



ASCE EWRI, SRH2D short course, Austin, TX, 2015

# Pre- and Post-processing of SRH-2D Results using Open Source Software

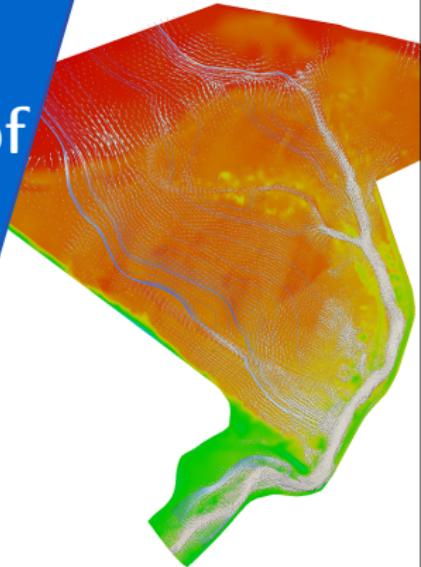
Xiaofeng Liu, Ph.D., P.E.

Assistant Professor

Department of Civil and Environmental Engineering

Pennsylvania State University

[xliu@engr.psu.edu](mailto:xliu@engr.psu.edu)



# What will be covered?

2 / 53

- ▶ Visualization using Paraview and other software
- ▶ Mesh Generation Using Gmsh

General introduction

Visualization of SRH-2D Results using ParaView

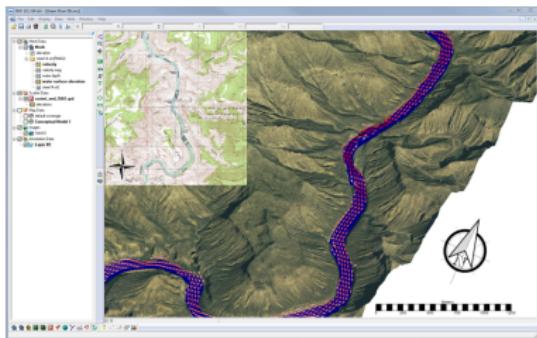
Mesh generation using Gmsh

# Why open source software for SRH2D?

4 / 53

What SRH2D uses now:

- ▶ SRH2D needs SMS for preprocessing and Tecplot for postprocessing
- ▶ Both cost \$\$\$



(a) SMS



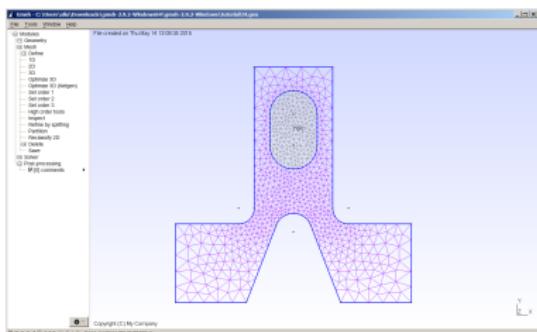
(b) Tecplot

# Why open source software for SRH2D?

5 / 53

Other (better) alternatives:

- ▶ For mesh generation:
  - Any 2D mesh generator will do: format conversion to SMS.
  - Gmsh has good reputation, well maintained, and large user base.
- ▶ For post processing:
  - ParaView (version 4.1 and later)
- ▶ Scripting and customization with Python



(a) Gmsh



(b) Paraview

# Preparation for this part of the short course

6 / 53

Some tools are needed:

- ▶ Installation of SRH2D v3.
- ▶ Installation of Gmsh
- ▶ Installation of ParaView (version 4.1 and later)
- ▶ Installation of Python: I use Continuum's Anaconda, which provides Python, IPython and others.

Pre-downloaded packages for Gmsh, Paraview, Anaconda for you to install to your laptop.

Postprocessing is the step during or after the simulation based on the simulation results to

- ▶ produce derivative data: e.g., calculate vorticity, turbulence quantities, max, min, average, integrate along a line
- ▶ get subset of data: sample, probe, etc.
- ▶ data conversion: to other formats
- ▶ ...

How to visualize flow field?

- ▶ Velocity vector
- ▶ Streamlines (instantaneous tangent lines of  $\mathbf{u}$ ), streaklines (dye test), pathlines (particle trajectory)
- ▶ Vorticity:

# Simple data processing and plotting

We have used the following tools to do data analysis and plotting

- ▶ Gnuplot
- ▶ Python and Matplotlib
- ▶ Octave

- ▶ Gnuplot is an open source, cross-platform plotting tool.
- ▶ <http://www.gnuplot.info/>
- ▶ It can generate 1D, 2D, and 3D plots.
- ▶ Support both interactive and scripting
- ▶ Support different output formats: pdf, png, jpeg, Latex, svg, etc.
- ▶ To use Latex for annotation is possible, but needs some extra steps.
- ▶ Scripting is good to keep all the generated figures have consistent style (font, size, color, etc.)

- ▶ Python and Matplotlib
- ▶ <https://www.python.org/>
- ▶ <http://matplotlib.org/>
- ▶ Python is more powerful than just plotting.

# A short tutorial of Paraview

- ▶ We will use Paraview v4.1 and above: [www.paraview.org](http://www.paraview.org)
- ▶ Paraview is an open-source, multi-platform parallel data analysis and visualization software
- ▶ Built up the VTK (Visualization Toolkit) library
- ▶ It supports a wide variety of data formats
  - Tecplot ← SRH2D output format!!!
  - Structure grid
  - Unstructured grid
  - Images
  - Multi-block
  - AMR
  - Time series are automatically supported

A note on the downside of Tecplot format: it can only save scalars, not vectors. So we have to tell the software which variables are  $U_x$  and  $U_y$  to plot velocity.

# A short tutorial of Paraview

- ▶ A rich selection of visualization tools
  - Contour
  - Isosurface
  - Cutting plane
  - Vectors (Glyphs)
  - Streamlines
  - Volume rendering
  - Clipping
  - ...
- ▶ Derived variables: Calculator and other filters
- ▶ Support of Python for scripting
- ▶ Support of export to different formats
- ▶ Support for parallel run

# A short tutorial of Paraview

- ▶ VTK format (<http://www.vtk.org/VTK/img/file-formats.pdf>)
  - structured points
  - structured grid
  - rectlinear grid
  - polygonal data
  - unstructured grid
  - ...
- ▶ VTK can store scalar, vector, tensor, etc.
- ▶ VTK has several formats: Legacy, and more recently XML
- ▶ Python has good support for VTK data format. A good selection of libraries.
- ▶ Paraview can be easily extended: for example, read a customized format

# A short tutorial of Paraview

The concept of pipeline:

- ▶ All processing operations on data set will produce new data set
- ▶ The new data set can be further processed (pipeline)
- ▶ Example:
  - Operation 1: Extract a contour surface
  - Operation 2: Plot velocity vector on the contour (glyphs)
  - ...
- ▶ With the combination of different operations, the result can be very complicated.
- ▶ For a list of all the operations (filters) available, have a look at the content inside menu filters

# A short tutorial of Paraview

15 / 53

Scripting using Python:

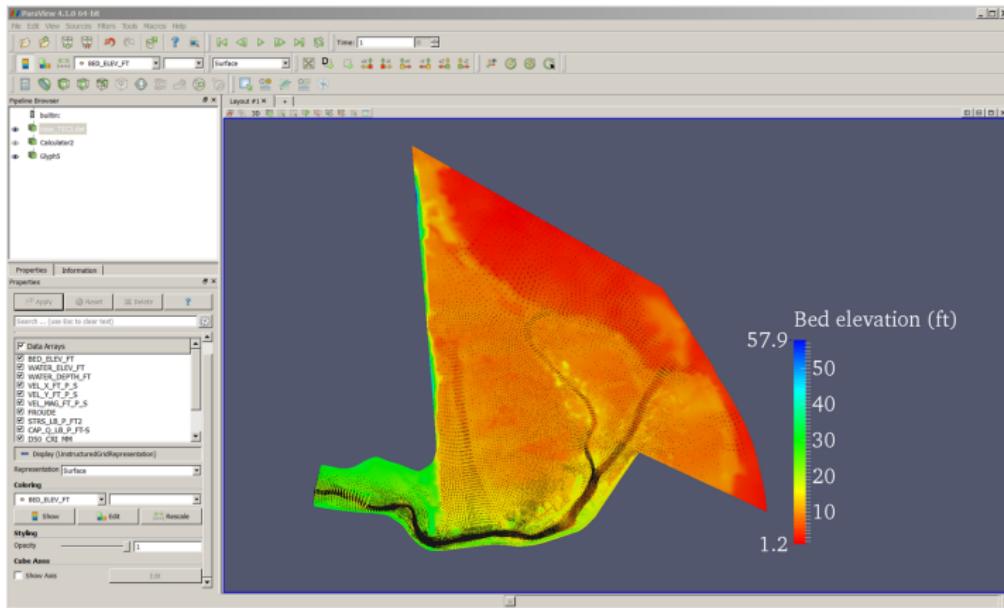
- ▶ The purpose: automate the process
- ▶ With repeated simulations, a script can be written and save a lot of labor

