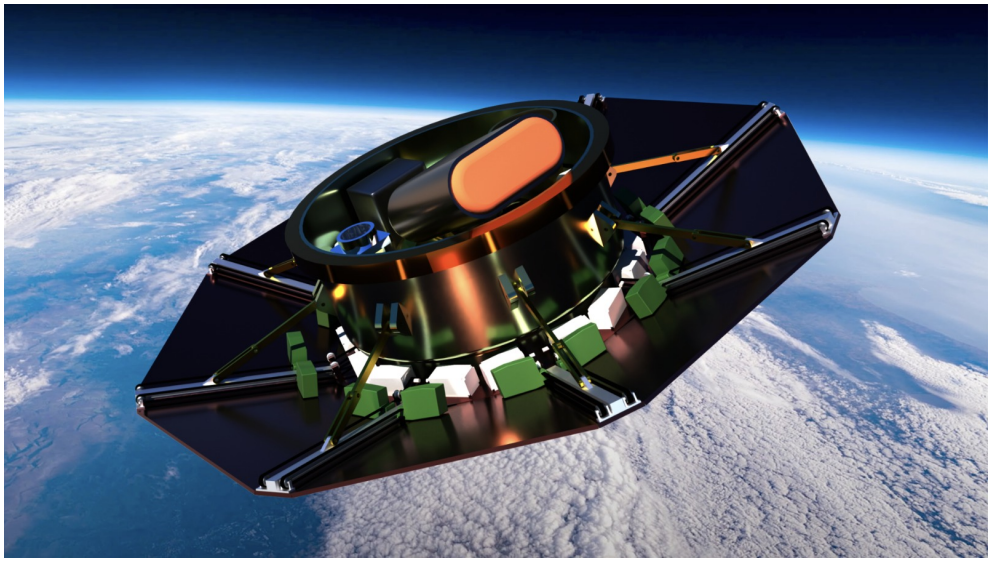
Imperial College London

FOLDABLE AEROSHELL DEMONSTRATOR

Space HAVEN: Aerothermodynamic Analysis of a Hypersonic Rocket Nosecone

FAD07 - Aerodynamic Analysis



Student: Nathanael Jenkins
CID: 01843784
Date: 16/06/23

Academic Responsible: Dr Paul Bruce
Supervisor: Dr Konstantinos Steiros

Department: Department of Aeronautics
Course: MEng Aeronautical Engineering
Module: AERO60004 - Group Design Project
Academic Year: 2022/2023

Department of Aeronautics
South Kensington Campus
Imperial College London
London. SW7 2AZ.
U.K.

Abstract

The aim of this study is to assess the aerodynamic performance of different nosecone designs for a conceptual, hypersonic sounding rocket. Drag forces and heating effects are evaluated using analytical and computational methods. Computational Fluid Dynamics (CFD) simulations are employed to quantify the drag coefficient, maximum nose temperature, and nosecone pressure distribution at different flight conditions. The CFD simulations are validated against experimental data obtained from wind tunnel tests. Adaptive mesh refinement is implemented to accurately capture shocks within an axisymmetric finite volume domain. The findings of this report provide valuable insights into the aerothermodynamic performance of the nosecone and contribute to the selection of an appropriate design.

FAD22: Space HAVEN		
Sub-Group	**Name**	**CID**
FAD01 - Project Co-ordination	Amy-Beth Curtis	01859027
	Eunice Lam	01861076
	Shapol M	01865267
FAD02 - Mission Design & Analysis	Joshua Goldspink	01870028
	Gul Kaur	01905296
	William Taylor	01771592
	Usmaan Yaqoob	01847219
FAD03 - Launcher Structural Design	Myka Abaquin	01857878
	Kunpeng Li	01909532
	Julian Ting	01843067
	Yichen You	01863827
FAD04 - Launcher Subsystem Design	Nicolas Bayle	01943039
	Lara Couceiro Alves	01877828
	Pablo Duhamel	01915106
	Artem Sheykin	01913084
FAD05 - Demonstrator Structural Design	Harry Allcock	01854056
	Imsara Samarasinghe	01949330
	Timur Uyumaz	01906145
	Raine Wong	01876352
FAD06 - Demonstrator Subsystem Design	Paula Gutierrez	01949543
	Cristian Vegas Medina	01874108
	Sevan Vlieghe	01926818
	Chris Ziyi Yao	01908293
FAD07 - Launcher & Demonstrator Aerodynamic Analysis	Yuzhe Dun	01849823
	Nathanael Jenkins	**01843784**
	Mohammad Kapadia	01862576
	Joel Tomas Pimentel	01881299

Contents

List of Figures

List of Listings

List of Tables

List of Symbols

A	Cross-sectional area
C_D	Drag coefficient
C_h	Conductive heat transfer coefficient
D	Rocket diameter
Kn	Knudsen number
M	Mach number
P_D	Power lost to drag
R	Nose radius
Re	Reynolds number
T	Static temperature
T_0	Stagnation temperature
V	Velocity
c_p	Specific heat at constant pressure
h	Enthalpy
k	Thermal conductivity
q	Heat flux
$\dot{q}$	Rate of heat flux
$y+$	Wall y+
γ	Ratio of specific heats
δ_c	Nose half-vertex angle
ϵ	Emissivity
μ	Dynamic viscosity
ρ	Density
σ	Stefan-Boltzmann constant
ϕ_{aw}	Flow property ϕ at an adiabatic wall
ϕ_e	Flow property ϕ at edge of boundary layer
ϕ_w	Flow property ϕ at the wall
ϕ_2	Flow property ϕ aft of normal (bow) shock
ϕ_∞	Flow property ϕ in freestream
ϕ^*	Reference value for flow property ϕ

List of Abbreviations

AIAA	American Institute of Aeronautics and Astronautics
AMR	Adaptive Mesh Refinement
CFD	Computational Fluid Dynamics
DSMC	Direct-Simulation Monte Carlo
EDL	Entry, Descent and Landing
FAD	Foldable Aeroshell Demonstrator
NASA	National Aeronautics and Space Administration

1 Introduction

1.1 Mission Overview

This project forms part of a mission to test a foldable aeroshell demonstrator in an effort to increase its Technical Readiness Level. The capability to transport large aeroshells using small rockets would enable the Entry, Descent, and Landing (EDL) of large payloads to planets such as Earth and Mars. NASA explains the value and importance of developing such technology in their 2020 'Technology Taxonomy' [8].

Full-scale aeroshells face extremely high temperatures during hypersonic re-entry. To experience these conditions without a costly orbital launch, the deployment of the demonstrator from a high altitude is necessary, approximately $300\,\mathrm{km}$ to reach Mach 7 on descent [9]. A sounding rocket has been deemed the optimal launch vehicle, requiring the design of a suitable nosecone. Preliminary trajectory analysis indicates that the rocket will exceed Mach 7 during ascent, necessitating consideration of aerothermal effects across subsonic, supersonic, and hypersonic regimes.

To test the demonstrator in the most extreme conditions, apogee must be maximised meaning drag on the rocket must be minimised. Given a conceptual design using constant-diameter body tubes [10], the primary opportunity for drag reduction lies in the nosecone design. Thermal analysis is important to ensure the nose does not warp, soften or, at worst, melt under aerodynamic heating. This could result in a catastrophic mission failure, underscoring the importance of understanding nosecone drag and heating for the success of this mission.

This work package experienced high interoperability with the launcher structural design team (FAD06), particularly work conducted by Myka Abaquin [2]. The early definition of an approximate rocket diameter by the mission design team (FAD02) meant that this package was not directly connected to any sub-teams outside of FAD06, enabling fast progress. The package was also disconnected from the rest of the aerodynamic analysis (FAD07) since the launcher and demonstrator must be treated differently. This is because the launcher is more slender and less blunted than the demonstrator. A roadmap outlining the work of the aerodynamic analysis team is given in figure 1; this report addresses the launcher nose shape task.

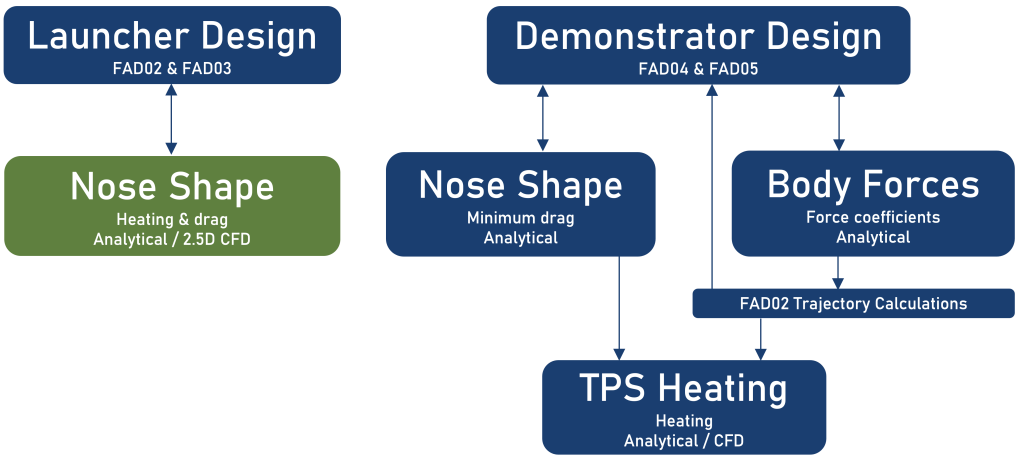


Figure 1: Aerodynamics (FAD07) project roadmap

1.2 Objectives

This report shall contribute to the nosecone design by providing information on the aerothermodynamic performance of various nosecone geometries by two means. Firstly, **analytical and empirical calculations** were conducted to estimate key parameters, namely wall temperature and drag coefficient. Secondly, **CFD simulations** were conducted at several Mach numbers to obtain more precise and relevant data including drag, heating, and pressure distributions. The pressure distributions were used in finite element simulations to guarantee the nose can withstand aerodynamic forces [2]. A successful outcome was established by the selection of a minimum-drag nose geometry based on robust data.

