Space Station Air Evacuation Time Analysis

Team 351, Problem A

November 3, 2024

Abstract

The space station, as a key part of the aerospace field, played an important role in various aspects. The topic requires us to model the gas evacuation from the capsule due to a micrometeorite impact. We propose assumptions about gas patterns, heat transfer, and station morphology to help us simplify the computational and modeling process for the specific environment of a space station in the universe. **First of all**, we propose a **spatial dynamics model**, assuming the gas as a particle flow, and by applying the Maxwell distribution of gas molecular velocities, combined with the kinetic energy and momentum of air molecules, we arrive at the final time evolution equations. **However**, due to the calculation of the Knudsen number, we found that in the state of the space station, the gas should be analyzed as a continuous flow rather than a particle flow in order to obtain more accurate results. **Therefore**, we improved the original modeling method to propose a second model, the **continuous medium model**, which uses the continuous flow as the object and treats the whole process as an isentropic process, and analyzes the answer that is relatively more in line with the actual situation. **Eventually**, a series of conclusions can be drawn from the modeling as well as the collation and comparison analysis between the two scenarios. We obtained the scheme that different gas state assumptions should be made under different Knudsen numbers, and at the same time, we used the program to solve the differential equations to derive the equation for the change of gas pressure in the space station cabin with time. We brought the problem conditions into the solution, and finally concluded that the gas evacuation to 0.3 atm at 1cm aperture takes about **48000s**.

Key Words: Space Station, Aerodynamic, Knudsen Number, Critical Flow

Contents

1 Introduction **4**
1.1 Background 4
1.2 Analysis of the Problem 4

2 Assumptions **5**
2.1 Common Assumptions 5
2.2 Model 1: Molecular Dynamics Assumptions 5
2.3 Model 2: Continuum Flow Assumptions 5

3 Notations **6**

4 Overview of Our Works **7**

5 Model 1: Aerodynamic Model **8**
5.1 Basic Molecular Flux Analysis 8
5.2 Kinetic Energy Distribution 9
5.3 Space Station Recoil Analysis 10
5.3.1 Initial Conditions and Assumptions 10
5.3.2 Momentum Conservation 10
5.3.3 Energy Conservation 10
5.3.4 System Evolution Equations 11
5.3.5 Final Solutions 11
5.4 Dismiss the Space Station's Recoil 12
5.5 Calculating Result 12

6 Model 2: Continuous Medium Model **14**
6.1 Two Main Assumptions 14
6.1.1 Knudsen Number 14
6.1.2 Critical Flow 14
6.2 Analysis 15
6.2.1 Basic Governing Equations 15
6.2.2 Mass Conservation and Flow Rate Analysis 16
6.2.3 Critical Flow Conditions 16
6.2.4 Temperature-Pressure Evolution 17
6.2.5 Derivation of the Time Evolution Equation 17
6.2.6 Solution Process 17

6.2.7 Final Integration and Result . . . 17
6.2.8 Physical Interpretation and Implications . . . 18

7 Results Discussion **20**
7.1 Discussion on the Relation Between Time and Diameters . . . 20
7.2 Discussion on Two Models . . . 20
7.2.1 Comparison on Figures . . . 20
7.2.2 Analysis . . . 22

8 Model Evaluation **23**
8.1 Strengths . . . 23
8.2 Weaknesses . . . 23

9 Conclusion **24**

References **25**

A Appendix **26**
A.1 Figure4.py . . . 26
A.2 Figure6.py . . . 28
A.3 Figure7.py . . . 30
A.4 Figure8.py . . . 32
A.5 Figure9.py . . . 33

1 Introduction

1.1 Background

In recent years, Space stations are playing a more and more important role in our world. From scientific research to space exploration, space stations will help astronauts finish various tasks. Space stations provide a safe environment for astronauts to conduct experiments in the cosmic environment. With the development of relevant advanced technology, more and more space stations will be built to benefit technological development. However, the risk analysis of space stations' security remains a big challenge. Because there are many micrometeorites in the orbit of the space station, there may be some risks. When facing dangerous situations, whether astronauts can have enough time to cope with the emergency is a crucial topic. It requires us to quickly analyze if astronauts can solve the problem to keep them secure. Therefore, such modeling analysis is of great help in the design of the space station as well as in the simulation of the gas evacuation problem in the capsule.

1.2 Analysis of the Problem

The question asks us to analyze and estimate the time required to reduce the air pressure in the space station to 0.3 atm for different pinhole diameters. Meanwhile, the problem has listed some given pre-conditions, such as the volume and the shape of the space station, and the primitive pressure and temperature in the space station. According to the pre-condition and some proper assumptions, we will figure out the result by modeling the physics and solving differential equations.

- Proper gas modeling should be constructed
- The gas may cause a change in the speed of motion of the space station due to the conservation of momentum during the gas evacuation process.
- Air is made up of nitrogen, oxygen, and other gases. So we have to simplify the gas types and molecular forms to help analysis.
- To clearly describe gas's status during the gas leakage process, we should make more assumptions and force analysis.