More tutorials can be found at:

- ▶ [http://www.paraview.org/Wiki/The\\_ParaView\\_Tutorial](http://www.paraview.org/Wiki/The_ParaView_Tutorial)
- ▶ [http://www.paraview.org/Wiki/Advanced\\_Tips\\_and\\_Tricks](http://www.paraview.org/Wiki/Advanced_Tips_and_Tricks)

# SRH-2D result visualization in ParaView

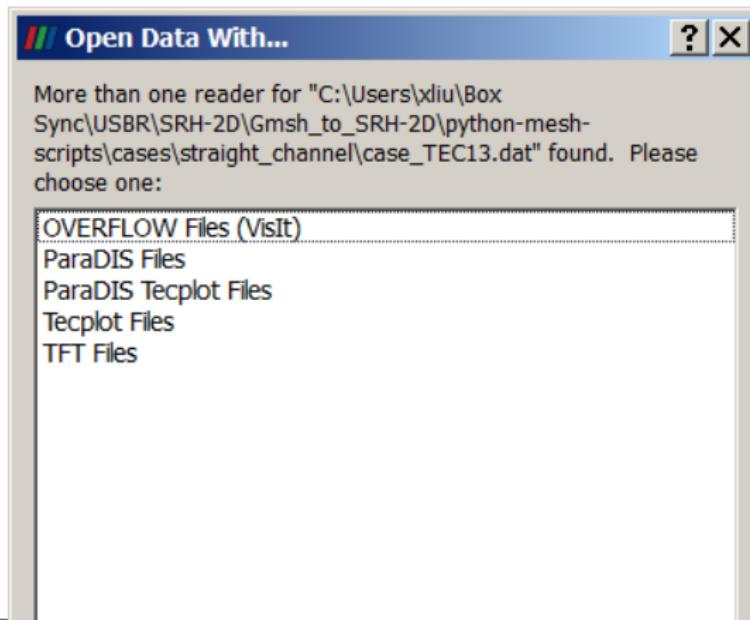
16 / 53



# SRH-2D result visualization in ParaView

To load result,

- ▶ simple do File → Open File and load the \_TECn.dat file.
- ▶ choose Tecplt file format and then click ok.
- ▶ Due to the way SRH-2D names the output file, the Tecplot output is nicely arranged in order. So you can open all the transient result file at once or simply select a subset.



We will demonstrate the following common tasks:

- ▶ Contour plot
- ▶ Clipping and threshold
- ▶ Calculator: e.g., combine velocity components into a vector (for vector plotting)
- ▶ Streamlines
- ▶ Warp (exaggeration in the vertical direction)
- ▶ Animation (both transient and static)
- ▶ Annotation and legend
- ▶ Export high quality figures and movies

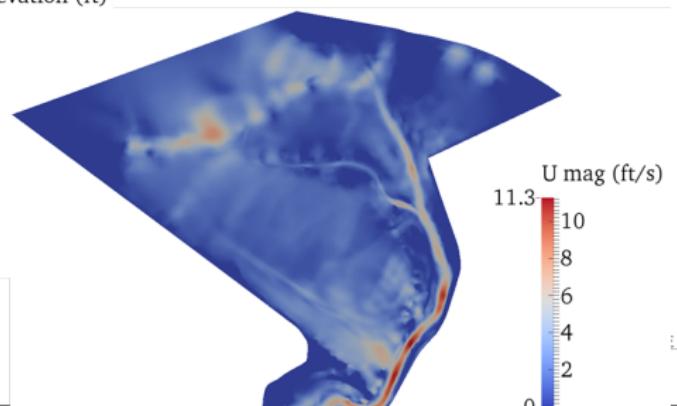
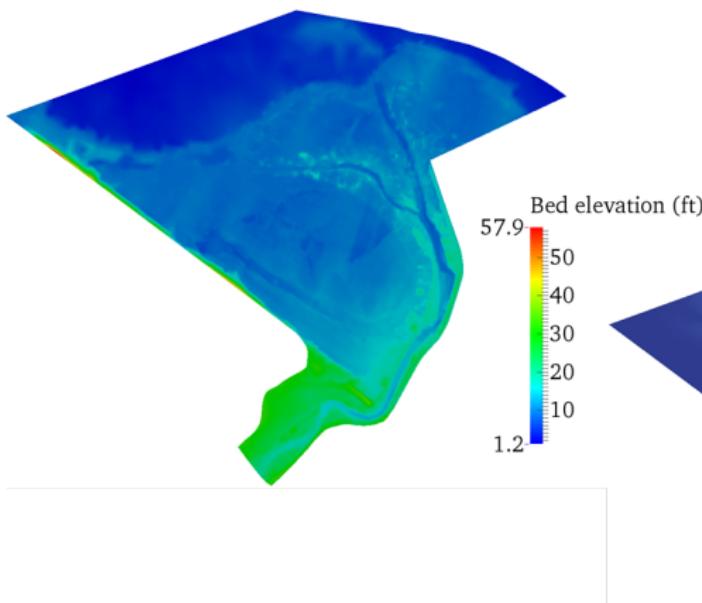
We will use the Dungeness River example in the Flow folder. The tutorial comes with some intermediate step results. Or you can run the case by yourself and generate some results.

# SRH-2D result visualization in ParaView

19 / 53

Contour plot:

- ▶ Load the data in Tutorial: Flow Modeling Cases → C2\_Tutorial\_Dungeness\_River.
- ▶ Practice the mouse usage: Left button (move or rotate depending on 2D or 3D view), Middle button (rotate), Left button (zoom)
- ▶ Edit legend, choose preset color scales

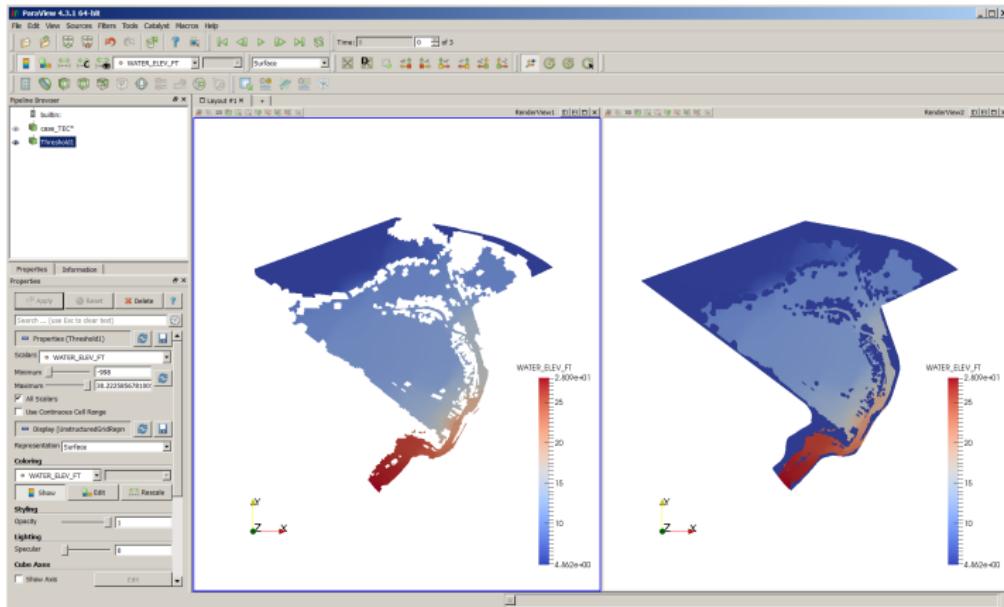


# SRH-2D result visualization in ParaView

20 / 53

## Clipping and threshold:

- ▶ Look at the Information pane and notice the range of each variables
- ▶ The water surface elevation has values of  $-999$  to signal the dry land.
- ▶ But you don't want to plot  $-999$  in the contour.
- ▶ To get rid of them, you can use the Threshold filter and use say  $-998$  as the lower threshold.

