# Abstract

This study focuses on validating the Computational Fluid Dynamics (CFD) component of an Aerodynamic Shape Optimization (ASO) framework using experimental data obtained from Wind Tunnel Tests (WTT). The objective is to assess the accuracy and reliability of the CFD-based predictions in replicating real-world aerodynamic performance. The comparison serves as a crucial step toward establishing confidence in the ASO framework for future optimization studies.

# Purpose of ASO

The purpose of Aerodynamic Shape Optimization (ASO) is to systematically improve the aerodynamic performance of a body—such as an aircraft wing, fuselage, or UAV component—by optimizing its geometry to achieve specific objectives like minimizing drag, maximizing lift-to-drag ratio, or enhancing stability. ASO enables engineers to explore complex design spaces that are often beyond human intuition, leading to more efficient and high-performance configurations. By integrating Computational Fluid Dynamics (CFD) solvers with optimization algorithms, ASO automates the iterative design process, reducing development time and improving design accuracy compared to traditional trial-and-error approaches.

Aerodynamic Shape Optimization (ASO) plays a vital role in modern aerospace design by enabling the development of efficient and high-performance configurations through the integration of Computational Fluid Dynamics (CFD) and optimization algorithms. With advances in high-fidelity solvers and computational power, CFD-based ASO frameworks have become powerful tools for exploring complex design spaces and improving aerodynamic efficiency. However, ensuring the accuracy of CFD predictions is essential before employing such frameworks for optimization. Therefore, this work focuses on validating the CFD component of the ASO framework by comparing its results against experimental data from Wind Tunnel Tests (WTT). This validation establishes the credibility of the numerical approach and strengthens its applicability to real-world aerodynamic design problems.

DAFoam is an open-source adjoint-based optimization framework that integrates tightly with OpenFOAM to enable gradient-based aerodynamic shape optimization. It allows for efficient computation of design sensitivities through discrete adjoint methods, making it suitable for high-fidelity and large-scale aerodynamic applications. In this study, the DAFoam framework is employed to perform CFD-based simulations as part of the validation process. The results are compared against Wind Tunnel Test (WTT) data to evaluate the accuracy of DAFoam’s flow predictions and its capability to capture key aerodynamic characteristics. This validation serves as a foundational step toward utilizing DAFoam for full aerodynamic shape optimization studies.

# ICEM

ANSYS ICEM CFD is utilized for generating high-quality meshes required for the CFD simulations within the DAFoam framework. It provides advanced tools for structured, unstructured, and hybrid meshing, allowing precise control over mesh density near critical regions such as leading and trailing edges, boundary layers, and wake zones. In this study, ICEM is used to create a mesh that ensures adequate resolution for capturing flow features accurately while maintaining computational efficiency. Proper mesh quality, including parameters like orthogonality, aspect ratio, and y⁺ values, is ensured to achieve reliable CFD results for validation against Wind Tunnel Test (WTT) data.

# FFD

In the DAFoam framework, geometry parameterization is achieved using Free-Form Deformation (FFD) boxes. The FFD method enables smooth and continuous shape modifications by embedding the geometry within a lattice of control points, which can be moved to deform the surface in a controlled manner. This approach allows for flexible and efficient representation of complex aerodynamic shapes while maintaining geometric smoothness and manufacturability. In this study, the FFD box is defined around the wing geometry to introduce design variables that influence the aerodynamic characteristics during the optimization or validation process.

# Meshing

The meshing process plays a critical role in ensuring the accuracy and stability of CFD simulations within the DAFoam framework. A high-quality mesh is essential to accurately resolve flow features such as boundary layer development, separation, and wake formation. In this study, the computational domain is discretized using ANSYS ICEM CFD, which allows precise control over mesh refinement near the wing surface and other regions of interest. Special attention is given to achieving appropriate near-wall resolution, ensuring that the y⁺ values fall within the desired range for the selected turbulence model. Mesh quality metrics such as skewness, orthogonality, and smoothness are also monitored to maintain numerical accuracy and convergence stability during the simulations.

The outer computational domain defines the far-field boundaries that simulate free-stream conditions around the aerodynamic body. In this study, the outer domain is designed to be sufficiently large to minimize the influence of boundary effects on the flow around the wing. Typically, the far-field boundaries are placed several chord lengths away from the geometry—upstream, downstream, and in all lateral directions—to ensure accurate pressure and velocity distributions. Appropriate boundary conditions are applied to represent the free-stream flow, such as velocity inlet, pressure outlet, and symmetry planes where applicable. This careful setup of the outer domain helps achieve realistic flow behavior and reliable comparison with Wind Tunnel Test (WTT) results.

