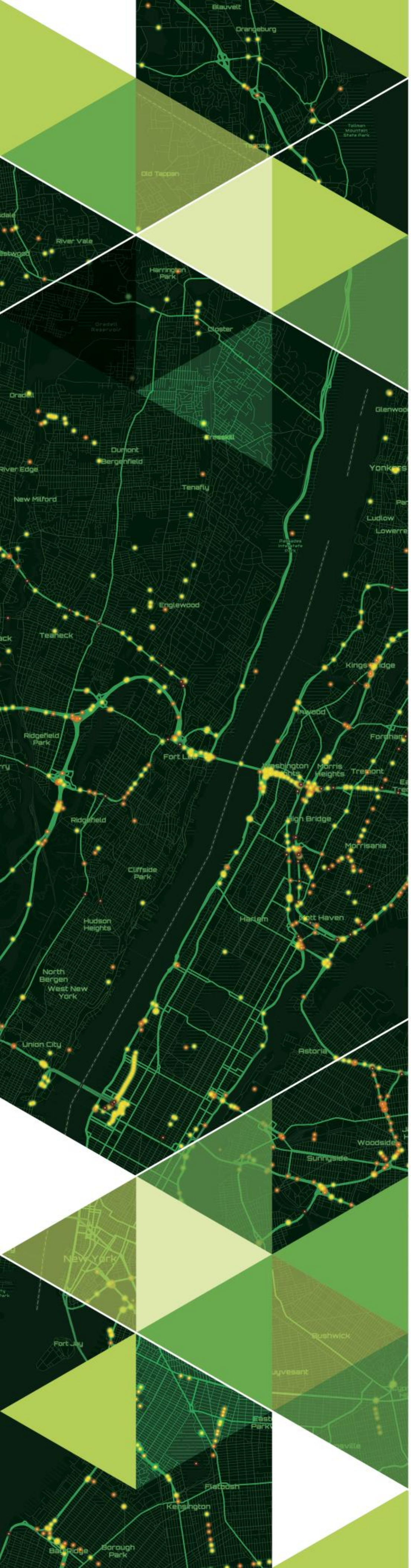


AN ESRI
TECHNICAL PAPER
July 2024

Flood Simulation in ArcGIS Pro 3.3

380 New York Street
Redlands, California 92373-8100 USA
909 793 2853
info@esri.com
esri.com

THE SCIENCE OF WHERE®

The information contained in this document is subject to change without notice.

Abstract

Flood events pose threats to human life and property. To effectively mitigate these disasters there is a need for a range of flood simulation tools. These range from engineering tools based on highly detailed physics models for hydraulic and civil engineering professionals to more rapid and approximate simulation tools for a wider audience looking for initial insight. In this paper we describe a new GPU based rapid flood simulation tool in the second category, that is integrated into GIS and runs on the same off-the-shelf modern GPU based hardware as the GIS. This flood simulation tool is designed to support rapid simulation and initial exploration but does not handle all the conditions and flow regimes that can occur in the real world. It is intended to complement and not replace existing engineering tools and models. It can be used by GIS users to obtain previews of water flows for the supported subset of conditions, for initial geo-design and exploration of design alternatives, and as a useful way to explore and validate data quality. The paper describes the new rapid simulation tool and its underlying model in more detail following which it describes the situations and conditions for which it can provide useful insights as well as situations and conditions that it does not model and cannot provide insight for.

1. Introduction

Floods are a topic of research in the field of hydrology and hydraulics and significantly impact many human activities including agriculture, civil engineering, and public health. Flood events pose threats to human life and property. To effectively mitigate these disasters there is a need for a range of flood simulation tools. At one end are engineering tools that model the detailed physics of the problem and are for use by civil and hydraulic engineering professionals. These tools run on specialized computational infrastructure outside the GIS and can exchange data and results with GIS. At the other end are simulation tools that are integrated into GIS and provide more approximate solutions while being accessible to a wider range of users. In this paper we describe a new GPU based rapid flood simulation tool in the second category that is integrated into GIS and runs on the same off-the-shelf modern GPU based hardware as the GIS.

The new rapid flood simulation tool uses an approximate physics-based model that makes tradeoffs between accuracy and speed so that the highly parallelized calculations can be performed on GPUs at interactive frame rates. The water flow model used by this new tool builds on approximate physics based models for simulating water flow and its impact on terrain that have been developed by researchers in the field of Computer Graphics for use in Visualization and Animation applications [1] [2] [3] [4]. Some of these solutions are implemented for popular game engines [5] [6]. In examining and integrating these models into GIS for use in a rapid simulation and exploration context we have analyzed the physics models and their underlying assumptions for applicability, and we provide guidelines for their use in the GIS context.

The new rapid flood simulation tool with its associated water flow model is designed to support rapid simulation and initial exploration for a subset of the flow conditions that can occur in the real world and provides approximate solutions for those conditions. It does not handle more complex conditions and flow regimes that can occur in the real world depending on the event and environment, sections 4 and 5 of this paper provide additional details. This tool is not intended to replace existing engineering tools and models and should not be used in place of them. It is intended for use by GIS users to obtain previews of water flows for the supported subset of conditions, for initial geo-design and exploration of design alternatives and as a useful way to explore and validate data quality.

The remainder of the paper describes the new dynamic flood simulation model in more detail following which it describes the situations and conditions for which it can provide useful insights as well as the situations and conditions that it does not model and cannot provide insight for.

2. Flow Simulation Model

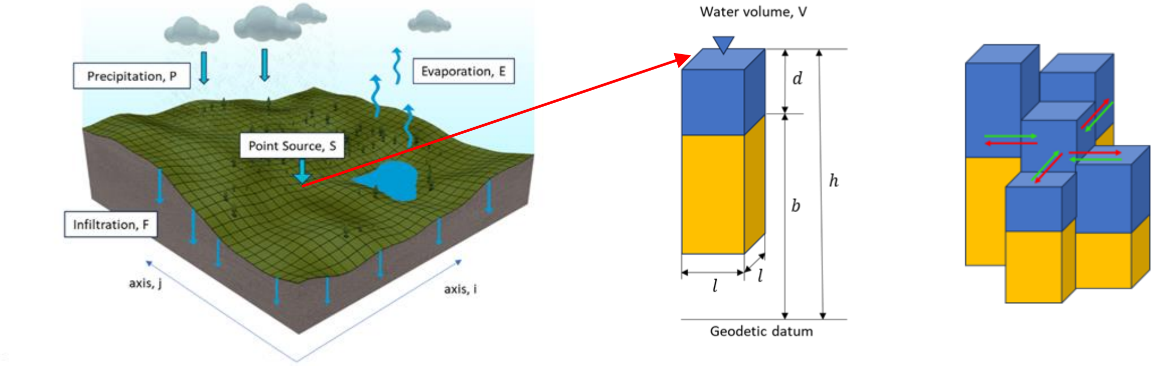
The movement of water through the landscape is described by two equations: a continuity equation and a momentum equation. The continuity equation is based on the principle of conservation of mass – water is neither created nor destroyed during the flow process. The momentum equation describes the flow dynamics – the effect of physical forces on the motion of the fluid. If a computational mesh of square cells is drawn over the landscape, such as by using a digital elevation of the land surface topography to provide the cell mesh, there are thousands of equations to be solved simultaneously to determine the volume of water in each cell and the flow of water between the cells. This task has some similarities to the requirements for computer animation using Graphics Processing Units (GPUs), which uses massive parallelization to solve large sets of equations simultaneously.

