Cloud-based CAD parametrization for engineering design workflows in OpenFOAM®

Joel Guerrero
University of Genoa + Wolf Dynamics

Luca Mantelli + Sahrish Naqvi
University of Genoa

Presented at the Fourth Midwest OpenFOAM® User Group Meeting
October 4-5, 2019. Minneapolis, Minnesota, USA.
https://sites.google.com/view/mofug2019v1/home

Who am I?

- My name is Joel Guerrero and I am a researcher 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, 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 (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 validation services for assessing the transition from commercial software to open-source applications.
- We offer training services and serve as an incubator for new graduates looking to learn more about scientific computing.

wolf dynamics

multiphysics simulations,
optimization & data analytics

www.wolfdynamics.com

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. **Live** demonstration
5. **Main** takeaways

1.

Parameter-based and parameter-free approaches for solid modeling

Parameter-based and parameter-free approaches for solid modeling

- When we talk about engineering design workflows, we refer to design optimization (DO) or design space exploration (DSE).
- We iterate in a scientifically sounded way to design the product (no trial-and-error).
- Engineering design loops highly rely in automating all tasks in the simulation workflow (solid modeling, meshing, case setup, post-processing, and so).
- Hereafter, we will focus on the solid modeling part which is the starting point of the simulation workflow.
- When it comes to solid modeling, there are two approaches:
    - Parameter-based or feature-based.
    - Parameter-free or direct-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.
  - 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. They do not interact via script files or 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.

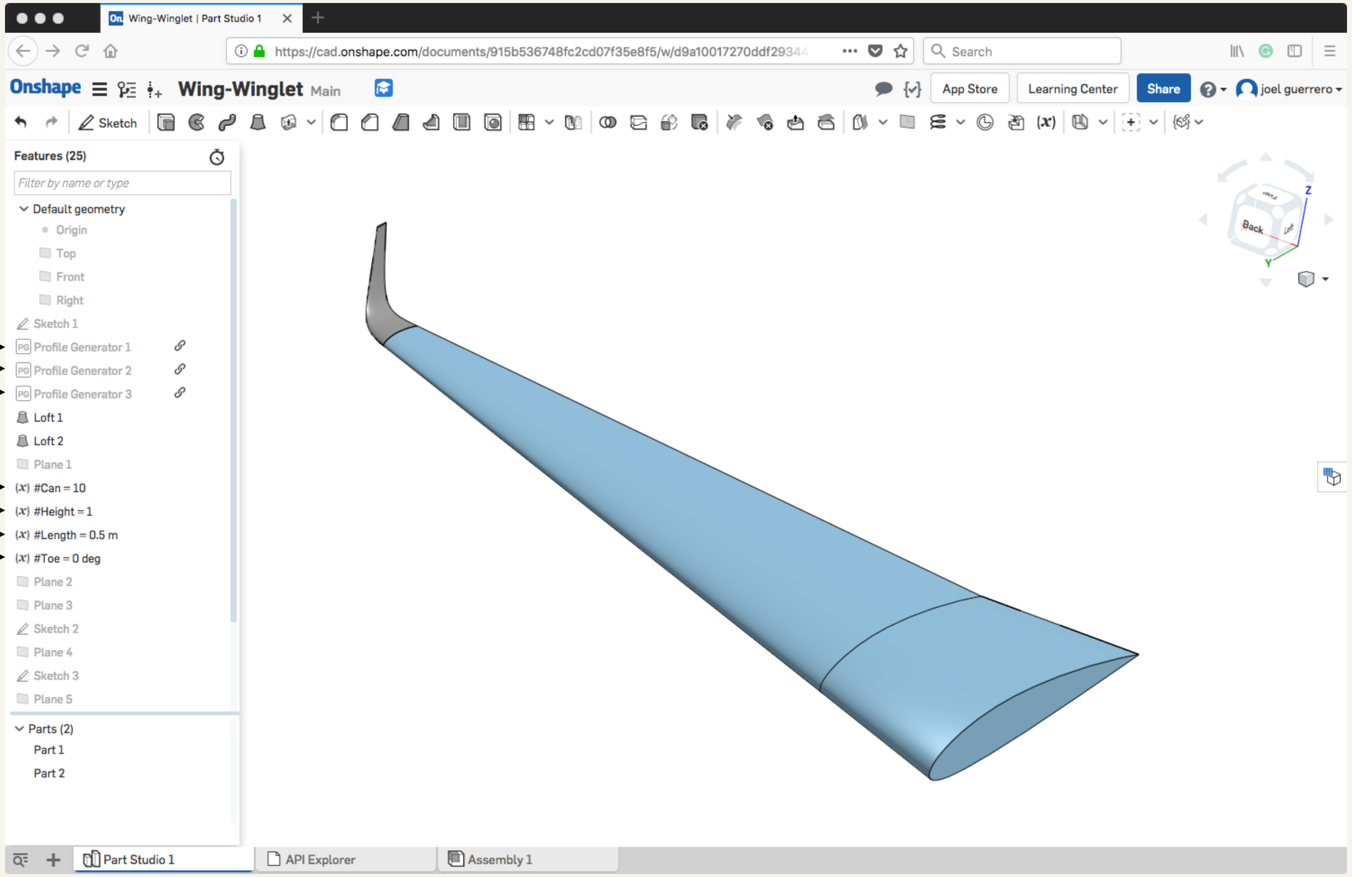
Parameter-based and parameter-free approaches for solid modeling