# Geometry

The baseline geometry used in this study was obtained from an existing aerodynamic model. The geometry was first imported into ANSYS DesignModeler, where a symmetry plane was applied to reduce computational cost and ensure uniform flow characteristics across the span. Subsequent geometry cleanup and surface repair were carried out in ANSYS ICEM CFD to remove gaps, overlaps, and unnecessary features that could affect mesh generation or simulation stability. This preprocessing ensured that the final geometry was watertight, smooth, and suitable for high-quality meshing and accurate CFD analysis within the DAFoam framework.

# CFD results

The aim of the CFD results section is to present and analyze the aerodynamic performance predicted by the DAFoam framework and compare it against Wind Tunnel Test (WTT) data for validation. This section focuses on evaluating key aerodynamic parameters such as lift coefficient (CL), drag coefficient (CD), and pressure distribution over the wing surface. The objective is to assess the accuracy of the CFD simulations, identify any discrepancies between numerical and experimental results, and understand the underlying flow behavior. This comparison establishes confidence in the DAFoam-based CFD setup as a reliable tool for future aerodynamic shape optimization studies.

**Cruise Regime (Transonic Flow):**  
In the cruise regime, simulations were conducted at a transonic Mach number representative of typical aircraft operating conditions. The flow field in this regime is characterized by the presence of local shock waves and boundary layer interactions, which significantly influence drag and lift performance. The DAFoam-predicted results were compared with Wind Tunnel Test (WTT) data, showing good agreement in terms of pressure distribution and aerodynamic coefficients. Minor deviations were observed near the shock regions, likely due to mesh resolution and turbulence model limitations, but overall the CFD results accurately captured the key aerodynamic behavior at cruise conditions.

**Subsonic Regime:**  
For the subsonic regime, the flow remains fully attached with smooth pressure gradients and negligible compressibility effects. The DAFoam simulations demonstrated strong correlation with the WTT measurements, especially in the prediction of lift coefficient (CL) and surface pressure distribution. The results indicate that the solver effectively models low-speed aerodynamic behavior with high accuracy. The validation under subsonic conditions confirms that the CFD setup and meshing strategy are robust for low-Mach-number applications.

**Supersonic Regime:**  
At supersonic Mach numbers, the flow is dominated by strong shock waves, expansion fans, and potential flow separation near the trailing edge. The DAFoam-based CFD simulations successfully captured these high-speed flow phenomena, and the computed aerodynamic coefficients were in reasonable agreement with WTT data. Slight discrepancies in drag prediction were attributed to shock-induced boundary layer effects and experimental uncertainties. Overall, the results validate DAFoam’s capability to simulate complex compressible flows and demonstrate its reliability across a wide range of Mach regimes.

**Transonic Regime:**  
In the transonic regime, the flow exhibits mixed characteristics of both subsonic and supersonic behavior, with local supersonic pockets forming over the wing surface followed by shock-induced flow separation. This regime is particularly sensitive to mesh quality and turbulence modeling, making it a critical test for CFD validation. The DAFoam simulations captured the onset and position of shock waves with reasonable accuracy when compared to Wind Tunnel Test (WTT) data. The pressure coefficient distributions and aerodynamic coefficients showed good agreement, confirming that the solver accurately resolves the complex flow interactions typical of transonic flight. Minor discrepancies near the shock region were observed, likely due to the limitations of steady-state modeling, but the overall trends validated the robustness of the DAFoam framework in transonic conditions.

# Turbulence models

DAFoam provides support for several turbulence models, enabling users to simulate a wide range of aerodynamic flow conditions with varying levels of fidelity and computational cost. Among the commonly used models are the **Spalart–Allmaras (SA)** model and the **k–ω SST (Shear Stress Transport)** model.

The **Spalart–Allmaras (SA)** model is a one-equation turbulence model designed primarily for external aerodynamic flows. It offers a good balance between accuracy and computational efficiency, making it well-suited for attached flows and preliminary optimization studies. The model is robust, easy to converge, and performs effectively in predicting boundary layer behavior for streamlined configurations such as wings and airfoils.

The **k–ω SST** model, on the other hand, combines the advantages of the k–ω model near walls and the k–ε model in the free stream. It provides improved accuracy in capturing flow separation, adverse pressure gradients, and shock–boundary layer interactions—features critical for transonic and supersonic simulations. In DAFoam, this model is often preferred for high-fidelity aerodynamic analyses where capturing complex flow phenomena is essential. Together, these turbulence models offer flexibility to balance accuracy and computational efficiency depending on the flow regime and design objectives.

