Anvil: An integration of artificial intelligence, sampling techniques, and a combined CAD-CFD tool

Harsh Vardhan*, Umesh Timalsina*, Michael Sandborn*, David Hyde*, Peter Volgyesi*, Janos Sztipanovits*

* Vanderbilt University, Nashville, TN, USA

Email: {harsh.vardhan, umesh.timalsina, michael.sandborn, david.hyde.1, peter.volgyesi, janos.sztipanovits}@vanderbilt.edu

Abstract—In this work, we introduce an open-source integrated CAD-CFD tool, *Anvil*, which combines FreeCAD for CAD modeling and OpenFOAM for CFD analysis, along with an AI-based optimization method (Bayesian optimization) and other sampling algorithms. *Anvil* serves as a scientific machine learning tool for shape optimization in three modes: data generation, CFD evaluation, and shape optimization. In data generation mode, it automatically runs CFD evaluations and generates data for training a surrogate model. In optimization mode, it searches for the optimal design under given requirements and optimization metrics. In CFD mode, a single CAD file can be evaluated with a single OpenFOAM run. To use *Anvil*, experimenters provide a JSON configuration file and a parametric CAD seed design. *Anvil* can be used to study solid-fluid dynamics for any subsonic flow conditions and has been demonstrated in various simulation and optimization use cases. The open-source code for the tool, installation process, artifacts (such as CAD seed designs and example STL models), experimentation results, and detailed documentation can be found at https://github.com/symbench/Anvil.

Index Terms—Computer-Aided Design (CAD), Computational Fluid Dynamics (CFD), Bayesian Optimization (BO), Surrogate Modeling, Design Optimization.

I. INTRODUCTION

Design optimization, which encompasses shape optimization, is a widely explored and essential task in computational physics [1]–[6]. Shape optimization involves modifying the geometry of a given topological architecture to achieve specific objectives by refining and improving the shape of an object. Modifications made to a shape under study might entail minimizing stress concentrations, maximizing strength tolerance on specified axes, or optimizing fluid flow characteristics. Our focus is on shape optimization design problems in engineering that require CAD modeling and fluid simulation to analyze a design.

On the optimization front, recent advancements in artificial intelligence (AI) have opened up exciting possibilities for these algorithms' use as sample-efficient optimizers [7]. The fusion of traditional CAD-CFD toolchains with AI has the potential to revolutionize the way engineering simulations and design processes are conducted.

In the context of this type of optimization problem, we note two main challenges:

1) **Lack of open-source, easy-to-use automated CAD-CFD evaluation tools**: The shape optimization process is hindered by the manual and disjointed use of CAD for design modifications and CFD for fluid dynamics analysis, necessitating manual file handling (e.g., creating and passing STL files between two programs). Seamless integration is essential for automating the design evaluation process, reducing manual intervention. Despite the existence of integrated CAD-CFD commercial solutions, their high license fees often make them an infeasible option for researchers.

2) **Time-consuming and computationally expensive evaluation process**: The evaluation process through CAD-CFD pipelines is notably time-consuming and computationally intensive, with simulations extending from minutes to days. This inefficiency is exacerbated in exhaustive search methods, particularly in complex design spaces. AI-based optimization methods offer a solution with their sample efficiency and superior convergence properties, underscoring the need for integrating these methods with CAD-CFD tools to enhance practicality and performance.

To address these challenges, we present an open-source, highly automated, and algorithmically efficient software tool, *Anvil*, for CAD-CFD design optimization problems. Our tool integrates existing open-source CAD and CFD tools and wraps this integrated functionality within a Python-based sampling and optimization framework.

The remainder of the manuscript is structured as follows. After a discussion of related work in Section II, Sections III presents the software architecture of *Anvil*, including the standard for creating parametric CAD designs, numerical settings, mathematical models used for CFD simulation, and implementation details of the tool. In Section IV, we show experiments conducted on various designs. No one tool is complete; accordingly, in Section V, we elucidate some of *Anvil*'s limitations. Concluding remarks and possible future directions are provided in Section VI.

