STUDENT SUMMER INTERNSHIP TECHNICAL REPORT

Exploration of Toolsets for Development of an Integrated Hydrology Model of Basin-6 near the Waste Isolation Pilot Plant (WIPP)

DOE-FIU SCIENCE & TECHNOLOGY
WORKFORCE DEVELOPMENT PROGRAM

Date submitted:

December 10, 2021

Principal Investigators:

Gisselle Gutierrez-Zuniga (DOE Fellow Student)
Florida International University

David Moulton, Ph.D. (Mentor)
Los Alamos National Laboratory

Daniel Livingston, M.S. (Mentor)
Los Alamos National Laboratory

Ravi Gudavalli, Ph.D. (Program Manager)
Florida International University

Leonel Lagos, Ph.D., PMP® (Program Director)
Florida International University

Submitted to:

U.S. Department of Energy
Office of Environmental Management
Under Cooperative Agreement # DE-EM0005213
Los Alamos Report: LA-UR-21-30104

FIU | Applied Research Center
FLORIDA INTERNATIONAL UNIVERSITY

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

This research work has been supported by the DOE-FIU Science & Technology Workforce Initiative, an innovative program developed by the U.S. Department of Energy's Office of Environmental Management (DOE-EM) and Florida International University's Applied Research Center (FIU-ARC). During the summer of 2021, a DOE Fellow intern, Gisselle Gutierrez-Zuniga, spent 10 weeks doing a remote summer internship with Los Alamos National Laboratory under the supervision and guidance of Dr. David Moulton and Mr. Daniel Livingston. The intern's project was initiated on June 1, 2021, and continued through August 6, 2021, with the objective of learning various toolsets required for development of an integrated hydrology model of Basin 6 of the Nash Draw west of the Waste Isolation Pilot Plant (WIPP) located in Carlsbad, New Mexico.

The WIPP is the nation's only deep geologic long-lived radioactive waste repository in operation which isolates transuranic waste 2,150 feet underground within the Salado Salt Formation. Scientists are concerned about the long-term vulnerability of the karst topography of this region and the influence of climate and karst features, such as sinkholes and brine lakes, which can contribute to the groundwater recharge and dissolution of the subsurface geological layers over time. These environmental and topographical characteristics of the region can potentially impact the long-term integrity and performance of the repository, an impact that is further exacerbated by incompatible land use activities occurring around the WIPP and surrounding basins. An excellent example of these karst regions near WIPP is Basin 6 within the Nash Draw, which developed through solution and erosion of upper Permian rocks creating an array of surface features, including sinkholes, swallets, and karst valleys. The overall objective of this research, therefore, is to study the impact of these surface features, in conjunction with soil properties and vegetation types, on the groundwater recharge in Basin 6 using model simulations of the hydrologic response to a range of storm events. Specifically, the hydrologic response of the coupled surface/subsurface flow system to those storm events is simulated with the Advanced Terrestrial Simulator (ATS). The ATS is an ecosystem-based, integrated, distributed hydrology simulator that is built on the underlying multi-physics framework provided by Amanzi, the high performance computing simulator developed in the Advanced Simulation Capability for Environmental Management (ASCEM) program. The output of the ATS model includes predictions of infiltration rates over selected regions of interest, such as sinkholes, and groundwater recharge, and hence ensembles of ATS simulations facilitate sensitivity and uncertainty analysis of groundwater and surface water flows. This internship documents the workflow and tools needed to perform an ensemble of ATS simulations, including the generation of meshes from digital elevation model (DEM) data, setting up meteorological forcing data, developing input files for the ATS, executing the simulations on local or remote systems, and analyzing the output. This documentation is through a series of Jupyter notebooks that detail each of the steps in this workflow, and was first created for a small well-studied watershed on the Canadian Forces Base Borden in Ontario, Canada. With this initial development of the workflow complete, each step is being adapted to work on a representative sub-region of Basin 6 where the FIU UAV-based survey data is available, and additional sinkhole post processing and analysis has been done. Ultimately, the experience and skills acquired during this internship will be used to implement the established workflow over the entire Basin 6 study area using a high-resolution DEM generated by UAV photogrammetry methods.

1. INTRODUCTION

The Waste Isolation Pilot Plant (WIPP), located in Carlsbad, New Mexico, is the nation's only deep geologic long-lived radioactive waste repository in operation that permanently isolates transuranic waste 2,150 feet underground within a layer of salt known as the Salado Formation. The WIPP is in a karst region in New Mexico that formed from the dissolution of soluble rocks such as limestone, dolomite, and gypsum. Above the Salado Formation lies the Rustler Formation which contains three-recognized fluid-bearing zones: the Rustler-Salado contact residuum, the Culebra dolomite and the Magenta dolomite. Of the three, the Magenta and Culebra are of prime concern because they extend over the WIPP site. West of WIPP is the Nash Draw, an enclosed basin that developed as a result of erosion of upper Permian rocks of the Rustler and Salado formations. Within the Nash Draw, there are thirty internally drained basins identified from topography and field surveys, including the focus of this work, Basin 6 (Figure 1).

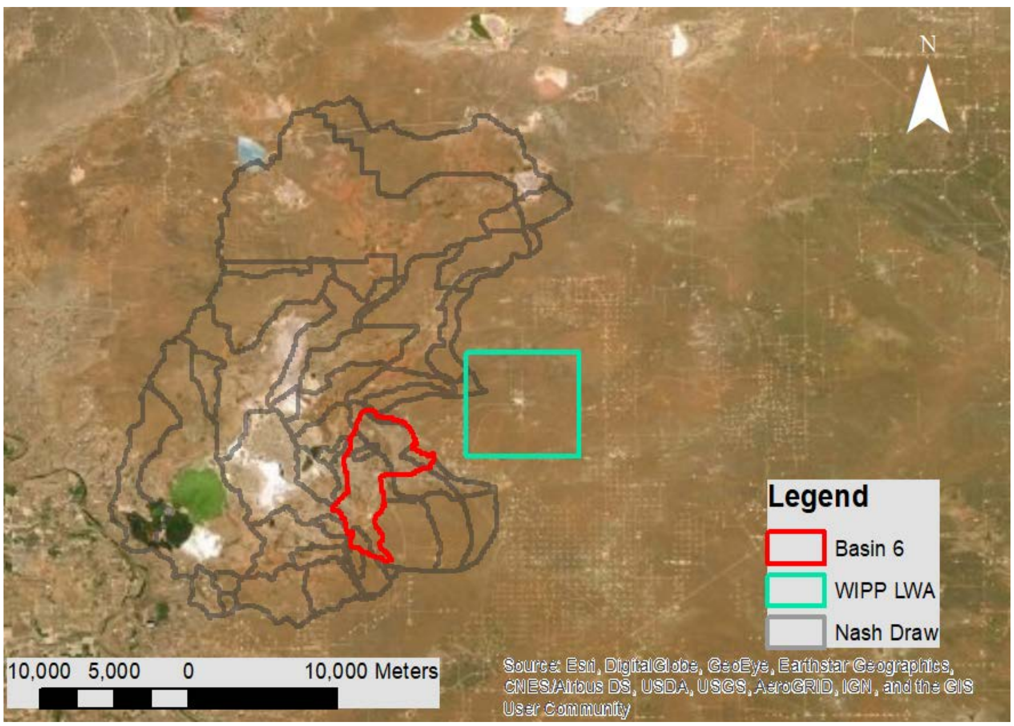

Figure 1. Map of thirty delineated basins within the Nash Draw. The study regions of interest in this work, Basin 6, is highlighted in red.

Scientists and researchers are concerned about the long-term vulnerability of the karst topography of this region and the influence of surface features, such as sinkholes, swallets, and karst valleys on groundwater recharge. Additionally, there is a need for improved understanding of the regional water balance, particularly the relation between the Culebra recharge and the intense, episodic precipitation events typical of a monsoon. This relationship is essential for understanding the rate of propagation of the shallow dissolution front, and the impact of land-use changes around the WIPP facility and Nash Draw on water levels and chemistry in compliance-monitoring wells. These types of analyses require a revision of the current site conceptual model to couple surface water and groundwater processes, which both require a high-resolution digital elevation model (DEM) to more accurately delineate surface features, including channels and sink holes, to better account for surface water routing and return flow. The outputs of the open-source Advanced Terrestrial Simulator (ATS) model being used in this study include predictions of infiltration rates over selected regions of interest, such as sinkholes, and groundwater recharge, and hence ensembles of ATS simulations facilitate sensitivity and uncertainty analysis of groundwater and surface water flows.

The ATS builds upon the multi-physics framework and low-level components (mesh infrastructure, discretizations, and solvers) provided by Amanzi, the high-performance computing simulator developed in the Advanced Simulation Capability for Environmental Management (ASCEM) program. Amanzi is used for environmental applications to provide flexible and extensible flow and reactive transport simulation capability. The ATS simulates ecosystem-based, integrated, distributed hydrology using various forms of Richards equation coupled to a surface flow equation along with needed sources and sinks for ecosystem and climate models (Coon et al, 2016). Configuring the ATS to simulate a specific coupled system may require a combination of several processes, which can be combined and organized in a tree structure with the individual physical models at the leaves of the tree called Process Kernels (PK). These PKs are then coupled by multi-process coordinators (MPCs), which can provide weak (sequential) or strong (fully implicit) coupling (Coon et al, 2013). The runtime control through a PK tree makes testing of coupled processes much easier because each test can be built up hierarchically, testing each process individually and then with multiple coupled processes together. It also makes it easier for scientists to start their studies with the simplest model possible, iteratively adding complexity as needed and as supported by data. Figure 2 shows an example schematic of a PK-tree for thermal integrated hydrology with meteorological forcing. Here the leaves are the brown boxes and represent the individual process models, including Richards model for variably saturated subsurface flow, diffusive wave model of surface flow, standard advection equation models of subsurface and surface energy, and a surface energy balance model, which includes snow, evaporation and precipitation.

