Bayesian Optimization with OpenFOAM

ML/OpenFOAM Hackathon

Mohammed Elwardi Fadeli [1], Tomislav Maric [1], Andre Weiner[2]

July 24, 2023

[1] Mathematical Modeling and Analysis (MMA), TU Darmstadt
[2] TU Braunschweig

Bayesian Optimization: An overview

Bayesian Optimization

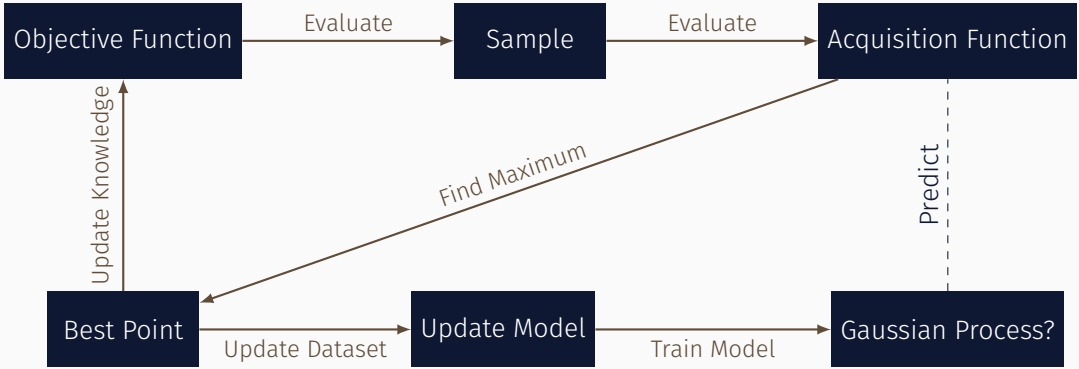


Figure 1: A Simplified Bayesian Optimization workflow

Bayesian Optimization

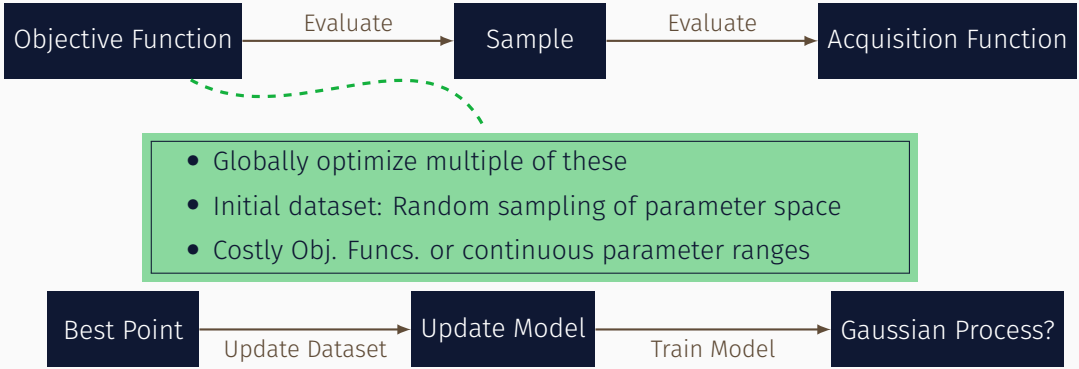


Figure 1: A Simplified Bayesian Optimization workflow

Bayesian Optimization