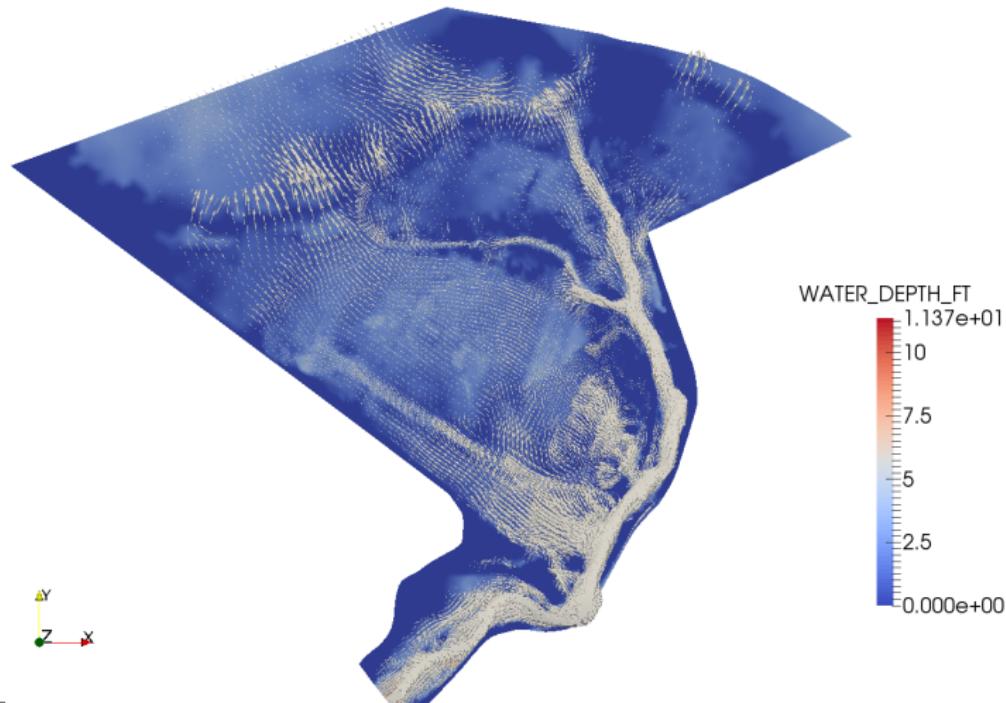


# SRH-2D result visualization in ParaView

21 / 53

## Calculator:

- ▶ You can use calculator to generate more variables based on existing ones
- ▶ We will demonstrate how to create the velocity vector.
- ▶ Vector plot is called **Glyph** in ParaView

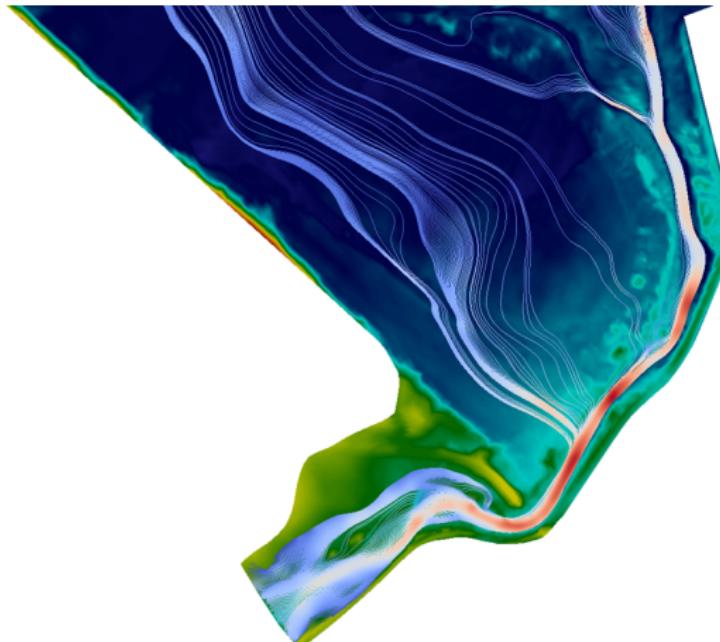


# SRH-2D result visualization in ParaView

22/53

## Streamlines:

- ▶ The streamline filter has two seeding modes: from point or line
- ▶ It need velocity vector information
- ▶ Make sure the line or the point intersection with the domain
- ▶ Parameters control how the integration in time is done to get the streamline

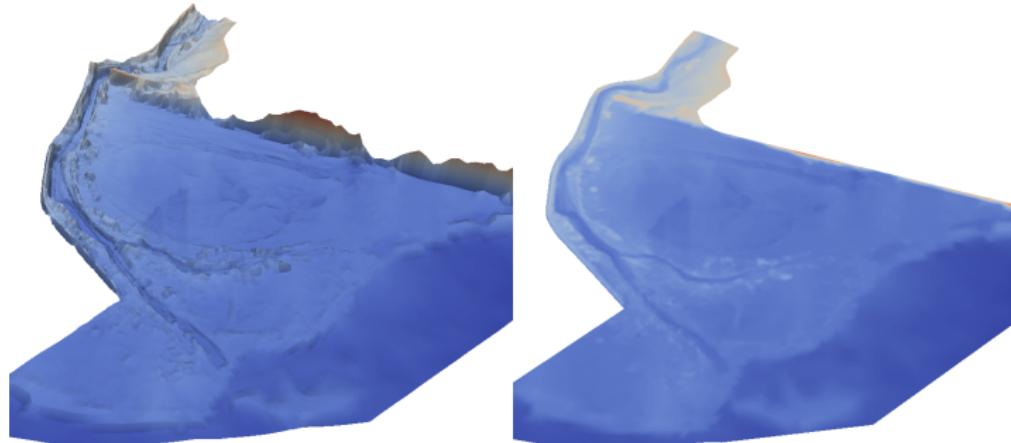


# SRH-2D result visualization in ParaView

23 / 53

## Warp:

- ▶ The mesh can be exaggerated in certain ways
- ▶ We demonstrate the vertical (elevation) exaggeration (by 10 times)
- ▶ Turn into 3D view
- ▶ Select the source (say the Tecplot data you loaded)
- ▶ Select Filter → Alphabetical → Warp by scalar.
- ▶ Select the bed elevation as the scalar to warp and use a scalar factor of 10

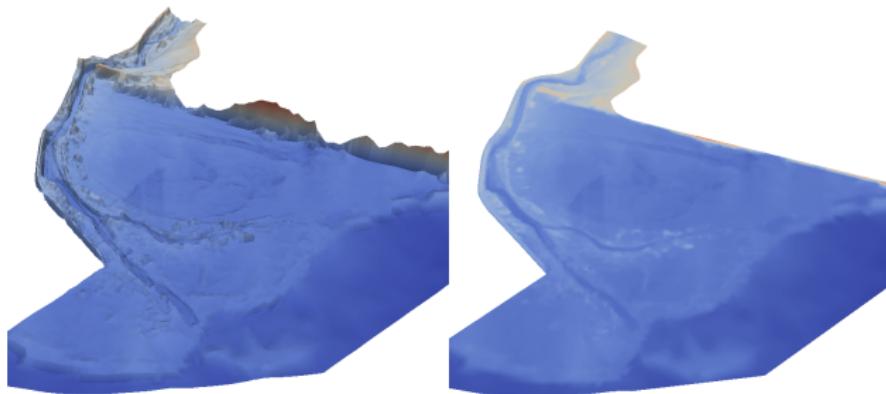


# SRH-2D result visualization in ParaView

24 / 53

Camera link and comparative view:

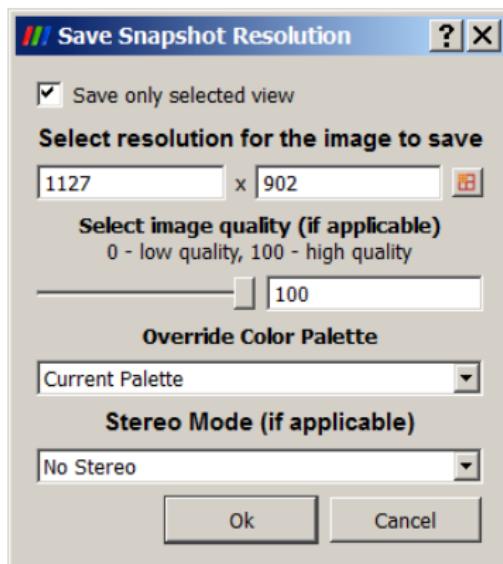
- ▶ As in the preview slide, the views on both sides are better to be the same for comparison.
- ▶ To do that, you need to split the window and create another view.
- ▶ Click on one of the views to select it
- ▶ Select Tools → Add Camera Link ...
- ▶ Click on another view which you like to link to
- ▶ Now the two views are linked.