2 Initial Calculations

2.1 Heating

In the subsonic and low-supersonic regimes (around M<1.5), aerodynamic heating is relatively insignificant. Above these speeds, however, the maximum temperature increases with the cube of velocity [11] and at Mach 7 this can exceed $2000\,^{\circ}\mathrm{C}$. Several analytical methods may be used to estimate the temperature of vehicles in high-speed flight. Starting with the most conservative estimate, increasingly relevant approaches were used to estimate aerodynamic heating.

2.1.1 Stagnation Temperature

Stagnation temperature T_0 describes the maximum temperature a flow could reach if brought to zero velocity (for example, at the tip of a blunt nosecone). It is defined in terms of the ratio of specific heats γ, static temperature T, and Mach number M in equation 1. Stagnation temperature is not changed across a shock.

$$T_0 = T(1 + \frac{\gamma - 1}{2}M^2) \tag{1}$$

For a sufficiently blunt body over a sufficiently long time, the wall temperature *may* tend to T_0. However, this fails to take into account the thermal boundary layer, transient effects, or the thermal properties of the air or nosecone material. As such, T_0 should be treated as an absolute maximum, but a poor estimate of actual temperature. Adiabatic wall temperature T_{aw} was also calculated using equations in Bruce [12], but was found to be very close to T_0 for M<7, so T_{aw} not used for further analysis.

2.1.2 Sutton-Graves and Fay-Riddell

The Sutton-Graves method calculates the rate of convective heat flux at the wall based on simple geometry and flow parameters for hypersonic flows. It is described in equation 2, where the constant $k = 1.74 \times 10^{-4}$ for Earth's atmosphere. Combining this with the Stefan-Boltzmann equation for radiative heat flux, given in equation 3, an equilibrium wall temperature can be found. While this relies on many simplifications, it often proves accurate enough to serve as an initial engineering estimate for hypersonic blunt bodies [13].

$$\dot{q}_{conv} = k\sqrt{\frac{\rho_\infty}{R_{eff}}}V_\infty^3 \tag{2}$$

$$\dot{q}_{rad} = \epsilon\sigma T^4 \tag{3}$$

A similar, more advanced equation is given by Fay and Riddell [13], although this requires the computation of a number of non-trivial quantities. Additionally, both the Sutton-Graves and Fay-Riddell methods are derived for blunt bodies and break down for sharp noses. A reasonable nose radius of $5\,\mathrm{cm}$ is sufficiently small that both methods predict peak temperatures far above T_0. Therefore, the results from these approaches have been neglected in further analysis.

2.1.3 Reference Temperature

Another popular method for predicting maximum wall temperature on a flat plate is given by Eckert's reference temperature T^*, modified by Meador-Smart and given in equation 4 for a compressible turbulent boundary layer [11]. It should be noted that the differences between the laminar and turbulent cases are relatively small.

$$\frac{T^*}{T_e} = 0.5(1 + \frac{T_w}{T_e}) + 0.1424(\frac{\gamma - 1}{2})M_e^2 \tag{4}$$

$$Re_x^* = \frac{\rho^* V_2 D}{\mu^*} \tag{5}$$

$$C_h = 0.0296(Re_x^*)^{-0.2} \tag{6}$$

$$\dot{q} = -C_h(T_w - T_{aw}) \tag{7}$$

Mazzoni et al. [14] show that the reference temperature can be used to calculate the reference Reynolds number Re_x^* in equation 5, where V_2 is the freestream velocity after a normal (bow) shock. This is used to calculate the heat transfer coefficient using equation 6 which in turn yields heat transfer rate $\dot{q}$ in equation 7 [11]. The equilibrium temperature can be found using the Stefan-Boltzmann relation from equation 3. The boundary

layer edge quantities T_e and M_e in equation 4 can be approximated using the freestream flow properties aft of a normal (bow) shock.

While this provides some of the most favourable predictions, it must be noted that these formulae apply to flow over a flat plate and not a sharp nose. While these results were presented to the structural design team, they were taken with a "pinch of salt".

2.1.4 Marvin-Diewert

Digging deep into the NASA archives, a method was found which was derived specifically for high-speed flow over a sharp, revolute nosecone. Marvin and Diewert predicted $\dot{q}$ on a sharp cone in air using equations 8 and 9 [15], where δ_c is the half-vertex angle of the cone. Again, the Stefan-Boltzmann equation was used to estimate wall temperature from this.

x represents the distance along the nose wall from its tip, meaning $\dot{q} \to \infty$ at the tip ($x \to 0$). To account for this, x was made equal to the nose radius R_N to provide a more realistic estimate. For very small radii, this gave much more realistic results compared to the Sutton-Graves method.

$$\dot{q}_{w_{conv.}} = 4.03 \times 10^{-5} \sqrt{\frac{\rho_\infty \cos(\delta_c)}{x}} V_\infty^{3.2} \sin(\delta_c)(1 - \frac{h_w}{h_{aw}}) \tag{8}$$

$$h_{aw} = h_\infty + 0.4 V_\infty^2 \tag{9}$$

The Marvin-Diewert method provided more realistic results since they were between the flat plate temperature prediction and T_0. Therefore, with some caution, these values were used as an estimate for nose heating. A plot of the most useful temperature estimates is given in figure 2 for the initial trajectory simulation, though these may have changed slightly for the final trajectory presented in [16]. The values in this figure assume a nose tip radius of $1\,\mathrm{cm}$, based on estimates provided by Abaquin [2]. These were shared with the structural design team, who applied safety factors and used the predictions to inform geometry and material choices. Validation of these results is mentioned in section 3.3.2.

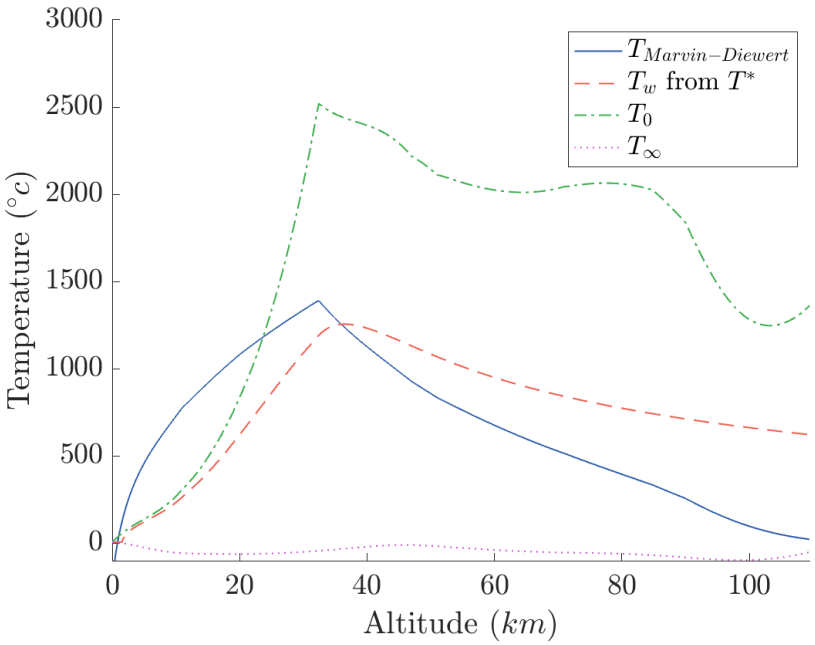


Figure 2: Nose temperature estimates

2.2 Drag Estimates

Estimating heating on a relatively sharp hypersonic nosecone was a challenge, but predicting the aerodynamic drag on such a body is even more complex. Analytical methods developed for hypersonic flow are almost exclusively derived for blunt bodies and/ or much higher Mach numbers than this sounding rocket will experience. While some data is available for the subsonic regime, this is expected (and confirmed in section 2.2.2) to be much less critical than supersonic and hypersonic drag.

2.2.1 Experimental Interpolation

It was not possible to identify any meaningful analytical methods for predicting the supersonic or hypersonic drag coefficient of a sharp nosecone so interpolation of experimental data was used for a crude estimate of drag. NASA [1] experimentally tested a range of conic nosecones, one of which is very similar in shape to a preliminary nose design presented by the launcher structures team [2]. A plot of this data is given in figure 3.

The data in figure 3 was obtained through wind tunnel testing, where C_D is the nosecone forebody drag coefficient, neglecting base drag. This corresponds to the value which will be later obtained from CFD simulations, meaning that this data could also be used to validate CFD results.

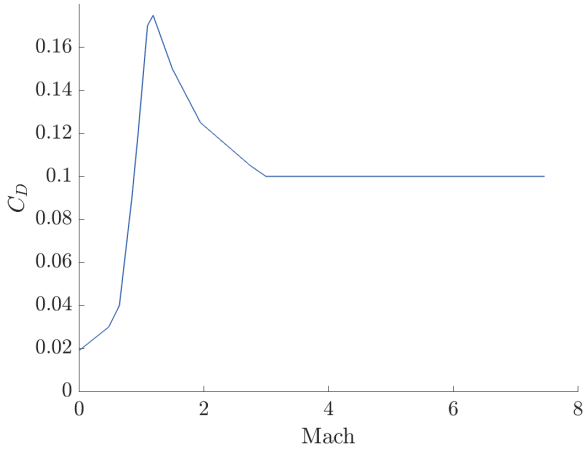


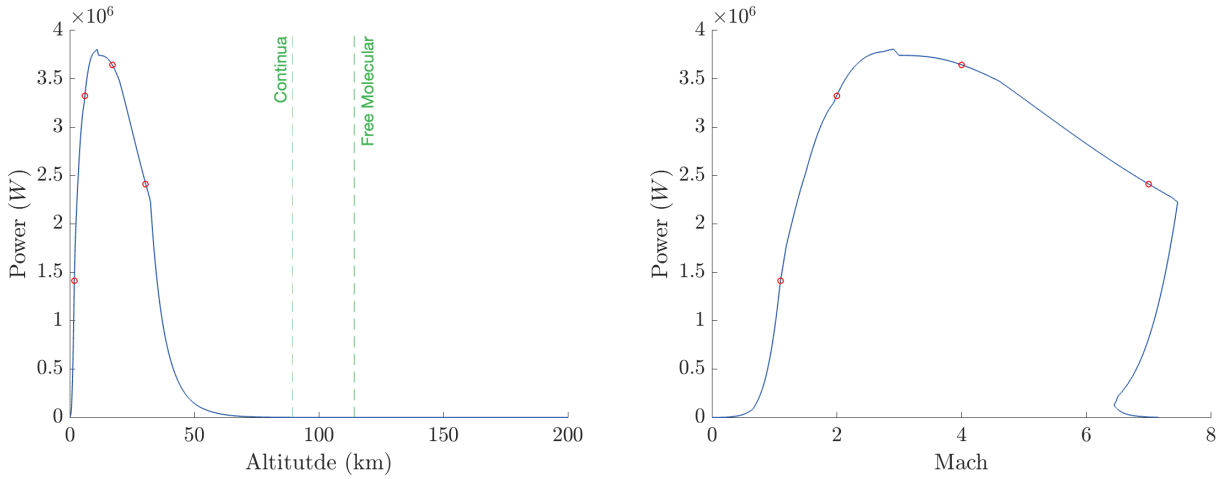
Figure 3: Experimental nosecone forebody drag data [1]

Some discussion was found in Maccoll and Taylor [17] on methods for predicting the forces on conic nosecones, although finding solutions using this approach (particularly for curved shapes) would be more involved than conducting CFD analysis and requires excessive assumptions about the flow.

