

# **3D Modeling Engine Representation Summary Report**

Steven Prescott  
Ramprasad Sampath  
Curtis Smith  
Timothy Yang

September 2014



The INL is a U.S. Department of Energy National Laboratory  
operated by Battelle Energy Alliance

# **3D Modeling Engine Representation Summary Report**

**Steven Prescott  
Ramprasad Sampath  
Curtis Smith  
Timothy Yang**

**September 2014**

**Idaho National Laboratory  
Idaho Falls, Idaho 83415**

**<http://www.inl.gov>**

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

#### **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, do 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.

## **ABSTRACT**

Computers have been used for 3D modeling and simulation, but only recently have computational resources been able to give realistic results in a reasonable time frame for large complex models. This summary report addressed the methods, techniques, and resources used to develop a 3D modeling engine to represent risk analysis simulation for advanced small modular reactor structures and components. The simulations done for this evaluation were focused on external events, specifically tsunami floods, for a hypothetical nuclear power facility on a coastline.

# CONTENTS

ABSTRACT.....	iv
FIGURES.....	vi
ACRONYMS.....	vii
1. BACKGROUND.....	1
2. SIMULATION ENGINES .....	3
2.1 Houdini FX.....	3
2.2 Neutrino.....	4
3. 3D MODELING DEVELOPMENT .....	6
3.1 Facility Model .....	6
3.2 Terrain mapping .....	7
3.3 Detailed Modeling.....	9
4. SIMULATION DEVELOPMENT.....	11
4.1 Tsunami Simulation Setup .....	11
4.2 Tsunami Simulations.....	12
4.3 Interior Flooding Simulation .....	12
4.4 Proposed Development Improvements.....	14
5. CONCLUSIONS .....	15
6. REFERENCES .....	16
Appendix A –Simulating Scenarios.....	21

## FIGURES

Figure 1: 3D modeling engine development supporting the advanced PRA Framework.....	2
Figure 2: Houdini FX user interface .....	3
Figure 3: Example of the Neutrino modeling framework.....	5
Figure 4: Quarter section for the 3D model of a nuclear reactor facility to be used for simulation testing. ....	6
Figure 5: Half facility model of a nuclear reactor facility to be used for simulation testing. ....	7
Figure 6: Terrain mapping application used to generate a 3D map of a specified area. ....	8
Figure 7: The 3D map generated from the Fukushima site.....	9
Figure 8: Detailed model of reactor building.....	10
Figure 9: Neutrino Wave Piston Setup with Initial Conditions. ....	11
Figure 10 : Simulation of 16 meter tsunami on the $\frac{1}{4}$ facility model .....	12
Figure 11: Reactor building flooding from the bay door. ....	13
Figure 12: Water flow flooding example in a reactor building.....	14

## ACRONYMS

API	Application programming interface
FLIP	Fluid-Implicit-Particle
FPS	Frames per second
GB	Gigabyte
GHz	Gigahertz
GPU	Graphics Processing Unit
HPC	High Performance Computing
INL	Idaho National Laboratory
INI	Initialization
PIC	Particle-In-Cell
SPH	Smoothed Particle Hydrodynamics

# 3D Modeling Engine Representation Summary Report

## 1. BACKGROUND

Through physics and complex mathematical models, we have a good understanding how our world around us behaves. However for anything other than small problems, these models become complicated when attempting to represent reality. The field of computational physics applies numerical approximations and decomposes a problem into a large number of simple mathematical operations that can be solved using a computer. [[http://en.wikipedia.org/wiki/Computational\\_physics](http://en.wikipedia.org/wiki/Computational_physics)]

Many fields of study use computational physics to do calculations from protein folding for medicine to realistic effects in visualization. With the expansion of computation power and distributed computing, larger and more complex problems are able to be solved. We have targeted some of these standard methods and tools used in other fields to analyze flooding events for risk analysis of nuclear facilities such as small modular reactors.

The software prototype development for this activity used a generic light water reactor (LWR) model to represent hypothetical facility building locations (including relative locations of buildings on-site), elevations (including facility topology information), and hazard representation (including potential tsunami flooding frequency and magnitudes). The main focus of this 3D modeling engine work was to demonstrate feasibility with a model that is not overly simplified. At the start of the activity in FY14, the project team recognized that a detailed Small Modular Reactor (SMR) or Advanced SMR model (where here, “model” implies the site characteristic, 3D building details, locations of key components) was not available – consequently the suggestions from the team was to demonstrate the process using an available model (based upon a LWR). It was felt that this approach would be effective for three primary reasons:

1. Starting with an existing model will facilitate quick development on the project
2. Simulating actual events allows for some validation of the mechanistic flooding model approach
3. Flooding events affect nuclear power plants in (mostly) the same way regardless of the type of reactor technology (they impact site infrastructure, where the details will depend on the specific site and plant type)

As additional detailed models become available to the research team specific to aSMRs, we plan on incorporating them into our analysis platform.

This work on the 3D modeling engine supports the overall advanced PRA Framework that has been proposed as illustrated in Figure 1.



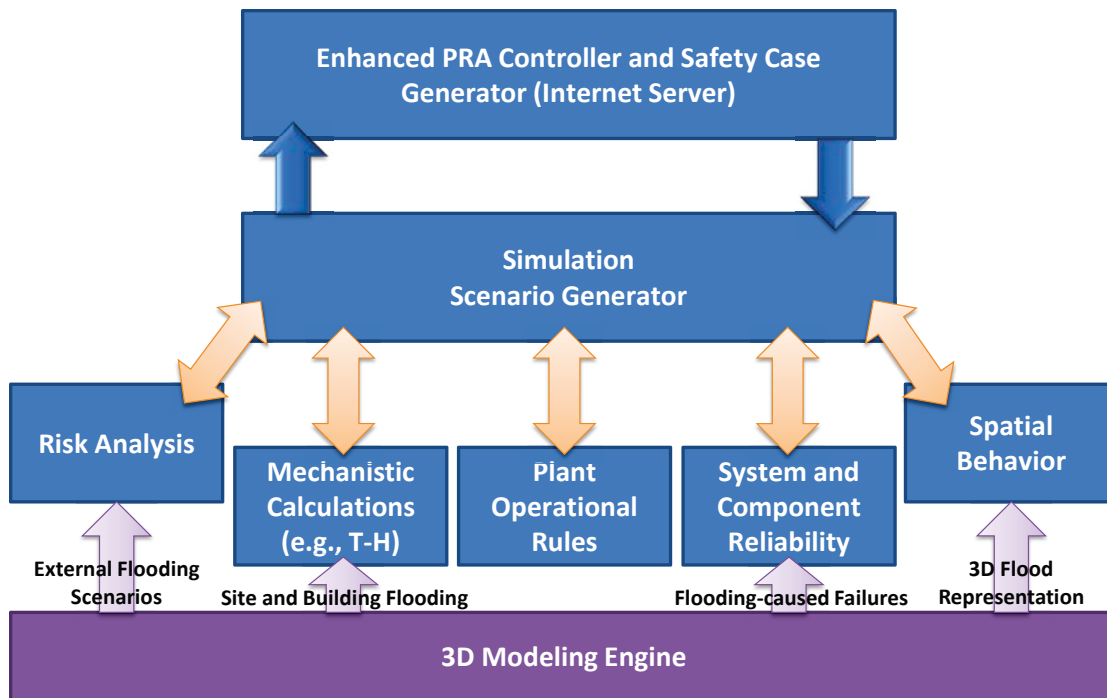


Figure 1: 3D modeling engine development supporting the advanced PRA Framework.