The solution of the continuity and momentum equations for shallow water flow over the land surface and in streams is accomplished by hydraulic routing models. Such models can be either one-dimensional (1-D) or two-dimensional (2-D). One-dimensional routing models describe flow through a river network where the only distance measure is length along the stream reaches being modeled. Two-dimensional models describe the motion of the water in the (x, y) coordinate directions across the landscape and treat flow in the stream channel as being part of the broader flow pattern. Hydraulic routing models account for pressure, friction, gravity and other forces, and their effect on the motion of the fluid in space and time. Fully accounting for all these components is a task of great complexity and involves long computation times.

The ArcGIS Flow Simulation model is a simplified 2-D hydraulic routing model designed to approximately represent flow in large, inundated areas, such as occur within a floodplain during a flood event.

Water Volume

Suppose that a computational mesh is drawn over the landscape as shown in Figure 1a. For a typical cell in the landscape, as shown in Figure 1b, the height of the land surface above geodetic datum is the land surface elevation, b, the water volume within the cell on the land surface has depth, d, and its water surface elevation is labelled as h.



(a) Computational mesh (b) Cell (i, j) (c) Group of five cells

Figure 1. Flow of water in a computational cell

It follows that the water surface elevation is the sum of the land surface elevation and the water depth, so

$$h_{ij} = d_{ij} + b_{ij} \qquad \textit{Eq. 1}$$

If the cells are square with length l, then the cell area is l^2, and the volume of water, V, in the cell is the product of the cell area and the water depth, hence

$$V = l^2 d \qquad \textit{Eq. 2}$$

The flow simulation is carried for a time period which is made up of small-time intervals of duration, Δt. At time, t, the volume of water in the cell is V_t and at time $t + \Delta t$, the corresponding volume of water is $V_{t+\Delta t}$, so the change in volume within the cell ΔV during this interval is

$$\begin{aligned} \Delta V &= V_{t+\Delta t} - V_t \\ &= l^2(d_{t+\Delta t} - d_t) \\ &= l^2 \Delta d \end{aligned} \qquad \textit{Eq. 3}$$

which just states mathematically that the change in volume of water over the time interval is the product of the cell area and the change in depth. Since the elevation of the land surface, b, is fixed, it follows from Eq. 1 that the change in volume over this time interval can also be written in terms of the change in water surface elevation, h, as

$$\Delta V = l^2 \Delta h \qquad \textit{Eq. 4}$$

Continuity Equation

The continuity equation states that over the time interval, Δt, the change in water stored within the cell is equal to the difference between the water inflow and outflow. As shown in **Error! Reference source not found.**, precipitation, P, falls on the cell, and infiltration into soil, F, absorbs water out of the cell. These are usually measured in dimensions of [L/T] such as inches per hour or mm/hr. There are also flows, Q, between the cells measured in dimensions of [L^3/T], such as cubic feet per second or liters per second. It follows that the continuity equation for a single cell can be written as

$$\frac{\Delta V}{\Delta t} = Inflow - Outflow$$

$$= Pl^2 + Sl^2 - Fl^2 - El^2 - \sum Q \qquad \textit{Eq. 5}$$

where $\sum Q$ represents the net outflow of water between the given cell and its four adjacent cells along the coordinate axes. It is a convention in fluid mechanics that outflows are always considered to be positive, so if water flows into the given cell from one of its neighbors, this is considered as a negative flow. Combining Eq. 4 and Eq. 5, leads to

$$l^2 \frac{\Delta h}{\Delta t} = Pl^2 + Sl^2 - Fl^2 - El^2 - \sum Q \qquad \textit{Eq. 6}$$

$$\frac{\Delta h}{\Delta t} = P + S - F - E - \frac{1}{l^2} \sum Q$$

which is a continuity equation relating the change in water surface elevation within the cell to the precipitation, infiltration, evaporation, sources and flows from adjacent cells that impact it. The values of precipitation, P, infiltration, F, evaporation, E, and sources, S, are supplied externally. The values of h and $\sum Q$ have to be determined as part of the simulation model solution.

Momentum Equation

As described earlier, the flows between adjacent cells are always considered positive when water leaves the cell, as shown in Figure 2. Hence,

$$\sum Q = Q^L + Q^T + Q^R + Q^B \qquad \textit{Eq. 7}$$

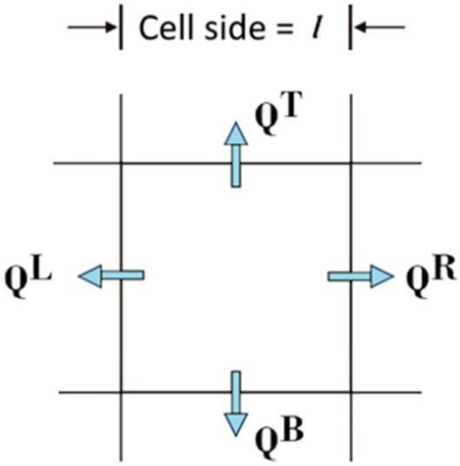


Figure 2. Outflow directions for a given cell (i, j)

The momentum equation for one-dimensional open channel flow can be written (with eddy losses, wind shear effect, and lateral inflow neglected and assuming Boussinesq coefficient β = 1.0) as

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\left(\frac{\partial h}{\partial x} + S_f\right) = 0 \qquad \textit{Eq. 8}$$

where Q is the discharge in the open channel, x is the distance along the channel, A is the cross-sectional area, g is acceleration due to gravity, h is the water surface elevation, and S_f is the friction slope, which is the energy loss per unit length of channel due to shear forces on the bed and banks of the channel [7]. This is an expression of Newton's Second Law of Motion applied to fluid flow in which the first two terms represent acceleration of the water in time (local acceleration) and space (convective acceleration), the third term accounts for the forces of pressure and gravity, and the final term represents friction forces. If the convective acceleration and friction forces are neglected, Eq. 8 reduces to:

$$\frac{\partial Q}{\partial t} + gA\left(\frac{\partial h}{\partial x}\right) = 0 \qquad \textit{Eq. 9}$$

