Computational Visualization of Space Capsule Atmospheric Reentry

João Alves
Master's Degree in Advanced Computing
University of Minho
Braga, Portugal
pg61439@alunos.uminho.pt

Luís Carmo
Master's Degree in Advanced Computing
University of Minho
Braga, Portugal
pg61440@alunos.uminho.pt

Miguel Cruz
Master's Degree in Advanced Computing
University of Minho
Braga, Portugal
pg61442@alunos.uminho.pt

***Abstract*—This project sets up a complete workflow to simulate a reentry capsule flying at Mach 5. We ran the physics using OpenFOAM's rhoCentralFoam solver on the Deucalion supercomputer (AMD EPYC 7742), configuring it for compressible flow. The simulation mimics stratospheric conditions: 1000 Pa pressure and 220 K temperature, with the capsule traveling at 1700 m/s. Finally, we used ParaView to visualize the results, allowing us to inspect the shock structures, temperature spikes, and pressure loads on the shield.**

I. Introduction

Atmospheric reentry involves hypersonic flow regimes where spacecraft decelerate from orbital velocities, generating extreme thermal and pressure loads through shock compression. Understanding the spatial distribution of these phenomena is critical for thermal protection system design and vehicle stability analysis.

Fig. 1: Command module Columbia, used in the Apollo 11 mission [1]

This work implements a complete pipeline from CFD simulation to scientific visualization for a reentry capsule at Mach 5, the capsule in question being a simpler version of the Columbia command module from the Apollo 11 mission, as seen in Fig. 1. We use OpenFOAM's rhoCentralFoam solver executing on the Deucalion supercomputer's x86 partition to simulate stratospheric reentry conditions. The simulation outputs native OpenFOAM field data which we visualize in ParaView to extract temperature distributions, pressure fields, shock structures, and flow topology.

II. Background and Physical Context

Hypersonic flow (Mach > 5) converts kinetic energy to thermal energy through shock compression. A strong detached bow shock forms ahead of the capsule, where the flow undergoes an abrupt transition from hypersonic to subsonic velocities. This creates a stagnation region where temperatures exceed 2000 K even at stratospheric conditions (220 K freestream). The thermal jump across the shock drives aerodynamic heating-the primary constraint for thermal protection systems.

The flow exhibits strong density and pressure gradients across the shock layer that require shock-capturing numerical schemes (rather than shock-fitting approaches). Compressibility effects are fundamental: density couples to pressure through the equation of state (ρ = p/RT for perfect gas), making the energy equation essential alongside mass and momentum conservation. These characteristics make hypersonic CFD computationally demanding, requiring fine spatial resolution near shocks and boundary layers, plus small timesteps to satisfy the Courant-Friedrichs-Lewy stability condition.

III. Simulation Methodology

We use OpenFOAM rhoCentralFoam solver [2], a density-based compressible flow solver with shock-capturing through Kurganov-Tadmor schemes. The simulation runs on Deucalion's x86 partition (2x AMD EPYC 7742, 2x64 cores per node).

Flow conditions represent stratospheric reentry at Mach 5: freestream velocity 1700 m/s, pressure 1000 Pa (~30 km altitude), temperature 220 K. We model air as a perfect gas with constant properties ($\mu = 1.8 \times 10^{-5}$ Pa·s, Cp = 1004.5 J/kg·K) and laminar flow. While real hypersonic flows are turbulent with temperature-dependent properties, this configuration captures essential shock physics while remaining computationally tractable.

The solver uses adaptive timestepping with initial (initial Δt = 1×10^{-6} s, maintaining Courant number ≤ 0.5 for numerical stability. Total physical time simulated is 0.05 s (50 ms). Field outputs (pressure p, temperature T, velocity U, density ρ, Mach number Ma) are written every 500 timesteps in OpenFOAM's binary compressed format.

