

pyHMT2D: A python package for automating 2D computational hydraulics modeling

Xiaofeng Liu, Penn State University, State College, PA, xzl123@psu.edu
Ali Mahdavi Mazdeh, Penn State University, State College, PA, ajm8717@psu.edu

Introduction

Hydraulics modeling has been extensively used for many practical applications such as flood forecasting, flood control design, floodplain mapping, stormwater management, transportation hydraulic studies, bridge scour, river restoration, habitat evaluation, and many others. Most of these models have graphical user interfaces (GUIs) which provide many built-in features for convenient model preparation and result demonstration. While they are easy to use, straightforward, and give direct control to the modelers, they are mostly developed for small number of simulations, and they are not designed for tasks that need to run many simulations, such as parametric studies, Monte Carlo simulations, automatic calibrations, optimizations, and parallel runs of many cases. These tasks are better handled with scripting languages such as Python. Specifically, Python scripting can be extremely helpful for well-defined workflows. Scripting to control simulation runs helps to automate the simulation process including preprocessing, simulation itself, as well as postprocessing. This presentation will demonstrate an open-source Python package, pyHMT2D, which is designed to automate hydraulics modeling (<https://github.com/psu-efd/pyHMT2D>).

pyHMT2D

pyHMT2D stands for 2D Hydraulics Modeling Tools with Python. It is a Python package that can control many steps in the pre-processing, simulation, and post-processing workflow of a modeling project. It currently supports SRH-2D (Lai 2008, 2010) and HEC-RAS (Brunner 1995). This Python package can parse and modify the inputs and outputs of the supported models. It can also control the model runs. With the versatile scripting capability and powerful software ecosystem of Python, pyHMT2D can automate many of the modeling workflows that a hydraulic engineer routinely performs, such as:

- Automate repetitive tasks such as calibration and optimization.
- Bridge between hydraulic models and the Python universe for doing statistics, machine learning/AI, GIS, parallel computing, cloud computing, visualization, etc.
- Monte-Carlo simulations, uncertainty quantification (UQ), sensitivity analysis
- Convert HEC-RAS 2D mesh to SRH-2D for fair model inter-comparison
- Convert HEC-RAS 2D and SRH-2D results to VTK format for flexible and efficient analyzing and visualizing of the results.
- Sample and probe simulation results on arbitrary points/ lines (with the functionality of VTK library) and visualize them using Python packages.

Three important capabilities of pyHMT2D will be demonstrated with examples, whose scripts can be used as templates for other projects. The first example is the conversion of a HEC-RAS 2D mesh and Manning's n to SRH-2D so both models can run with the same mesh and

roughness coefficient. The second example shows how to control the runs of SRH-2D. The third example is a more complex one and it shows how to perform Monte Carlo simulations. All these examples and many more can be found in the open source repository on GitHub.

Convert HEC-RAS 2D mesh and Manning's n to SRH-2D

In practice, it is sometime desired to compare the results of different models, for example SRH-2D and HEC-RAS 2D. Using pyHMT2D, a modeler can convert HEC-RAS 2D mesh, boundary conditions, and Manning's n data into SRH-2D format such that both models can run a case with exactly the same mesh and roughness coefficient. A side benefit is that HEC-RAS can be used as a mesh generator for SRH-2D. Figure 1 shows the Python code to convert the HEC-RAS 2D model, using its HDF and TIFF files, to SRH-2D. The converter creates the SRH-2D data files with the exception of the srhydro file, which needs to be edited manually.

```
import pyHMT2D

def main():
    #build the RAS_to_SRH_Converter object
    my_ras_to_srh_converter = pyHMT2D.Misc.RAS_to_SRH_Converter("Muncie2D.p01.hdf",
        "Terrain/TerrainMuncie_composite.tif", "SRH-2D/Muncie")

    #convert to SRH-2D (mesh and material) and save
    my_ras_to_srh_converter.convert_to_SRH()

if __name__ == "__main__":
    main()
```

Figure 1. Sample script for converting HEC-RAS 2D input data files to SRH-2D.

Run an SRH-2D simulation

pyHMT2D can control the runs of hydraulics models. Internally, the control is through different mechanisms. For example, for SRH-2D, it is a simple system call of its executable files. For HEC-RAS, it is through Windows's COM interface. Regardless the internal difference, a user of pyHMT2D controls simulation runs with similar command. In this example, SRH-2D is run on a prepared case with only several lines of Python code (Figure 2).

```

import pyHMT2D

#create a SRH-2D model instance
my_srh_2d_model = pyHMT2D.SRH_2D.SRH_2D_Model(version, srh_pre_path, srh_path, extra_dll_path, faceless=False)

#initialize the SRH-2D model
my_srh_2d_model.init_model()

#open a SRH-2D project
my_srh_2d_model.open_project("Muncie.srhhydro")

#run SRH-2D Pre to preprocess the case
my_srh_2d_model.run_pre_model()

#run the SRH-2D model's current project
my_srh_2d_model.run_model()

#close the SRH-2D project
my_srh_2d_model.close_project()

#quit SRH-2D
my_srh_2d_model.exit_model()

```

Figure 2. Sample script to run SRH-2D using pyHMT2D.

