Cloud-based CAD parametrization and image recognition for support of engineering design workflows using numerical simulations

Joel Guerrero
University of Genoa + Wolf Dynamics

Luca Mantelli + Sahrish Naqvi
University of Genoa

Presented at the 15th OpenFOAM workshop.
June 22-26, 2020. Arlington, VA, USA.
http://www.cpe.vt.edu/ofw15/

Who am I?

- My name is Joel Guerrero and I am an assistant professor at the University of Genova (Italy).
- I am also the CTO of Wolf Dynamics.
- My main areas of research are multi-physics simulations, numerical optimization, turbulence modeling, exploratory data analysis, data analytics, and interactive data visualization.
- Lately, I have been evangelizing about cloud computing, visual storytelling, and agile simulations.

Who is Wolf Dynamics?

- Wolf Dynamics is a spin-off of the University of Genova (innovative start-up).
- It was created to fill the gap between University and Industry in the Liguria region in Italy (and the world).
- We work with SMEs to help them become agile, innovate, and more competitive by using numerical simulations.
- But we also work with LEs mainly offering simulation software support, second opinion and validation services, and benchmarking services between commercial and open-source simulation applications.
- We also offer training services in the field of numerical simulations and serve as an incubator for new graduates looking to learn more about scientific computing.

wolf dynamics

multiphysics simulations, optimization & data analytics

Agenda

1. **Parameter-based** and parameter-free approaches for solid modeling
2. **Automatic loop** for design optimization or design space exploration
3. **Cloud-based** CAD – **Onshape**
4. **Image recognition** approach – **Python**
5. **Sample application** and live demonstration
6. **Future** vision
7. **Main** takeaways

1. **Parameter-based** and parameter-free approaches for solid modeling

Parameter-based and parameter-free approaches for solid modeling

- **Parameter-based** approaches work at the CAD level.
  - Gives the designer incredible level of control over the geometry.
  - A couple of parametrical variables are enough to make significant and well controlled changes in the final geometry.
  - Changes can be introduced easily.
  - The final geometry is ready to use for manufacturing or production.
  - It is a very mature method and widely used in industry.

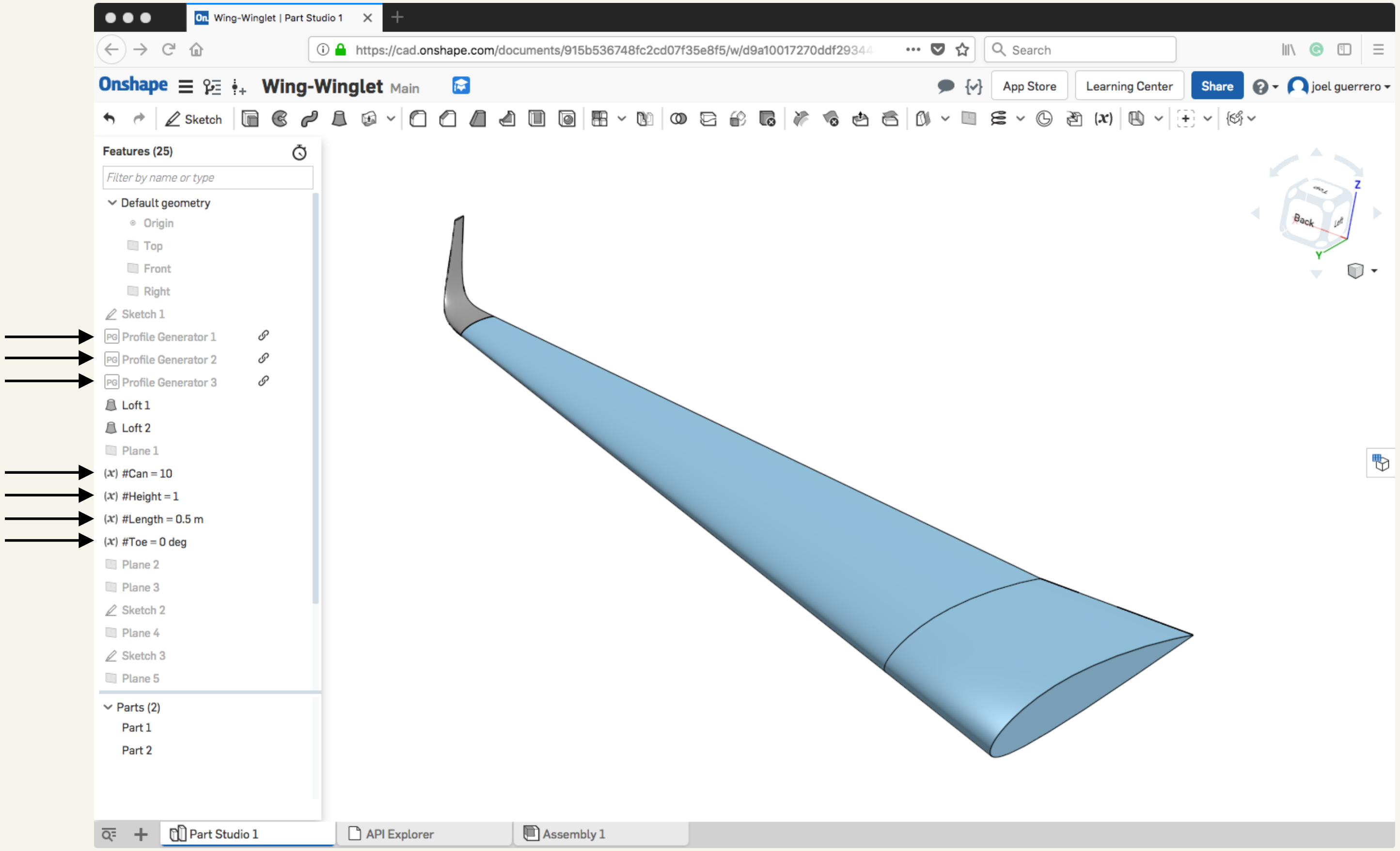
Parameter-based and parameter-free approaches for solid modeling

- **Parameter-based** approaches work at the CAD level.
    - The main difficulty of using a parameter-based approach when used in an automatic loop, is making the CAD application interact with the code coupling tool.
        - CAD applications are strongly coupled with the GUI.
        - Most of the CAD applications do not interact via script files or a programmatic way.
        - No application program interface (API) available.
        - They work in dedicated workstation running Windows OS.
        - The simulation software most of the time runs in Linux workstations or HPC hardware using UNIX like OS.

Parameter-based and parameter-free approaches for solid modeling

- In engineering design, **parameter-based** approaches are usually used with gradient-based methods, derivative-free optimization methods, and design space exploration techniques.

Parametric variables



Cant angle and toe angle variations

Parameter-based and parameter-free approaches for solid modeling

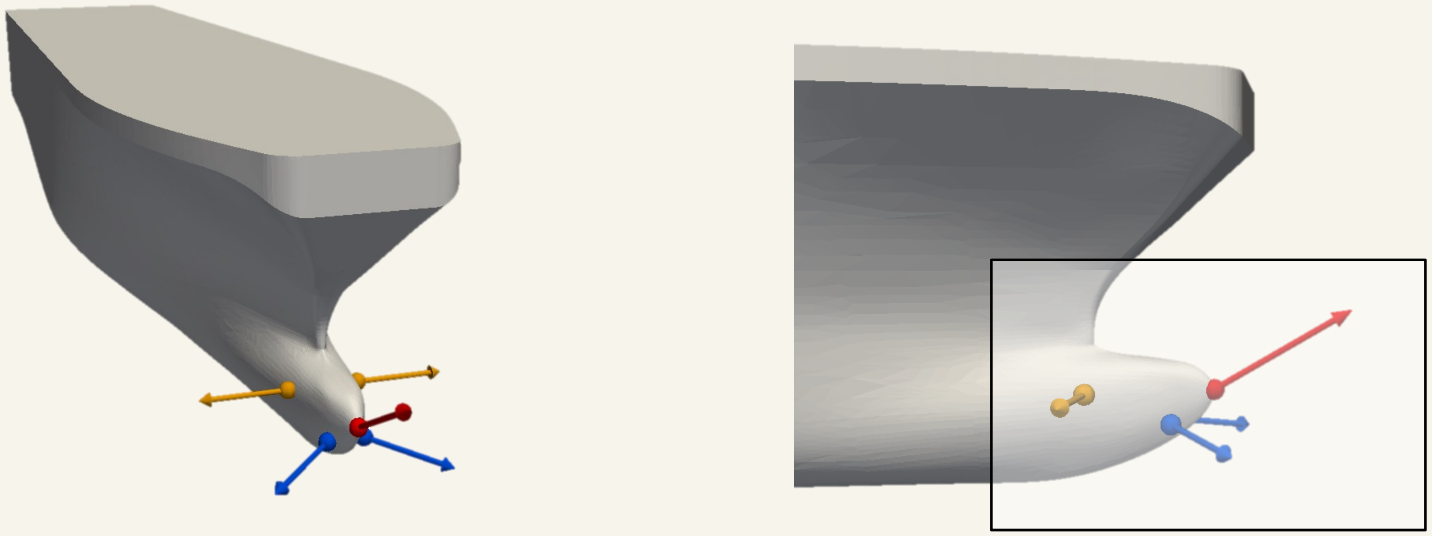
- **Parameter-free** approaches are usually related to surface modeling or direct modeling.
    - Conversely to parameter-based approaches, there is no need to assign parametrical variables.
    - These methods are very flexible as any point of the surface mesh can used to deform the solid model.
    - However, the flexibility gained does not necessarily means that the designer has complete control over the surface deformation.
    - It requires the careful selection of control points, lattice boxes or surfaces in order to define solid model deformation.
    - This method can also be used to modify the volumetric mesh without remeshing.