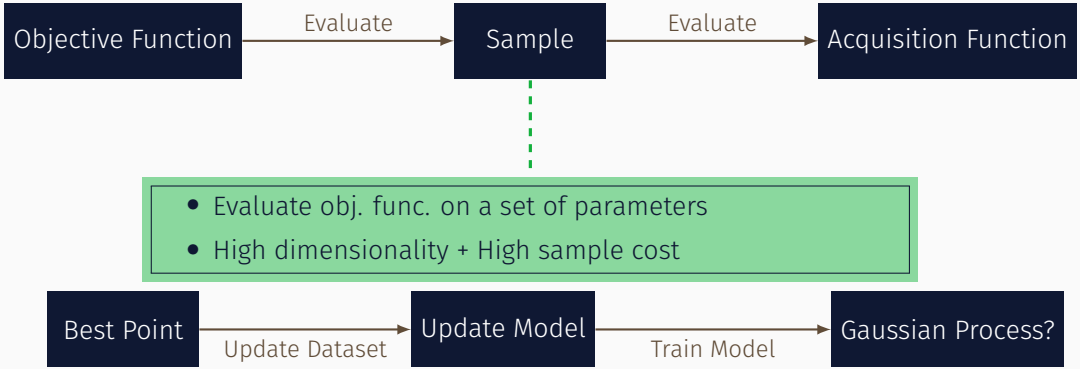


Figure 1: A Simplified Bayesian Optimization workflow

Bayesian Optimization

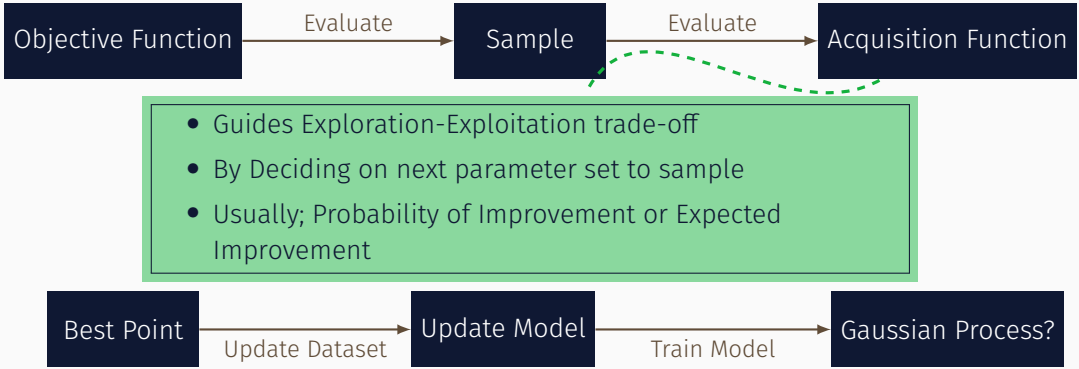


Figure 1: A Simplified Bayesian Optimization workflow

Bayesian Optimization

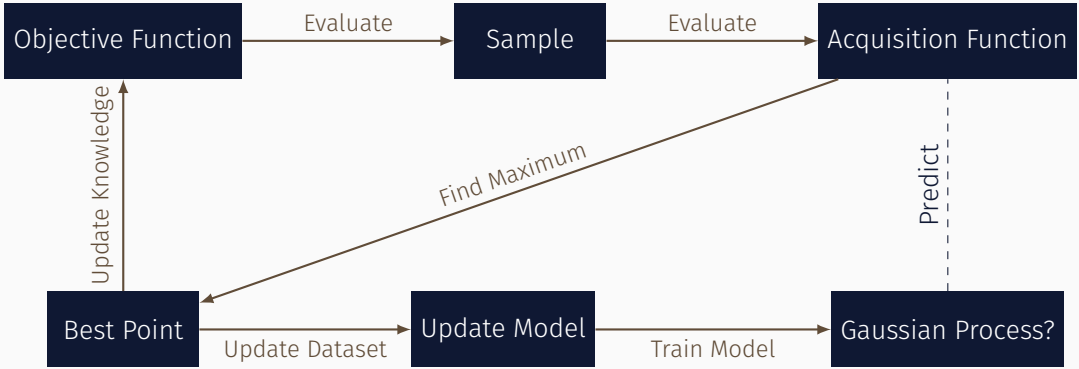


Figure 1: A Simplified Bayesian Optimization workflow

Bayesian Optimization

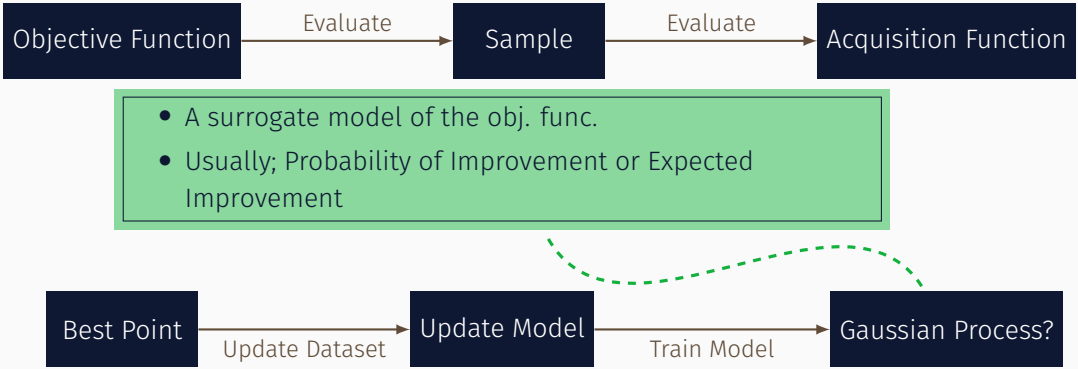


Figure 1: A Simplified Bayesian Optimization workflow

Bayesian Optimization

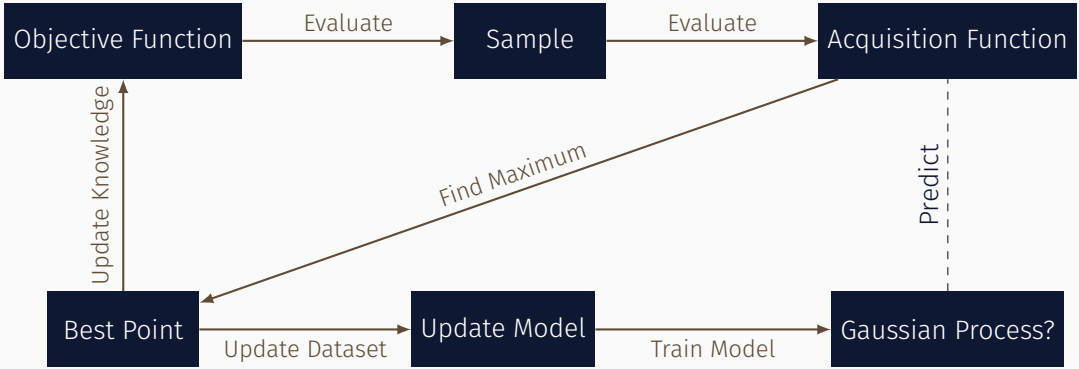


Figure 1: A Simplified Bayesian Optimization workflow

OpenFOAM Multi-Objective Optimization: Toolkit status

Global Multi-Obj. Optimization of OpenFOAM cases

- Fine it here: **FoamScience/OpenFOAM-Multi-Objective-Optimization**
- Based on Facebook's **Adaptive Experimentation Platform**
- Automatically picks optimization settings/components for you (Based on parameters configuration).
- You only need **a config file** and your **OpenFOAM case**

- An experimental PoC project at this point.
- You're invited to PR your improvements.