Additionally, Amanzi provides an advanced mesh infrastructure for the ATS that is capable of reading large 3D meshes in parallel, subsetting meshes, managing multiple meshes, and deforming them. For this project, the open-source Python module TINerator was used for the creation of unstructured 3D and 2.5D meshes from GIS data sources. With TINerator, a DEM is imported and an unstructured triangulated mesh is generated to represent the surface topography in the ATS model. This unstructured mesh is refined (smaller triangles) near selected topographic features, such as stream networks and sinkholes. Those topographic features can be identified in a number of ways, including externally provided shapefiles and internal processing of the DEM by TINerator. The development of the ATS model for Basin 6 requires the development of an unstructured mesh using TINerator and will serve as the input for the ATS.

This internship will document the workflow and tools needed to perform a series of ATS simulations, including generating meshes from a DEM, setting up meteorological forcing data, developing input files for the ATS, executing the simulations on local or remote systems, and analyzing the output. This documentation is through a series of Jupyter notebooks that detail each of the steps in this workflow and is developed first for a small well-studied watershed on the Canadian Forces Base Borden in Ontario, Canada. With this initial development and understanding of the workflow complete, each step will be applied to a representative sub-region of Basin 6 where the FIU UAV-based survey data is available, and additional sinkhole post processing and analysis has been done. Ultimately, the experience and skills acquired during this internship will be used to implement the established workflow over the entire Basin 6 study area using a high-resolution DEM generated by UAV photogrammetry methods.

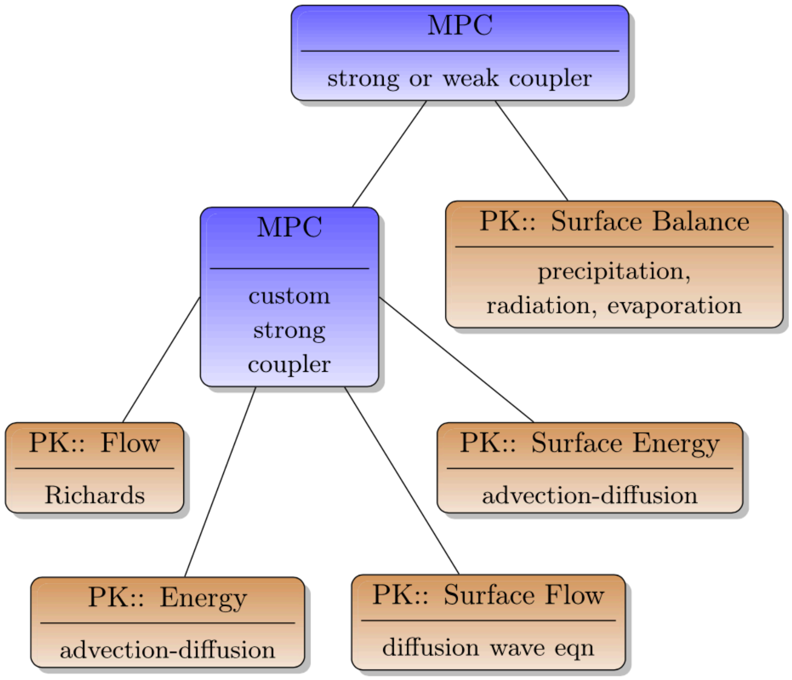

Figure 2. PK-tree for coupled surface/subsurface thermal hydrology, driven by a surface energy balance model, as in ATS.

2. RESEARCH DESCRIPTION

The general workflow to develop an ATS integrated hydrology model is comprised of the steps shown in Figure 3. Model development begins with the collection of geospatial data, such as DEMs, GIS shapefiles of surface features, soil and vegetation types, and meteorological forcing variables. This raw data is then processed to improve quality, ensure common georeferenced coordinates and measurement units, and to create files in a format supported by the ATS. Once the available data has been collected and pre-processed, it can be imported into a mesh generation tool that is capable of producing meshes in the ExodusII format needed by ATS. With these steps complete, a specific model and forcing scenario is identified (e.g., surface flow with a single rainfall event), and an input file for the ATS is created. Using the input file, the ATS is run to simulate that scenario, and results can be visualized using open-source software such as VisIt and ParaView.

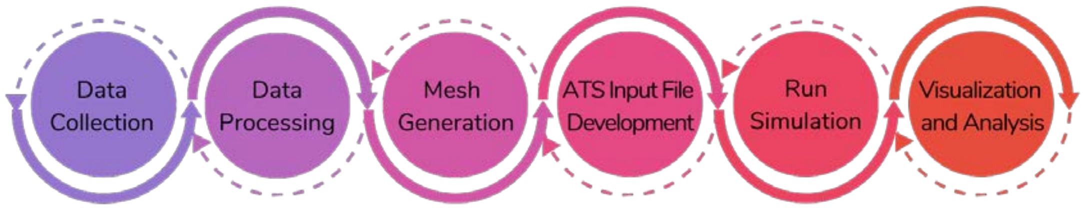

Figure 3. The model development workflow begins with Data Collection and proceeds to the right (solid arrows) but issues may arise at any step and result in backtracking to the previous step or require iterations (dashed arrows).

In practice, the steps in this workflow do not simply follow the direct left to right path shown by the solid arrows in Figure 3. Instead, problems may arise at any step requiring iteration within a step, or even backtracking and iteration on a previous step. This iterative aspect of the workflow is shown with the dashed lines in Figure 3. For example, once a mesh is generated it may be apparent that there is a problem in some part of the DEM, leading to review and improvements to the processing of the DEM before returning to mesh generation. Similarly, once surface flow is simulated it may be clear that more refinement of the mesh is required around a particular feature, sending the scientist back to the mesh generation step before proceeding with more simulations.

To effectively support these iterative workflows and ensure that they are transparent and transferable, Jupyter notebooks are used at each step. Jupyter notebooks provide a web-based open-source tool that supports a combination of explanatory text (including mathematical expressions), software scripting and graphical output, to form an interactive narrative. In this case, such an interactive narrative is developed for each step of the workflow, making it easy for someone to reproduce the analysis at this site, or make minor changes to apply it to another site.

2.1 Data Collection and Data Processing

A significant amount of geohydrologic data is publicly available and can be downloaded interactively through web-based forms or using scripts. For example, the USGS provides 10m DEM data and the USDA provides the SSURGO database of soil properties over the coterminous United States. However, to develop an understanding for the workflow a small well-studied site was selected, and the limited data needed for the ATS model in this case was available from a recent model intercomparison study (Kollet et al, 2017). Specifically, from that study a 50cm and

1m DEM of a small catchment of the Canadian Forces Base Borden in Ontario, Canada (referred to as Borden) was obtained, along with other model parameters. The DEM, which provided limited metadata in these simple text files and marked the watershed boundary with no data values, is shown in Figure 4.

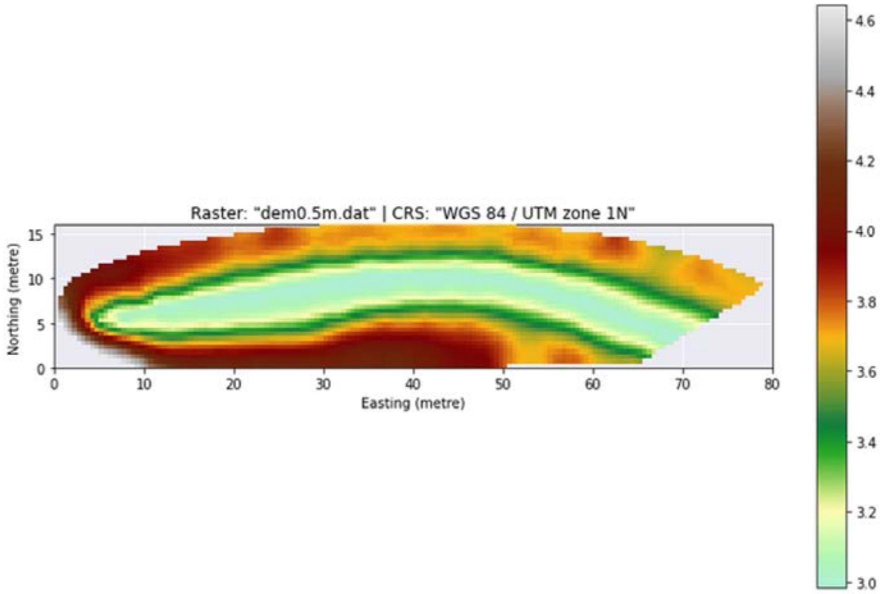

Figure 4. 50 cm-resolution DEM of Borden watershed.