2 Assumptions

2.1 Common Assumptions

To simplify our analysis while maintaining physical accuracy, we establish the following common assumptions for both models:

- The space station is adiabatic to the outer space
- The gas in the space station is considered the ideal gas.
- The external environment is perfect vacuum (pressure ≈ 0 Pa)
- All gas molecules are treated as diatomic molecules
- No rotation of the space station
- The gas inside the space station was always in equilibrium during the leak

2.2 Model 1: Molecular Dynamics Assumptions

For our molecular dynamics-based approach, we make the following specific assumptions:

- Energy and momentum of the system are conserved in molecular collisions

2.3 Model 2: Continuum Flow Assumptions

For our continuum mechanics-based approach, we make the following specific assumptions:

- The flow is isentropic at the orifice
- One-dimensional flow at the hole
- Critical flow conditions apply at the orifice
- Continuum flow assumptions valid (verified by Knudsen number)

3 Notations

Symbols	Description	Numeric value	Unit
v	Molecule Velocity	-	m/s
D_0	Diameter of the station	4	m
L	Length of the station	50	m
d	Diameter of the small hole	0.1	cm
ρ	Air density	-	kg/m^3
c	Speed of sound	343	m/s
A	Orifice area	7.854×10^{-5}	m^2
V	Volume of the station	12.566	m^3
γ	Adiabatic index	$\frac{7}{5}$	-
T	Temperature	-	K
T_0	Initial temperature	293.15	K
N	Number of gas molecules	-	-
n	Gas molecule density	-	m^{-3}
m	Gas molecule mass	4.81×10^{-26}	kg
k	Boltzmann constant	1.381×10^{-23}	J/K
R	Ideal gas constant	289	J/(kg·K)
C_d	Discharge coefficient	0.62	-
p_0	Initial pressure	1.013×10^5	Pa
p	Final pressure	-	Pa
t	Time	-	s
E_k	Average kinetic energy	-	J
dN	Change in the number of gas molecules	-	-
dP	Change in pressure	-	Pa
$d\rho$	Change in density	-	kg/m^3
$\dot{m}$	Mass flow rate	-	kg/s
K_n	Knudsen value	-	-