## 2. SIMULATION ENGINES

With our focus on flooding, a fluid simulation engine was needed. There are different 3D physics simulation engines using a variety of methods. Each has its advantages or disadvantages depending on the intended use. In this report, we describe a couple of approaches, and provide demonstration cases where these may be applicable for risk analysis.

### 2.1 Houdini FX

For this research, we initially started with a software package called Houdini FX. This application is a dynamic and widely used 3D simulation environment for visual effects. It also has an API for custom modifications which allowed us to communicate with it through other applications during each frame of the simulation. (See Figure 2) This feature makes it useful for incorporation into risk analysis modeling since the scenario evolution can be controlled (e.g., a failure can be triggered) during the calculations being performed.

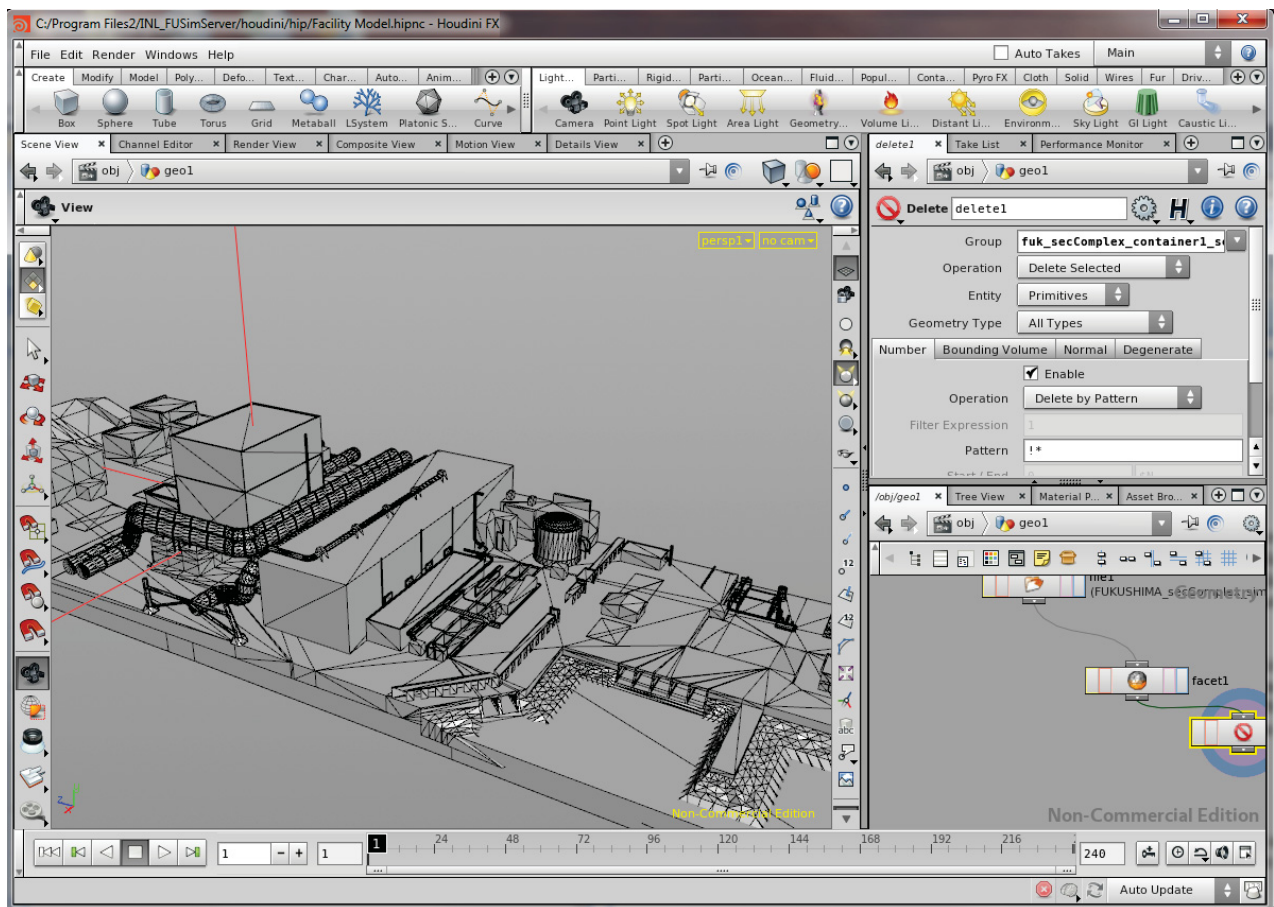


Figure 2: Houdini FX user interface

The Houdini package worked well for smaller simulation such as water flow inside of a room. However, it had two issues with larger simulations. First, the solver that is used in Houdini was grid based Fluid In Particle (FLIP) method. PIC/FLIP based solvers are extensively used in visual effects and produces visually interesting dynamic motion because it uses custom particle advection methods to combat numerical diffusion problems resulting in diffuse fluids. However, using a Smoothed Particle Hydrodynamics (SPH) solver with physics based modifications to handle boundaries guarantees conservation of mass with computation of pressure from weighted contribution of neighboring particles [1]. Although this option was present in Houdini to use its SPH solver, it was not suited for our needs. When Houdini's FLIP solver was used to generate a solution wave like a tsunami wave, the wave would quickly lost momentum and diffuse out. We tried to overcome this numerical diffusion by using higher resolution grids. But, in using higher resolution grids, we ran into the second problem. The Houdini engine was not able to support the memory requirement needed to run the larger simulations. Moreover even running higher resolutions to a maximum of what could be handled, this work around still did not produce a wave which preserved energy and it had excessive numerical diffusion

## **2.2 Neutrino**

To compare the results from Houdini as well as to combat the problems we encountered with Houdini's FLIP fluid solver, we decided to try "Neutrino." The Neutrino fluid solver developed by Neutrino Industries is based on Smooth Particle Hydrodynamics with a pressure solve to handle incompressible fluids. The Neutrino fluid solver also factors in accurate boundary handling, and adaptive time stepping to help to increase accuracy and calculation speed [2].