2.2 Model Development and Execution

2.2.1 TINerator

A major component in developing the ATS integrated hydrology model consists of developing a mesh that accurately represents the geologic structures and engineered systems, and hence, supports accurate discretizations of models of fluid flow and mass transport. TINerator is a mesh generator that was created at Los Alamos National Laboratory (LANL) and the mesh can be exported in an ExodusII format which encodes important geometrical and geophysical information about the model and is designed for the ATS. It is designed as a Python module that can be used in python scripts or within Jupyter notebooks.

The general workflow with TINerator consists of 1) loading the DEM and surface feature data, 2) watershed delineation, 3) mesh generation, and 4) faceset generation.

1. Loading DEM and Surface Feature Data

For the Borden watershed, 1 meter and 0.5-meter resolution DEMs were loaded in TINerator using the function shown in Figure 5. Additionally, TINerator provides the tools to reproject a DEM given a coordinate reference system (CRS) which could be read in as an EPSG code, a WKT string, a Pyproj4 string, or a pyproj4 dict.

```
dem_1m = tin.gis.load_raster(data.dem_1m)
dem_1m.fill_depressions()

dem_50cm = tin.gis.load_raster(data.dem_50cm)
dem_50cm.fill_depressions()

dem_1m.plot()
dem_50cm.plot()
```

Figure 5. Loading GIS data and fill depressions function on TINerator.

2. Watershed Delineation

In order to prepare the DEM as input for mesh generation, an automatic watershed delineation is done. A watershed delineation consists of 1) filling the depressions, 2) resolving flats, and 3) performing flow accumulation (Figure 6). Of particular importance is the flow accumulation step as it identifies an approximate stream network and stores this information in a raster for visualization and use by other tools. In this workflow, TINerator can use this raster to guide the refinement of the variable resolution unstructured mesh around the streams. Flow accumulation algorithms generate this raster of accumulated flow by accumulating the weight of all cells that flow into adjacent cells with negative elevation data ('downslope cells'). With TINerator, there are numerous types of flow accumulation algorithms that may be used:

1. D8 (default)
2. D4
3. D-Infinity
4. Rho8
5. Rho4
6. Quinn
7. Freeman
8. Holmgren

Once a flow accumulation algorithm is defined, the flow network (Figure 7) can be extracted with an accumulation threshold, as all cells above that threshold are defined to be a part of the network.

```
ws_flow = tin.gis.watershed_delineation(dem_50cm, threshold=100., method='D8')
ws_flow.plot(layers=[dem_50cm])
```

Figure 6. Watershed delineation function including threshold and method parameters on TINerator.

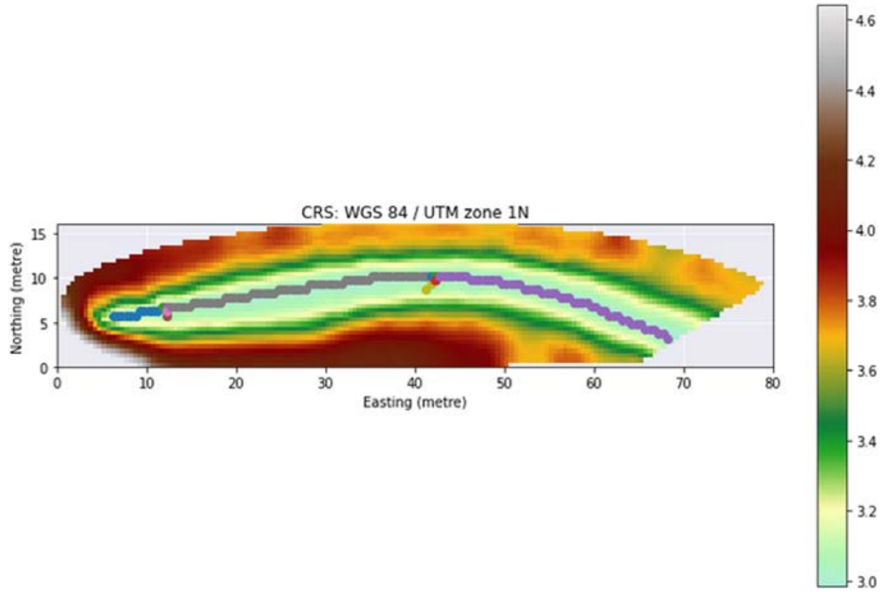

Figure 7. Watershed delineation on Borden watershed DEM using a threshold of 100. The main stream is clearly visible in the centerline of the watershed, starting with blue dots at the left followed by gray in the middle and then purple at the right. Two small tributaries are also captured in this flow network.

3. Generating a Mesh

Once the DEM has been processed, the next step is to generate a triangulated mesh (Figure 8). To perform this mesh generation in TINerator, the 'tin.meshing.triangulate' method is used and the following parameters must be defined:

- Minimum edge length (min_edge_length): The minimum edge length for created triangles.
- Maximum edge length (max_edge_length): The maximum edge length for created triangles.
- Scaling type: Can either be relative or absolute. "Relative" meaning the edge length parameters will be a percent of the DEM extent. "Absolute" means the edge length parameters will be interpreted in the units that the DEM is in.
- Method:
    - Jigsaw (default)
    - MeshPy
    - Gmsh

Here the "Method" parameter selects a particular mesh generation tool. In this study, the Jigsaw mesh generator is chosen because it has the strongest capabilities for developing a high-quality Delaunay triangulation on the surface, which is desirable for the accurate discretizations of flow and transport (Figure 9).

```
triangular_surface = tin.meshing.triangulate(
    dem_50cm,
    min_edge_length=0.01,
    max_edge_length=0.1,
    method='jigsaw',
    refinement_feature=ws_flow,
    scaling_type='relative',
)

triangular_surface.view(window_size=(400,400))
```

Figure 8. Generation of triangulated mesh which includes the parameters of min edge length, max edge length, method, refinement feature, and scaling type.

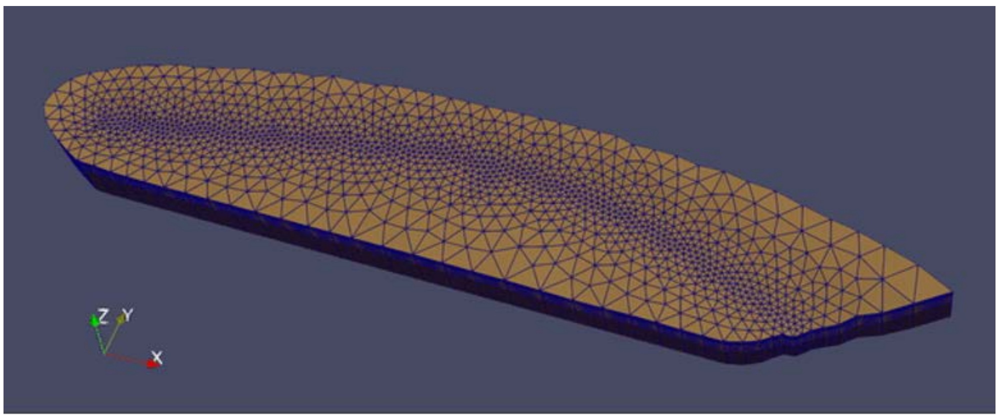

Figure 9. Triangulated mesh generated for Borden and visualized using ParaView.

The surface mesh is then extruded into the subsurface to create a volume mesh. This extrusion is built around adding layers, each of which can have any number of sublayers. In addition, each layer can be terrain following, use a reference surface or be set to a particular elevation. The number of elements in the volumetric mesh is directly proportional to the total number of sublayers.

$$\mathrm{N_{volume}} = \mathrm{N_{surface}} \times \mathrm{N_{sublayers}}$$

The layering schema is:

- Layer 1: 50 cm thick, with 5 sublayers
- Layer 2: 1 meter thick, with 5 sublayers in 20 cm increments
- Layer 3: flat bottom at `z = 0 m` , with 5 sublayers

```
layers = [
    ("constant", 0.5, 5, 1),
    ("constant", 1., 5, 2),
    ("snapped", 0., 5, 3),
]

volume_mesh = tin.meshing.extrude_mesh(triangular_surface, layers)
volume_mesh.view(window_size=(800,800))
```

Figure 10. Example layering schema for Borden watershed.

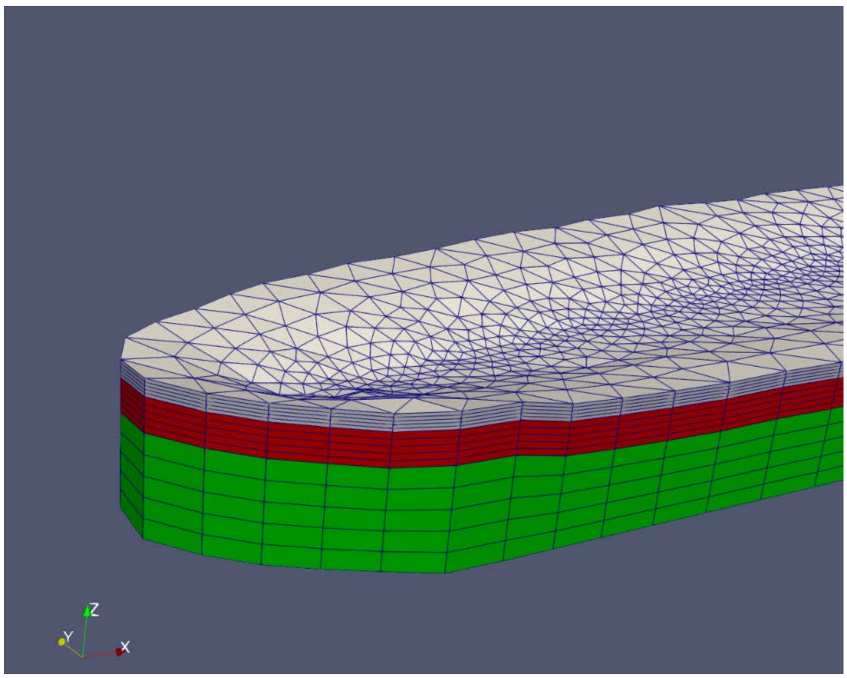

Figure 11. Five layer prism mesh of Borden watershed with layer depths exaggerated for effect.