2.2.2 Critical Drag Regions

Based on interpolations of the data in figure 3, the power of drag on the nosecone could be estimated using equation 10, which can be plotted for the preliminary trajectory in figure 4. These plots proved vital in determining which flight regimes to focus on when analysing drag; red markers indicate the points at which CFD simulations were conducted. A change to the trajectory late in the design process [16] means these profiles have now changed, though these changes were small ($<6\,\%$).

$$P_D = \frac{1}{2}\rho_\infty V_\infty^3 C_D A \tag{10}$$



(a) Variation with altitude (ascent only)

(b) Variation with Mach number (ascent only)

Figure 4: Power lost to nosecone forebody drag

Firstly, it can be seen from figure 4a that most power lost to drag occurs in the fluid continua regime, significantly simplifying analysis. The continuum regime was defined based on the criteria outlined by Anderson [11] that Kn<0.03 for Navier-Stokes solutions to apply, without the consideration of wall slip conditions. Additionally, figure 4b indicates that supersonic and hypersonic drag is most significant. The subsonic and transonic regimes, however, can be largely neglected.

3 Computational Fluid Dynamics

The vehicles most similar to hypersonic sounding rockets are military missiles. This means most publicly available information relevant to the design of a minimum-drag, minimum-heating hypersonic nosecone is found in old declassified documents which exclude state-of-the-art material. Analytical methods cannot handle the geometries being analysed in this work package, as mentioned in section 2.2. Therefore, to properly understand the drag on different shaped nosecones, computational analysis was deemed necessary.

The feasibility of CFD analysis was verified in part by figure 4, which indicates that the most significant aerodynamic drag occurs when fluid can be treated as a continuum. As a result, Direct-Simulation Monte Carlo (DSMC) methods are not required [11], and Navier-Stokes-based CFD can be relied upon. Data from the trajectory was used to inform four CFD test cases, outlined in table 1 and circled in figure 4. These were selected to provide data at a range of Mach numbers, with the goal of capturing flow behaviour in the transonic, low-supersonic, high-supersonic, and hypersonic regimes.

Table 1: CFD test cases

M	ρ_∞ $(\mathrm{kg\,m^{-3}})$	T (K)
1.1	1.040	272.5
2.0	0.671	241.1
4.0	0.139	211.6
7.0	0.016	225.0

A commercial finite-volume package, StarCCM+, was used to perform CFD simulations. This was selected because of its excellent built-in meshing tool, wide range of physics models, and the ability to easily integrate parameters and macros for automation. StarCCM+ comes with excellent documentation, helping to prevent operator error. It also provides scope for future integration of multiphase (fluid-solid) thermal modelling to simulate the transient behaviour of the nosecone material, although that is beyond the scope of this project.

3.1 CFD Setup

3.1.1 Geometry

A conic nose geometry was used to set up the initial test case; this was also used for the mesh refinement study discussed in section 3.2. This was easily replaced with any nose geometries presented for analysis by the launcher structures team, who held design authority for the nosecone. Some of the geometries are illustrated in figure 5; the 'aerospike' geometry uses a power-series cone and information on each geometry can be found in Abaquin [2].

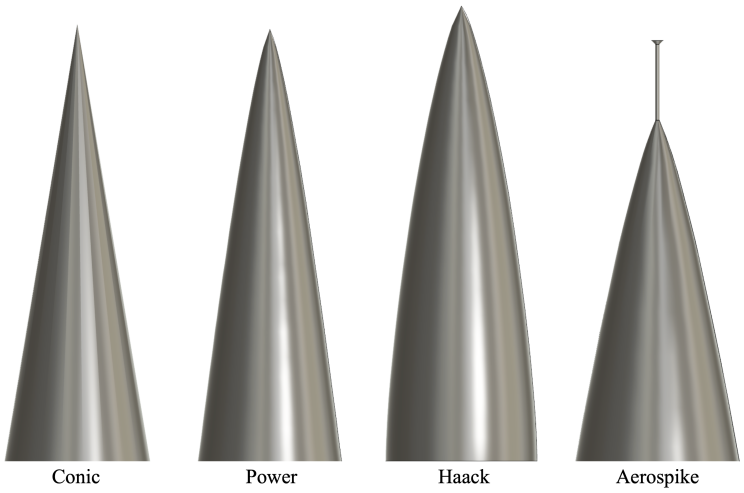


Figure 5: Various nose shapes for CFD testing [2]

Because the rocket is axisymmetric, the domain could be simplified into a 2D 'slice' of half of the rocket, and physics models were used to simulate the axisymmetric nature of the flow. This limited simulations to 0° angles of attack, which is allowable since other angles of attack were not a part of this test plan. A quadrant ($\frac{1}{4}$ circle) domain was used to prevent reflection shocks reflecting off boundaries, with a length of the body tube extending behind the nose and through the outlet. The domain can be seen in visualisations presented in appendix C.3. For a $2\,\mathrm{m}$ long nosecone and $2\,\mathrm{m}$ of body tube, a $5\,\mathrm{m}$ domain radius was used, informed by [18]. Because information is not propagated upstream in supersonic or hypersonic flow, the distance between the inlet and the geometry can be small.

This setup means that it is easy to measure forebody drag - that is, the drag coefficient of the nosecone when it is attached to the body tube, ignoring the effects of the aftbody/ wake. This is lower than the rocket's total drag coefficient but easier to simulate and appropriate for comparing different nose geometries, assuming the rocket body is much longer than the nosecone.

3.1.2 Mesh

A 2D polyhedral mesh was generated on the domain using StarCCM+, which provides excellent unstructured mesh generation tools. A polygonal mesh was selected because its cells neighbour many others, providing higher resolution gradient approximations [19] which are particularly vital when simulating shocks. Additionally, research has found that polygons are more computationally efficient than other mesh types [20], and often converge more consistently [21]. Adaptive mesh refinement was used to ensure accurate modelling of shocks. This is explained in section 3.2.

Near wall prism layers were *not* implemented for two reasons. Firstly, they had little effect on the accuracy of results; testing of models with and without prism layers found that the relative change in C_D was less than $5\,\%$. The test with prism layers used models from [22] to predict the δ_{99} (prism layer total thickness) height, which was to the order of $5\,\mathrm{mm}$. The relatively small effect of the prism layers can be attributed to the characteristics of two physics models discussed in section 3.1.3; StarCCM+ is known to have a high-quality 'any y+' wall treatment, and the k-ω SST turbulence model is designed to handle a range of y+ values so small near-wall cells are not *necessary*. The second reason was that, near the tip of the nose, the prism layers resulted in divergence for high-Mach cases. This is because gradients in flow properties at the tip are not only extreme in the wall-normal direction but also parallel to the wall. Therefore, using high-aspect prism-layer cells at the tip resulted in worse performance.

3.1.3 Physics Models

The selection of appropriate physics models is critical to attaining accurate and relevant CFD results. The selected models are outlined and justified in table 3.1.3.

Table 2: CFD physics models

Type	Model	Justification
Spatial	Axisymmetric	Given the axisymmetric geometry, computational cost can be reduced using this model.
Temporal	Steady	Transient effects are not anticipated. This was validated by visually checking that the residuals did not exhibit any oscillatory behaviour after convergence.
Coupling	Coupled	Segregated flow models are not appropriate for compressible flow, where the interdependencies between momentum and energy become more significant.
EoS	Ideal Gas	As chemical (dissociation) effects are negligible at these velocities [11], the ideal gas model provides an accurate and computationally efficient model for air.
Turbulence	k-ω SST (Menter)	This model has been widely validated in hypersonic flows [23, 24, 25] and benefits from the advantages of both the $k-\epsilon$ and $k-\omega$ models. $k-\epsilon$ performs best for attached flows, whilst the $k-\omega$ model improves performance near the wall and any regions of separation [26]. The necessity of a turbulence model is demonstrated in section C.1, where simulations are conducted using k-ϵ, k-ω SST, and laminar models.
Wall Treatment	All-y+	The good performance of $k-\omega$ SST near walls and the well-designed 'all-y+' model mean that no complicated prism layers are required near the wall. Instead, provided y+<1 or y+>30 (verified during meshing), this model is known to be accurate [27].

The inflow boundary and initial conditions require a turbulence intensity and viscosity ratio when using the $k-\omega$ model. The turbulence intensity was set to $1\,\%$ based on the low expected turbulence in stationary air [28]; the viscosity ratio was then calculated using standard relations given in [22], using body diameter as a reasonable approximation of the turbulent length scale, based on Guerrero [28]. No transition model was implemented since $y+>1$ [29] and *'the accurate prediction of transition at hypersonic speeds is one of the leading state-of-the-art questions'* [11].

3.1.4 Boundary Conditions

Also critical to accurate and relevant results are the choices of boundary conditions. These are outlined and justified in table 3.