- **Parameter-free** approaches are usually related to surface modeling or direct modeling.
    - There is no need to assign parametrical variables. This gives a lot flexibility as any point 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 selection of control points, lattice boxes or surfaces in order to define solid model deformation (it also can be used to modify the volumetric mesh without remeshing).
    - Surface modeling, free-form deformation, and direct modeling are ways of doing parameter-free solid modeling.
    - This kind of approaches are heavily used in applications where organic shapes are required (e.g., animation industry).
    - In the CFD and Multiphysics simulations is usually used with adjoint methods. However, this approach is often limited to small and localized deformations.
    - Adjoint methods do not integrate naturally with feature-based CAD applications.

Parameter-based and parameter-free approaches for solid modeling

- In engineering design, **parameter-based** approaches are usually used with gradient-based and derivative-free optimization methods, and design space exploration studies.
- It is relative to easy to implement and control.

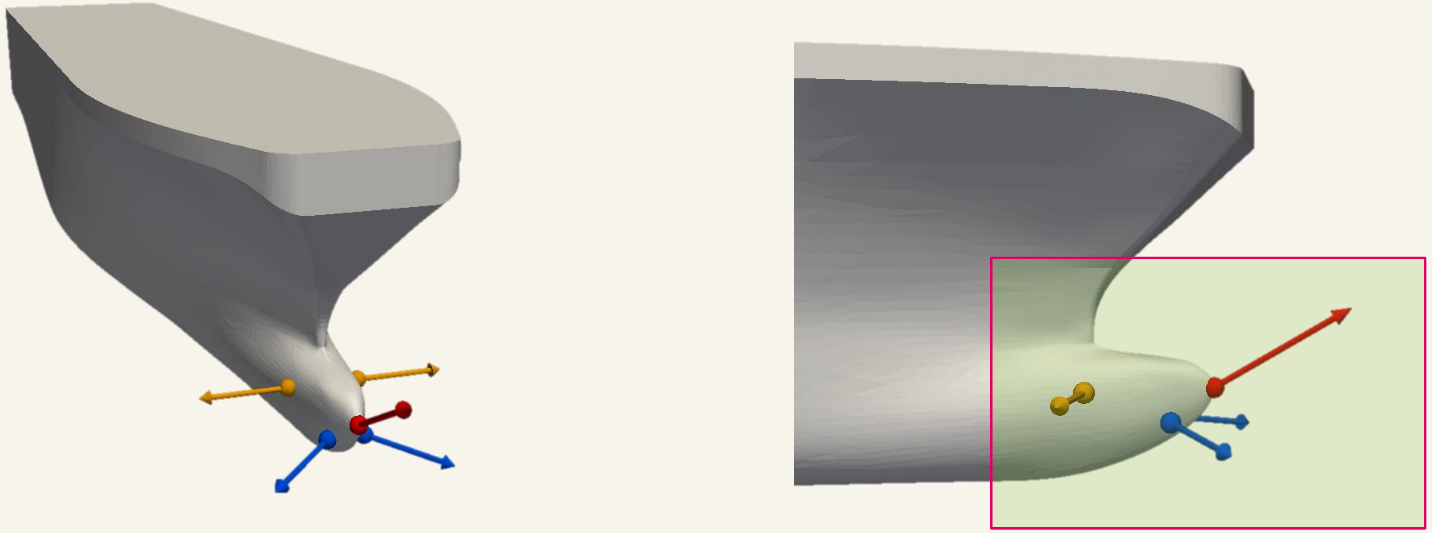
Parametric variables



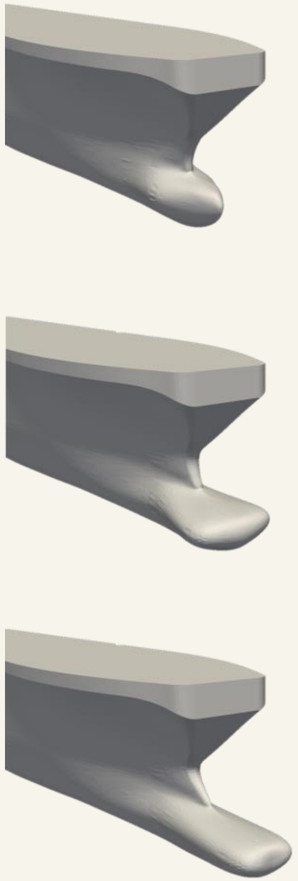
Cant angle and toe angle variations

Parameter-based and parameter-free approaches for solid modeling

- In engineering design, **parameter-free** approaches are usually used with adjoint optimization methods.
- It is not easy to control, and it can generate exoteric geometries (with are optimal but not easy to parametrize or manufacture).



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

- We will illustrate the automatic loop using a simple application.
- 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).
- 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 acceptable levels of convergence.

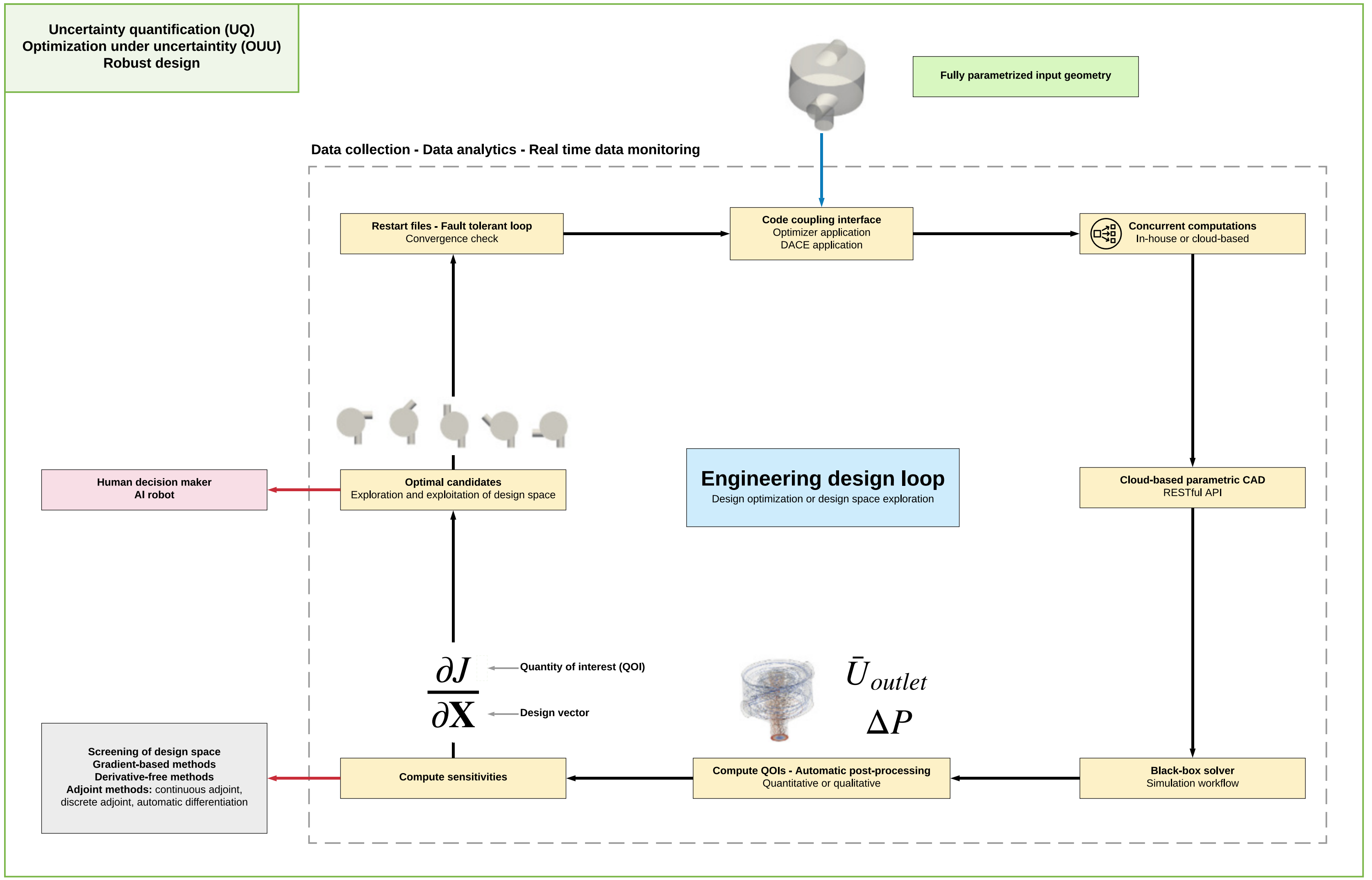
SL = statistical learning
ML = machine learning