Optionally, each layer can be assigned a material ID which aids in identifying which layer an element is in. For example, the material ID allows a region to be defined in the ATS input file, and then that region can be used to assign values to model parameters, such as the permeability.

4. Generating sets

To work with the boundary of a volume mesh, particularly to define sets of faces that will be used to define spatial regions in the ATS model, TINerator provides several handy methods. In this part of the workflow the first step is to generate a surface mesh from a volume mesh:

```
# Extract the surface mesh
surface_mesh = volume_mesh.surface_mesh()
surface_mesh.view()
```

Figure 12. TINerator function to create a surface mesh from a volume mesh.

Following the terminology of the ExodusII mesh format, sets of faces used to define spatial regions are referred to as facesets, and sets of elements (or cells) are element sets. The most common application for facesets is to define spatial regions over which boundary conditions or source terms are applied, while element sets define objects such as wells. TINerator provides methods to help define those facesets over arbitrary sub-domains of the mesh's surface (Livingston, 2020). In addition, TINerator provides three ‘Naïve’ facesets: top, bottom, and sides. Once the surface mesh is created with ‘vol_mesh.surface_mesh()’, the naïve facesets are auto-generated. Using ‘vol_mesh.view()’, the faceset and nodes can be visualized.

```
# Extract top and bottom faces, and side nodes
top_faces = surface_mesh.top_faces
bottom_faces = surface_mesh.bottom_faces
side_faces = surface_mesh.side_faces

print('Sets')
print('----')
print(top_faces)
print(bottom_faces)
print(side_faces)

# View them
volume_mesh.view(sets=(top_faces, bottom_faces, side_faces), window_size=(800, 800))
```

Figure 13. TINerator function to extract and view top, bottom, and side faces of the surface mesh.

Additionally, there are advanced set operations that allow for regions to be defined and analyzed separately. In this example, an outlet was identified as a set of nodes (in clockwise ordering) using the ‘discretize_sides’ function in order to capture the faces along the perimeter of the outlet. The nodes are in (x, y) pairs which are assumed to be in the same CRS as the mesh. The ‘at_layer’ function allows for a specific layer of interest to be identified instead of every side.

```python
# Generates two sets:
#  1 - Outlet (between the line segment in `outlet_pts`)
#  2 - No flow (everywhere else)

outlet_pts = [
    (71.9, 4.7),
    (67.3, 2.1),
]

# Returns list of two sets
outlet = surface_mesh.discretize_sides(
    outlet_pts, close_ends=True, #at_layer=(1, 1), set_name_prefix="Outlet"
)

# Change the set names
outlet[0].name = "Outlet"
outlet[1].name = "NoFlow"

print('Outlet sets: ')
print('-------------')
print(outlet)

# Visualize
volume_mesh.plot(sets=[outlet])
```

Figure 14. TINerator function to create and extract a region of interest of the surface mesh.

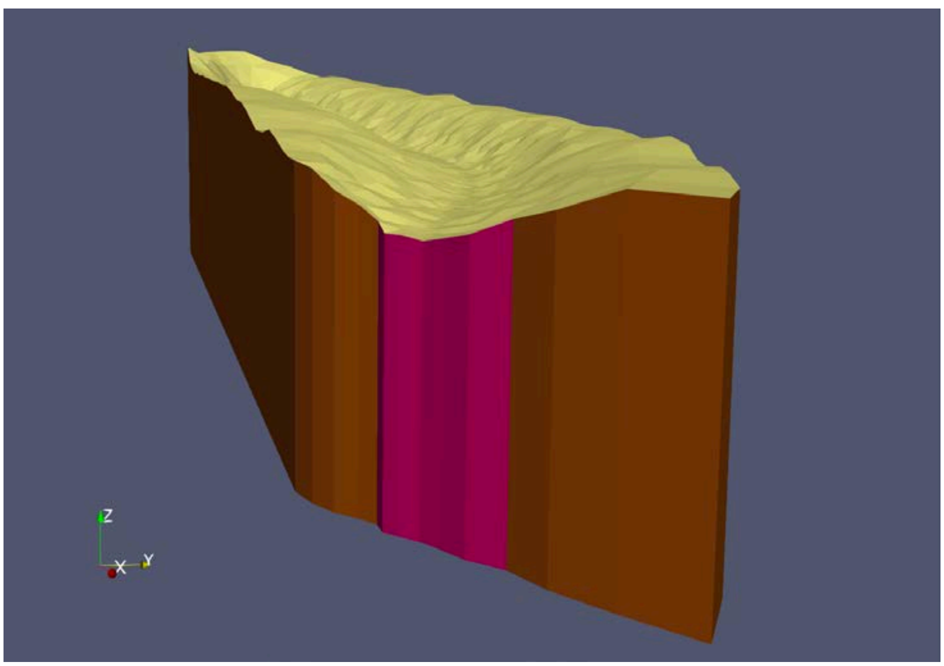

Figure 15. Highlighted facesets (top, bottom, sides, and outlet) of Borden.

After the facesets are created, the volumetric mesh can be exported in an ExodusII format as can be seen in the following figure:

```python
volume_mesh.save("Borden.exo", sets=[outlet, top_faces, bottom_faces])
```

Figure 16. Export of volume mesh in ExodusII format using TINerator.

2.2.2 ATS Input File Development

As described previously, the ATS is a flexible multi-physics simulation capability that allows a specific model (set of coupled processes) to be configured through an XML input file at run time. Specifically, the XML input file defines all aspects of the simulation, including reading or creating the mesh, defining geometric or geophysical spatial regions, the start and end time, external forcing such as precipitation, and output for visualization and analysis. Perhaps most importantly the input file also describes the PK tree, which defines the processes and couplers used to represent the model in this simulation, and the corresponding system, which includes all the variables and predefined fields used in the simulation. These important parts of the ATS input file comprise the major sections of the XML input file:

- "mesh" [mesh-typed-spec-list]: A list of Mesh objects.
- "regions" [region-typed-sublist-spec-list]: A list of Region objects.
- "cycle driver" [coordinator-spec]: Includes the PK tree
- "PKs" [pk-typed-spec-list]: Details of each PK and MPC used in the PK tree
    - Examples include:
        - Flow PKs
            - Richards PK
            - Permafrost Flow PK
            - Overland Flow PK
        - Transport PK
        - Energy PKs
        - Surface Energy Balance PKs
        - Biogeochemistry
        - Deformation
        - Multi-process Couplers (MPCs)
            - coupled water
            - permafrost surface and subsurface
- "visualization" [visualization-spec-list]
- "observations" [observation-spec-list]
- "checkpoint" [checkpoint-spec]

This work first considers a model of surface flow alone. In this case the PK tree has a single leaf, which holds the overland flow PK, and it is in this element where the initial conditions, boundary conditions, and model parameters are defined. Simulation of a rainfall event and dryout is performed with this model to evaluate the quality of the surface mesh through observations of stream flow.

Second, an input file is developed for coupled surface and subsurface flow, which is commonly referred to as integrated hydrology. This PK tree has two leaves, one holding the overland flow PK, which is associated with the surface mesh, and the other holding Richards PK, which is associated with the subsurface mesh. In each of these PK elements of the input file the initial conditions, boundary conditions and parameters are defined. The "coupled water" MPC couples these PKs, and within this element any options about the coupling conditions that apply between the models are added.

The following figure shows a portion of the input file for the Borden watershed example. The entire input file can be found in Appendix A.

```xml
<!-- Begin: ATS Simulation Input -->
<ParameterList name="Main" type="ParameterList">

  <!-- Begin: Mesh -->
  <ParameterList name="mesh" type="ParameterList">
    <ParameterList name="domain">
      <Parameter name="mesh type" type="string" value="read mesh file" />
      <ParameterList name="read mesh file parameters" type="ParameterList">
        <Parameter name="file" type="string" value="borden_3-layers_unamed-sets_v1.exo" />
        <Parameter name="format" type="string" value="Exodus II" />
      </ParameterList>
    </ParameterList>
    <ParameterList name="surface">
      <Parameter name="mesh type" type="string" value="surface" />
      <ParameterList name="surface parameters" type="ParameterList">
        <Parameter name="surface sideset name" type="string" value="surface" />
      </ParameterList>
    </ParameterList>
  </ParameterList>
  <!-- End: Mesh -->

  <!-- Begin: Regions -->
  <ParameterList name="regions" type="ParameterList">

    <ParameterList name="computational domain" type="ParameterList">
      <ParameterList name="region: all" type="ParameterList">
      </ParameterList>
    </ParameterList>
    <ParameterList name="surface domain" type="ParameterList">
      <ParameterList name="region: all" type="ParameterList">
      </ParameterList>
    </ParameterList>

    <!-- ATS extracts the surface boundary (as a handy default option) from your mesh -->
    <ParameterList name="surface boundary" type="ParameterList">
      <ParameterList name="region: boundary" type="ParameterList">
```

Figure 17. XML input file required for ATS.