Suppose there are two adjacent cells, labelled cell (i, j) and cell (m, n). Their water surface elevations $h_{i,j}$ and $h_{m,n}$ apply at the cell centers, which are a distance l apart, and the water is flowing outward from cell (i, j) to cell (m, n). Then

$$\frac{\partial h}{\partial x} = \frac{h_{m,n} - h_{i,j}}{l} \qquad \textit{Eq. 10}$$

and applying Eq. 9 in discrete time

$$\frac{\partial Q}{\partial t} = \frac{\Delta Q}{\Delta t} \qquad \textit{Eq. 11}$$

Substituting Eq. 10 and Eq. 11 into Eq. 9 yields

$$\frac{\Delta Q}{\Delta t} + gA\left(\frac{h_{m,n} - h_{i,j}}{l}\right) = 0 \qquad \textit{Eq. 12}$$

If this equation is applied to flow between two adjacent cells and if the cross-sectional area is approximated by $A = l^2$, then Eq. (12) can be rewritten as:

$$\Delta Q = \Delta t\ g\ l\ (h_{i,j} - h_{m,n}) \qquad \textit{Eq. 13}$$

The computational process uses the values of h at time t in Eq. 13 to get the change in discharge ΔQ in the interval Δt between time t and time $t + \Delta t$, in each of the four coordinate directions. These ΔQ values are added to the four coordinate discharges at time t, Q_t, to get the corresponding values at time $t + \Delta t$, $Q_{t+\Delta t}$, and those values are used in Eq. 6 to get $\sum Q$.

To conserve the water mass, two corrections are made:

- Negative flow rates cannot occur.
- Water flow from a cell cannot be greater than the volume of the water in that cell.

In the first correction, the flow in a direction D is set to zero if it's negative with

$$Q^D_{ij,t+\Delta t} = max\left(0, Q^D_{ij,t} + \Delta t\ l\ g\ \Delta h^D_{ij}\right) \qquad \textit{Eq. 14}$$

where D is one of the directions L (left), R (right), T (top) or B (bottom) and Δh^D_{ij} is the water depth difference with the column in direction D. The second correction is for the case when the outgoing water exceeds the amount of water in that cell. In that case, the flow rate is scaled down by a factor K so that no water is lost. The scaling factor is calculated by:

$$\mathrm{K} = min\left(1, \frac{V_{ij}}{Q\ \Delta t}\right) \qquad \textit{Eq. 15}$$

where V is the water volume in the cell and Q is the total flow rate in four directions. Then K becomes:

$$\mathrm{K} = min\left(1,\ d_{ij}\ \frac{l^2}{\Delta t(\ Q^L\ +\ Q^R\ +\ Q^T\ +\ Q^B)}\right) \qquad \textit{Eq. 16}$$

Adding in the precipitation, P, infiltration, F, evaporation, E, and sources, S, during this time interval Δt, the corresponding change in water surface elevation Δh in cell (i, j) is computed by Eq. 6. This process is repeated for all cells and for all time intervals to get the distribution of the discharge and water surface elevation in space and time.

Δt must be a very small number to prevent the simulation from becoming unstable. To speed up the simulation progress, a different time step ΔT was introduced which represents the "actual" time taken for each iteration. Δt can be interpreted as an internal time step that is used for calculations, whereas ΔT is the external time step which represents the time elapsed for an iteration. To normalize the water speed across different cell sizes, the external time step ΔT is scaled linearly with respect to l with the equation:

$$\Delta T = 0.5l \qquad \textit{Eq. 17}$$

Internal time step Δt is also calculated as a function of l to ensure stability with

$$\Delta t = \sqrt{l + 0.1}\ * 0.01416 \qquad \textit{Eq. 18}$$

In the derivation of the continuity and momentum equations, the following key assumptions were made:

1. Water is incompressible and single-phase media.
2. Uniform velocity distribution across cell.
3. In derivation of momentum equation, we neglected:
   a. Eddy losses.
   b. Wind shear effect.
   c. Lateral inflow.
   d. Convective acceleration.
   e. Friction forces.
4. Depth of water is similar to cell width/length.

3. ArcGIS Pro Implementation and Usage

The approximations in the water flow model help to make the equations for each cell independent from each other so we can take advantage of GPU parallelism. The GPU implementation works on 2-dimensional textures which are arrays of pixels similar to rasters and the textures are stored on the GPU.

Significant initialization and configuration of the scenario is required before the iterations begin, and both visualization and caching capabilities are included to review the results after the calculations are run. The values can also be exported out to a set of data slices through time to support additional workflows, such as creating maps, reports and service data.

Figure 3 presents the overview of the flood simulation process and workflow.

