Expanded 1-Month CFD Solver & MDO Project Plan

Overall Scope & Tech Stack

Language: C++17 with CGNS API, NLopt/Pagmo, Eigen, CLI11, and plotting via CSV+Python or matplotlib-cpp. Post-processing with ParaView/Tecplot. Build with CMake and Catch2 for testing.

Week 1 – Foundation Setup (Mesh, CGNS, Baseline Euler Solver)

Goal: Read a CGNS mesh, run a steady 2D compressible Euler solver, validate with wedge shock. Steps: Create meshes in Gmsh (airfoil & duct), import via CGNS, build cell/face structs, implement HLLC Riemann solver with MUSCL reconstruction and RK3. Apply BCs for inflow, outflow, slip walls. Validate shock angle with oblique-shock theory.

Week 2 – Project 1: Non-reacting Airfoil MDO

Goal: Optimize airfoil shape for drag minimization at fixed lift. Geometry via Bézier control points or Hicks–Henne bumps, remeshed in Gmsh per iteration. CFD black-box objective: drag with lift penalty. Optimizer: NLopt/Pagmo, 3–5 design vars, ~50–100 evals. Post-process Cp curves, Mach contours, objective history.

Week 3 – Project 2 Start: Reacting-flow Extension

Goal: Extend solver to multi-species Euler with simple chemistry. Equations include mass fraction transport + global 1-step Arrhenius reaction. Thermo from mixture cp(T), γ(T). Use operator splitting: Euler convection + chemistry ODE integration (implicit BE or sub-cycled explicit). Test case: supersonic duct with side fuel injection, validate physicality (T rise, p0 loss).

Week 4 – Project 2 MDO (Reacting Combustor)

Goal: Optimize combustor parameters (e.g., wall divergence, injector angle/position, fuel rate). Objectives: maximize efficiency/thrust, minimize total pressure loss with penalties. Budget: 5–10 evals. Deliverables: Mach/T/species contours, efficiency/thrust history, optimized design vs baseline.

Code Architecture

Proposed structure: /mesh (CGNS reader, BCs), /numerics (Riemann, limiter, RK3), /euler, /reacting, /io, /mdo, /utils, main.cpp with CLI. Configs in YAML with freestream, CFL, chemistry constants, optimizer settings.

Numerical Details & Guardrails

Use HLLC flux, Venkatakrishnan limiter, positivity-preserving fixes. Ramp CFL, save restarts, warm-start optimization. Chemistry integration stabilized by clipping T/Ys, enforcing sum(Ys)=1. Residual smoothing and coarse-to-fine mesh strategy for efficiency.

Day-by-Day Milestones

D1–2: CGNS reader + mesh, D3–4: HLLC+MUSCL+RK3, D5: wedge validation, D6–8: airfoil param geo+optimizer hook, D9–10: run optimization + report, D11–13: chemistry extension + duct case, D14: combustor optimization + report.

Deliverables

Codebase, optimization configs, plots (Cp, contours, objective history), and two concise reports (airfoil, combustor).