# Computational Fluid Dynamics Analysis of TPS Tile Discontinuities During Atmospheric Reentry

a project presented to
The Faculty of the Department of Aerospace Engineering
San Jose State University

in partial fulfillment of the requirements for the degree
**Masters of Science in Aerospace Engineering**

by

**Javaneh Keikha**

December 2024

approved by

Dr. Periklis Papadopoulos
Faculty Advisor

ABSTRACT

**CFD Analysis of TPS Tile Discontinuities During Atmospheric Reentry**

Javaneh Keikha

For this report, a probabilistic method called DSMC is utilized to determine which configuration of the TPS tile spacing is optimal to reduce the aerodynamic heating during reentry of Earth's atmosphere. This method was developed in the 1960s by Graeme A. Bird to model the molecular properties of the flow. For the case of rarefied flow, where the density of the domain is low enough that the molecular collisions can no longer be ignored, this computational method is crucial. The TPS discontinuity is modeled in 3D as a cavity using the VSS model. There are three total cases for the cavity configuration: regular cavity with no modifications, a cavity with a rounded reattachment corner, and a cavity with an inclined reattachment corner. The altitude for all three cases is constant at 80 km, along with the freestream conditions. From running the simulation using SPARTA as a DSMC software of choice, given the buildup of pressure and density, rounding the reattachment corner is the best configuration to reduce the aerodynamic heating during reentry.

Acknowledgements

I would like to give my sincerest thanks to my dad, mom, brother, and my close family members for all the support and guidance they have given me throughout my life, as well as shaping me into the person I am today. I am also grateful for my friends that I have met during my master's program. They have given me the belief and motivation to succeed in my graduate studies. Also, I would like to acknowledge Dr. Yawo Ezunkpe for choosing me for the ISA position during the 2023 year. I have gained a lot of experience in decision-making, as well as in-depth knowledge of aerodynamics and propulsion. Additionally, I would like to express my gratitude to my faculty advisor, Dr. Periklis Papadopoulos, for his guidance and assistance whenever I needed his help for this project.

Table of Contents

List of Tables

List of Figures

List of Symbols

| Symbol | Definition | Units (SI) |
|---|---|---|
| A | Speed of sound | m/s |
| B | Closest approach for particle during undisturbed trajectories | m |
| C | Velocity of the particle | m/s |
| D | Diameter of molecule | m |
| D | Depth of the cavity | mm |
| E | Internal energy | J/kg |
| fnum | Ratio of real to simulated particles | --- |
| F | Density function of particles in phase space | --- |
| $k_b$ | Boltzmann constant | J/K |
| Kn | Knudsen Number | --- |
| L | Characteristic Length | m |
| L | Length of the cavity | mm |
| M | Molecular mass | kg |
| N | Number density of particles | --- |
| N | Number | - |
| P | Pressure | K |
| $S^2$ | Sphere of particle | --- |
| T | Time | S |
| T | Temperature | K |
| T | Time | s |
| V | Velocity of the flow | m/s |
| Subscripts: | | |
| Cav | Cavity | --- |
| Cell | Cell | --- |
| R | Relative | --- |
| Dom | Domain | --- |
| Par | Particles | --- |
| Ref | Reference | --- |

| | | |
|---|---|---|
| Real | Real | --- |
| Sim | Simulated | --- |
| I | Ionization | --- |
| W | Wall | --- |
| ∞ | Freestream | --- |
| I | Internal mode type | --- |
| Trans | Translational | --- |
| Rot | Rotational | --- |
| Vib | Vibrational | --- |
| El | Electronic | --- |
| Superscripts | | |
| ’ | After collision | --- |
| — | Average | --- |
| Greek Symbols: | | |
| Λ | Mean free path | m |
| Α | Diffusion coefficient | --- |
| Δ | Difference | --- |
| Ζ | Number of degrees of freedom | --- |
| Ω | Temperature dependence of Viscosity | --- |
| Θ | Characteristic temperature of the gas | K |
| Acronyms: | | |
| 2D | Two dimensional | --- |
| 3D | Three dimensional | --- |
| ASCII | American Standard Code for Information Interchange | --- |
| CAD | Computer Aided Design | --- |
| CFD | Computational Fluid Dynamics | |
| DSMC | Direct Simulation Monte Carlo | --- |
| HS | Hard Sphere | --- |
| MPI | Message Passing Interface | --- |
| SPARTA | Stochastic Parallel Rarefied-gas Time-accurate Analyzer | --- |
| STL | Stereolithography | --- |
| TCE | Total collision | --- |
| TPS | Thermal Protection System | --- |
| VHS | Variable Hard Sphere | --- |
| VSS | Variable Soft Sphere | --- |
| N2 | Nitrogen Gas | --- |
| O2 | Oxygen Gas | --- |

# 1. Introduction

## 1.1. Motivation

TPS has been used as an active thermal control method for spacecrafts during their mission. For certain spacecrafts such as NASA challenger, SpaceX Starship, and the upcoming Dream Chaser, their entire bodies are covered by thousands of TPS tiles. Generally, these tiles are arranged in a staggered pattern, where each of these tiles are spaced out, creating small gaps. All the TPS tiles are part of the heat shield for the spacecraft. This kind of thermal control is crucial, especially for atmospheric re-entry [1]. When this situation occurs, the spacecraft is entering atmosphere with speeds reaching as significantly large as Mach 30. At this point, the temperature reaches to an extremely high range. As a result, aerodynamic heating will affect the spacecraft; specifically, this type of heating will occur on the spacings between the TPS tiles. This will cause thermal expansion and contraction of these tiles, which will ultimately affect the performance of the TPS system. The goal of this project is to determine an optimal configuration to reduce the effects of the aerodynamic heating on the tile spacing.

## 1.2. Literature Review

This section is broken down into a couple of parts to understand the topic and decide an approach that will be best suited for this type of situation. The first subsection delivers background information about reentry flow and the second showcases the configurations previously done.

### 1.2.1. Atmospheric Reentry

Before diving into earlier works, understanding the atmosphere's condition during reentry is crucial to determine the simulation method. When the vehicle enters atmosphere, the temperatures grow significantly large. The flow goes through all the three regimes: continuum, slip, transition, and free molecular [2]. At a certain range of altitudes, the flow is considered to be rarefied [3][4]. Another name for it is low-density flow. In other words, the density of the medium is low enough that the motion of the particles cannot be neglected anymore [5]. For this reason, the flow has to be looked into at a microscopic level, so typical methods of CFD tools will create inaccurate results. This is where DSMC, which is a stochastic method of flow determination, will be useful since it considers the aero-thermal effects from the molecular properties.

### 1.2.2. TPS Tile Spacing Modeling

#### 1.2.2.1. Cavity Configurations

Previous works that have modeled the TPS tile discontinuity have created the model as a cavity [7][8][9][10][11][12]. This makes sense to model as a cavity due to how the tiles are generally spaced out. As seen in Fig. 1.1, the tiles for the space shuttle are placed together on the surface with thin gaps in between each other. Another example can be seen in Fig. 1.2 to better visualize the geometry.

Fig. 1.1: Space shuttle TPS tiles [6]

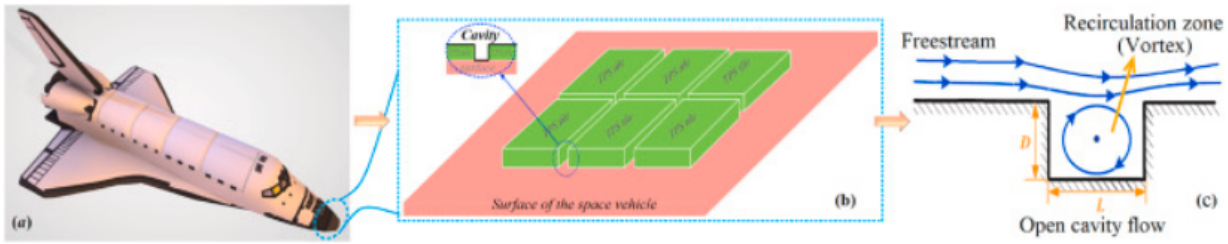



Fig. 1.2: Space shuttle with cavity flow diagram [10]

In the journal article [7] by Jin et al., a configuration used was rounding the corners as seen in figure below. The L/D ratio was kept as a constant with L = D = 3 cm, while investigating about which rounding corner shows improvement in reducing aerodynamic heating. The setting took place at 80 km above ground level in the atmosphere of Earth, which is around the altitude where the spacecraft would be during reentry. The flow characteristics being investigated include density and baroclinic pressure gradient [7]. The journal article also investigates aerodynamic surface quantities such as surface pressure and surface heat flux. The setup for the simulation can be seen below in Fig. 1.3. Since the flow was rarefied, DSMC was the computational method of choice with SPARTA as the program. After simulating the model, rounding the reattachment corner results in less concentration buildup in density as seen in Fig 1.4. On the other hand, rounding the separation corner does not show any difference in decreasing the density concentration of the cavity [7]. Generally, there is a direct correlation between the density buildup and heat. Thus, rounding the reattachment corner makes a positive difference in reducing aerodynamic heating.

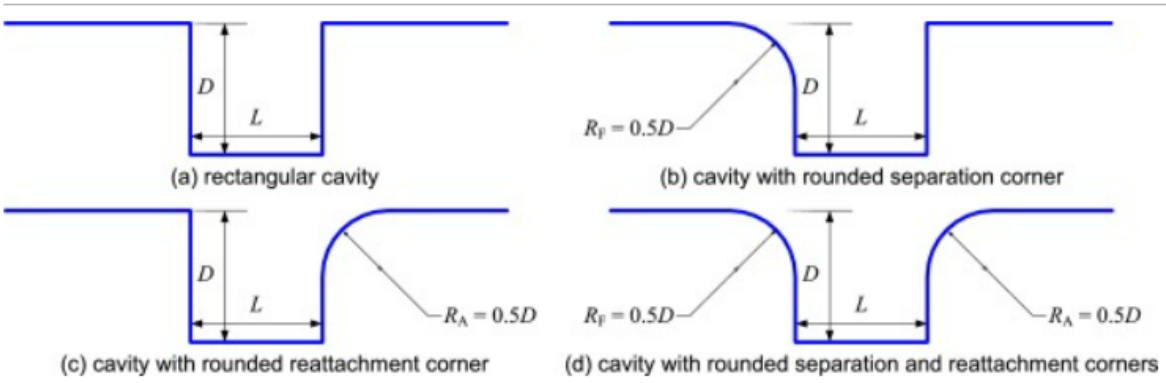



Fig. 1.3: Rounding corner geometry [7]

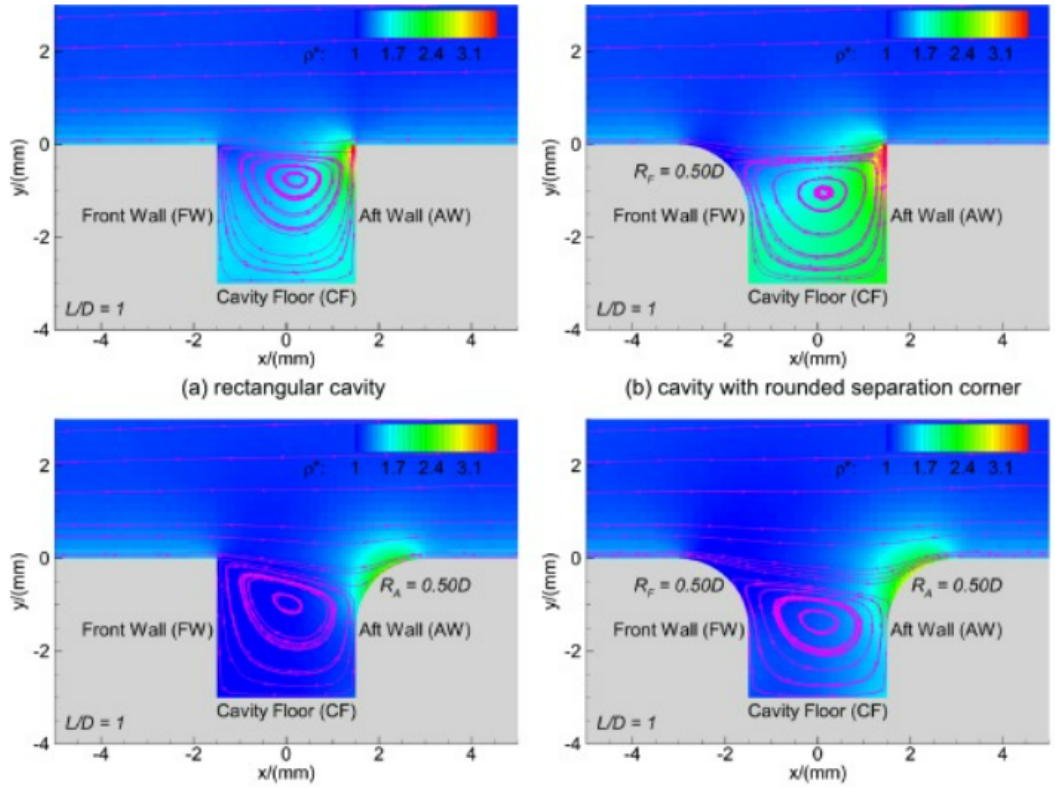



Fig. 1.4: Density contour for rounding corner [7]

Another configuration explored was implementing inclined planes on the reattachment and separation corners. The idea is the same as the previous method in keeping the L/D ratio constant along with free stream parameters. However, the main difference is the inclusion of the 3D geometry and investigating the differences between itself and the 2D simulations. The performance parameters are the same as the article from Jin et al. The method of choice was DSMC using the SPARTA program. The results indicate that creating an inclined plane for the separation corner decreases the density concentration, while modifying the reattachment corner with incline plane still has high density range, especially on the rear corner [8]. This result for density is the same as the journal from Jin et al. Therefore, changing the rear corner is an option to reduce the aerodynamic heating.

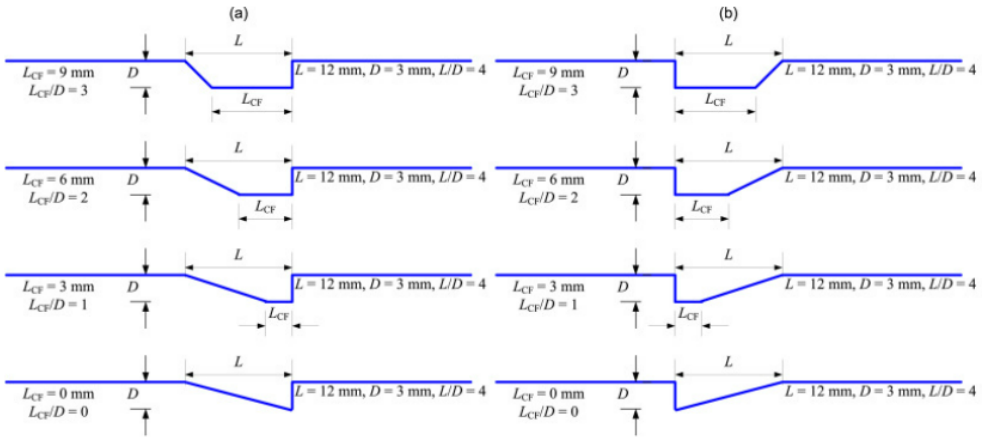



Fig. 1.5: Inclined plane geometry cases [8]

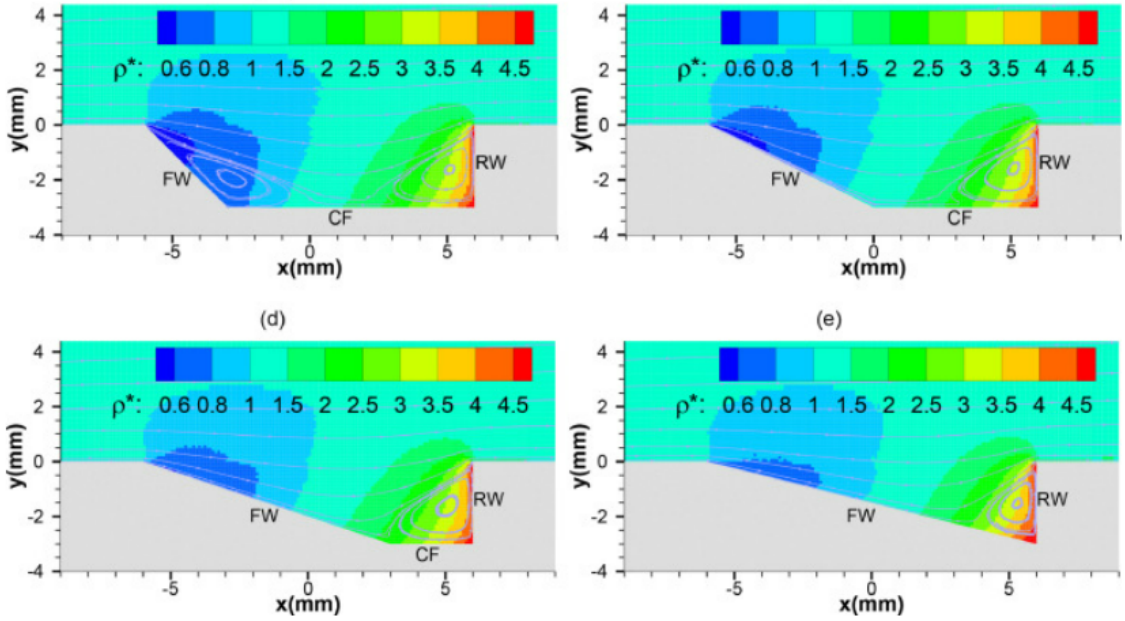



Fig. 1.6: Density contour for forward wall inclination [8]