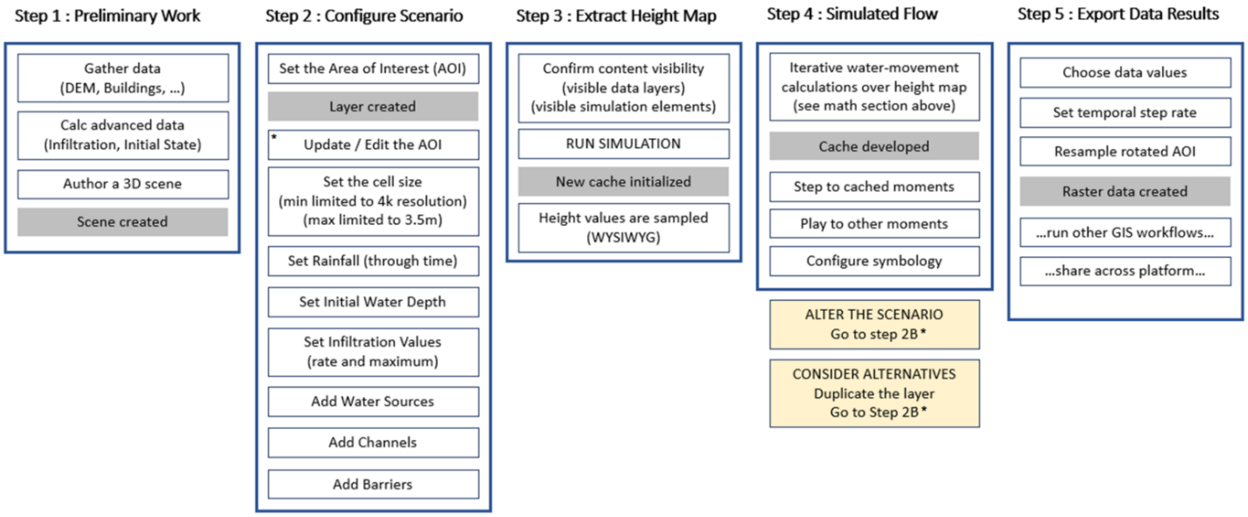

Step 1 : Preliminary Work
Gather data (DEM, Buildings, …)
Calc advanced data (Infiltration, Initial State)
Author a 3D scene
Scene created
Step 2 : Configure Scenario
Set the Area of Interest (AOI)
Layer created
* Update / Edit the AOI
Set the cell size (min limited to 4k resolution) (max limited to 3.5m)
Set Rainfall (through time)
Set Initial Water Depth
Set Infiltration Values (rate and maximum)
Add Water Sources
Add Channels
Add Barriers
Step 3 : Extract Height Map
Confirm content visibility (visible data layers) (visible simulation elements)
RUN SIMULATION
New cache initialized
Height values are sampled (WYSIWYG)
Step 4 : Simulated Flow
Iterative water-movement calculations over height map (see math section above)
Cache developed
Step to cached moments
Play to other moments
Configure symbology
ALTER THE SCENARIO Go to step 2B*
CONSIDER ALTERNATIVES Duplicate the layer Go to Step 2B*
Step 5 : Export Data Results
Choose data values
Set temporal step rate
Resample rotated AOI
Raster data created
…run other GIS workflows…
…share across platform…

Figure 3. Overview of the entire flood simulation process

Step 1: Preliminary work

The simulation is run by generating a height map (a texture containing elevation values) derived from sampling the orthographically rendered 3D view. This means that *anything rendered in the view* – the ground elevation surface, 3D database objects (such as buildings), symbolized features (such as collapsed bridges), and other elements (such as barrier walls) – will be captured in the 2.5D surface and used as the surface upon which water flow will be modelled. That is, the processing is run in a what-you-see-is-what-you-get (WYSIWYG) manner.

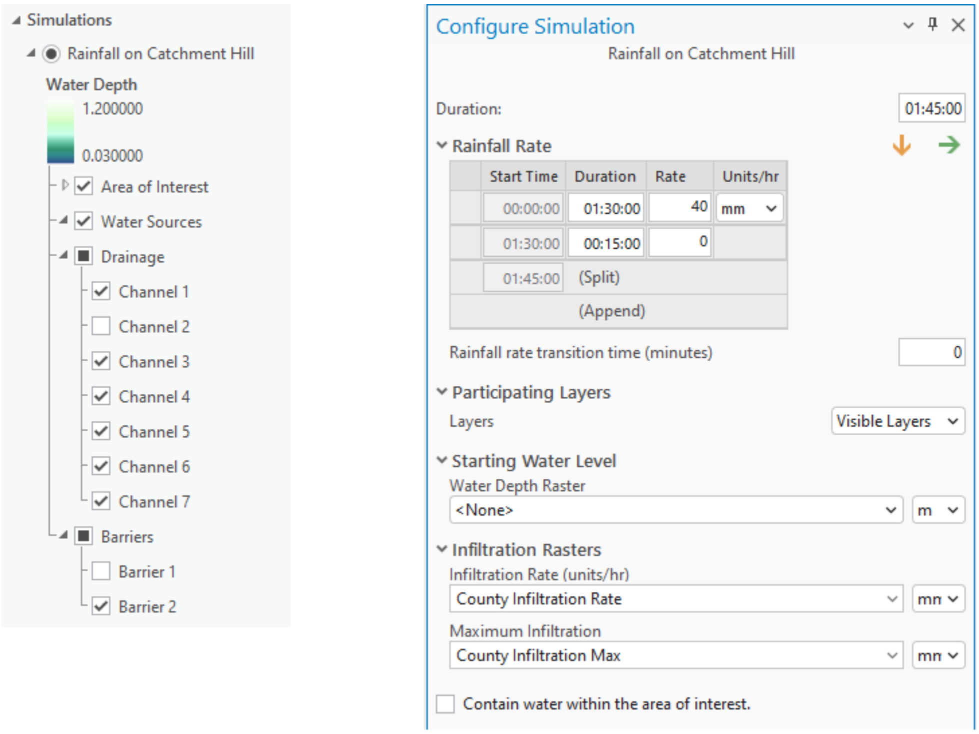
This approach makes it simple to simulate alternate scenarios that, for example, include/exclude new construction, make changes in the ground surface, and test out mitigation alternatives using barriers. It also means that every element drawn in the scene needs to be included *on purpose* and meet your analysis accuracy requirements. For example, using a single vertically extruded bounding box for a city block, instead of a collection of individually oriented 3D models, could incorrectly redirect water flow, and symbolized features like 3D tree symbols will act as large barriers that can block water flow in unexpected ways.

The main preliminary steps are:

a. Gather data – create, import, or reference a digital elevation model (DEM) that includes the topography of the ground surface, plus any additional 3D vector data that will impact water flow, such as buildings. You can also gather data for alternate future states, such as the updated topography, street network and buildings for a proposed subdivision.
b. Calculate advanced data – create, import, or reference additional data that can help describe a flooding scenario, including potential initial-water depth rasters (e.g., the 100-year flood state generated in another flood modeling package) and infiltration rates and maximum infiltration values (e.g., calculated from a combination of land use and soil type).
c. Author a 3D scene – use the DEM to define the ground surface, and add in other 3D vector data, like buildings and dam walls, to define the 3D world within which the flood simulation will be run. The data layers can represent the current, future, or historical state of the world.

Step 2: Configure Scenario

A flood simulation scenario runs within an area-of-interest (AOI), and this rectangular extent, plus at least one source of water, is the only required parameter for every simulation. There are several additional (optional) configuration elements, such as rainfall, water sources and barriers, that can be used in a variety of combinations. All the properties are stored in the simulation layer, with optional references to raster layers inside the scene and editable elements listed in the Contents pane (Figure 4).



Simulations
- ● Rainfall on Catchment Hill
  - Water Depth
    - 1.200000
    - 0.030000
  - ☑ Area of Interest
  - ☑ Water Sources
  - ■ Drainage
    - ☑ Channel 1
    - □ Channel 2
    - ☑ Channel 3
    - ☑ Channel 4
    - ☑ Channel 5
    - ☑ Channel 6
    - ☑ Channel 7
  - ■ Barriers
    - □ Barrier 1
    - ☑ Barrier 2

Configure Simulation

Rainfall on Catchment Hill

Duration: 01:45:00

Rainfall Rate

Start Time	Duration	Rate	Units/hr
00:00:00	01:30:00	40	mm
01:30:00	00:15:00	0	
01:45:00	(Split)		
	(Append)		

Rainfall rate transition time (minutes): 0

Participating Layers

Layers: Visible Layers

Starting Water Level

Water Depth Raster: <None> m

Infiltration Rasters

Infiltration Rate (units/hr): County Infiltration Rate mn

Maximum Infiltration: County Infiltration Max mn

□ Contain water within the area of interest.