For this purpose, Neutrino Industries collaborated with the Idaho National Laboratory (INL) by providing the use of their solve engine and making custom modifications to the code base to help with analysis. Neutrino was able to handle not only the memory requirements needed for large simulations, but provided more accurate fluid movement with less numerical diffusion which preserved the solitary tsunami wave momentum required for our simulation. Neutrino's simulation framework was very flexible and provides a python based expression system (see Figure 2) to accurately model the movement of the wave machine based on Goring's 1978 numerical wave model (see Appendix A). Neutrino also provided a variety of tools to measure parameters in a section of the fluid simulated. This includes the wave height at a specific point, the average pressure and average velocity in a certain area/volume, as well as the flow rate across a certain area/volume.

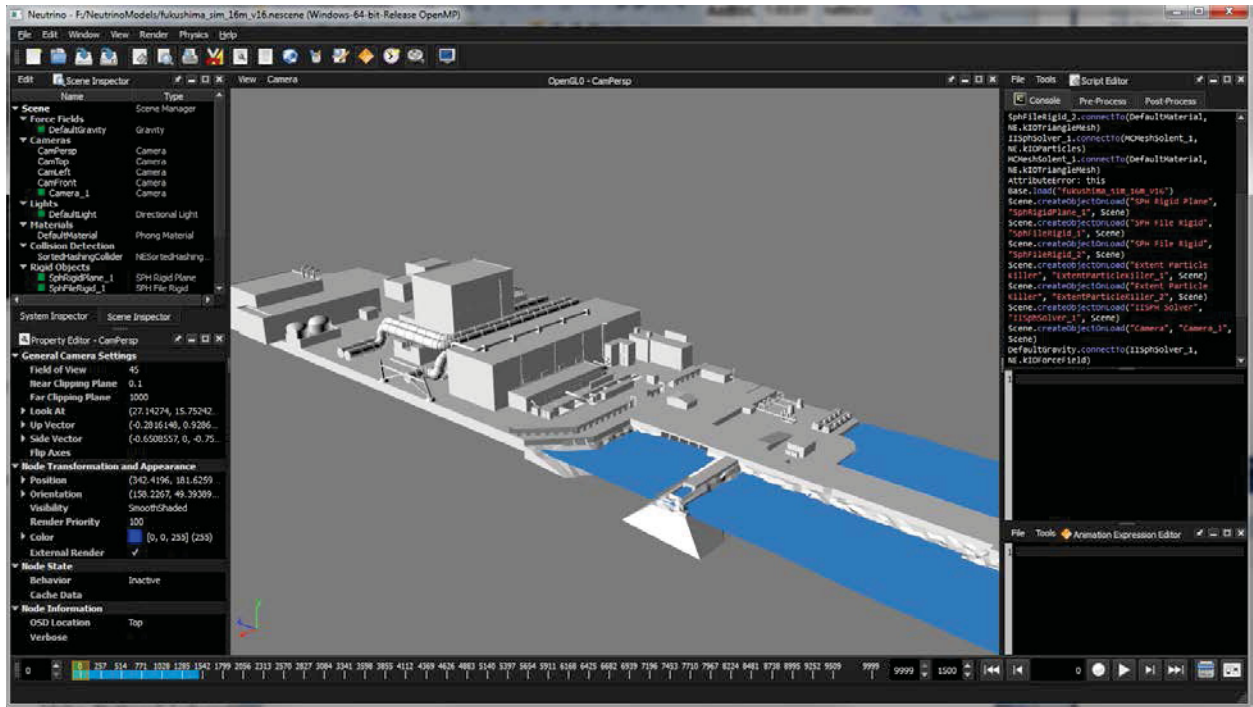


Figure 3: Example of the Neutrino modeling framework.

### 3. 3D MODELING DEVELOPMENT

Before simulations can be performed, a model of the desired facility must be made. Many tools exist for 3D modeling and, in general, it does not matter which tool is used as long as it can export the model to a format used by the simulation engine. Many model used for simulation consists of two parts, the visual model, and the collision model. The visual model consists of an accurate representation of the physical appearance of the facility and usually has image maps applied to the 3D structures to make them appear realistic. For our purposes, little emphasis is placed on the visual appearances, as long as an analyst can easily understand where and what items are such as buildings and components. More important is the collision model and in our case it is almost identical to the visual model except with additional invisible boundary items to bound calculations within the desired risk analysis scope.

#### 3.1 Facility Model

For this advanced analysis, a detailed 3D model of a nuclear facility was obtained through a 3D modeling clearing house for a nominal cost. We started with this visual model and, while it provided a good starting point, some additional work was required since it was designed with only visual uses in mind, not as a collision model.

Our initial facility modeling consisted of a  $\frac{1}{4}$  slice of the facility which included one reactor, one turbine building and other various structures. (See Figure 4) The second stage modeling consisted of the south half of the facility including support building. (See Figure 5) A larger model can show secondary effects such as water flowing around other buildings and terrain features forcing water in from other directions.