# SRH-2D result visualization in ParaView

Export high quality images:

- ▶ File → Save Screenshot
- ▶ Increase the resolution and lock the aspect ratio
- ▶ If you have multiple views, you can export the selected view or all views.
- ▶ Click OK, you will specify the location and the export file name, as well as the format (PNG, BMP, PDF, etc.)



# SRH-2D result visualization in ParaView

26 / 53

Animate a static vector field:

Play the static animation.

Animate a static vector field:

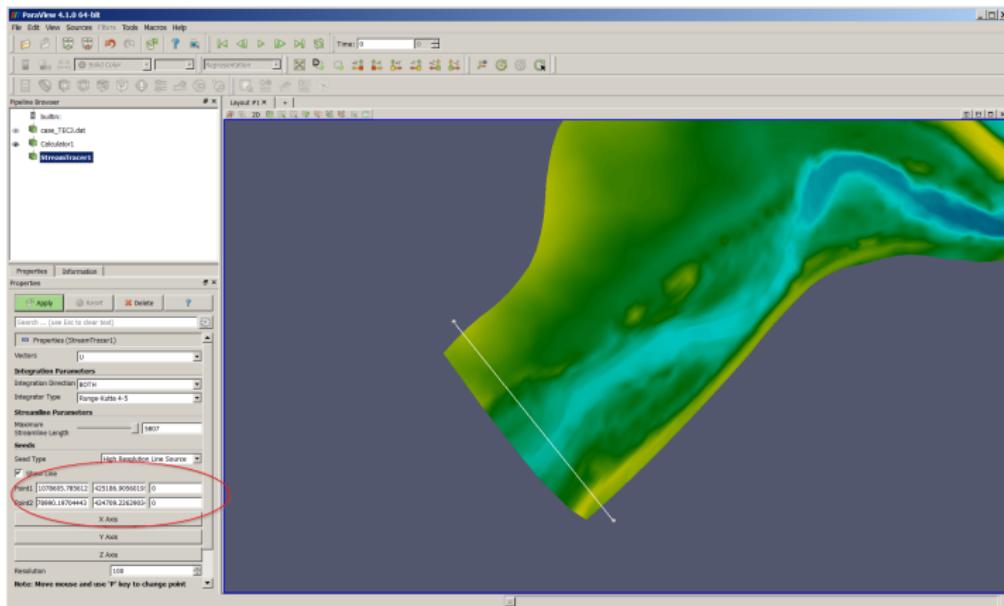
- ▶ If you have a vector field (solution at one time step), you can animate it.
- ▶ The basic idea is to place points or vectors on the streamlines and animate them down the streamline.
- ▶ It is good to see the flow paths and compare the residence time.
- ▶ Create the vector field  $U$  using the filter Calculator.
- ▶ Create streamlines as before.
  - Use a line source and make sure the z coordinates of the two points are zero.
  - Increase the Maximum Streamline Length to a large value, say 100,000.
  - Choose the Integration Direction as FORWARD
- ▶ Apply the Contour filter to the streamline on Integration Time. Use Add a range of values to add contour levels.

Animate a static vector field:

- ▶ Apply the Glyph filter to the contour with the created velocity  $U$ . Uncheck the Mask Points and Random Mode. I also turn off the Scale Mode and set a constant Scale Factor as 200.
- ▶ Turn on the Animation View: View → Animation View
  - Mode: Sequence
  - No. Frames: 100
  - Change the Pulldown box near the blue + to be Contour you generated, then click the + button.
  - Now, you can play the animation by clicking the Play button to top.

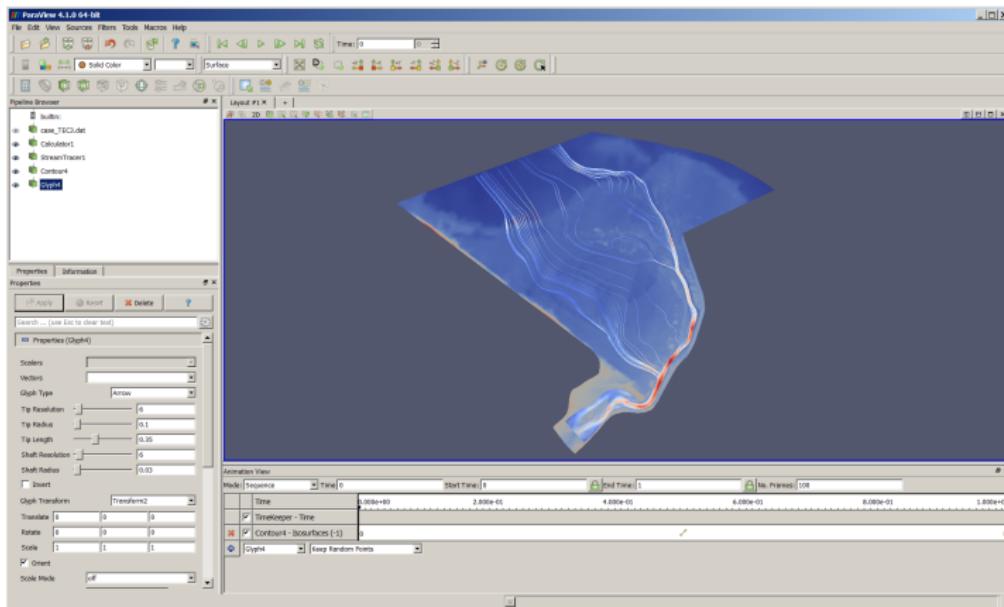
# SRH-2D result visualization in ParaView

29 / 53



# SRH-2D result visualization in ParaView

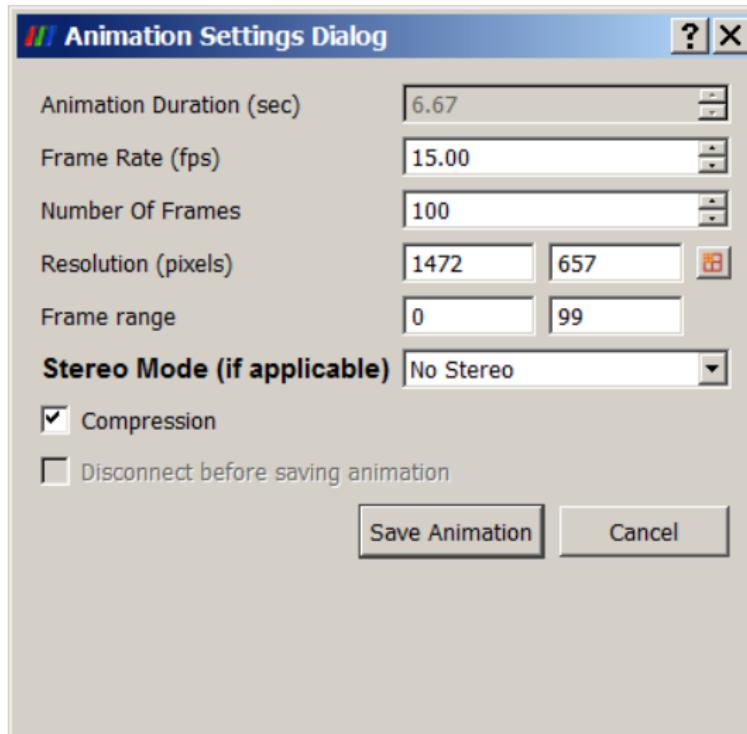
30 / 53



# SRH-2D result visualization in ParaView

Animate a static vector field:

