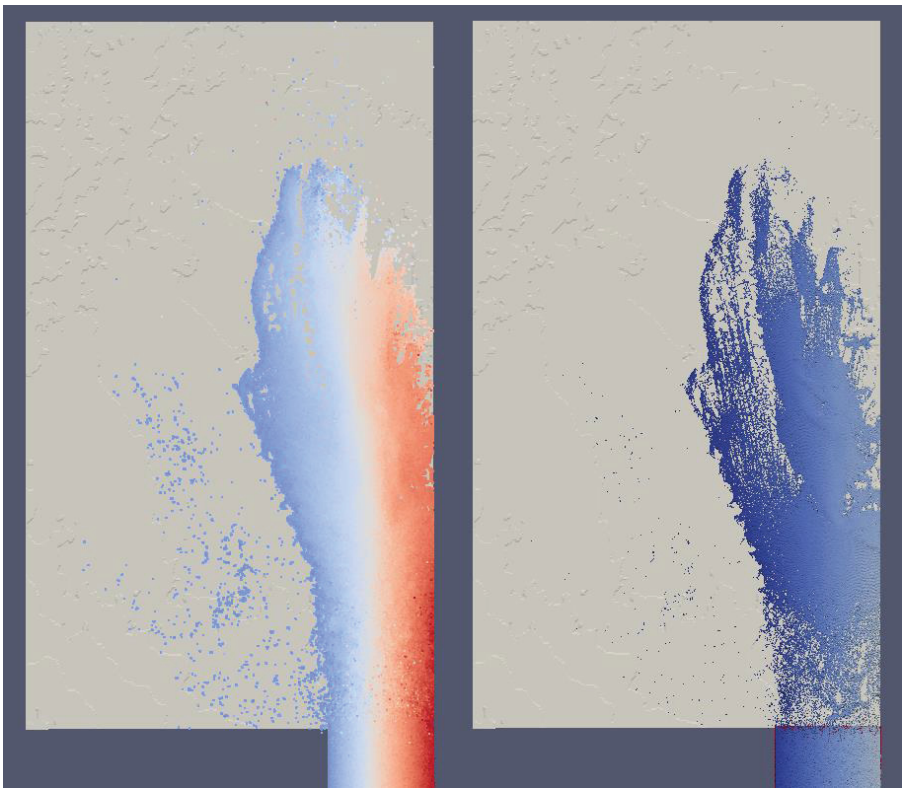


Figure 56. Case 4 300 seconds

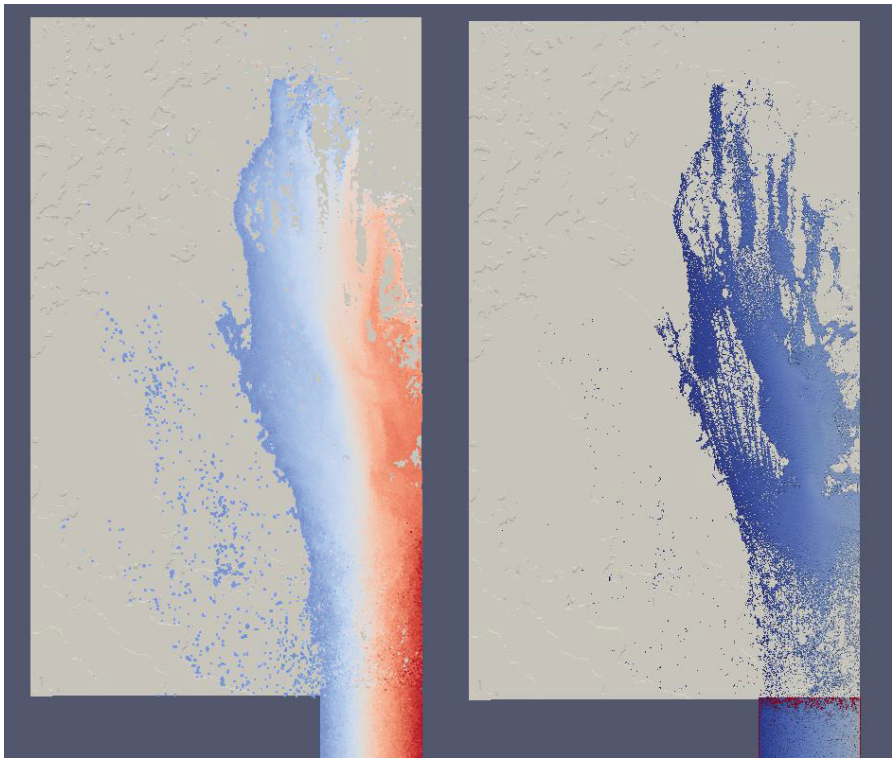


Figure 57. Case 4 375 seconds

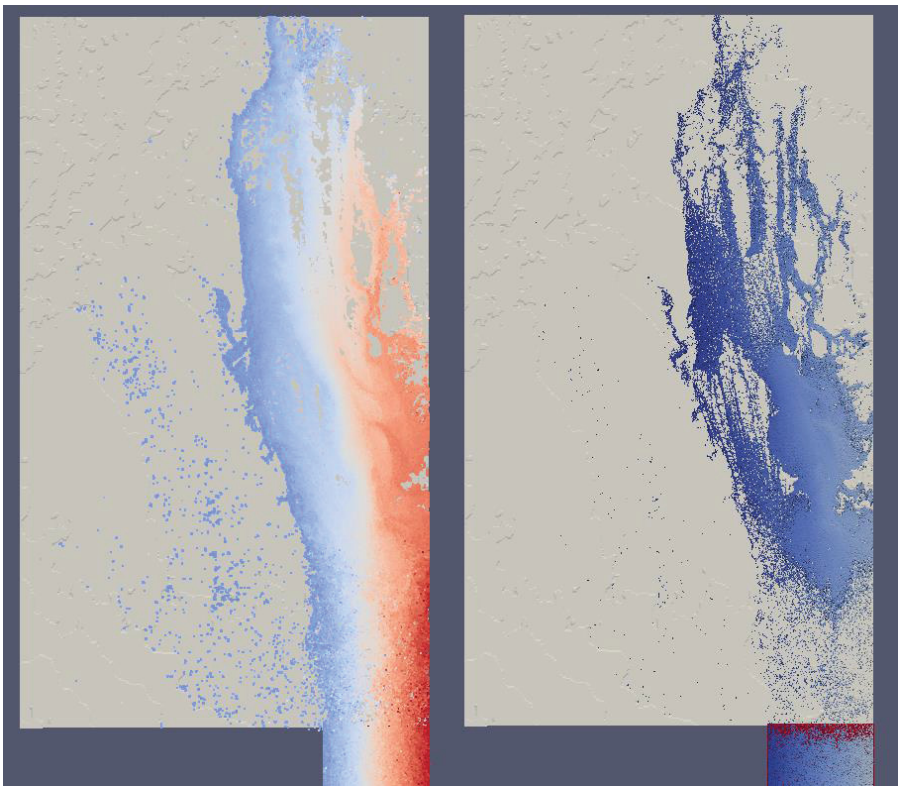


Figure 58. Case 4 450 seconds

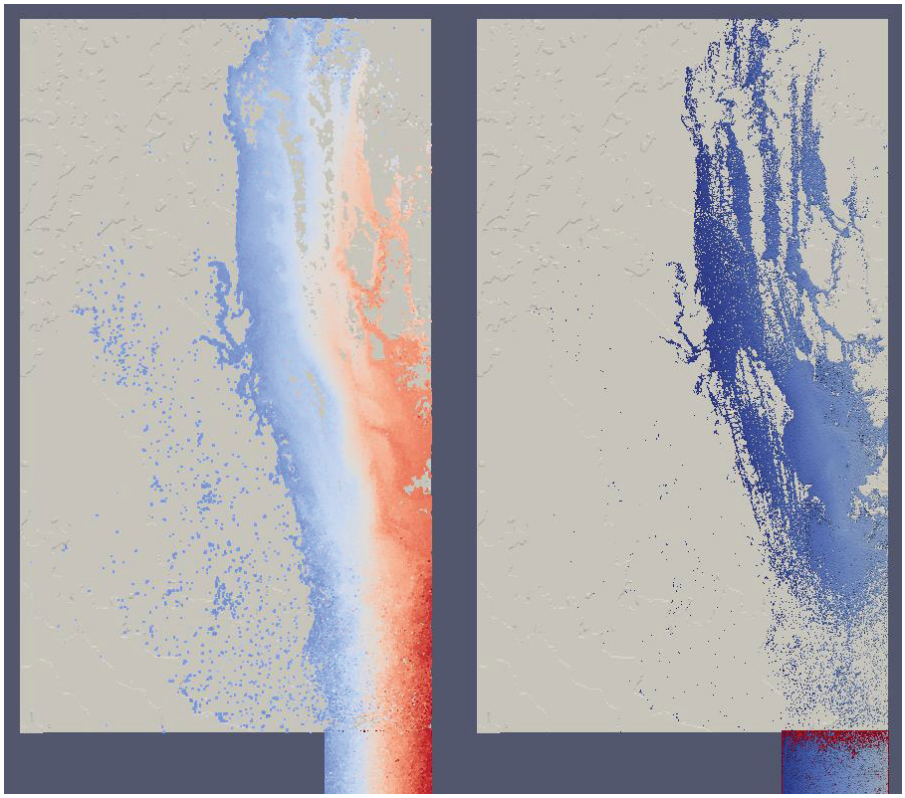


Figure 59. Case 4 525 seconds

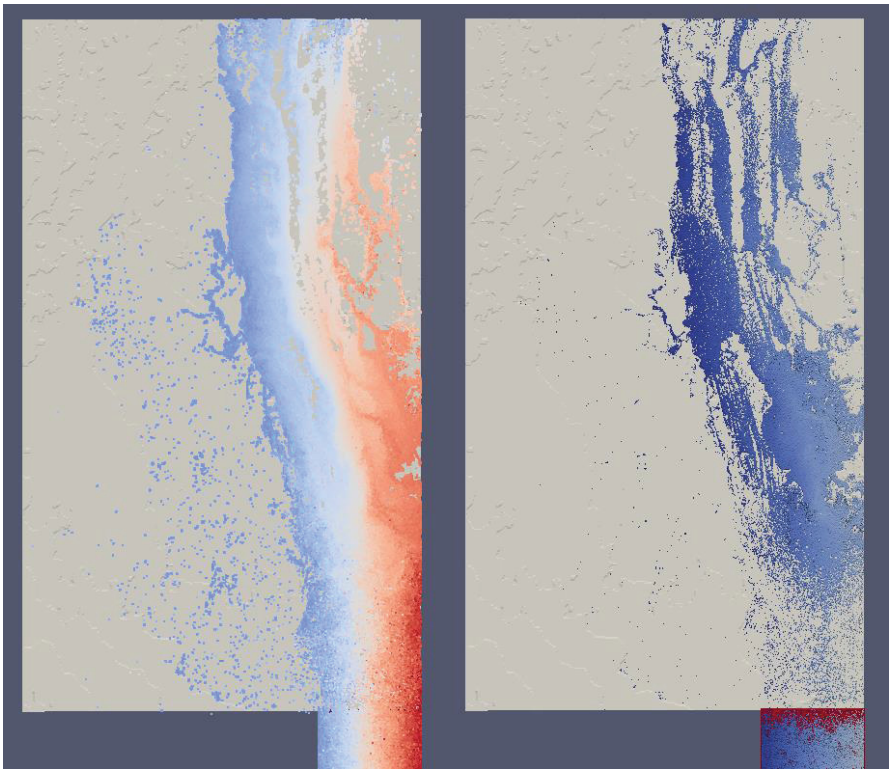


Figure 60. Case 4 600 seconds

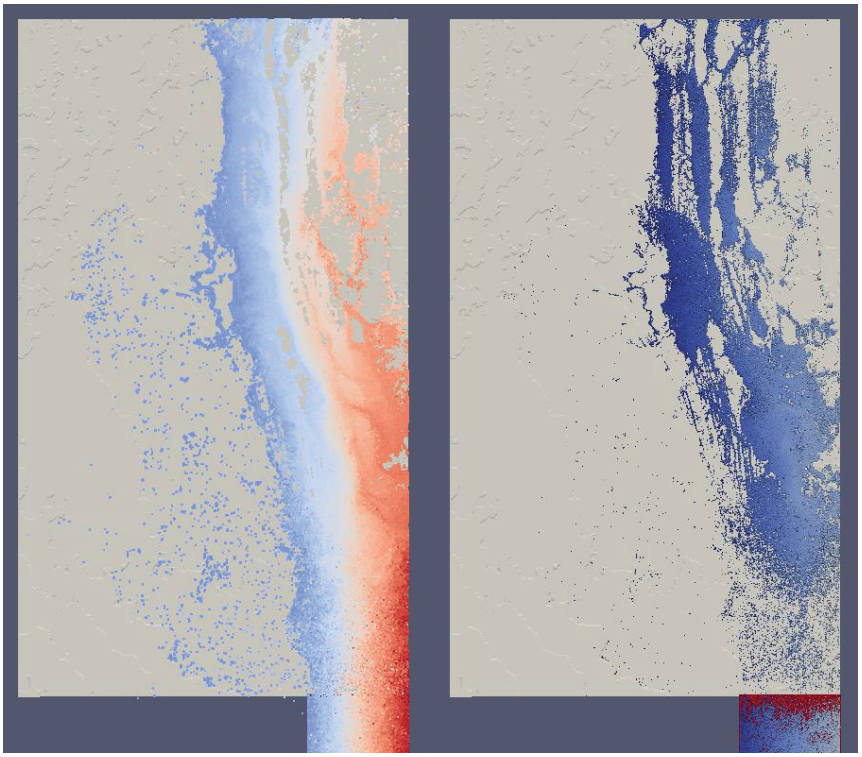


Figure 61. Case 4 675 seconds

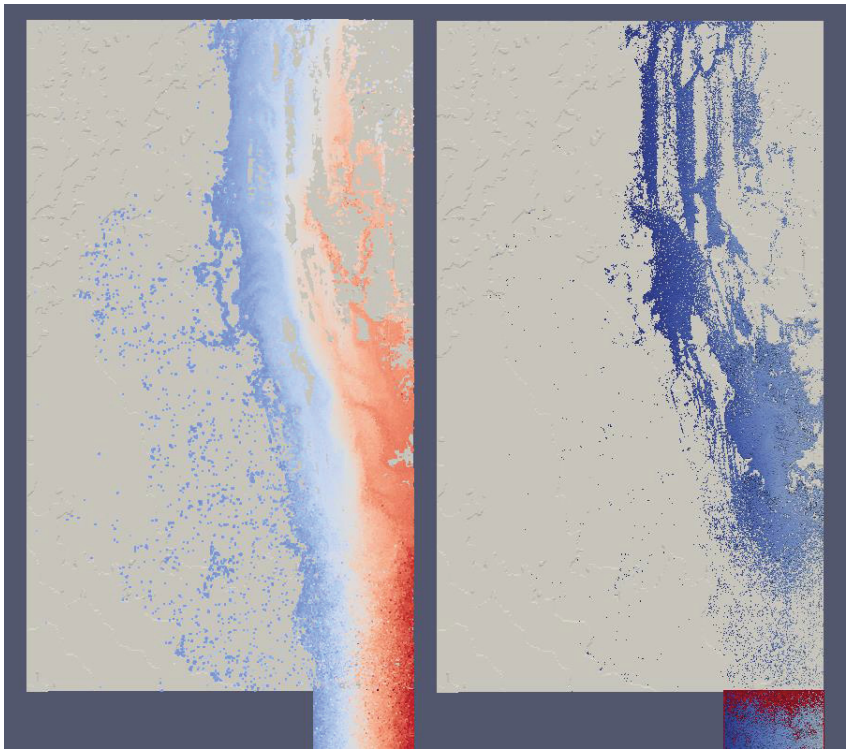


Figure 62. Case 4 750 seconds

3.4 Large Scale SPH Summary