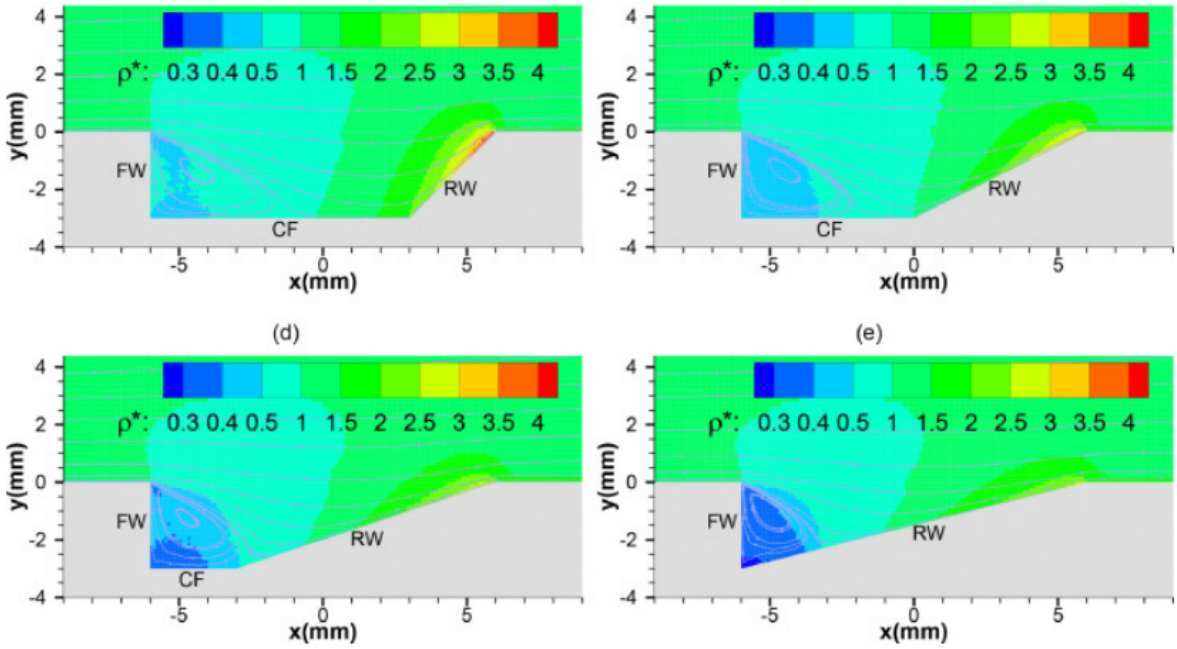



Fig. 1.7: Density contour of rear wall inclination [8]

The third explored configuration is changing the L/D ratio by varying L as seen in one of the journal articles [9] by Guo and Luo. Along with varying L, they also simulated 3 flow regime cases (by changing the altitude: 20 km, 50 km, and 70 km), while using DSMC as a simulation tool. The setup of the model can be seen in Fig. 1.6. The results analyzed were temperature, pressure, and velocity profiles [9]. From the results, increasing the length causes the flow starts to become unstable. Thus, keeping L close to D would be a preferred configuration.

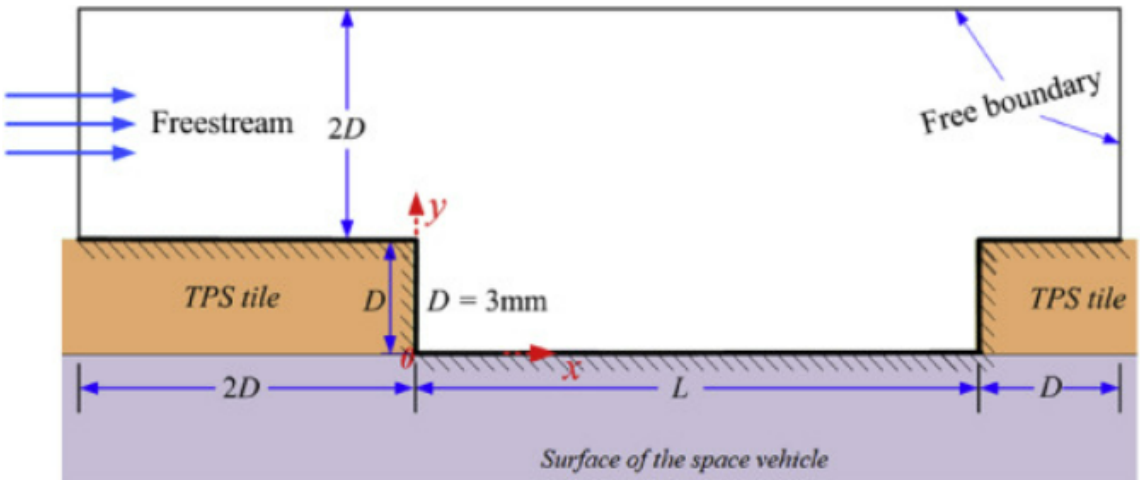



Fig. 1.8: Setup of the cavity geometry [9]

## 1.3. Project Proposal

Given the research and the problem definition, a proposed method to understand the aerothermal properties is to perform comparison studies on different configurations of the

cavities. This process will involve performing simulations to observe pressure, temperature, and dimensionless mass density fields. The next section contains in-depth information on obtaining these fields at certain conditions.

## 1.4. Methodology

For this project, the TPS discontinuity will be modeled as a 3D rectangular cavity to understand the flow behavior in all three directions. The flow through a gap will be characterized at selected flight conditions for rarefied flow environments taking place in the atmosphere of Earth. To solve the issue, comparison studies will be done to determine which cavity configuration is most optimal. This involves taking the best results from previous work and comparing them. This includes rounding reattachment corner of the cavity, varying the cavity depth, and increasing the spacing length.

From the literature review, a method that will be used is implementing DSMC to simulate the cavity flow and calculate the aerothermal properties. The Kn value indicates which computational method would best obtain accurate results. In the case of rarefied flow, which occurs when the density is extremely low, the motion and collision of the particles will not be negligible anymore. This means the mean free path of the particles will be high, resulting in the increase of Kn due to the direct relationship between the two quantities. This means general CFD software will not be able to generate accurate results. This is the reason why DSMC will play a huge role for this project. There are plenty of softwares that are based on DSMC implementation. However, only one program will be used to perform the simulations. More details will be discussed in the next chapter.

# 2. DSMC Method

## 2.1. Introduction

Chapter 2 will cover the details of the methodology in-depth to achieve the goals of this project. This includes the derivations of DSMC, different models, and the software that will be used for this project to utilizes this solver.

## 2.2. Background

Properties of the flow are split into two parts: macroscopic and microscopic. In other words, to describe these parts are continuum and discrete respectively. The last category takes into account the properties at a micro level such as the particle's interaction with each other and the surfaces. There are different models as seen in the figure below that are utilized depending on the flow regime. A quantity used to determine the appropriate model is Kn, which can be calculated using equation 2.1.

$$\mathrm{Kn} = \frac{\lambda}{\mathrm{l}} \tag{2.1}$$

As seen in the same figure, when Kn reaches to 0.1, continuum models are no longer valid. This means the microscopic method is needed to get an accurate simulation [13]. Boltzmann's equation, unlike Navier Stokes or Euler, takes into account the microscopic properties of the flow. The general form of Boltzmann equation is shown below. The left side is the free motion of the particles without the collision, while the other side of the equation is the Boltzmann Collison operator, which can also be written as equation 2.3 [14].

$$\frac{\partial \mathrm{F}}{\partial \mathrm{t}} + \mathrm{v} \cdot \nabla_{\mathrm{x}} \mathrm{F} = \mathrm{Q}(\mathrm{F}, \mathrm{F}) \tag{2.2}$$

$$\mathrm{Q}(\mathrm{F}, \mathrm{F})(\mathrm{v}) = \int_{\mathrm{R}^3} \mathrm{dv}_* \int_{\mathrm{S}^2} \mathrm{d}\sigma \mathrm{B}(|\mathrm{v} - \mathrm{v}_*|, \sigma)[\mathrm{F}'_* \mathrm{F}' - \mathrm{F}_* \mathrm{F}] \tag{2.3}$$

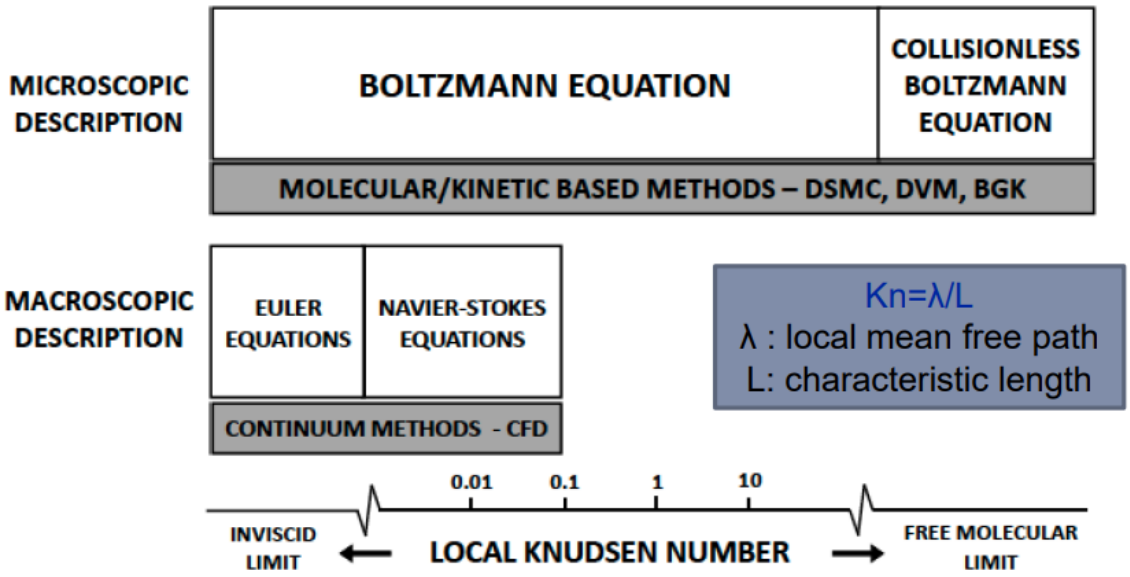



Fig. 2.1: Diagram of physical models for flow modeling [13]

In the 1960s, Graeme A. Bird developed a computational solver called DSMC that uses a stochastic approach to solve the Boltzmann equation [13]. Back then, typical machines were not capable enough to handle the computational power to perform these calculations. However, around the same time that Bird developed the Molecular Dynamics method in 1994, computers became faster and capable of complicated tasks [15]. The next subsection goes through the logistics of DSMC in-depth.

## 2.3. Breakdown of DSMC Models

Generally, the programs that implement DSMC contain the same algorithm. The flow chart for general DSMC solvers is seen in figure 2.2. Since DSMC is probabilistic, the particles are initialized at random conditions [16]. Then, they move every particle at a distance for each $\Delta t$. The particles are then indexed into cells for the next step, which is taking data for particle-to-particle collision. If they collide, new velocity set will be created depending on how the particles are modeled. The collision model is broken down into numerous types as seen in the sub-subsections. Determining the chemical behaviors of the collisions that fits the application is crucial to obtain an accurate simulation. Once the collision is modeled and the flow data for macroscopic properties is sampled, the code loops until reaching the number of iterations set by the program.

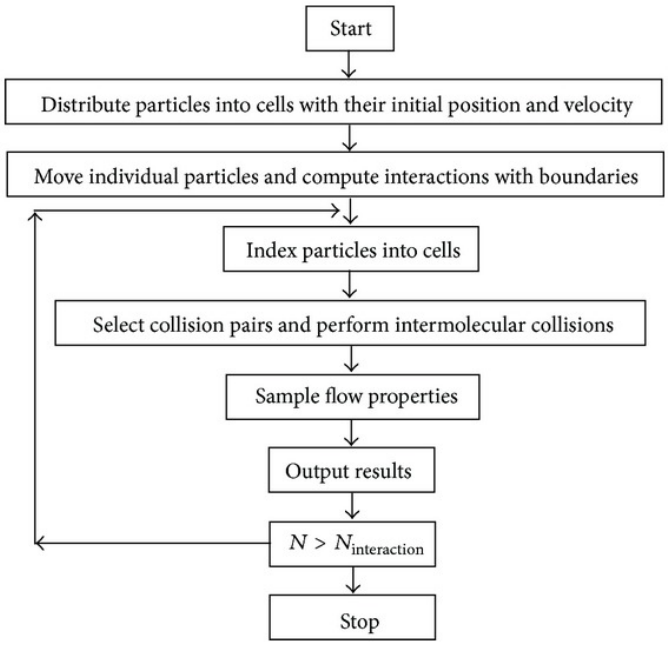



Fig. 2.2: DSMC code flowchart [16]

### 2.3.1. Particle-To-Surface Collision

When they collide with surface, the action of the solver is dependent on whether the collision is specular, diffuse, or a mix of both. The collision is specular if the particle bounces off at the same angle as its initial angle before impact. This can be seen on the left in figure 2.3. However, in reality, no surface is fully specular; thus, the specular condition will not be applied for this problem.

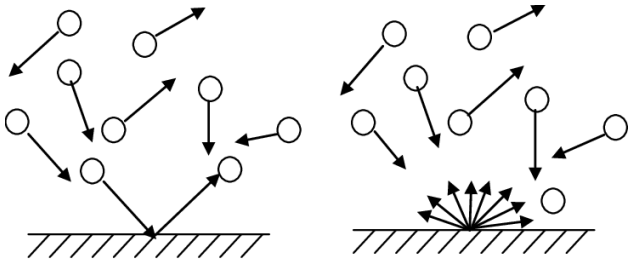


Fig. 2.3: Specular (left) and diffuse (right) [17]

### 2.3.2. Elastic Collision Models

In a 3D setting, the molecular collision occurs as seen on the figure below. The parameters during this phase vary depending on the chosen cross-sectional model chosen for the simulation. There have been three well-known models that have been devised that will be introduced in this section.

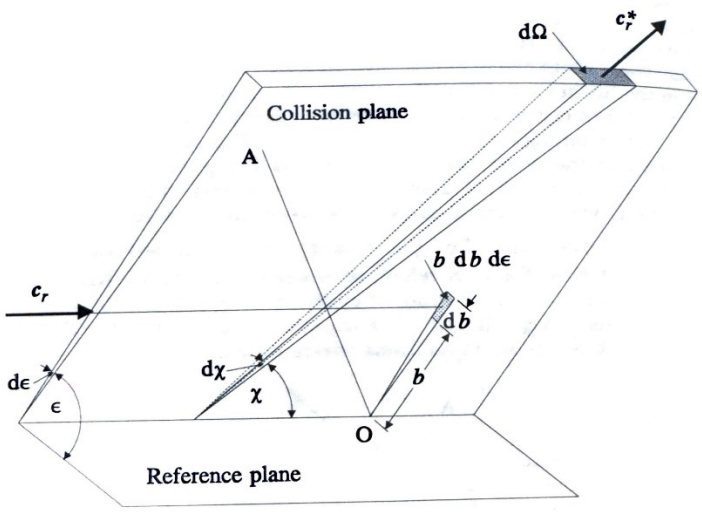


Fig. 2.4: Molecular collision diagram [20]

One of them is HS, developed by Bird in 1963, is the simplest in that the particle has constant total cross-section and numerically easier to model [15]. The diameter of the particle is constant at $d_{ref}$. However, the downside is the inaccuracy of the model by not taking into account the inverse power law [13]. This law is also known as the point center of repulsion model, which shows the intermolecular forces between the two molecules at a given distance [20]. The equation that models this law can be seen below.

$$F = \frac{\kappa}{r^{\eta}} \tag{2.4}$$

Later, in 1981, VHS was developed by Bird to account for viscosity variation based on the inverse law. It affects the diameter of the particle as seen in equation 2.5 [20], where $c_r$ is seen after [13]. Also, the deflection angle for VHS model is given in equation 2.7 [13].

$$d = d_{ref}\left(\frac{c_{r,ref}}{c_r}\right)^{\omega-0.5} \tag{2.5}$$

$$c_r = \sqrt{\frac{2\epsilon}{m}} \tag{2.6}$$

$$\chi_{VHS} = 2\cos^{-1}\left(\frac{b}{d}\right) \tag{2.7}$$

The third model VSS, which was developed by Koura and Matsumoto, is the same as VSS in terms of the diameter. However, the difference is the additional assumption of diffusion that affects only the deflection angle as seen in equation 2.8. [13]. The last two elastic models are the ones most used for DSMC applications.

$$\chi_{VSS} = 2\cos^{-1}\left(\frac{b}{d}\right)^{1/\alpha} \tag{2.8}$$

### 2.3.3. Internal Energy Modes

This section deals with the inelastic collision of the molecules with the internal modes. Generally, the molecular system contains internal energy, which is different from external energy types such as kinetic and potential. For every molecule, depending on the assumptions and the type of problem, there are four types for degrees of freedom [13].