II. RELATED WORK

The development of integrated CAD-CFD tools for shape optimization has been a focus in commercial tools like PTC Creo and AutoDesk Fusion360. However, the open-source

equivalent of this capability is not readily available. Previous works have utilized Bayesian optimization in loop with computational fluid dynamics (CFD) for design optimization [7]–[10], but these works did not make their entire toolchains available. Moreover, these solutions generally require manual intervention during optimization process , which is a bottle neck. The design evaluation tool provided in [7] is only demonstrated for Myring hull design optimization in the context of unmanned underwater vehicles (UUVs). On the learning front, most traditional works explored the use of genetic algorithms combined with CFD simulations to optimize shapes [11]. Although effective in identifying optimal designs, genetic algorithms can be computationally expensive and may not always converge to the global optimum [7]. In recent times, other learning models like random forest [12], deep learning [13], [14], and Bayesian optimization [15], [16] are gaining more attention due to their performance drastically exceeding classical approaches. The application of Bayesian optimization for shape optimization in fluid dynamics has been less common, with recent studies such as [7], [17] demonstrating its potential in optimizing complex geometries with a limited number of simulations [18]. However, these approaches have not been fully integrated into a user-friendly tool that combines CAD modeling and CFD analysis.

III. INTEGRATION ARCHITECTURE

In this section, we provide a high-level overview of *Anvil*. In computer-aided engineering (CAE), for shape optimization problems, computer-aided design (CAD) and computational fluid dynamics (CFD) are two essential technologies. CAD allows designers to create and modify models of solid objects, while CFD enables the analysis and simulation of fluid flow and related phenomena. The primary goal of CAD-CFD software architecture is to bridge the gap between the geometric design of objects and understanding objects' interactions with fluids.

At its core, *Anvil* comprises a unified software platform that integrates CAD and CFD tools, facilitating automated and accurate analysis of fluid behavior within complex geometries while also allowing flexibility to create, modify, and analyze 3D models of their designs. We then build a Python toolchain around these integrated software components, where the key advantage is the ability to automatically generate design variations, conduct CFD evaluation, and perform parametric optimizations. Accordingly, our architecture empowers users to incorporate optimization methods or automatic sampling methods to optimize their designs or generate data.

Figure 1 shows the overall software architecture and workflow of *Anvil*. The significant components shown in the diagram are detailed in the following subsections.

A. User inputs

In *Anvil*, the user needs to provide two things: first, a configuration JSON file and second, a parametric CAD seed design. The configuration JSON file contains various parameters, options, or preferences that the user can customize to

control how the tool should operate or process the data. By using a parametric CAD seed design, *Anvil* can leverage its parametric capabilities to adjust or adapt the design based on the parameters defined in the CAD model.

B. Parametric CAD seed design

A parametric CAD model provides the flexibility to change the design of a 3D CAD model based on one or more parameters. The parameters define the shape of the design; this may be viewed as a reduced-order model where the model is controlled by this typically small set of parameters. We use FreeCAD [19] as our our CAD design tool. To follow the rigorous design practices, we suggest for fully constraining the sketches of CAD design. To standardize the interaction between the CAD tool (FreeCAD) and Python scripts, we use the `Spreadsheet` functionality of FreeCAD. A spreadsheet must be attached to the parametric seed design with the name 'Spreadsheet' that consists of all the possible parameters that can be varied during optimization. The experimenter must follow specific rules while designing this parametric CAD design so that *Anvil* can access these parameters and automatically modify the design during experimentation. This rule of writing a spreadsheet is such that all the parameters' variable names with the default values must be provided. These cells in the spreadsheet must be linked with the sketch parameters in the design. By doing this, we do not need to understand the sketch properties and how these sketches interact, and only by interacting with the spreadsheet wrapper, we standardize the interface between Python-based scripts with different FreeCAD designs.

C. STL generation

Anvil uses the STL file format to exchange the shape generated by a parametric model in FreeCAD to our chosen CFD software. STL (Standard Tessellation Language) is a standard file format used and supported by many other software packages and is widely used for rapid prototyping, 3D printing, and computer-aided manufacturing. Although the actual standard of the STL file format is dimensionless, FreeCAD assumes that the length unit used in STL files is millimeters (mm). Consequently, all units and parameters in *Anvil* are taken to be consistent with using millimeters as the unit of length.