2.2.3 Scenario Description and Running ATS

Once the input file has been created, the ATS simulation can be run in the Docker container; or for large scale data, it can be run on a supercomputer, such as Cori provided in the National Energy Research Scientific Computing Center (NERSC). For training purposes, a 12-hour rainfall event was simulated on the Borden watershed. Results of this simulation can be seen in the next subsection.

2.3 Visualization and Analysis

2.3.1 Borden Watershed Simulation

Results produced by TINerator and the ATS can be visualized using open-source software such as ParaView and VisIt. ParaView began as a collaborative effort between Kitware Inc. and Los Alamos National Laboratory and can be described as an open-source, multi-platform data analysis and visualization application. Similarly, VisIt is an open-source, interactive, scalable, visualization, animation, and analysis tool that was originally developed by the Department of Energy (DOE) Advanced Simulation and Computing Initiative (ASCI). Both were used throughout the internship in order to visualize meshes, facesets, and the results produced by ATS. Using the XML input parameter file created for the Borden watershed, a 12-hour rainfall simulation was performed and visualized using VisIt (Figure 18).

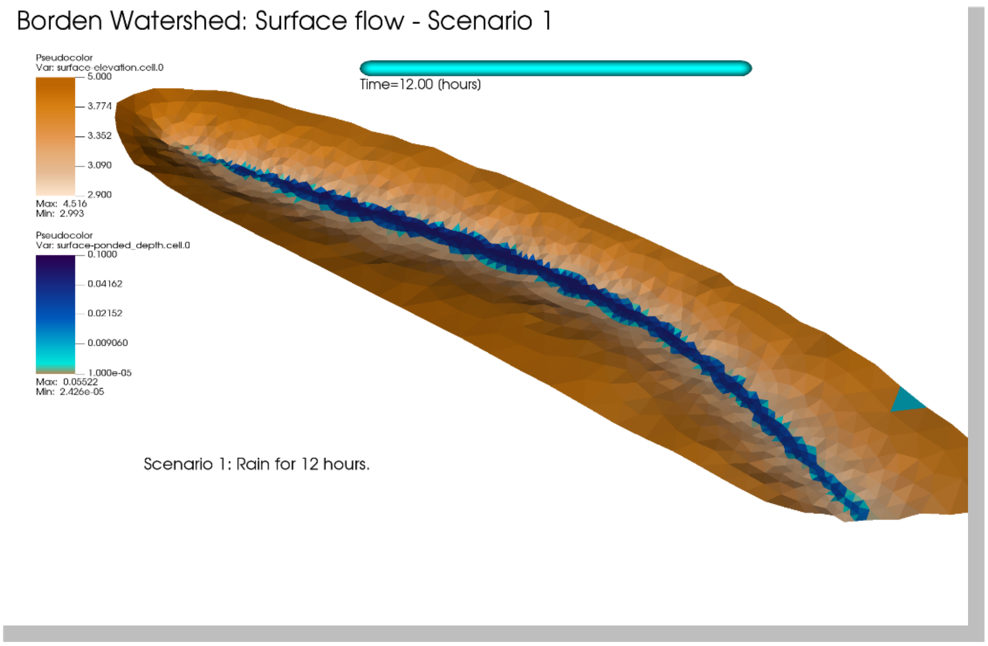

Figure 18. Overland flow scenario of 12-hour rainfall event on Borden watershed.

3. RESULTS AND ANALYSIS

After establishing a workflow using the Borden watershed as a test case, the same methodology was implemented on a sub-section of data for the Basin 6 study area as described below. A Jupyter notebook was created for the Basin 6 sub-section using the UAV-generated DEM from a previous field trip to New Mexico. As per the established workflow, the data was collected and processed, and a mesh was generated using TINerator and visualized using ParaView and VisIt. Future work beyond this report will focus on developing the input file for the ATS and running rain event scenarios.

3.1 Data Collection and Processing

As with the Borden example, the data needed to develop an ATS integrated hydrology model for a subset of Basin 6 includes the DEM, meterological forcing data, and vegetation and soil types that prescribe model parameters. For this study of a sub-region of Basin 6, the 0.05 m DEM generated from aerial imagery collected from a previous field trip to Carlsbad, NM using a UAV was used. Using the high-resolution DEM, two GIS-based sinkhole detection methods, Context and Morphology Based Extraction (CM-BE) and Multi-Depth Threshold Approach (MDTA), were tested and the results were compared with a sinkhole inventory that was created from GIS field mapping (Goodbar et al, 2020). Results from both methods either overlapped or were in close proximity to the sinkhole inventory. Additional depressions were also discovered using default parameters in each method. Upon further inspection, the Goodbar and MDTA methods produced point shapefiles whereas the CM-BE produced polygon shapefiles which are easier to use for refining the mesh. Therefore, the sinkholes mapped by the CM-BE method were used for refinement of the mesh of the sub-section of Basin 6 and were subsequently loaded onto TINerator.

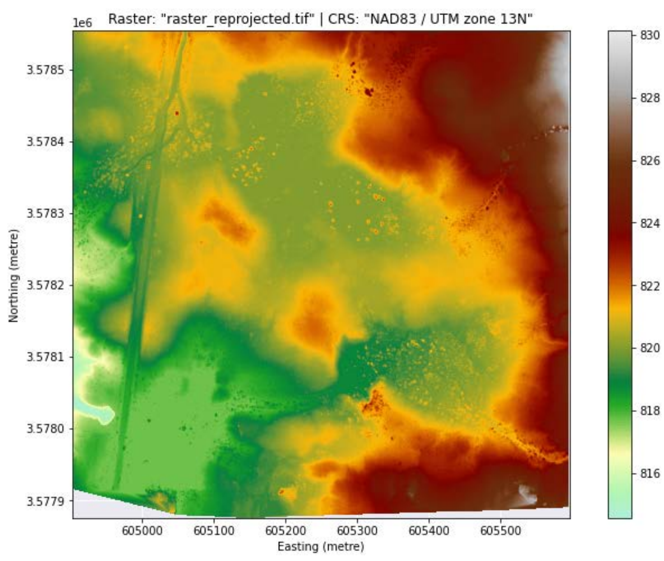

Figure 19. Section of Basin 6 DEM generated from UAV photogrammetry methods.

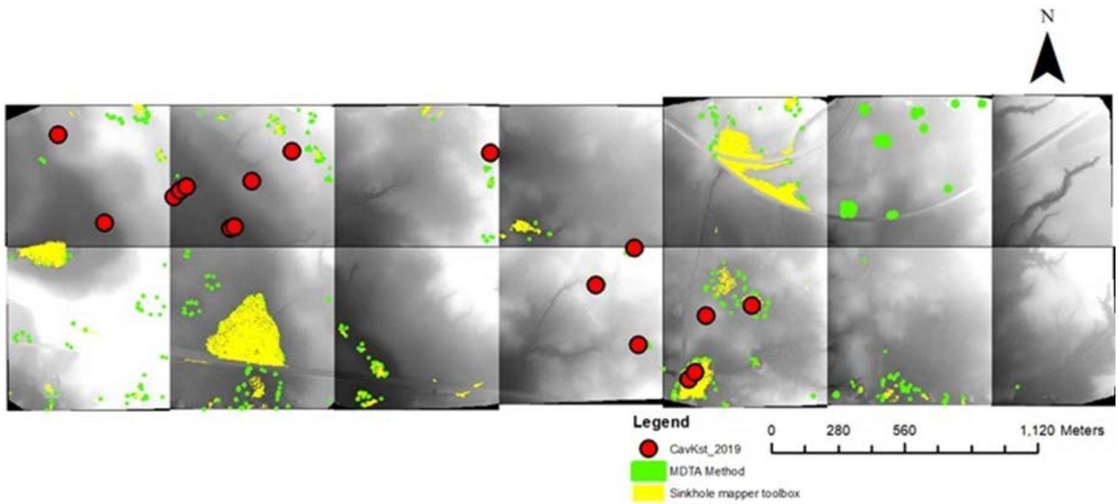

Figure 20. Map of sinkholes delineated in the Basin 6 Pilot Study Area from a high-resolution DEM using the Goodbar, MDTA, and CM-BE Methods.