Parameter-based and parameter-free approaches for solid modeling

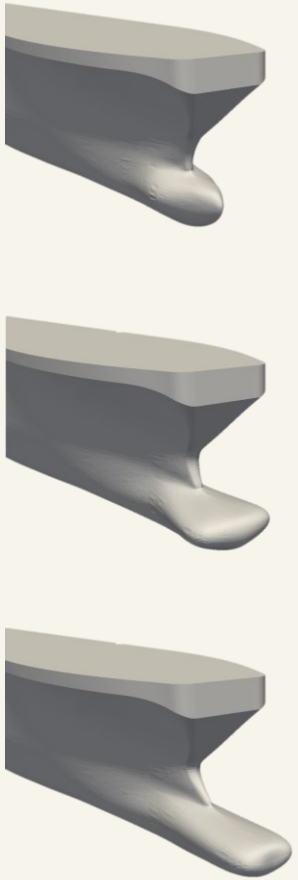
- **Parameter-free** approaches are usually related to surface modeling or direct modeling.
    - Surface modeling, free-form deformation, and direct modeling are ways of doing parameter-free solid modeling.
    - These kind of approaches are heavily used in applications where organic shapes are required (e.g., animation industry).
    - In CFD and Multiphysics simulations is usually used with adjoint methods.
    - Adjoint methods do not integrate naturally with feature-based CAD applications.
    - This approach is often limited to small and localized deformations.
    - With parameter-free methods, the designer loose complete control over the parametrization of the geometry.
    - These methods are not easy to control, and they tend to generate exoteric designs, which still are optimal, but neither easy to manufacture nor aesthetically pleasant.

Parameter-based and parameter-free approaches for solid modeling

- In engineering design, **parameter-free** approaches are usually used with adjoint optimization methods.



Control points and control box selection



2. **Automatic loop** for design optimization or design space exploration

Automatic loop for design optimization and design space exploration

- The automatic loop covers the workflow of a simulation:

Solid modeling → Meshing → Case setup → Simulations and monitoring → Post-processing

- We will illustrate the automatic loop using a simple application with many applications interacting.
- But have in mind that the framework can be easily extended to any engineering application (aerospace, automotive, HVAC, AEC, medical devices, thermal management, naval, and so on).

Automatic loop for design optimization and design space exploration

- The automatic loop covers the workflow of a simulation:

Solid modeling → Meshing → Case setup → Simulations and monitoring → Post-processing

- For the case that we will be presenting, the simulations are run using a pre-specified level of accuracy and iterative marching (which is not bad).
- However, by using data and metadata (data-of-data) to compute basic descriptive statistics and by leveraging a few concepts of SL/ML, the design loop can freely iterate until it reaches an acceptable level of convergence.

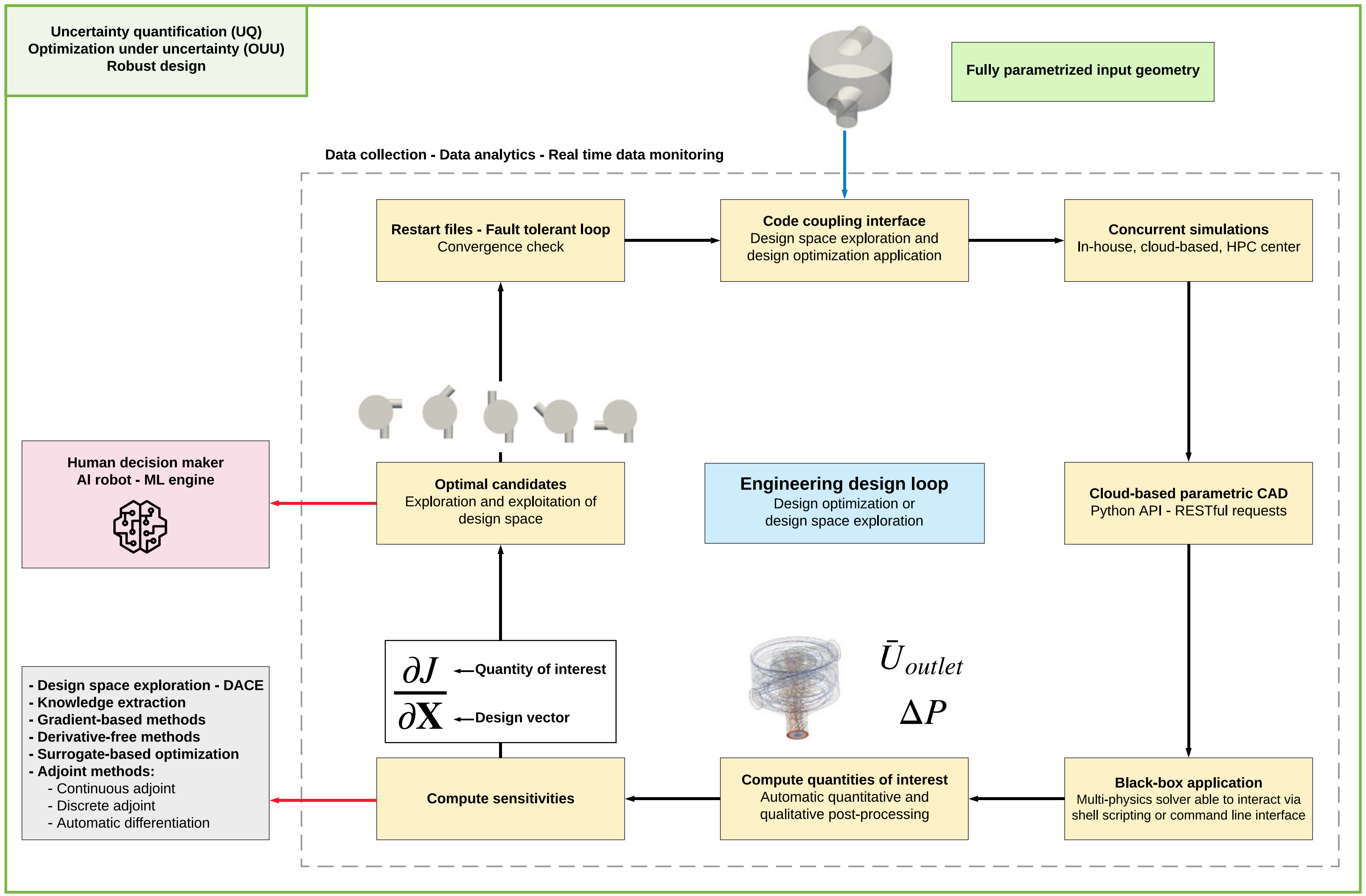
SL = statistical learning
ML = machine learning

Automatic loop for design optimization and design space exploration

- A few comments on the framework:
    - The framework is automatic and to some extent fault tolerant.
    - But in the case of fatal failure, the user can restart from the latest stable solution.
    - In the case of anomalies while the loop is running, the input parameters can be changed on-the-fly to stabilize the solution, this can be done automatically (a lot SL/ML involved) or manually.
    - To achieve this, a lot of things need to be monitored.
    - Therefore, it is important to monitor all the QOIs and KPIs real-time.
    - Every single modification is recorded and reported to the user.
    - <u>The bottleneck is the meshing stage.</u>
    - In case of meshing failure or bad quality meshes, the domain is remeshed using more robust parameters (which will increase the meshing time and mesh size).
    - If the mesh issues cannot be repaired in an automatic way, the user must fix the problems manually, which is not desirable.

Automatic loop for design optimization and design space exploration

- Graphical summary of an engineering design loop using a feature-based CAD – Tools to be used.