- Translational
- Rotational
- Vibrational
- Electronic

For the first type, it deals with the translational motion of the molecules (along the directions x, y, and z). Depending on the type of setting (whether the domain is 2D or 3D), $\zeta$ is based on the number of directions that the molecule can move. For the case of the project, taking place in 3D space, $\zeta_{\text{trans}} = 3$ [13].

The second one deals with the rotation of the molecules about its own axes. Unlike the previous mode, $\zeta$ depends on the type of molecule (whether the molecule is diatomic or nonlinear) [13]. In other words, the number of degrees of freedom depend on the formation type of the molecules (whether the molecules are formed straight or bent). The prefix di- is defined as two, so diatomic infers a molecule composed of two or more atoms. For the case of air composed mainly of nitrogen gas $N_2$ and oxygen gas $O_2$, both of those are classified as linear. This can also be proven through the Lewis Structure, which deals with the valence properties of the molecule and the physical structure of the bonds [21]. The Lewis structure for both nitrogen and oxygen gas can be seen in figures 2.5 and 2.6 respectively. Since both cases show they are connected in a pair, these molecules are automatically classified as diatomic. Always, for any fully excited diatomic molecule, $\zeta_{\text{rot}} = 2$.

Fig. 2.5: Lewis structure of nitrogen gas

Fig. 2.6: Lewis structure of oxygen gas

Unlike the previous two types, the third one depends on the temperature of the system rather than being constant at any case. The equation for the degrees of freedom regarding vibration can be seen in equation 2. [20]. This follows the assumption that the molecules can be treated as a simple harmonic oscillator. $\Theta_{vib}$ depends on the type of molecule, which can be seen in table 2.1 for the case of air composed of $N_2$ and $O_2$.

$$\zeta_{vib} = \frac{2\Theta_{vib}/T}{\exp(\Theta_{vib}/T)-1} \tag{2.8}$$

Table 2.1: Characteristic vibrational temperature for air [20]

| Gas | $\Theta_{vib}$ |
|---|---|
| $N_2$ | 3371 |
| $O_2$ | 2256 |

Not all DSMC applications take into account electronic energy due to the high levels of energy. However, if taking into account quantum energy, the number of degrees of freedom (similar to vibrational energy) depend on temperature as seen in equation below [13].

$$\zeta_{el} = \frac{2\Sigma_I g_{el}\Theta_I e^{-\Theta_I/T}}{T_{el}\Sigma_I g_i e^{\Theta_I/T}} \tag{2.9}$$

Regardless of the internal mode type, assuming equilibrium, the total internal energy can be expressed as seen in equation 2.10.

$$E_i = \frac{1}{2}\zeta_i k_b T_i \tag{2.10}$$

## 2.4. DSMC Software

As mentioned from the previous chapter, there are plenty of softwares that implement the DSMC solver and take into account all the models explained in the previous sections of this chapter. The list of software to choose from can be seen below.

- DSMCFoam
- NASA DAC

- NASA MAP
- DS3V
- SPARTA

For this project, the software that will be used to perform DSMC analysis is an open-source called SPARTA, which was developed by Steve Plimpton in the Sandia National Laboratories in 2013 [18]. The software requires using a Linux based terminal and is designed to run in parallel mode using MPI. The commands in the source file are based on C language.

According to the SPARTA manual, the process in designing an input script (which is in a text file format) involves several steps [18].

- Initialization
- Defining Problem
- Settings
- Running Simulation

The first stage is defining the parameters before going through the problem setup. Those parameters include dimension (choosing either 2D or 3D), seed number for the random generator, and the units. The second step involves setting the boundary conditions, geometry and grid style. The geometry file that the script reads is a text file as well and can be established in different ways. To make creating a 3D geometry efficient, there are python scripts in the tools directory. One of them can convert an STL file in ASCII format from a CAD program into SPARTA surface file. For the grid, the SPARTA program uses cartesian style only. In the third stage, the settings are specifically the chemistry models that are used for the simulation. These include the surface and collision models detailed in the previous subsection. This depends on the problem description. Finally, the last stage sets up the number of runs for the simulation and generating the data. After establishing the input script, using an executable MPI, the simulation can run and generate results, creating a log. Although images can be seen in the dump file after outputting results, post processing has to be done for better visualization of the flow behavior. This process will be discussed with the results section.

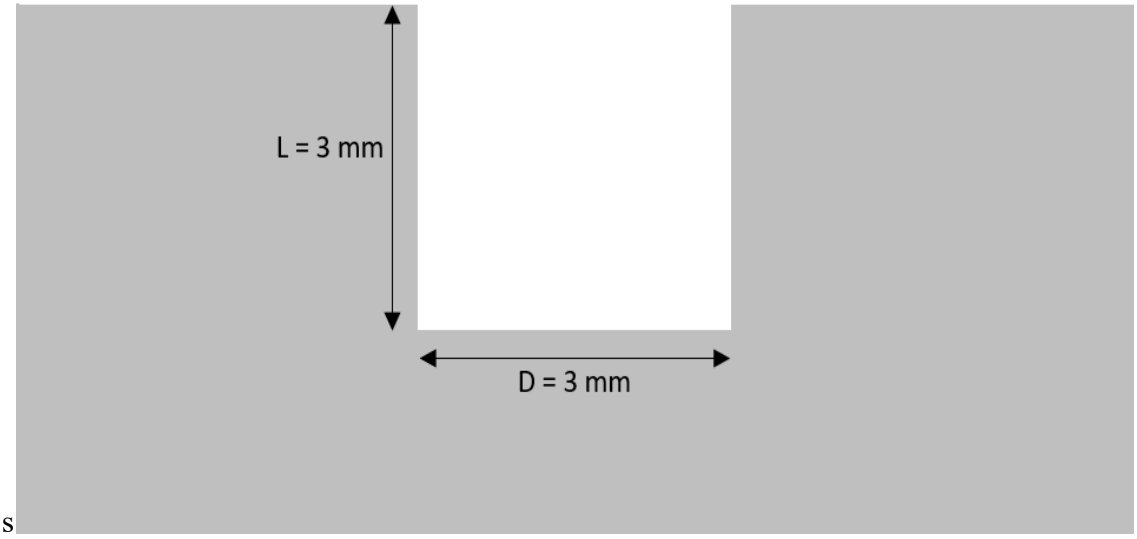
# 3. Problem Setup

## 3.1. Introduction

This chapter will go over the setup for this project. This includes defining the dimensions of geometry and setting up the boundary conditions that define and represent the problem in an accurate way.

## 3.2. Geometry

For this project, from the literature review, there will be 3 cases (one with each type of configuration). One of them is a rectangular cavity with L = D = 3 mm as seen in Fig. 3.1, which was chosen since one of the research articles showed that keeping L close to D would make the flow stable [9]. The second case is the same as the previous one except with the reattachment corner being rounded as seen in Fig. 3.2. This had to do with the study that showed rounding only the rear corner reduced the aerothermal heating on the cavity, while changing the front corner did not make a difference [7]. The third case is implementing the incline plane of 45 deg onto the rear corner as seen in Figure 3.3. This configuration was inspired by a research article [8]. The incline plane on the rear corner is beneficial for the same reasoning as rounding the rear corner.




s

Fig. 3.1: Case 1: rectangular cavity

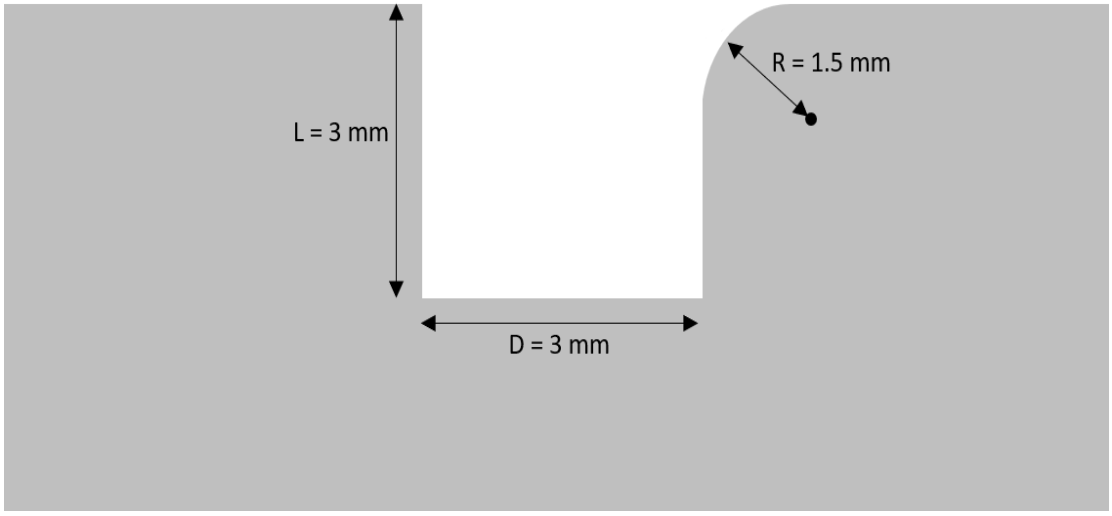


Fig. 3.2: Case 2: rectangular cavity with rear corner rounded

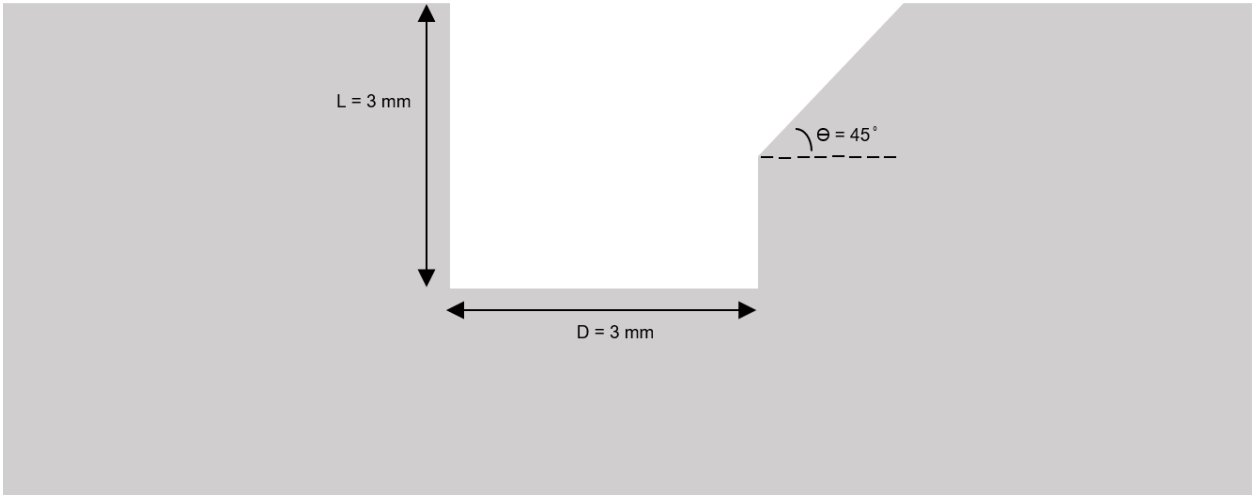


Fig. 3.3: Case 3: rectangular cavity with Incline Plane of 45°

As mentioned in the previous chapter, the 3D geometry can be constructed in different ways. However, for efficiency, this can be done by converting the STL file to surface file. The first step is using a CAD software to create the geometry, which in this case is SolidWorks. The 3D models for each case are seen below. Since the cavity is the only part of the geometry that matters in the simulation, the depth of the block was set to an arbitrary value. In SPARTA, the geometry can be clipped, so that the bottom part of the block will not be taken into account during simulation. Also, the z distance of the cavity was set to an appropriate value that the flow can be observed. A major requirement when creating any geometry is to make sure the surface is watertight. This means the surface elements have to be closed off to the point where nothing can go through. If the STL file is not watertight, SPARTA will convert the geometry regardless. However, there will be a warning that the surface is not watertight. This restriction cannot be ignored since the simulation would not run at all and will just display an error message for this

issue. For this reason, the cavity configuration for case 3 ended up resulting as seen in figures 3.3 and 3.6 respectively.

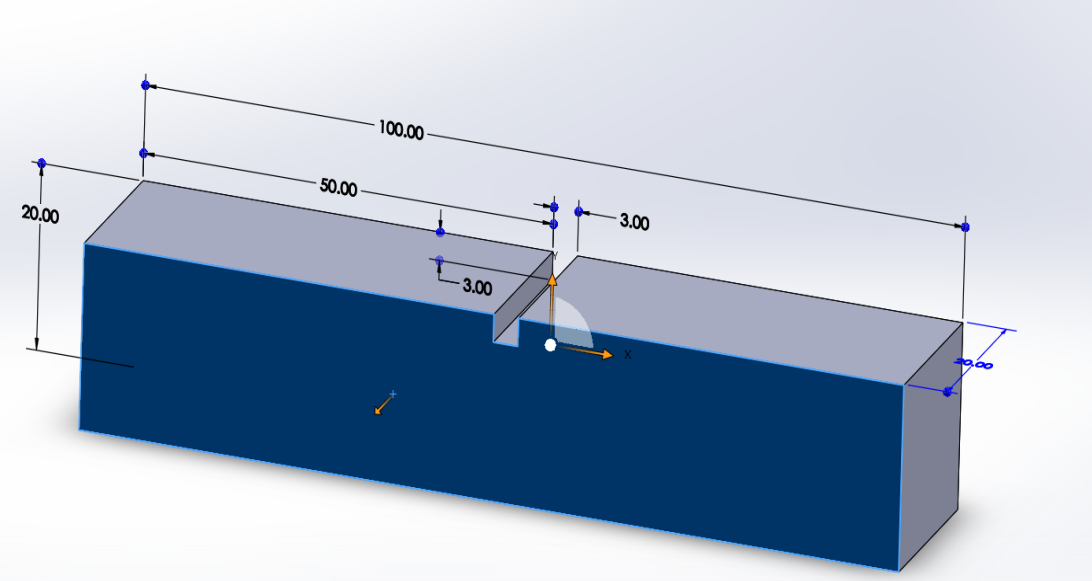



Fig. 3.4: 3D model of case 1

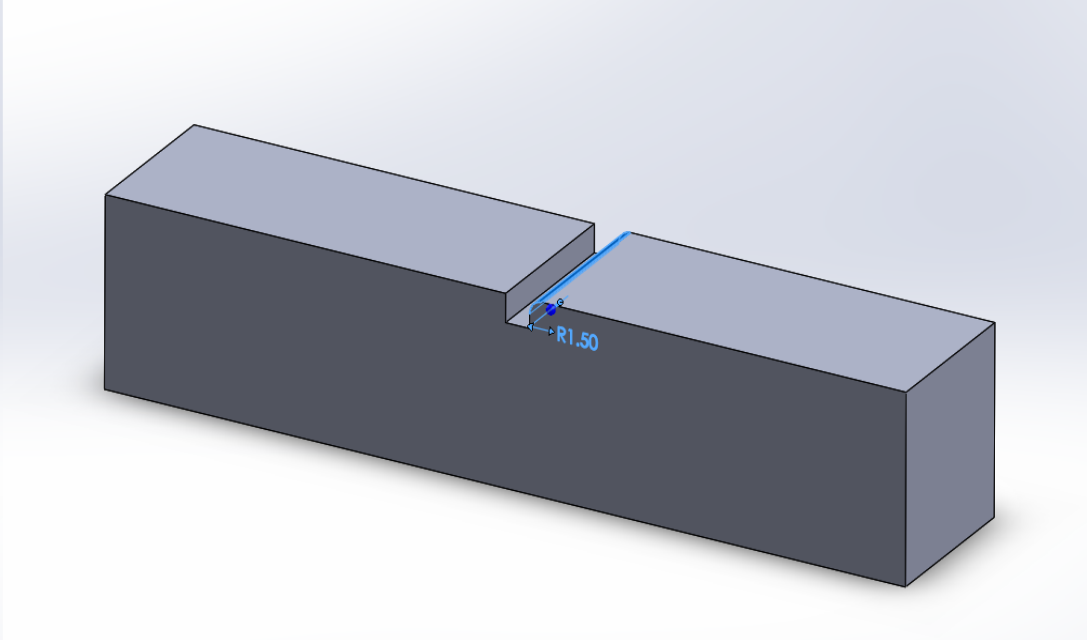



Fig. 3.5: 3D model of case 2

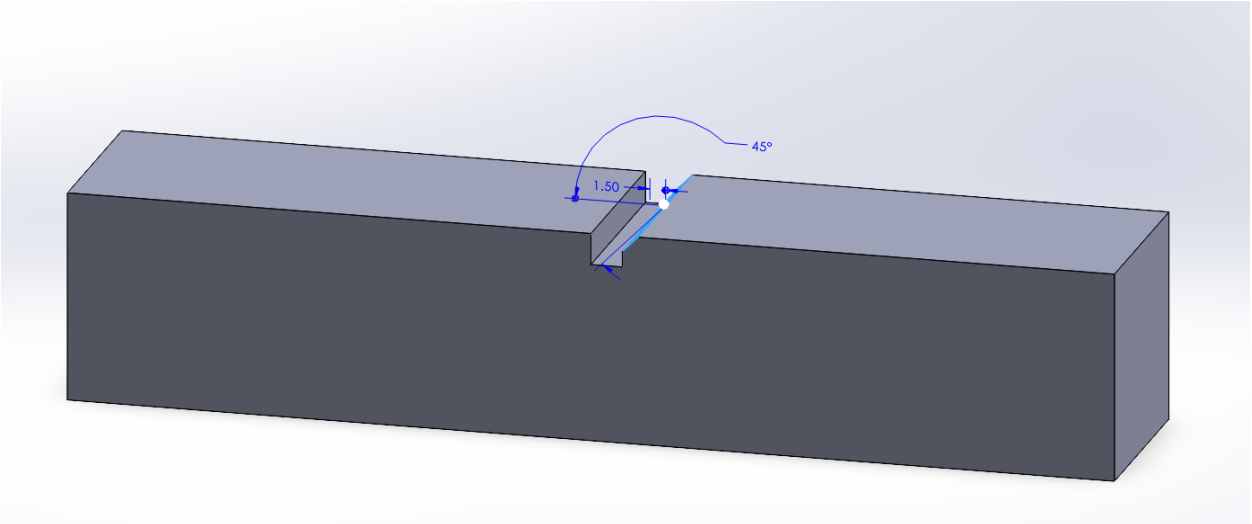



Fig. 3.6: 3D model of case 3

Next, the SPARTA program requires converting the part file to STL in ASCII format. If the format is not selected correctly, the program will not work and will result in an error. Then, from the tools directory, only one command is required, which calls python and runs the script. Once that command is executed, the SPARTA surface file (which are in units of meters) can be observed as seen in Appendix A for all three cases. Generally, SPARTA surface file is constructed in a way that shows the number of points and the coordinates of each points in the x, y, and z direction along with the number of triangles (since this is 3D) that connect all the points that make up the geometry.