D. Computational Fluid Dynamics

In our problems of interest, it is necessary to evaluate the behavior of a specific solid design immersed in a turbulent fluid. We assume the fluid to be Newtonian, incompressible, and isothermal. Rather than more computationally expensive direct numerical simulation of the governing equations, we adopt a commonly-employed approach of using Reynolds-averaged Navier-Stokes (RANS) paired with a turbulence model.We employ RANS with the k-ω shear stress transport (SST) model for turbulence physics [20]. The k-ω SST model is based on the Boussinesq hypothesis that assumes a relationship between the Reynolds stress tensor and the mean rate-of-strain (deformation) tensor and the turbulent kinetic energy
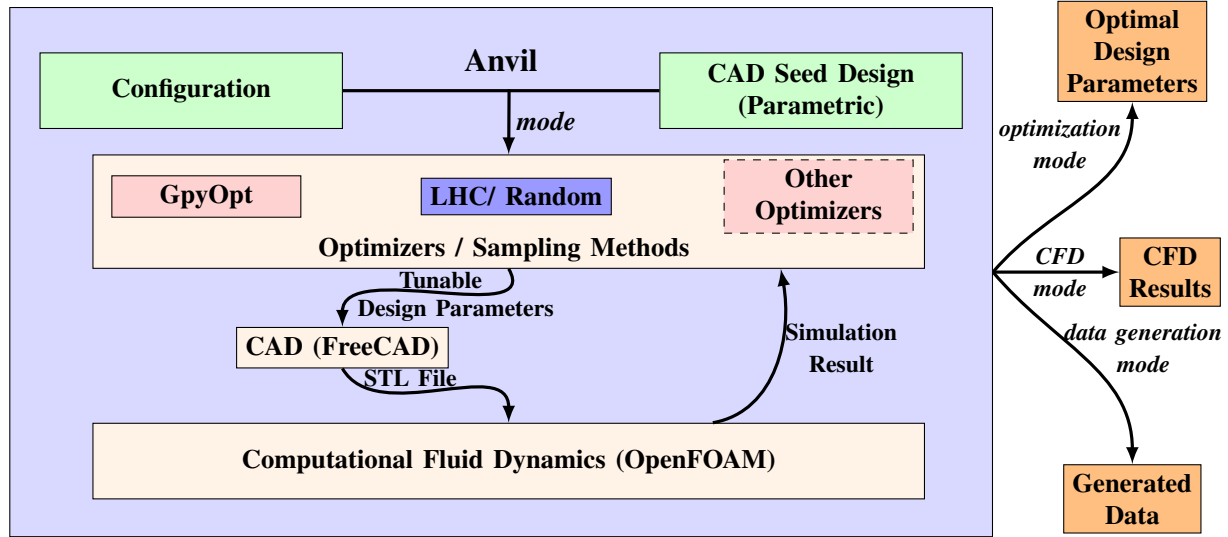
17

Fig. 1: Anvil Architecture Diagram. With a configuration JSON file and a parametric CAD seed design, designers can operate Anvil in three modes: (i.) Data Generation, where the CFD results by sampling provided design space can be saved in batches (ii.) CFD, for running CFD on a single parametric design (iii.) Optimization, for searching for optimum design parameters provided a design space and CFD drag, with Bayesian Optimization

k. To maximize accuracy, the k-ω SST model utilizes the k-ω model [21] near boundaries and the k-ϵ model [22] in the free stream away from the boundary layer. The k-ω SST model has been successfully validated in various real-world turbulent flow problems [20], [23], [24]. *Anvil* uses OpenFOAM [25] as our fluid simulation tool, which provides an implementation of RANS with k-ω SST out of the box.

E. Mesh Generation and auto-meshing

For volumetric meshing of the fluid portion of the computational domain, we leverage OpenFOAM's built-in meshing tools: blockMesh and snappyHexMesh. Meshing is a two steps process: the first step starts with the creation of a castellated 3D parametric volumetric mesh to fill the 3D volume using the blockMesh utility. This volume meshing process uses only hexahedral elements whose number is defined by the parameter in a configuration file. The volumetric mesh is further split and refined in the vicinity of the body surfaces. Once the mesh in the locality of the body is refined and split, cells inside the body shape are removed, keeping the constraint that at least one volumetric region must be bounded inside the domain. This results in separating the body shape from the volume mesh and creating a volume mesh around the surface. In the next step, refinement of the mesh near the surface layer is performed to generate three-dimensional unstructured or hybrid meshes consisting of hexahedra (hex) and split-hexahedra (split-hex) elements. Performed with the snappyHexMesh tool, this step removes cells that violate the mesh quality parameters near body surfaces.