The comparison between the Spalart–Allmaras (SA) and k–ω SST turbulence models revealed consistent aerodynamic predictions at low angles of attack (AOA), where the flow remains mostly attached and steady. In this regime, both models produced nearly identical results for lift, drag, and pressure distribution, indicating that either model is suitable for low-AOA aerodynamic analysis. However, as the angle of attack increased, the k–ω SST model demonstrated superior performance in capturing flow separation and stall onset behavior. This is attributed to its ability to better resolve adverse pressure gradients and complex boundary layer interactions. The improved agreement of the k–ω SST results with Wind Tunnel Test (WTT) data at high AOA highlights its suitability for cases involving strong flow curvature, shock–boundary layer interaction, and separation, where the SA model tends to oversimplify the turbulence physics.

This study is necessary to establish the reliability and accuracy of the CFD component within the DAFoam-based Aerodynamic Shape Optimization (ASO) framework before it is employed for design optimization tasks. By validating CFD results against experimental Wind Tunnel Test (WTT) data across different flow regimes and turbulence models, the study ensures that the numerical setup can accurately predict aerodynamic performance under realistic conditions. Such validation is a crucial step to build confidence in the framework’s predictive capability and to identify the most suitable turbulence model for different optimization scenarios.

While the **k–ω SST** model provides higher accuracy—especially at high angles of attack where flow separation and adverse pressure gradients dominate—it also demands more computational resources. In contrast, the **Spalart–Allmaras (SA)** model offers a good compromise between accuracy and efficiency, particularly for attached and moderate-flow conditions where its assumptions remain valid. Therefore, for preliminary design studies or optimization cases focused on low to moderate angles of attack, the SA model can be effectively used due to its faster convergence and lower computational cost. However, for high-fidelity optimization involving separated or transonic flows, the k–ω SST model remains the preferred choice.

**Engine modeling**

**Aim:**  
The aim of this part of the study is to evaluate the impact of different boundary conditions—specifically, a wall boundary versus a freestream condition—at the engine exhaust on the overall aerodynamic predictions. The objective is to determine which setup provides results that are more consistent with Wind Tunnel Test (WTT) data and realistic flow behavior around the aircraft configuration.

**Necessity:**  
Selecting appropriate boundary conditions at the engine exhaust is crucial because it directly influences the local flow field, pressure distribution, and downstream aerodynamic forces. An incorrect boundary assumption, such as using a wall where the flow should exit freely, can lead to artificial pressure buildup and unrealistic flow separation. Therefore, analyzing both wall and freestream cases helps identify the configuration that best replicates real aerodynamic conditions and ensures the credibility of the CFD setup.

**Findings and Interpretation:**  
The results showed that applying a freestream boundary condition at the engine exhaust produced better agreement with WTT data compared to the wall condition. This indicates that allowing the flow to exit naturally from the exhaust region provides a more physically accurate representation of the aerodynamic environment. The wall condition, on the other hand, likely caused artificial flow blockage and pressure rise, distorting the aerodynamic coefficients. This finding highlights the importance of realistic boundary condition selection in CFD modeling, particularly when simulating complex configurations involving jet exhausts or open-flow regions.

# Mesh study

**Aim — y⁺ Sensitivity Study:**  
The aim of this study is to assess how near-wall resolution (two mesh families: y⁺ < 5 and y⁺ ≈ 300–400) affects the fidelity of CFD predictions in the DAFoam simulations. Specifically, we compare aerodynamic coefficients and pressure distributions from both meshes against Wind Tunnel Test (WTT) data to determine which near-wall treatment better captures boundary-layer behavior across different flow regimes and angles of attack.

**Why this is necessary:**  
Near-wall resolution controls whether the boundary layer is fully resolved (low y⁺) or modeled with wall functions (high y⁺). This choice strongly influences predictions of skin-friction, separation onset, and shear-layer behavior—key contributors to lift, drag, and pressure distribution. Testing both extremes verifies the robustness of the CFD setup and helps choose an appropriate mesh strategy for optimization trade-offs between accuracy and computational cost.