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

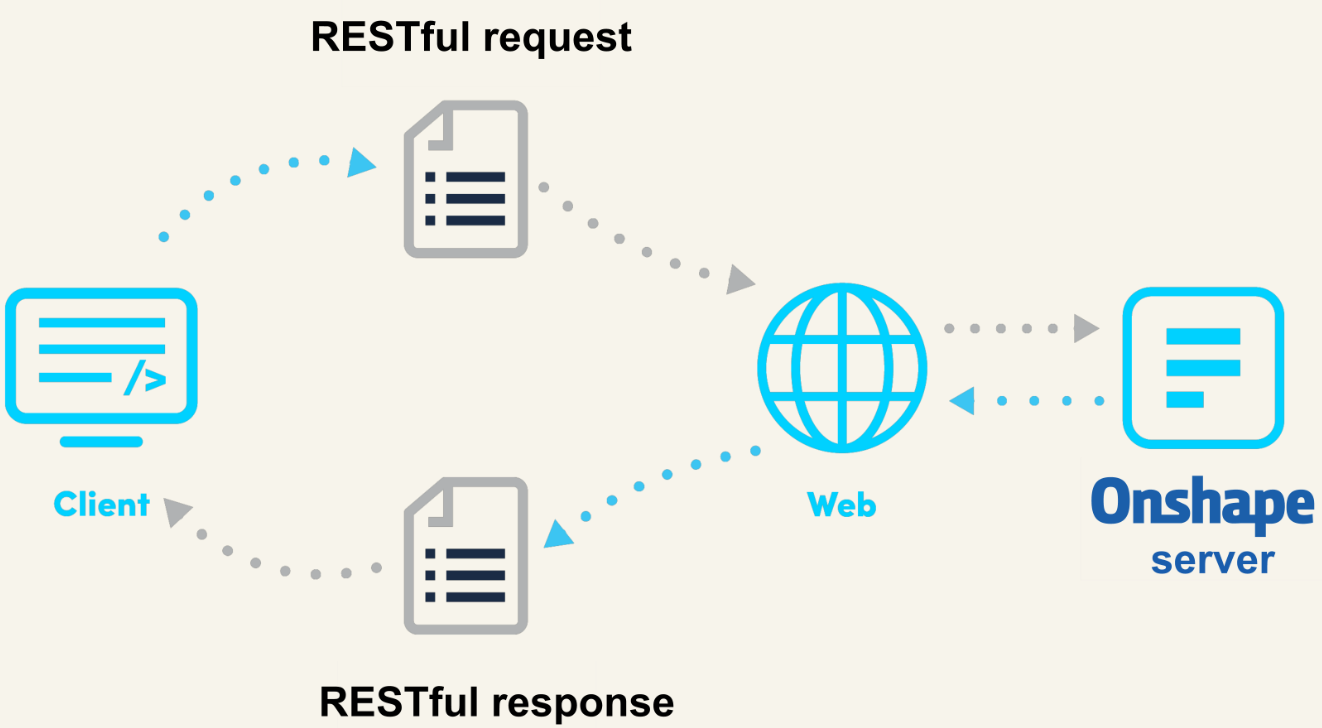
3. Cloud-based CAD – Onshape

Cloud based CAD - Onshape

- Using a feature-based, fully parametric CAD application gives the designer incredible level of control over the solid model.