- **Code coupling/Optimizer:**
  DAKOTA
- **Concurrent computations scheduler:**
  DAKOTA
- **Parametric CAD:**
  Onshape (API)
- **Black-box solver:**
  OpenFOAM
- **Quantitative and qualitative post-processing:**
  Python, paraview, JavaScript
- **Real time data monitoring:**
  Python, R, BASH
- **Exploration and exploitation of design space:**
  Python, R, BASH
- **Additional automation scripting:**
  Python, BASH

All tools are open-source

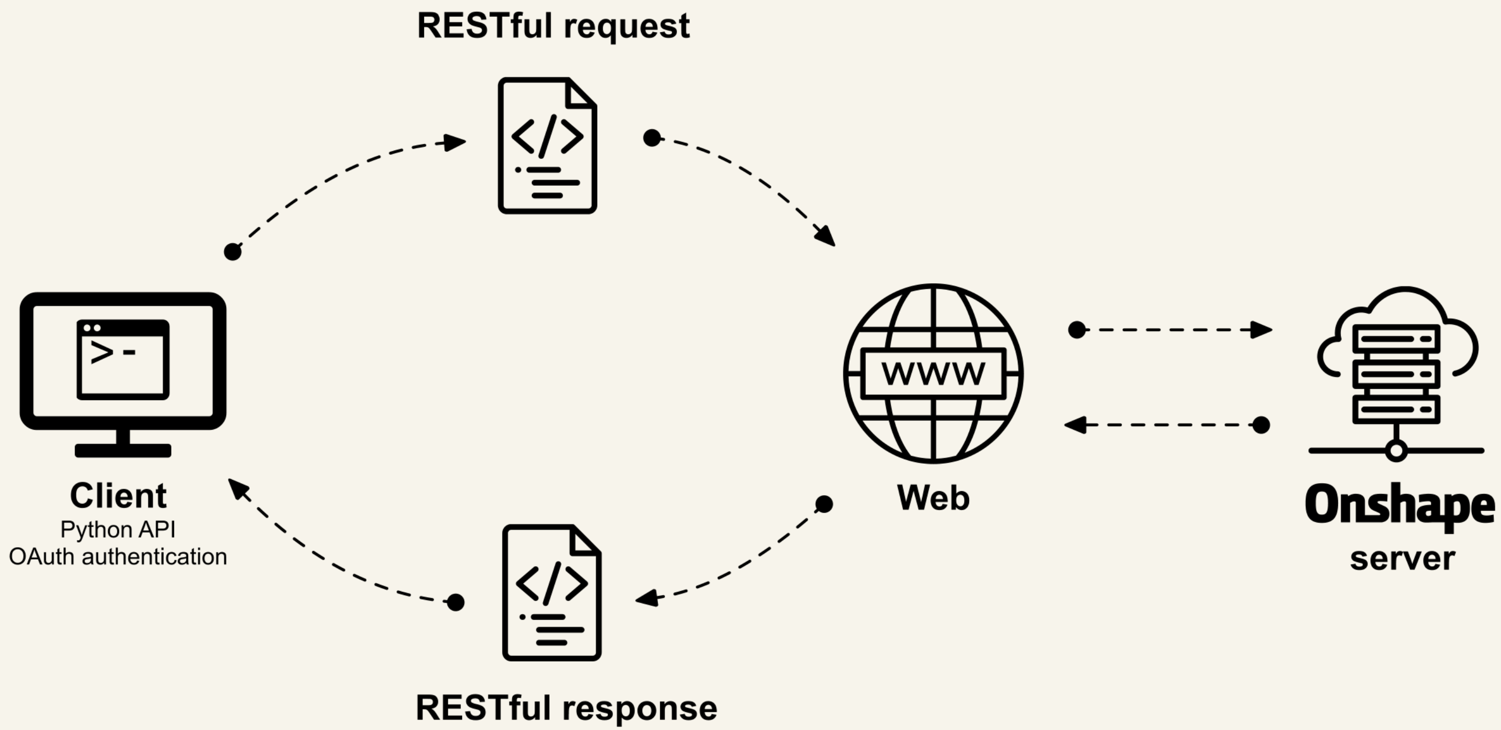
3. Cloud-based CAD – Onshape

Cloud based CAD - Onshape

- Using a feature-based, fully parametric CAD application gives the designer incredible level of control over the solid model.
- The problem with most CAD applications is that they do not work in Linux and they do not take input parameters using a programmatic language.
- To overcome this problem, we use Onshape (www.onshape.com).
    - Full cloud-based professional 3D CAD system.
    - Fully collaborative and simultaneous real time editing.
    - Version control, document management, data analytics, and sharable documents.
    - It runs on any device with a working web browser.
    - No need to install any software (besides the web browser).
    - Academic and public versions → Free.
    - Professional version → Monthly/annual subscription.
    - All versions share same capabilities.
    - RESTful API, so it can be scripted using python or nodeJS.

Cloud based CAD - Onshape

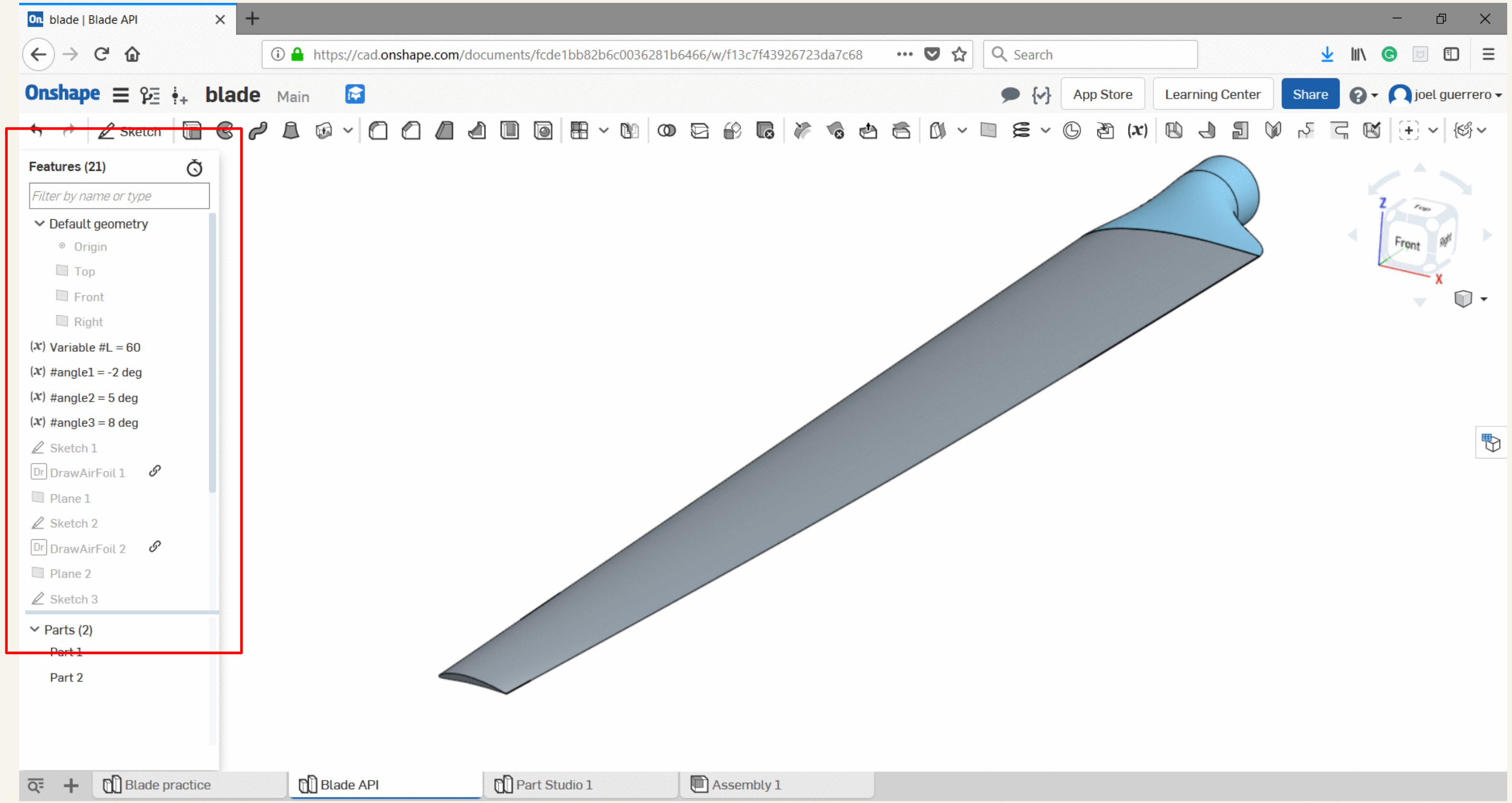
- By using Onshape RESTful API, we can close our design loop using a fully parametric CAD system.



Cloud based CAD - Onshape

- When using the API we do not interact with the GUI.
- RESTful requests → POST, GET, PUT, DELETE
  - Request a feature/document update or change.
  - Get the request response.
  - Download the new solid model in STL format or any supported CAD exchange format.