For the meshing process, we need to provide the mesh size (level of refinement). The selection of mesh size depends on the size and the shape of the design. Since there is no analytical closed form solution to find the right mesh size,

accordingly if an inappropriate mesh size is provided, the simulation fails. To address this, we employ an *auto-meshing* capability, where the mesh size is selected automatically by capturing log errors from the meshing tool and by refining the mesh size accordingly to find the suitable mesh size.

F. Boundary and Initial Conditions

For all simulations, the appropriate boundary conditions used are the standard Dirichlet boundary conditions. On the left-hand side of the simulation domain, a velocity inlet boundary condition is imposed, while a zero pressure boundary condition is applied on the right-hand side. The hull surface is subjected to a no-slip boundary condition. The remaining surfaces, including the side wall, symmetry wall, and far-field, are set to have symmetry boundary conditions. The initial values for the variables k and ω are determined using the same procedure described in equations [7]. These initial values are calculated based on the assumption of the Newtonian model of the fluid, which considers the kinematic viscosity to be constant. The user has the option to provide the value of kinematic viscosity based on empirical data in the configuration file. The initial velocity inlet and values of k and ω are also entered in the configuration JSON file.

IV. COMPUTATIONAL EXPERIMENTS

We classify our experiments into three categories based on the different modes in which *Anvil* can be utilized.

A. CFD simulation experiments

We first demonstrate how our tool simplifies computational fluid dynamics (CFD) analysis, requiring minimal effort from the user. To conduct a CFD analysis, the designer simply

18

needs to provide an STL file of the design and a configuration file to specify turbulence models, fluid parameters, and other settings. An example configuration file is available in the anvil repository[1]. We chose to experiment with three different problems: unmanned underwater vehicles (UUVs), land vehicles, and unmanned aerial vehicles (UAVs). UUVs encompass both autonomous and remotely operated variants. In the design process of UUVs, CFD analysis is crucial for optimizing the hull shape, which significantly influences the vehicle's drag and, consequently, its range and efficiency. For land vehicles, CFD simulation is a valuable tool for analyzing and optimizing the aerodynamic performance. It is employed for various purposes, such as reducing drag, optimizing cooling systems, and studying crosswind stability. In the design of UAVs, CFD is deployed for early design studies and optimization that allow analyzing the aerodynamic performance of UAVs under various flight conditions without the need for physical prototypes or wind tunnel testing. When a CFD simulation reaches steady state, we obtain flow fields on the volumetric mesh that give the user a sense of how fluid flows around a candidate design (shown in Figures 3c, 3b, 4c, 3b, 4c, and 4b). Other physical metrics of interest like drag and lift can be derived using the flow field. Below we present the configuration parameters and CFD flow field results conducted using *Anvil*:

1) Unmanned Underwater Vehicle (UUV): For this experiment, a UUV design is taken from current research efforts [10], [26]. The UUV is simulated at a speed of 2.25 miles/hr, with fluid properties set to open sea density ($1027.0 Kg/m^3$), dynamic viscosity ($1.789 10^{-5}$ N−s/m^2), and turbulence intensity of 4%, indicating medium turbulence (refer to Figure 2).

2) Land vehicle: In this experiment, we designed a representative land vehicle and conducted a simulation with the following configuration: the speed of the vehicle was set to be 70 miles/hr, the fluid density is air density on surface i.e. $1.225 Kg/m^3$, dynamic viscosity as $1.789 \times 10^{-5} N − s/m^2$, and turbulence intensity is at 1% (refer to Figure 3).