Figure 4. A simulation layer can contain elements for water sources, drainage channels, and barriers (left), as well as configuration properties for rainfall rates and referenced raster layers for initial water depth and infiltration properties (right).

The main steps are:

a. Set the AOI – define the area within which the simulation will run. This can be digitized in manually or based on another property of the map, such as the extent of a selected feature. Once the AOI has been accepted, a layer is added into the scene to hold all the other potential properties for the scenario.
   a. The boundary of the AOI can also be configured to allow water to exit – such as for a mountainous flash flood event – or to be contained – such as for ocean water rise.

b. Update / Edit the AOI – the extent of the simulation can be moved, resized, and rotated, as needed. This will not impact the other configuration properties of the scenario.

c. Set the cell size – this is the physical size of each cell in the simulation. The scene will be sampled from vector objects into this resolution when running the scenario. The minimum cell size is a factor of the physical extent of the AOI – that is, 1/4096 of the longest side. For example, an AOI that is 2km by 1.5km would have a minimum allowed cell size of ~0.5m (i.e., 2000m / 4096). This value can be changed to a value *larger* than the minimum, such as 1m or 2m, with the maximum physical cell size locked at 3.5m (Cell values greater than 3.5m lead to larger values of Δt and cause instability).

d. Set rainfall through time – set the rate of rainfall being applied to the AOI, such as 2 inches per hour. The rate can change over time, with the option to linearly move between a set of fixed rates over a specified transition time. The rainfall rate is applied equally over the AOI. Rainfall rates are included in the simulated water flow modeling in Step 4.

e. Set evaporation – the rate of water "lost" to evaporation during the simulation. The value should be set to a rate per hour using the same units as the rainfall rate. It is applied at a constant rate across the area of interest for the duration of the simulation. Evaporation is included in the simulated water flow modeling in Step 4.

f. Set initial water depth – add water into the simulation as an initial step before allowing it to move. Water depth values must be defined in a raster layer where the cell values are in a depth unit, such as meters or feet. For example, start a scenario already in the 100-year flooding state and then make a change, such as adding more rainfall or blocking a river, to simulate what happens next. Initial water depth is included in the first phase of simulated water flow modeling in Step 4.

g. Set infiltration values – define how fast water infiltrates into the ground (which removes that water from the simulation) and when that infiltration stops. Use a raster layer to vary these values over the AOI (such as in an urban environment with roads and grass parks), or use a fixed rate for the whole area (such as in a rural or mountainous region). Varying infiltration values must be defined in a raster layer where the cell values are in a flow rate unit, such as mm/hr. Similarly, using a raster layer to set varying maximum infiltration rates over the AOI will need to use values in a depth unit, such as millimeters or inches. Infiltration values are included in the simulated water flow modeling in Step 4.

h. Add water sources – define locations and flow rates for adding water into the simulation to model options such as upstream water flow entering the AOI, burst pipes, and venting dams. The volume of water is added into the simulation by being averaged over the cells that are within 5m of the water source point. This is done to reduce the effect of "mounding" at the input location, where water is being added faster than it can move away. For high incoming water flow, consider modeling it across multiple water source locations. Water sources are included in the simulated water flow modeling in Step 4.

i. Add channels – enter in two-point lines, with a width value, to add channels into the ground surface without having to edit it. This will allow water movement past undesired blockages in the terrain – such as low bridges that cross streams or drainage canals – without having to edit the DEM surface. Note that water no longer flows along the bridge but will also fall into the channel. The channel effectively updates the captured elevation values from the scene with a linear progression of elevation values between the start and end points of the line. Channels are incorporated into the height map in Step 3.

j. Add barriers – add in potential contingency barriers by digitizing in a path and setting the width and height of the barrier. Each vertex of the path will be placed on the ground and the intermediate sections will be connected linearly. To have the barrier more closely fit the ground surface, digitize in more vertices. The barrier will be captured in the height map in Step 3.

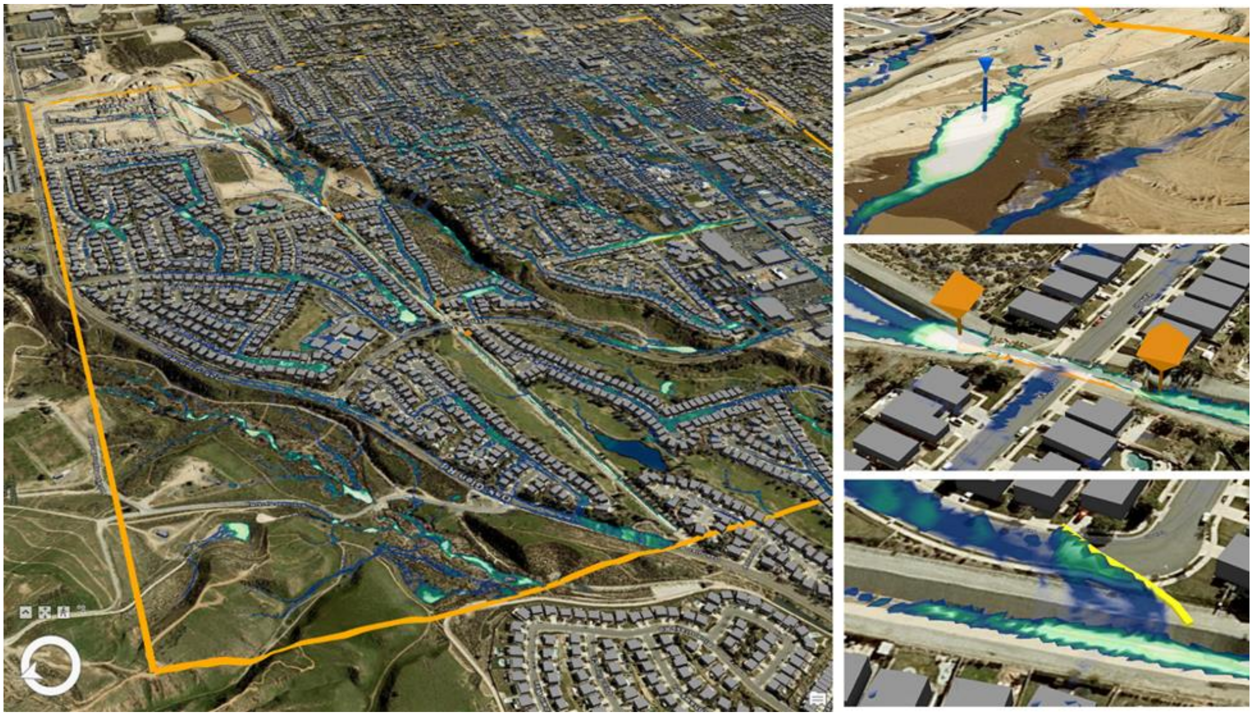


Figure 5. Rainfall simulation for an area of interest, with water source points, channels, and barriers