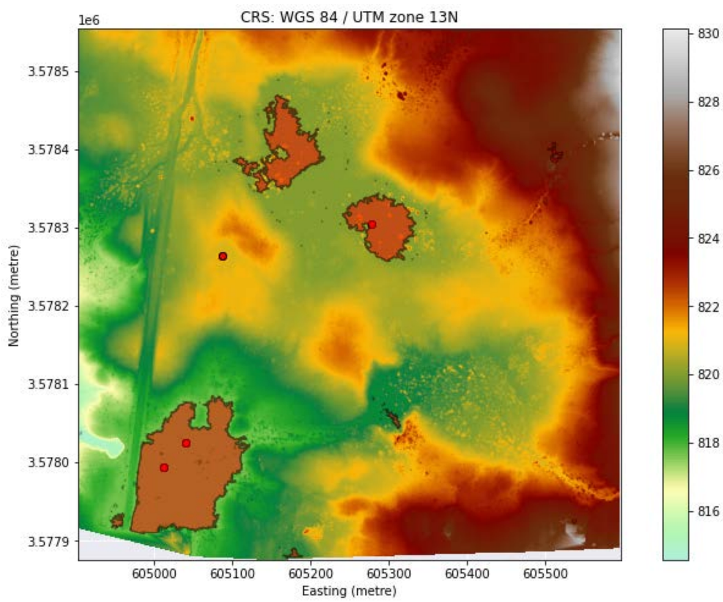

Figure 21. Map of sinkholes identified by Goodbar (red dots) and CM-BE Method (red polygons) on the DEM of a small sub-section of the Basin 6 Pilot Study Area.

3.2 Model Development

As mentioned previously, a Jupyter notebook was created for the sub-section of Basin 6 to document the process in TINerator. The high-resolution DEM was imported as well as the resulting sinkhole shapefile from the CM-BE method. The same workflow as applied in the Borden test case was applied which included loading the DEM, watershed delineation, creation of the faceset, and mesh generation. For the watershed delineation, a threshold of 40,000 was used as well as the D8 method provided by TINerator.

```
ws_flow = tin.gis.watershed_delineation(dem, threshold=40000)
ws_flow.plot(layers=[dem])
```

Figure 22. Watershed delineation function including a threshold of 40,000 for the sub-section of Basin 6 in TINerator.

Once the DEM was refined and processed, a triangulated mesh was generated. To generate the triangulated mesh, the following parameters were used:

- Min_edge_length= 0.01
- Max_edge_length=0.1
- Method= Jigsaw
    - Verbosity= 1
    - Mesh iterations = 20,000
    - Init_near= 0.1
    - Optm_qlim= 0.9
- Refinement feature= Watershed flowlines
- Scaling type= ‘Relative’

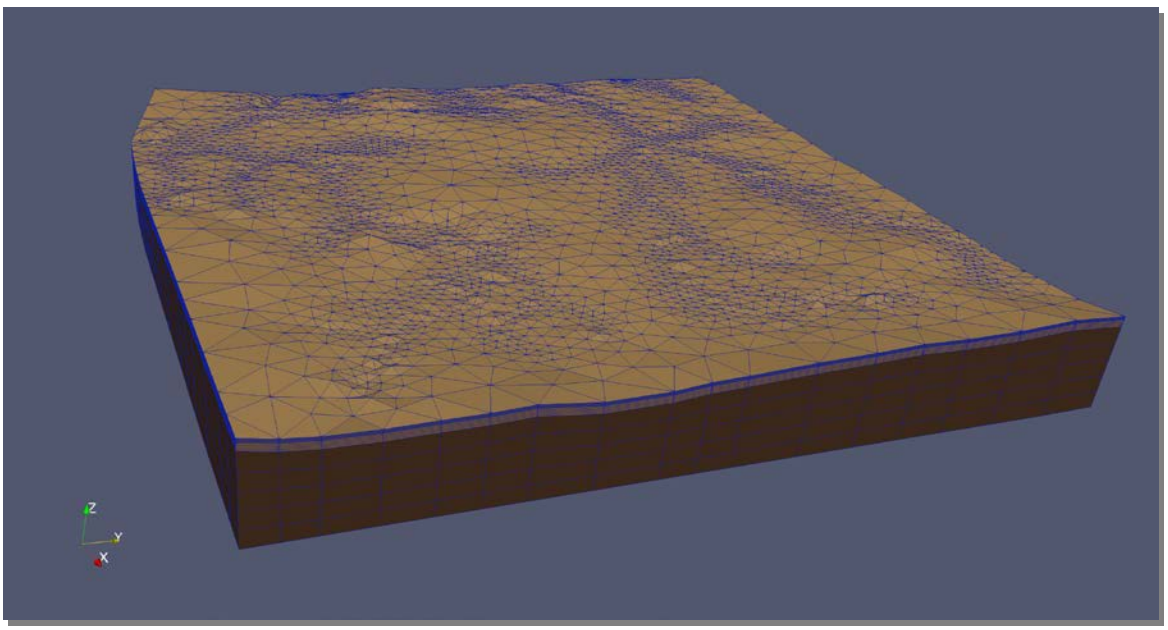

Figure 23. Triangulated mesh generated for a sub-section of Basin 6 and visualized using ParaView.

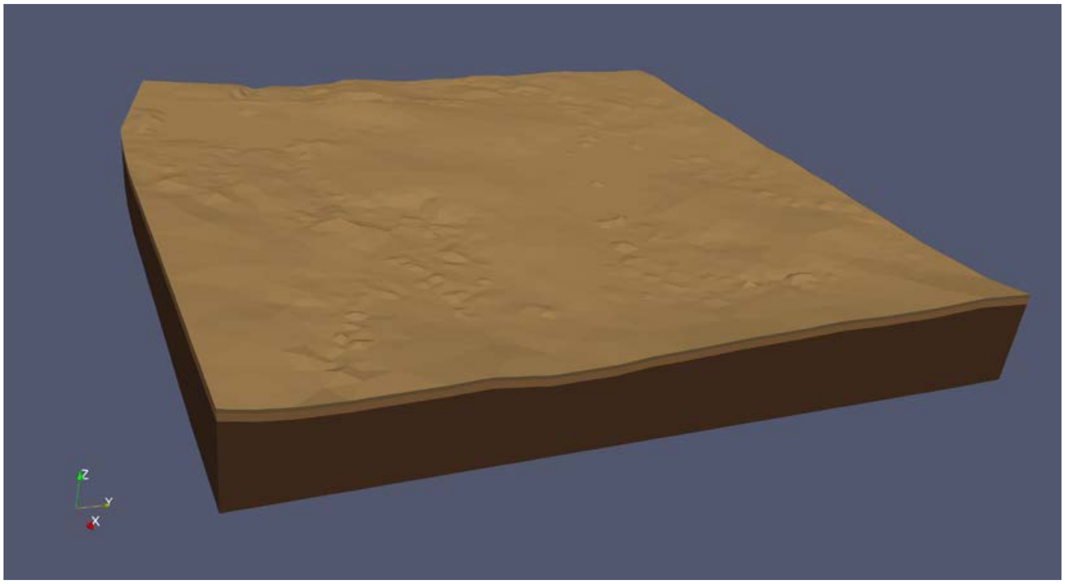

Figure 24. Surface mesh of a sub-section of Basin 6 with layer depths exaggerated for effect.

From the surface mesh, the volume mesh was generated to support integrated hydrology models that include the subsurface and so support the creation of the facesets, which are needed to specify boundary conditions. In addition to the auto-generated naïve facesets, the CM-BE sinkhole shapefile was imported so that these regions can be passed onto the ATS through the ExodusII mesh file. These regions will be used for model observations of infiltration rates in the sinkholes.

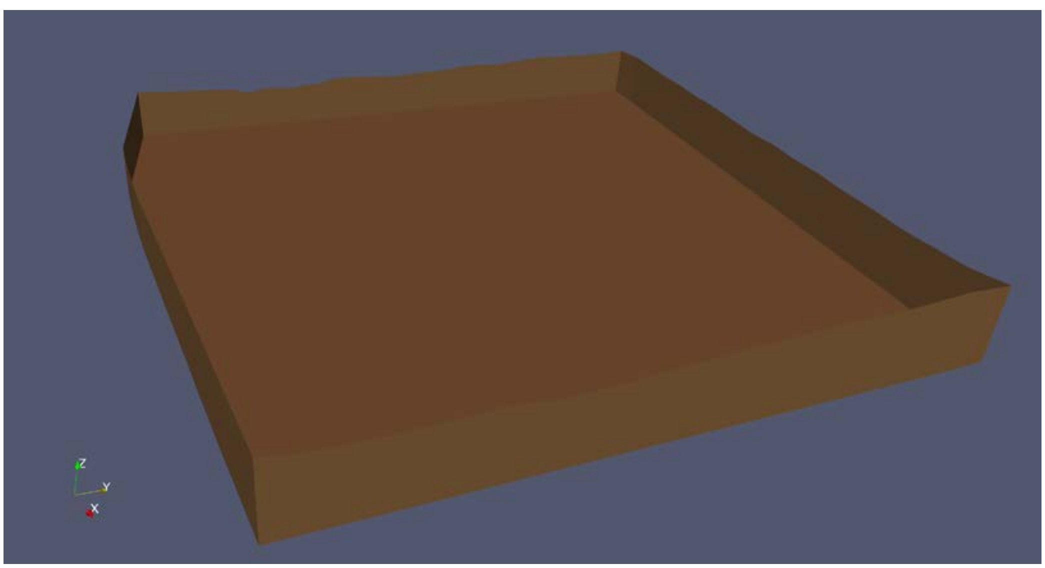

Figure 25. Highlighted facesets (bottom and sides) of a sub-section of Basin 6.

