CSE6730 Project: Jet Engine Flow Simulation Checkpoint 2

Sijian Tan, Kaiqun Peng, Cheng Zhang

Abstract

This project conducts a simulation of supersonic flow through a 2D scramjet engine inlet, utilizing an adaptive, first-order finite-volume method implemented in Python. Emphasizing total pressure recovery calculation at the engine inlet's end, the project also offers insightful visualizations of Mach and total pressure contours. The study aims to advance supersonic inlet aerodynamics and integrate computational science and engineering (CSE) to solve complex aerospace challenges. Our implementation is available on the course github repository.

1 Progress Summary

In Checkpoint 1, we detailed the project's background, motivations, and objectives, and clearly defined the system and model used. We also generated the 2D CAD file, created the corresponding mesh file, and successfully plotted the mesh using Python.

Over the recent weeks, our team has achieved significant advancements in our project. We began by thoroughly studying the well-known Computational Fluid Dynamics (CFD) method, Roe flux, and subsequently integrated it into our Python framework as a helper function. In addition to this, we have developed various aerodynamics-related helper functions, including those for calculating Mach number, normal vectors, and total pressure. The core of our work is included in 'PressureRecovery.py'. This script examines each element along the internal and external edges within a for loop. Here, the Roe flux and other crucial parameters are computed, corresponding to various boundary conditions. By performing intricate calculations for each cell, we have successfully implemented the total pressure computation, marking a pivotal milestone in our project.

2 Works Remained

In the upcoming phase of our project, our primary objective will be to assign distinct colors to each cell within the geometry, reflecting various profiles such as Mach number, velocity, and pressure. We aim to visualize these distinctions by plotting contours in Python. Furthermore, we will undertake a comprehensive convergence and residual analysis to evaluate our model's performance. Based on the insights gained from this analysis, we plan to refine and optimize our model as necessary, ensuring its accuracy and effectiveness.

3 Division of Work

1. Sijian Tan: Roe flux implementation, Mach/pressure contour plot generation, mesh adaption, ANSYS validation
2. Kaiqun Peng: CAD, mesh generation, total pressure recovery implementation, mesh adaption
3. Cheng Zhang: Github version control, FVM implementation, convergence analysis