Any feature in the tree can be modified using Onshape RESTful API



Oauth authentication

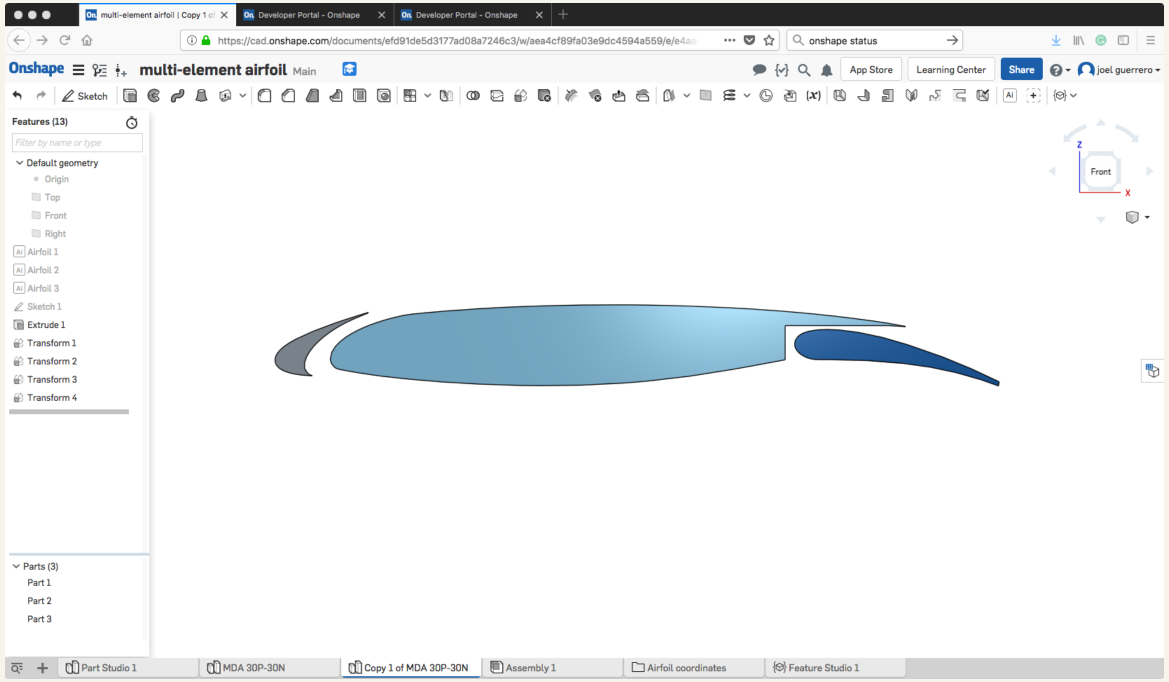
{ RESTful Request }

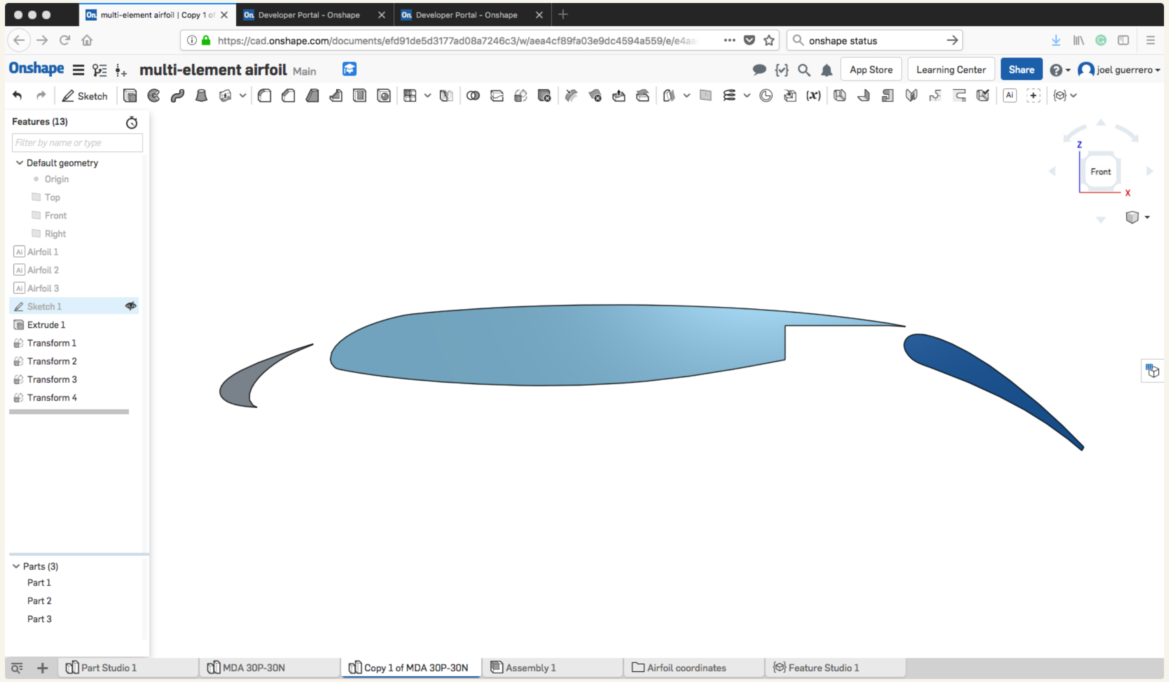
{ RESTful Response }

API – Python interface

Cloud based CAD - Onshape

- By using a fully parametric CAD, things such as this high-lift wing can be easily parametrized.
- Doing such modifications using mesh morphing is not that easy and robust.
- But if you are still interested in working at the mesh level, a workaround can be the use of overset meshes.





4. Image recognition approach – Python

Image recognition approach – Python

- The field of image recognition, image processing, object detection, and pattern recognition is very rich.
- In this work, to compute the similarity between two images, we propose the use of the Structural Similarity Index (SSIM) method, which is widely used for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos.
- The SSIM method consists in computing one single metric (SSIM index), which depends on the luminance (brightness) and contrast of the image.
- The closer the SSIM index is to one, the more similar the images are.

Target image

Image to compare
SSIM index = 0.7219

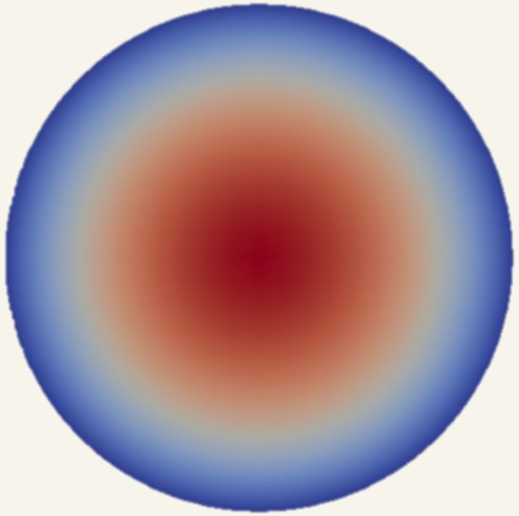


Image to compare
SSIM index = 0.1203

Image recognition approach – Python

- The SSIM index is computed as follows,

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

- The SSIM index evaluation is based on three characteristics quantities: luminance (l), contrast (c), and structure (s).

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$\mu_x = \frac{1}{N}\sum_{i+1}^{N} x_i$$

Luminance

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i+1}^{N}(x_i - \mu_x)^2}$$

Contrast

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i+1}^{N}(x_i - \mu_x)(y_i - \mu_y)$$

Structure

Image recognition approach – Python

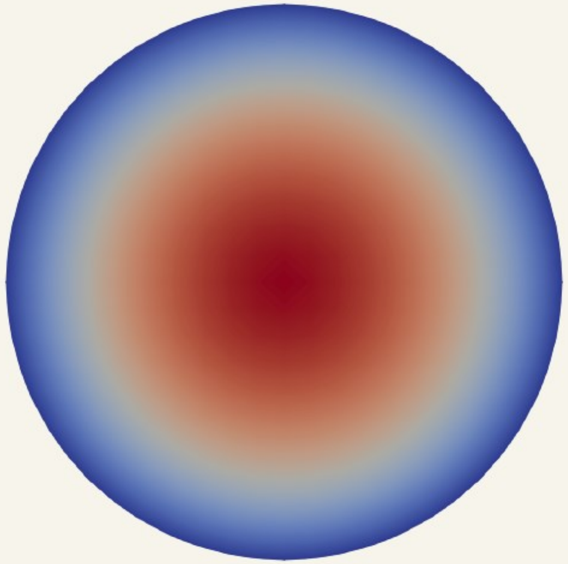
- In this work, we used the Python library scikit-image to process and analyze the images.

- Steps to compute the SSIM index:
  - The images to compare, one target image and the output image of the process that we are running, are saved as color images in digital format.
  - At processing time, the digital images are read and converted into arrays. They are also separated into RGB color channels (red, green, and blue).
  - At this point, each channel is a monochrome picture so that it can be treated as a grey-scale picture, and its SSIM index can be computed.
  - Then, the SSIM index of the two images can be obtained as the average of the SSIM of the RGB color channels of the images to compare.
  - The SSIM index value is a number between 0 and 1, where 1 means a perfect matching between the images. That is, the closer the value is to 1, the more similar the images are.