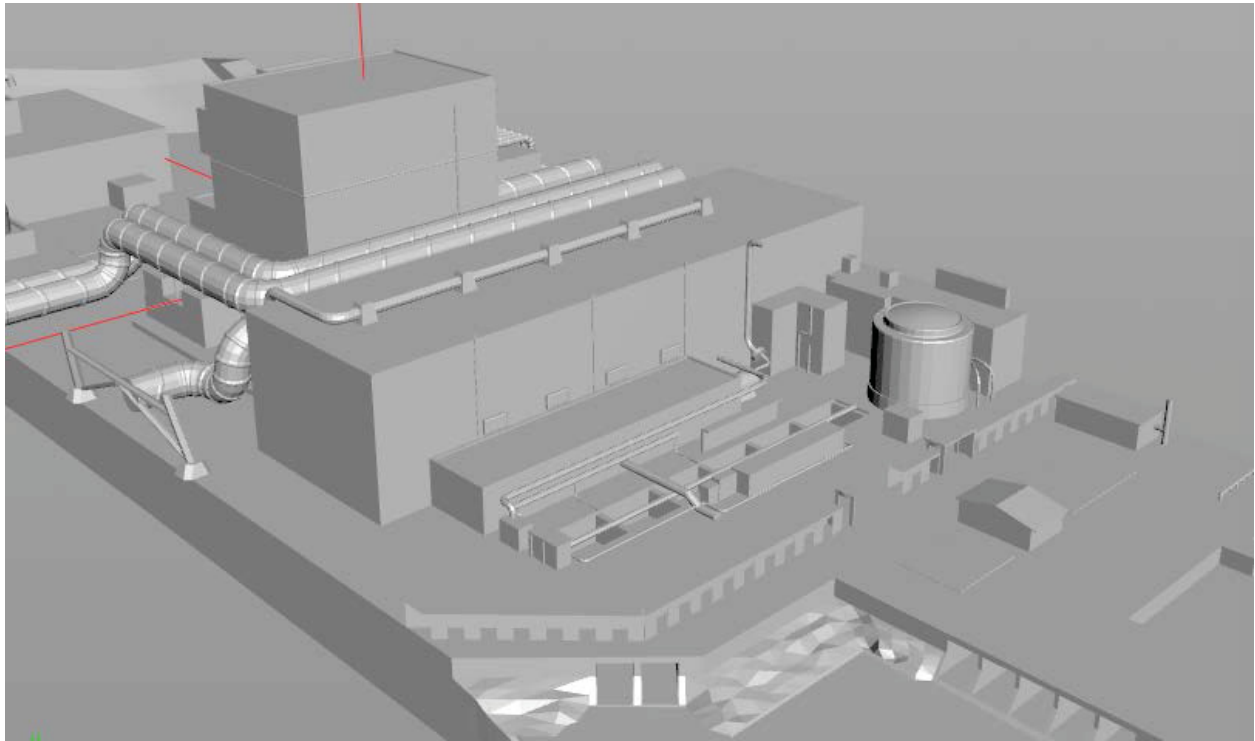


Figure 4: Quarter section for the 3D model of a nuclear reactor facility to be used for simulation testing.

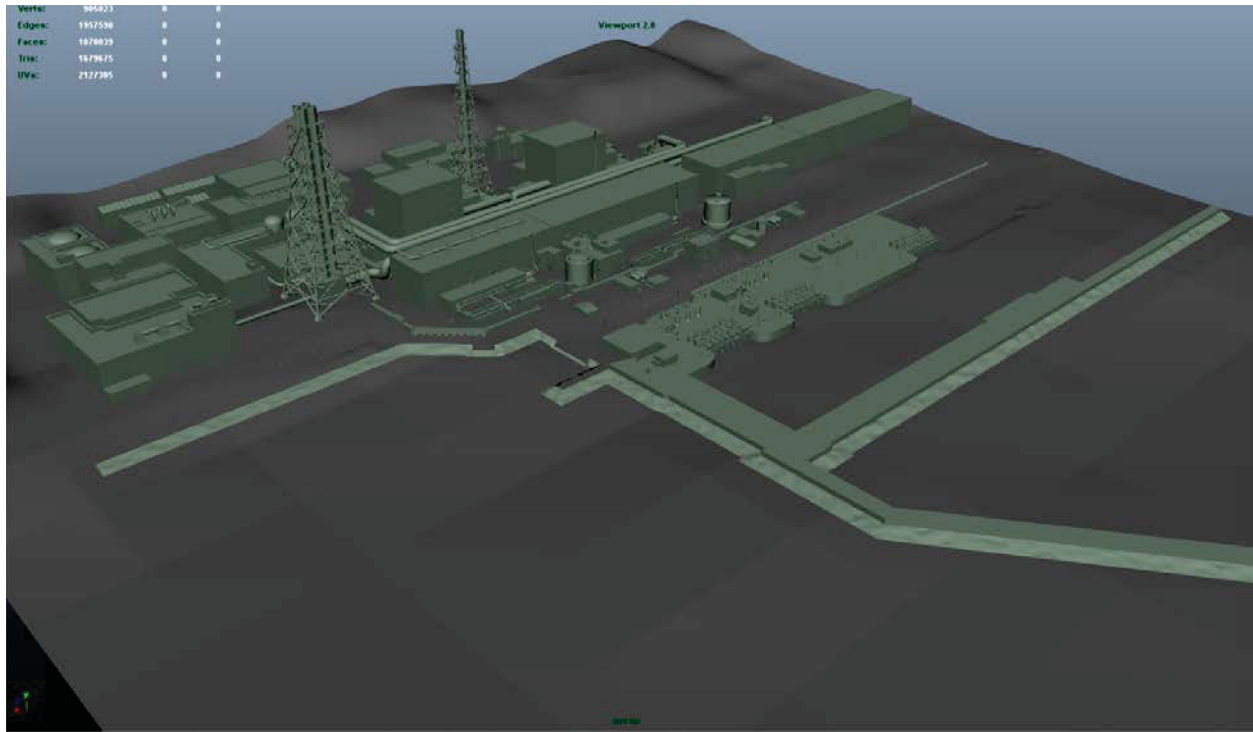


Figure 5: Half facility model of a nuclear reactor facility to be used for simulation testing.

## 3.2 Terrain mapping

Initial testing was done with a flat gradient on the facility and simple gradual slope for the ocean floor. However, to produce better results, an accurate terrain map showing the topography of the area is needed. To easily obtain this information, a standalone applet was created using Google's Elevation API. (See Figure 6) The Elevation API allowed for us to retrieve elevation levels from a set of points in a rectangular area anywhere on the surface of the earth. The user inputs latitude, longitude, distance and resolution data for the desired location. Using this information, the application gives a visual representation of the defined area on a Google map (and which can also be saved as a 3D model for incorporation into the facility 3D model). For our testing we used a location with a scale of five meters between grid points.