Application of pyHMT2D in Monte Carlo simulations

Scripting to control simulation runs is well suited for situations where many runs are needed such as Monte Carlo simulations. In this example, a HEC-RAS 2D case was created with the data of the Muncie example from HEC-RAS. The example case and script can be found in pyHMT2D repository on GitHub. In this example the main channel manning's n is assumed to vary within a truncated normal distribution (mean= 0.04, min = 0.03, max = 0.05, std = 0.005). The script sampled one hundred Manning's n randomly from this distribution and then it ran the model 100 times using these Manning's n values with the help of pyHMT2D. Figure 3 (left) shows the histogram of sampled main channel Manning's n values. After each run, the water surface elevation (WSE) at a desired location within the channel was recorded using pyHMT2D. These recorded WSE's were later used to calculate the exceedance of probability. Figure 3 (right) shows the respective water surface elevation exceedance of probability at the monitoring point.

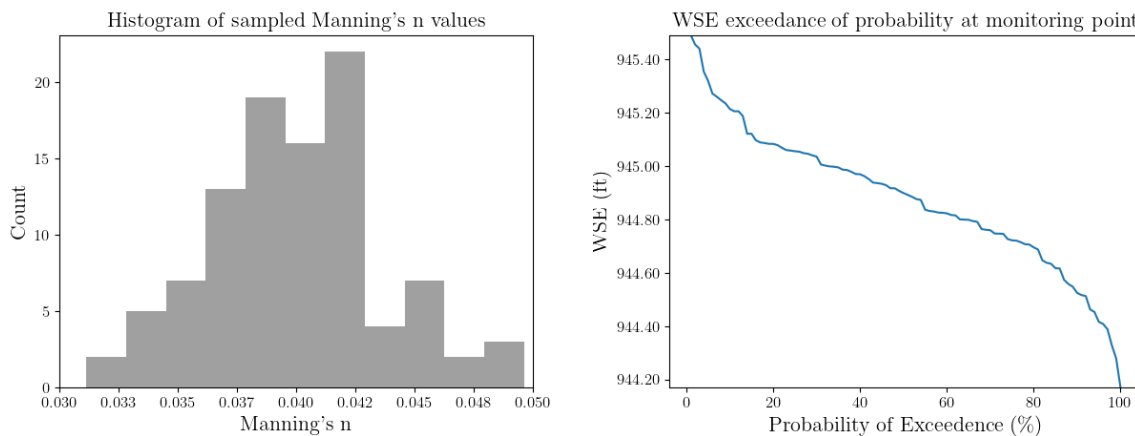


Figure 3. Histogram of sampled Manning's n values (left), WSE's exceedance of probability curves

Other Applications

With the enormous and powerful Python packages and the ability of pyHMT2D in communicating with HEC-RAS 2D and SRH-2D, there are many use examples for a modeler to solve different practical problems. One another example is automatic calibration. There are many advanced Python packages for optimization. These packages along with pyHMT2D can be used to calibrate the model by minimizing the error between simulation and measurement. Many models have pre-defined errors if their GUI is used. However, with pyHMT2D, a modeler can define a customized error for calibration or optimization.

Another important application is in Machine learning (ML) and AI. Hydraulic models are increasingly used to produce needed data for the training of AI models. To generate “big” data, it is often required to run a model many times. Since most of ML/AI packages are developed in python, pyHMT2D is natively useful to serve as a bridge between computational hydraulics and ML/AI.

Computational resources and cost are one major concern among modelers, especially as 2D modeling becomes prevalent. With pyHMT2D, it is possible to automatically and intelligently utilize advanced computing techniques and resources, such as cloud computing, to increase computing speed, expand computing capacity, and reduce cost.

Conclusion

pyHMT2D is a Python package for automated hydraulic modeling that currently supports SRH-2D and HEC-RAS 2D. Some of its features and applications such as converting the HEC-RAS model to SRH-2D, running SRH-2D, and Monte Carlo analysis are introduced. Additionally, pyHMT2D can be used in conjunction with other Python packages to be more powerful.

References

- Lai, Y. G. (2008). “Srh-2d version 2: Theory and user’s manual.” Sedimentation and River Hydraulics—Two-Dimensional River Flow Modeling, US Department of Interior, Bureau of Reclamation.
- Lai, Y. G. (2010). “Two-dimensional depth-averaged flow modeling with an unstructured hybrid mesh.” *Journal of Hydraulic Engineering*, 136(1), 12–23.
- Brunner, G. W. (1995). HEC-RAS River Analysis System. Hydraulic Reference Manual. Version 1.0. (Tech. Rep.). Hydrologic Engineering Center Davis CA