Riverine flooding was simulated using an SPH code. From the results shown in Chapter 3, the topography did influence the flow of the water as would be seen when an increase of water is experienced in a watershed like the one simulated. This flooding scenario could be modified to see:

- Different depths of water flow across the topography as a function of different initiating events
- Different speeds flow across the topography
- Which facilities may be impacted, if any
- Duration of the event in order to understand timing issues that may be relevant

Additional improvements could be made by determining a better way to create the flood. An example would be to have the topography file cover more land so that water could start further away from the site of interest. This modification would allow the water to take a more natural flow into the area of interest rather than have a slab of water. Alternatively, we might explore the use of particle emitters. It might also be of interest to use another code, such as GPUSPH or Neutrino, to simulate the flooding scenarios. This would allow the outputs to be compared to see if the flooding results are similar for validation.

Although it has been shown that SPH can be used to demonstrate large scale runoff patterns, the restrictions of large particle size and long runtimes reduce the usefulness and advantages of using SPH methods for these scenarios. The detail needed for effects on a facility would impossible without an adaptive strategy.

4. COUPLING 2D AND SPH METHODS

By coupling 2D simulation methods with SPH, we could take advantage of the speed and large domain of the 2D methods and the detailed 3D interactions and calculation capabilities of SPH. To develop this methodology, we added a fictional NPP test area near the Teton River, downstream of the dam failure described in section 2.3. Here we will describe how to couple the flood data from the dam failure to run a more refined 3D SPH simulation for the NPP site.