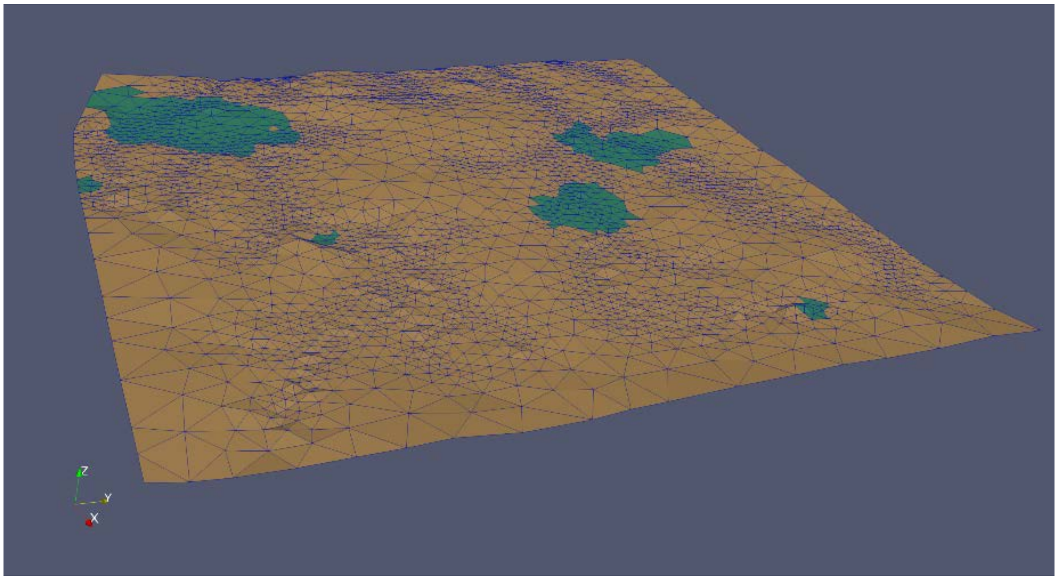

Figure 26. Sinkhole faceset on a triangulated surface mesh of a sub-section of Basin 6 mesh.

After the facesets were created, the volumetric mesh was exported in an ExodusII format. Further refinements will be made using the sinkhole shapefiles generated from the CM-BE method and applied to all of the Basin 6 study area. Further work will involve applying the complete workflow including developing the ATS input file and running ATS simulations. Visualization and analysis of the Basin 6 sub-section was not possible due to insufficient time and will be pursued in the future work to complete the workflow, and eventually apply it to the entire Basin 6 study area.

4. CONCLUSION

The overall objective of the project was to study the impact of surface features (such as sinkholes) in conjunction with soil properties and vegetation types, on the groundwater recharge over a range of storm events in Basin 6. This internship provided the knowledge and expertise in using several open-source software tools (TINerator, ParaView, VisIt, etc.) to generate meshes from DEM data, set up meteorological forcing data, and develop input files for the ATS to perform simulations. Using the Borden watershed allowed for an understanding of the general workflow from importing the DEM to running simulations using the ATS and which will now be applied to the entirety of Basin 6 using the high-resolution DEM generated from UAV photogrammetry methods as well as the sinkhole data generated from GIS-based methods. Future work will involve improving the mesh for Basin 6 and studying the impact of surface features (sinkholes, swallets, etc.) in conjunction with soil properties and vegetation types on the groundwater recharge using the ATS code.

5. REFERENCES

Ahrens, J., Geveci, B., & Law, C. (2005). Paraview: An end-user tool for large data visualization. The visualization handbook, 717(8).

Coon, E., Berndt, M., Garimella, R., Moulton, J. D., Manzini, G., & Painter, S. L. (2013, December). Computational advances in the Arctic terrestrial simulator: modeling permafrost degradation in a warming Arctic. In AGU Fall Meeting Abstracts (Vol. 2013, pp. DI31A-2195).

Coon, E. T., Moulton, J. D., & Painter, S. L. (2016). Managing complexity in simulations of land surface and near-surface processes. Environmental modelling & software, 78, 134-149.

Goodbar, A. K., Powers, D. W., Goodbar, J. R., & Holt, R. M. (2020). Karst and sinkholes at Nash Draw, southeastern New Mexico (USA).

Kollet, S., Sulis, M., Maxwell, R. M., Paniconi, C., Putti, M., Bertoldi, G., ... & Sudicky, E. (2017). The integrated hydrologic model intercomparison project, IH-MIP2: A second set of benchmark results to diagnose integrated hydrology and feedbacks. Water Resources Research, 53(1), 867-890.

Livingston, D. (2020). TINerator API Reference. Los Alamos National Laboratory.