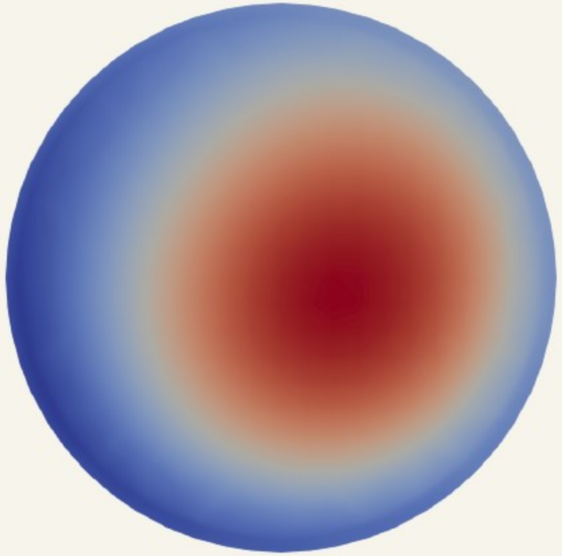
- We want to stress that more sophisticated methods exists, but for our purposes this method works very well.
- It is also interesting to mention that many image and pattern recognition techniques have been developed in the field of medical imaging.

Image recognition approach – Python

- Example output of the SSIM index.
  - The target image represents a reference velocity distribution (which can come from an experiment or designed manually).
  - The output image is the outcome of the simulation or process we are running.
  - The color scale is not important, what is important is to have images with similar brightness and contrast, similar color palettes, and the same image resolution.



Target image

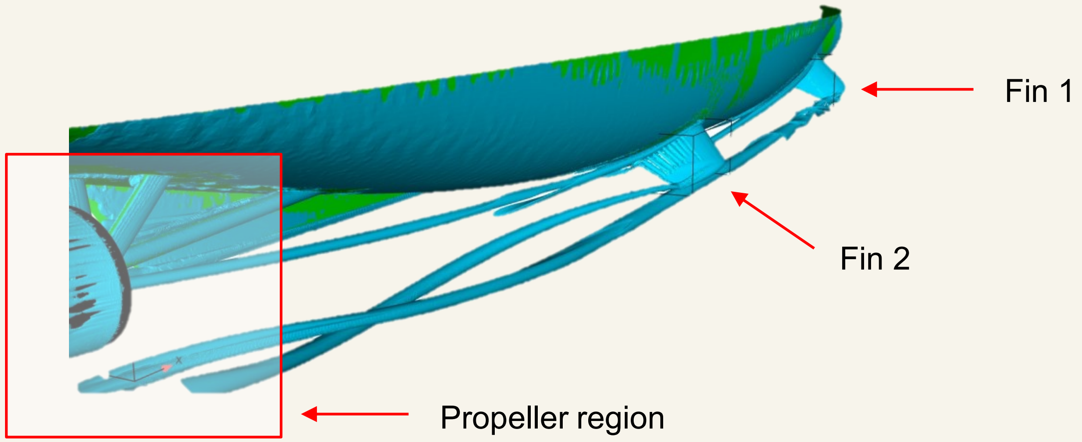


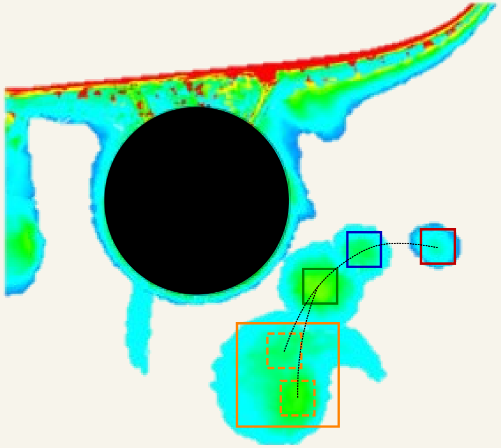
Output image (Image to compare)

SSIM = 0.9220752891

Image recognition approach – Python

- Even tough the SSIM method appears to be a little bit too simplistic, we have used it to deal with far more difficult applications.
    - For example, we have used image processing together machine learning techniques for vortex identification.
    - By using this approach, it is possible to track the trajectories of the vortex core for different configurations (in this case for different fins).
    - With the information gathered, we can construct regression models to predict the trajectories of the vortices.
    - In the example shown, all the image processing, statistical learning, and machine learning was done by an expert data scientists with no knowledge at all in fluid dynamics, CFD, or naval applications.

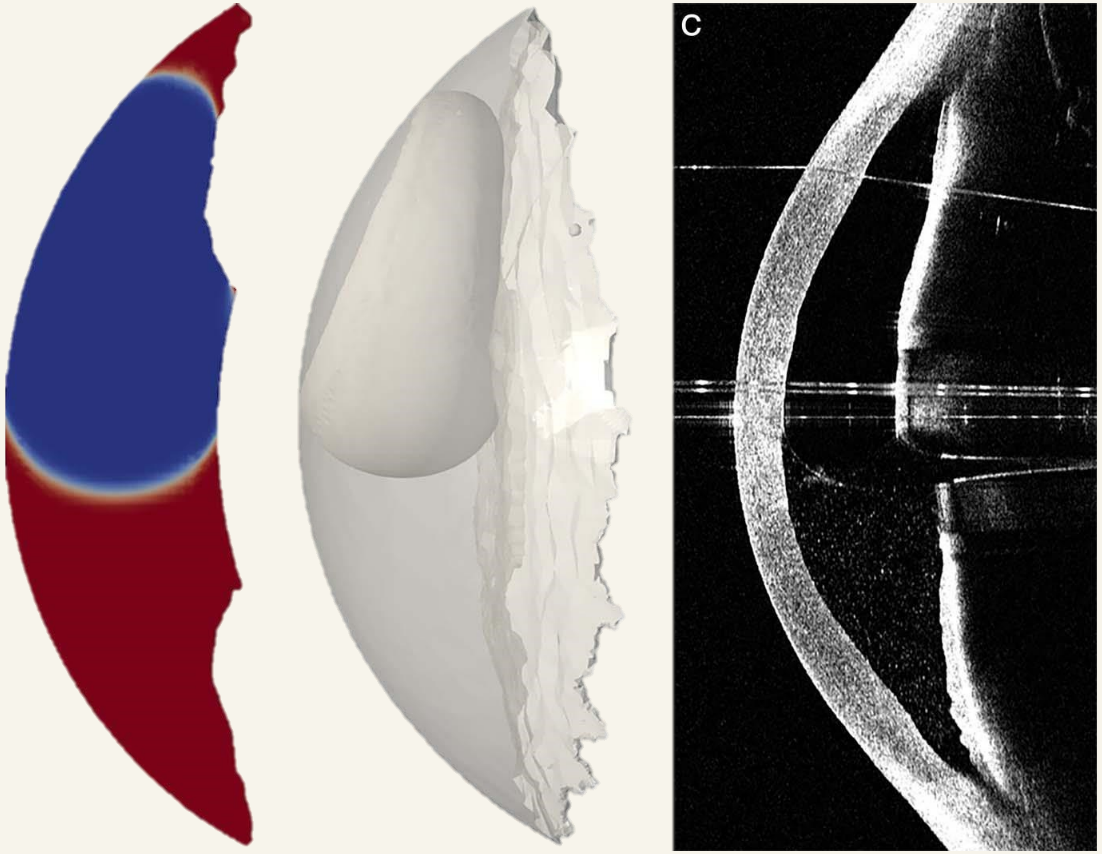


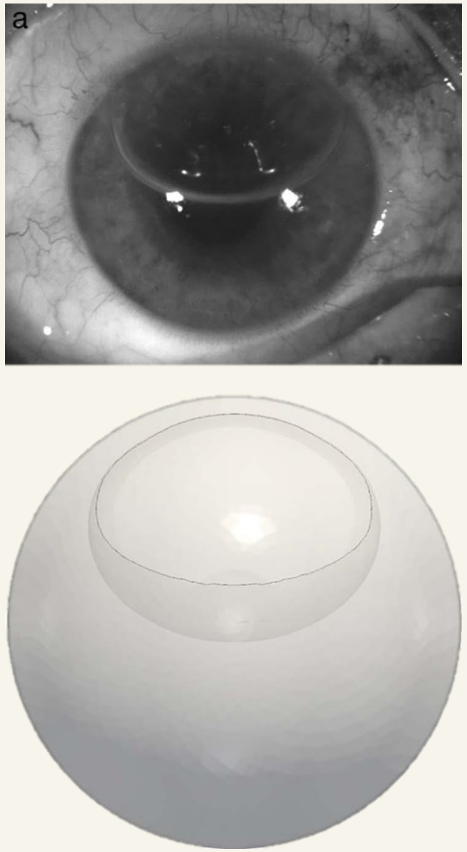


Vortex interaction in the propeller region

Image recognition approach – Python

- Even tough the SSIM method appears to be a little bit too simplistic, we have used it to deal with far more difficult applications.
  - This is a work in progress where we will use image recognition and numerical simulations for medical applications.