IV. Visualization

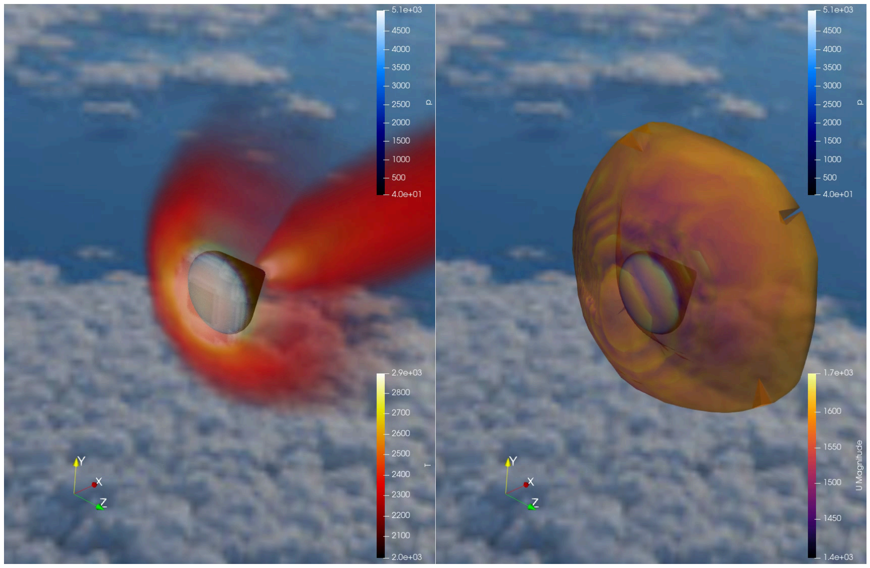

Fig. 2: Visualization of the aerothermodynamic flow field around the Apollo command module during hypersonic reentry

To illustrate the flight conditions, the simulation was rendered in ParaView, against a backdrop of high-altitude clouds, representing the vehicle's descent through the stratosphere. Due to having too many parameters to correctly display in a single pane, the view was split into two panes. This visualization allows us to get a much better look at what happens in the simulation, instead of just the values given by OpenFOAM.

The left pane utilizes volume rendering to visualize the intense heat surrounding the capsule. This technique creates a semi-transparent "fog" that represents the **plasma sheath**, the layer of superheated air caused by the capsule smashing into the atmosphere at hypersonic speeds. As seen in Fig. 2, the temperature mostly fluctuates between 2000K and 2900K.

The right pane uses an isosurface (a 3D contour) to reveal the shape of the bow shock. This "glass shield" represents the exact boundary where the air abruptly changes from undisturbed to compressed. It effectively maps the invisible aerodynamic wall formed ahead of the vehicle. This surface is colored by velocity, showing how the air slows down drastically as it passes through the shock wave.

Finally, in both images, the capsule itself is colored to show the pressure applied to its surface. The light blue / white area on the heat shield shows it directly taking the brunt of the aerodynamic force, which means the rest of the capsule has significantly lower pressure, as shown by its dark blue color.

V. Parallelism

Considering we're using OpenFOAM for our simulation, which already supports parallelism through multiple processes with MPI. In fact, due to the complexity of the simulation, it was executed on two x86 nodes in the Deucalion Supercomputer [3], using 256 processes across four AMD EPYC 7742 CPUs, which made it possible to calculate many more time steps than using a personal laptop.

The simulation was run in an HPC environment. MPI was used, with OpenFOAM's native runParallel function on the Deucalion cluster in the x86 partition. To assess the benefit of parallelisation, tests were carried out with different numbers of MPI processes, from 2 to 512 cores, using 1/5 of the complete timesteps to optimise the analysis time. Open Foam does not have Open MP available, so an MPI process per CPU core had to be used.

The results confirmed that the simulation benefits greatly from the use of parallelisation. The solver execution time was reduced from 553 seconds with 2 cores to 49 seconds with 64 cores, a speedup of 11.29x. The configurations that used between 16 and 64 cores presented the best ratio between performance and efficiency, with time reductions of over 85% compared to the execution with 2 cores.

The parallel efficiency analysis shows that the best use of resources occurs with 4 cores (56.7% efficiency), which remains reasonable up to 16 cores (43.2%). Efficiency decreases with the increase in the number of MPI processes: 32 cores reach 30.9% and 64 cores only 17.6%. This indicates that MPI communication and synchronisation overhead becomes increasingly impactful. In multi-node configurations (256 and 512 cores), a very high degradation is observed, with efficiencies of 4.0% and 1.9% respectively. Once again, this shows that inter-node communication overhead completely dominates the useful work performed, i.e., it indicates limitations due to communication and memory.