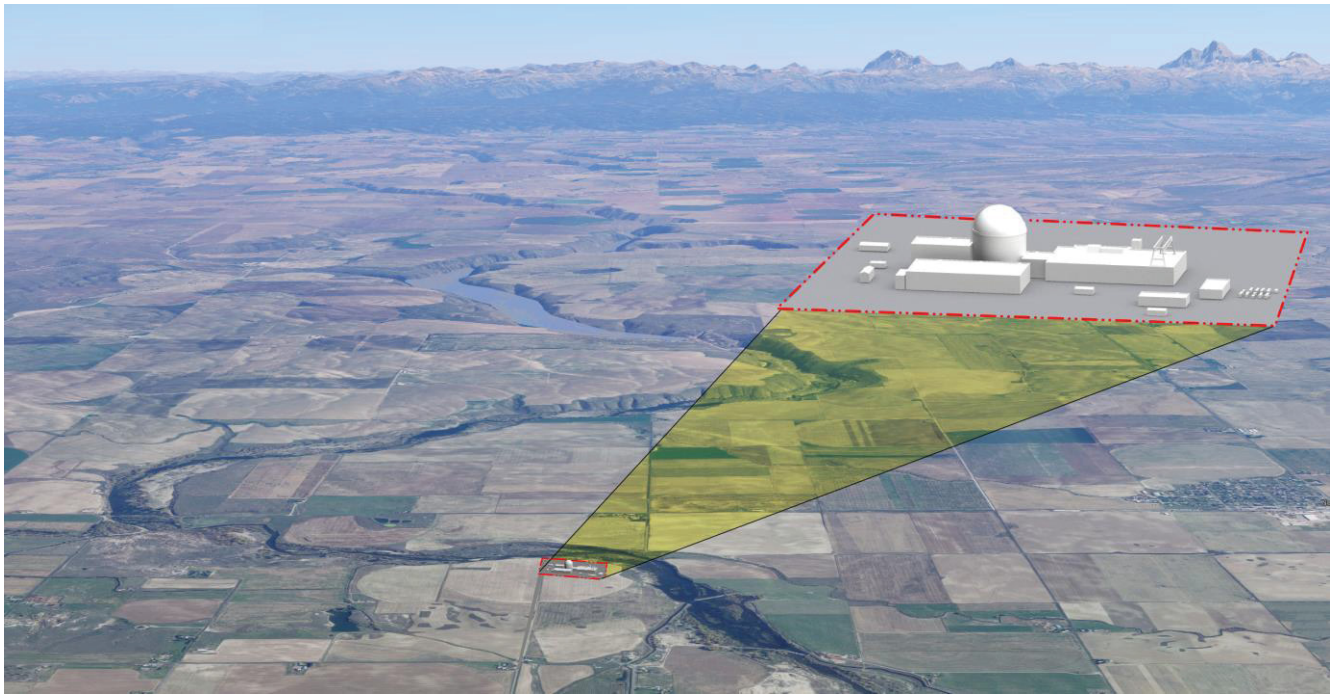


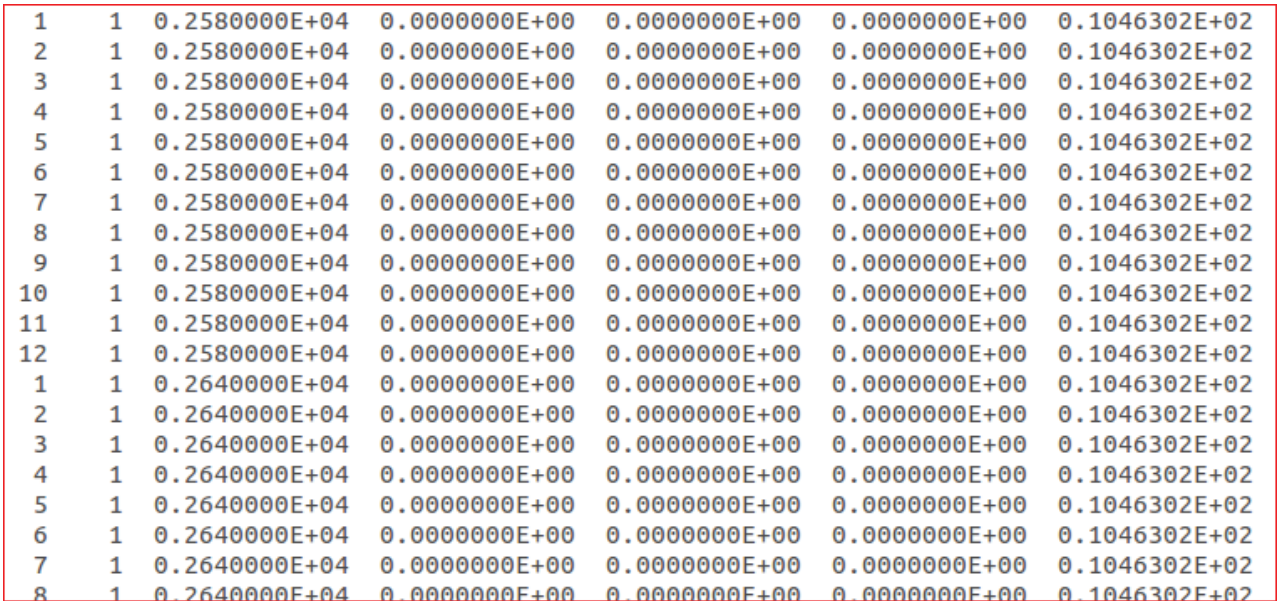
Figure 63. Map of a NPP added below the Teton dam.

4.1 Boundary Data

Simulations from GeoClaw output the water height and velocity data for the defined grid area over time. Only a small area of this data is relevant to the flooding of a specific facility.

GeoClaw simulations result in one output file per time step, each containing data from several AMR grids that collectively cover the whole computational domain. However, only data along a boundary surrounding a facility is needed to couple GeoClaw simulations with the 3D SPH simulations. To extract this data GeoClaw gauges were used. GeoClaw gauges allow the user to specify certain boundaries on a computation domain, at which solutions will be saved to an output file.

As mentioned in section 2.2, GeoClaw numerically computes solutions to the conservative shallow wave equations. These equations are a system of partial differential equations comprised of the dependent variables h, u and v, where h, u and v are functions that give the fluid column height, the velocity in the x direction, and the velocity in the y direction. Gauge output files contain seven columns (see Figure 64). The columns contain, the gauge number, AMR level, time step, height, x-momentum (expressed as hu), y-momentum (expressed as hv), and η (the difference between the water's current level and the water's natural level) on the specified boundary.



1	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
2	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
3	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
4	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
5	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
6	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
7	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
8	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
9	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
10	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
11	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
12	1	0.2580000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
1	1	0.2640000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
2	1	0.2640000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
3	1	0.2640000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
4	1	0.2640000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
5	1	0.2640000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
6	1	0.2640000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
7	1	0.2640000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02
8	1	0.2640000E+04	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1046302E+02

Figure 64. An image of GeoClaw gauge output files.

Once gauges have been placed and the gauge output files have been obtained, the next step is to extract data needed specifically for SPH methods. A python script was written for this purpose. The python script takes in the Neutrino time step as a parameter and extracts only data with a corresponding time step from the gauge file. The python script then calculates the horizontal velocities from the given momentums and writes the results to a new file.