Step 3: Extract Height Map

When the simulation is run, the 3D view is converted into a height map by sampling the area of interest at the spacing defined by the cell size. For example, an area of interest that is 2km by 2km in size and is using the (almost) maximum resolution of 0.5m, the scene will be sampled at around 16 million regularly gridded locations (i.e., 4096 x 4096) to make the texture for holding the elevation values. Anything rendered in the scene, including the ground surface and vector objects like buildings and barrier elements (per step 2.j), will be captured in the height map. There can be a loss in the fidelity of vector elements, such as the crisp edges of buildings, as the height values are converted into a gridded texture format for processing in the GPU.

Note that the elevation values are captured directly from the displayed surface mesh in the 3D view and are *not* directly queried from the underlying data source/s. This means that the way that the surface mesh is triangulated in the view, potentially when working with multiple data sources or coarse resolution data, can have an impact on how elevation values are interpolated.

The processing steps are:

a. The texture resolution for the AOI is calculated by dividing the longest AOI side by the cell size.
    i. Note that when the cell size is automatically calculated, the resolution of the longest side will be 4096 cells (i.e., using a 4k resolution in the GPU)
b. The 3D view is sampled at the center point of each cell, honoring the list of visible layers
    i. This occurs at a single level-of detail, as though the scene is being viewed from above
    ii. Barrier elements, which are rendered in the view, will be included here

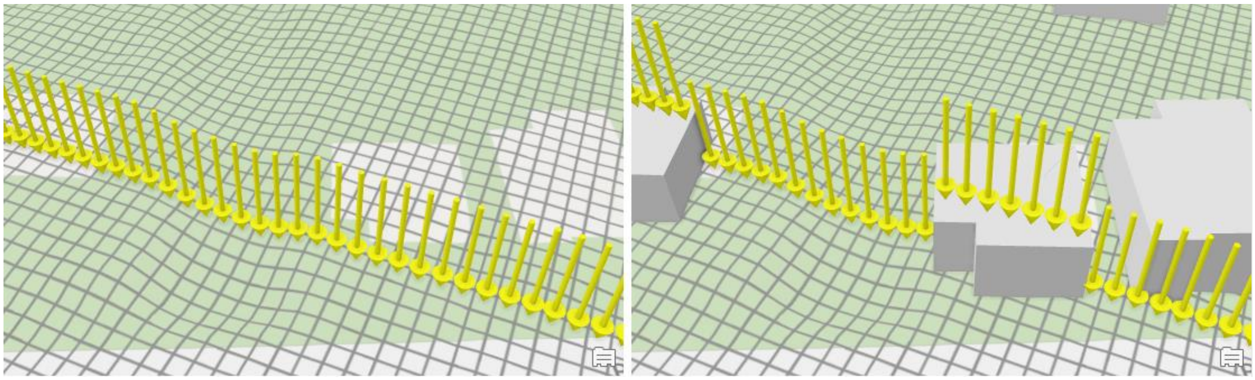


Figure 6. The scene is sampled based on the cell size (i.e., 2m), and honors what is being rendered (left: without buildings and right: with buildings in the scene).

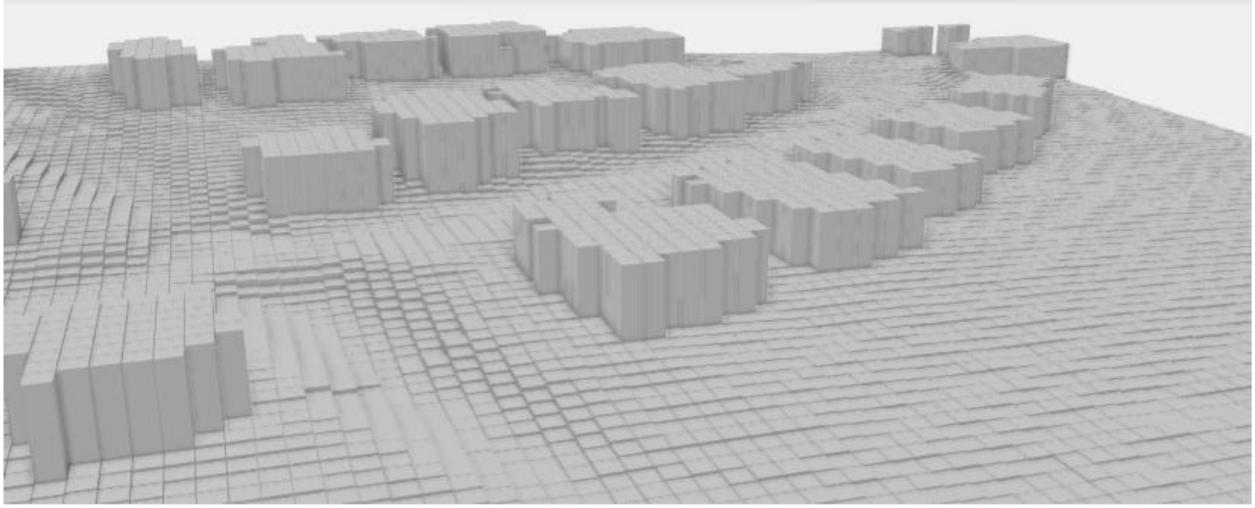


Figure 7. The height map for an area of interest, resampled for a 2m cell size (with buildings)

c. For all Channel elements, cell centers that are along the defined path and within the channel width will be updated with new elevation values based on a linear interpolation of height values between the start and end points of the channel.

d. All computed height values are stored in a texture as terrain height b_{ij}.

This texture will be used to provide the elevation values by the GPU when running the analysis.

Step 4: Simulated Flow

Once the ground height map has been calculated, the water flow simulation can be run. There are two phases. The first is initialization of water on top of the surface before any flow is calculated. And the second is multiple iterations of calculations as water is added (e.g., through rainfall and water sources), removed (e.g., through infiltration and evaporation), and moved across the landscape using the math described in the 'Water Flow Model' section, above.

The processing steps for the initialization phase are:

a. If an initial water depth raster is supplied, water depth d_{ij} is initialized for each cell.

   i. Otherwise, it is set to zero.

b. The flow rate of each cell is initialized to zero.

After the ground height and water depth initialization is performed, iterations for each time step are calculated until the simulation completes its total duration.