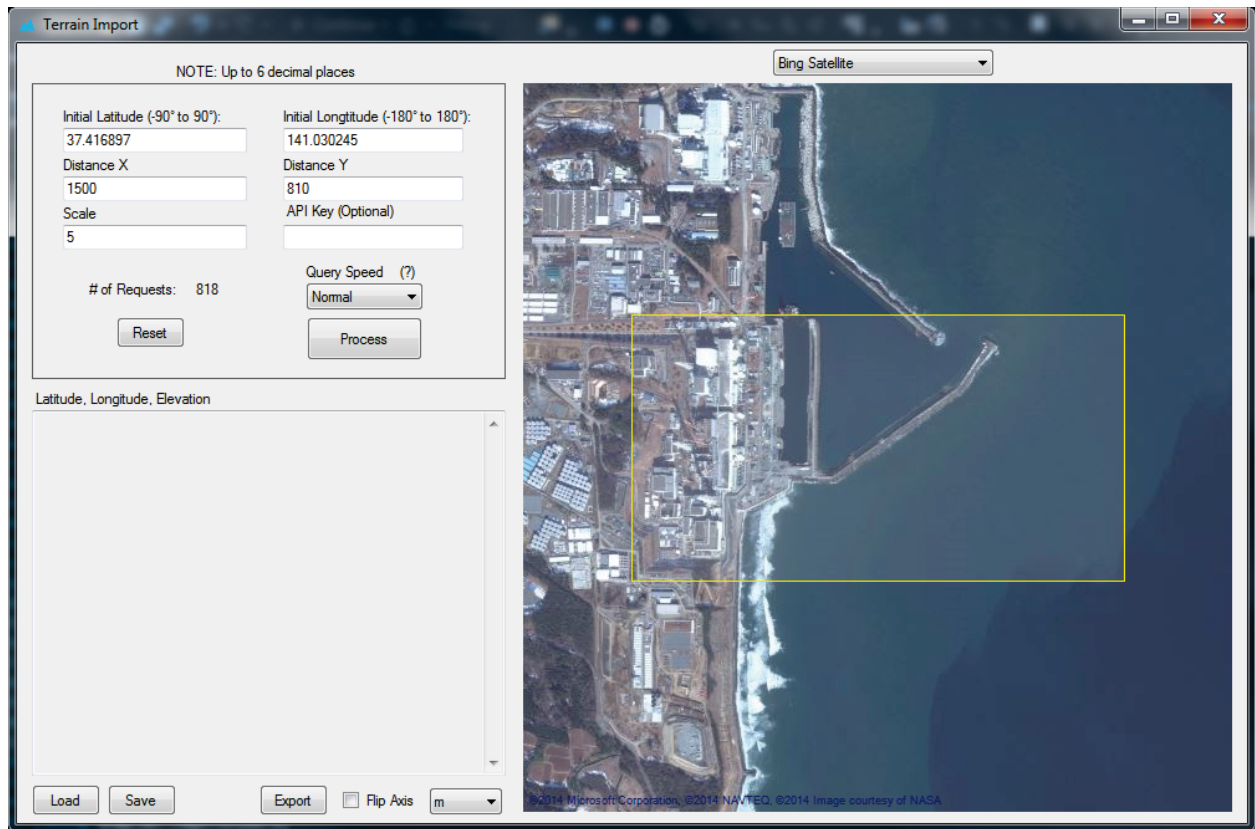


Figure 6: Terrain mapping application used to generate a 3D map of a specified area.

Using the data from the inputs, the terrain map application then calculates all the points within the rectangular area using the Haversine formula (See Appendix A) and proceeds to query the Google terrain database for elevation data for each point that was calculated. After all the queries are complete, the application is able to export all the elevation data gathered and convert it into a set of representative points. The exported file is saved as Wavefront Technologies' most common geometry interchange file format: the OBJ file format.

The OBJ file created from the facility area was imported to our facility model and used in the simulations of the facility. (See Figure 7)

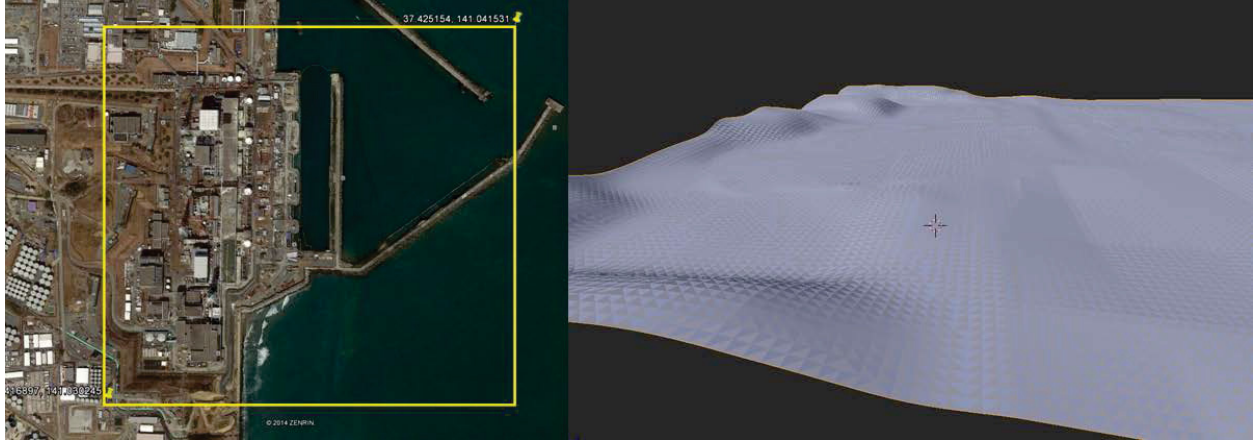


Figure 7: The 3D map generated from the Fukushima site.

### 3.3 Detailed Modeling

A large facility model gives a generalization of an event at the facility-level, but in order to have a complete scenario, we need to know what effects the event has inside critical facility buildings. To demonstrate this, a mockup of the inside of a hypothetical reactor building was modeled. It included the general structure, reactor core, a few components, and access points/stairways between levels. (See Figure 8)





Figure 8: Detailed model of reactor building.

## 4. SIMULATION DEVELOPMENT

Simulations were performed for both tsunamis of various heights and flooding into the reactor building.

### 4.1 Tsunami Simulation Setup

Initial simulations to show proof of concept were performed using the  $\frac{1}{4}$  facility slice. Initially, this model lacked accuracies such as actual terrain and wave formula representations. For the  $\frac{1}{2}$  slice of the facility, the ground terrain and ocean depth were obtained by using our custom mapping application as explained in section 3.2. Then, a boundary container was added to the model outside of the facility to represent the deeper part of the ocean. A bounding container was also modeled around the area of simulation with the measured ocean depth as its floor. The container was filled with water particles using volumetric operations and Boolean operations to remove any particles inside solid geometry. This served as a start point of the simulation (i.e., still water). A wave piston machine which acted as a collision object was setup to follow Goring's 1978 model (see Appendix A) for generating waves. This wave piston (Figure 9) was placed at the far end in the "deep ocean." An additional rotational component of the wave piston was also generated using a sinusoidal equation to represent the wave part of the water.

