

Honors Final Reflection Essay

Aditya Chadha

Over the last three years, I have worked in three high-impact settings that demanded real outcomes and accountability. As Aerodynamics Lead in NC State's High-Powered Rocketry Club, I was responsible for performance-driven decisions that affected a vehicle flying in the NASA Student Launch competition. In the BEFAST Lab, I supported CFD work related to rotating detonation engine research and learned how modeling assumptions shape the trustworthiness of results. During my Continuous Improvement internship at Ontic, I worked on manufacturing-focused feasibility work that forced me to confront the gap between promising ideas and what holds up under testing. Across these experiences, the common thread was learning to integrate academic knowledge with real constraints, transfer methods across contexts, and communicate results in a way others could trust.

My central takeaway is that credibility is built through integration, not isolated competence. Each environment pushed me to connect what I knew from coursework to the messy details of real systems, and each one exposed different failure modes in my thinking. HPRC taught me that leadership is earned through consistent ownership and the ability to coordinate people under time pressure. BEFAST taught me that simulation results matter only if the physics and numerical choices behind them can withstand scrutiny. Ontic taught me that even a strong presentation can collapse if the validation plan is weak. By connecting these experiences, I can show how I moved from simply doing technical work to making decisions that hold up across disciplines, stakeholders, and constraints.

My experience in HPRC is the clearest example of how learning became real through responsibility. I joined the club early because I wanted to work on something hands-on rather than see aerospace concepts in lectures. At first, I contributed where I could, helping with smaller tasks and learning how the team operated. Over time, I moved through roles that gave me broader exposure, and eventually I took on aerodynamics responsibilities that directly affected flight performance. When I became Aerodynamics Lead, the expectations shifted. It was no

longer enough to be curious or hardworking. I had to build systems, justify choices, and ensure the team could act on the information I provided.

A major part of that work was understanding how drag changes with Mach number and with airbrake deployment. In class, drag and compressibility can feel like clean concepts with clean equations. In HPRC, the messy reality is that drag is a function of geometry, deployment state, and flow regime, and must be understood well enough to be useful for decision-making. I learned that the value of an aerodynamic model is not whether it looks impressive, but whether it helps the team predict performance, design for reliability, and plan for uncertainty. That required taking coursework on compressible flow and aerodynamics and turning those ideas into a workflow that supported real design decisions.

BEFAST gave me a different kind of pressure and a different kind of learning. In a research setting, the question is not only whether you can produce a result, but whether your result is defensible. The work I supported involved CFD analysis connected to rotating detonation engine research, with attention to boundary-layer physics and heat transfer. This forced me to treat the near-wall region as the center of the problem rather than a detail. In many class contexts, the boundary layer can be presented as something you study conceptually, then move past. In BEFAST, it became clear that if you do not resolve or model the near-wall region appropriately, the heat-flux results can be misleading or meaningless. That shift changed how I interpreted any simulation. Instead of viewing a plot as the final product, I learned to view it as a claim that mesh resolution, modeling choices, and sanity checks must support.

My internship at Ontic added a third perspective, completing the triangle. I worked on Continuous Improvement initiatives and led feasibility research for a machine-learning vision-inspection concept. This was valuable because it required me to speak to technical and non-technical stakeholders and to define success in practical terms. The initial idea had a strong appeal. The challenge was that the demo results underperformed expectations, and that experience was a turning point for how I think about verification and communication. I learned that feasibility is not a vibe or a pitch. It is evidence that survives realistic constraints such as lighting conditions, part variation, throughput requirements, and integration costs. Ontic taught me to ask a different set of questions early, especially questions about validation design, edge cases, and what would count as a meaningful performance threshold.

When I look at these experiences together, I see progression in how I define good engineering work. In HPRC, good work meant enabling a rocket to perform reliably. In BEFAST, good work meant building confidence in a simulation result by defending the assumptions behind it. In Ontic, good work meant grounding an idea in practical validation and stakeholder needs. Each setting demanded a different kind of maturity, and integrating across them strengthened my judgment.

Two disciplinary threads run through all three experiences. The first is aerodynamics and simulation. The second is thermal fluids and heat transfer, including the idea that a design is only as good as its most limiting constraint. In HPRC, I learned to treat drag as a system-level driver rather than a number. In BEFAST, I learned that credible simulation is a chain where every link matters. Mesh choices, turbulence modeling assumptions, boundary conditions, and post-processing decisions can all influence the physical story you think you are seeing. In Ontic, I saw a parallel lesson in a different domain. A model or algorithm is only valuable if it performs under real data conditions, which is the manufacturing version of simulation credibility.

What changed my thinking is that performance is never only aerodynamic. In BEFAST, heat flux became a feasibility driver. A geometry that performs well aerodynamically can still fail if the thermal loads are not survivable, and survivability is not a theoretical afterthought. The analysis forced me to connect compressible flow behavior to boundary-layer heating and material limitations. That mindset transfers directly to rocketry. Airbrakes can improve apogee targeting by changing drag, but they introduce constraints in deployment reliability, structural loading, and control logic. If I only optimize aerodynamic performance without integrating those constraints, the design is fragile.

The strongest learning came from combining multiple forms of evidence. In HPRC, I leaned on simulation outputs and flight performance expectations. In BEFAST, I had to connect numerical outputs to physical reasoning about near-wall gradients and heat flux. In Ontic, I had to connect metrics from a demo to operational realities, such as throughput and variance. The synthesis across these disciplines is what made the learning integrative. In each case, I had to answer not only what the result was, but why it should be believed, what it implies, and what it cannot guarantee.

