Python-Based GUI Tool for Automated CFD Design with Parameterized Geometry

Shubh Dwivedi, Aerospace Engineering
Mentor: Jeonglae Kim, Assistant Professor
Ira A. Fulton Schools of Engineering

Figure 1: First few GUI parameter inputs

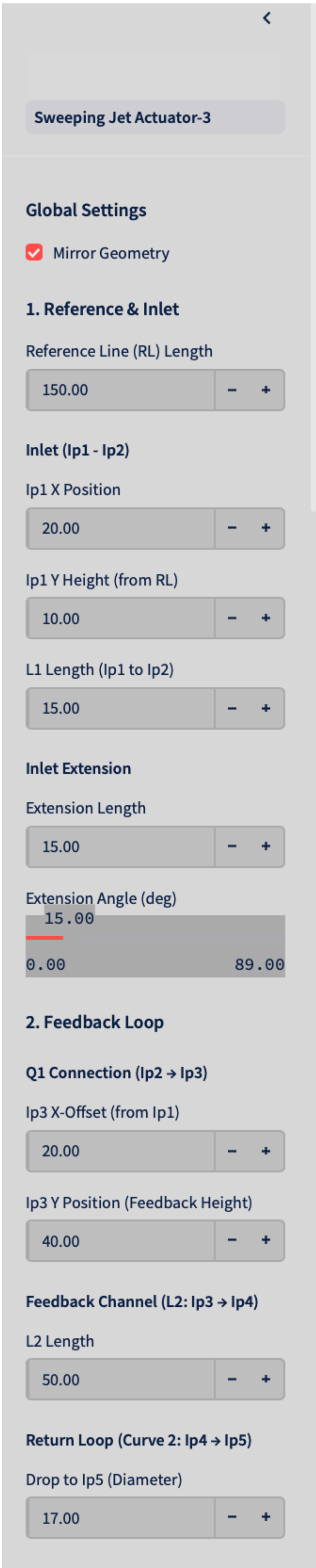


Figure 2: GUI live geometry preview window

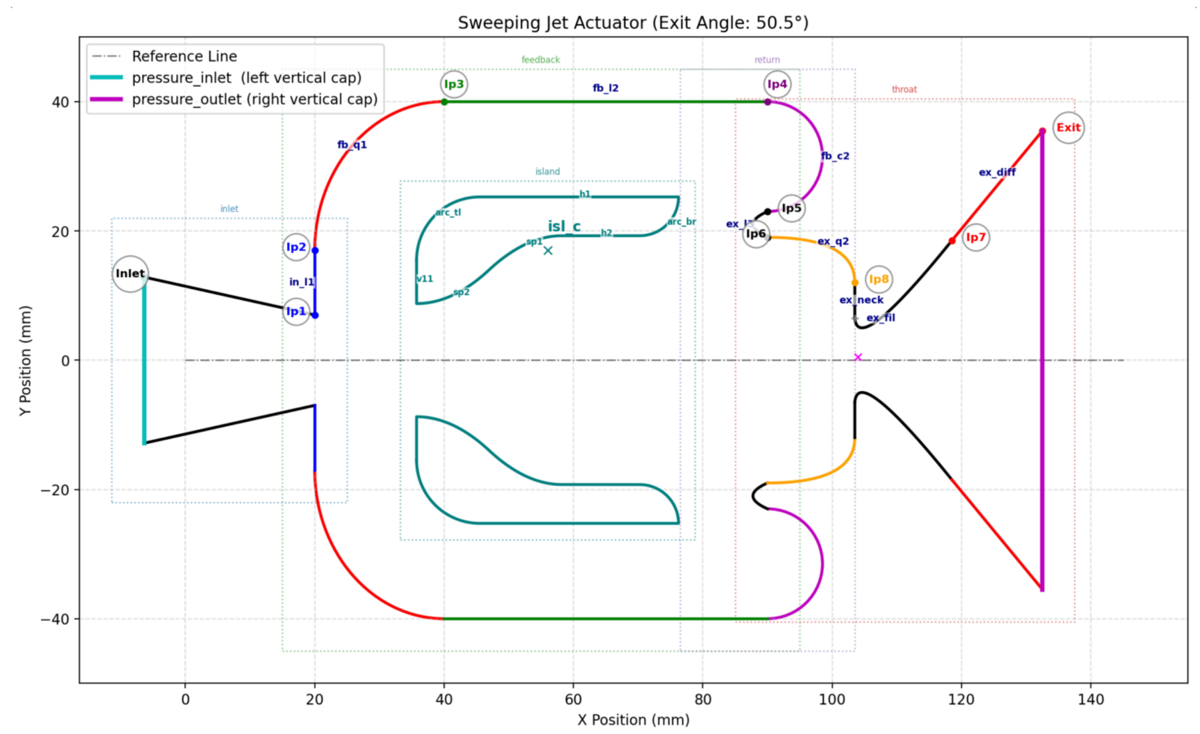


Figure 3: Tool-generated mesh

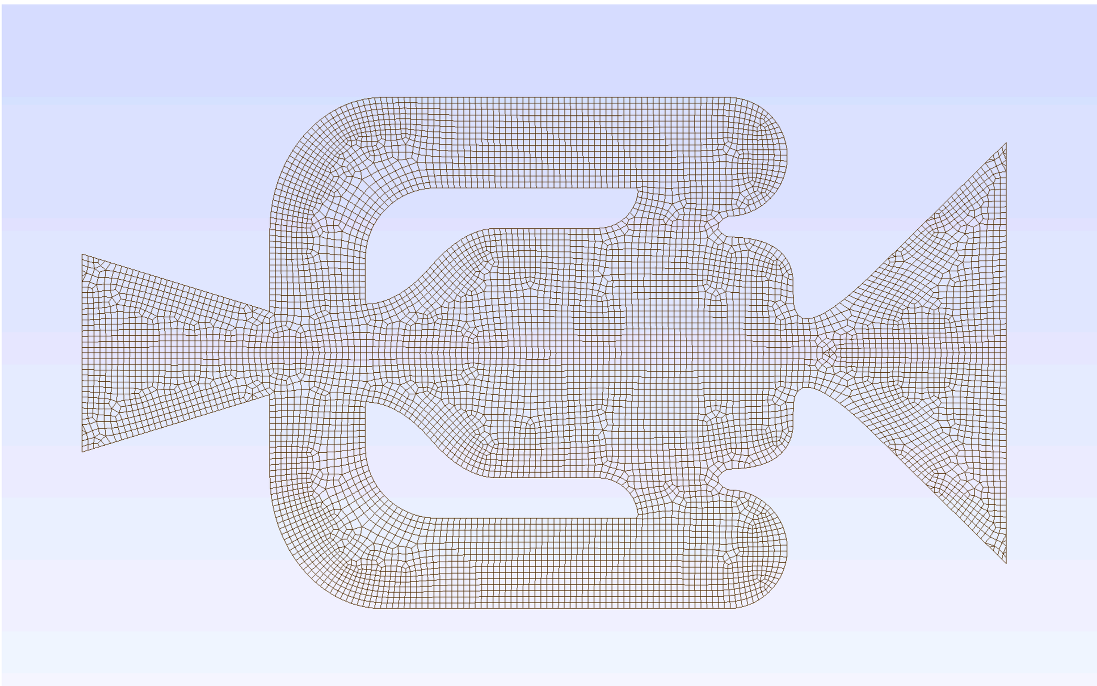
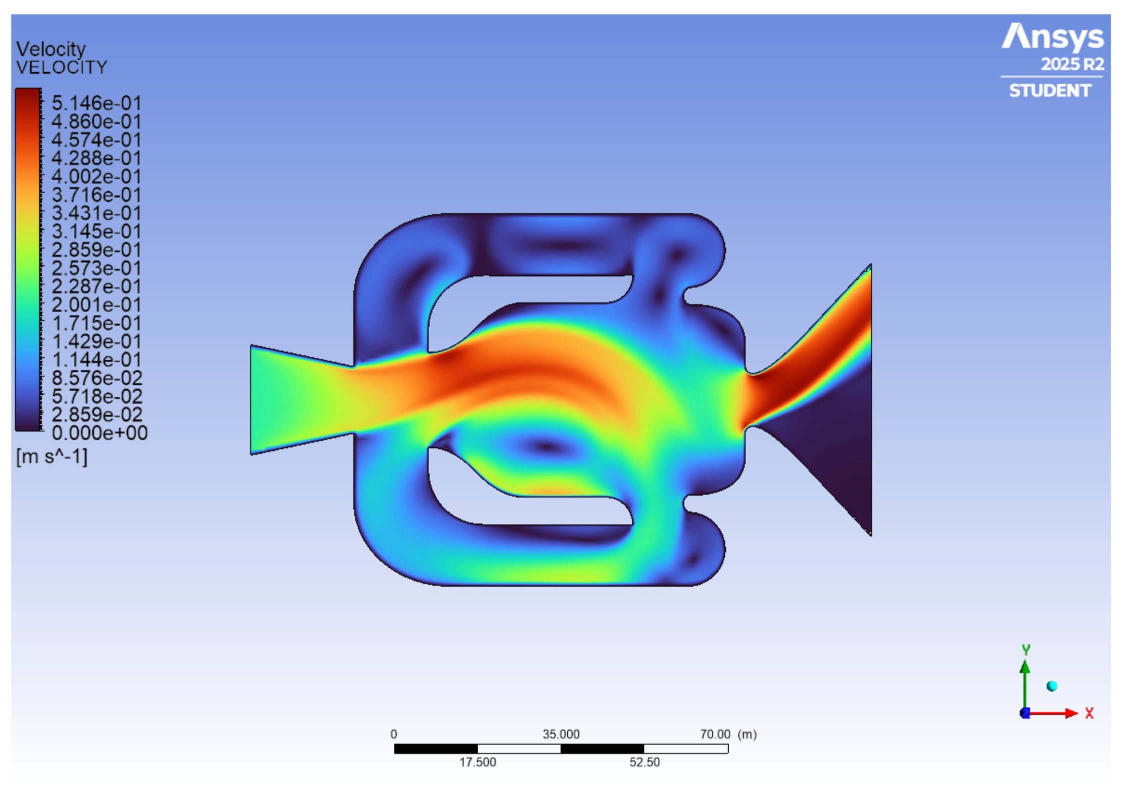