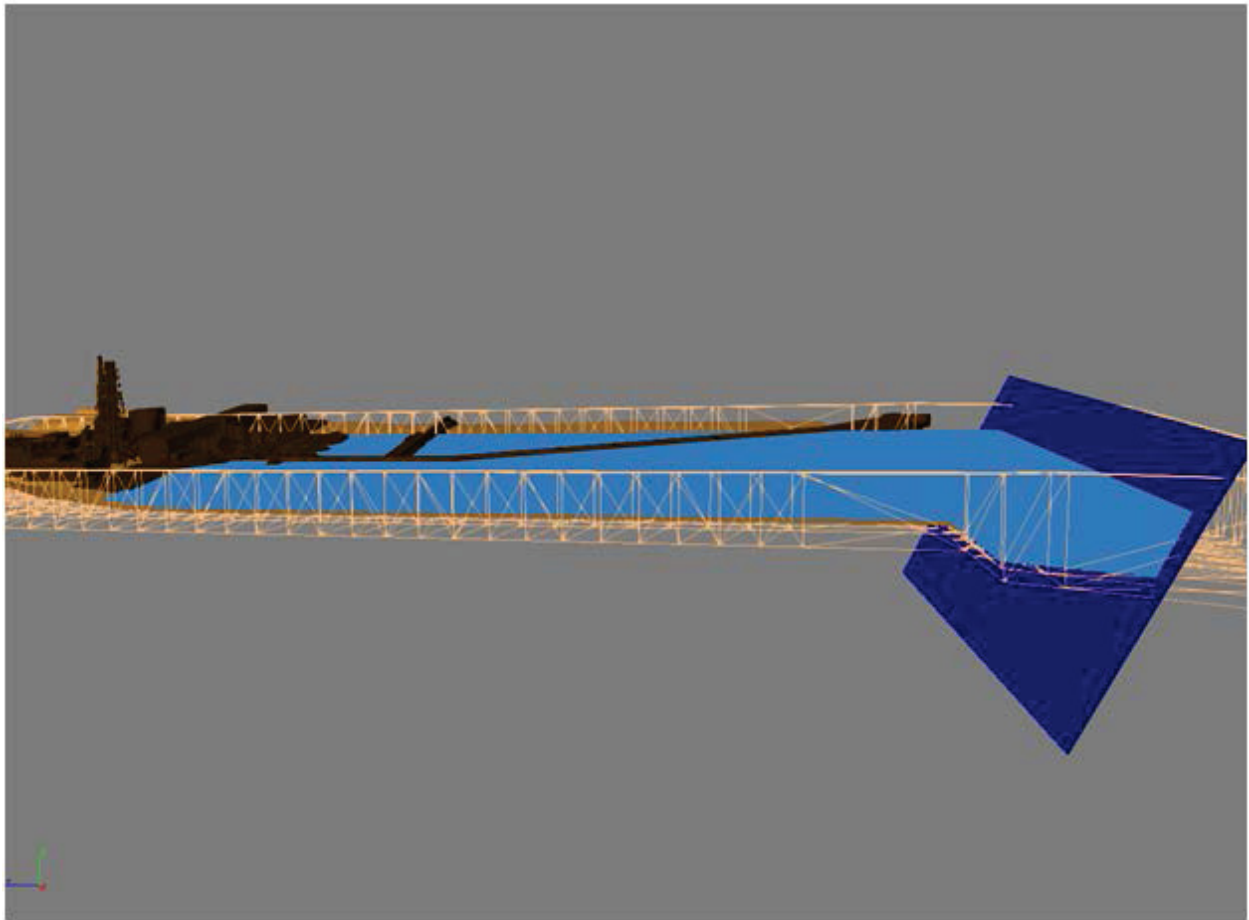


Figure 9: Neutrino Wave Piston Setup with Initial Conditions.

## 4.2 Tsunami Simulations

Initial testing was done using the  $\frac{1}{4}$  slice of the facility. Wave heights (in the ocean) from 14 to 38 meters were generated to determine impact data on the facility. These different waves were generated by varying the parameters to the wave piston. Twelve million particles at  $\frac{1}{2}$  meter resolution were used for the ocean water and tsunami wave, requiring approximately 16 GB of ram for simulation. Each frame simulated at 24fps required about 2-3 minutes on a 24 thread 12 core dual processor Xeon 2.8 GHz machine. (See Figure 10)

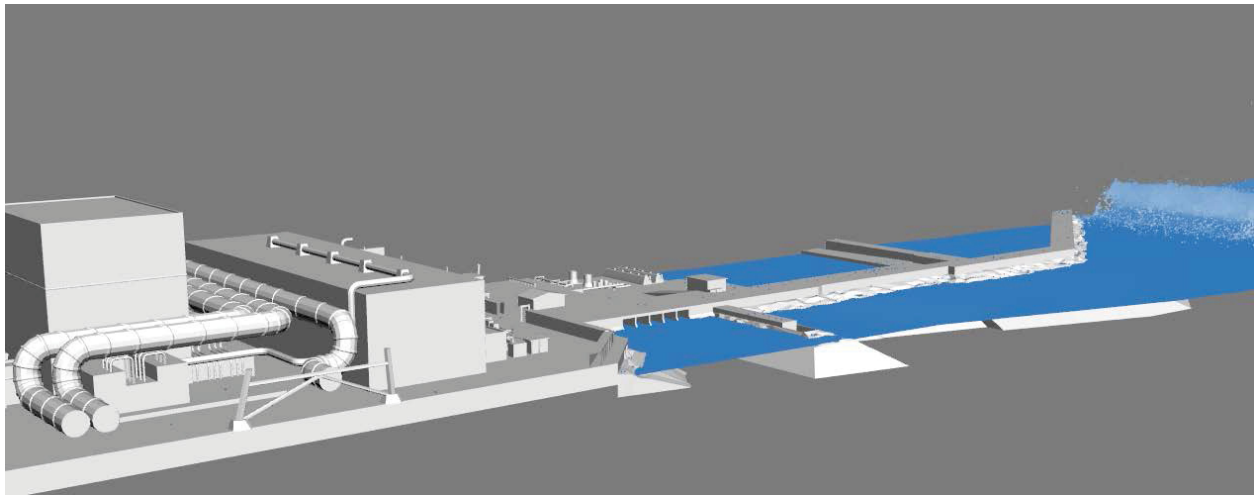


Figure 10 : Simulation of 16 meter tsunami on the  $\frac{1}{4}$  facility model

The second stage of simulations used the  $\frac{1}{2}$  the facility model. Currently it has been running at a lower resolution of 1 meter but on a much faster 40 thread, 20 core machine, requiring approximately 12 GB of memory and simulated at about 20 seconds a frame at 24fps. Two wave heights, 19 and 34 meters were generated to determine water movement into a much larger section of the facility. For each simulation, an initial run-up of about 100 frames is used to settle the particles into the initial configuration. The total simulation for about 2,000 frames took about 3 days for  $\frac{1}{4}$  slice facility at 0.5 meter resolution on the 24 thread/12 core, dual processor Xeon 2.8 Ghz machine while the simulation took about 24 hours for 2,000 frames for the  $\frac{1}{2}$  slice facility at 0.5 meter resolution on the 40 thread/20 core dual processor Xeon 2.6 GHz machine.

