Postprocessing Scripts for OpenFOAM in Open Channel Hydraulics

Your Name

September 30, 2024

Abstract

This document outlines various postprocessing scripts designed for OpenFOAM simulations related to open channel hydraulics. The scripts are categorized into GNU scripts, Python scripts, and ParaView scripts, each detailing their functionalities and applications.

Contents

1 Introduction

Provide an overview of the importance of postprocessing in CFD simulations, specifically for open channel hydraulics using OpenFOAM.

2 GNU Scripts

GNU tools, like Gnuplot, are popular for their reliability and flexibility in handling data. One benefit of using these tools is that they are usually available on most Linux operating systems. Many versions of Linux come with essential GNU packages already installed, so users can easily access this without needing anything extra. This makes it convenient for plotting residuals, though python or other coding languages are also sufficient.

2.1 GNU Script: Residuals Plot

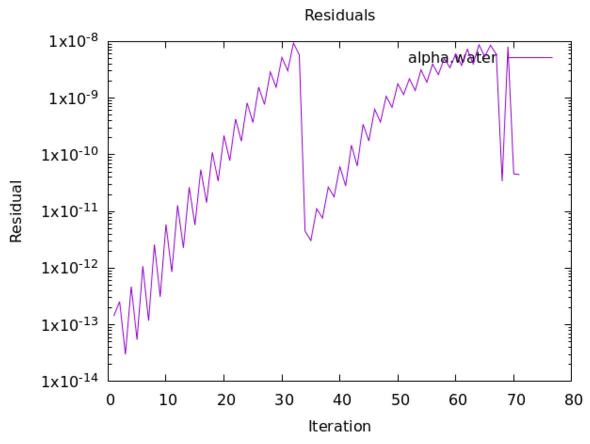

Figure 1: Example residuals plot for alpha.water

2.1.1 Script Description

This GNU script generates a plot of the residuals from an OpenFOAM `interfoam.out` file, particularly focusing on the `alpha.water` field. The script uses `gnuplot` to filter and process the residuals, and outputs a plot with logarithmic scaling on the y-axis.

Below is the script:

```
set terminal pngcairo
set output fileout
set logscale y
set title "Residuals"
set ylabel 'Residual'
set xlabel 'Iteration'
plot "< cat interfoam.out | grep 'Solving for alpha.water' | cut -d' ' -f13 |
    tr -d ','" title 'alpha.water' with lines
set output
```

Script Breakdown: - `set terminal pngcairo`: Sets the terminal type to `pngcairo` for generating PNG images. - `set output fileout`: Specifies the output file for the plot. - `set logscale y`: Applies logarithmic scaling to the y-axis to better visualize residuals across a wide range of values. - `set title "Residuals"`: Sets the title of the plot. - `set ylabel 'Residual'`: Labels the y-axis as "Residual". - `set xlabel 'Iteration'`: Labels the x-axis as "Iteration". - `plot "< cat interfoam.out | grep 'Solving for alpha.water' ..."`: Filters the `interfoam.out` log file for lines containing `alpha.water`, extracts the residual value, and plots it as a line. - `set output`: Completes the script by closing the output.

Usage Notes: - `fileout` should be replaced with the desired file name for the PNG output. - The script assumes that the residuals for `alpha.water` can be found in the `interfoam.out` file, and adjusts the log output accordingly.

2.1.2 Example Usage

The following commands demonstrate how to use the GNU script with `gnuplot` to generate residual plots. Each command specifies a different output file for the plots:

```
gnuplot -p -e 'fileout="Resid_turb.png"' GNUscripts/Residuals_turb
gnuplot -p -e 'fileout="Resid_pressure.png"' GNUscripts/Residuals_pressure
gnuplot -p -e 'fileout="Resid_alpha.png"' GNUscripts/Residuals_alpha
```

In these examples: - `gnuplot -p` runs `gnuplot` in persistent mode, allowing the plot window to remain open. - The `-e` option allows you to set variables directly from the command line. - Each command specifies a different script located in the `GNUscripts` folder to generate specific residual plots.

2.2 Script Title 2

```
# Add your GNU script code here
```

2.2.1 Description

Briefly describe the functionality and purpose of this script.

3 Python Scripts

3.1 Script Title 1

```
# Add your Python script code here
```

3.1.1 Description

Briefly describe the functionality and purpose of this script.