The processing steps for each iteration are:

c. Add water for rain. Rainfall rate is provided in height/time units. This value is converted accordingly and added to d_{ij}. In this implementation rainfall is applied to cells homogeneously.

d. Add water for water sources (optional). Water sources can be placed on a map with units of volume/time. Water depths d_{ij} around a water source are increased by the appropriate amount calculated from the water source flow rate.

e. Remove water for evaporation. Evaporation is supplied as height/time value. It is subtracted from each water depth d_{ij}.

f. Remove water for infiltration. If a constant value or an infiltration raster is supplied, water is removed from each cell accordingly. The total infiltration amount per cell can be limited to a value if required.

g. Calculate the corrected flow rate change and add to the previous value.

h. Calculate volume change by using Q_{in} and Q_{out}.

i. Calculate water depth change from volume change and update d_{ij}.

j. If the option to contain water is turned on, water flow to the boundary cells is not allowed.

k. Calculate flow rate and velocity for every cell so that those values can be shown on the user interface for each iteration. Flow rate and velocity are recalculated for ad hoc querying of individual cells.

l. Go to the next iteration.

As the simulation runs, a cache of the progress of the water is captured in discrete steps. This allows fast navigation back to a cached state. To get to an intermediate (non-cached) moment in the simulation, the simulation needs to go to the nearest earlier cached moment and then run the mathematics from there.

The calculated water depth is converted into a mesh at the same resolution as the elevation surface used to run the analysis, and then rendered in the 3D view. By using the same texture resolution, the height values of the water levels will align with the ground elevation. The coloring of the displayed content can be configured to represent water depth or water flow, and the mesh can be rendered using edges, a fill, or a combination of the two.

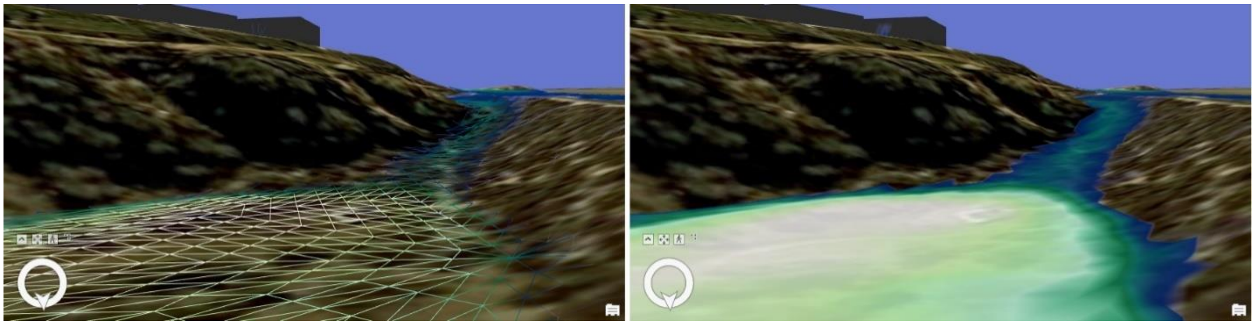


Figure 8. *Water depths align with the terrain as a triangulated mesh (left) and a colored fill (right)*

At this point of running a flood simulation, you could start to consider running alternatives. You can return to Step 2.b and make any changes you like, including the AOI extent, the amount of rainfall, and the barrier elements being used. To compare scenarios, you can duplicate the simulation layer in the scene and create an identical one with just one or two specific changes. The 3D map can contain multiple flood simulation layers, but only one can be active at a time.

Step 5: Export Data Results

Once you've investigated one or more flooding scenarios, you can export the results as GIS data for more workflows, such as creating maps, generating charts and reports, and sharing the results with others. The simulation runs over a period of time, so there are any number of moments within that duration that could be exported. The export process allows you to set a regular time step value – such as every 10 minutes – where values will be exported. As the analysis is processed as an image texture in the GPU, the output will also be in image format (TIF files).

The main steps are:

a. Set the folder location and base name for the exported files
   i. All created files will be saved in the same folder and have a common filename prefix
b. Choose which data values you want to export, including:
   i. Water depth through time, in a specified depth unit (e.g., meters, feet)
      i. Numeric values for the relative depth of the water as it flows and accumulates in the view
   ii. Absolute water depth (water surface elevation) through time, in a specified elevation unit (e.g., meters, feet)
      i. Numeric values for the absolute height of the water (i.e., the elevation of the ground surface added to the depth of the water) as it flows and accumulates in the view
   iii. Water velocity through time, as a two-band raster for the net flow vector
      i. The U and V components of the net-flow vector (i.e., the direction and magnitude) for water as it flows and accumulates in the view
c. Set the temporal step value
   i. The step value in time between exported moments
   ii. Can be defined by time (e.g., every 10 minutes) or count (e.g., 20 slices)
d. Georeferenced raster files are written to disk
   i. A raster file will be created for each combination of *value* and *time-slice*
   ii. Where possible, the export process is accelerated using the cache

For this first release, the exported folder of image files is limited to WGS84. As we move forward, we will support export in the coordinate system of the scene, such as a UTM zone, and provide the option to write the content directly to a multidimensional CRF (cloud raster format) file. The use of the CRF format, in particular, allows the analysis results to be easily queried to generate products like temporal charts and time aware services hosted in a web map.

4. Approximations and limitations

The presented model trades accuracy for computation speed in the form of approximations and assumptions. These limit the use cases and the model outputs/inputs. The assumptions can be listed as:

a. The water surface behaves like a height field. Although this simplification restricts the representation of phenomena like splashing and wave breaking, it remains valid under gentle forces.
b. Water surface variation from cell to cell is limited. The model does not respond well to cases such as dam breaks and tsunami waves where there is a lot of variation in the water surface height field across small areas.

c. The water velocities do not have vertical components. Hence, when steep waves emerge due to disturbances, the model's accuracy diminishes.
d. Vertical isotropy along a column is assumed. This leads to horizontal velocity component within a vertical column of water to remain relatively constant. This assumption may fail under conditions which require simulation of phenomena such as turbulent flow or bottom friction.
e. Viscosity effects are ignored. Chemical/petroleum spills and landslides cannot be simulated accurately.
f. Waterfront can only move one pixel per iteration which limits the maximum speed. This causes inaccuracies in scenarios in which huge water volumes are required to move at high speeds such as dam breaks.
g. Water behaves more accurately for slopes less than 5m/km. As the slope increases, the model cannot calculate the velocities accurately.
h. The model cannot take initial water speeds as input, but only water depths. Therefore, scenarios such as incoming large waves cannot be modeled accurately.

With these assumptions and approximations, the model performs best in applications where the water does not move at high speeds (e.g. finding where water gathers during rainfall). However, it performs poorly at estimating when water will hit critical locations in cases where water moves at high speeds such as dam breaks and tsunamis. The method can still predict the water flow patterns relatively accurately even for the high-water-speed cases.