Global Multi-Obj. Optimization of OpenFOAM cases

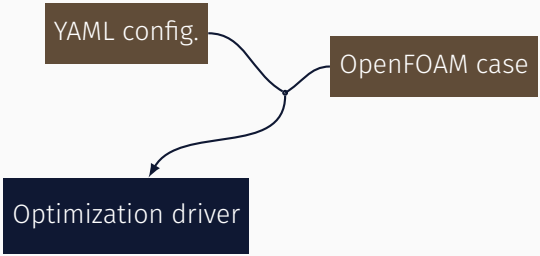


Figure 2: A configuration-only workflow for optimization on OpenFOAM cases

Global Multi-Obj. Optimization of OpenFOAM cases

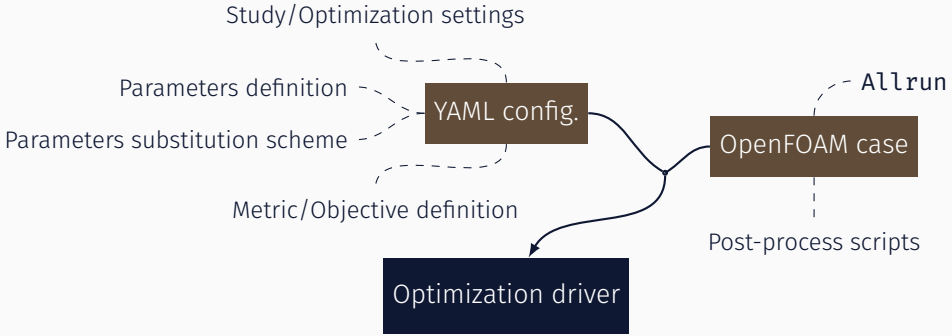


Figure 2: A configuration-only workflow for optimization on OpenFOAM cases

Global Multi-Obj. Optimization of OpenFOAM cases

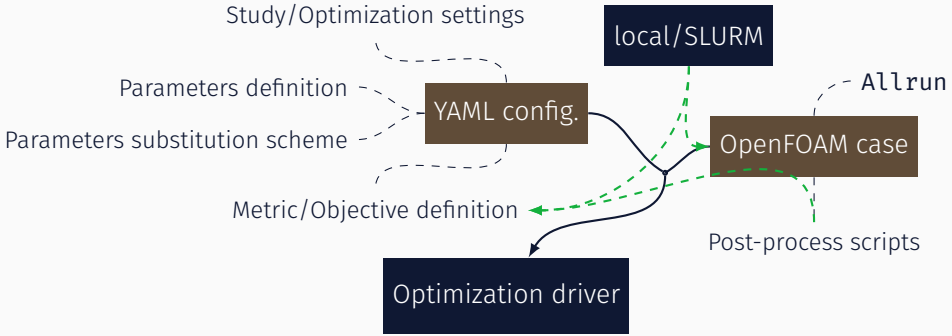


Figure 2: A configuration-only workflow for optimization on OpenFOAM cases

Hackathon challenge: Starter code

Challenge 1: Shape optimization on OpenFOAM cases

- Detailed documentation at **docs.md**
- Objective: Minimize Pressure drop between inlet and outlet patches in `pitzDaily` case for varying shapes of the `lowerWall` patch.

Stage 01
- Go through the starting code (Case setup, Configuration file, Live-metrics)
- Reach a conclusion about the best shapes

Stage 02
- What parameters are most important?
- Investigate usage of dependent parameters

Stage 03
- Add one more objective (Turbulence, or mesh surface area)
- Study trade-offs between these two objectives