3) Unmanned Air Vehicle (UAV): Our test UAV design features four propellers positioned at the front, primarily responsible for generating thrust. Additionally, it includes three wings: one dedicated to lift generation, and the other two tasked with controlling roll and yaw. The aircraft is also equipped with a cargo box for storage and a separate compartment housing electronics, batteries, and other essential components. The design is simulated at a speed of 50 meters/sec with fluid density is air density $1.225 Kg/m^3$, dynamic viscosity $1.789 \times 10^{-5} N − s/m^2$, turbulence intensity 1% (refer to Figure 4).

B. Optimization experiment

We also demonstrate the capability of *Anvil* to operate in a design optimization mode. As a test of this mode, we chose a design optimization problem of an underwater vehicle hull.

[1]https://github.com/symbench/Anvil/blob/main/artifacts/sample_input_config_winged_design.json
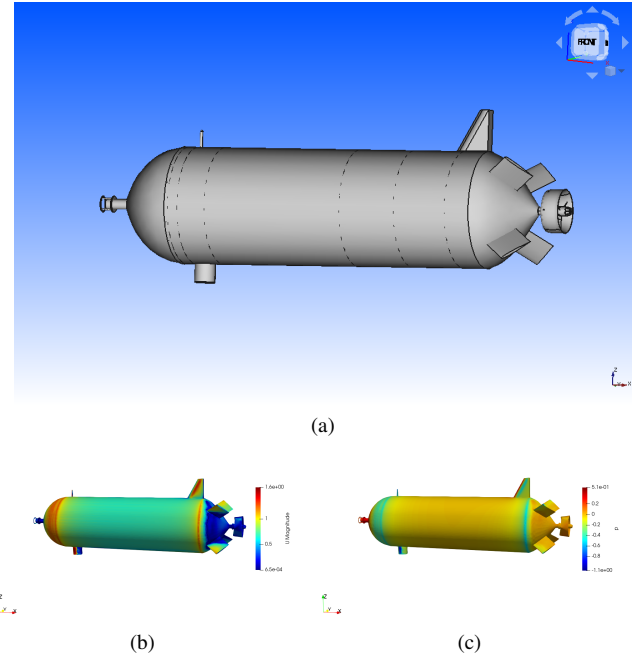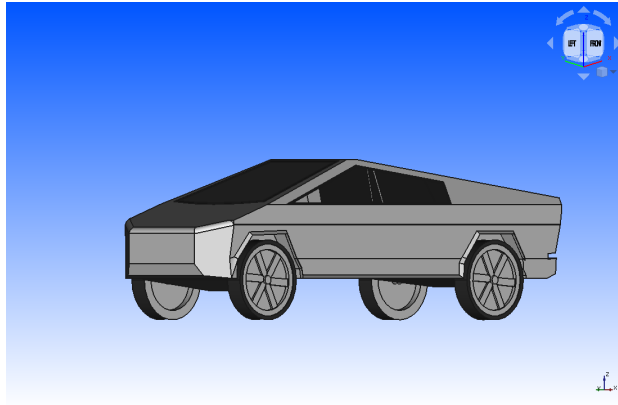

(a)

(b)   (c)

Fig. 2: (a) The UUV design. Steady state flow at the surface layer of design, with the flow field colored by (b) Velocity field (meters/second), (c) Pressure field (Pascals).

The problem of finding the optimal hull and its design space is taken from [27]. It considers the 1 meter length UUV design with 6 control points to morph the shape of the design (refer to Figure 5). The design space is in 7 dimensions: six control points which are allowed to range between 0 and 0.2 meters, and a seventh dimension, the UUV nose length with a search range between 10 cm - 900 cm. The tail section is a derived parameter and is defined as 1000 cm minus the nose length. The optimization goal is to find a hull design with minimal drag. For this we employ sequential Bayesian optimization using with a budget of 50 iterations. The fluid parameters are set according to 1% turbulence intensity and inlet velocity of 2.5 meter/sec. The details of the Bayesian optimization approach can be found in [7], [15]. The hyperparameters for Bayesian optimization and optimization goal and budgets are set in an *Anvil* configuration file (we chose Lower Confidence Bound (LCB) as our acquisition function). The mixture of radial basis function (RBF) kernel with small white noise is used as the learning kernel for multivariate Gaussian Process regression. The optimal design found after exhaustion of budget is shown in Figure 6.