## 3.3. Boundary Conditions

### 3.3.1. Assumptions

One of the major steps before attempting the problem is to establish assumptions that can be applied for the simulation. The list can be seen in this section below.

- 3D
- Constant wall temperature
- No ablation
- No ionization

The first item is self-explanatory and is mentioned in the beginning of the report. For the second assumption, this has to do with the geometry. Specifically, as seen in the three cases, all of their surfaces are flat (except the second case with the rounded reattachment corner), so they can be modeled as a flat plate. For any flat plate, the wall temperature is constant, so this assumption applies for the project. In terms of ablation, although TPS systems generally contain ablative material, which helps the heat shield reduce the aerodynamic heating. However, since DSMC is very complex and modeling ablation is computationally expensive on the computer,

this project will not consider the ablation of the TPS tiles. For the next item, ionization occurs when the speed of the object reaches higher temperatures to the point where the surrounding gas becomes charged (or ionized) that the medium becomes plasma. The temperature in which this process occurs depends on the type of medium. For the case of air, the temperature for N2 and O2 is 9000 K as seen on figure 3.1. However, since the software inputs simulated particles, inputing charged particles will take computational power. For this project, to avoid running out of computer memory for the simulation, the project will not take into account ionization.

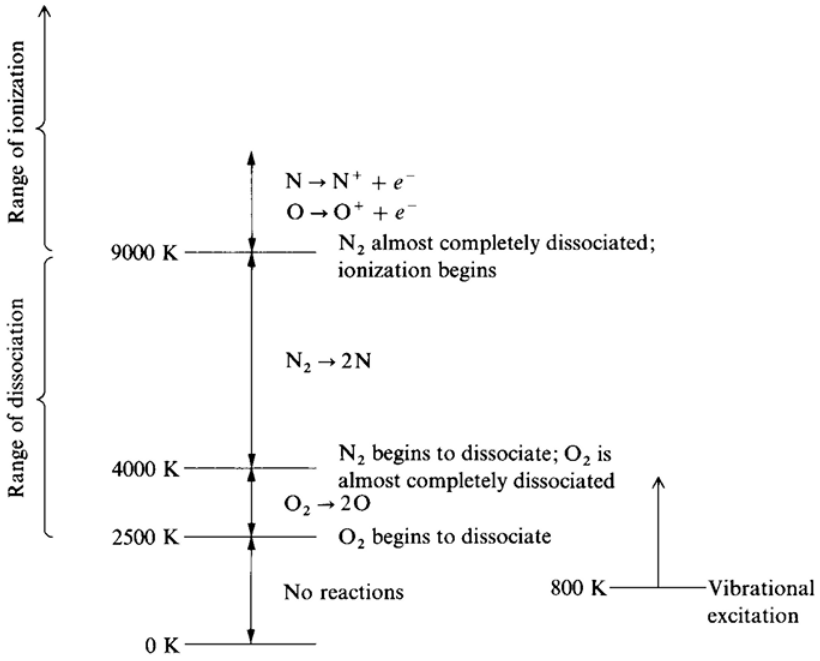



Fig. 3.7: Diagram for N2 and O2 range [5]

### 3.3.2. Atmospheric Conditions

To keep this study consistent and best determine which configuration is most optimal, the altitude will be kept constant for each case. The value for altitude is crucial for determining the free-stream conditions, which are seen in Table 3.1. For this work, the altitude is 80 km and $T_w$ will be kept constant at 1000 K. The wall temperature occurs at a high level due to the fact that this occurs during reentry at the stagnation point. The flow will be entering the domain parallel to the surface at a velocity of 7565.7 m/s.

Table 3.1: Free-stream parameters at 80 km [19]

| $V_\infty$ | $T_\infty$ | $P_\infty$ | $n_\infty$ | $\rho_\infty$ | $\lambda_\infty$ |
|---|---|---|---|---|---|
| 7565.7 | 198.63 | 1.05 | $3.837 * 10^{20}$ | $1.84 * 10^{-5}$ | $4.402 * 10^{-3}$ |

### 3.3.3. Collision Model

Given the different aspects of DSMC, the particles will be modeled as VSS. Fortunately, SPARTA contains collision parameters for the VSS model of each species in the program as defined in Table 3.2. The only species that will be involved in this simulation are N2 and O2, which are the main species composed for air. Also, using the

Table 3.2: VSS model for air [18]

| Species | d | $\omega$ | $\alpha$ | $T_{ref}$ |
|---|---|---|---|---|
| $N_2$ | $3.96 * 10^{-10}$ | 0.77 | 1.4 | 273.15 |
| $O_2$ | $4.07 * 10^{-10}$ | 0.74 | 1.6 | 273.15 |

# 4. Defining DSMC Simulation

## 4.1. Introduction

From the given assumptions, geometries, and boundary conditions made from the earlier chapter, this section will cover the crucial steps of setting up the input script in SPARTA to properly run the simulation for each of the three cases. These include generating the mesh, discretizing the time, and creating the particles. All three of these steps will be important before creating the input files for each case used for the project. However, the calculations have been performed using Matlab, where the code can be seen in Appendix B. There is also a discussion on conducting a mesh independent study based on the cell size.

## 4.2. Mesh Generation

In any numerical flow solver, the domain needs to be split into an enormous number of tiny cells, in which the numerical solver can calculate the thermal properties for each of them. In a flow simulator, selecting the mesh configuration depends on the software and the problem at hand. For the shape of the cell, that is predetermined due to the fact that SPARTA only utilizes a cartesian grid. An example of this type of mesh can be seen in Figure 4.1 for the case of a circle in 2D simulation.

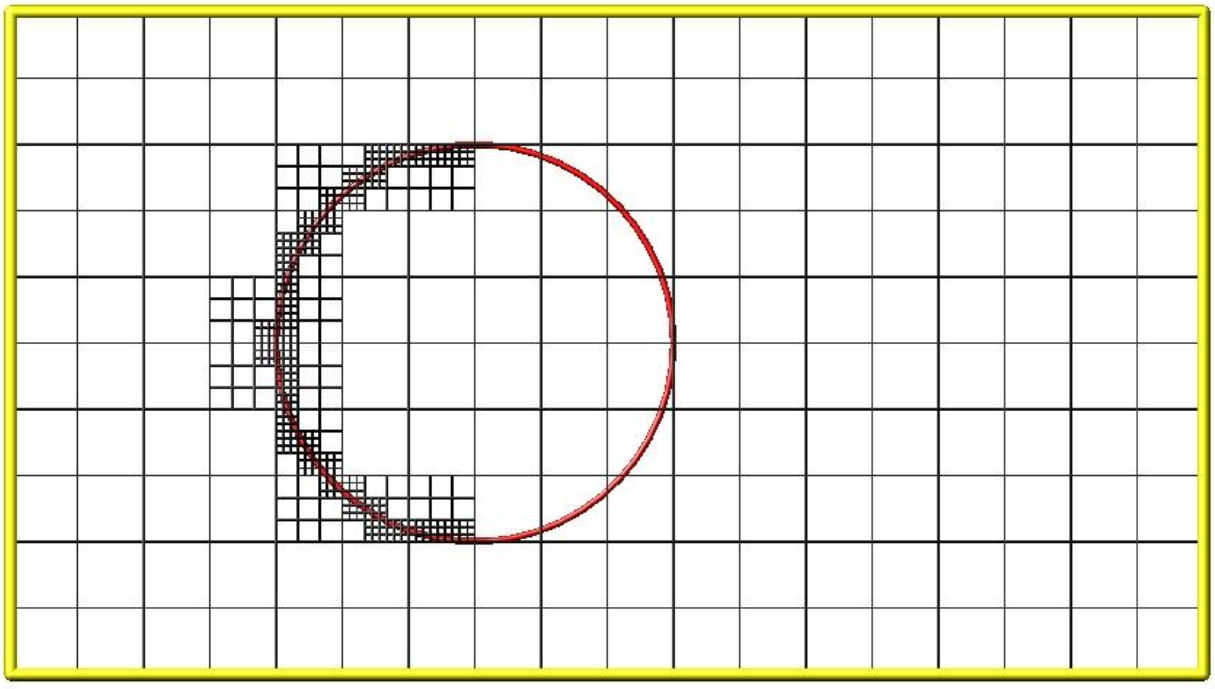


Fig. 4.1: Example of cartesian mesh in SPARTA [18]

Unlike deciding the shape, however, the size of those cells will not be as simple to plan. Theoretically, the higher the cell count, the more exact the simulation will be. However, there is also a direct correlation between mesh count and computational power. This means too many cells can cause the simulation to take an unnecessary amount of time to run when the result can be

achieved with a smaller number of cells. For DSMC, a general method for quantifying the size needs to follow the constraint as seen in Equation 4.1 [19][21]. In other words, the cell size needs to be much smaller than the local mean free path to capture the possible particle collisions and the effect it has on a macroscopic level.

$$\frac{\Delta x}{\lambda} \ll 1 \tag{4.1}$$

The value for mean free path depends on the chosen elastic model. Since the model chosen is VSS, the equation is derived as seen through equation 4.2 [20]. The mean free path, for this equation, is local (which means this value is dependent on the temperature of that specific location). This information is not fully known until running the simulation.

$$\lambda = \frac{1}{\sqrt{2}\pi d_{ref} n \left(\frac{T_{ref}}{T}\right)^{\omega - 0.5}} \tag{4.2 [20]}$$

For this reason, the mean free path used will be the free-stream case since that is the value already known. For the unrefined settings, the cell size will be initially equal to $\frac{\lambda_\infty}{10}$, which is $4.402 * 10^{-4}$ m. This quantity can be useful for computing the number of cells (which is the information that SPARTA needs when creating the input script). The number of cells can be determined by equation 4.3. This equation is only valid since the mesh is classified as a structured grid. The only information left that is unknown is the domain size.

$$N_{cells,i} = \frac{L_{dom,i}}{\Delta x} \tag{4.3}$$

The domain itself needs to be big enough to capture geometry, but not too large that the cavity cannot be captured. The coordinates along the x, y, and z direction from one end to another will be (0.035, 0.075) m, (0.01, 0.04) m, and (0, 0.02) m respectively. This means $L_x$, $L_y$, and $L_z$ are set to be 0.12 m, 0.05 m, and 0.04 m respectively. Thus, the number of cells in the x, y, and z direction will be approximately 90, 64, and 45 cells for a total of 281,360 cells. Since the dimensions of the geometry are relatively the same, the mesh configuration will be the same for all of them. The mesh for each case will be shown in the next chapter accompanied by the results of the simulations.

Although figuring out the cell size is important for the unrefined setting, the cells need to be refined in certain areas (especially in the cavity). To accomplish this task, the cells around the surface will be refined, which can be done easily by one of the commands in SPARTA. It does the refinement process in one step and not continuous.

## 4.3. Time Discretization

Another factor that is crucial for generating an accurate result is determining the timestep required for each step. The general rule is the given in equation 4.4 to capture the possible collisions that could occur when running the simulation [22]. To find the mean collision time (or the average time that takes particles to collide), this requires using equation 4.5 [23]. As

mentioned in the previous section, the local mean free path is not easily found without the simulation being performed. Therefore, the option would need to be to use the free-stream value, which is already known. For the mean velocity of the particles, this can be determined by equation 4.6 [24]. The Boltzmann constant is $1.38 * 10^{23}$ J/K for every case.

$$\frac{\Delta t}{\bar{\tau}} < 1 \tag{4.4}$$

$$\bar{\tau} = \frac{\lambda}{\bar{c}} \tag{4.5}$$

$$\bar{c} = \sqrt{\frac{2kT}{m}} \tag{4.6}$$

In terms of the mass, this requires determining the molar mass of the gas as seen in 4.7. Since air is a mixture of species and not a homogeneous gas, this will require some calculation to get this quantity. This depends on the concentration of each species as seen in equation 4.8 with the expanded version like in equation 4.9. Since the species that will be involved for this case are N2 and O2, the concentration at 80 km altitude are 76.3% and 23.7% respectively. Also, the molar masses for N2 and O2 are 28 g/mol and 32 g/mol respectively. From these values, the molar mass for the mixture is 28.64 g/mol. From there, using equation 4.7, the mass of air per particle can be calculated. This conversion can be seen in equation 4.9 (where $N_A = 6.023 * 10^{23}$ particles), which results in $m_{air} = 4.75 * 10^{23}$ kg/particle.

$$m_{air} = \frac{M_{air}}{N_A} \tag{4.7}$$

$$M_{air} = \sum C_i M_i \tag{4.8}$$

$$M_{air} = C_{N2}(M_{N2}) + C_{O2}(M_{O2}) \tag{4.9}$$

Finally, the rest of the calculations can be performed. The temperature that will be used to solve equation 4.6 is the free-stream temperature for the same reason as the mean free path. Thus, $\bar{c}$ and $\bar{\tau}$ are 10.68 m/s and $4.12 * 10^{-4}$ s respectively. The time step, for all cases of this project, will be set to tau/10, which is $4.12 * 10^{-5}$ s.

## 4.4. Particle Generation

When running any DSMC simulation, the particles need to be inputted into the simulation to be able to generate the results from sampling the data and understand the macroscopic behaviors of the molecular collisions. Besides adding the number density in the input script (which is the free-stream value given in table 3.1), the fnum value needs to be found for the program to generate particles in the simulation. The fnum value can be seen in equation 4.10 [20].

$$fnum = \frac{N_{real}}{N_{sim}} \tag{4.10}$$

However, these two values are unknown and need to be computed. For the number of real particles in the domain, this is given by equation 4.11. The volume of the domain is the same as the volume of a rectangular prism, which is 0.000024 $m^3$ in this case. Thus, the number of real particles results as $9.21 * 10^{15}$ molecules.

$$N_{real} = n_{Inf} * V_{dom} \tag{4.11}$$

The next step is to find the number of simulated particles, which can be found by the equation below. The only unknown value left is the number of particles per cell. There is a direct correlation between the number of particles per cell and the accuracy of the simulation. On the other hand, this can also increase the computational power required to perform the simulation. A rule of thumb is the particles per cell need to be greater than 5 [23]. Each simulated particle adds memory required for the simulation. In other words, the computational cost increases with the number of particles in the simulation. Thus, for all three cases of this project, $N_{par}$ will be set to 10 particles per cell. From this value and the number of cells determined in the previous section, $N_{sim} = 2.813 * 10^6$; thus, $fnum = 3.27 * 10^9$. Now, the next step is to create the input scripts for each case, which will be discussed in the next chapter.

$$N_{sim} = N_{par} * N_{cell} \tag{4.12}$$

# 5. Results

## 5.1. Introduction

This chapter will showcase the results from SPARTA for all the three cases. The input scripts for each can be seen on Appendix C. The process for developing the input file follows the steps as discussed in chapter 2. Although the simulations were done in SPARTA to gather grid and surface data, post-processing needs to be performed to be able to visualize the results. There are a couple of options for post-processing softwares that can convert SPARTA results for visualization: tecplot and paraview. The first option requires requesting a student license, which is costly and takes time to receive. However, the other choice doesn't have these issues. As a matter of fact, the software is free, open-source that can perform the tasks just as effectively. Thus, paraview was the program of choice for post-processing for this project. SPARTA has tools for post-processsing (specifically for paraview, which involves using the paraview python interpreter pvpython to convert the grid and surface files to paraview format. Since the simulation is 3D, when converting the file to paraview format for the grid, a requirement is to input the location of the planes that are slicing the geometry and domain. The appearance of the paraview images with all the mesh and grid properties can be seen in the next section of this chapter.