Challenge 1: Shape optimization on OpenFOAM cases

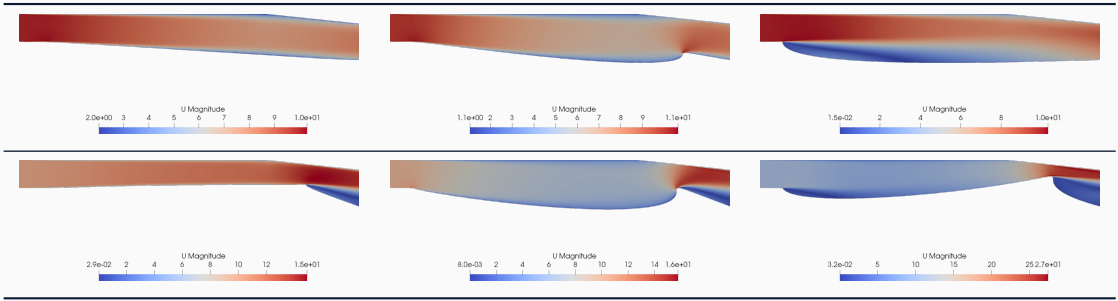


Figure 3: Example samples produced with the starter code

Few words on the environment

Your opening routine

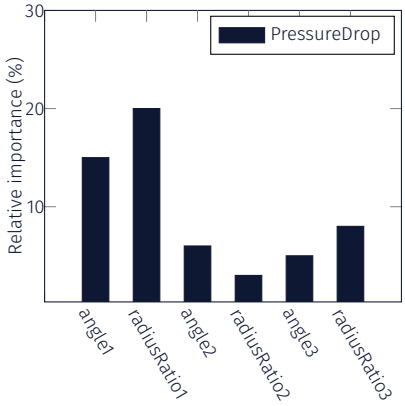
```
ssh -XC ubuntu@<hostname-will-be-shared-with-you>
conda activate /fsx/bias/bias_env
source /fsx/OpenFOAM/OpenFOAM-v2212/etc/bashrc
# Do this only once
mkdir -p /fsx/bias/<your-name>
cd /fsx/bias/<your-name>
git clone https://github.com/FoamScience/OpenFOAM-Multi-Objective-Optimization omoo
cd omoo && git checkout mlhackathon_07_2023
# ---- Optional stuff ---- #
# If you want to see "live" images of your trials
# Get a free Key from imgbb.com
export IMGBB_API_KEY=<your-key>
# If you're a VIM user, do this for an easier life
alias vim=lvim
# And this will take you to the roadmap (In no particular order)
vim +'Neorg index'
```

The OpenFOAM case uses: **simpleFoam** solver, **cfMesh**, **OpenSCAD** and **ParaView**

Relative feature importance for design decision

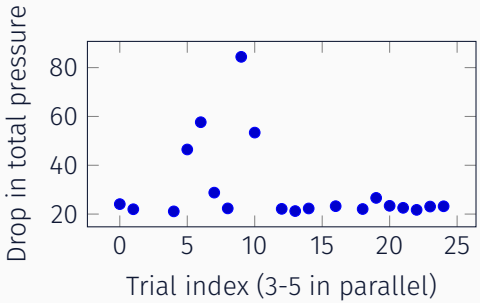
Contribution of parameters to the outcomes (Fictional data)

```yaml
problem:
  name: ShapeOpt
  template_case: 'pitzDaily'
  parameters:
    angle1:
      type: range
      value_type: float
      bounds: [-90.0, 0.0]
      log_scale: False
    radiusRatio1:
      type: range
      value_type: float
      bounds: [0.1, 0.5]
      log_scale: False
    # ....
  objectives:
    PressureDrop:
      mode: 'shell'
      command: ['./getPressDiff.sh']
      threshold: 6
      minimize: True
      lower_is_better: True
```



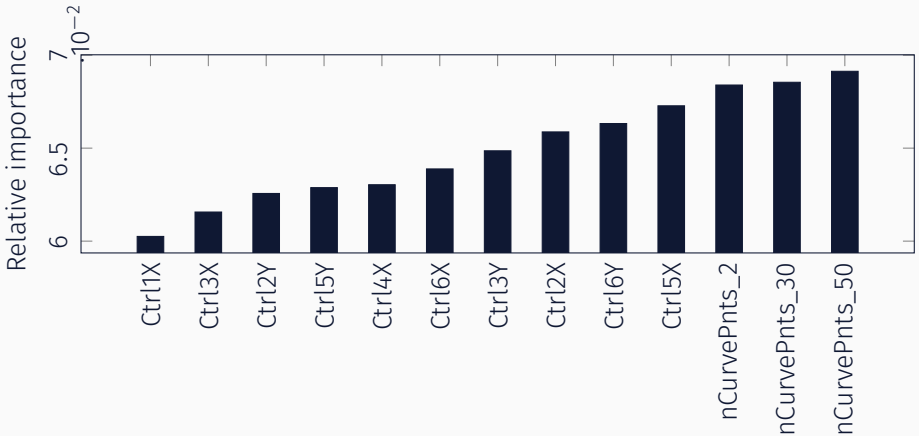
Hackathon challenge: Report

Hackathon challenge report: Stage 01



- Parametrize: Xs and Ys of 8 control points to build the bezier curve for the lower wall.
- Objective: Minimize pressure drop
- 6 trial cases diverged
- GP+EI performed well; can we stop after only 25 trials?
- SAASBO, which is more advanced, performed worse!
- But it had one more ”continuous parameter” + ordered choice parameters