- In our design loop, 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

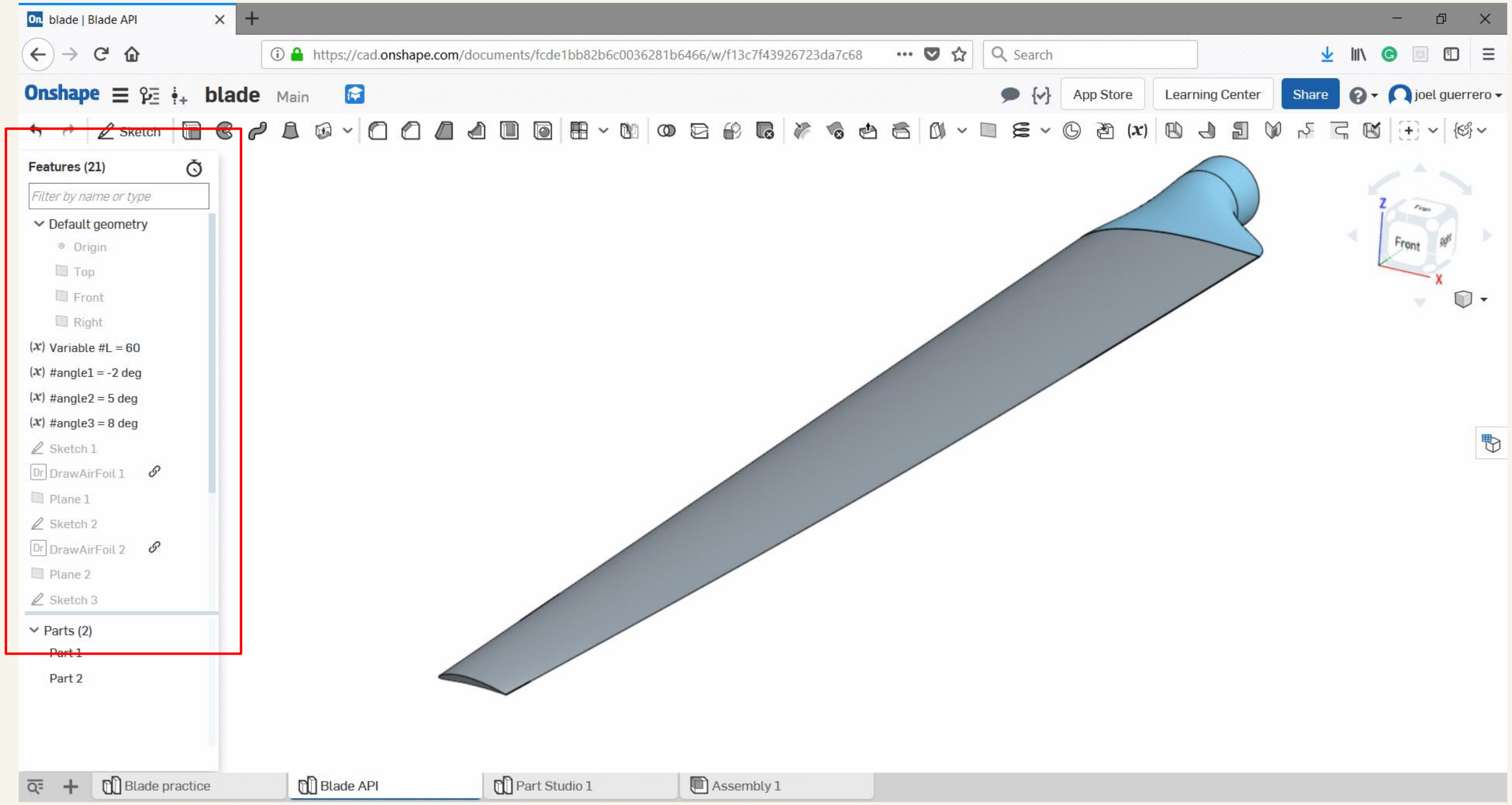
- By using Onshape RESTful API, we can close our design loop using a fully parametric CAD system.