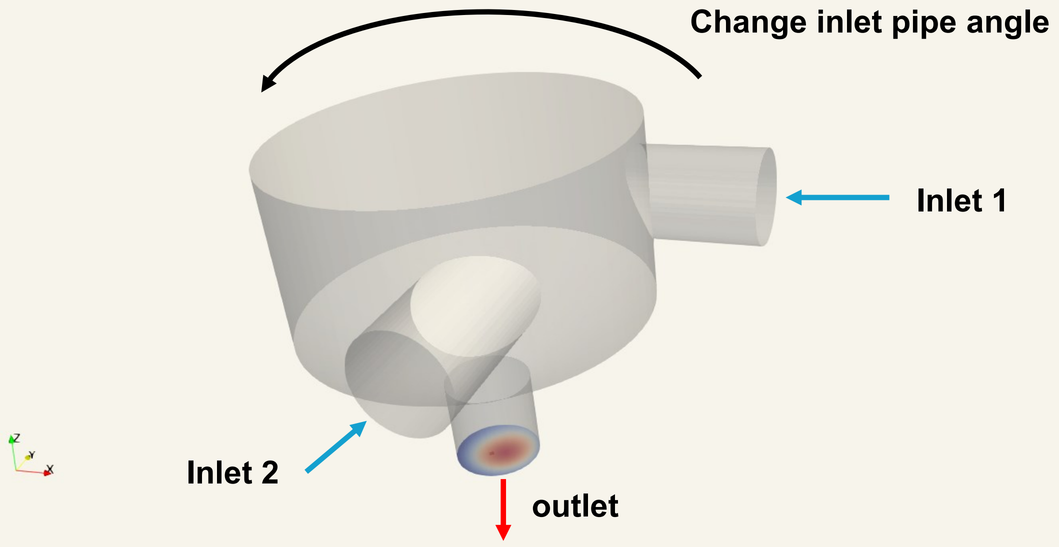


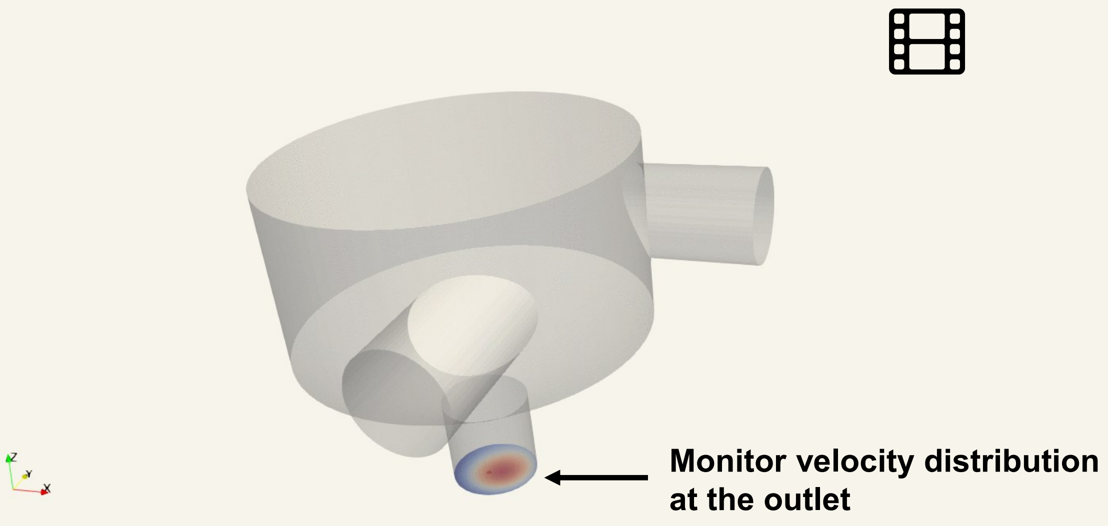
J. Pralits, M. Alberti, J. Cabrerizo. Gas-graft coverage after DMEK: a clinically validated numerical study. Translational Vision Science & Technology, Vol. 8(6), pp 1-14, 2019.

5. **Sample application** and live demonstration

Sample application and live demonstration

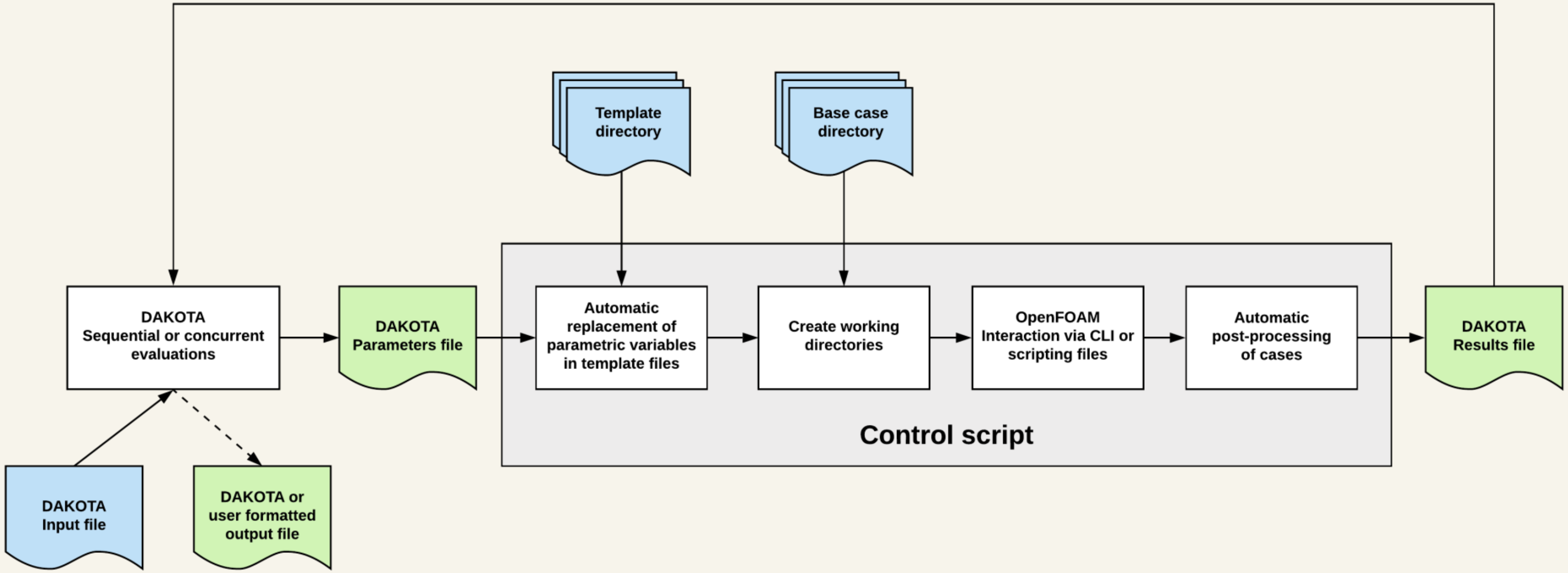
- Let us see the engineering design framework in action using a sample application.
- The main goal in this case is to obtain a given velocity distribution at the outlet by changing the angle of the inlet pipe 1 (refer to the figure below).
- The velocity distribution field at the outlet was designed in such a way that the velocity normal to the outlet surface has a paraboloid distribution.
- Then, by using the SSIM index method we can compare the target image with current image.





Sample application and live demonstration

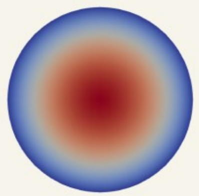
- Workflow for data exchange between DAKOTA and OpenFOAM.
  - The white rectangles denote process blocks, light-shaded blue document symbols denote unchanging sets of files, and light-shaded green document symbols indicate files that change with each set of design parameters generated by DAKOTA or after the end of the evaluation of the QoI.
  - The light-shaded grey area denotes the domain of the control script that automatically prepares the case; this includes, CAD geometry, mesh generation, launching the solver, quantitative and qualitative post-processing, and automatic formatting of input and output files.
  - It is worth mentioning that the workflow is similar for different black-box applications, the only difference is in the formatting of the input and output files, and the data structure.



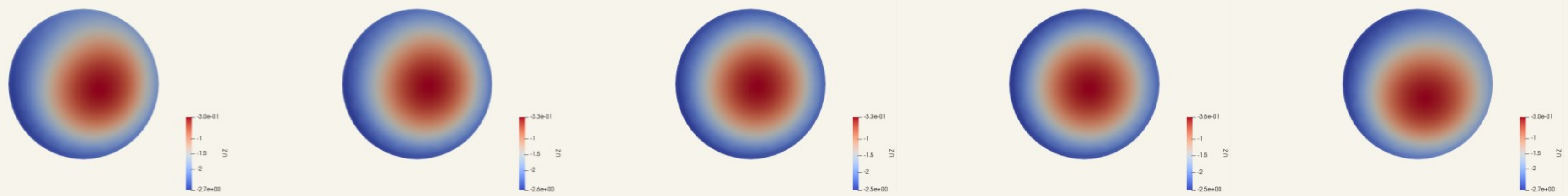
Sample application and live demonstration

- Qualitative comparison of velocity distribution at the outlet.
- The SSIM method was used to compare the images.
- The SSIM index value is bounded between 0 and 1 → A value of 1 means that the images are identical.