Hackathon challenge report: Stage 01



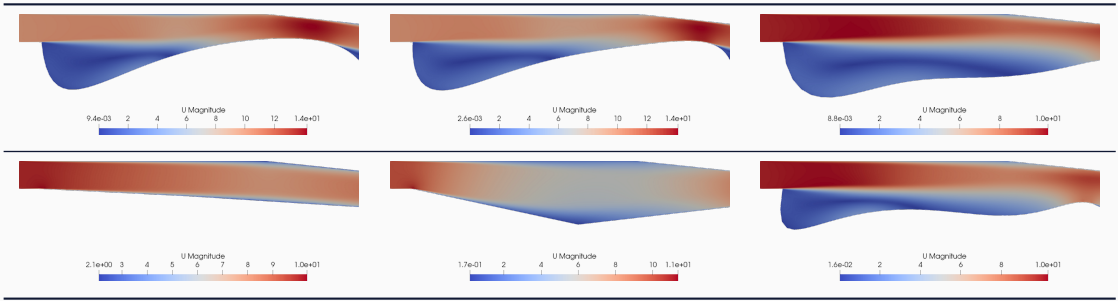
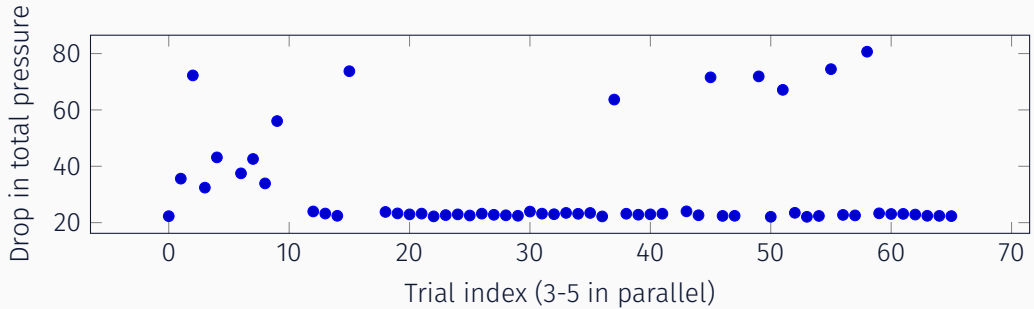
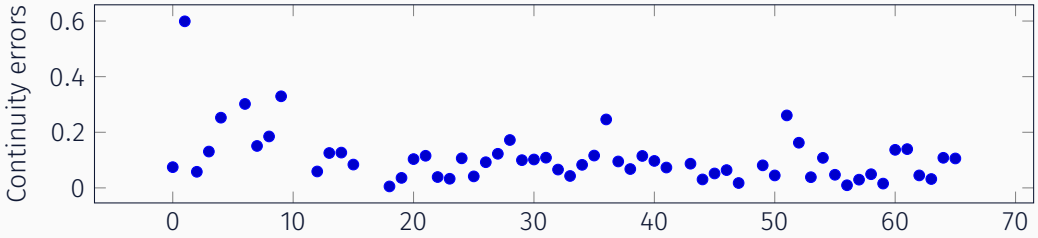


Figure 4: Highlights produced with GPEI (Optimal: 2nd row - center)