Cloud based CAD - Onshape

- When using the API when do not interact with 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

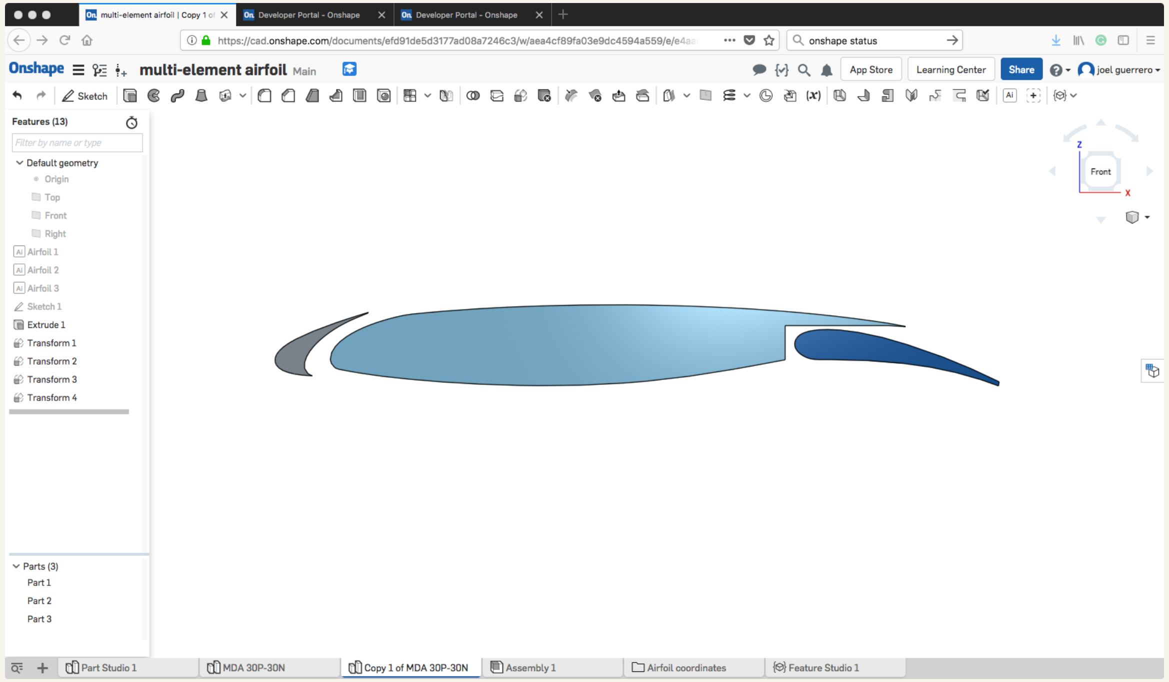
{ **RESTful Request** }

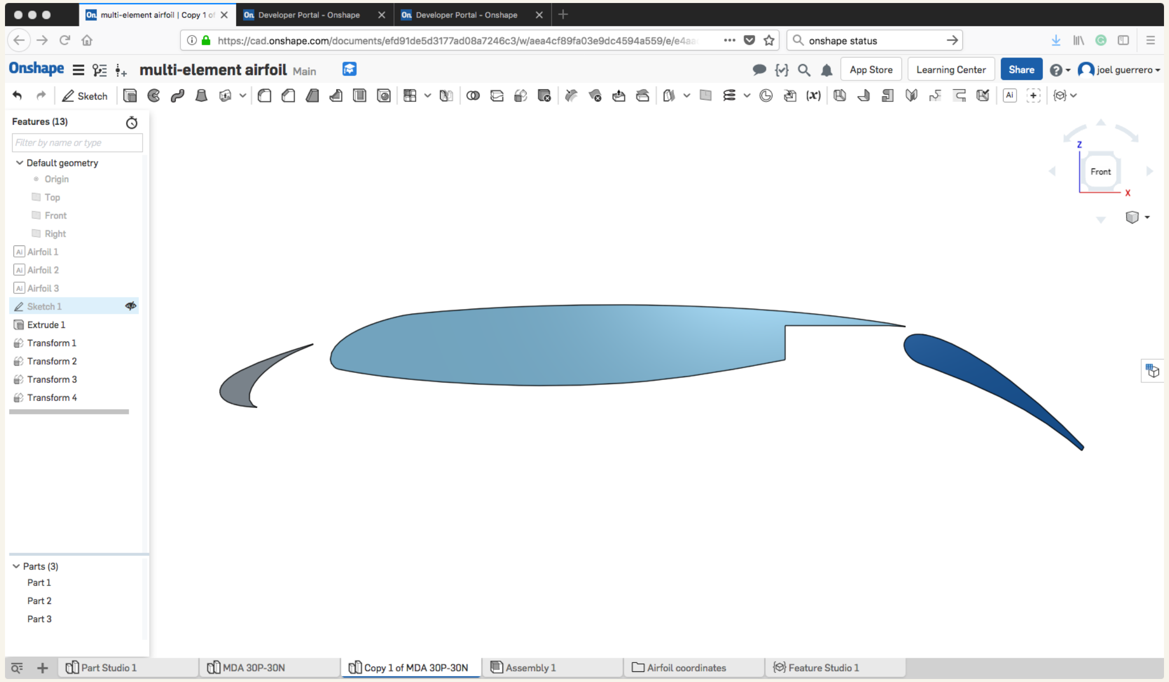
{ **RESTful Response** }

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.

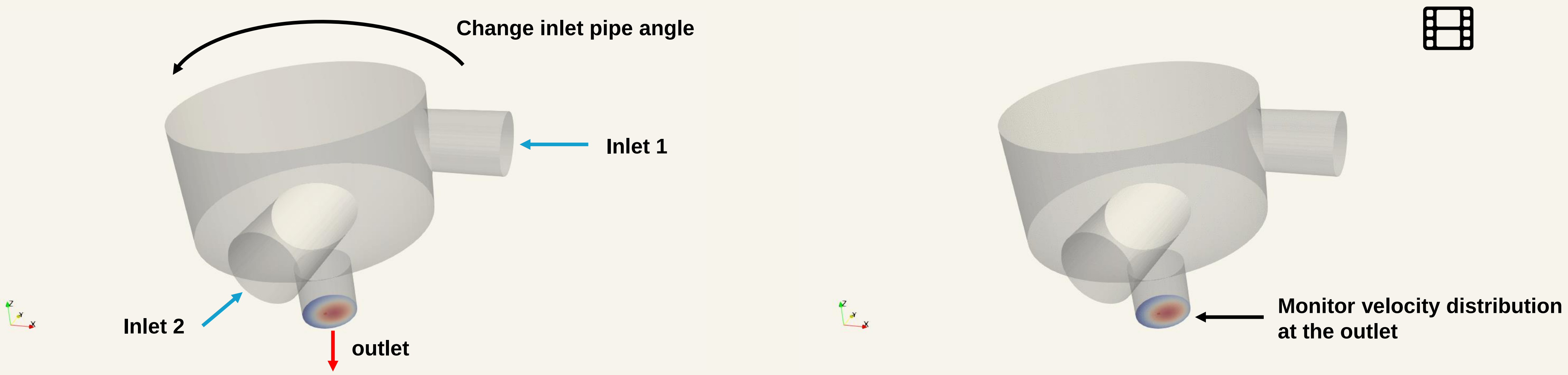




4. Live demonstration

Live demonstration

- Let us see the design loop and the cloud-based CAD tool (via the API) in action.
- We will change the angle of the inlet pipe, as shown in the figure.

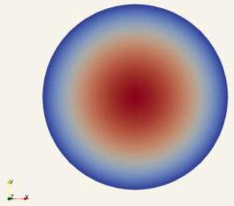


- Besides monitoring mass flow, average pressure, and maximum velocity in the inlet and outlet patches, we will also monitor the velocity distribution at the outlet (using coprocessing). Remember, in design loops is highly advisable to monitors many quantities.
- We will use the SSIM method (structural similarity) to compare the actual velocity distribution at the outlet with a target velocity distribution (qualitative comparison).
- The twist in this case is using a qualitative measure to assess the results (usually we use a quantitative measure).