Figure 4: ANSYS Fluent transient simulation (tool-generated geometry and mesh)



Problem / Motivation

Computational Fluid Dynamics (CFD) simulates fluid flow to evaluate the performance of engineered systems. Many fluidic devices, such as sweeping jet actuators (fluid oscillators with no moving parts), require testing multiple geometric variations to identify effective designs.

Current workflows rely on repeated manual geometry creation, meshing, and file transfer between tools. This process is time-consuming, error-prone, and inefficient for iterative design-intensive optimization studies.

Objective

Develop a parametric web-based tool that enables real-time geometry control and automated mesh generation within a unified workflow, allowing rapid and consistent evaluation of design variations.

Example Geometry: Sweeping Jet Actuator

A sweeping jet actuator is a fluidic oscillator that generates an oscillating jet through internal flow feedback, without moving parts. It is a relatively recent technology being actively researched for aerospace flow control applications.

Because its behavior is governed primarily by geometry, it is well-suited for parametric design studies. As a result, it serves as an effective demonstration case for this tool, where small geometric changes directly influence performance and require many design iterations for real-world design optimization.

Tool Architecture / Framework

A graphical user interface (GUI) built with Streamlit serves as the control interface for geometry and meshing parameters. The GUI accepts user input through sliders and text fields and updates the geometry in real time, allowing immediate visualization of parameter changes. It can also generate geometry as pointwise coordinate files (.dat), which can be used to reconstruct surface geometry or be further processed for 3D modeling or printing, eliminating the need for manual CAD sketching.