Target velocity distribution at outlet →



Velocity distribution at outlet →



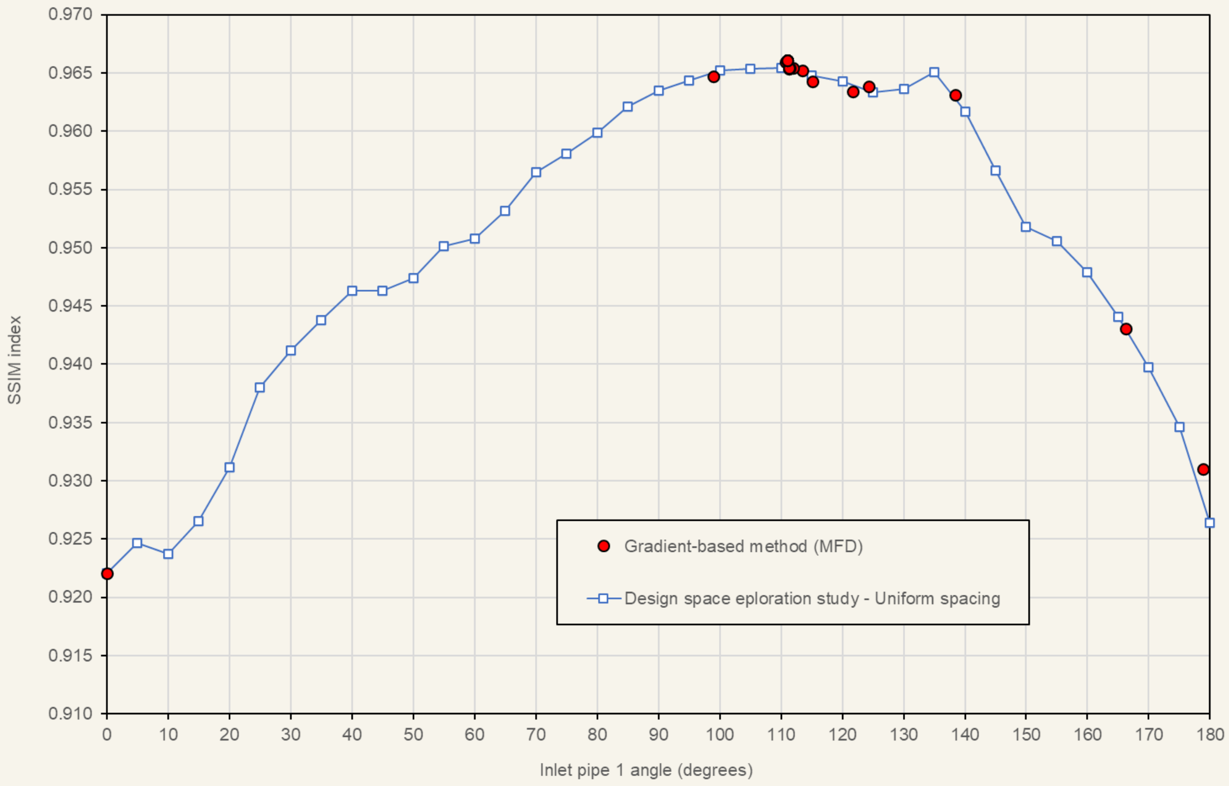
Geometry variations →

Inlet pipe angle →	0°	45°	90°	135°	180°
SSIM →	0.9220752891	0.946307357	0.9634644796	0.9650353853	0.9263740248

Sample application and live demonstration

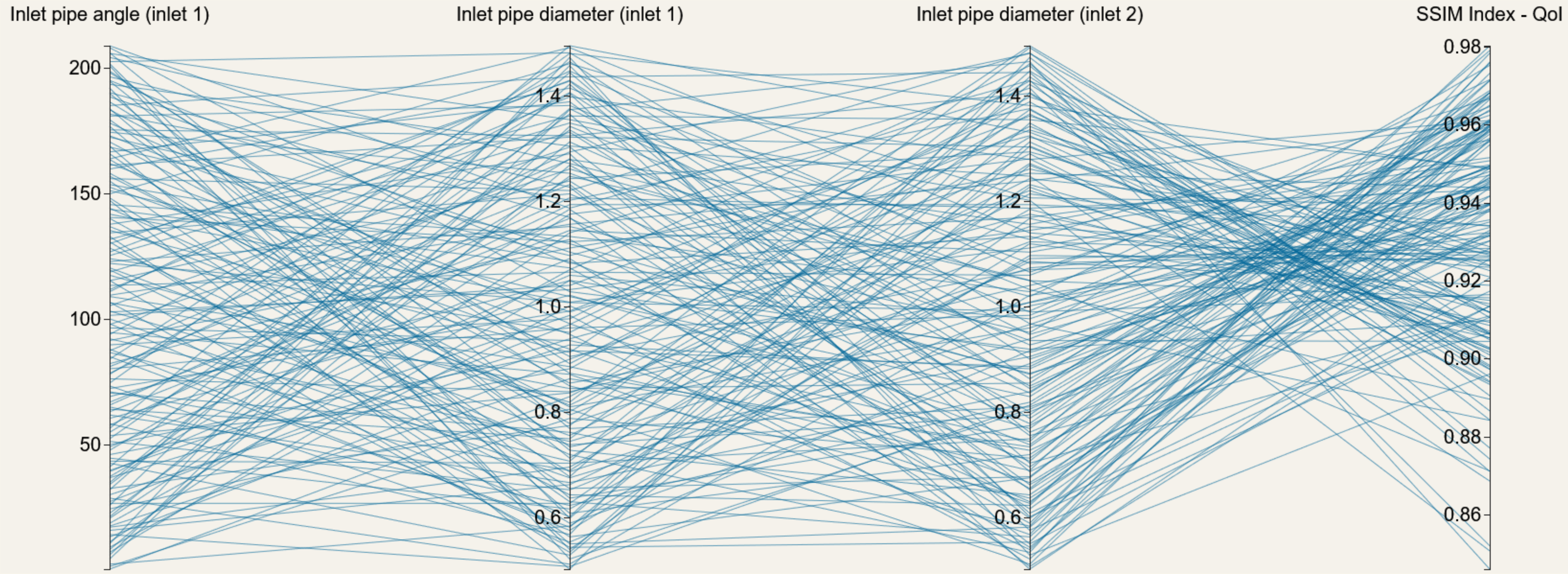
- Comparison of the outcome of a DO study and a DSE study.

- The DO study was conducted using the method of feasible directions (gradient-based method) with numerical gradients computed using forward differences.
- For the DO case, the starting point was 0 degrees, and the case converged to the optimal value in 31 function evaluations.
    - Optimal value: pipe angle equal to 111.0549 degrees and SSIM index equal to 0.9660
- In the DSE case, we explored the design space from 0 to 180 degrees, in steps of 5 degrees (36 function evaluations).
- So roughly speaking, we used the same number of function evaluations as for the DO case.
- The DSE study, while not formerly converging to the optimal solution, gives more information about the design space than the DO method.



Sample application and live demonstration

- This case can be easily extended to more design variables.
- The use of exploratory data analysis techniques is of extremely importance when studying high dimensional design spaces.
- In the figure below, the outcome of a case with three design variables is visualized using parallel coordinates (interactive).