To compute the horizontal velocities, each element of the columns hu and hv must be divided by h. This poses a problem, as $\frac{hu}{h}$ and $\frac{hv}{h}$ are not numerically stable when h is small (meaning that small changes in the machines precision can result in unreasonably large velocities). To avoid this, we consider cells containing less than 1mm of water to have a velocity of zero.

For the dam break simulation, a rectangular boundry closely surrounding the fictional NPP test. The north-east corner of the rectangular boundry was located at 43.913661° N, 111.620150° W and measured 302.6m x 302.56m. One gauge was stationed on each vertex of the rectangular parameter, and two were stationed along each edge. (See Figure 65-67)

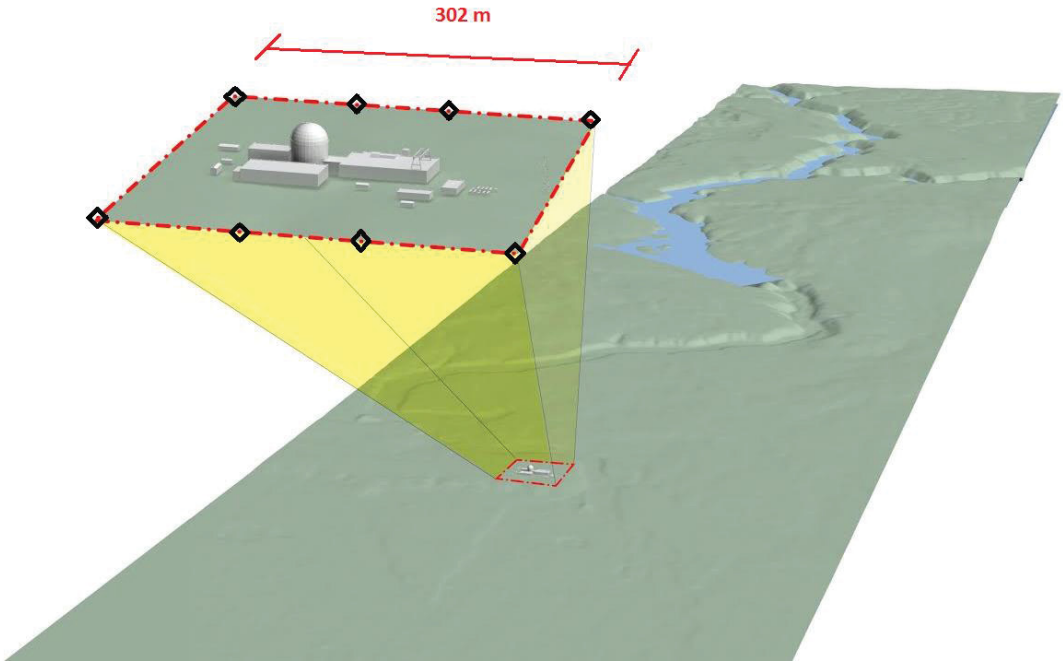


Figure 65. The boundary area for the NPP along the flood path and simulation area.

The simulation was run, and the python script was used to extract data from the gauge file. An image of the resultant data is shown below.