**Observations / Results:**  
The fine mesh with y⁺ < 5 produced improved agreement with WTT data in regions sensitive to boundary-layer gradients (e.g., pressure distribution near the leading edge and separation locations), especially at higher angles of attack and in transonic/supersonic regimes. The coarse mesh with y⁺ ≈ 300–400 gave similar results in attached, low-AOA flows but underpredicted separation effects and showed differences in drag and near-wall pressure compared to experiments.

**Interpretation:**  
The low-y⁺ mesh resolves the viscous sublayer and works well with models (like k–ω SST) that depend on near-wall quantities, yielding higher-fidelity results for separated or high-gradient flows. The high-y⁺ mesh relies on wall-function assumptions (suitable with SA in some setups), which can be adequate for attached flows but fail to capture separation and shock-boundary-layer interactions accurately. Thus, the best mesh depends on the flow regime and the required fidelity.

**Recommendation:**  
For preliminary optimization and many low-AOA, attached-flow studies—where computational budget is limited—the higher-y⁺ mesh (with appropriate wall functions and SA model) is an acceptable and efficient choice. For validation, transonic/supersonic cases, or high-AOA scenarios where separation and shocks are important, use the low-y⁺ mesh combined with a higher-fidelity turbulence model (k–ω SST) to ensure accurate predictions. If possible, perform a local refinement study (targeted low-y⁺ regions) to balance cost and accuracy.

Conclusion:  
The validation study successfully demonstrated the reliability of the DAFoam-based CFD framework for aerodynamic analysis and shape optimization applications. By comparing numerical results with Wind Tunnel Test (WTT) data across various Mach regimes—subsonic, transonic, and supersonic—the framework showed strong predictive capability for aerodynamic coefficients and flow features. The comparison of turbulence models revealed that while the Spalart–Allmaras (SA) model performs adequately at low angles of attack, the k–ω SST model provides superior accuracy at higher angles where flow separation and shock interactions dominate.

Boundary condition analysis further emphasized the importance of physical accuracy, with the freestream condition at the engine exhaust producing results that closely matched experimental trends, unlike the wall condition, which introduced artificial flow blockage. The mesh sensitivity study highlighted that fine near-wall resolution (y⁺ < 5) is essential for high-fidelity predictions, especially in separated or transonic flows, while coarser meshes (y⁺ ≈ 300–400) remain suitable for attached, low-AOA cases.

Overall, the study validates the CFD setup in DAFoam as a dependable and accurate foundation for future aerodynamic shape optimization work. The insights gained regarding turbulence model selection, boundary conditions, and mesh resolution will help guide the development of efficient and robust optimization frameworks tailored to specific aerodynamic regimes.

References

<https://mdolab-mach-aero.readthedocs-hosted.com/en/latest/machFramework/MACH-Aero.html>

<https://dafoam.github.io/index.html>

# DAFOAm

* **Ping He, Charles A. Mader, Joaquim R.R.A. Martins, Kevin J. Maki. DAFoam: An open-source adjoint framework for multidisciplinary design optimization with OpenFOAM. AIAA Journal, 58:1304-1319, 2020. https://doi.org/10.2514/1.J058853**
* **Ping He, Charles A. Mader, Joaquim R.R.A. Martins, Kevin J. Maki. An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM. Computer & Fluids, 168:285-303, 2018. https://doi.org/10.1016/j.compfluid.2018.04.012**

# Mach aero

PyGeo

H. Hajdik, A. Yildirim, E. Wu, B. Brelje, S. Seraj, M. Mangano, J. Anibal, E. Jonsson, E. Adler, C. A. Mader, G. Kenway, and J. R. R. A. Martins. pyGeo: A geometry package for multidisciplinary design optimization. Journal of Open Source Software, 8(87), 5319, July 2023. <https://doi.org/10.21105/joss.05319>

G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins, “A CAD-Free Approach to High-Fidelity Aerostructural Optimization”, in Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, 2010.

**PyHyp**

**N. Secco, G. K. W. Kenway, P. He, C. A. Mader, and J. R. R. A. Martins, “Efficient Mesh Generation and Deformation for Aerodynamic Shape Optimization”, AIAA Journal, 2021.**[**doi:10.2514/1.J059491**](https://doi.org/10.2514/1.J059491)

**Idwarp**

**N. Secco, G. K. W. Kenway, P. He, C. A. Mader, and J. R. R. A. Martins, “Efficient Mesh Generation and Deformation for Aerodynamic Shape Optimization”, AIAA Journal, 2021.**[**doi:10.2514/1.J059491**](https://doi.org/10.2514/1.J059491)