Table 3: CFD boundary conditions

Boundary	Condition	Justification
Inlet	Freestream	For supersonic and hypersonic simulations, the 'freestream' boundary is widely recommended instead of the 'velocity inlet' which is known to perform poorly for such simulations [30].
Axis	Axis	This provides an axis of revolution for the axisymmetric physics model.
Outlet	Pressure Outlet	This is the standard StarCCM+ outflow boundary and also recommended in [30]. Note that gauge pressure is set to zero to improve the accuracy of pressure calculations [27].
Wall	Wall (no-slip)	A no-slip wall best models the surface of the rocket; Kn$<$0.03 means slip conditions can be neglected [11]. The wall is modelled as adiabatic for simplicity.

3.2 Adaptive Mesh Refinement

Since all test cases (table 1) have Mach numbers above 1, shocks and expansions are expected. Accurately modelling shocks in finite volume codes is a great challenge and an active field of research [18]. For finite volume codes, a popular method is 'Adaptive Mesh Refinement' (AMR), which iteratively refines and coarsens the mesh size based on a specified function in order to ensure the mesh is sufficiently small near shocks to capture the flow behaviour, but not so small everywhere that the simulations become prohibitively expensive.

StarCCM+ recently implemented built-in AMR for 3D simulations [31], but it does not yet support this for 2D meshes. Therefore, AMR was implemented manually using a custom field function, table, and StarCCM+ macro using the method outlined in [30]. Refinement was conducted based on the field functions given in listing 1. It's important to note that, while cell size can usually be found using the cube root of volume, for an axisymmetric model the cell size is a function of radial coordinate.

```
$PressureGradMag = mag(grad($Pressure))

$CellSize = ${RadialCoordinate}>0?sqrt($Volume/(${RadialCoordinate})):$BaseSize

$RefinementType = $PressureGradMag>$RefinementLimit?1:($PressureGradMag<$CoarsenLimit?-1:0)

$Refinement = min(max($PressureGradMag>$RefinementLimit?0.75*${CellSize}:($PressureGradMag<$CoarsenLimit?2*
 ${CellSize}:${CellSize}), $MinCellSize), 100*$BaseSize)
```

Listing 1: AMR field functions

The `Refinement` field function is fed into a user-defined table in StarCCM+ which is used for 'table-based refinement' in the mesh generation parameters. The macro given in appendix B.1 is then implemented to automatically execute several short sets of solution iterations of execution and mesh refinement, before a longer execution is conducted with the final, refined mesh.

3.3 Verification and Validation

3.3.1 Verification

The AIAA defines verification as *'The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model.'* [32] This was achieved by conducting a mesh convergence study and monitoring key quantities during execution.

A mesh convergence study was conducted on a blunted conic nose at Mach 4. The minimum cell size was set to $20\,\%$ of the base size which was varied during the study. Figure 6 illustrates the results for the pressure drag coefficient and nose tip temperature, with the final $25\,\mathrm{mm}$ base size marked. Originally, a $50\,\mathrm{mm}$ base size was planned, which was reduced following this study to properly predict C_D.

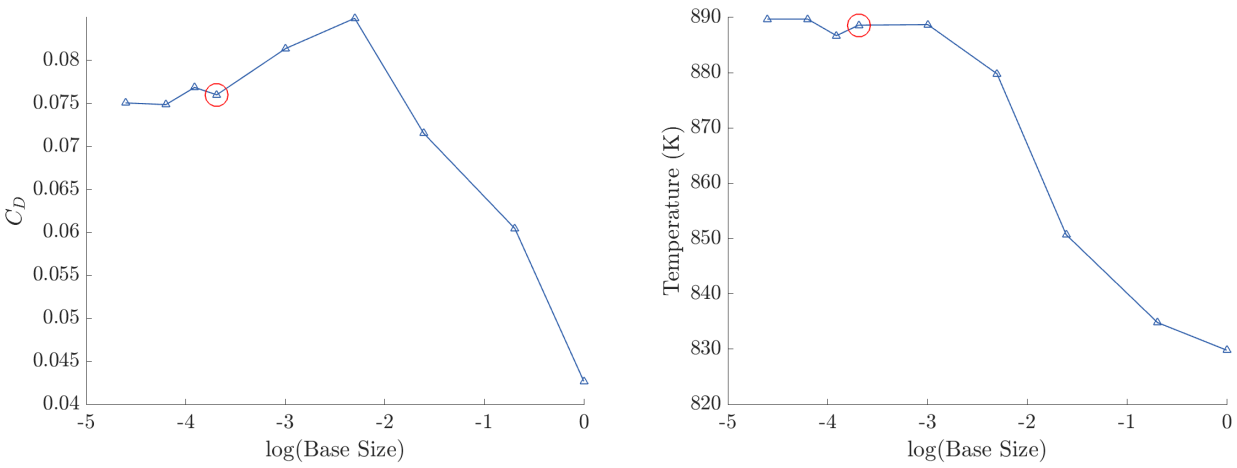


Figure 6: Mesh convergence study

Verification is also achieved by monitoring residuals, C_D, and temperature across all iterations. The AMR script in section B.1 implements automatic stopping criteria for its final run which enforces asymptotic behaviour of C_D and the normalised energy residual, which is also required to fall below 10^{-4}. While an even lower residual value would be desirable, the steady-state value of the energy residual sometimes did not fall far lower than this (usually for Mach 7 test cases). In the case where residuals fail to meet these criteria, a 1500-iteration fixed steps criteria prevented excessively long runs and monitors were checked manually to verify convergence.

3.3.2 Validation

The AIAA defines validation as *'The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.'* [32] This is best achieved by comparing CFD results to experimental data and analytical calculations where available.

The forebody pressure drag coefficient was extracted from CFD results and compared to several sources of experimental data from the USA [4, 5, 6, 7] and the UK [3]. The experiments took place largely in supersonic/ hypersonic wind tunnels, and all measured the forebody drag by subtracting a base drag value from the total measured drag. It is important to note that the experiments did not test noses with a body tube attached but they did all take place at similar Reynolds numbers $\mathcal{O}(10^6)$. The Mach 7 CFD case was at a slightly lower Reynolds number of 7.3×10^5 due to the lower density, which could explain the slight discrepancy for those values.

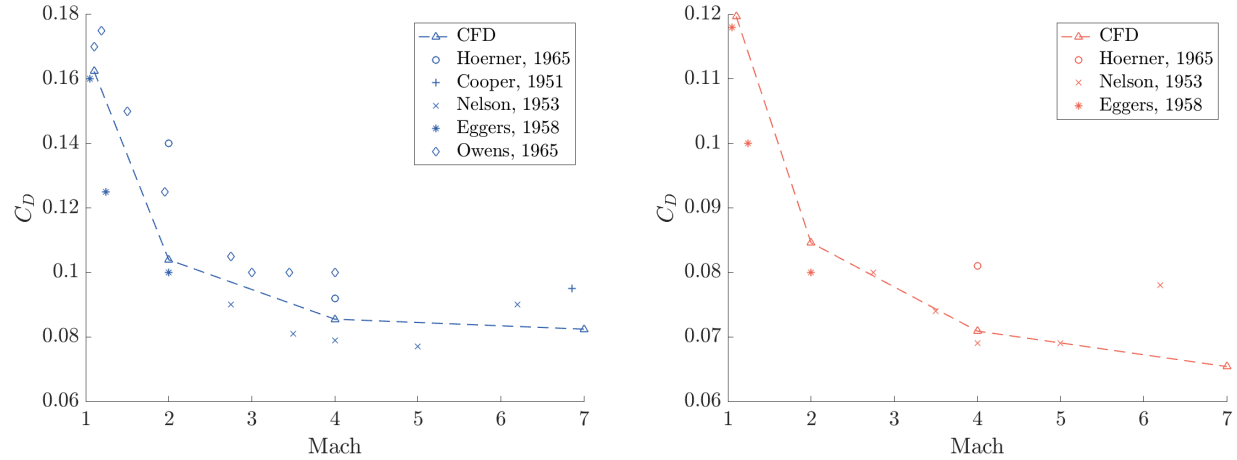


Figure 7: Drag coefficient validation [3, 4, 5, 6, 7]

Further validation was conducted by comparing the maximum air temperature for blunt nose test cases with the analytical stagnation temperature. Since CFD should provide 'perfect' numerical solutions, one would expect these numbers to become equal as the mesh size becomes small. Table 7 in appendix C.2 shows how these comparisons further validate CFD results to a very high level of accuracy (around $0.5\,\%$).

Finally, the `localHeatTransferCoefficient` and `localHeatTransferReferenceTemperature` fields were extracted from StarCCM+ and used to predict the rate of heat flux $\dot{q}$ at the nose tip assuming an initial wall temperature of $15\,^{\circ}\mathrm{C}$, using equation 7 [27]. Using the Stefan-Boltzmann equation it was possible to find the equilibrium wall temperature based on CFD data. Figure 8 illustrates a comparison between this value and three analytical methods for predicting wall temperature from section 2.1. This demonstrates consistency between analytical and computational methods, further validating those results.

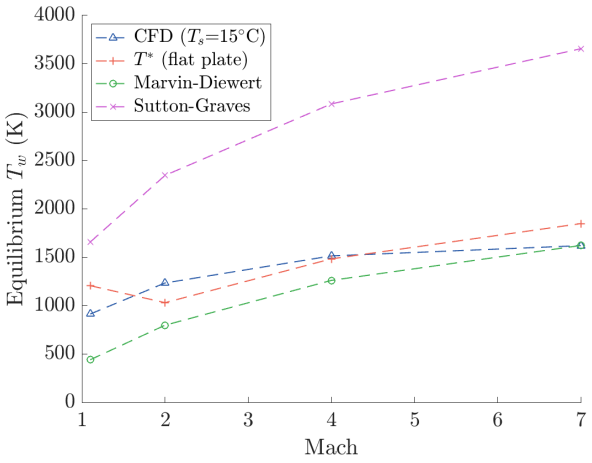


Figure 8: Equilibrium wall temperature validation

3.4 Results

Having verified and validated the CFD model, a series of tests were executed based on the requirements of the launcher structural design team. They created noses of different shapes, fineness ratios, and bluntness ratios [2]. The results of these tests are illustrated in figures 9 and 10. These results informed the selection of a $\frac{3}{4}$-Power series nose with a fineness ratio of 3 and bluntness ratio less than $10\,\%$ [2]. The pressure distribution (see section C.2) and peak temperatures for the selected nose were also shared with the design team for material selection [33].