## 4.3 Interior Flooding Simulation

Two interior flooding scenarios of the reactor building were performed. The first consisted of flooding through the bay door into the interior of the reactor building. (See Figure 11) The second was to demonstrate water flow through the facility from a tank rupture. (See Figure 12) The first simulation was also set up to allow the flow-rate of inflow of fluid to be determined by the external facility-level flooding simulation. This coupling allowed for dynamic simulation of reactor flooding at any flow rate. Because

of the comparatively low number of particles, these internal flooding simulations took only a couple seconds per frame.

In the Houdini FX module to simulate both scenarios, two custom plugins, along with a server program, was written to test the scenarios. The server program first runs acting as the simulation command center. This is a Houdini python script which gets automatically executed during Houdini start up and keeps running in the background waiting for commands which are sent from the risk analysis simulation. Upon an initiating event simulation, the server activates the plugin which read a file describing the interested components and the actions to be taken upon the initiator. On receiving a simulation event, the server activates the simulation and sends events of interest back to the remote risk analysis program and waits for subsequent commands.

When coupling two varying time dependent messaging simulations, there were several race conditions which had to be handled. In our testing, we had to handle these in order not to miss any commands or events.

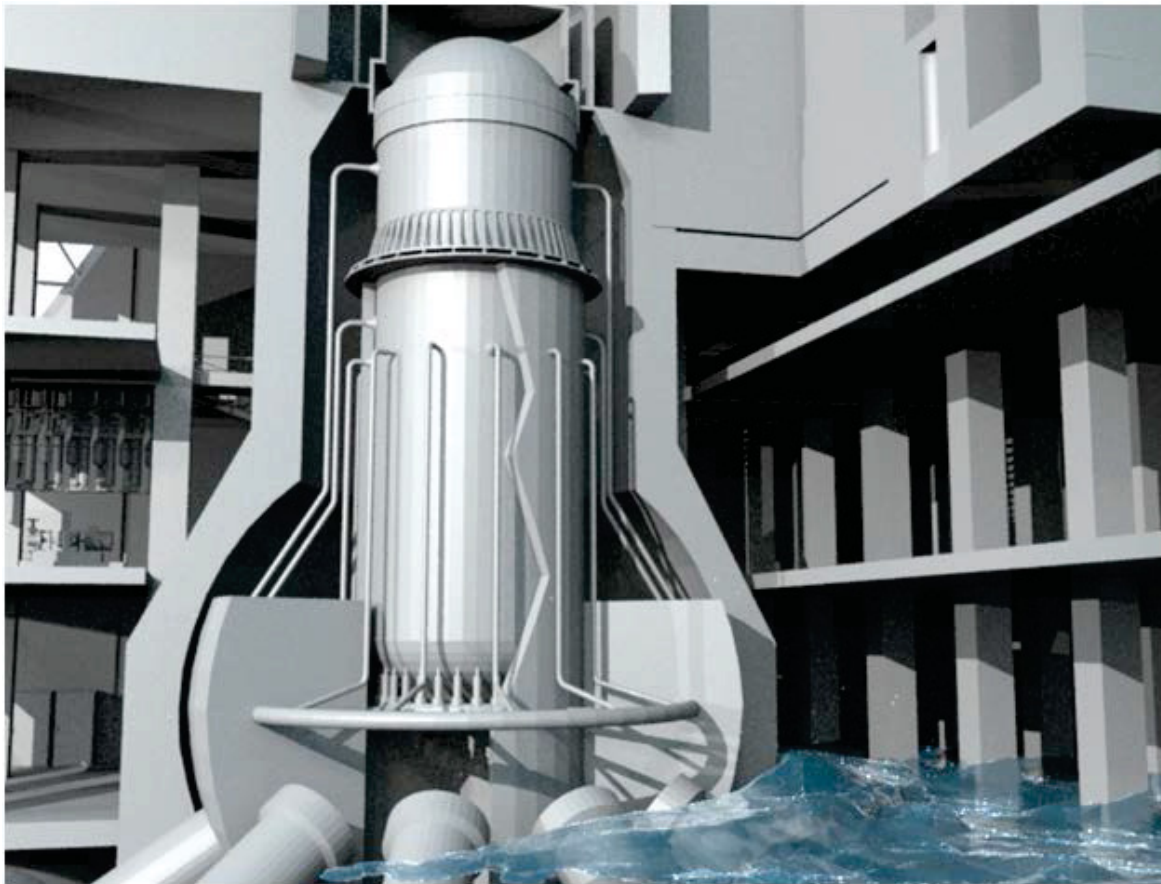


Figure 11: Reactor building flooding from the bay door (cutaway view).

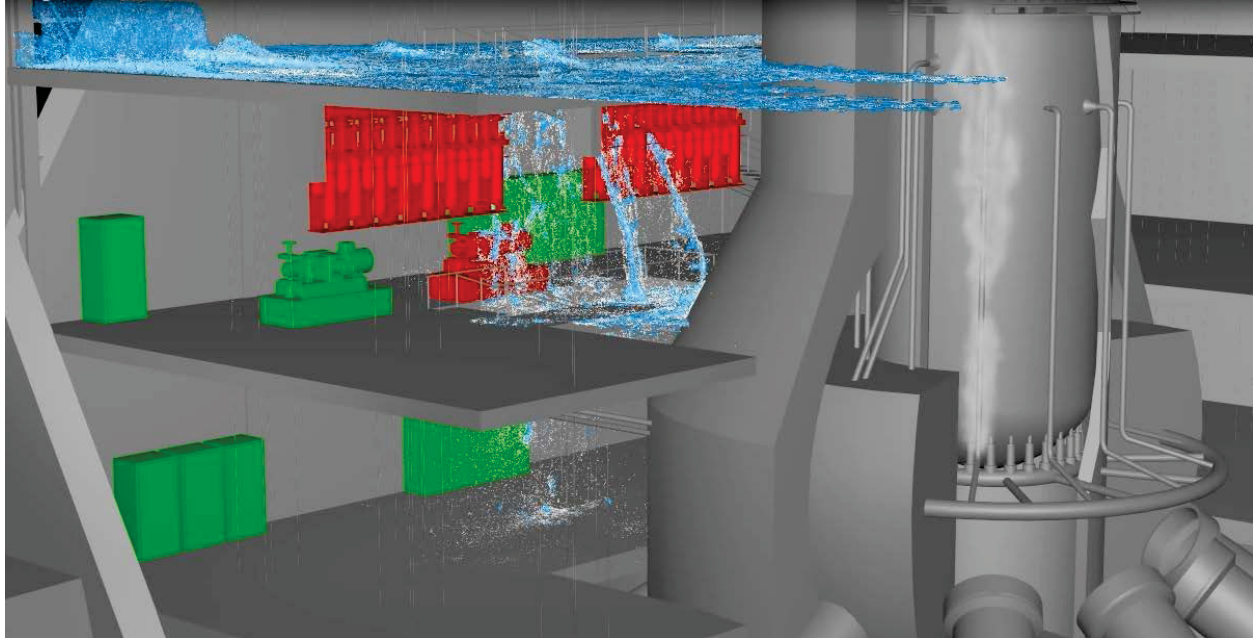


Figure 12: Water flow flooding example in a reactor building.