- ▶ To save the animation: File → Save Animation
- ▶ Set the parameters to have high quality video

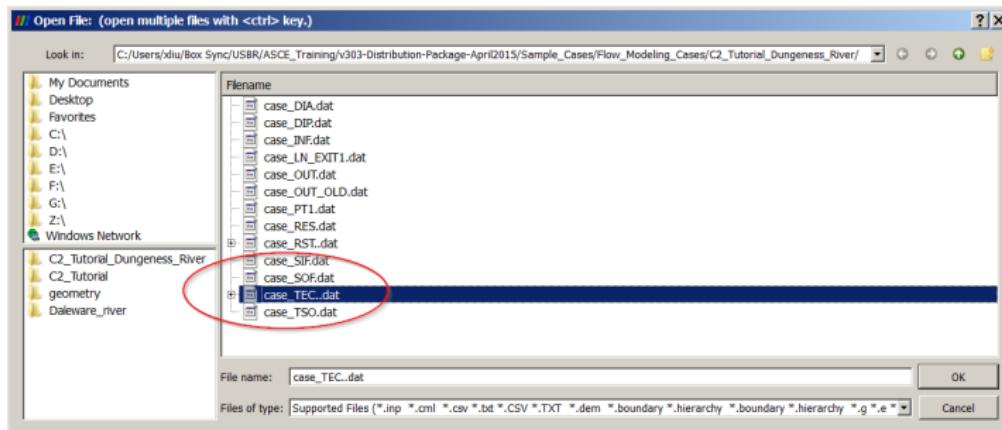


# SRH-2D result visualization in ParaView

32 / 53

Animate a time sequence of field:

- ▶ Much simpler
- ▶ Just load all the Tecplot result files by selecting the whole (not the individual file).
- ▶ Set up the view (contour, vector, etc.)
- ▶ Click Play button, or save to video file

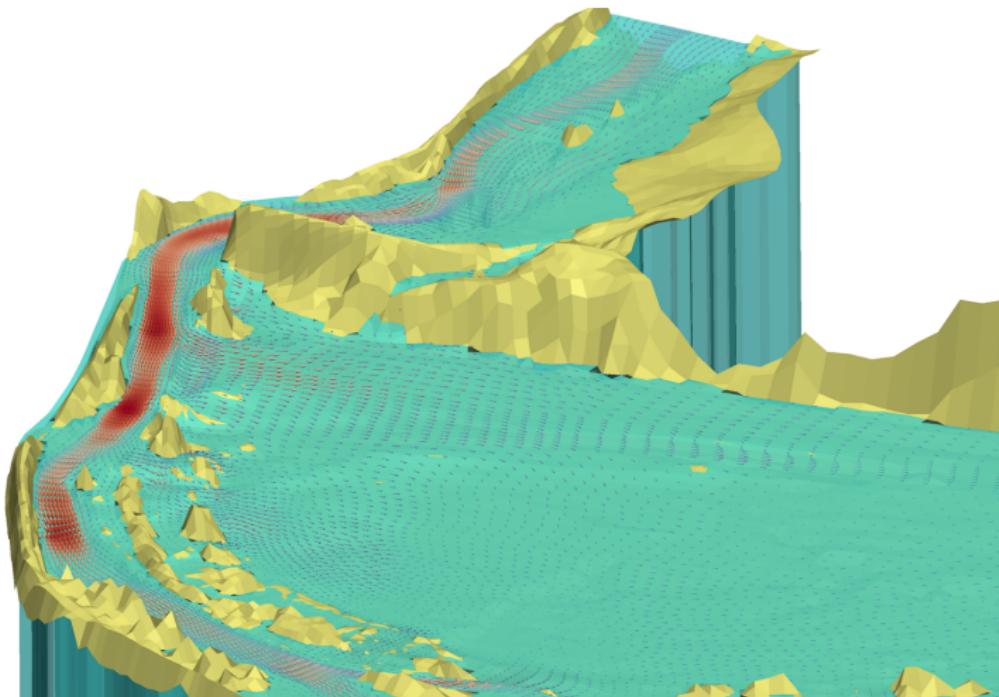


# SRH-2D result visualization in ParaView

33 / 53

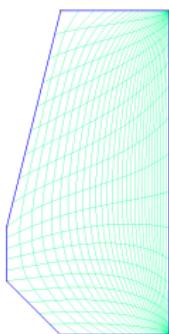
A take-home exercise: To generate a 3D inundation view of the SRH-2D result.

- ▶ Filters used:
  - Contour (with some transparency), Glyph, Threshold
  - Warp

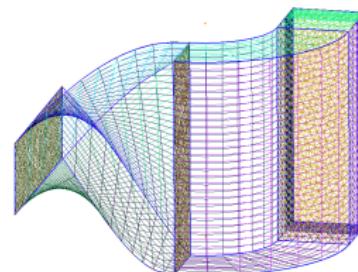


- ▶ Why Gmsh (<http://geuz.org/gmsh/>)?

- a free and robust mesh generator for both 2D and 3D unstructured grid.
- open source and cross-platform (Windows, Linux, and Mac)
- Easy user interface and scripting (.geo file)
- Export to Gmsh format, which is readable by many software
- Also export other formats: e.g., vtu for ParaView

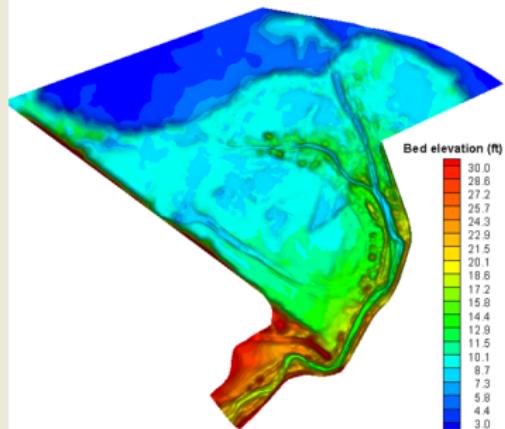
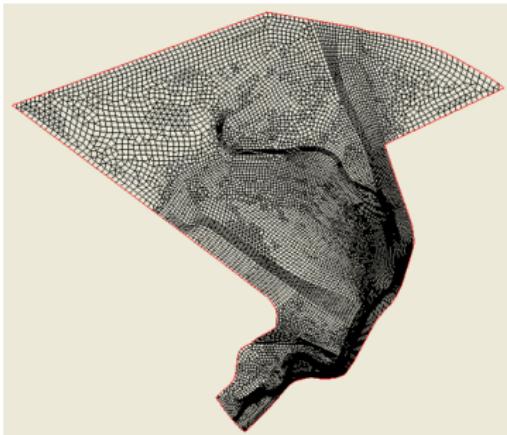


(a) SMS



(b) Tecplot

- ▶ Drawback:
  - No direct support for SRH2D (SMS format)
  - No support for boundary delineation
  - No support for elevation assignment (Z coordinate)
- ▶ All these drawbacks can be overcome
- ▶ More user friendly interface can be developed in open source GIS platforms such as QGIS and GRASS.