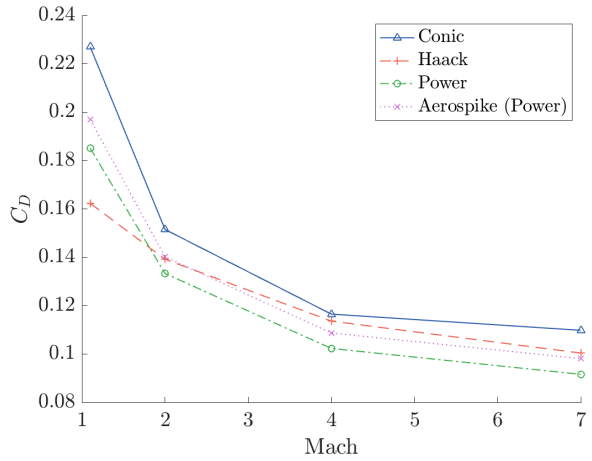
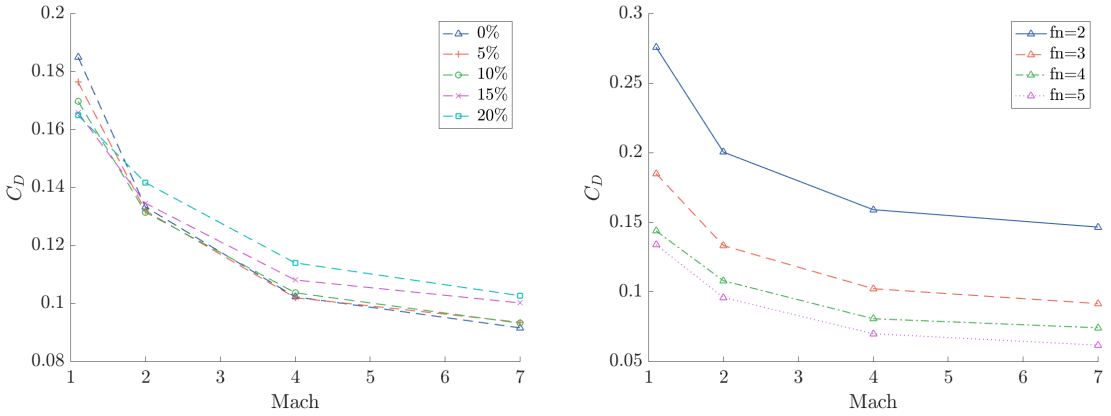


Figure 9: Effect of nose shape on drag coefficient



(a) Varying bluntness ratio

(b) Varying fineness ratio

Figure 10: Further CFD results; $\frac{3}{4}$-power series nose

4 Conclusions

This report has achieved its aims by delivering accurate and relevant information on the performance of different rocket nosecones. A power-series nose with a fineness ratio of 3 was selected [2] by the launcher structures team. The objectives outlined in section 1.2 have been reached, with both analytical and computational means used to profile the performance of various nose concepts.

While analytical analysis fell short in determining drag, the report acknowledges the possibility of exploring more intricate analytical approaches, as discussed in section 2.2. However, due to the complexity involved, a CFD analysis proved to be a more suitable engineering solution, leveraging the computational power of modern computers. On the other hand, temperature estimation proved to be more manageable using analytical methods. The lack of analyses relevant to sharp nosecones is disappointing, but the Marvin-Diewert method is believed to be reasonably accurate. The method is certainly superior to conservative approaches such as stagnation temperature since it predicted a peak heating much closer to that attained from CFD analysis.

CFD analysis has also been deemed successful, with an excellent validation outcome. The robustness of results has been demonstrated, and computational efficiency maximised by leveraging axial symmetry and AMR. These results can therefore inform the nose design with a good degree of confidence.

In addition to the core research, the author also dedicated time to creating a video animation of the mission. Blender animation software was utilized, and the rendering process was accelerated by executing it on high-performance computing servers. For compiling various renders into a video suitable for presentation to industry experts, Davinci Resolve was employed.

5 Future Improvements

Experimental testing could further validate analytical heating calculations, though this would come at a high cost. CFD simulations could be 'improved' no end, although mostly with little benefit and significant cost. One more valuable improvement would be implementing a 3D model for predicting drag and lateral forces at various small angles of attack, which would be useful in trajectory simulations developed by the mission design team [9]. Surface roughness and the effects of pressure-tapping holes could also be considered in the future since these are likely to significantly impact the aerodynamic performance of the nose. Additionally, some of the methodologies in this report could be extended to enable simulation of the whole rocket. It may be of particular value to lengthen the domain to include the aftbody and wake, which will contribute significantly to drag.

While the improvements discussed above may be of some value, greater mission value can be attained in the future by applying the tools used in this report to a detailed analysis of the fins designed by the FAD02 [10]. These have not been analysed with the same rigour as the nose, so there is likely greater scope for drag (and heating) reduction here. Since the fins are not axisymmetric, different analytical methods would be required and 2D CFD simulations of the fin cross-section could be conducted. Such CFD simulations could still leverage many of the features used for the nosecone analysis, including AMR and the selected turbulence models. Depending on the choice of computational domain, modelling of inflow conditions is likely to be one of the most significant challenges for such a simulation.

References

[1] Robert V Owens. Aerodynamic characteristics of spherically blunted cones at mach numbers from 0.5 to 5.0. NASA Technical Note TN D-3088, George C. Marshall Space Flight Centre, Huntsville, Alabama, December 1965.

[2] Myka Abaquin. Spacehaven: Launch vehicle nose cone and payload deployment design. Group project report, Imperial College London, 2023.

[3] Sighard F Hoerner. *Fluid-Dynamic Drag*. Hoerner Fluid Dynamics, Bakersfield, CA 93390, 1992. Library of Congress Catalog Card Number 64-19666.

[4] Ralph D Cooper and Raymond A Robinson. An investigation of the aerodynamic characteristics of a series of cone-cylinder configurations at a mach number of 6.86. NACA Research Memorandum RM L51J09, Langley Aeronautical Laboratory, Langley Field, Virginia, December 17 1951. Declassified October 31, 1958.

[5] A J Eggers, M Resnikoff, and David H Dennis. Bodies of revolution having minimum drag at high supersonic speeds. NACA Report 1306, Ames National Laboratory, Moffett Field, California, January 1958.

[6] Robert L Nelson and William E Stoney. Pressure drag of bodies at mach numbers up to 2.0. NACA Research Memorandum RM L53I22c, Langley Aeronautical Laboratory, Langley Field, Virginia, November 1953.

[7] Robert V Owens. Aerodynamic characteristics of spherically blunted cones at mach numbers from 0.5 to 5.0. NASA Technical Note TN D-3088, George C. Marshall Space Flight Center, Huntsville, Alabama, December 1965.

[8] National Aeronautics and Space Administration. NASA technology taxonomy. `https://www.nasa.gov/sites/default/files/atoms/files/2020_nasa_technology_taxonomy_lowres.pdf`, 2020. Accessed June 14, 2023.

[9] Usmaan Yaqoob. Space haven: Demonstrator re-entry trajectory planning and optimisation. Group project report, Imperial College London, 2023.

[10] William Taylor. Space haven: Launcher static stability and control authority. Group project report, Imperial College London, 2023.

[11] John D Anderson Jr. *Hypersonic and High-Temperature Gas Dynamics*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston, Virginia, 3rd edition, 2019. ISBN 9781624105142.

[12] Paul Bruce. Aerothermodynamics of launch and re-entry vehicles. Imperial College London, 2021. Autumn Term.

[13] Paul Bruce. Aerothermodynamics of launch and re-entry vehicles. Imperial College London, 2021. Autumn Term.

[14] Jose Augusto Mazzoni, Jose Bezerra Pessoa Filho, and Humberto Araujo Machado. Aerodynamic heating on vsb-30 sounding rocket. In *Proceedings of the 18th International Congress of Mechanical Engineering*, Ouro Preto, November 2005.

[15] Michael E Tauber. A review of high speed, convective, heat transfer computation methods. NASA Technical Paper NASA-TP-2914, Ames Research Center, Moffett Field, California, 1989.

[16] Josh Goldspink. Space haven: Trajectory design, atmospheric modelling and uncertainty analysis. Group project report, Imperial College London, 2023.

[17] J.W. Maccoll and G.I. Taylor. The air pressure over a cone moving at high speeds. *Proceedings of the Royal Society*, 139:278–311, 1933. doi: 10.1098/rspa.1933.0017.

[18] Russel M. Cummings, William H. Mason, Scott A. Morton, and David R. McDaniel. *Applied Computational Aerodynamics*. Cambridge Aerospace Series. Cambridge University Press, New York, 2015. ISBN 978-1-107-05374-8.

[19] Mr CFD. Y+ and Grid Types. `https://www.mr-cfd.com/y-and-grid-types/#:~:text=The%20mesh%20quality%20determines%20the,wedge%2C%20pyramid%2C%20and%20polyhedral`, 2023. Accessed June 5, 2023. Falmouth, UK.

[20] Georgios Balafas. Polyhedral mesh generation for CFD-analysis of complex structures. Technical report, Faculty of Civil Engineering and Geodesy, Technische Universität München, Munich, October 2014.

[21] Symscape. Polyhedral, tetrahedral, and hexahedral mesh comparison. https://www.symscape.com/polyhedral-tetrahedral-hexahedral-mesh-comparison, February 25 2013. Accessed June 14, 2023.

[22] Aidan Wimhurst. Fluid mechanics 101 calculators. https://www.fluidmechanics101.com/pages/tools.html, 2021. Accessed June 14, 2023.

[23] Florian R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32:1598–1605, 1994.

[24] Florian R Menter, M Kuntz, and RB Langtry. Ten years of experience with the sst turbulence model. In *AIAA Paper 2003-4031*, 2003.

[25] Junji Huang, Jorge-Valentino Bretzke, and Lian Duan. Assessment of turbulence models in a hypersonic cold-wall turbulent boundary layer. *Fluids*, 4(1):37, 2019. doi: 10.3390/fluids4010037.

[26] Aidan Wimhurst. The k - omega SST turbulence model. https://www.youtube.com/watch?v=myv-ityFnS4, March 2019. Accessed June 5, 2023.