## 5.2. Mesh Appearance

Before calculating the grid properties, the input script runs the command to write the grid with the refinement already taken place. Since the cavity is rectangular and the cross section does not change along the z direction, the behavior can be assumed the same in any location along the z direction. For this reason, the xy plane can be sliced anywhere as long as it goes through the cavity geometry. The result of the 3D mesh can be seen in figure 5.1. When zooming in to the cavity, as seen in 5.2, the cells are refined in the surface to showcase any flow behavior in the surface. This feature is essential especially for the corners of the cavity.

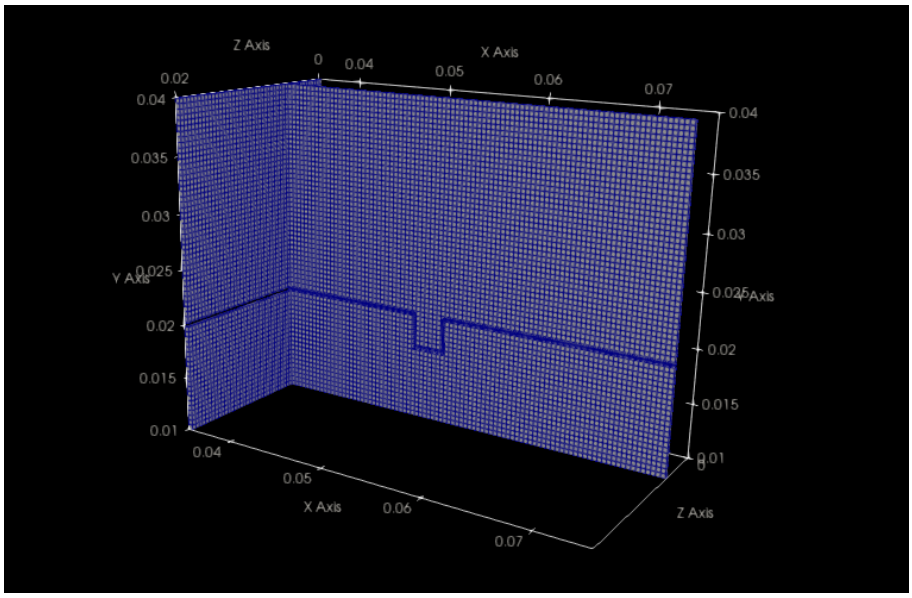


Fig. 5.1: Mesh configuration overview

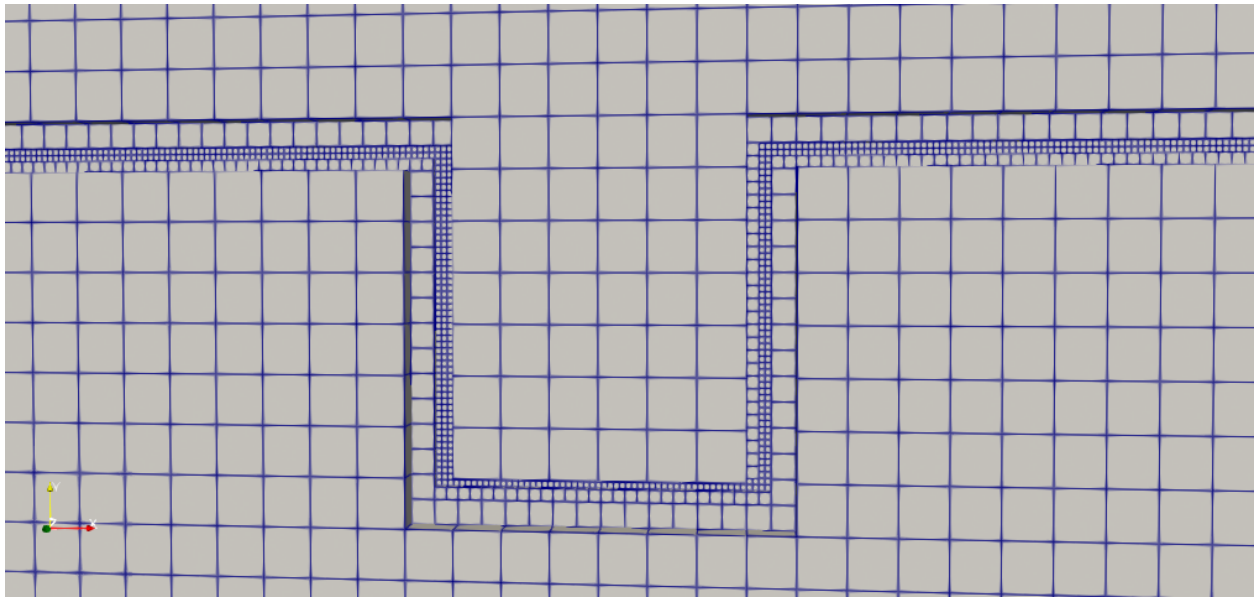


Fig. 5.2: Mesh configuration zoom view

## 5.3. Grid Properties

### 5.3.1. Case 1: regular cavity

As explained in chapter 1, the quantities that will be discussed to determine which configuration reduces aerodynamic heating are density, pressure, and temperature. The dimensionless density contour (which is a ratio of mass density to freestream mass density) can be used to verifythe results from the simulation to the reference from the lit review for the regular cavity case. As seen in figure 5.3, when observing the xy plane, there is buildup in the

corner reaching up to 3.867, which is higher than the 3.6358 value from the research paper [7]. When calculating the percentage difference between the simulation and research result, using the equation below, the result is about 6.35%.

$$\% \, change = \frac{\left(\frac{\rho}{\rho_\infty}\right)_{peak,exp} - \left(\frac{\rho}{\rho_\infty}\right)_{peak,act}}{\left(\frac{\rho}{\rho_\infty}\right)_{peak,act}} * 100 \tag{5.1}$$

This slight difference could be due to the nature of the DSMC method as it is a probabilistic approach to generating a simulation of the flow behavior. However, the behavior of particle buildup does match on both sides, which shows the accuracy of the simulation for this project. There is also buildup of pressure, which can be seen by the pressure contour in figure 5.5 with the highest point reaching to 176.65 Pa. The temperature contour reaches to extremely large values, where the peak temperature is about 19418.9 K. This is reasonable since the speed is approximately 8000 m/s.

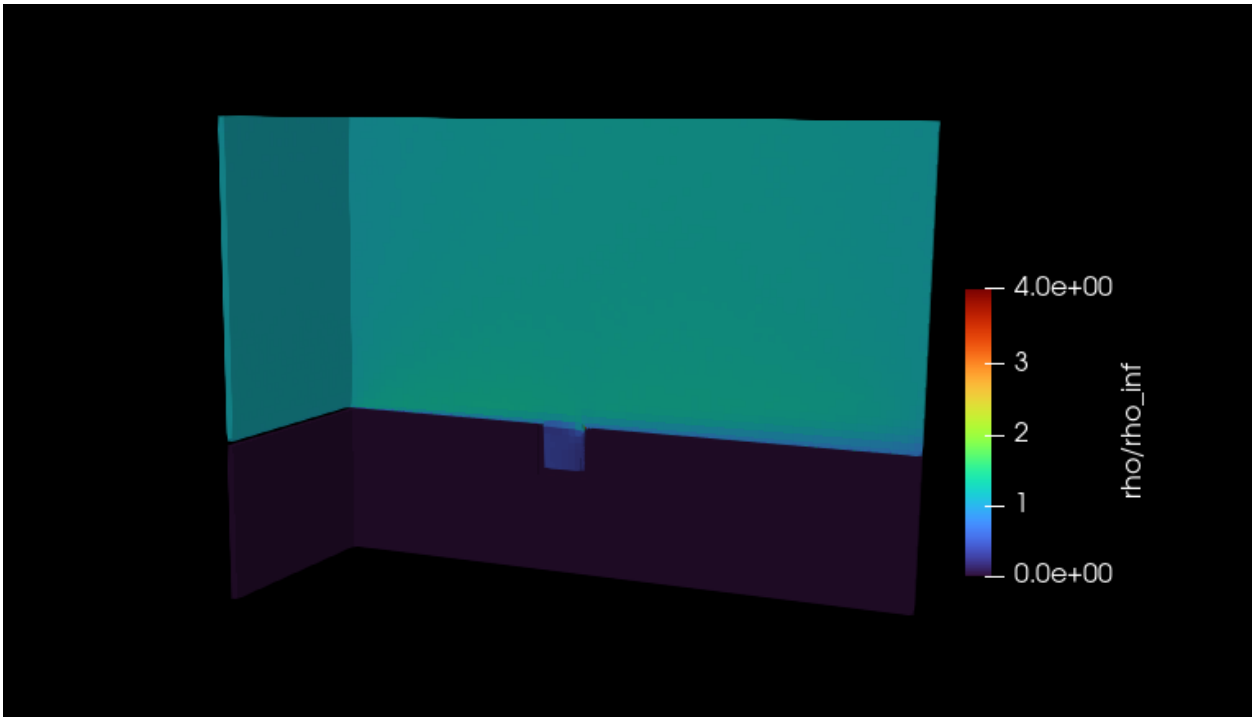



Fig. 5.3: Overview of mass density ratio contour for case 1

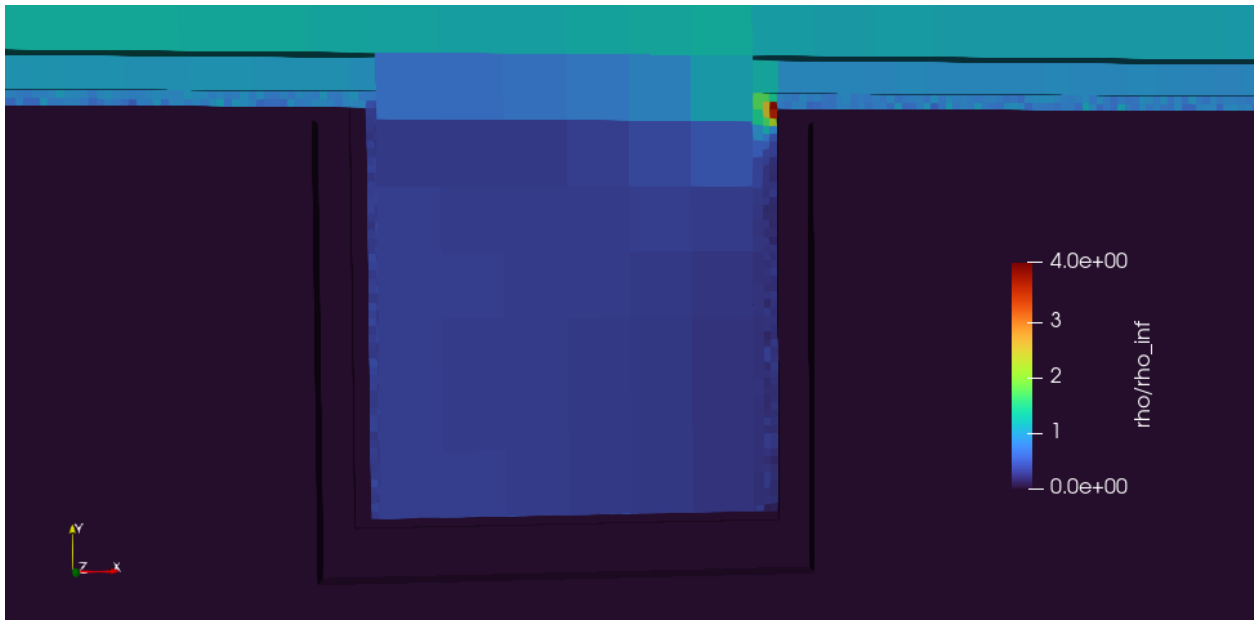


Fig. 5.4: Closeup of mass density ratio contour for case 1

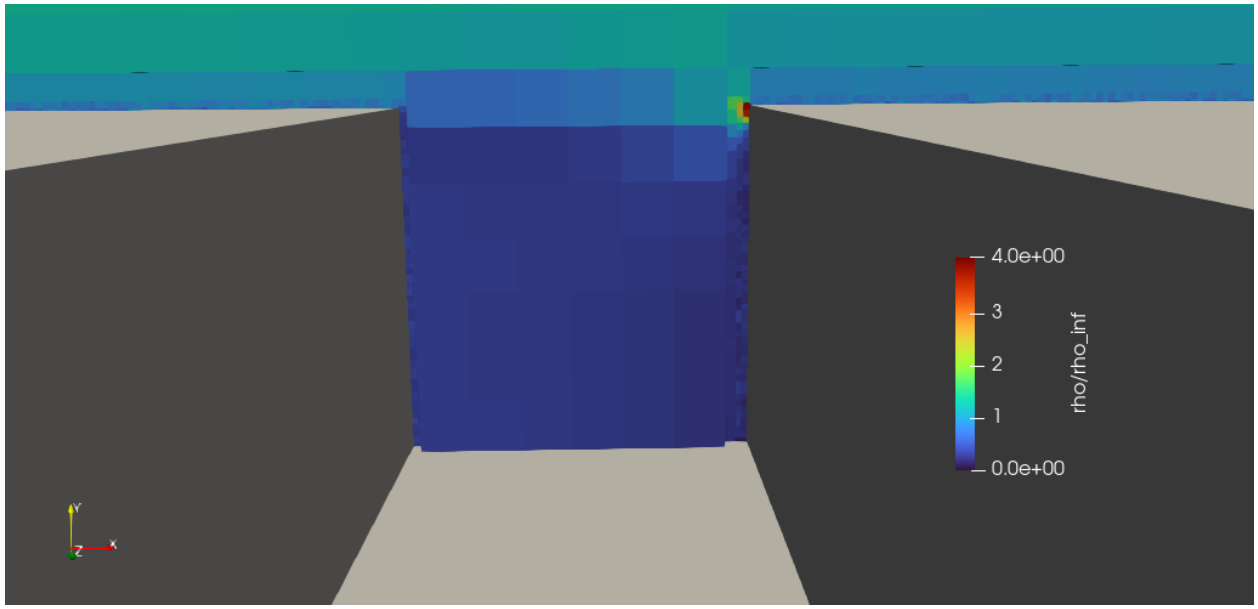


Fig. 5.5: Closeup of Closeup of mass density ratio contour with geometry for case 1