General procedures to use Gmsh for SRH2D:

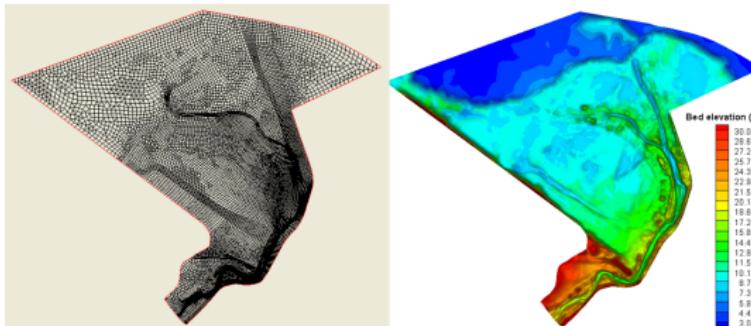
1. In Gmsh:

- Define control coordinates and boundaries of the domain:
- Define meshing control
- Generate and export 2D mesh (in Gmsh format):

2. Prepare the bathymetry data ( $x, y, z$  point cloud format) covering the domain

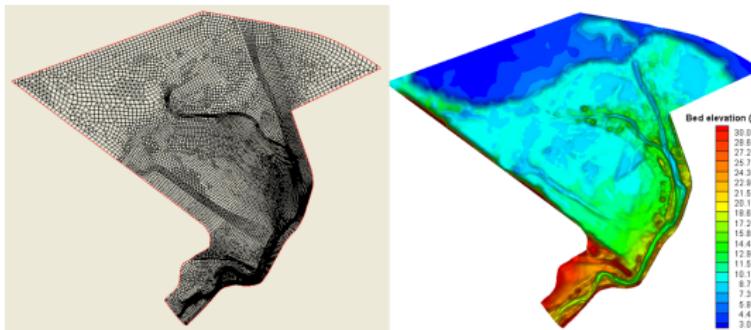
3. Use the provided Python script to convert Gmsh mesh to SMS.

- Translate Gmsh format to SMS format
- Assign elevation Z to each node by interpolation



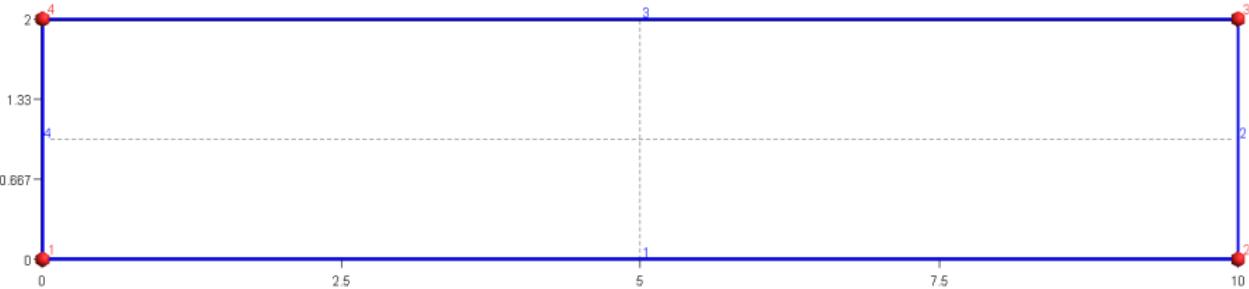
Some notes about using Gmsh:

- ▶ Gmsh has two modes: interactive and script (.geo file)
- ▶ Experienced user and for complicated domain geometry should use the script directly.
- ▶ Coordinates of points can be imported.



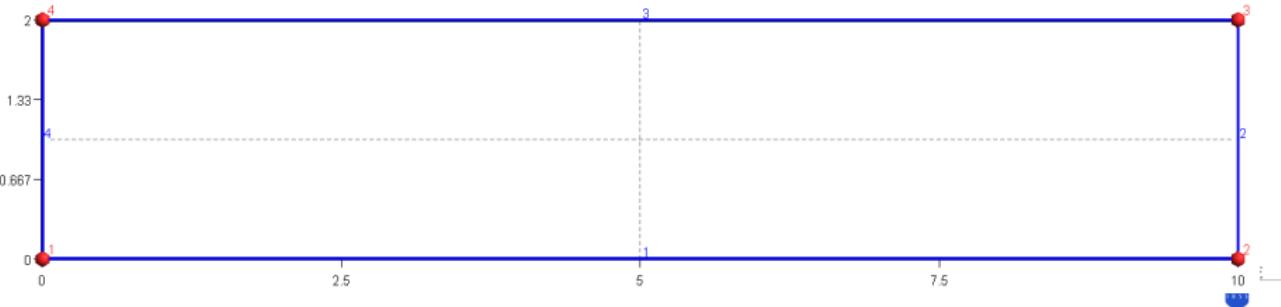
## Basic concepts in Gmsh:

- ▶ Point: e.g., `Point(3) = {0, 1, 0, 0.5};` defines a point (ID=3) at  $(x, y, z) = (0, 1, 0)$  with a target nearby mesh size 0.5
- ▶ Line: e.g., `Line(2) = {1,2,3,4};` which the line (ID=2) connecting Point 1, 2, 3 and 4 (in that order).
- ▶ Line Loop: e.g., `Line Loop(2) = {1,3,2,4};` define a closed line loop connecting Line 1, 3, 2 and 4 (in that order).
  - Line has directions. So if the line direction is the opposite to the direction of the line loop, put a negative sign in front of it.
  - e.g. `Line Loop(2) = {1,3,-2,4};`



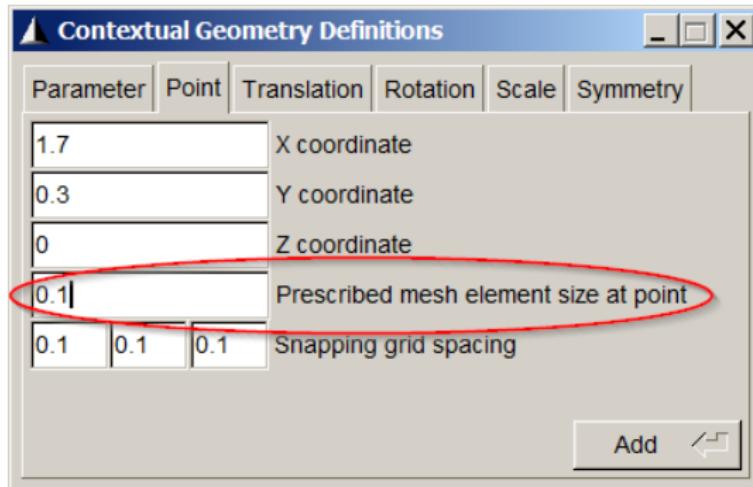
Try Gmsh before we do examples. Construct a rectangular channel ( $10 \text{ m} \times 2 \text{ m}$ ):

1. Create points: Geometry → Elementary entities → Add → Point
  - Point 1 (0, 0, 0), Point 2 (10, 0, 0), Point 3 (10, 2, 0), Point 4 (0, 2, 0).
2. Create lines: Geometry → Elementary entities → Add → Line
  - Line 1: connecting Point 1 and 2
  - Line 2: connecting Point 2 and 3
  - Line 3: connecting Point 3 and 4
  - Line 4: connecting Point 4 and 1
3. Create Line Loop: Geometry → Elementary entities → Add → Plane Surface
  - Connecting line 1, 2, 3 and 4



Construct a rectangular channel ( $10 \text{ m} \times 2 \text{ m}$ ):

1. Create points: Geometry → Elementary entities → Add → Point
  - When create point, you can specify the mesh size control around the point.



Construct a rectangular channel ( $10 \text{ m} \times 2 \text{ m}$ ):

4. Create physical line (our boundary condition will be based on their IDs):  
Geometry → Physical groups → Add → Line

- Physical Line 1 (inlet): use Line 4. Select it and it will highlight. Press e to end the selection.
- Physical Line 2 (outlet): use Line 2. Select it and it will highlight. Press e to end the selection.
- Physical Line 3 (wall): use Line 1 and 3. Select them and they will highlight. Press e to end the selection.