C. Data generation experiment

Anvil can also be used to generate designs sampled from a design space. For our test problem, we made a parametric design that represents a winged air vehicle. In the design of this aerospace vehicle body, there are three main components: a hemispherical nose, a conical tail, and a cylindrical body. The wings of the body are another crucial component, and for symbolic representation, we deploy rectangular wings.
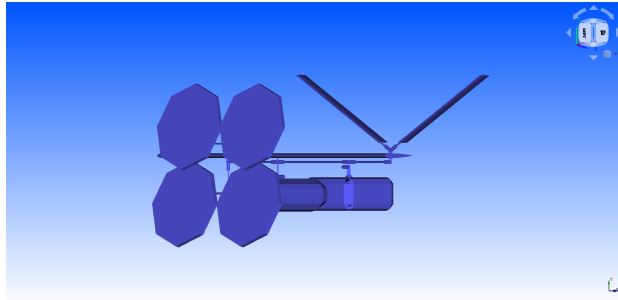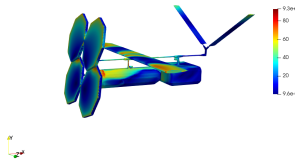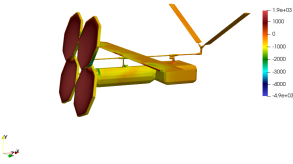
(a)
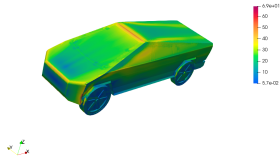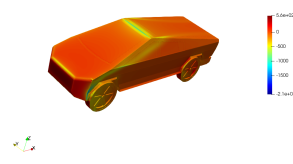
(b)   (c)

Fig. 3: (a) The land vehicle design. Steady state flow at the surface layer of design, with the flow field colored by (b) Velocity field (meters/second), (c) Pressure field (Pascals).

(a)

(b)   (c)

Fig. 4: (a) The UAV design. Steady state flow at the surface layer of design, with the flow field colored by (b) Velocity field (meters/second), (c) Pressure field (Pascals).

Various parameters are considered that can alter the vehicle's shape, including the length of the nose section (denoted as $nose\_radius$), the length of the body section (or $fuselage\_length$), the tail section's length (referred to as $tail\_length$), the thickness of the wing ($thickness\_wing$), the span of one wing ($half\_span$), and the chord length of the wing. The range of the design space for these parameters
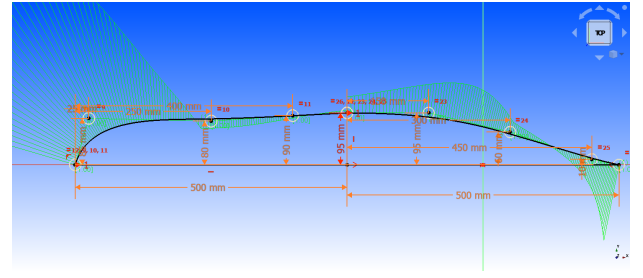

Fig. 5: A UUV hull sketch in a CAD environment, with control points are shown in black.
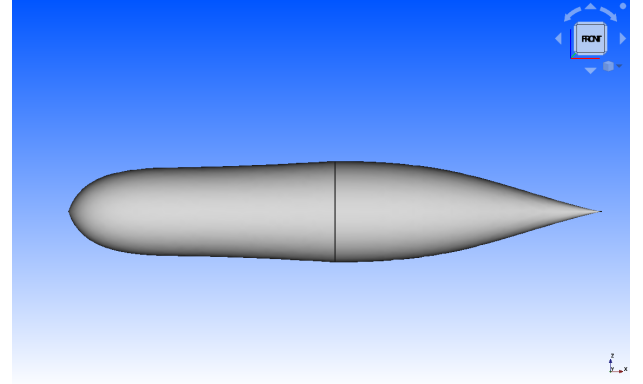

Fig. 6: UUV hull shape in a CAD environment after optimization.

is shown in Table I.

The budget (the number of labeled data points to generate) and sampling methods can both be specified in an *Anvil* configuration file. Currently *Anvil* supports uniform random sampling, Latin hypercube based on minimizing the maximum correlation, and Latin hypercube basewd on maximizing the minimum distance between points. For this experiment, we chose uniform random sampling from the design space and a budget of 100 samples.