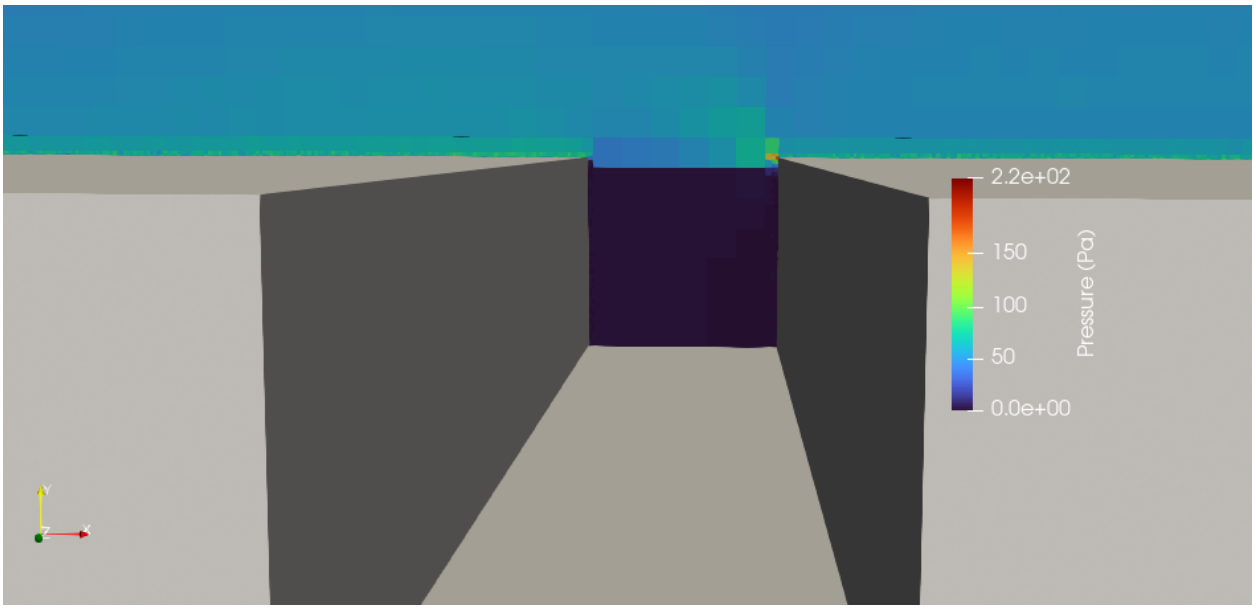


Fig. 5.6: Closeup of pressure contour with geometry for case 1

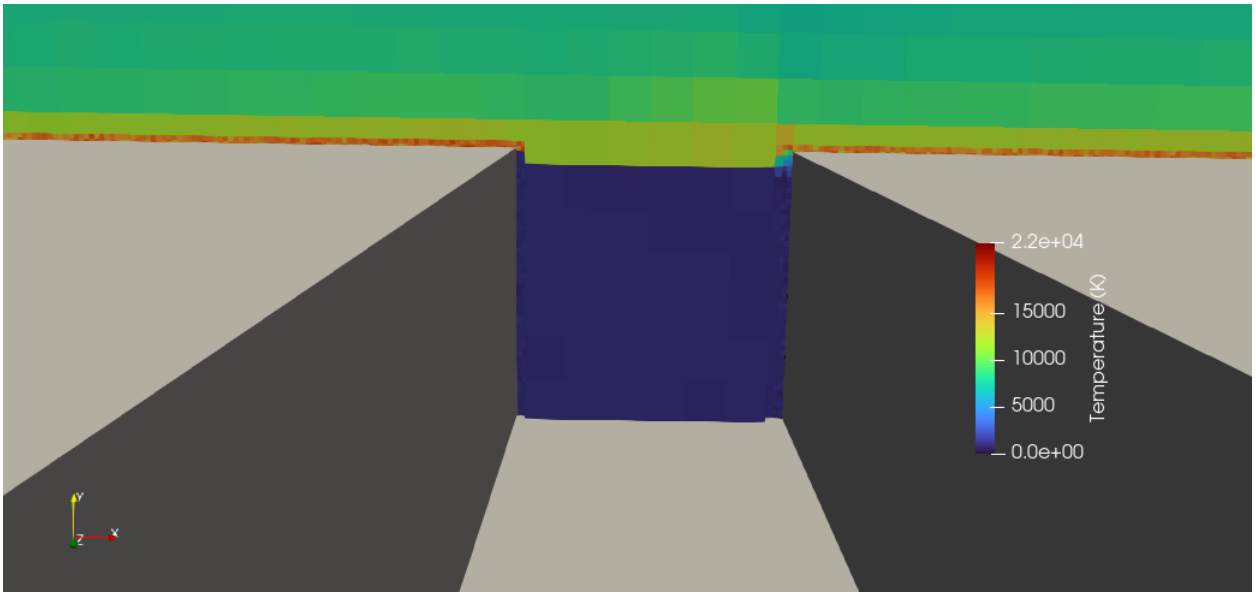


Fig. 5.7: Closeup of temperature contour with geometry for case 1

### 5.3.2. Case 2: rounded reattachment corner

The rounded reattachment corner case seems similar when observing figure 5.8. However, when looking at the closeup in figures 5.9 and 5.10, there is a difference in the peak value for dimensionless mass density. In fact, the value is smaller than the previous case, which is about 2.2. In other words, there is not as much particle buildup. This can also be seen in figure 5.11 with the pressure contour with the peak value reaching to 162 Pa. For the temperature contour in figure 5.12, the peak value reaches up to 18460.9 K, which is smaller than the regular

cavity case. In other words, from the three flow contours, this second configuration for 3D is an improvement from the regular cavity.

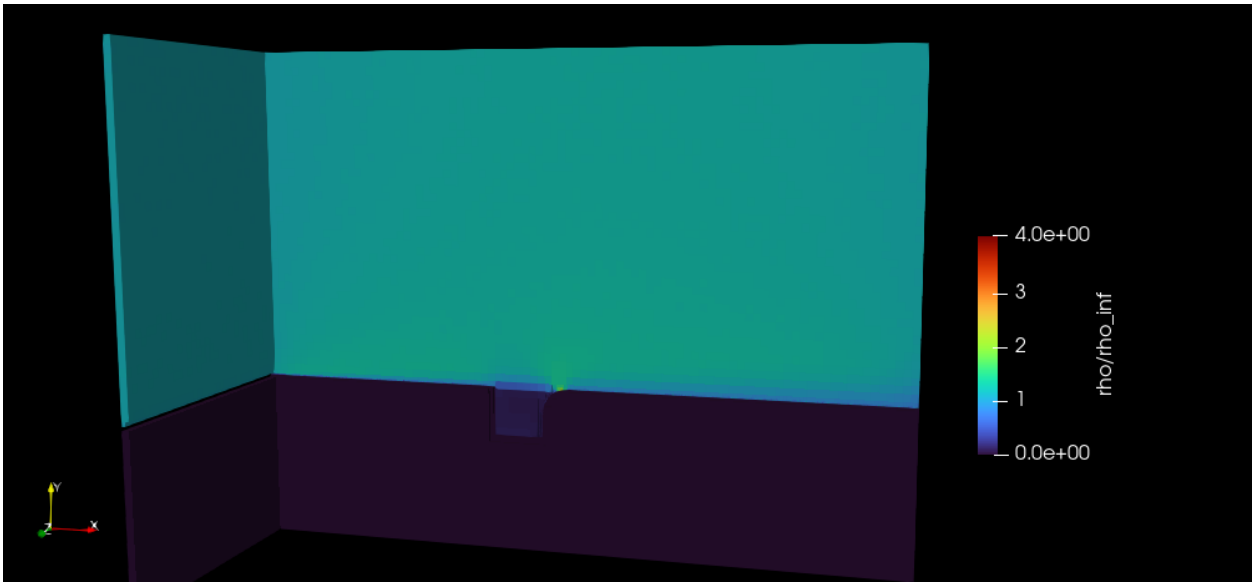



Fig. 5.8: Overview of mass density ratio contour for case 2


Fig. 5.9: Closeup of mass density ratio contour for case 2

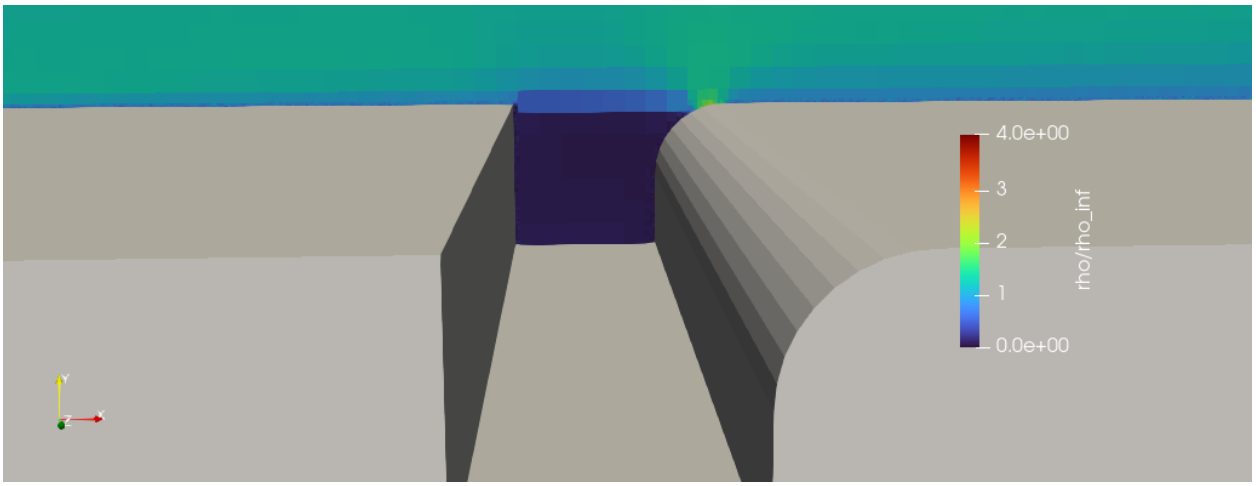


Fig. 5.10: Closeup of mass density ratio contour with geometry for case 2

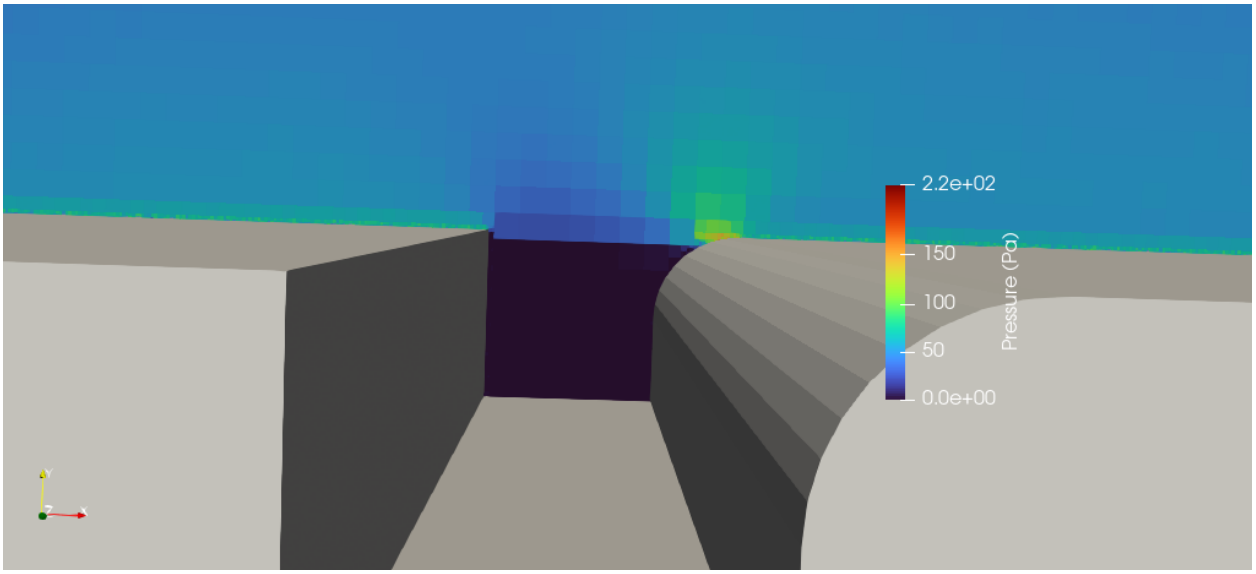


Fig. 5.11: Closeup of pressure contour for case 2

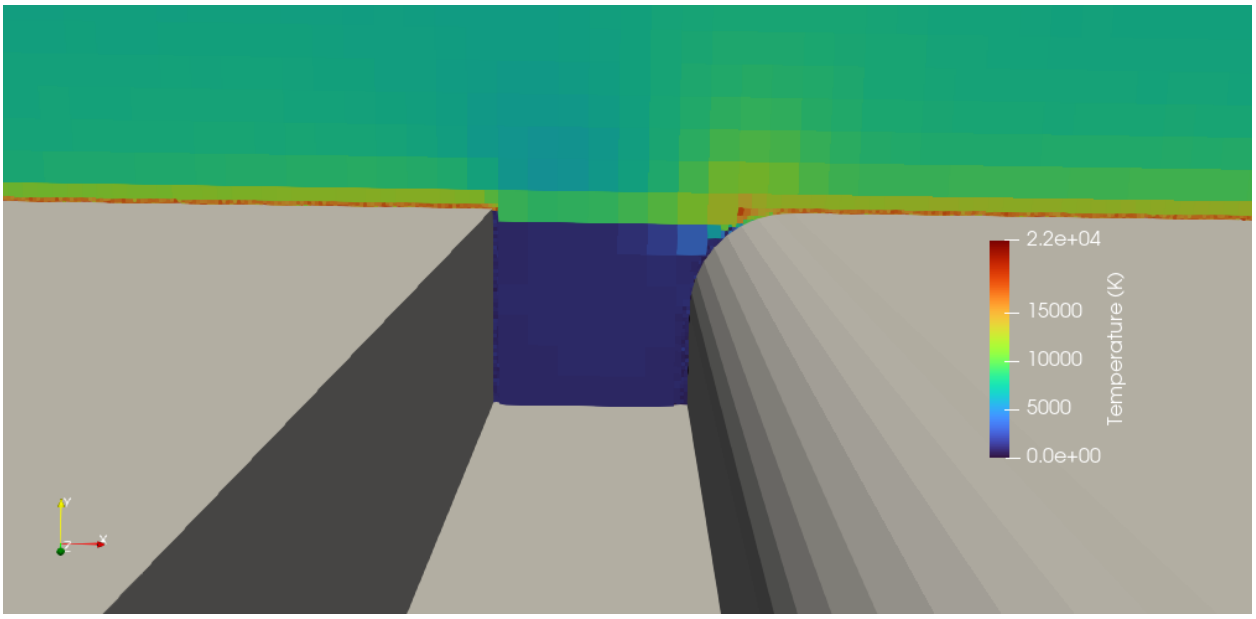


Fig. 5.12: Closeup of temperature contour with geometry for case 2

### 5.3.3. Case 3: incline on reattachment corner

The overview of the third case can be seen in figure 5.13. Just like the first two cases, the difference is unnoticeable. However, when observing closeup at the cavity in figures 5.14 and 5.15, the mass density ratio peak is higher as the regular cavity case by reaching up to about 5.2. In other words, there is more particle buildup that could cause deformation during reentry. There is also a pressure buildup in the reattachment corner as well with the largest value reaching up to 167.031 Pa, which is not as large as the first case. This can be seen in figure 5.16. However, the second configuration has a smaller peak pressure value than this last configuration. Also, for the temperature contour (seen in figure 5.17), the peak temperature value reaches to about 19230.2 K. Just like the pressure contour, the third cavity configuration has a smaller peak value in comparison to the regular cavity. However, the second configuration still has a reduced temperature value in comparison to this case. Overall, the third case creates a negative outcome for reducing the aerodynamic heating.