[27] Siemens PLM Software. *StarCCM+ User Guide*, 2019.2 edition, 2019.

[28] Joel Guerrero. Turbulence and CFD models: Theory and applications. http://www.dicat.unige.it/guerrero/turbulence2020/slides/4practical_/estimates.pdf, 2020. Accessed June 14, 2023.

[29] Aidan Wimhurst. The transition SST ($\gamma - re_\theta$) model. https://www.youtube.com/watch?v=5htknS9uVEk, 2019. Accessed June 14, 2023.

[30] Peter G Cross and Michael R West. Simulation of hypersonic flowfields using star-ccm+. Technical Report NAWCWD TM 8824, Naval Air Warfare Center Weapons Division, China Lake, California, January 2019.

[31] Clay Hearn. Adaptive mesh refinement in simcenter star-ccm+. https://appliedcax.com/resources/star-ccm-resources/adaptive-mesh-refinement-in-simcenter-star-ccm, August 2021. Accessed June 14, 2023.

[32] American Institute of Aeronautics and Astronautics. *Guide For The Verification And Validation Of Computational Fluid Dynamics*. American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Drive, Reston, VA 20191-4344 USA, 1998.

[33] Julian Ting. Space haven: Launch vehicle material selection and airframe design. Group project report, Imperial College London, 2023.

Appendices

A Analytical Calculations Script

A.1 Main Script

```
% Foldable Aeroshell Demonstrator | GDP
% Launcher aerodynamic calculations
% Nathanael Jenkins, 2023

clear
load('Trajectory.mat')

%% Parameters
% Trajectory Data
dataFreq = 10;
M = transpose(out.mach.Data(1:dataFreq:end));
alt = transpose(out.altitude.Data(1:dataFreq:end));
time = transpose(out.tout(1:dataFreq:end));
[~, indx] = max(alt);
M = M(1:indx);
alt = alt(1:indx);
time = time(1:indx);

% Location
launchLat = 35.0568; % deg N (Mojave)
launchLong = -118.1578; % deg E (Mojave)

% Nose
R_N = 0.01; % (m)
eps = 0.2; % Al
RBody = 0.325;
fnConic = 3; % Fineness ratio

% Enable plotting
plotOn = 0;

%% Constants
% Air
gma = 1.4;
c_p = 1005;
R = 287;
molMass = 28.96*10^-3;
% Universal constants
sigma = 5.6704*10^(-8);
Rbar = 8.314;

%% Parameter calculations
% NRLMSISE-00 atmosphere model; @ Null Island, with default F107A, F107, APH
[tmp, tmp2] = atmosnrlmsise00(alt, 45, 45, 2023, 1, 0, 150, 150, 4);
T = tmp(:, 2)'; % Altitude freestream temperature, K
rho = tmp2(:, 6)'; % Total mass density (kg/m3)
% Freestream
c = sqrt(gma*R*T); % Speed of sound (m/s)
V = M.*c;
% Knudsen number
mu = 1.458*10^(-6)*T.^1.5./(T+110.4);
nu = mu./rho;
Kn = nu./(sqrt(8*Rbar*T./(pi*molMass)));
% Reynolds Number
Re = rho.*V.*RBody./mu;
% Geometric
CSA = pi*RBody^2;

%% Nosecone heating
% LEVEL 1 (ALL REGIMES)
% Most conservative - stagnation temperature
T0 = stagTemp(M, T, gma);

% Also conservative - adiabatic wall temp'
```

```
% Assumes turbulent BL
% (Turbulent coefficient hotter, therefore conservative)
T2 = T.*((2*gma*M.^2-(gma-1)).*((gma-1).*M.^2+2)./((gma+1)^2.*M.^2));
M2 = sqrt(((gma-1).*M.^2+2)./(2*gma*M.^2-(gma-1)));
V2 = M2.*sqrt(gma*R.*T2);
Taw = (c_p.*T2+0.89*0.5*V2.^2)/c_p;
Taw(M<1) = NaN;

% LEVEL 2 (HYPERSONIC REGIMES, FLAT PLATE)
% SUTTON-GRAVES, 1971
% Only considers convective heating
[TwSG, qdot_SG] = suttonGraves(rho, R_N, V, eps, sigma);
TwSG(M<1) = NaN;
% Transient effects (small)
TintSG = zeros(1, length(qdot_SG));
TintSG(1) = T(1);
qInt = zeros(1, length(qdot_SG));
qInt(1) = 0;
for i=2:length(qdot_SG)
    qInt(i) = qdot_SG(i-1) - eps*sigma*TintSG(i-1)^4;
    TintSG(i) = TintSG(i-1)+(time(i)-time(i-1))*qInt(i)/c_p;
end

% REF TEMP METHOD
Tawtmp = Taw;
Tawtmp(isnan(Tawtmp)) = T(1); % Prevent NaNs
refT.Tw = zeros(1, length(Tawtmp));
refT.Tw(1) = T(1);
refT.qc = zeros(1, length(Tawtmp));
refT.qc(1) = 0;
convCoeff = zeros(1, length(Tawtmp));
convCoeff(1) = 0;
rho2 = rho.*((gma+1)*M.^2./((gma-1)*M.^2+2)); % Normal shock relation
V2(M<1) = V(M<1);
refT.Treal = zeros(1, length(Tawtmp));
refT.Treal(1) = T(1);
refT.qreal = zeros(1, length(Tawtmp));
for i=2:length(Tawtmp)
%     TStar = T(i-1)*(1+0.032*M(i-1)^2+0.58*((refT.Tw(i-1)/T(i-1))-1)); % Anderson
    TStar = T0(i-1)*(0.5*((refT.Tw(i-1)/T0(i-1))+1)+0.16*0.89*((gma-1)/2)*M2(i-1)^2); % Meador-Smart adaptation
     ; Turbulent BL
    muStar = 1.458*10^(-6)*TStar.^1.5./(TStar+110.4); % Sutherland
    rhoStar = rho2(i-1)*((TStar/T2(i-1))^(1/(gma-1))); % Ideal gas relation
    Restar = RBody*V2(i-1)*rhoStar/muStar; % Anderson/ VSB-30
    if Restar == 0
        convCoeff(i) = 0; % Prevent NaNs
    else
        convCoeff(i) = 0.02296*rho2(i-1)*V2(i-1)*c_p*(Restar^-0.139);
    end
    refT.qc(i) = -real(convCoeff(i-1)*(refT.Tw(i-1)-Tawtmp(i-1)));
    refT.Tw(i) = refT.Tw(i-1)+(time(i)-time(i-1))*refT.qc(i)/c_p;
    qreal(i) = refT.qc(i-1) - eps*sigma*refT.Treal(i-1)^4;
    refT.Treal(i) = refT.Treal(i-1)+(time(i)-time(i-1))*qreal(i)/c_p;
end
refT.Tequilibrium = real((refT.qc./(eps*sigma)).^0.25);

% LEVEL 3 (HYEPRSONIC) NASA METHOD for SHARP CONE
deltaC = (pi/2)-atan(2*fnConic); % Half cone angle (rad)
% x=R_N (Asusming small R_N)
hwhaw = (c_p.*T+0.5*V.^2)./(c_p.*T+0.4.*V.^2);
SC.qdot = -real((4.03*10^(-5)).*sqrt(rho.*cos(deltaC)./R_N).*V.^3.2.*sin(deltaC).*(1-hwhaw));
SC.T = real((SC.qdot./(eps*sigma)).^0.25);

% Plotting
if plotOn
    plot(alt/1000, TwSG-273.15, alt/1000, Taw-273.15, alt/1000, T0-273.15, alt/1000, T-273.15)
    legend('Sutton-Graves $T_w$', '$T_{aw}$', '$T_0$', '$T_\infty$', 'Interpreter', 'Latex')
    title('Nosecone heat estimates (for all continua altitudes)')
    xlabel('Time (s)')
    ylabel('Temperature ($^\circ$c)')
    nonContAlts = alt(Kn>=0.2); % Kn=0.2 => NS invalid (Anderson)
    xlim([0 nonContAlts(1)/1000])
end
```

```
137
138 %% Nosecone drag
139 % Experimental data fit for conic nose
140 CdC = noseDrag(M);
141 PowC = 0.5*rho.*V.^3.*CdC.*CSA;
142 CdC(Kn>=0.2) = 0; % Adjustment for non-continuum flow
143 EnrgC = trapz(alt, CdC.*0.5.*rho.*V.^2*CSA);
144 CdC(Kn>=0.2) = NaN;
145
146 %% Nosecone CFD Conditions
147 M_target = [1.1, 2, 4, 7]; % Selected using Mach-DragPower curve
148 Mcontinua = M(Kn<0.2);
149 [~, indx] = max(Mcontinua);
150 Minterp = Mcontinua(1:indx);
151 q = 0.5*rho.*V.^2;
152 q_CFD = interp1(Minterp, q(1:indx), M_target);
153 V_CFD = interp1(Minterp, V(1:indx), M_target);
154 rho_CFD = interp1(Minterp, rho(1:indx), M_target);
155 alt_CFD = interp1(Minterp, alt(1:indx), M_target);
156 T_CFD = interp1(Minterp, T(1:indx), M_target);
157 CdC_CFD = interp1(Minterp, CdC(1:indx), M_target);
158 disp('')
159 disp('========================================')
160 disp('          NOSECONE CFD PARAMETERS')
161 disp('========================================')
162 disp('    M    rho      T       V     alt    CdC')
163 for i=1:length(M_target)
164     fprintf('%4.3f  %4.3f  %4.1f  %6.1f  %6.0f  %5.2f\n', M_target(i), rho_CFD(i), T_CFD(i),  V_CFD(i), alt_CFD
      (i), CdC_CFD(i))
165 end
```

Listing 2: Initial analysis of heating and drag on a rocket nosecone

A.2 Stagnation Temperature

```
1 function T0=stagTemp(M, T, gma)
2     % T0=stagTemp(M, T, gma)
3     % Calculate fluid stagnation temperature(s)
4     T0 = T.*(1+((gma-1)/2)*(M.^2));
5 end
```