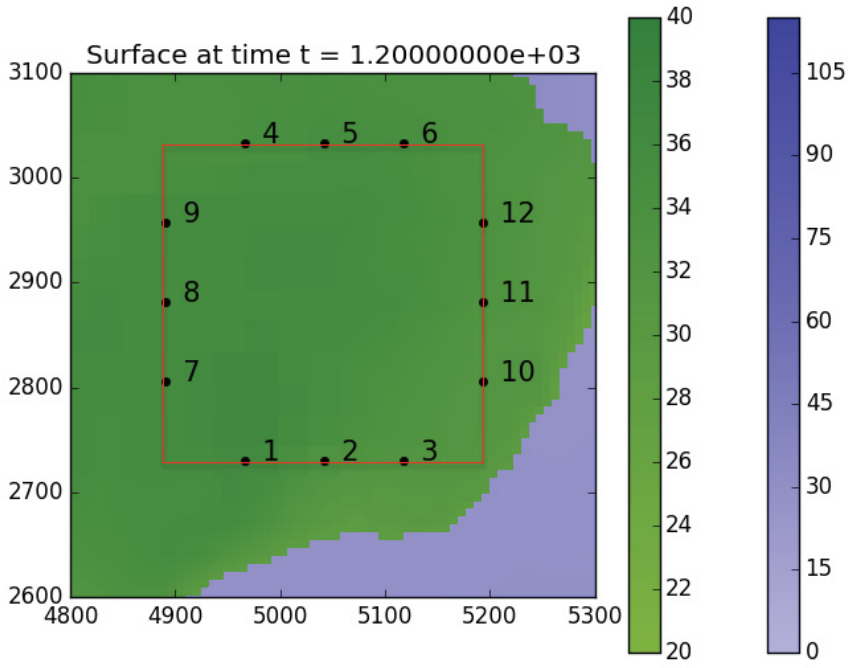


Figure 66. GeoClaw gauge location and simulation data at time = 1.2e3

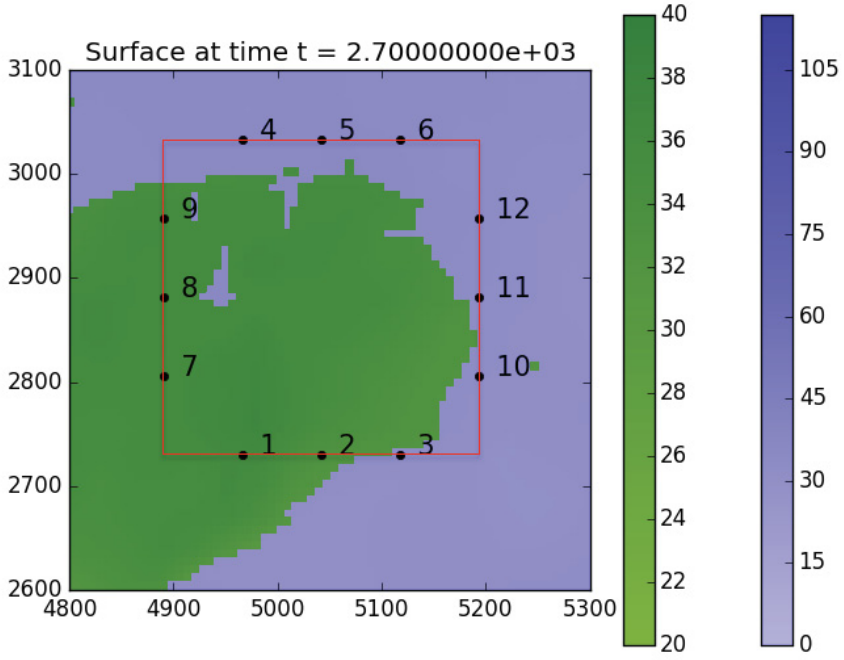


Figure 67. GeoClaw gauge location and simulation data at time = 2.7e3

4.2 SPH Coupling

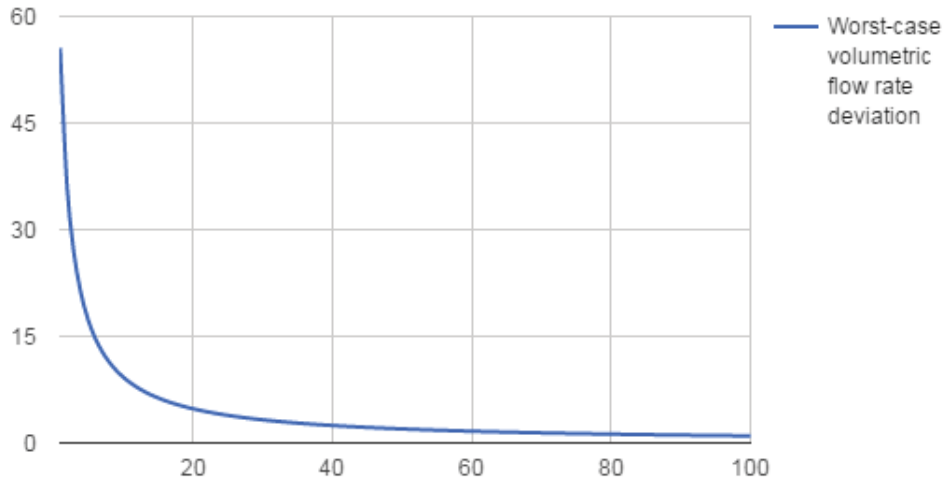
The 2D to 3D SPH coupling development has been initiated. Because data from GeoClaw is preprocessed, initial coupling will be one-way. In general this is adequate for most scenarios with a single 3D SPH area because what happens inside the boundary would have only minor effects on the 2D equations, and practically no secondary feedback into the boundary area. However, in special conditions it may be necessary to have 2-way coupling. The coupling will be integrated into the IISPH Solver and Simulation System "Neutrino". The following open boundary conditions will be implemented along with a particle emitter with specified velocities from the boundary conditions sampling of data from GeoClaw gauges positioned at the boundary of the 3D domain overlapping with the GeoClaw domain.

The chief component in Neutrino of the coupling with GeoClaw is an emitter providing inflow open boundary for one-way coupling from GeoClaw to Neutrino. But a future extension is planned to provide outflow open boundary for complete two-way coupling for Neutrino data back to GeoClaw.

The following are the main aspects of the flow rate emitter.

- Rectangular in shape.
- Is defined by a position, a height, a width and a flow rate, or alternatively a velocity.
- Generates fluid particles with a rectangular, uniform grid-packed configuration.
- Minimizes particle spacing deviation from the particle rest spacing.
- Has a (near) perfect volumetric/mass flow rate accuracy if the height and width are multiples of the particle rest spacing.
- Has, otherwise, a decreasing worst-case flow rate deviation when the height and width are large compared to the particle rest spacing.
    - Considering a square emitter, noting n the number of particles generated in each height and width direction, and err the volumetric flow rate deviation, we have: err <= 1-((2n)^2)/((2n+1)^2). Here are some examples of the error in flow introduced by the flow emitter.
        - n = 1 -> err <= 55.6%
        - n = 2 -> err <= 36.0%
        - n = 5 -> err <= 17.4%
        - n = 10 -> err <= 9.3%
        - n = 20 -> err <= 4.9%
        - n = 50 -> err <= 2.0%
        - n = 100 -> err <= 1.0%

          These results are shown below:



The following features will be implemented for handling open boundaries.