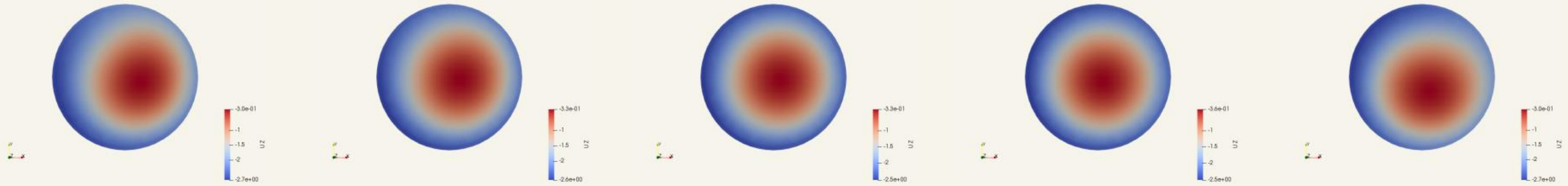
Live demonstration

- Qualitative comparison of velocity distribution at the outlet.
- The SSIM method was used to compare the images.
- In the SSIM, a value of 1 means that the images are identical.

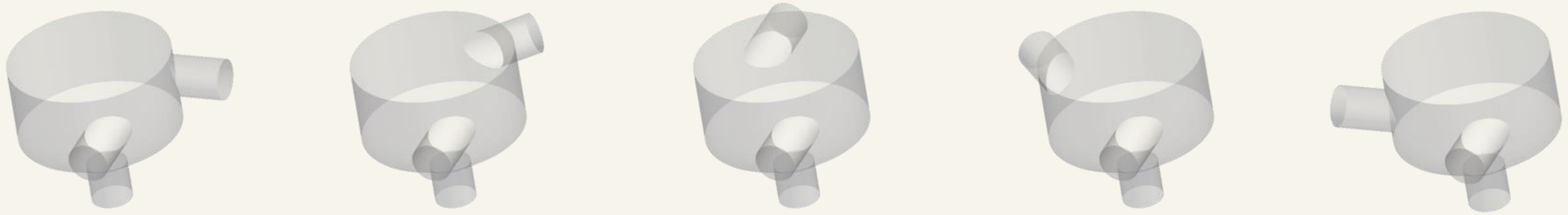
Target velocity distribution at outlet →



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

5. **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.
- We also opened a new dimension, namely, everyone working together on the same document, real-time, on any-device, anywhere.
- Implementing an engineering design loop is a meticulous and thoughtful process that requires careful planning.
- Validate and calibrate your design loop, be sure that is fault tolerant, accurate, and robust.
- Always monitor and analyze your data (quantitative or qualitative) real-time.
- Open-source technology is mature enough to be used with industrial applications.
- And the cloud is here to stay.

We take very seriously this vision – It is always in our radar

We aim at addressing the findings and recommendations reported in the NASA contractor report "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences", where the authors state (among many things):

> "A single engineer/scientist must be able to conceive, create, analyze, and interpret a large ensemble of related simulations in a time-critical period (e.g., 24 hours), without individually managing each simulation, to a pre-specified level of accuracy."

FYI – Useful links

- All solid models used in this presentation are shared publicly in Onshape.
- Just look for the following documents:
    - midwest_OF_ahmed_bodies
    - midwest_OF_wing-winglet
    - midwest_OF_static_mixer_opt_configuration
    - midwest_OF_static_mixer
    - midwest_OF_multi_element_airfoil

- You can download the working cases from GitHub:
    - https://github.com/wolfDynamics/midwest_OF_2019

- Remember to add your Onshape credentials in the file `creds.json` located in the directory `support_files` of each case.
- To generate the API credentials go to https://dev-portal.onshape.com/keys

Thank you very much for your attention

Be collaborative, be innovative, be cloud

wolf dynamics

multiphysics simulations,
optimization & data analytics

www.wolfdynamics.com

guerrero@wolfdynamics.com

www.wolfdynamics.com