**Adflow**

1. Kenway, C. A. Mader, P. He, and J. R. R. A. Martins. Effective adjoint approaches for computational fluid dynamics. Progress in Aerospace Sciences, 110:100542, October 2019. [doi:10.1016/j.paerosci.2019.05.002](https://doi.org/10.1016/j.paerosci.2019.05.002)
2. **Mader, G. K. W. Kenway, A. Yildirim, and J. R. R. A. Martins. ADflow—an open-source compu- tational fluid dynamics solver for aerodynamic and multidisciplinary optimization. Journal of Aerospace Information Systems, 2020.**[**doi:10.2514/1.I010796**](https://doi.org/10.2514/1.I010796)
3. Yildirim, G. K. W. Kenway, C. A. Mader, and J. R. R. A. Martins. A Jacobian-free approximate Newton– Krylov startup strategy for RANS simulations. Journal of Computational Physics, 397:108741, Nov. 2019. ISSN 0021-9991. [doi:10.1016/j.jcp.2019.06.018](https://doi.org/10.1016/j.jcp.2019.06.018)

PyOptSparse

1. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. R. A. Martins. pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems. Journal of Open Source Software, 5(54), 2564, October 2020. <https://doi.org/10.21105/joss.02564>

ICEM

<https://ansyshelp.ansys.com/public/Views/Secured/corp/v251/en/pdf/Ansys_ICEM_CFD_Help_Manual.pdf>

<https://ansyshelp.ansys.com/public/Views/Secured/corp/v251/en/pdf/Ansys_ICEM_CFD_Programmers_Guide.pdf>

Opefoam

<https://wiki.openfoam.com/Main_Page>

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

<https://www.openfoam.com/documentation/user-guide/a-reference/a.1-standard-solvers>

# MACH-Aero Framework Manual

## Table of Contents

1. [Introduction](#_1._Introduction)
2. [Core Modules](#_2._Core_Modules)
   * [baseClasses](#_baseClasses)
   * [pySpline](#_pySpline)
   * [pyGeo](#_pyGeo)
   * [IDWarp](#_IDWarp)
   * [ADflow](#_ADflow)
   * [pyOptSparse](#_pyOptSparse)
3. [Optional Modules](#_3._Optional_Modules)
   * [pyHyp](#_pyHyp)
   * [multiPoint](#_multiPoint)
   * [cgnsUtilities](#_cgnsUtilities)
   * [DAFoam](#_DAFoam)
4. [Workflow Overview](#_4._Workflow_Overview)
5. [Installation and Setup](#_5._Installation_and)
6. [Acknowledgments](#_6._Acknowledgments)

## 1. Introduction

The MACH-Aero framework is a comprehensive and extensible suite of software tools for high-fidelity aerodynamic shape optimization. Built around adjoint-based optimization techniques, it integrates geometry manipulation, mesh deformation, flow simulation, and optimization in a scalable and modular fashion. The framework is suitable for both academic research and industrial design tasks involving computational fluid dynamics (CFD) and design optimization.

## 2. Core Modules

### baseClasses

This module provides a set of abstract base classes and infrastructure tools that are shared across nearly all MACH-Aero packages. It defines common utilities and ensures a consistent development style across modules.

**Key functionalities:**

- **File I/O Management**: Interfaces for reading and writing standard file formats.

- **Timing and Profiling**: Functions to benchmark code performance.

- **Parallel Communication Tools**: Interfaces with mpi4py for distributed computing.

- **Error Handling and Logging**: Centralized logging system and error reporting.

- **Command-Line Argument Parsing**: Provides standardized methods for parsing user inputs.

Use Case: Facilitates interoperability and code consistency across the MACH ecosystem.

### pySpline

pySpline implements B-spline functions for geometric curve and surface representation.

**Key functionalities:**

- **Curve and Surface Definitions**: Construct and evaluate B-spline curves and tensor-product B-spline surfaces.

- **Point Interpolation**: Generate splines that pass through or approximate a set of user-defined points.

- **Geometry Manipulation**: Modify shapes while maintaining smoothness and continuity.

Use Case: Define fuselage centerlines, airfoil cross-sections, or control curves for FFD boxes.

### pyGeo

pyGeo is a geometry engine that defines and manipulates parameterized geometry for optimization.

**Key functionalities:**

- **Free-Form Deformation (FFD)**: Deforms the underlying geometry via control points.

- **Design Variable Linking**: Enables interdependent variable configurations.

- **Constraint Definition**: Maintain physical realism with constraints on volume, thickness, LE/TE control, etc.