Listing 3: StagTemp Matlab function

A.3 Sutton-Graves Temperature Estimation

```
1 function [Tw, qdot_conv] = suttonGraves(rho, R_N, V, eps, sigma)
2     % [Tw, qdot_conv] = suttonGraves(M, rho, R_N, V, eps, sigma)
3     % Calculate peak wall temperature based on Sutton-Graves relation
4     % Assumes earth atmosphere; hypersonic mach numbers
5     qdot_conv = (1.74*10^-4)*sqrt(rho./R_N).*(V.^3);
6     % Solve Stefan-Boltzmann (radiative) relation
7     Tw = (qdot_conv./(eps*sigma)).^0.25;
8 end
```

Listing 4: SuttonGraves Matlab function

A.4 Empirical Drag Model

```
1 function CD = noseDrag(M)
2
3     Mfit = [0.048, 0.48, 0.65, 0.85, 0.95, 1.1, 1.19, 1.5, 1.95, 2.75, 3.0, 3.45, 4.0, 4.45];
4     Dfit = [0.02, 0.03, 0.04, 0.09, 0.12, 0.17, 0.175, 0.15, 0.125, 0.105, 0.10, 0.1, 0.1, 0.1];
5
6     CD = interp1(Mfit, Dfit, M, 'linear', 'extrap');
7
8 end
```

Listing 5: NoseDrag Matlab function

B StarCCM+ Scripts

B.1 StarCCM+ Macro

```
// Simcenter STAR-CCM+ macro: RunAMR.java
// Nathanael Jenkins, 2023
// FAD GDP Adaptive Mesh Refinement (Nosecone)
package macro;

import java.util.*;

import star.common.*;
import star.base.neo.*;
import star.twodmesher.*;
import star.meshing.*;
import star.base.report.*;
import star.flow.*;
import star.vis.*;

public class RunAMR extends StarMacro {

  public void execute() {
    execute0();
  }

  private void execute0() {

    /* Initialisation */

    Simulation simulation_0 =
      getActiveSimulation();

    Solution solution_0 =
      simulation_0.getSolution();

    solution_0.clearSolution(Solution.Clear.History, Solution.Clear.Fields, Solution.Clear.LagrangianDem);

    /* Java variables */

    AutoMeshOperation2d autoMeshOperation2d_0 =
      ((AutoMeshOperation2d) simulation_0.get(MeshOperationManager.class).getObject("Automated Mesh (2D)"));

    DualAutoMesher2d dualAutoMesher2d_0 =
      ((DualAutoMesher2d) autoMeshOperation2d_0.getMeshers().getObject("Polygonal Mesher"));

    MeshPipelineController meshPipelineController_0 =
      simulation_0.get(MeshPipelineController.class);

    XyzInternalTable xyzInternalTable_0 =
      ((XyzInternalTable) simulation_0.getTableManager().getTable("Refinement"));

    // Drag stopping criteria
    MonitorIterationStoppingCriterion monitorIterationStoppingCriterion_1 =
      ((MonitorIterationStoppingCriterion) simulation_0.getSolverStoppingCriterionManager().
     getSolverStoppingCriterion("CD Monitor Criterion"));

    MonitorIterationStoppingCriterionAsymptoticType monitorIterationStoppingCriterionAsymptoticType_1 =
      ((MonitorIterationStoppingCriterionAsymptoticType) monitorIterationStoppingCriterion_1.getCriterionType()
     );

    // Energy stopping criteria
    MonitorIterationStoppingCriterion monitorIterationStoppingCriterion_0 =
      ((MonitorIterationStoppingCriterion) simulation_0.getSolverStoppingCriterionManager().
     getSolverStoppingCriterion("Energy Criterion"));

    MonitorIterationStoppingCriterionAsymptoticType monitorIterationStoppingCriterionAsymptoticType_0 =
      ((MonitorIterationStoppingCriterionAsymptoticType) monitorIterationStoppingCriterion_0.getCriterionType()
     );

    // Fixed steps stopping criteria
    FixedStepsStoppingCriterion fixedStepsStoppingCriterion_0 =
```

```
    ((FixedStepsStoppingCriterion) simulation_0.getSolverStoppingCriterionManager().
  getSolverStoppingCriterion("Fixed Steps"));

  IntegerValue integerValue_0 =
    fixedStepsStoppingCriterion_0.getFixedStepsObject();

  integerValue_0.getQuantity().setValue(50.0);

  // Min Cell Size
  ScalarGlobalParameter scalarGlobalParameter_2 =
    ((ScalarGlobalParameter) simulation_0.get(GlobalParameterManager.class).getObject("MinCellSize"));

  scalarGlobalParameter_2.getQuantity().setValue(0.01); // Default value

  ScalarGlobalParameter scalarGlobalParameter_3 =
    ((ScalarGlobalParameter) simulation_0.get(GlobalParameterManager.class).getObject("BaseSize"));

  scalarGlobalParameter_3.getQuantity().setValue(0.025); // Default value

  /* First run (default mesh) */

  monitorIterationStoppingCriterionAsymptoticType_1.getMaxWidth().setValue(1.0E-3);
  monitorIterationStoppingCriterionAsymptoticType_0.getMaxWidth().setValue(1.0E-3);

  dualAutoMesher2d_0.setFieldFunctionRefinementObject(null);

  meshPipelineController_0.generateVolumeMesh();

  dualAutoMesher2d_0.setFieldFunctionRefinementObject(xyzInternalTable_0);

  solution_0.initializeSolution();

  simulation_0.getSimulationIterator().run();

  /* 1st stage mesh refinement */

  for (int i=0; i<19; i++) {
    xyzInternalTable_0.extract();

    meshPipelineController_0.generateVolumeMesh();

    simulation_0.getSimulationIterator().run();
  }

  /* 2nd stage mesh refinement */
  monitorIterationStoppingCriterionAsymptoticType_1.getMaxWidth().setValue(1.0E-3);
  monitorIterationStoppingCriterionAsymptoticType_0.getMaxWidth().setValue(1.0E-3);
  integerValue_0.getQuantity().setValue(100.0);

  for (int i=0; i<2; i++) {
    xyzInternalTable_0.extract();

    meshPipelineController_0.generateVolumeMesh();

    simulation_0.getSimulationIterator().run();
  }

  /* Execute final run */
  monitorIterationStoppingCriterionAsymptoticType_1.getMaxWidth().setValue(1.0E-4);
  monitorIterationStoppingCriterionAsymptoticType_0.getMaxWidth().setValue(1.0E-4);
  integerValue_0.getQuantity().setValue(1500.0);

  xyzInternalTable_0.extract();

  meshPipelineController_0.generateVolumeMesh();

  simulation_0.getSimulationIterator().run();

  /* Run reports */
  ForceCoefficientReport forceCoefficientReport_0 =
    ((ForceCoefficientReport) simulation_0.getReportManager().getReport("CD"));
  forceCoefficientReport_0.printReport();
  MaxReport maxReport_0 =
```

```
      ((MaxReport) simulation_0.getReportManager().getReport("MaxWallTemp"));
    maxReport_0.printReport();
    }
}
```

Listing 6: AMR macro

B.2 HPC Batch Submission Script

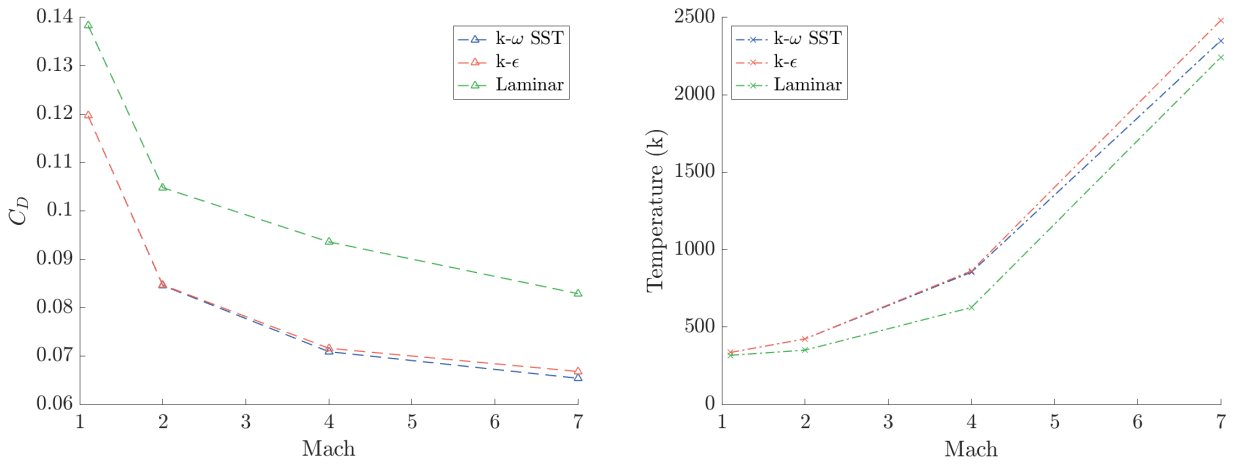
```
#!/bin/bash

# Batch script for submission of StarCCM+ simulation jobs
# Nathanael Jenkins | 2023

#SBATCH --nodes=1
#SBATCH --ntasks=2              # Threads allocated to job (0-40 recommended)
#SBATCH --mem=16000MB           # Memory allocated to job (~32-64 GB recommended)
#SBATCH --time=1:00:00          # Job time limit (each partition comes with a hard limit)
#SBATCH --job-name=<job>        # Short job name for reference in the queue
#SBATCH --partition=short       # Partition (small, medium, large)
#SBATCH --output=<job>.out      # Logfile name

module load star-ccm+
# Make sure to specify -np equal to ntasks above to make full use of available computational resources
# Make sure to use appropriate PoD key and filename below
starccm+ -power -podkey <podKey> -licpath 1999@flex.cd-adapco.com -np 2 "$SLURM_SUBMIT_DIR/<filename>.sim" -
    batch "$SLURM_SUBMIT_DIR/RunAMR.java"
```

Listing 7: Batch submission script

C CFD Data

C.1 Turbulence Model Study

The test cases outlined in table 1 experience Reynolds numbers between 8.7×10^6 and 7.3×10^5, which is sufficiently large to assume that flow is turbulent. This was validated by conducting CFD tests with three different turbulence models. The k-ϵ turbulence model was identified as a reasonable alternative to the k-ω SST model and was tested to confirm this. Additionally, a laminar model was tested to confirm whether it was reasonable to assume turbulence is of importance in this case. The results of these tests are outlined in figure 11, which clearly shows that a turbulence model is necessary to accurately predict flow behaviour. The k-ω SST and k-ϵ models are both adequate, although the k-ω SST model is more widely validated [23, 24, 25, 30] so was selected for this project. The laminar model does not accurately model drag or temperature for these flow conditions.