The results in Fig. 3 show that the parallel approach is essential to enable complex CFD simulations such as the atmospheric re-entry of an Apollo 11-style capsule. They also show that there is an optimal point of parallelisation, which in this case is between 16 and 64 cores. Configurations above 128 cores show better times, which may perfectly justify their use even if less efficient, since in Deucalion the use of x86 nodes, except for GPUs, is exclusively for those who have reserved them.

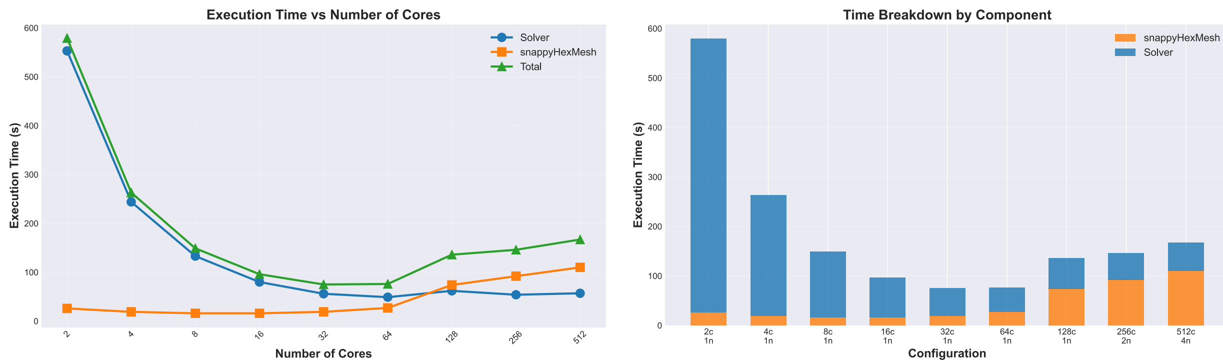

Fig. 3: Graphical analysis of the parallel performance of simulations

VI. Difficulties

Due to having low experience with OpenFOAM and CFD software in general, setting it up took a significant portion of the assignment time, by having to change values and then observe them in ParaView until they were correct. The fact that this simulation was extremely complex did not help, as visualizing it is very demanding on computer hardware. We also tried to increase the speed to mach 20, the actual reentry speed when coming back from the moon, as the Apollo 11 mission did, but implementing it in OpenFOAM is a very difficult task prone to multiple errors, so we decided to go with mach 5 instead, which is still hypersonic speed.

VII. Conclusion

This assignment implemented a complete pipeline from hypersonic CFD simulation to visualization for a Mach 5 reentry capsule. OpenFOAM's rhoCentralFoam solver captured the bow shock structure, shock layer heating reaching 2000 K, and stagnation point pressure loads.

Volume rendering revealed the temperature distribution and shock standoff distance. The velocity-colored isosurface made the bow shock geometry measurable. Surface pressure visualization identified peak thermal protection loads. These techniques extracted physical insight more directly than examining raw field data.

Parallelization analysis showed optimal performance at 16-64 cores, achieving 11× speedup at 64 cores. Beyond 128 cores, communication overhead dominates-512 cores achieved only 1.9% efficiency. The simulation scales well in the sweet spot but becomes communication-bound at large scale.

The main limitation was simplified physics. Laminar flow with constant properties kept the simulation tractable but underestimates real turbulent reentry with temperature-dependent viscosity.

The pipeline successfully demonstrates CFD-to-visualization workflow for hypersonic problems, though production work would require higher-fidelity models.

References

[1] “Apollo 11 Command Module 'Columbia'.” Accessed: Jan. 29, 2026. [Online]. Available: https://www.si.edu/object/nasm_A19700102000

[2] SimFlow Technologies, “rhoCentralFoam - OpenFOAM Solver Documentation.” Online, 2026. Accessed: Jan. 30, 2026. [Online]. Available: https://help.sim-flow.com/solvers/rho-central-foam

[3] MACC - FCCN, “Deucalion User Guide.” Accessed: Jan. 30, 2026. [Online]. Available: https://docs.deucalion.macc.fccn.pt/