V. LIMITATIONS

Anvil is capable of addressing the shape optimization problem for a wide range of air and water-based designs, provided they operate at velocities below the transonic range. At low Mach numbers, air can be considered nearly incompressible, or isentropic, because the changes in flow variables are small and gradual. The assumption of incompressibility is generally

Parameter	Symbol	Minimum	Maximum
Length of nose section	$nose\_radius$	100 mm	800 mm
Length of body section	$fuselage\_length$	100 mm	800 mm
Length of tail section	$tail\_length$	100 mm	800 mm
Thickness of wing	$thickness\_wing$	5 mm	50 mm
Span of one wing	$half\_span$	50 mm	200 mm
Chord length of wing	$chord$	50 mm	200 mm

TABLE I: Design Space: Range of design space parameters for data generation.
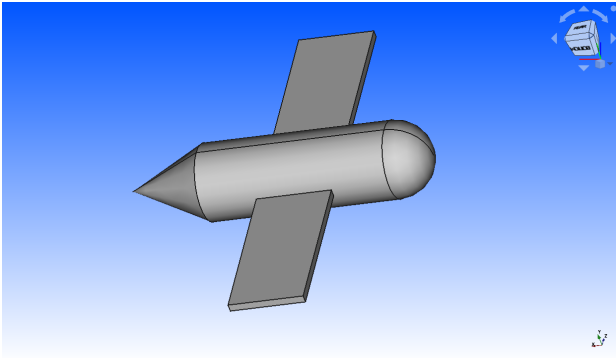
20

Fig. 7: The winged air vehicle design in CAD environment

safe up to 1 Mach, or until the flow becomes transonic [28]. A second limitation of Anvil is that, despite its adaptive mesh size selection, it does not perform a convergence test of the outcomes using different refined mesh sizes. Finally, the tool is limited to handling design optimization problems with up to 20 dimensions due to the constraints of the underlying optimization algorithm, Bayesian Optimization [15].

VI. CONCLUSIONS & FUTURE WORK

Our work presents *Anvil*, an open-source integrated CAD-CFD tool through the combination of FreeCAD, OpenFOAM, Bayesian optimization, and other algorithms. *Anvil* proves to be a versatile scientific machine learning tool, capable of handling data generation, CFD evaluation, and shape optimization for a wide range of subsonic flow conditions, as demonstrated in various use cases. In future work, efforts will be directed towards extending *Anvil*'s capabilities to handle transonic and supersonic flow conditions. Additionally, we plan to enhance the tool's functionality by incorporating convergence testing with varied mesh sizes and expanding its capacity to tackle optimization problems in higher dimensions.

REFERENCES

[1] S. Wang, K. Lim, B. Khoo, and M. Wang, "An extended level set method for shape and topology optimization," *Journal of Computational Physics*, vol. 221, no. 1, pp. 395–421, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999106002968

[2] S. Hazra, V. Schulz, J. Brezillon, and N. Gauger, "Aerodynamic shape optimization using simultaneous pseudo-timestepping," *Journal of Computational Physics*, vol. 204, no. 1, pp. 46–64, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999104004061

[3] J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, and E. Hachem, "Direct shape optimization through deep reinforcement learning," *Journal of Computational Physics*, vol. 428, p. 110080, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999120308548

[4] O. Alexandrov and F. Santosa, "A topology-preserving level set method for shape optimization," *Journal of Computational Physics*, vol. 204, no. 1, pp. 121–130, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999104004103

[5] S. Amstutz and H. Andrä, "A new algorithm for topology optimization using a level-set method," *Journal of Computational Physics*, vol. 216, no. 2, pp. 573–588, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999105005656

[6] G. Allaire, F. Jouve, and A.-M. Toader, "Structural optimization using sensitivity analysis and a level-set method," *Journal of Computational Physics*, vol. 194, no. 1, pp. 363–393, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002199910300487X

[7] H. Vardhan, D. Hyde, U. Timalsina, P. Volgyesi, and J. Sztipanovits, "Sample-efficient and surrogate-based design optimization of underwater vehicle hulls," *arXiv preprint arXiv:2304.12420*, 2023.