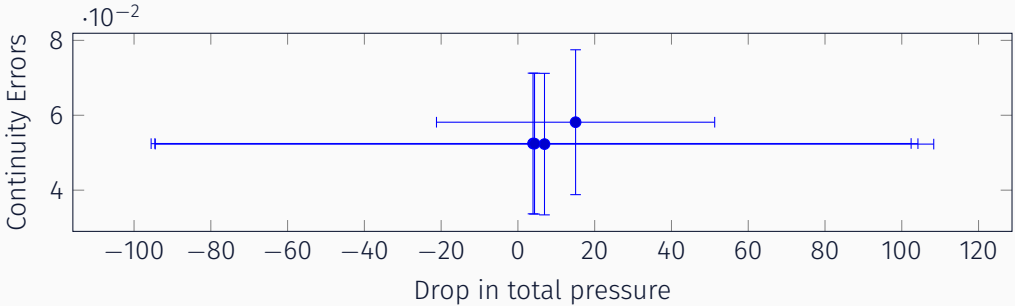
Hackathon challenge report: Stage 03





Hackathon challenge report: Stage 03

Garbage all over the place for the Pareto Frontier!



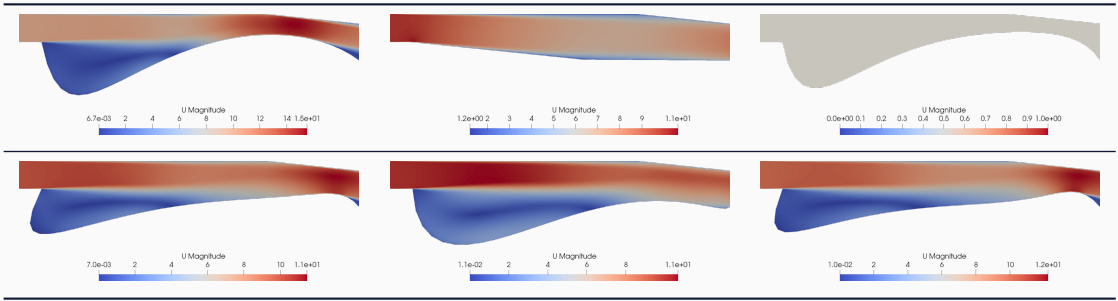


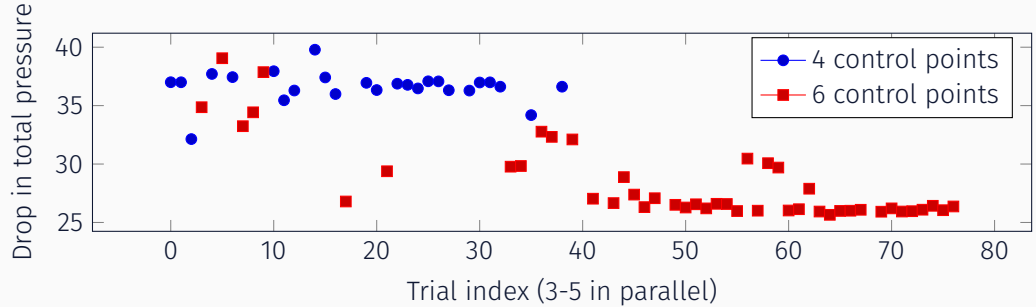
Figure 5: Highlights produced with GPEI (Don't really care about optimal configs...)