3.2 Script Title 1

```
# Add your Python script code here
```

3.2.1 Description

Briefly describe the functionality and purpose of this script.

3.3 Script Title 1

```
# Add your Python script code here
```

3.3.1 Description

Briefly describe the functionality and purpose of this script.

4 ParaView Scripts

4.1 Calculation of Depth and Water Surface elevation using an alpha.water contour and a bottom surface

This section provides a technical note on the postprocessing of OpenFOAM simulations for open channel hydraulics using ParaView. The focus is on comparing multiple simulation cases, resampling surface water elevation (WSE), and calculating depths on a specific bottom patch. This script allows for flexible post-processing either from the current working directory or from specified locations. The script processes different cases, computes Water Surface Elevation (WSE), and interpolates and compares the depths from these cases.

4.1.1 Script Overview

The script is designed to perform the following operations:

- Load a base OpenFOAM case and optionally other cases for comparison.
- Resample WSE onto a specific patch (e.g., bottom patch).
- Calculate the depth by subtracting the WSE from the bottom patch.
- Combine the depth results from multiple cases for visualization and comparison.

The script can be run in two modes:

1. Use the current working directory.
2. Specify case locations for comparison.

4.1.2 Script Details

Below is the Python script that uses the `paraview.simple` module to process OpenFOAM data:

```python
import os
from paraview.simple import *
import numpy as np

# Specify the patch to use for viewing depth (e.g., bottom)
patch_name = 'patch/bottom'  # You can change this as needed

# Set the mode for running the script
use_current_directory = False  # Set to False to use specified case locations
    and compare cases

if use_current_directory:
    dir = os.path.dirname(os.path.abspath(__file__))
    filename = os.path.join(dir, 'case.foam')
    cases_to_resample = [filename]
else:
    base_case_location = r'C:\path\to\Ubend0.2'
    cases_to_resample = [
        r'C:\path\to\Ubend0.2\case.foam',
        r'C:\path\to\Ubend0.2_2\case.foam',
    ]

# Load the base case
casefoam = OpenFOAMReader(registrationName='base_case', FileName=os.path.join(
    base_case_location, 'case.foam'))
casefoam.UpdatePipeline()
casefoam.MeshRegions = [patch_name]

depth_calculators = []

for case in cases_to_resample:
    base_folder_name = os.path.basename(os.path.dirname(case))
    casefoam_1 = OpenFOAMReader(registrationName=f'{base_folder_name}_case.foam_1', FileName=case)
    casefoam_1.CellArrays = ['alpha.water']
    casefoam_1.UpdatePipeline()

    contour1 = Contour(registrationName=f'{base_folder_name}_Contour1', Input=casefoam_1)
    contour1.ContourBy = ['POINTS', 'alpha.water']
    contour1.Isosurfaces = [0.5]
    contour1.UpdatePipeline()

    calculator1 = Calculator(registrationName=f'{base_folder_name}_Calculator1', Input=contour1)
    calculator1.Function = 'coordsZ'
    calculator1.ResultArrayName = 'WSE'
    calculator1.UpdatePipeline()

    convertToPointCloud1 = ConvertToPointCloud(registrationName=f'{base_folder_name}_ConvertToPointCloud1', Input=calculator1)
    convertToPointCloud1.UpdatePipeline()

    depthCalculator = Calculator(registrationName=f'Depth_{base_folder_name}', Input=convertToPointCloud1)
    depthCalculator.Function = 'WSE - coordsZ'
    depthCalculator.ResultArrayName = f'Depth_{base_folder_name}'
    depthCalculator.UpdatePipeline()

    depth_calculators.append(depthCalculator)

if depth_calculators:
    appendAttributes2 = AppendAttributes(registrationName='AppendAttributes2', Input=depth_calculators)
    appendAttributes2.UpdatePipeline()

    appendAttributesDisplay = Show(appendAttributes2, GetActiveViewOrCreate('RenderView'))
```

4.1.3 Usage Notes

- The patch name `'patch/bottom'` can be modified depending on the case geometry and the desired comparison patch.
- Set `use_current_directory` to `True` if processing is desired for the current working directory.
- Modify the `cases_to_resample` list to add more cases for comparison with the base case.

5 Blender Scripts

6 References