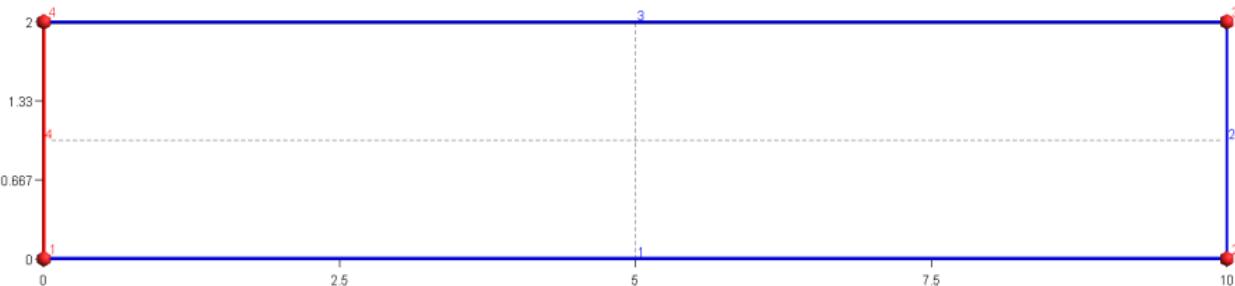
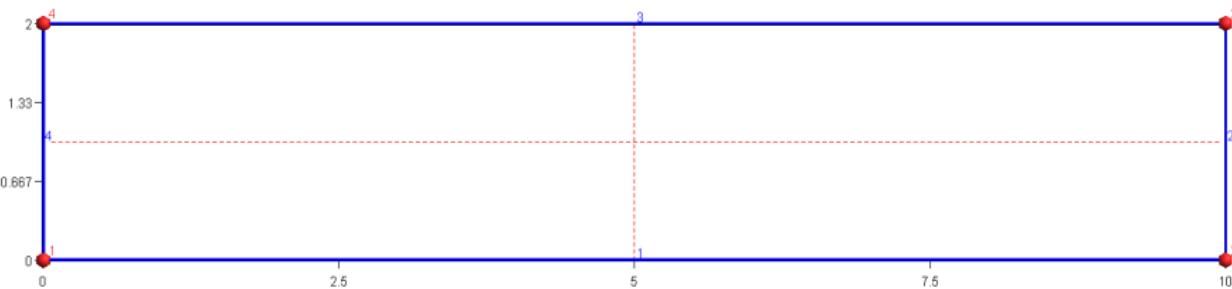


Figure: Line 4 is selected and highlighted to construct Physical Line 1 (inlet)

Construct a rectangular channel ( $10 \text{ m} \times 2 \text{ m}$ ):

5. Create Physical Surface 1: use the defined Plane Surface 1. Geometry → Physical groups → Add → Surface

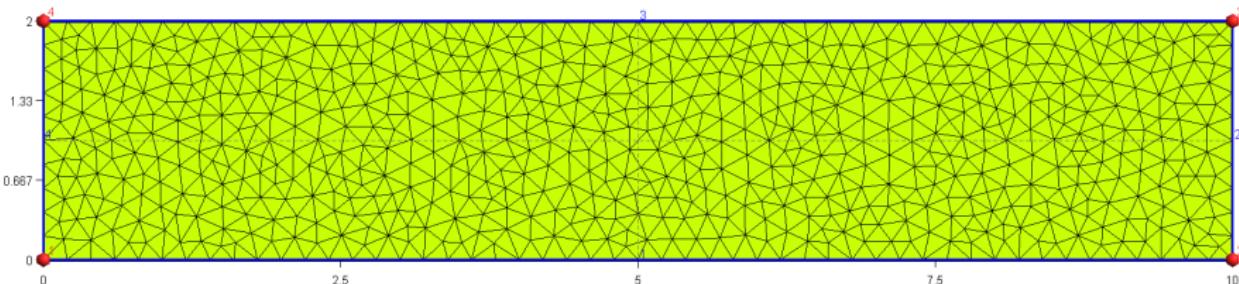
- Select the crossing of the two dash lines and the rectangular will be highlighted. Press e to end the selection and then press q to quit.
- Note: We can have multiple physical surfaces to represent different zones (e.g., different Manning's  $n$ )



**Figure:** Plane Surface 1 is selected and highlighted to construct Physical Surface 1 (domain)

Construct a rectangular channel ( $10 \text{ m} \times 2 \text{ m}$ ):

6. Create the 2D mesh:
  - Mesh → 2D
7. Save the script file which records what you have done: File → Save As. Select the .geo format. The file is a simple text file which can be viewed and edited.
8. Save the mesh: File → Save Mesh. By default, the mesh file has the same name as the .geo file.



More about the .geo file. It is very simple text file. The one we created looks like this:

```
Point(1) = {0, 0, 0, 0.2};  
Point(2) = {10, 0, 0, 0.2};  
Point(3) = {10, 2, 0, 0.2};  
Point(4) = {0, 2, 0, 0.2};  
  
Line(1) = {1, 2};  
Line(2) = {2, 3};  
Line(3) = {3, 4};  
Line(4) = {4, 1};  
Line Loop(6) = {1, 2, 3, 4};  
Plane Surface(6) = {6};  
  
Physical Line(7) = {4};  
Physical Line(8) = {2};  
Physical Line(9) = {1};  
Physical Line(10) = {3};  
Physical Surface(1) = {6};
```

Two important notes:

- ▶ Lines not connecting can not be grouped into one physical line
- ▶ The physical surface ID starts from 1 and be consecutive.

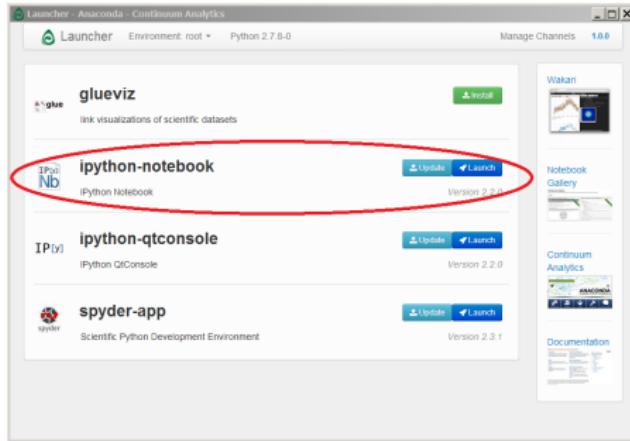
About the generated .msh file. It is also very simple text file.

```
$MeshFormat  
2.2 0 8  
$EndMeshFormat  
$Nodes  
714  
1 0 0 0  
...  
$EndNodes  
$Elements  
1426  
1 1 2 9 1 1 5  
...  
$EndElements
```

See Gmsh reference manual for the details.

Next, use the provided Python script to convert Gmsh format to SMS format.

1. Load Anaconda Launcher
2. Launch ipython-notebook. This should bring up your browser. Here IPython only needs a web browser to run.
3. Navigator to your case directory and make sure the Python script is there. Open the script by clicking.

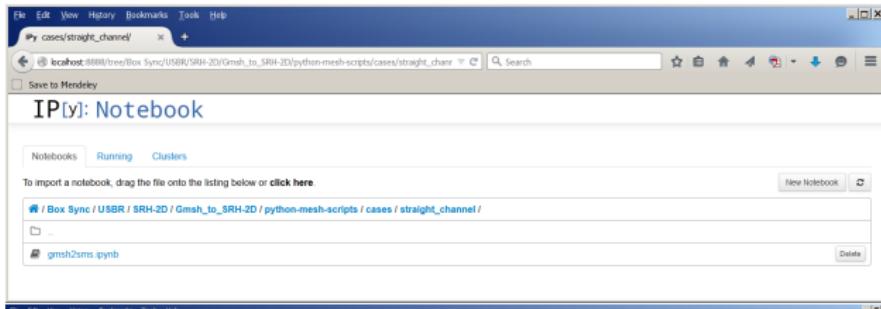


Note: The Python script is adapted from the following:

<https://github.com/lukeolson/python-mesh-scripts>

Next, use the provided Python script to convert Gmsh format to SMS format.

4. The IPython interface looks like this and click on the gmsh2sms.ipynb file.

A screenshot of an IPython Notebook interface showing a code cell. The title bar says "IP[y]: Notebook gmsh2sms (restarted)". The code cell contains the following Python script:

```
In [7]: %matplotlib inline
import numpy as np
import scipy as sp
import meshio
from os import path
from matplotlib import patches
import sys

class Mesh:
    """
    Store the verts and elements and physical data
    attributes
    -----
    Verts : array
        rank 2 array of 3d coordinates (npts x 3)
    Edges : dict
        dictionary of tuples
        rank 1 array of physical ids, rank 2 array of element to vertex ids
        (mesh & pgen) each array in the tuple is of length nverts thus : dict
        keys and names
    methods
    -----
    read_msh:
        read a 2.2 ascii gmsh file
    write_vtu:
```

Next, use the provided Python script to convert Gmsh format to SMS format.