#### 4.4 Proposed Development Improvements

Because of shared memory requirements, these simulations were done on individual multi-core computers. Future work will investigate how to distribute the calculations in order to speed simulation results. For smaller simulations, such as inside the reactor building, a GPU processor could be used to possibly give real-time results. This option is machine dependent and limited to small simulations depending on the memory capacity of the GPU. The second option is a dedicated co-processor which can add hundreds of additional cores to an existing machine. The principle is similar to the GPU but makes memory management much simpler because of the similar architecture to the main core and the code remains the same even if there is no co-processor on the machine. Lastly, distributed computing could be implemented to take advantage of clusters of networked computers. These High Performance Computing (HPC) systems could possibly run large simulations in close to real time (depending on the size of the system). However, this requires modifications to 3D engine we are using in order to insure they work in unison with an effective memory management system.



## 5. CONCLUSIONS

Computers have been used for 3D modeling and simulation, but only recently have computational resources been able to give realistic results in a reasonable time frame for large complex models. In this report, we described the methods, techniques, and resources which are being developed at the INL to support a 3D modeling engine used to represent risk analysis simulation for advanced small modular reactor structures and components. This capability is important since explicit, scenario-based analysis of plant safety will play a key role in licensing of small modular reactors, and significant emphasis will be placed on selection of events to be analyzed. Consequently, the need for extensive analysis, and representative models, supporting the safety case will be paramount.

During FY14, the 3D modeling engine development focused on four aspects:

1. Providing the creation of external flooding scenarios by representing an initiating event (potential flooding event entering a coastal plant site) through an entire risk analysis scenario.
2. Providing mechanistic calculations specific for flood representation where we “generate” the simulated flood occurring in the ocean, entering the coastal site boundary, and tracing the water as it enters and moves through the site buildings.
3. Providing flooding-caused failures for components that are inside buildings that may be impacted as seawater enters building through doors or other penetrations.
4. Providing an overall spatial representation of a flooding hazard by depicting how a tsunami might impact various buildings and components that are distributed at different locations at a hypothetical plant site.

The ability to represent hazards such as floods is integral to having a robust PRA approach for aSMRs for a variety of different siting conditions.

## 6. REFERENCES

- [1] [http://en.wikipedia.org/wiki/Smoothed-particle\\_hydrodynamics](http://en.wikipedia.org/wiki/Smoothed-particle_hydrodynamics)
- [2] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler and M. Teschner, "Versatile Rigid-Fluid Coupling for Incompressible SPH," *ACM Transactions on Graphics (Proc. SIGGRAPH 2012)*, vol. 31, no. 4, 2012.

# APPENDIX A

## A.1 Google Elevation API

Google's Elevation API places a limit on the number of possible query requests. Normally, a free user has up to request up to 2500 points per day. The API also tries to limit automated requests, so in order for the terrain map generator not to be blocked, time pauses were also added between requests. Optionally, Google provides an advanced subscription that can be to purchase which enables more requests. A terrain portion is determined by:

Haversine formula:

$$a = \sin^2(\Delta\phi/2) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Where  $\phi$  is latitude,  $\lambda$  is longitude,  $R$  is earth's radius (mean radius = 6,371km).

Note that angles need to be in radians.

[<http://www.movable-type.co.uk/scripts/latlong.html>]

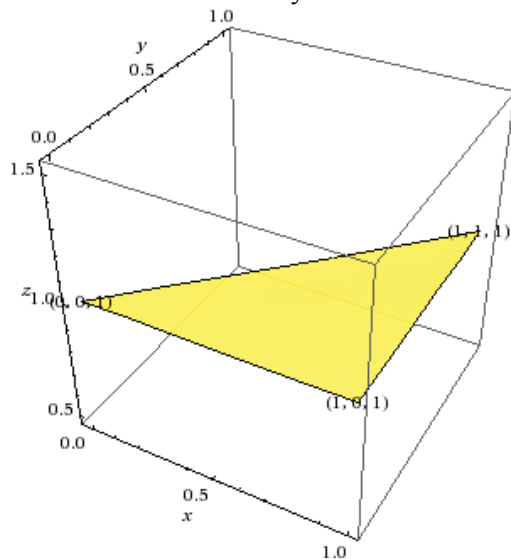
The OBJ file format is generated first by defining points in a parameter space of curve or surface.

As an example, if “v 5 15 34.483” were to be generated, it would mean that a vertex (v for vertex) at rectangular point coordinate (5, 15) with z-axis (elevation) 34.483.

Secondly, the faces are generated to produce the texture from the vertices. Suppose the following were to be generated in the OBJ file:

```
v 0 0 1
v 1 0 1
v 1 1 1
v 0 1 1
f 1 2 3
```

This would generate a triangle connecting the three vertices numbered 1, 2, and 3. The number of the vertices is determined by their order from the top of the list going down.



(Visual Representation generated by WolframAlpha)



However the faces generated from the applet is generated as a rectangle and that only requires that the face generated have one additional vector number at the end.

## A.2 Goring Solitary Wave Generation – Numerical Model

Goring (1978) proposed a model for the purpose of laboratory solitary wave generation. The surface profile (x,t) of a solitary wave can be described using the following equation:

$$\eta(x, t) = H \operatorname{sech}^2[\kappa(Ct - X_0)] \quad (\text{A-1})$$

$$C = \sqrt{g(H + h)} \quad (\text{A-2})$$

$$\kappa = \sqrt{\frac{3H}{4h^3}} \quad (\text{A-3})$$

Where  $C$  is the wave celerity or phase velocity,  $X_0$  is the wave displacement,  $H$  is the wave height and  $h$  is the depth of the ocean. Applying equation A-1 to the wave maker piston results in

$$X_0(t) = \frac{H}{\kappa h} (\tanh(\kappa(Ct - X_0))) \quad (\text{A-4})$$

Using this equation one can solve for the wave piston displacement and wave piston duration using newton iterations resulting in

$$S = \sqrt{\frac{16Hh}{3}} \quad \text{and} \quad t_f = \frac{2(3.80 + \frac{H}{h})}{\kappa C}$$

Where  $S$  is the displacement and  $t_f$  is the time taken for the displacement.

Goring, D. G. (1978). Tsunamis – The Propagation of Long Waves Onto a Shelf. Doctoral Dissertation, Report No. KH-R-38, Keck Laboratory of Hydraulics and Water Resources, California Institute of Technology, Pasadena, California.