5. Recommendations for using dynamic flood simulation in ArcGIS Pro

The ability to run flood simulation scenarios – with combinations of initial water, rainfall, and water sources – directly upon a rendered 3D scene in ArcGIS Pro makes flooding analysis very accessible and flexible. However, it is important to understand that the mathematical model used for calculating water flow in the GPU is not well-suited for all potential scenarios. Additional analysis, using other models and techniques, is always recommended for deeper understanding of water flow, especially before making critical decisions.

General scenarios, and the suitability of each for dynamic flood simulation, are described below.

i. Shallow water flow

Shallow water flow models the movement of water over the ground elevation surface and around physical structures. In these cases, water will flow together to merge into streams and accumulate in depressions. Water will be directed by the slope of the landscape and any physical elements built into it, like ditches and banks, as well as by any 3D vector objects being displayed, like buildings, walls, and sandbags. Water will accumulate behind obstructions and only overtop them after enough water has accumulated at that location. The water level is mostly shallow and can accumulate.

Dynamic flood simulation is generally well-suited for this use case. Examples include heavy rainfall in urban or rural areas, the controlled venting of dams, and non-explosive rupture of pipes. Note that in some cases there can be a large build up, and subsequent fast release, of water behind a swale in the landscape, at which point the limitations described in ‘Large water release’ below may be observed.

ii. Water encroachment

Water encroachment models water flowing into a landscape from either a point location, like a burst pipe, or from an areal region, such as the ocean. As water moves uphill into an area, it will push into the terrain to reveal all the nooks and crannies. When water runs downhill from a source, it will find the easiest and steepest path to take. And, in both cases, water will accumulate and coalesce in places that have indentations and containment areas.

Dynamic flood simulation is generally well-suited for this use case. Examples include sea level rise, burst pipes, and filling up of dams. Note that the limited area-of-interest of a simulation, which is also tied to the cell size, might restrict the scope or accuracy of the analysis.

iii. Geodesign / What-If scenarios

Running geodesign and what-if scenarios for flood simulation is the process of comparing multiple possibilities from a core starting point, such the before and after analysis of heavy rainfall on an area of proposed construction or comparing potential locations of sandbag barriers. Each flood simulation scenario is configured as a separate layer, and each can contain one or more differences – such as different rainfall rates or alternate ground surfaces and 3D objects. Water flow will be impacted by these changes and the results of each scenario can be compared visually or empirically (as exported temporal rasters).
Dynamic flood simulation is generally well-suited for this use case. Examples include evaluating urban planning decisions, testing mitigation options, and running best-case versus worst-case scenarios for predicted weather. Note that all scenarios will need to be evaluated against the known limitations of the water flow model.

iv. Large water release

Large water release models tall columns of water collapsing and rushing through the landscape. This kind of analysis requires water to travel at different speeds within a single vertical column of water so it can “break” and collapse forward at higher speeds. The simplifications made in the flow model to support it in the GPU specifically removed the inclusion of this factor – see Approximations and Limitations for more details.

Dynamic flood simulation is not well-suited for this use case. Examples include catastrophic dam failures, tsunamis, and deep water moving down steep slopes.

v. Timed movement of water

The timed movement of water models how long it takes to move water between locations based on certain conditions. For cell-based analysis, one of the simplifications made when moving the flow model into the GPU was that water cannot move across multiple cells in one time step. This restricts water speed, especially in turbulent and deep water scenarios, and can lead to inaccuracies in water flow speed – see Approximations and Limitations for more details.

Dynamic flood simulation is not well-suited for this use case. Examples include calculating precisely when water vented from a dam will arrive at a point downstream, or how many hours a reservoir will take to fill up. While the model will route water flow and accumulation, the timing of the water movement is estimated.

vi. Wide-area floodplains

Wide-area floodplain modeling works with water flow over very large areas, such as a hundred square kilometers of flat land where multiple rivers converge or a long valley that collates water from multiple watersheds. This analysis requires water to move freely across a wide expanse, which can quickly exceed the limited AOI extent used in the GPU based modeling.

Dynamic flood simulation is not currently well-suited for this use case. Potential future work on monitoring where and when water leaves the boundary of an area-of-interest, and using that as input for subsequent analysis areas, may help in some cases.

6. Summary

Dynamic flood modeling in ArcGIS Pro provides fast processing of water flow using minimal configuration steps and easy reconfiguration to experiment with alternative situations. By necessity, multiple simplifications and limitations have been applied to the water flow model to allow it run quickly in the GPU, and being aware of what these simplifications mean to the accuracy of the results is critical. This new capability should be used as an addition to the flood modeling and planning capabilities already in use, with a focus on experimentation and general understanding of water flow for a region.

Bibliography

[1] X. Mei, P. Decaudin and B.-G. Hu, "Fast Hydraulic Erosion Simulation and Visualization on GPU," in *15th Pacific Conference on Computer Graphics and Applications*, 2007.

[2] J. F. O'Brien and J. K. Hodgins, "Dynamic Simulation of Splashing Fluids," in *Computer Animation*, 1995.

[3] M. Kass and G. Miller, "Rapid, stable fluid dynamics for computer graphics," *SIGGRAPH Comput. Graph.,* vol. 24, no. 4, p. 49–57, August 1990.

[4] O. Št'ava, B. Beneš, M. Brisbin and J. Křivánek, "Interactive terrain modeling using hydraulic erosion," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2008.

[5] J. Hawkins, "Interactive Erosion," [Online]. Available: https://github.com/Scrawk/Interactive-Erosion.

[6] B. Shishov, "GPU Terrain Erosion," [Online]. Available: https://github.com/bshishov/UnityTerrainErosionGPU.

[7] V. Chow, D. Maidment and L. Mays, Applied Hydrology, McGraw-Hill, 1988.

THE SCIENCE OF WHERE®

Esri, the global market leader in geographic information system (GIS) software, offers the most powerful mapping and spatial analytics technology available.

Since 1969, Esri has helped customers unlock the full potential of data to improve operational and business results. Today, Esri software is deployed in more than 350,000 organizations including the world's largest cities, most national governments, 75 percent of Fortune 500 companies, and more than 7,000 colleges and universities. Esri engineers the most advanced solutions for digital transformation, the Internet of Things (IoT), and location analytics to inform the most authoritative maps in the world.

Visit us at esri.com.

Contact Esri

380 New York Street
Redlands, California 92373-8100 USA

1 800 447 9778
T 909 793 2853
F 909 793 5953
info@esri.com
esri.com

Offices worldwide
esri.com/locations

For more information, visit
esri.com