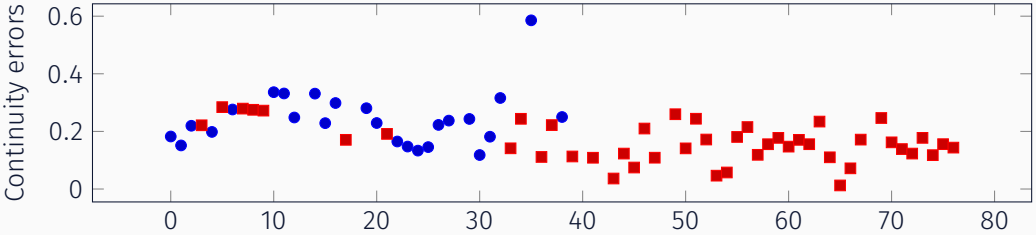
Hackathon challenge report: Stage 02

Parameter sets depening on number of curve control points

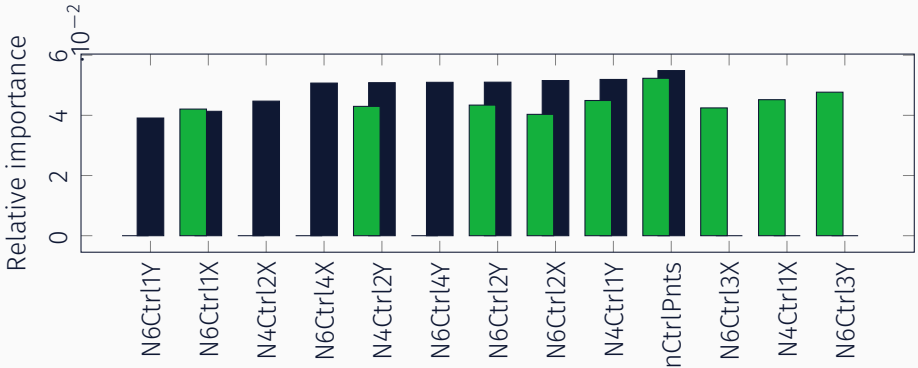
```yaml
parameters:
  nCtrlPnts:
    type: choice
    value_type: str
    values: ["4", "6"]
    is_ordered: False
    dependents:
      - "4": [N4NCurvePnts, N4Ctrl1X, N4Ctrl1Y, N4Ctrl2X, N4Ctrl2Y]
      - "6": [N6NCurvePnts, N6Ctrl1X, N6Ctrl1Y, N6Ctrl2X, N6Ctrl2Y, N6Ctrl3X, N6Ctrl3Y, N6Ctrl4X, N6Ctrl4Y]
  N4NCurvePnts:
    type: choice
    value_type: int
    values: [2, 30, 50]
    is_ordered: False
  # ...
  N6NCurvePnts:
    type: choice
    value_type: int
    values: [2, 30, 50]
    is_ordered: False
  N6Ctrl1X:
    type: range
    value_type: float
    bounds: [0, 0.0725]
    log_scale: False
  # ...
```

Hackathon challenge report: Stage 02





Hackathon challenge report: Stage 02



Green: Continuity errors, Dark blue: Pressure drop

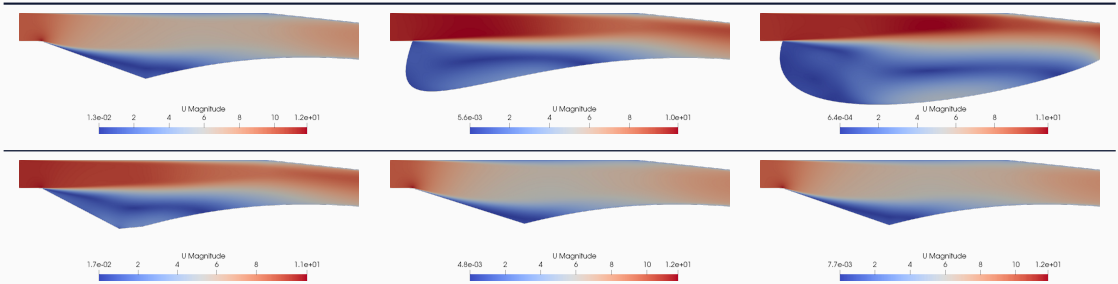


Figure 6: Highlights produced with GPEI (Optimal for Pressure drop: 2nd row - center, for continuity errors: 2nd row - right)

Conclusion

- Many features added to the toolkit!
  1. Manual selection of generation models (can chain models)
  2. Online visualization
  3. Automatic model selection based on parameter properties/constraints
  4. Better handling of dependent parameters
- Valuable insights from running the 1600+ into the optimal shape that **we can simulate**
- IF parameters naturally depend on each other; These relationships **must** be expressed.