APPENDIX A.

```xml
<!-- Begin: ATS Simulation Input -->
<ParameterList name="Main" type="ParameterList">

  <!-- Begin: Mesh -->
  <ParameterList name="mesh" type="ParameterList">
    <ParameterList name="domain">
      <Parameter name="mesh type" type="string" value="read mesh file" />
      <ParameterList name="read mesh file parameters" type="ParameterList">
        <Parameter name="file" type="string" value="borden_3-layers_unamed-sets_v1.exo" />
        <Parameter name="format" type="string" value="Exodus II" />
      </ParameterList>
    </ParameterList>
    <ParameterList name="surface">
      <Parameter name="mesh type" type="string" value="surface" />
      <ParameterList name="surface parameters" type="ParameterList">
        <Parameter name="surface sideset name" type="string" value="surface" />
      </ParameterList>
    </ParameterList>
  </ParameterList>
  <!-- End: Mesh -->

  <!-- Begin: Regions -->
  <ParameterList name="regions" type="ParameterList">

    <ParameterList name="computational domain" type="ParameterList">
      <ParameterList name="region: all" type="ParameterList">
      </ParameterList>
    </ParameterList>
    <ParameterList name="surface domain" type="ParameterList">
      <ParameterList name="region: all" type="ParameterList">
      </ParameterList>
    </ParameterList>

    <!-- ATS extracts the surface boundary (as a handy default option) from your mesh -->
    <ParameterList name="surface boundary" type="ParameterList">
      <ParameterList name="region: boundary" type="ParameterList">
      </ParameterList>
    </ParameterList>

    <!-- Top surface (face set) of the 3D mesh -->
    <ParameterList name="surface" type="ParameterList">
      <ParameterList name="region: labeled set" type="ParameterList">
        <Parameter name="label" type="string" value="1" />
        <Parameter name="file" type="string" value="borden_3-layers_unamed-sets_v1.exo" />
        <Parameter name="format" type="string" value="Exodus II" />
        <Parameter name="entity" type="string" value="face" />
      </ParameterList>
    </ParameterList>

    <!-- Outlet on the surface, derived from the 3D mesh -->
    <ParameterList name="stream outlet" type="ParameterList">
      <ParameterList name="region: labeled set" type="ParameterList">
        <Parameter name="label" type="string" value="3" />
        <Parameter name="file" type="string" value="borden_3-layers_unamed-sets_v1.exo" />
        <Parameter name="format" type="string" value="Exodus II" />
        <Parameter name="entity" type="string" value="face" />
      </ParameterList>
    </ParameterList>

  </ParameterList>
  <!-- End: Regions -->
```

```xml
<!-- Top-level execution control -->
<ParameterList name="cycle driver" type="ParameterList">

  <Parameter name="start time" type="double" value="0.0" />
  <Parameter name="start time units" type="string" value="s" />
  <Parameter name="end time" type="double" value="0.5" />
  <Parameter name="end time units" type="string" value="d" />

  <!-- PK Tree:  Surface Flow (only) -->
  <ParameterList name="PK tree" type="ParameterList">
    <ParameterList name="overland flow" type="ParameterList">
      <Parameter name="PK type" type="string" value="overland flow, pressure basis" />
    </ParameterList>
  </ParameterList>
</ParameterList>

<!-- Begin: Declarations of Multi-Process Couplers (MPCs) and Process Kernels (PKs) -->
<ParameterList name="PKs" type="ParameterList">

  <!-- Begin: PK::Overland Flow -->
  <ParameterList name="overland flow" type="ParameterList">
    <Parameter name="PK type" type="string" value="overland flow, pressure basis" />
    <Parameter name="primary variable key" type="string" value="surface-pressure" />
    <Parameter name="domain name" type="string" value="surface" />
    <Parameter name="source term" type="bool" value="true" />
    <Parameter name="initial time step" type="double" value="1.0" />
    <Parameter name="water source in meters" type="bool" value="true" />
    <Parameter name="source term is differentiable" type="bool" value="true" />
    <!-- tidal head boundary condition uses this information for the switch -->
    <Parameter name="min ponded depth for tidal bc" type="double" value="0.01" />
    <ParameterList name="verbose object" type="ParameterList">
      <Parameter name="verbosity level" type="string" value="low" />
    </ParameterList>

    <ParameterList name="diffusion" type="ParameterList">
      <Parameter name="discretization primary" type="string" value="fv: default" />
    </ParameterList>

    <ParameterList name="diffusion preconditioner" type="ParameterList">
      <Parameter name="Newton correction" type="string" value="true Jacobian" />
    </ParameterList>

    <ParameterList name="inverse" type="ParameterList">
      <Parameter name="preconditioning method" type="string" value="block ilu" />
      <Parameter name="iterative method" type="string" value="gmres" />
      <ParameterList name="verbose object" type="ParameterList">
        <Parameter name="verbosity level" type="string" value="low" />
      </ParameterList>
      <ParameterList name="gmres parameters" type="ParameterList">
        <Parameter name="preconditioning strategy" type="string" value="left" />
        <Parameter name="error tolerance" type="double" value="1e-06" />
        <Parameter name="convergence criteria" type="Array(string)" value="{relative residual,make
        <Parameter name="maximum number of iteration" type="int" value="80" />
      </ParameterList>
    </ParameterList>

    <ParameterList name="time integrator" type="ParameterList">
      <Parameter name="extrapolate initial guess" type="bool" value="true" />
      <Parameter name="solver type" type="string" value="nka_bt_ats" />
      <Parameter name="timestep controller type" type="string" value="smarter" />
      <ParameterList name="nka_bt_ats parameters" type="ParameterList">
        <Parameter name="nka lag iterations" type="int" value="2" />
        <Parameter name="max backtrack steps" type="int" value="5" />
        <Parameter name="backtrack lag" type="int" value="0" />
        <Parameter name="backtrack factor" type="double" value="0.5" />
        <Parameter name="backtrack tolerance" type="double" value="1.0" />
        <Parameter name="nonlinear tolerance" type="double" value="1e-06" />
        <Parameter name="diverged tolerance" type="double" value="100000.0" />
        <Parameter name="limit iterations" type="int" value="21" />
        <ParameterList name="verbose object" type="ParameterList">
          <Parameter name="verbosity level" type="string" value="low" />
        </ParameterList>
      </ParameterList>
      <ParameterList name="verbose object" type="ParameterList">
        <Parameter name="verbosity level" type="string" value="high" />
      </ParameterList>
      <ParameterList name="timestep controller smarter parameters" type="ParameterList">
        <Parameter name="max iterations" type="int" value="18" />
        <Parameter name="min iterations" type="int" value="10" />
        <Parameter name="time step reduction factor" type="double" value="0.5" />
        <Parameter name="time step increase factor" type="double" value="1.25" />
```

```xml
        <Parameter name="max time step" type="double" value="10000000000.0" />
        <Parameter name="min time step" type="double" value="1e-10" />
        <Parameter name="growth wait after fail" type="int" value="2" />
        <Parameter name="count before increasing increase factor" type="int" value="2" />
      </ParameterList>
    </ParameterList>

    <!-- PK:: Overland Flow - Initial Condition -->
    <ParameterList name="initial condition" type="ParameterList">
      <ParameterList name="function" type="ParameterList">
        <ParameterList name="initial pressure" type="ParameterList">
          <Parameter name="region" type="string" value="surface domain" />
          <Parameter name="components" type="Array(string)" value="{cell,boundary_face}" />
          <ParameterList name="function" type="ParameterList">
            <ParameterList name="function-constant" type="ParameterList">
              <Parameter name="value" type="double" value="101325.0" />
            </ParameterList>
          </ParameterList>
        </ParameterList>
      </ParameterList>
    </ParameterList>

    <!-- PK:: Overland Flow - Boundary Conditions
               - Natural boundary conditions (Neumann/No Flow) are used on any face not specified
               - Define critical depth on the seaward side of the domain, defined by the "tidal river
    -->
    <ParameterList name="boundary conditions" type="ParameterList">
      <ParameterList name="critical depth" type="ParameterList">
        <ParameterList name="outlet" type="ParameterList">
          <Parameter name="regions" type="Array(string)" value="{stream outlet}" />
        </ParameterList>
      </ParameterList>
    </ParameterList>

  </ParameterList>
  <!-- End: PK::Overland Flow -->

</ParameterList>
<!-- End: Declarations of Multi-Process Couplers (MPCs) and Process Kernels (PKs) -->


<!-- Begin: State - Field declarations and evaluators -->
<ParameterList name="state" type="ParameterList">

  <ParameterList name="field evaluators" type="ParameterList">

    <ParameterList name="surface-water_content" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="overland pressure water content" />
    </ParameterList>

    <ParameterList name="surface-ponded_depth" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="ponded depth" />
    </ParameterList>

    <ParameterList name="surface-ponded_depth_bar" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="ponded depth" />
      <Parameter name="ponded depth bar" type="bool" value="true" />
      <Parameter name="height key" type="string" value="ponded_depth_bar" />
    </ParameterList>

    <ParameterList name="surface-water_source" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="independent variable" />
      <ParameterList name="function" type="ParameterList">
        <ParameterList name="domain rain" type="ParameterList">
          <Parameter name="region" type="string" value="surface domain" />
          <Parameter name="component" type="string" value="cell" />
          <ParameterList name="function">
            <ParameterList name="function-tabular" type="ParameterList">
                          <!-- Rain-fall:
                               - time [0,12][h] = [0,43200][s]
                               - rate 18 [mm/h] = 5.0e-6 [m/s]  - very heavy rain, or heavy shower
                               - (rain fall rate: subjective names https://water.usgs.gov/edu/activit
                          -->
              <Parameter name="x values" type="Array(double)" value="{0.0,43200.0}" />
              <Parameter name="y values" type="Array(double)" value="{5e-06,0.0}" />
              <Parameter name="forms" type="Array(string)" value="{constant}" />
            </ParameterList>
          </ParameterList>
        </ParameterList>
      </ParameterList>
    </ParameterList>
```

```
    <ParameterList name="surface-source_molar_density" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="eos" />
      <Parameter name="EOS basis" type="string" value="molar" />
      <Parameter name="molar density key" type="string" value="surface-source_molar_density" />
      <ParameterList name="EOS parameters" type="ParameterList">
        <Parameter name="EOS type" type="string" value="constant" />
        <Parameter name="density [kg/m^3]" type="double" value="1000.0" />
      </ParameterList>
      <ParameterList name="verbose object" type="ParameterList">
        <Parameter name="verbosity level" type="string" value="low" />
      </ParameterList>
    </ParameterList>

    <ParameterList name="surface-molar_density_liquid" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="eos" />
      <Parameter name="EOS basis" type="string" value="both" />
      <Parameter name="molar density key" type="string" value="surface-molar_density_liquid" />
      <Parameter name="mass density key" type="string" value="surface-mass_density_liquid" />
      <ParameterList name="EOS parameters" type="ParameterList">
        <Parameter name="EOS type" type="string" value="constant" />
        <Parameter name="density [kg/m^3]" type="double" value="1000.0" />
      </ParameterList>
      <ParameterList name="verbose object" type="ParameterList">
        <Parameter name="verbosity level" type="string" value="low" />
      </ParameterList>
    </ParameterList>

    <ParameterList name="surface-manning_coefficient" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="independent variable" />
      <Parameter name="constant in time" type="bool" value="true" />
      <ParameterList name="function" type="ParameterList">
        <ParameterList name="domain" type="ParameterList">
          <Parameter name="region" type="string" value="surface domain" />
          <Parameter name="components" type="Array(string)" value="{cell,boundary_face}" />
          <ParameterList name="function" type="ParameterList">
            <ParameterList name="function-constant" type="ParameterList">
              <Parameter name="value" type="double" value="0.0036" />
            </ParameterList>
          </ParameterList>
        </ParameterList>
      </ParameterList>
    </ParameterList>

    <ParameterList name="surface-elevation" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="meshed elevation" />
      <Parameter name="dynamic mesh" type="bool" value="false" />
      <ParameterList name="verbose object" type="ParameterList">
        <Parameter name="verbosity level" type="string" value="low" />
      </ParameterList>
    </ParameterList>

    <ParameterList name="surface-effective_pressure" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="effective_pressure" />
    </ParameterList>

    <ParameterList name="surface-temperature" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="independent variable" />
      <Parameter name="constant in time" type="bool" value="true" />
      <ParameterList name="function" type="ParameterList">
        <ParameterList name="domain" type="ParameterList">
          <Parameter name="region" type="string" value="surface domain" />
          <Parameter name="components" type="Array(string)" value="{cell,boundary_face}" />
          <ParameterList name="function" type="ParameterList">
            <ParameterList name="function-constant" type="ParameterList">
              <Parameter name="value" type="double" value="274.65" />
            </ParameterList>
          </ParameterList>
        </ParameterList>
      </ParameterList>
    </ParameterList>

    <ParameterList name="surface-overland_conductivity" type="ParameterList">
      <Parameter name="field evaluator type" type="string" value="overland conductivity" />
      <ParameterList name="overland conductivity model" type="ParameterList">
        <Parameter name="Manning exponent" type="double" value="0.666666666667" />
        <Parameter name="slope regularization epsilon" type="double" value="0.01" />
      </ParameterList>
    </ParameterList>

  </ParameterList>
  <ParameterList name="initial conditions" type="ParameterList">
```

```
      <ParameterList name="atmospheric_pressure" type="ParameterList">
        <Parameter name="value" type="double" value="101325.0" />
      </ParameterList>

      <ParameterList name="gravity" type="ParameterList">
        <Parameter name="value" type="Array(double)" value="{0.0,0.0,-9.80665}" />
      </ParameterList>

    </ParameterList>
  </ParameterList>
  <!-- End: State - Field declarations and evaluators -->


  <!-- Begin: Visualization -->
  <ParameterList name="visualization">
    <ParameterList name="surface" type="ParameterList">
      <Parameter name="file name base" type="string" value="visdump_surface" />
      <Parameter name="times start period stop" type="Array(double)" value="{0.0,600.0,-1.0}" />
    </ParameterList>
  </ParameterList>
  <!-- End: Visualization -->

</ParameterList>
```