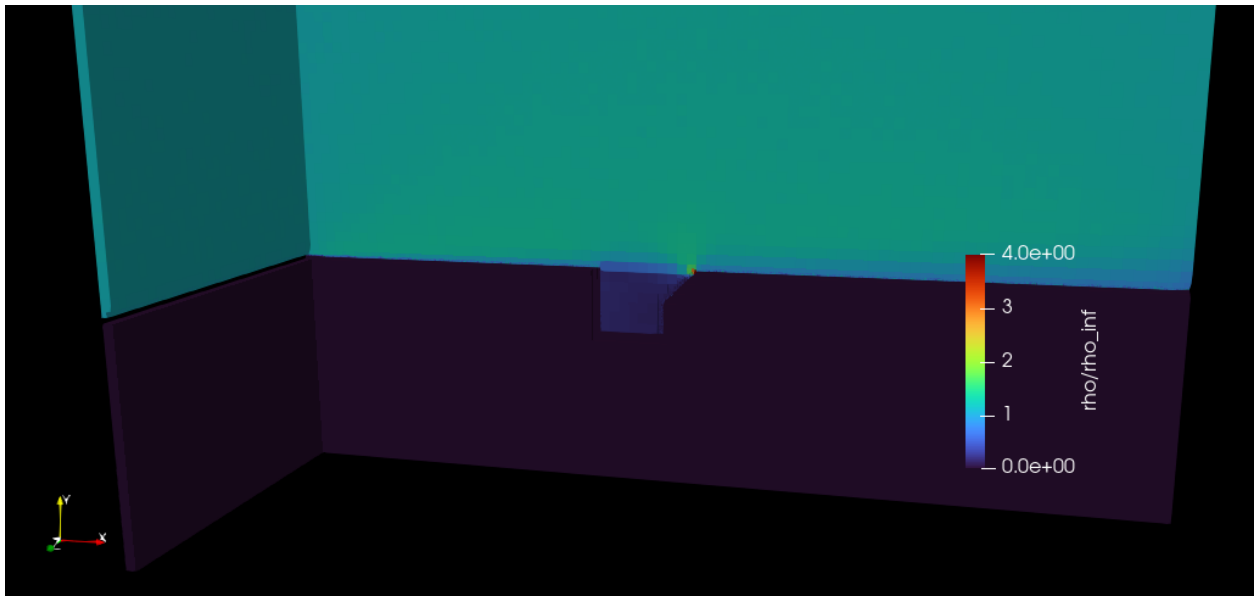


Fig. 5.13: Overview of mass density ratio contour for case 3

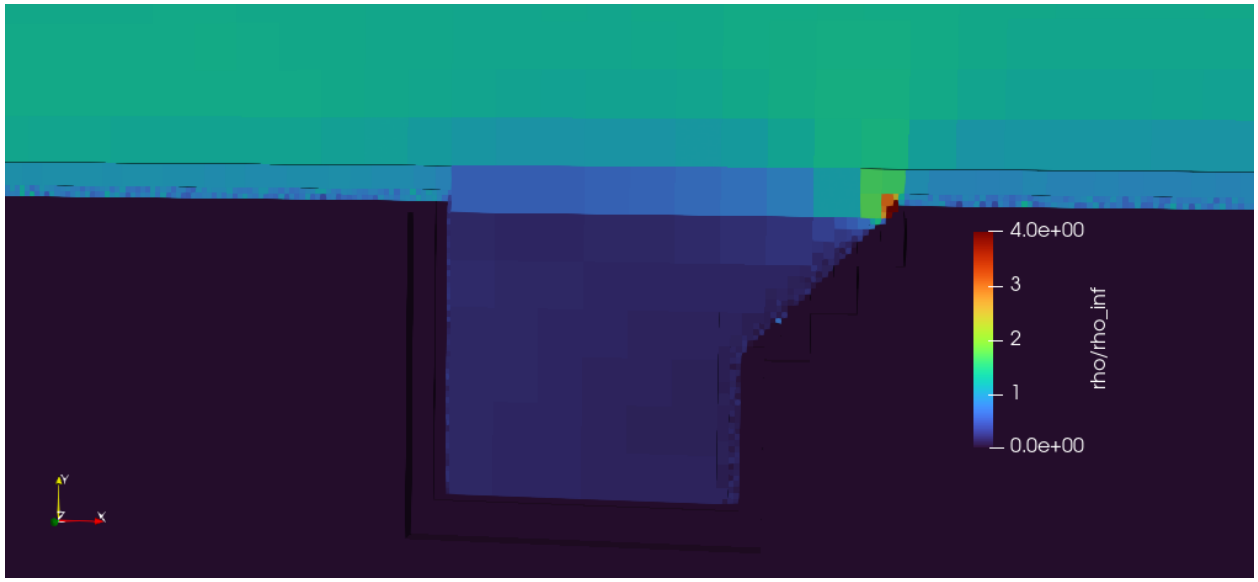


Fig. 5.14: Closeup of mass density ratio contour for case 3

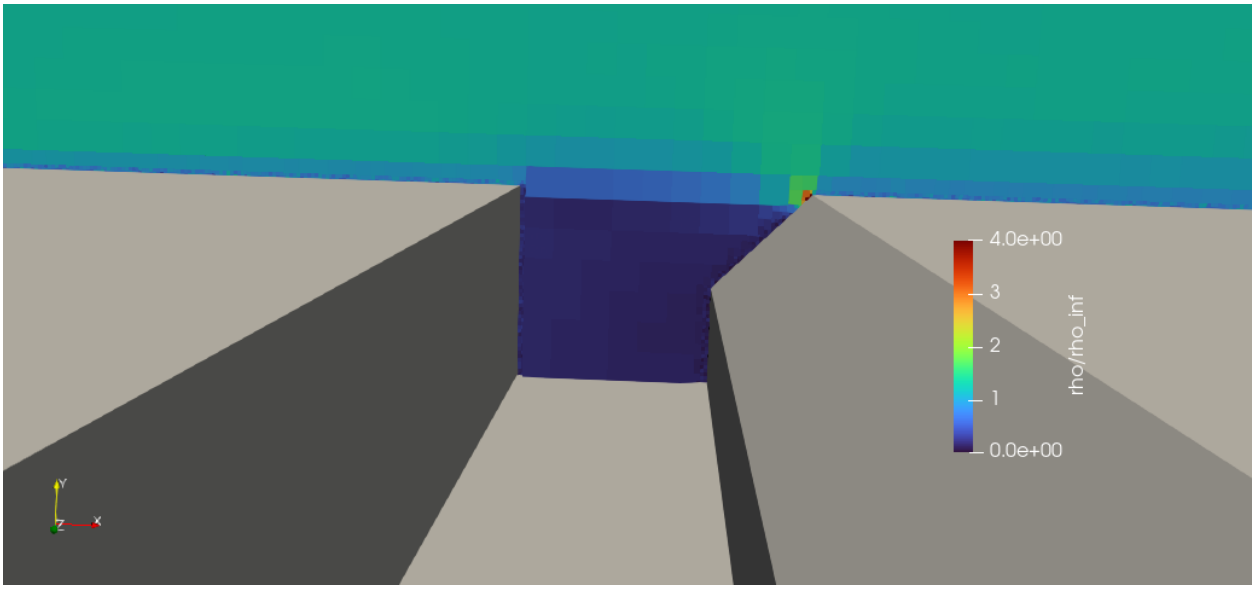


Fig. 5.15: Closeup of mass density ratio contour with geometry for case 3

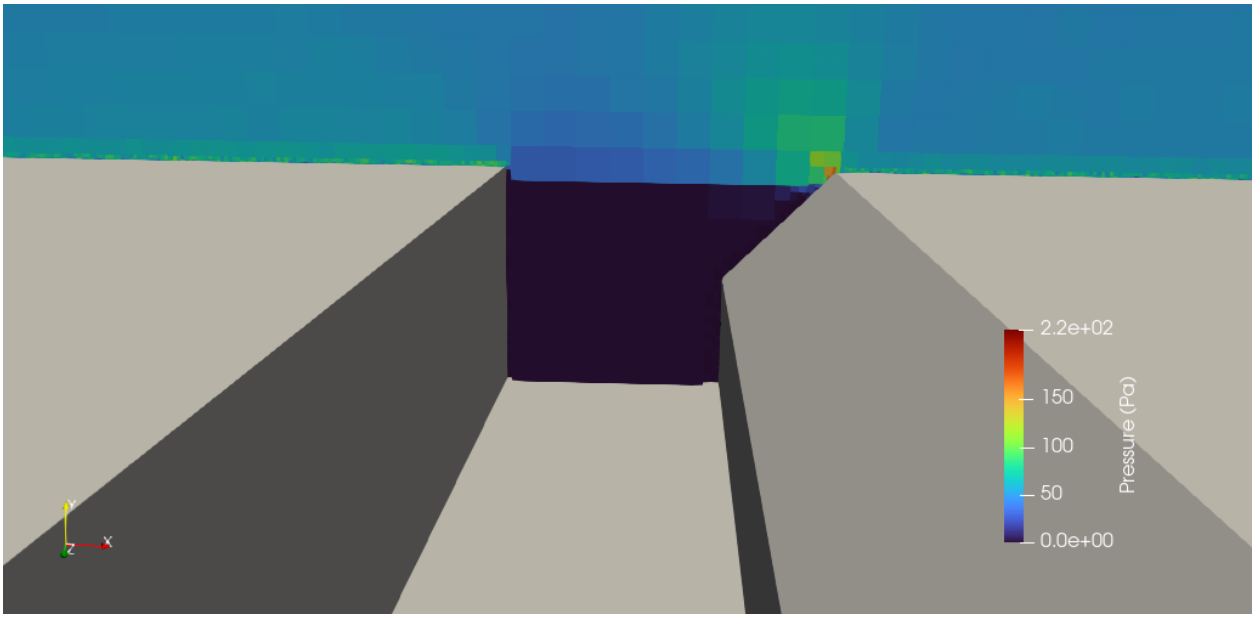


Fig. 5.16: Closeup of pressure contour with geometry for case 3

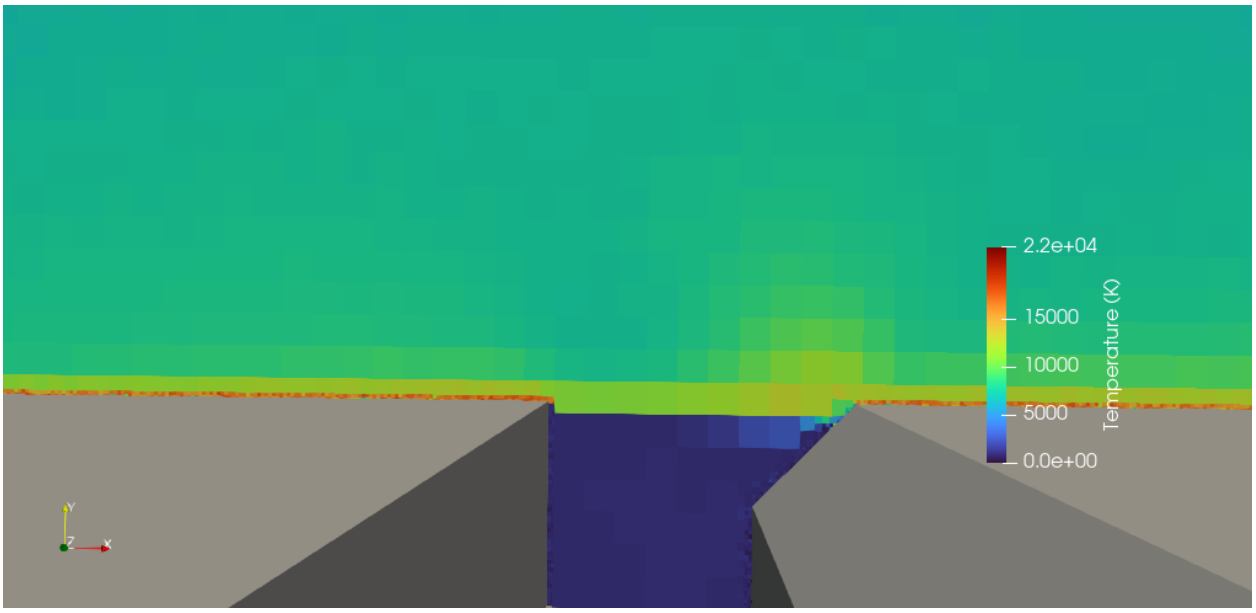


Fig. 5.17: Closeup of temperature contour with geometry for case 3

# 6. Conclusion

The project was able to showcase which configuration works best to reduce the aerodynamic heating of the TPS cavity given the DSMC simulation, which is rounding the reattachment corner. There is more buildup in the corners for the reattachment corner with the incline and regular cavity than the rounded corner case. Since there is more mass and pressure buildup from the particles, this can cause deformation of the surface. Also, the temperature is the lowest for the second configuration in comparison to the two other cases. Thus, rounding the reattachment corner is the most optimal configuration.

Despite the project being able to accomplish its main goal, there are a couple of recommendations that can be implemented in the future to improve the performance of the simulations. One of them is to decrease the size of the unrefined cells to capture the flow behavior. As seen in chapter 4, this increases the number of cells in all directions (assuming the domain size is kept constant) and the number of simulated particles in the domain. Although these would improve the performance, there will also be a rise computational cost. This would mean obtaining and utilizing a powerful computer with a large amount memory and cores.

Another way to improve the project is to simulate ablation of the TPS tiles. Generally, during atmospheric reentry, these tiles contain ablative material that would burn off to reduce the heating of the spacecraft. This method of improving the project would involve researching into different types of TPS material and determining the properties needed for SPARTA to generate the material and run the simulation. Also, just like adding grid cells and simulated particles, increasing the amount of surface elements requires more memory for the computer.

# References

[1] Cleland, J., and Iannetti, F., "Thermal Protection System of the Space Shuttle," NASA CR-4227, 1989.

[2] Bird, G. A., "Introduction to DSMC Method," The DSMC Method, Wiley, Sydney, 2013, pp.13-32.

[3] Josyula, E., and Burt, J., "Review of Rarefied Gas Effects in Hypersonic Applications," NASA, Jan 2011.

[4] Schouler, M., Prévereaud, Y., and Mieussens, L., "Survey of flight and numerical data of hypersonic rarefied flows encountered in Earth orbit and Atmospheric Reentry," Progress in Aerospace Sciences, vol. 118, Oct. 2020.

[5] Anderson, J. D., Hypersonic and high-temperature gas dynamics, Reston: American Institute of Aeronautics and Astronautics, 2006.

[6] Pearlman, R. Z., "NASA offers Space Shuttle tiles to schools," Space.com. Retrieved 23 May 2024 from https://www.space.com/9641-nasa-offers-space-shuttle-tiles-schools.html.

[7] Jin, X., Wang, B., Cheng, X., Wang, Q., and Huang, F., "Effects of corner rounding on aerothermodynamic properties in rarefied hypersonic flows over an open cavity," Aerospace Science and Technology, vol. 110, Mar. 2021.

[8] Jin, X., Cheng, X., Wang, Q., and Wang, B., "Numerical Analysis of rarefied hypersonic flows over inclined cavities," International Journal of Heat and Mass Transfer, vol. 214, Nov. 2023.

[9] Guo, G., & Luo, Q. (2019). Flowfield structure characteristics of the hypersonic flow over a cavity: From the continuum to the transition flow regimes. Acta Astronautica, 161, 87–100. https://doi.org/10.1016/j.actaastro.2019.05.023

[10] Guo, G., Jiang, S., Chen, H., and Zhu, L., "Influence of flow control on aerodynamic Properties of an open cavity in rarefied hypersonic flows," Acta Astronautica, vol. 191, Feb. 2022, pp. 404–416.

[11] Jiang, Q., Cai, G., Chen, Y., Yuan, J., He, B., and Liu, L., "Effects of cavity shapes and sizes on rarefied hypersonic flows," International Journal of Mechanical Sciences, vol. 245, May 2023.

[12] Guo, G., Luo, Q., and Wu, J., "Numerical investigation on flow and Aerothermal characteristics of rarefied hypersonic flows over slender cavities," Aerospace Science and Technology, vol. 144, Jan. 2024.

[13] Borner, A., and Liechty, D., “Analysis 101: Intro to DSMC,” EDL Summer Seminar, 2024.

[14] Gressman, P. T., and Strain, R. M., “Global classical solutions of the Boltzmann equation with long-range interactions,” Proceedings of the National Academy of Sciences, vol. 107, Mar. 2010, pp. 5744–5749.

[15] Bird, G. A., “Introduction to DSMC Method,” *The DSMC Method*, Wiley, Sydney, 2013, pp.13-32.

[16] Xiao, H., Shang, Y., and Wu, D., “DSMC simulation and experimental validation of shock interaction in hypersonic low-density flow,” The Scientific World Journal, vol. 2014, 2014, pp. 1–10.

[17] Acharya, T., Falgoust, J., Rasmussen, R., and Martin, M. J., “Disk spin-down measurements in the free-molecular flow regime: A new method for measurement of tangential momentum accommodation coefficients,” Vacuum, vol. 126, Apr. 2016, pp. 70–79.

[18] “Sparta Users Manual”, Retrieved 23 May 2024 from https://sparta.github.io/doc/Manual.html.

[19] U.S. standard atmosphere, 1976, Washington: National Oceanic and Atmospheric Administration, 1976.

[20] Bird, G. A., *Molecular gas dynamics and the direct simulation of gas flows*, Oxford: Clarendon Press, 1994.

[21] Hasanayn, F., Holland, P. L., Goldman, A. S., and Miller, A. J. M., “Lewis structures and the bonding classification of end-on bridging dinitrogen transition metal complexes,” *Journal of the American Chemical Society*. Retrieved 23 May 2024 from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9983020/#ref33.

[22]
Lakkis, I. and Shamseddine, M., “A New Parallel Adaptive DSMC Algorithm for Unsteady Flows in Complex Geometries”, *The Twelfth International Conference on Quantum, Nano/Bio, and Micro Technologies,* pp. 1-6.

[23] Roohi, E., and Darbandi, M., “Recommendations on performance of parallel DSMC algorithm in solving subsonic nanoflows,” *Applied Mathematical Modelling*, vol. 36, May 2012, pp. 2314–2321.

[24] Rader, D. J., Gallis, M. A., Torczynski, J. R., and Wagner, W., “Direct simulation monte carlo convergence behavior of the hard-sphere-gas thermal conductivity for Fourier heat flow,” *Physics of Fluids*, vol. 18, Jul. 2006.