[8] M. J. Garcia, P. Boulanger, and S. Giraldo, "Cfd based wing shape optimization through gradient-based method," in *EngOpt 2008-International Conference on Engineering Optimization*, vol. 1. Citeseer, 2008.

[9] V. Griaznov, N. Trigui, H. Affes, and D. Smith, "Cfd based shape optimization of ic engine," *Oil & Gas Science and Technology*, vol. 54, no. 2, pp. 297–307, 1999.

[10] H. Vardhan, P. Volgyesi, and J. Sztipanovits, "Constrained bayesian optimization for automatic underwater vehicle hull design," *arXiv preprint arXiv:2302.14732*, 2023.

[11] N. Marco, J.-A. Désidéri, and S. Lanteri, "Multi-objective optimization in cfd by genetic algorithms," Ph.D. dissertation, INRIA, 1999.

[12] H. Vardhan, P. Volgyesi, and J. Sztipanovits, "Machine learning assisted propeller design," in *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*, 2021, pp. 227–228.

[13] H. Vardhan and J. Sztipanovits, "Deep learning-based fea surrogate for sub-sea pressure vessel," *arXiv preprint arXiv:2206.03322*, 2022.

[14] H. Vardhan, U. Timalsina, P. Volgyesi, and J. Sztipanovits, "Data efficient surrogate modeling for engineering design: Ensemble-free batch mode deep active learning for regression," *arXiv preprint arXiv:2211.10360*, 2022.

[15] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[16] J. R. Lepird, M. P. Owen, and M. J. Kochenderfer, "Bayesian preference elicitation for multiobjective engineering design optimization," *Journal of Aerospace Information Systems*, vol. 12, no. 10, pp. 634–645, 2015.

[17] Y. Morita, S. Rezaeiravesh, N. Tabatabaei, R. Vinuesa, K. Fukagata, and P. Schlatter, "Applying bayesian optimization with gaussian process regression to computational fluid dynamics problems," *Journal of Computational Physics*, vol. 449, p. 110788, 2022.

[18] P. I. Frazier and J. Wang, "Bayesian optimization for materials design," *Information science for materials discovery and design*, pp. 45–75, 2016.

[19] J. Riegel, W. Mayer, and Y. van Havre, "Freecad," 2016.

[20] F. R. Menter, "Improved two-equation k-omega turbulence models for aerodynamic flows," NASA, Tech. Rep. 103975, October 1992.

[21] D. C. Wilcox *et al.*, *Turbulence modeling for CFD*. DCW industries La Canada, CA, 1998, vol. 2.

[22] B. E. Launder and D. B. Spalding, "The numerical computation of turbulent flows," in *Numerical prediction of flow, heat transfer, turbulence and combustion*. Elsevier, 1983, pp. 96–116.

[23] P. Tide and V. Babu, "A comparison of predictions by sst and wilcox kw models for a mach 0.9 jet," in *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008, p. 24.

[24] K. Nikiforow, P. Koski, H. Karimäki, J. Ihonen, and V. Alopaeus, "Designing a hydrogen gas ejector for 5 kw stationary pemfc system–cfd-modeling and experimental validation," *International Journal of Hydrogen Energy*, vol. 41, no. 33, pp. 14 952–14 970, 2016.

[25] H. Jasak, A. Jemcov, Z. Tukovic *et al.*, "Openfoam: A c++ library for complex physics simulations," in *International workshop on coupled methods in numerical dynamics*, vol. 1000. IUC Dubrovnik Croatia, 2007, pp. 1–20.

[26] M. Maroti, W. Hedgecock, and P. Volgyesi, "Rapid design space exploration with constraint programming," in *2022 IEEE Workshop on Design Automation for CPS and IoT (DESTION)*. IEEE, 2022, pp. 27–33.

[27] H. Vardhan and J. Sztipanovits, "Search for universal minimum drag resistance underwater vehicle hull using cfd," *arXiv preprint arXiv:2302.09441*, 2023.

[28] T. Benson. (2021) Mach number: Role in compressible flow. [Online]. Available: https://www.grc.nasa.gov/www/BGH/machrole.html