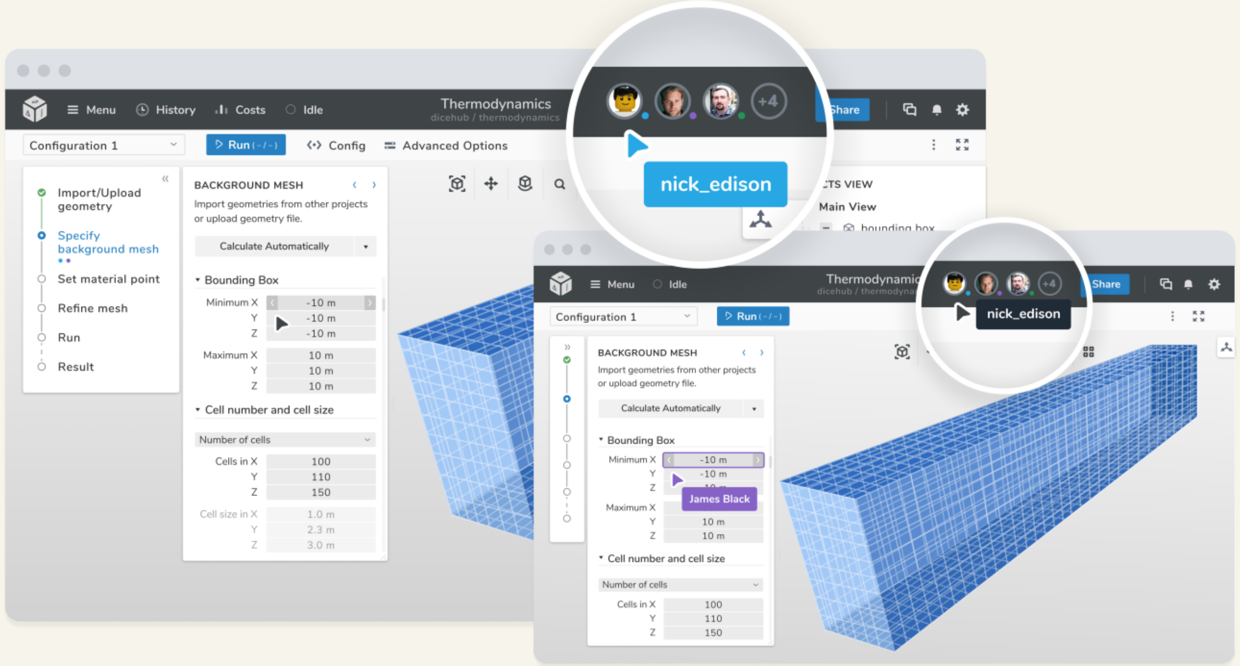


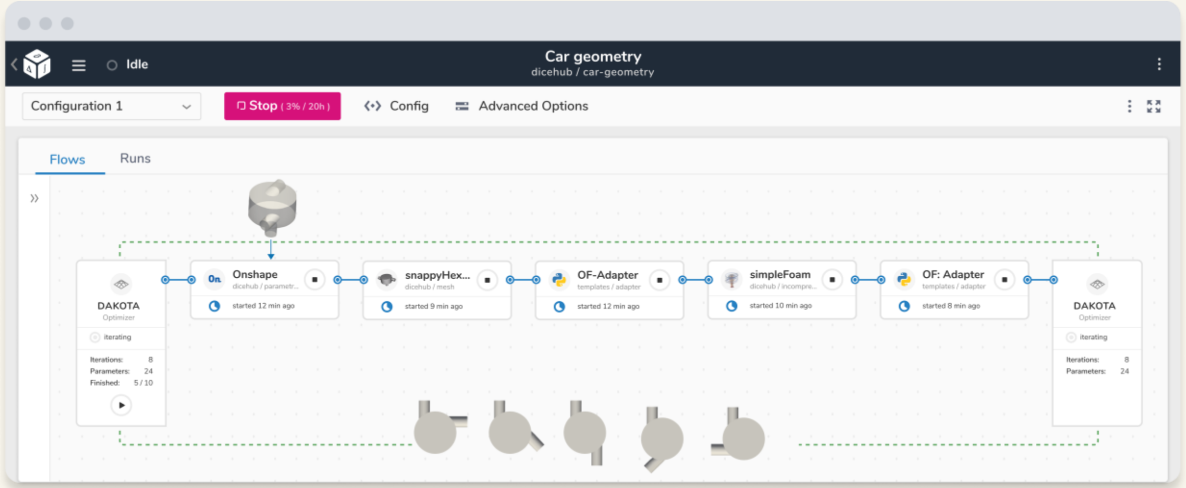
https://joelguerrero.github.io/parallel_coordinates_dse_case/

6. **Future** vision

Future vision

> “Everyone working together on the same document, real-time, on any-device, anywhere.”





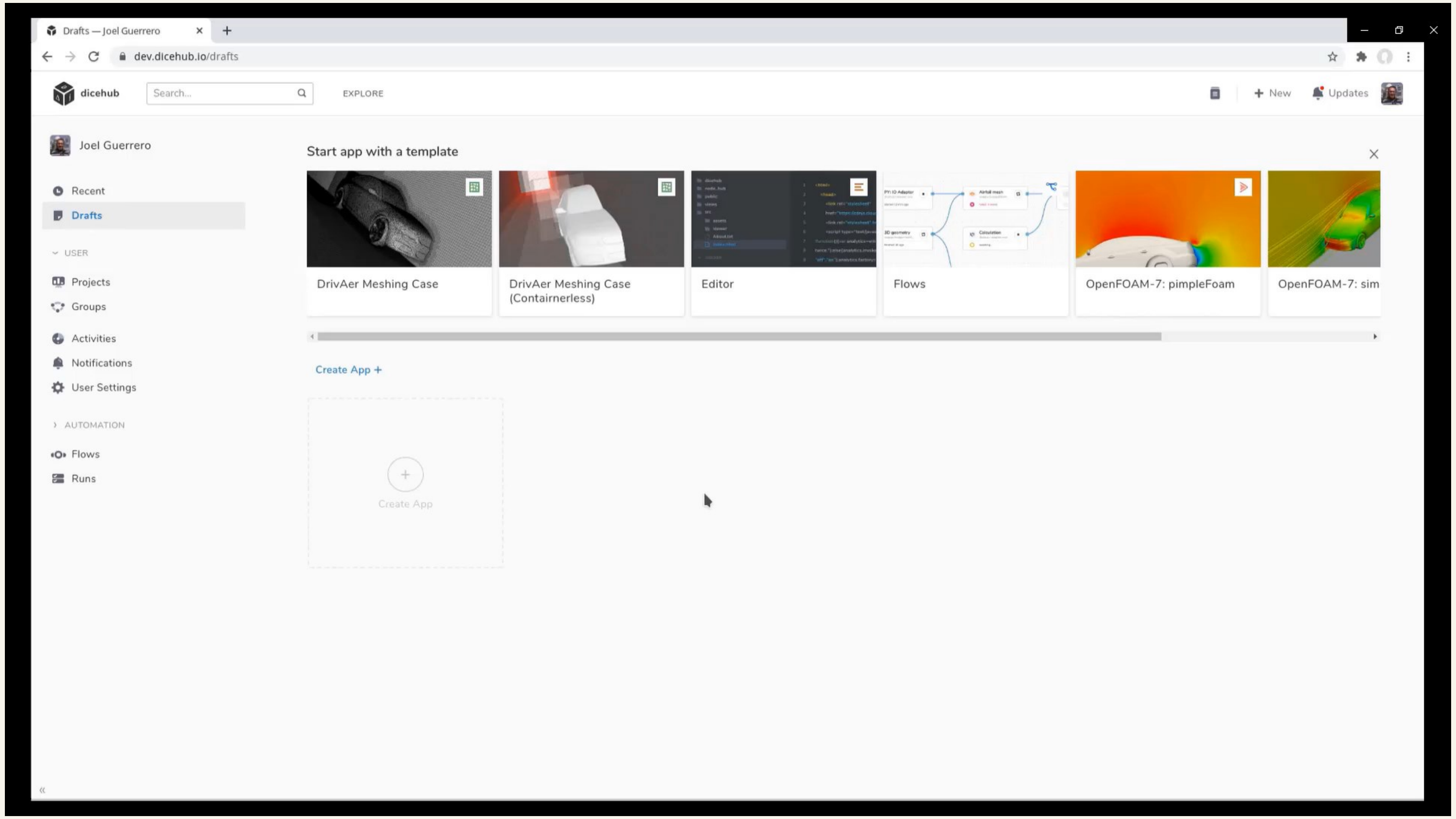
- Agile product development, real-time collaboration, automation, concurrent tasks, rapid iterations, data driven insight, interactive data analysis

Future vision

- Our vision is shared with the developers of dicehub (https://about.dicehub.com/).
- dicehub is a cloud based all-in-one application simulation operations (SIMOPS) with,
  - Built-in real-time collaboration.
  - Data management solutions.
  - Workflow automation.
  - Machine learning and data analysis functionalities.
  - Code coupling capabilities.
  - High-performance scaling capabilities on local resources, HPC centers, or the cloud.

Future vision

dicehub in action.



http://www.wolfdynamics.com/training/OF_WS2020/dicehubvideo/dicehubvideo_player.html

- **Contact information:**
  - https://about.dicehub.com/
  - rostyslav.lyulinetskyy@dicehub.com

7. Main takeaways

Main takeaways

- The use of the cloud-based CAD application (Onshape), allowed us to implement a CAD feature-based engineering design loop.
- Thanks to cloud-based technologies we overcame the problems related to installation, operating system compatibility, access to files, and so on.
- Collaborative tools and the cloud are making their way into CFD to help us become more agile and deliver increased design innovation.
    - Everyone working together on the same document, real-time, on any-device, anywhere.

Main takeaways

- Implementing an engineering design loop is a meticulous and thoughtful process that requires careful planning.
- Always monitor and analyze your data (quantitative or qualitative) real-time.
- Validate and calibrate your design loop, be sure that is fault tolerant, accurate, and robust.
- We all want rapid iterations; however, do not sacrifice solution accuracy over solution speed. Design engineering loops are time consuming.
- Leverage your computational resources (local, remote, or on the cloud) and deploy concurrent tasks.

FYI – Useful links

- You can download the working case at the following link (GitHub),
  - https://github.com/joelguerrero/cloud-based-cad-paper

- You can find a complete description of the methodology presented at the following link,
  - https://www.mdpi.com/2311-5521/5/1/36

- Applications used:
  - OpenFOAM 7 → https://openfoam.org/
  - DAKOTA 6.11 → https://dakota.sandia.gov/
  - Anaconda Python (versions 2.7 and 3.7) → https://www.anaconda.com/
  - Onshape (web-based app. and API) → https://www.onshape.com/
  - Paraview 5.6.1 (headless mode) → https://www.paraview.org/

Be collaborative, be innovative, be cloud

wolf dynamics
multiphysics simulations,
optimization & data analytics

Let's connect

guerrero@wolfdynamics.com

www.wolfdynamics.com