# Appendix A: SPARTA Surface Files

For Case 1:

# SPARTA surface file, from STL file 3DCavityTest.STL with name 3DCavityTest

16 points
28 triangles

Points

1 5.300000e-02 1.700000e-02 2.000000e-02
2 5.300000e-02 2.000000e-02 2.000000e-02
3 5.300000e-02 1.700000e-02 0.000000e+00
4 5.300000e-02 2.000000e-02 0.000000e+00
5 5.000000e-02 1.700000e-02 2.000000e-02
6 5.000000e-02 1.700000e-02 0.000000e+00
7 5.000000e-02 2.000000e-02 2.000000e-02
8 5.000000e-02 2.000000e-02 0.000000e+00
9 0.000000e+00 2.000000e-02 2.000000e-02
10 0.000000e+00 2.000000e-02 0.000000e+00
11 0.000000e+00 0.000000e+00 2.000000e-02
12 0.000000e+00 0.000000e+00 0.000000e+00
13 1.000000e-01 0.000000e+00 2.000000e-02
14 1.000000e-01 0.000000e+00 0.000000e+00
15 1.000000e-01 2.000000e-02 2.000000e-02
16 1.000000e-01 2.000000e-02 0.000000e+00

Triangles

1 1 2 3
2 3 2 4
3 5 1 6
4 6 1 3
5 7 5 8
6 8 5 6
7 9 7 10
8 10 7 8
9 11 9 12
10 12 9 10
11 13 11 14
12 14 11 12
13 15 13 16
14 16 13 14
15 2 15 4
16 4 15 16

17 7 9 5
18 5 9 11
19 5 11 1
20 1 11 13
21 1 13 2
22 2 13 15

For Case 2:

# SPARTA surface file, from STL file 3DCavityRoundCorner.STL with name 3DCavityRoundCorner

34 points
64 triangles

Points

1 5.300000e-02 1.850000e-02 0.000000e+00
2 5.300000e-02 1.700000e-02 0.000000e+00
3 5.300000e-02 1.850000e-02 2.000000e-02
4 5.300000e-02 1.700000e-02 2.000000e-02
5 5.000000e-02 1.700000e-02 2.000000e-02
6 5.000000e-02 1.700000e-02 0.000000e+00
7 5.000000e-02 2.000000e-02 2.000000e-02
8 5.000000e-02 2.000000e-02 0.000000e+00
9 0.000000e+00 2.000000e-02 2.000000e-02
10 0.000000e+00 2.000000e-02 0.000000e+00
11 0.000000e+00 0.000000e+00 2.000000e-02
12 0.000000e+00 0.000000e+00 0.000000e+00
13 1.000000e-01 0.000000e+00 2.000000e-02
14 1.000000e-01 0.000000e+00 0.000000e+00
15 1.000000e-01 2.000000e-02 2.000000e-02
16 1.000000e-01 2.000000e-02 0.000000e+00
17 5.450000e-02 2.000000e-02 2.000000e-02
18 5.450000e-02 2.000000e-02 0.000000e+00
19 5.320096e-02 1.925000e-02 2.000000e-02
20 5.309046e-02 1.901303e-02 2.000000e-02
21 5.302279e-02 1.876047e-02 2.000000e-02
22 5.335093e-02 1.946418e-02 2.000000e-02
23 5.423953e-02 1.997721e-02 2.000000e-02
24 5.398697e-02 1.990954e-02 2.000000e-02
25 5.375000e-02 1.979904e-02 2.000000e-02
26 5.353582e-02 1.964907e-02 2.000000e-02
27 5.302279e-02 1.876047e-02 0.000000e+00
28 5.309046e-02 1.901303e-02 0.000000e+00
29 5.320096e-02 1.925000e-02 0.000000e+00

30 5.335093e-02 1.946418e-02 0.000000e+00
31 5.353582e-02 1.964907e-02 0.000000e+00
32 5.375000e-02 1.979904e-02 0.000000e+00
33 5.398697e-02 1.990954e-02 0.000000e+00
34 5.423953e-02 1.997721e-02 0.000000e+00

Triangles

1 1 2 3
2 3 2 4
3 5 4 6
4 6 4 2
5 7 5 8
6 8 5 6
7 9 7 10
8 10 7 8
9 11 9 12
10 12 9 10
11 13 11 14
12 14 11 12
13 15 13 16
14 16 13 14
15 17 15 18
16 18 15 16
17 3 4 17
18 17 4 15
19 7 9 5
20 5 9 11
21 5 11 4
22 4 11 13
23 4 13 15
24 19 20 21
25 19 21 22
26 3 17 21
27 21 17 23
28 21 23 24
29 24 25 21
30 21 25 26
31 21 26 22
32 10 8 12
33 12 8 6
34 12 6 14
35 14 6 2
36 14 2 16
37 16 2 1
38 16 1 18

39 18 1 27
40 18 27 28
41 28 29 18
42 18 29 30
43 18 30 31
44 32 33 31
45 31 33 34
46 31 34 18
47 17 18 34
48 17 34 23
49 23 34 33
50 23 33 24
51 24 33 32
52 24 32 25
53 25 32 31
54 25 31 26
55 26 31 30
56 26 30 22
57 22 30 29
58 22 29 19
59 19 29 28
60 19 28 20
61 20 28 27
62 20 27 21
63 21 27 1
64 21 1 3

For Case 3:

# SPARTA surface file, from STL file 3DCavityInclinePlane.STL with name 3DCavityInclinePlane

16 points
24 triangles

Points

1 5.000000e-02 1.699964e-02 2.000000e-02
2 5.300000e-02 2.000000e-02 2.000000e-02
3 5.000000e-02 1.699964e-02 -6.938894e-18
4 5.300000e-02 2.000000e-02 -6.938894e-18
5 5.000000e-02 2.000000e-02 2.000000e-02
6 5.000000e-02 2.000000e-02 0.000000e+00
7 5.000000e-02 1.699964e-02 0.000000e+00
8 0.000000e+00 2.000000e-02 2.000000e-02
9 0.000000e+00 2.000000e-02 0.000000e+00

10 0.000000e+00 0.000000e+00 2.000000e-02
11 0.000000e+00 0.000000e+00 0.000000e+00
12 1.000000e-01 0.000000e+00 2.000000e-02
13 1.000000e-01 0.000000e+00 0.000000e+00
14 1.000000e-01 2.000000e-02 2.000000e-02
15 1.000000e-01 2.000000e-02 0.000000e+00
16 5.300000e-02 2.000000e-02 0.000000e+00

Triangles

1 1 2 3
2 3 2 4
3 5 1 6
4 6 1 7
5 8 5 9
6 9 5 6
7 10 8 11
8 11 8 9
9 12 10 13
10 13 10 11
11 14 12 15
12 15 12 13
13 2 14 16
14 16 14 15
15 1 5 8
16 14 2 12
17 12 2 1
18 12 1 10
19 10 1 8
20 7 16 15
21 9 6 11
22 11 6 7
23 11 7 13
24 13 7 15

# Appendix B: Matlab Code for Simulation Setup

```
%% DSMC Masters Project Appendix B

% Description: The purpose of this code is to perform calculations
%              for setting up the DSMC SPARTA input file. This step is for
%              Chapter 4 of the report.
%
%------------------------------------------------------------------------
%
%% Setting up number of cells in x, y, and z direction
%
% Given:
%       L_dom_x: the length of the domain in the x direction, scalar, units
%                in m
%       L_dom_y: the length of the domain in the y direction, scalar, units
%                in m
%       L_dom_z: the length of the domain in the z direction, scalar, units
%                in m
%       lambda_inf: freestream mean-free path, scalar, units in m
%
clc;clear;
lambda_inf = 4.402E-3; % given from Ch 3
L_dom_min_x = 0.035;
L_dom_min_y = 0.01;
L_dom_min_z = 0;
L_dom_max_x = 0.075;
L_dom_max_y = 0.04;
L_dom_max_z = 0.02;
L_dom_x = L_dom_max_x-L_dom_min_x;
L_dom_y = L_dom_max_y-L_dom_min_y;
L_dom_z = L_dom_max_z-L_dom_min_z;
cell_z = lambda_inf/10;
cell_y = lambda_inf/10;
cell_x = lambda_inf/10;
N_cell_x = L_dom_x/cell_x
N_cell_y = L_dom_y/cell_y
N_cell_z = L_dom_z/cell_z
N_cells = N_cell_z*N_cell_y*N_cell_x
%% Setting up time step
%
% Given:
%       k: boltzmann's constant, scalar, units are in J/K
```

```
%       T_inf: freestream temperature, scalar, units are in K
%       C_N2: concentration of nitrogen gas, scalar, unitless
%       C_O2: concentration of oxygen gas, scalar, unitless
%       M_N2: molar mass of nitrogen gas, scalar, units are in g/mol
%       M_O2: molar mass of oxygen gas, scalar, units are in g/mol
%       N_A: avogadro's number, scalar, units are in particles/mol

C_N2 = 0.763;
C_O2 = 0.237;
M_N2 = 28;
M_O2 = 32;
M_air = C_N2*M_N2 + C_O2*M_O2; % molar mass of air, scalar, units are in g/mol
N_A = 6.023E23;
m_air = M_air/N_A; % mass of the air, scalar, units are kg per particle
T_inf = 198.64;
k = 1.38E-23;
c_bar = sqrt((2*k*T_inf)/(m_air));
tau_bar = lambda_inf/c_bar;
dt = tau_bar/10;
%% Setting up fnum
%
% Given:
%       N_par: number of particles per cell, scalar, units are part. per
%            cell
%       n_inf: freestream number density, scalar, units are in part./m^3
%
n_inf = 3.837E20;
V_dom = L_dom_x*L_dom_y*L_dom_z; % volume of domain, scalar, units are in m^3
N_real = n_inf*V_dom;
N_par = 10;
N_sim = N_par*N_cells;
fnum = N_real/N_sim
```

# Appendix C. Input Files for Each Case

```
# This input script is for the 3D regular cavity and no modifications.
# The values for the grid, boundary conditions, etc. from the report are inputted in this file.

# Initialization

seed            12345
dimension       3
global          gridcut 0.0 comm/sort yes
boundary        o r r

# Setting up simulation domain with grid

create_box      0.035 0.075 0.01 0.04 0 0.02
create_grid     90 64 45 # number of cells in x, y, and z respectively
balance_grid    rcb cell
global          nrho 3.837e20 fnum 3.273e9

# Defining species for air. Referencing article for composition that the article for rounded corner
uses, which uses N2 and O2

species         air.species N2 O2 # referencing the article that uses N2 and O2
mixture                 air N2 O2 vstream 7565.7 0 0 temp 198.63 # freestream velocity
and temperature are referenced from CH 3
mixture         air N2 frac 0.763
mixture         air O2 frac 0.237

# Defining Geometry

read_surf       data.3DCavityTest clip

# Particle-to-Surf Collision

surf_collide    1 diffuse 1000 1.0 # All surf collide with the wall temp being 1000 K
surf_react      2 prob air.surf
surf_modify     all collide 1
surf_modify     all react 2

# Particle-to-Particle Collision

collide         vss air air.vss
collide_modify          vremax 100 yes vibrate smooth # particles are unsteady
```

```
react                   tce air.tce

# Setting up Particles

fix                     in emit/face air xlo twopass

timestep                4.12e-5

stats                   1
stats_style             step cpu np nattempt ncoll nscoll nscheck maxlevel

run                     100 # Checking to see if it works and how it's running

# Adapting cells around surface (will see if i need to use python tools or not)

adapt_grid              all refine surf all 0.000001 iterate 5 # adapts grid once, doesn't keep
iterating like the fix adapt command
balance_grid            rcb cell
#write_surf             surf.3DCavRef3
run                     5000

# Computing properties per surface such as normal and shear pressure

compute                         surf_press surf all air press px py pz shx shy shz
fix                     surf_press ave/surf all 1 500 500 c_surf_press[*] ave running
dump                    surf_press surf all 1000 Surf3DCavZoomRefineSurfaceFinal*. id
f_surf_press[*]
write_surf              3DCavZoomRefineSurfaceFinal.surf # for post processing

# Computing Properties per Grid such as temp, press, dens, heat flux, etc while averaging the
data to get the results later

compute                 therm_grid thermal/grid all air temp press
compute                         grid_thermprop grid all air n nrho massrho u v w temp trot tvib
variable                massrhoratio grid c_grid_thermprop[3]/1.84e-5

fix                     avg_grid ave/grid all 1 500 500 c_therm_grid[*] c_grid_thermprop[*]
v_massrhoratio ave running
dump                    dump_thermgrid grid all 1000 Thermal3DCavZoomRefineSurfaceFinal*.
id f_avg_grid[*]
write_grid              Therm3DCavZoomRefineSurfaceFinal.grid # writing grid data with all
IDs for post processing
```

```
compute         heat_grid eflux/grid all air heatx heaty heatz
fix             heat_grid ave/grid all 1 500 500 c_heat_grid[*] ave running
dump            heat_grid grid all 1000 Heat3DCavZoomRefineSurfaceFinal*. id
f_heat_grid[*]
write_grid      Heat3DCavZoomRefineSurfaceFinal.grid # writing grid data with all IDs
for post processing

run             10000

# This input script is for the 3D cavity with rounded reattachment corner.
# The values for the grid, boundary conditions, etc. from the report are inputed in this file.

# Initialization

seed            12345
dimension       3
global          gridcut 0.0 comm/sort yes
boundary        o r r

# Setting up simulation domain with grid

create_box      0.035 0.075 0.01 0.04 0 0.02
create_grid     90 64 45 # number of cells in x, y, and z respectively
balance_grid    rcb cell
global          nrho 3.837e20 fnum 3.273e9

# Defining species for air. Referencing article for composition that the article for rounded corner
uses, which uses N2 and O2

species         air.species N2 O2 # referencing the article that uses N2 and O2
mixture         air N2 O2 vstream 7565.7 0 0 temp 198.63 # freestream velocity
and temperature are referenced from CH 3
mixture         air N2 frac 0.763
mixture         air O2 frac 0.237

# Defining Geometry

read_surf       data.3DCavityRounded clip # the clipping function clips the surface when
it takes over the domain

# Particle-to-Surf Collision

surf_collide    1 diffuse 1000 1.0 # All surf collide with the wall temp being 1000 K
```

```
surf_react          2 prob air.surf
surf_modify         all collide 1
surf_modify         all react 2

# Particle-to-Particle Collision

collide             vss air air.vss
collide_modify          vremax 100 yes vibrate smooth # particles are unsteady
react               tce air.tce

# Setting up Particles

fix                 in emit/face air xlo twopass

timestep            4.12e-5

stats               1
stats_style         step cpu np nattempt ncoll nscoll nscheck maxlevel

run                 100 # Checking to see if it works and how it's running

# Adapting cells around surface (will see if i need to use python tools or not)

adapt_grid          all refine surf all 0.000001 iterate 5 # adapts grid once, doesn't keep
iterating like the fix adapt command
balance_grid        rcb cell
#write_surf         surf.3DCavRef3
run                 2000

# Computing properties per surface such as normal and shear pressure

compute                 surf_press surf all air press px py pz shx shy shz
fix                 surf_press ave/surf all 1 500 500 c_surf_press[*] ave running
dump                surf_press surf all 1000 Surf3DCavRoundZoomRefineSurfaceFinal*. id
f_surf_press[*]
write_surf          3DCavZoomRoundRefineSurfaceFinal.surf # for post processing

# Computing Properties per Grid such as temp, press, dens, heat flux, etc while averaging the
data to get the results later

compute             therm_grid thermal/grid all air temp press
compute                 grid_thermprop grid all air n nrho massrho u v w temp trot tvib
variable            massrhoratio grid c_grid_thermprop[3]/1.84e-5
```

```
fix             avg_grid ave/grid all 1 500 500 c_therm_grid[*] c_grid_thermprop[*]
v_massrhoratio ave running
dump            dump_thermgrid grid all 1000
Thermal3DCavRoundZoomRefineSurfaceFinal*. id f_avg_grid[*]
write_grid      Therm3DCavRoundZoomRefineSurfaceFinal.grid

compute             heat_grid eflux/grid all air heatx heaty heatz
fix             heat_grid ave/grid all 1 500 500 c_heat_grid[*] ave running
dump            heat_grid grid all 1000 Heat3DCavRoundZoomRefineSurfaceFinal*. id
f_heat_grid[*]
write_grid      Heat3DCavRoundZoomRefineSurfaceFinal.grid

run             10000

# This input script is for the 3D inclined cavity at reattachment corner.
# The values for the grid, boundary conditions, etc. from the report are inputted in this file.

# Initialization

seed            12345
dimension       3
global          gridcut 0.0 comm/sort yes
boundary        o r r

# Setting up simulation domain with grid

create_box      0.035 0.075 0.01 0.04 0 0.02
create_grid     90 64 45 # number of cells in x, y, and z respectively
balance_grid    rcb cell
global          nrho 3.837e20 fnum 3.273e9

# Defining species for air. Referencing article for composition that the article for rounded corner
uses, which uses N2 and O2

species         air.species N2 O2 # referencing the article that uses N2 and O2
mixture             air N2 O2 vstream 7565.7 0 0 temp 198.63 # freestream velocity
and temperature are referenced from CH 3
mixture         air N2 frac 0.763
mixture         air O2 frac 0.237

# Defining Geometry
```

```
read_surf               data.3DCavityInclineTest2 clip

# Particle-to-Surf Collision

surf_collide            1 diffuse 1000 1.0 # All surf collide with the wall temp being 1000 K
surf_react              2 prob air.surf
surf_modify             all collide 1
surf_modify             all react 2

# Particle-to-Particle Collision

collide                 vss air air.vss
collide_modify                  vremax 100 yes vibrate smooth # particles are unsteady
react                   tce air.tce

# Setting up Particles

fix                     in emit/face air xlo twopass

timestep                4.12e-5

stats                   1
stats_style             step cpu np nattempt ncoll nscoll nscheck maxlevel

run                     100 # Checking to see if it works and how it's running

# Adapting cells around surface (will see if i need to use python tools or not)

adapt_grid              all refine surf all 0.000001 iterate 5 # adapts grid once, doesn't keep
iterating like the fix adapt command
balance_grid            rcb cell
#write_surf             surf.3DCavRef3
run                     1000

# Computing properties per surface such as normal and shear pressure

compute                         surf_press surf all air press px py pz shx shy shz
fix                     surf_press ave/surf all 1 500 500 c_surf_press[*] ave running
dump                    surf_press surf all 1000 Surf3DCavInclineZoomRefineSurfaceFinal*. id
f_surf_press[*]
write_surf              3DCavZoomInclineRefineSurfaceFinal.surf # for post processing
```

```
# Computing Properties per Grid such as temp, press, dens, heat flux, etc while averaging the data to get the results later

compute         therm_grid thermal/grid all air temp press
compute             grid_thermprop grid all air n nrho massrho u v w temp trot tvib
variable        massrhoratio grid c_grid_thermprop[3]/1.84e-5

fix             avg_grid ave/grid all 1 500 500 c_therm_grid[*] c_grid_thermprop[*]
v_massrhoratio ave running
dump
dump_thermgrid grid all 1000 Thermal3DCavInclineZoomRefineSurfaceFinal*. id
f_avg_grid[*]
write_grid      Therm3DCavInclineZoomRefineSurfaceFinal.grid

compute             heat_grid eflux/grid all air heatx heaty heatz
fix             heat_grid ave/grid all 1 500 500 c_heat_grid[*] ave running
dump            heat_grid grid all 1000 Heat3DCavInclineZoomRefineSurfaceFinal*. id
f_heat_grid[*]
write_grid      Heat3DCavRoundInclineRefineSurfaceFinal.grid

run             5000
```