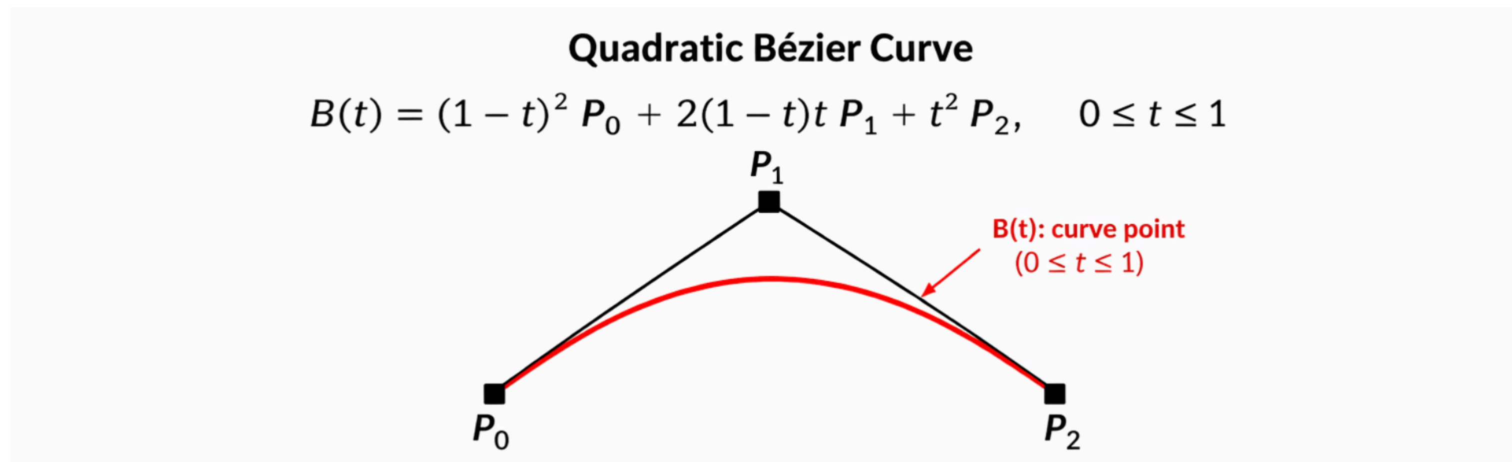
Geometry is generated programmatically using parametric primitives (lines, arcs, Bézier curves), enabling continuous shape variation without manual reconstruction.

The GUI directly controls Gmsh, an open-source mesh generator, which runs in the background to automatically mesh user-generated geometries based on selected mesh settings. From the GUI, users can directly export custom meshes in .cgns format for simulation use. The system includes 30+ tunable parameters governing geometry and mesh behavior.

Key Parameters

- Inlet Height / Length → sets incoming flow profile and development
- Feedback Height & Length → controls oscillation behavior and jet switching
- Throat Width → governs velocity acceleration and pressure drop
- Exit Angle → determines jet direction and spreading
- Island Size & Position → drives flow asymmetry and instability formation
- Mesh Settings → controls overall element type, size, and quality

Figure 5: Quadratic Bézier curve (actuator throat region)



Acknowledgments: Sincere thanks to Prof. Jeonglae Kim for his mentorship, guidance, and continued support throughout this project, along with valuable technical insight and direction.

Design Insights

Small changes in geometric parameters, particularly channel curvature and dimensions, produce significant differences in flow behavior, highlighting strong sensitivity to localized geometry.

Parametric control enables rapid, systematic exploration of these variations, while the GUI consolidates geometry generation and meshing into a single interface, reducing iteration time and manual effort.

Combining multiple parameterization methods (lines, arcs, Bézier curves, splines) provides both local and global control, allowing complex geometries to be modified efficiently with a compact set of inputs.

Future Work

Future work will expand the tool to additional no-moving-part fluidic devices beyond sweeping jet actuators. Further development will focus on advanced parameterization methods (higher-order curves and piecewise surfaces) to enable more flexible and precise control of complex fluid geometries.

Figure 6: 3D printed model from tool-generated geometry

Scan to view tool demonstration video, generated geometries, meshes, and 3D printed models.

FURI

Ira A. Fulton Schools of Engineering
Arizona State University