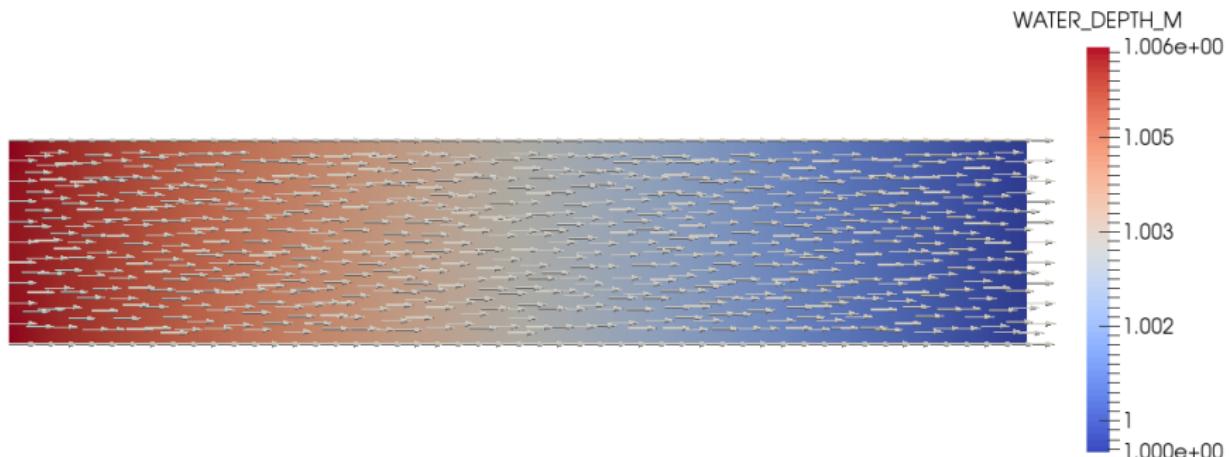
5. You can have a look at the IPython notebook. At the end of the file, you will see a line:

```
mesh = readmesh('gmsh_straight.msh')
```

If you want reuse this code, you need to change the Gmsh file name accordingly.

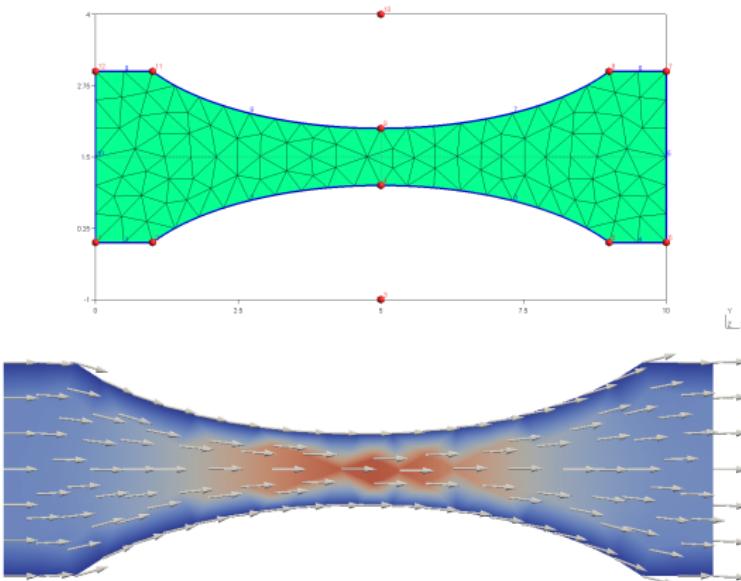
6. Run the Python code by pressing *CTL+L*. A SMS file `gmsh_straight.msh` will be generated.

Next, use the generated SMS mesh file to proceed as usual.



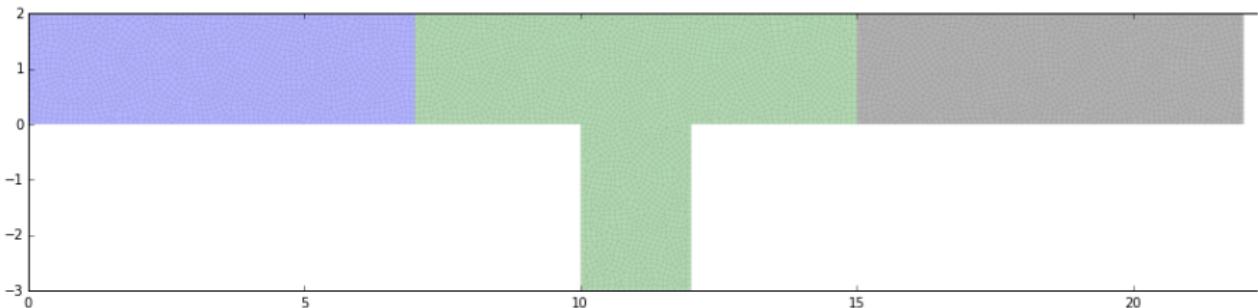
## Gmsh example 2: A river contraction

1. The geometry was adapted from <http://femwiki.wikidot.com>
2. Similar to straight channel case, but with curves in Gmsh.

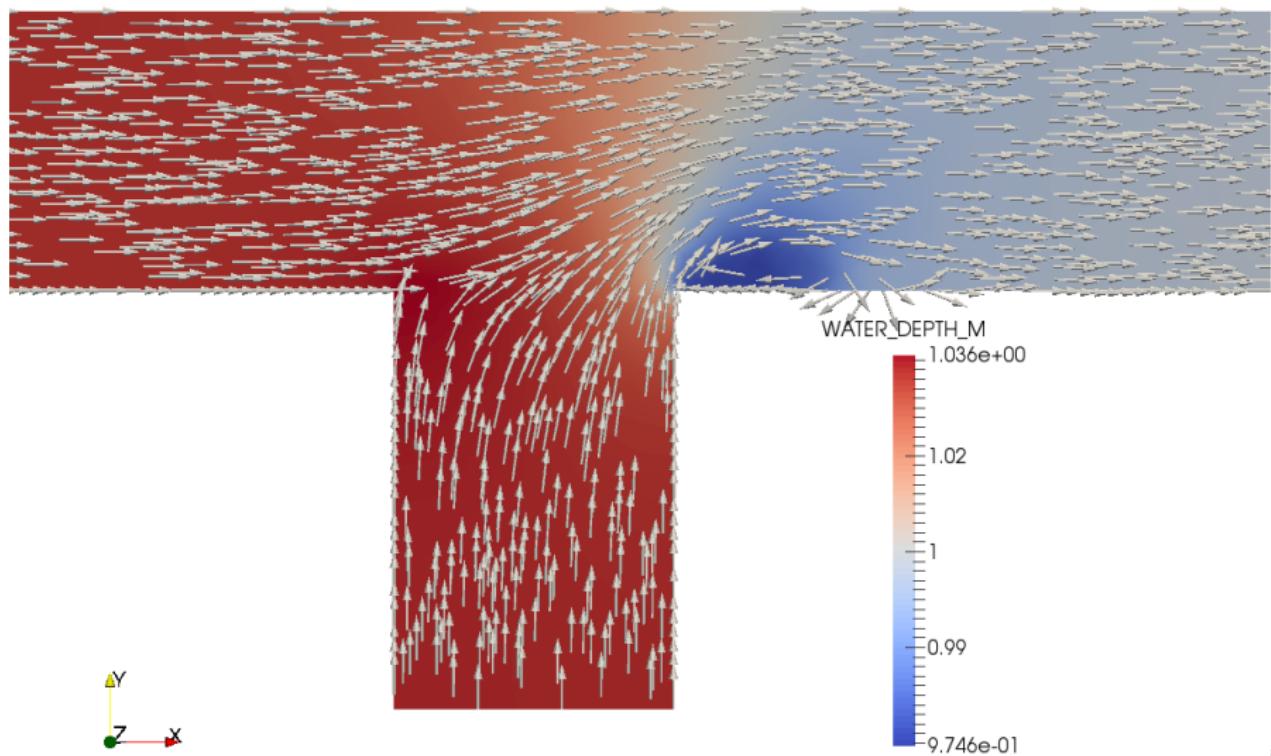


## Gmsh example 3: A junction flow

1. Two inlets and one outlet
2. The domain is divided into three zones: 1-main channel inlet zone, 2-junction zone, and 3-main channel outlet zone.
3. Different zones can assign different properties (such as Manning's  $n$ )



## Gmsh example 3: A junction flow



This part of the short course covered:

- ▶ Use ParaView to visualize SRH2D result
- ▶ Use Gmsh + Python to prepare mesh for SRH2D

The End!