- Modified flow rate emitter using the ghost particle concept.
- The emitter is positioned outwards from the boundary by h+s/2, with h the interaction radius and s the particle rest spacing. This domain of size h+s/2 is referred to as ghost domain.
- Ghost particles are generated with their velocities set to the velocity corresponding to the defined flow rate.
- Ghost particle pressures are extrapolated from adjacent fluid particles. These pressures are updated at each pressure iteration.
- Ghost particles are included in the neighborhood to improve the accuracy of the SPH computations near the boundary.
- Ghost particles are advecated in the flow using a null acceleration.
- Ghost particles become fluid on exit from the ghost domain.
- Additional constraints are set to the time step update to ensure the stability.

This system therefore ensures the mass conservation due to the flow rate accuracy of the emitter.

To summarize, the following are constraints are placed for the one way coupling from GeoClaw to Neutrino.

- Inflow boundary conditions are used for each boundary cell, or alternatively for each measurement point, at the Shallow Water (SW)/SPH boundaries.
- The heights and velocities of the inflow boundaries are set using the height and velocity Shallow Water data.
- Bi-linear interpolation is used to initialize the SPH fluid domain from Shallow Water data.
- Synchronizes the SPH and SW simulations are synchronized by linearly interpolating in time the SW data.

The same methods used for coupling with GeoClaw, provides a template to couple with other 2D simulation software. Implementing this coupling methodology will expand the external event domain capabilities for SPH.

5. CONCLUSIONS

Different simulation methods are better suited for particular problem scopes. For example, 2D methods embodied by GeoClaw, SWMM, HEC-HMS are best for very-large scale flooding scenarios and have been validated for their areas of use. Although SPH can be used in some degree to duplicate large scale flooding, with current computational resources, it cannot achieve the accuracy and speed of the 2D methods. SPH simulation is ideal for midsize- and smaller-scope problems and can provide detailed evaluation in those areas as demonstrated in Reference [10]. However, by coupling 2D approaches with 3D SPH simulation methods, large scale flooding simulations can give detailed system structure and component flooding and failure data in a computationally cost effective manner.

6. REFERENCES

[1] C. Smith, C. Rabiti, R. Martineau and R. Szilard, "Risk Informed Safety Margin Characterization (RISMC) Pathway Technical Program Plan," Idaho National Laboratory, INL-EXT-11-22977, Revision 3, 2015.

[2] FEMA, "Hydrologic Models Meeting the Minimum Requirement of National Flood Insurance Program," FEMA, 12 6 2015. [Online]. Available: http://www.fema.gov/hydrologic-models-meeting-minimum-requirement-national-flood-insurance-program. [Accessed 20 10 2015].

[3] EPA, "Storm Water Management Model (SWMM)," EPA, [Online]. Available: http://www2.epa.gov/water-research/storm-water-management-model-swmm. [Accessed 20 10 2015].

[4] A. A. Smith, "Miduss Version 2 Reference Manual," AAS Inc, Ontario, 2004.

[5] US Army Corps of Engineers, "Hydrologic Engineering Center," US Army Corps of Engineer, [Online]. Available: http://www.hec.usace.army.mil/software/hec-hms/. [Accessed 20 Oct 2015].

[6] D. L. G. a. M. J. B. R. J. LeVeque, "Tsunami modelling with adaptively refined finite volume methods," *Acta Numerica,* pp. 211-289, 2011.

[7] D. L. George, "Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959)," *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS,* no. 66, pp. 1000-1018, 2010.

[8] "DualSPHysics code v3.0," DualSPHysics, [Online]. Available: http://www.dual.sphysics.org. [Accessed 14 July 2015].

[9] "ParaView," Kitware, [Online]. Available: http://www.paraview.org/. [Accessed 28 Sept 2015].

[10] M. S. S. Prescott, "3D simulation of External Flooding Events for the RISMC Pathway," INL/EXT-15-36773, Idaho Falls, 2015.

[11] D. Gaston, G. Hansen and C. Newman, “MOOSE: A Parallel Computational Framework for Coupled Systems for Nonlinear Equations,” in *International Conference on Mathematics, Computational Methods, and Reactor Physics*, Saratoga Springs, NY, 2009.

[12] A. Crespo, J. Dominguez, A. Barreiro, M. Gomez-Gesteira and B. Rogers, "GPUs, a new tool of acceleration in CFD: Efficiency and reliability on Smoothed Particle Hydrodynamics methods," PLoS ONE, 2011. [Online]. Available: http://www.dual.sphysics.org.

[13] "Blender," [Online]. Available: http://www.blender.org/. [Accessed 28 Sept 2015].