

CFD Through OpenFOAM

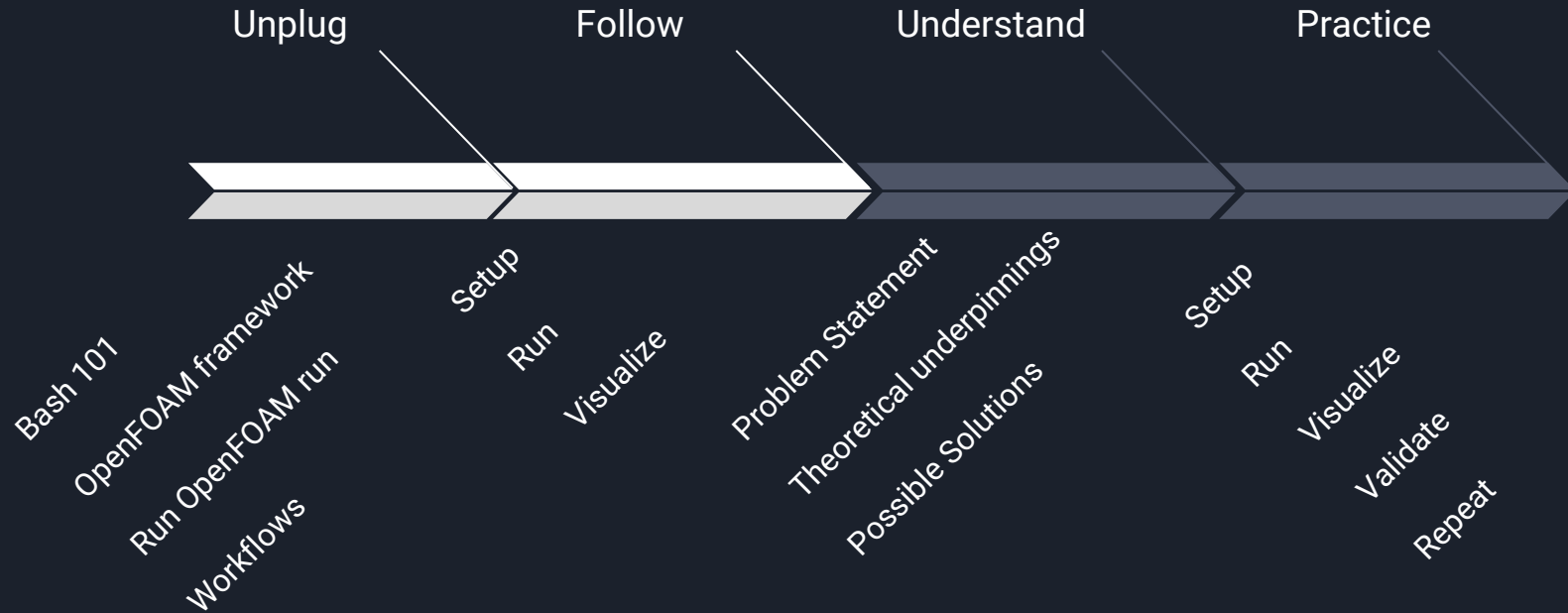


CACDS Workshop, Jan 23-25, 2018

Taher Chegini : tchegini@uh.edu

Amit Amritkar: armakritk@central.uh.edu

Roadmap





Acquaintanceship

01 **Linux** users? Open Source vs. Closed Source, Terminal vs. Mouse

02 **CFD** experience? FVM, Numerical Methods, Scientific Computing, Visualization

03 **OpenFOAM** users? C++, Source Code, Parallel Programming

Terminal

Where all the
magic happens:

Boosting
productivity

Automation

Empowering

```
[taher@cf-d-lab ~]$ ofdev
[taher@cf-d-lab ~]$ run
[taher@cf-d-lab run]$ cd validation_cases/
[taher@cf-d-lab validation_cases]$ ls
damBreakWithObstacle  lidDrivenCavity  schemesComparison  shockTube
[taher@cf-d-lab validation_cases]$ cd lidDrivenCavity/
[taher@cf-d-lab lidDrivenCavity]$ ./Allrun
Cleaning up the case
Setting Re to 100
Generating mesh
Renumbering Matrices
Running icoFoam
```

Takes a little getting
used to:

Perseverance

Risky

Source Codes

01 Official versions:

<https://openfoam.org/>

<https://www.openfoam.com/>

<https://sourceforge.net/projects/foam-extend/>

02 Installation methods:

<https://github.com/OpenFOAM/OpenFOAM-5.x>

<https://hub.docker.com/r/openfoam/openfoam5-paraview54/>

<http://dl.openfoam.org/ubuntu/>

03 Useful resources:

<https://www.cfd-online.com/Forums/openfoam/>

<https://cpp.openfoam.org/v5/>

<https://www.openfoam.com/documentation/cpp-guide/html/>

[A Youtube Channel](#)

[Håkan Nilsson website](#)

[CFD Direct](#)



Workflows: Packages

Useful Free Packages:

Case manager: Git (github education, gitlab, ...)

Data processing: python (numpy, matplotlib), gnuplot

Rendering: paraview python module, ffmpeg, ImageMagick

Text editor: Vim, Emacs

Image editor: GIMP





Workflows: Example

1. Choosing a solver: interFoam
2. Finding a similar case in tutorials: damBreakWithObstacle
3. Finding related data for validation: [ComFLOW - dambreak experiment](#)
4. Validating the solver: water level (H2, H4) and pressure sensors data (P1, P7)
5. Fine tuning the simulation parameters: using bash scripts and python
6. Establishing an effective pipeline based on the desired outputs:
 - a. Github: easy transfer of data
 - b. Python: plotting comparison graphs
 - c. Paraview.simple module: rendering the final results into image sequences
 - d. foamCreateVideo: converting images to a video
7. Preparing the case for running locally or on the cluster:
 - a. Preparing a slurm file to run on Sabine
 - b. Transferring the outputs to github repository



Hands-on



```
$ run  
  
$ mkdir UHOF  
  
$ cd UHOF  
  
$ git clone https://github.com/taataam/UHOFWorkshop.git  
  
$ cd UHOFWorkshop/validation_cases
```