(a) Effect on C_D

(b) Effect on temperature

Figure 11: Turbulence model validation; power-series nosecone

C.2 Additional Data

All CFD data is collated in the tables below. The mesh refinement study was conducted for a conic nose at Mach 4, with minimum cell size set to $20\,\%$ of the base size. The fineness ratio is described using fn and the bluntness ratio using br. The 'blunted' nosecones have a nose radius of $5.2\,\mathrm{cm}$, corresponding to br=0.16. The pressure distribution over the selected nosecone at Mach 7 (figure 12) was shared with the launcher structures team [33] to enable accurate prediction of the forces on its surface for material selection.

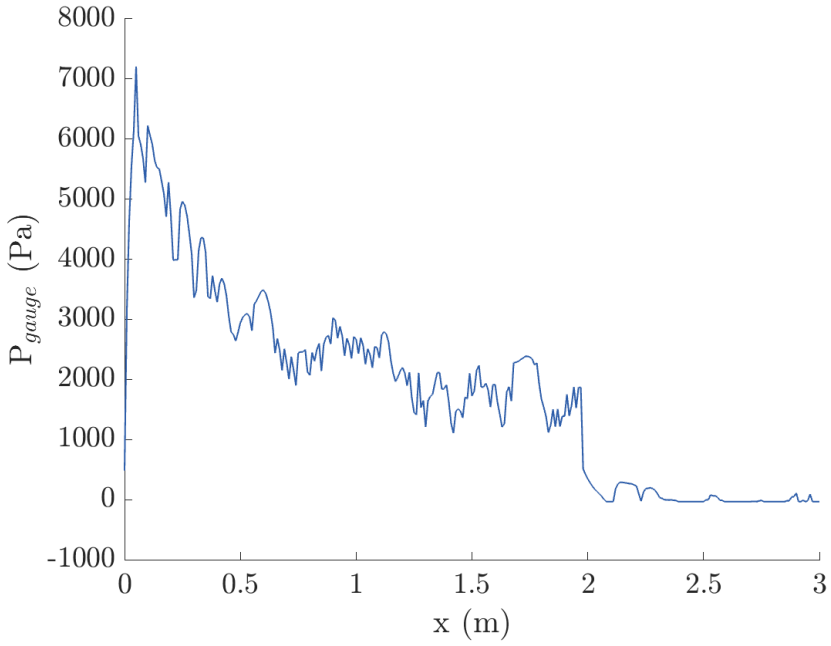


Figure 12: Pressure distribution over a 2m long power-series nose at Mach 7

Table 4: Mesh refinement study

Base Size (m)	**1.0**	**0.5**	**0.2**	**0.1**	**0.05**
C_D	0.04268	0.06045	0.07153	0.08490	0.08319
T_{max} (k)	829.8	834.8	850.7	879.8	888.7

Base Size (m)	**0.025**	**0.02**	**0.015**	**0.01**
C_D	0.0760	0.07686	0.07486	0.07506
T_{max} (k)	888.6	886.7	889.7	889.7

Table 5: Forebody drag coefficients

	$C_{D_{pressure}}$				$C_{D_{shear}}$			
Mach	1.1	2.0	4.0	7.0	1.1	2.0	4.0	7.0
Conic	0.1624	0.1039	0.0855	0.0824	0.0646	0.0476	0.0310	0.0274
Power ($\frac{3}{4}$)	0.1197	0.0846	0.0709	0.0654	0.0654	0.0487	0.0314	0.0262
Haack	0.0943	0.0892	0.0815	0.0744	0.0680	0.0500	0.0321	0.0260
Spike (Conic)	0.1587	0.1050	0.0840	0.0773	0.0594	0.0462	0.0300	0.0258
Spike (Power)	0.1329	0.0929	0.0780	0.0716	0.0641	0.0473	0.0307	0.0265
Conic, blunted	0.1315	0.0972	0.0837	0.0750	0.0618	0.0466	0.0289	0.0235
Power, blunted	0.1264	0.1004	0.0887	0.0784	0.0654	0.0489	0.0307	0.0257
Haack, blunted	0.0946	0.0947	0.0880	0.0848	0.0631	0.0484	0.0304	0.0235
Power, fn=2	0.2138	0.1549	0.1296	0.1214	0.0622	0.0456	0.0295	0.0252
Power, fn=4	0.0745	0.0563	0.0476	0.0457	0.0694	0.0517	0.0331	0.0285
Power, fn=5	0.0616	0.0427	0.0354	0.0332	0.0723	0.0533	0.0345	0.0285
Power, br=0.05	0.1104	0.0835	0.0708	0.0670	0.0660	0.0485	0.0312	0.0265
Power, br=0.10	0.1035	0.0835	0.0730	0.0681	0.0663	0.0480	0.0307	0.0251
Power, br=0.15	0.0998	0.0873	0.0783	0.0747	0.0661	0.0473	0.0298	0.0255
Power, br=0.20	0.0983	0.0948	0.0855	0.0792	0.0667	0.0469	0.0285	0.0235

Table 6: Forebody maximum wall temperatures (Kelvin)

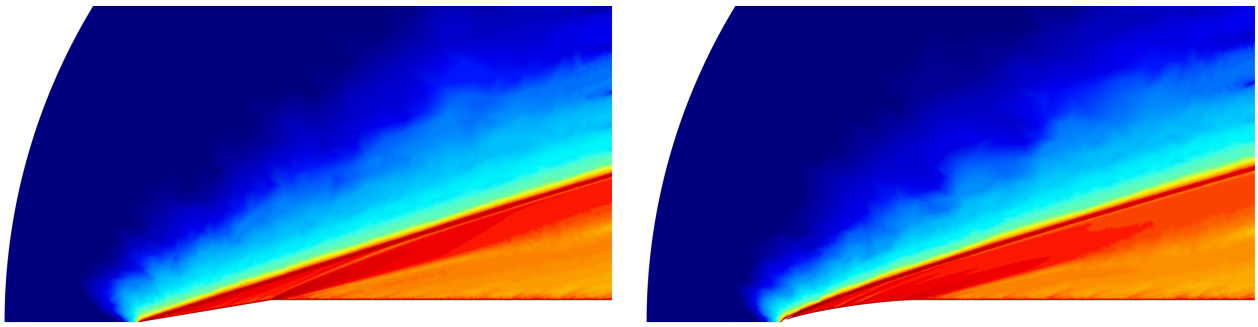
Mach	1.1	2.0	4.0	7.0
Conic	335.2	422.6	852.3	2343.5
Power ($\frac{3}{4}$)	335.4	423.2	853.5	2349.8
Haack	336.5	424.6	859.7	2364.2
Aerospike (Conic)	341.5	434.2	889.7	2428.8
Aerospike (Power)	339.4	434.2	887.3	2401.0
Conic, blunted	338.2	433.8	888.6	2420.0
Power, blunted	338.6	434.2	889.1	2444.9
Haack, blunted	338.3	434.0	888.5	2408.2
Power, fn=2	336.5	425.0	859.2	2370.0
Power, fn=4	335.3	422.7	854.5	2342.6
Power, fn=5	336.0	423.4	854.8	2349.4
Power, br=0.05	335.1	428.0	854.6	2348.6
Power, br=0.10	337.9	433.8	886.8	2400.0
Power, br=0.15	338.3	433.4	888.9	2427.1
Power, br=0.20	338.4	434.2	889.1	2437.9

Table 7: Stagnation temperature validation (Kelvin)

M	$T_{0_{analytical}}$	Conic	Power	Haack	**Mean error**
1.1	338.4	338.2	338.6	338.3	**0.05%**
2.0	433.9	433.8	434.2	434.0	**0.04%**
4.0	888.8	888.6	889.1	888.5	**0.03%**
7.0	2428.7	2420.0	2444.9	2408.2	**0.62%**

C.3 Visualisations

Visualisations were not of primary importance when analysing results, but they served an important purpose in understanding errors and mistakes, as well as providing some explanation for the differences between certain results. For the interest of the reader, some visualisations are shown below. Note that the pressure gradient plots are clipped at a certain height up the domain to save space; the domain does not have a flat horizontal top surface. In figure 13, dark blue represents a value of $0.0\,\mathrm{Pa\,m^{-1}}$, and red represents a gradient of $1.0 \times 10^6\,\mathrm{Pa\,m^{-1}}$.



(a) Conic

(b) Haack-series

(c) $\frac{3}{4}$ Power-series

(d) Aerospike (power-series)

Figure 13: Pressure gradient magnitude for various nosecone geometries at Mach 4.

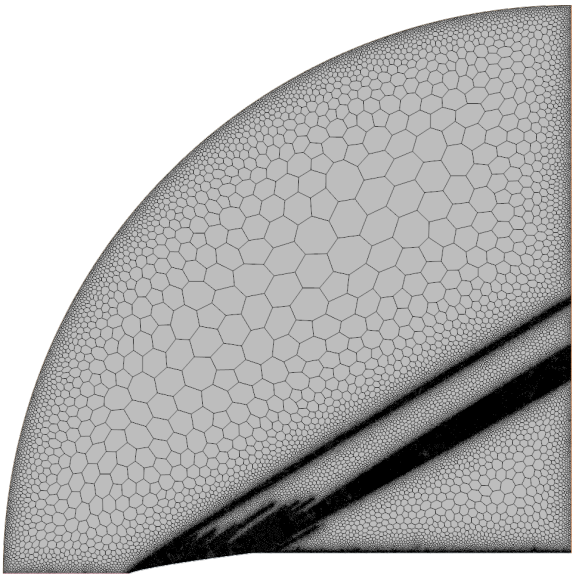


Figure 14: Refined mesh for a $\frac{3}{4}$ power series nose at Mach 2