The clearest transfer across contexts is the simulation workflow mindset and the need to adapt it when assumptions break. In coursework, it is easy to apply a method as a sequence of steps. In practice, the sequence is never stable because real problems contain constraints that are not explicitly stated in the instructions. Across BEFAST and HPRC, I learned to explicitly define assumptions, run the pipeline, validate reasonableness, and iterate until the results tell a coherent story. That sounds straightforward, but the hard part is that you have to know what “coherent” means physically, not only visually.

A specific example of transfer happened during a compressed research deadline. I had a presentation due in a matter of days, and the initial figures were not accepted because they did not communicate progress convincingly in class, which could have turned into a scramble to produce a plot. In the lab environment, that approach would have failed because the group needed results they could build on. I adapted by tightening the loop between setup, diagnostics, and post-processing. Instead of treating simulation runs as long black boxes, I introduced more frequent checks, clearer criteria for what counted as improvement, and more disciplined post-processing. Hence, the figures matched the story we could defend. The outcome was not simply that I produced better plots. The deeper transfer was learning how to run a workflow under pressure without sacrificing credibility.

That same adaptation mindset shows up in HPRC. When a team is close to a milestone, the temptation is to treat simulation as a confirmation tool rather than a reasoning tool. I learned to resist that. If a result looks too good, I need to ask what assumptions could be hiding risk. If a result looks confusing, I need to isolate variables and verify inputs. Transfer, for me, is not copying techniques from one setting to another. It involves recognizing the underlying logic of a method and reshaping it to accommodate new constraints.

Another major area of growth across these experiences was communication, especially in integrating evidence rather than relying on confidence. In each environment, the audience was different, and I had to adjust the format while keeping the reasoning clear.

In HPRC, an effective communication artifact might be a contour plot of drag versus Mach number versus airbrake extension level. The point of including that figure is not to decorate a report. The figure is a decision-making tool. It helps the team understand when airbrakes meaningfully change drag, how sensitive performance is to deployment level, and where the

system might see peak loads. The integrated part explains what the figure shows, how it was created, and what decisions it supports. If I include it without explanation, it does not improve understanding. If I explain it without connecting it to decisions, it becomes trivia.

In BEFAST, an effective artifact might be a plot of heat flux across operating conditions or geometry variants. The key is to explain the pattern, why it happens, and what it implies for feasibility. The integrated communication requirement matters here because heat flux results must be paired with a discussion of near-wall resolution, boundary conditions, and model assumptions. A plot without that context can mislead. A plot with that context can guide design decisions and research directions.

In Ontic, an effective artifact could be a feasibility comparison table that includes accuracy metrics, throughput impact, integration complexity, and cost. The lesson I learned is that a table can be more persuasive than a narrative if it captures the constraints stakeholders actually care about. The failure of the demo results underperformance taught me to communicate uncertainty clearly and to avoid overstating readiness. That experience improved my ability to explain limitations while still presenting a path forward.

Across all three settings, I learned that integrated communication means tying artifacts directly to claims. It also means naming limitations. If a simulation result depends on assumptions, I need to say which assumptions. If a demo is incomplete, I need to say what remains untested. That kind of honesty increases trust because it shows I understand the boundaries of my evidence.

These experiences revealed strengths that I can substantiate. One strength is that I can push myself socially and technically when I feel behind. In HPRC, I learned that showing up consistently and asking questions, even when I felt unprepared, was the path to becoming useful and eventually being trusted with leadership. Another strength is that I can deliver under pressure through focused work blocks and systematic iteration. In BEFAST, I learned to work through debugging, reruns, and post-processing until results became coherent and defensible.

They also revealed challenges that I am still working on. Leadership is not only execution. It is coordination, expectation-setting, and sometimes uncomfortable conversations when contributions slip or decisions must be made quickly. As a lead, I had to learn that avoiding those moments does not protect the team. It delays the problem and increases risk. Another challenge

is that I can sometimes over-trust a promising path early, especially when the idea is exciting and the timeline is tight. Ontic showed me that a strong pitch still needs a strong verification plan, and that confidence is not a substitute for validation.

One assumption that changed across these experiences is what it means to be competent. Earlier, I believed competence was mostly technical, meaning you could derive the right equations, run the right software, or deliver the right output. Now I see competence as the ability to integrate. Can I connect physics to constraints, results to decisions, and ideas to verification? Can I communicate in a way that makes other people better, not just informed? That shift is a major outcome of these high-impact experiences.

Going forward, I have a specific plan for continuing learning. First, I want to strengthen my ability to produce heat-flux and near-wall results that are credible across different modeling choices. I will do this by building a repeatable verification checklist that forces me to justify near-wall resolution choices, turbulence model selection, and sanity checks on gradients and flux trends. I will measure progress through lab meeting feedback and by whether my figures survive critical review without major rework. Second, I want to improve my leadership execution by developing a clearer team communication cadence. That means setting expectations earlier, delegating ownership in ways that match people's strengths, and documenting decisions so the team can move quickly under pressure. I will measure progress by tracking whether major deliverables meet milestones, with fewer last-minute crises, and by seeking direct feedback from team members on clarity and support.

Together, these experiences taught me that engineering growth happens when I integrate technical methods with people, constraints, and verification. HPRC taught me ownership and leadership under integration pressure, where the cost of a wrong assumption can be a failed flight outcome. BEFAST taught me that simulation results are only valuable when the modeling choices behind them can be defended and when the physics story makes sense. Ontic taught me humility and the importance of practical validation, especially when presenting ideas to decision-makers who care about feasibility more than elegance. Moving forward, I want to build a career at the intersection of simulation and real-world validation, with a long-term focus on CFD, boundary-layer physics, and heat transfer in advanced propulsion systems. These experiences did

not just increase my skills. They changed how I think, how I communicate, and what I demand from my own work before I call it credible.