- **Surface Projection**: Project FFD boxes onto CAD or mesh surfaces.

Use Case: Model wing shapes, fuselages, nacelles, etc., and introduce shape parameters for optimization.

### IDWarp

IDWarp is responsible for high-quality mesh deformation using inverse distance weighting.

**Key functionalities:**

- **Surface-to-Volume Mesh Propagation**: Ensures consistent deformation of CFD meshes after geometry updates.

- **Parallel Deformation**: Scalable deformation algorithm.

- **Robust Handling of Complex Geometries**: Works well for highly curved and multi-component geometries.

Use Case: Update the CFD mesh in response to geometry changes during shape optimization.

### ADflow

ADflow is a parallel CFD solver with discrete adjoint capabilities. It is designed for aerodynamic simulations and gradient evaluations.

**Key functionalities:**

- **Steady/Unsteady RANS and Euler Solvers**: Models various compressible/incompressible flow regimes.

- **Turbulence Models**: Includes Spalart-Allmaras, k-ω SST, and others.

- **Discrete Adjoint Solver**: Efficient gradient computation for optimization.

- **Multi-block Structured Grids**: Optimized for high-performance computing.

Use Case: Simulate aerodynamic performance of geometries and compute accurate gradients for use in optimization.

### pyOptSparse

pyOptSparse is a general-purpose optimization library that integrates with the MACH-Aero stack.

**Key functionalities:**

- **Interfaces to Solvers**: SNOPT, IPOPT, SLSQP, NSGA2.

- **Support for Sparse Jacobians**: Efficient optimization for large-scale problems.

- **Flexible API**: Define objectives, constraints, and design variables.

- **Parallel Evaluations**: Run constraints and objective evaluations in parallel.

Use Case: Perform gradient-based or gradient-free optimization of aerodynamic configurations.

## 3. Optional Modules

### pyHyp

pyHyp generates structured volume meshes using a hyperbolic extrusion technique.

**Key functionalities:**

- **Boundary Layer Control**: Fine control over first cell height and stretching.

- **Grid Orthogonality**: Maintains high mesh quality near boundaries.

- **Surface Conformity**: Preserves geometric features while growing mesh volume.

Use Case: Generate high-quality meshes around airfoils or wing-body configurations.

### multiPoint

This module manages multi-point and multi-condition optimization.

**Key functionalities:**

- **Case Aggregation**: Combine results from various flow conditions.

- **Scenario Weighting**: Weight different conditions in the objective.

- **Design Variable Consistency**: Ensure shared design variables across cases.

Use Case: Optimize across cruise, takeoff, and landing conditions for robust designs.

### cgnsUtilities

A set of command-line tools for interacting with CGNS-formatted meshes.

**Key functionalities:**

- **Mesh Inspection**: Visualize zones, elements, and boundaries.

- **Mesh Manipulation**: Translate, rotate, or mirror mesh blocks.

- **Format Conversion**: Convert between CGNS and other formats.

Use Case: Pre-process and validate meshes used in ADflow or other CFD solvers.

### DAFoam

DAFoam is an OpenFOAM-based optimization framework with adjoint support.

**Key functionalities:**

- **Built on OpenFOAM**: Leverages popular open-source CFD platform.

- **Adjoint Solver**: Supports efficient derivative evaluations.

- **Custom Solver Integration**: Plug into existing OpenFOAM solvers.

Use Case: Perform shape optimization when OpenFOAM-specific solvers or models are needed.

## 4. Workflow Overview

The general workflow of the MACH-Aero framework involves:

1. **Geometry Parameterization**: Define geometry and design variables using pyGeo.
2. **Mesh Deformation**: Update CFD mesh with IDWarp to match new geometries.
3. **CFD Simulation**: Use ADflow or DAFoam to evaluate aerodynamic performance.
4. **Adjoint Evaluation**: Compute gradients using adjoint solvers.
5. **Optimization**: Run the optimization loop with pyOptSparse.

Each step is modular and extensible, allowing customization for specific applications.

## 5. Installation and Setup

Refer to the [official documentation](https://mdolab-mach-aero.readthedocs-hosted.com/en/latest) for installation instructions, dependencies, and compatibility notes. Each module may have specific requirements (e.g., compilers, MPI, PETSc, Python version).

## 6. Acknowledgments

MACH-Aero is developed and maintained by the Multidisciplinary Design Optimization Laboratory (MDOLab). For detailed guidance, bug reports, and contributions, please consult the official documentation and GitHub repositories associated with each module.