Table 1: List of Notations

4 Overview of Our Works

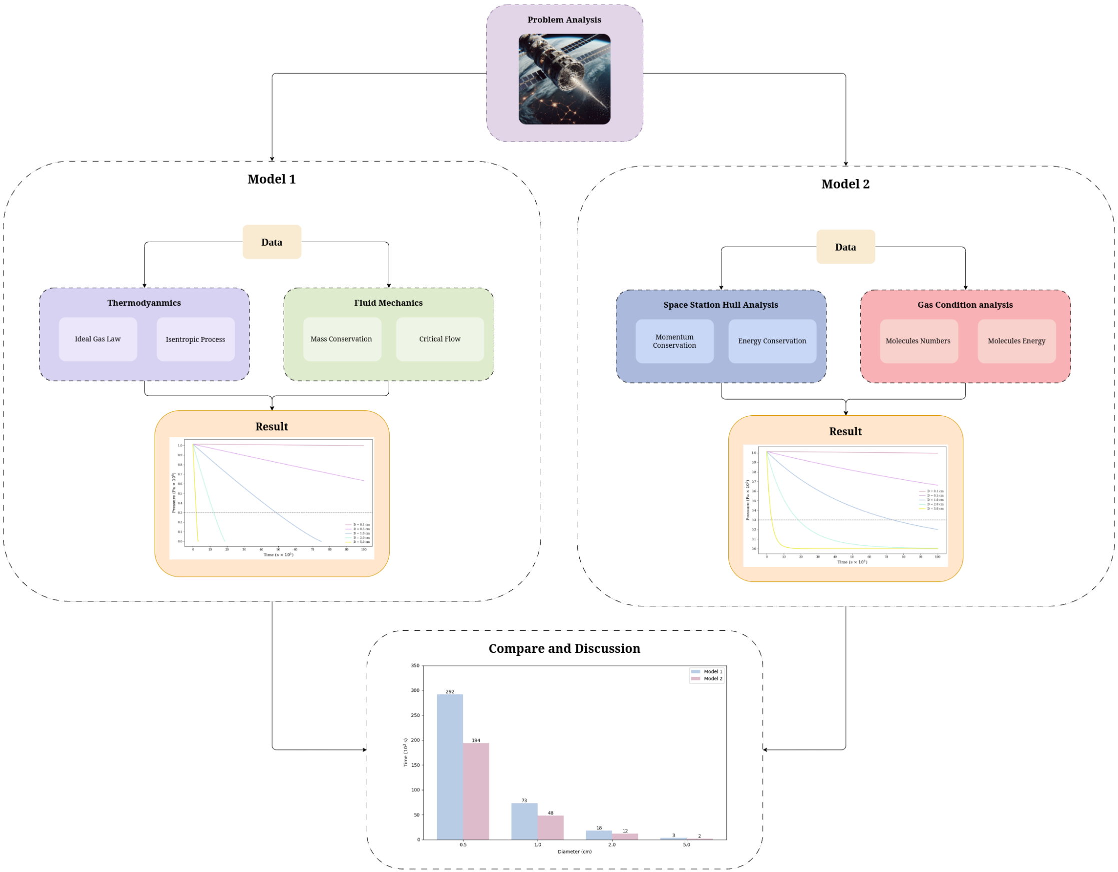

Figure 1: Basic Structure

- **Model 1**: Aerodynamic model; The gas is analyzed as a molecular flow, the kinetic energy and momentum of the gas is calculated, the number and kinetic energy of the escaping gas molecules are calculated, and the final result is obtained by solving the differential equations associatively.
- **Model 2**: Continuous Medium model; The gas is treated as a continuous flow and the entire isentropic process is analyzed and calculated so that the equation for the change in time under that condition can be derived and brought to numerical values to produce results.
- **Comparison**: Based on the calculation and discussion of the Knudsen number, we discuss the assumptions of the two models for the type of gas and conclude that model 2 is the more realistic scenario.

5 Model 1: Aerodynamic Model

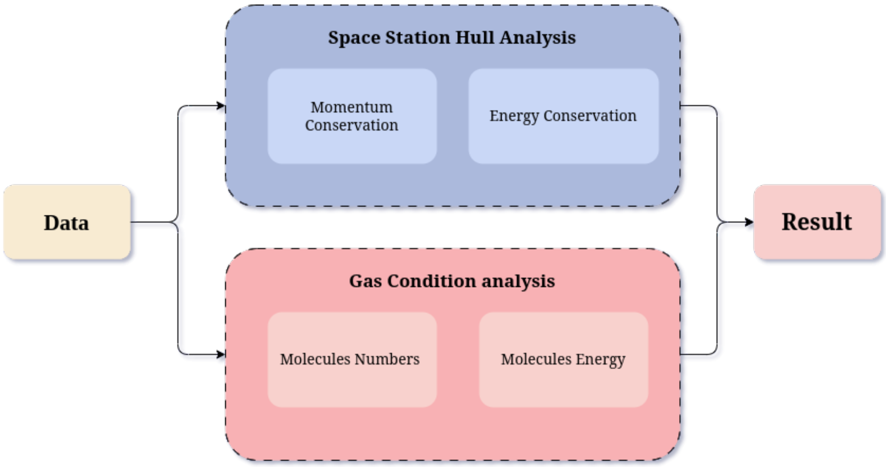

Figure 2: Model 1 overview diagram

5.1 Basic Molecular Flux Analysis

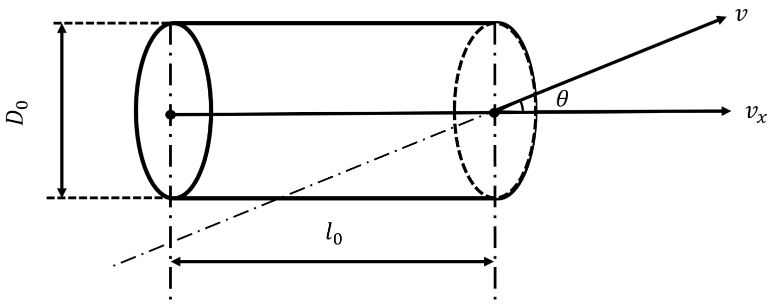

Figure 3: Schematic diagram of Aerodynamic model

From Figure4, let's assume the small hole is perpendicular to the x direction. Inside the space station, near the small hole, gas molecules move at a certain speed v in a certain direction, and their velocity component in the x-direction is v_x.

We should also note that:

$$A = \frac{\pi D^2}{4}, \tag{1}$$

$$V = \frac{\pi D_0^2 L}{4}. \tag{2}$$

These two parameters will be frequently used below.

The number of gas molecules that escape through the small hole within a time interval dt is equal to the number of gas molecules within the volume $v_x Adt$ on the side of the small hole, where the velocity component perpendicular to the small hole is within the range from v_x to $v_x + dv_x$. The number of gas molecules that meet this condition in the cylinder is $n_{v_x} Av_x dtdv_x$, where n_{v_x} is the number density of gas molecules with velocity v_x, and its relationship with the number density n is:

$$n_{v_x} = nf(v_x) = n\sqrt{\frac{m}{2\pi kT}} \exp\left(-\frac{mv_x^2}{2kT}\right). \tag{3}$$

The average number of molecules of gas emitted per unit time per unit area at the hole is [1]:

$$N_{\text{Leap}} = \int_0^\infty n_{v_x} v_x dv_x = \sqrt{\frac{m}{2\pi kT}} \int_0^\infty v_x n \exp\left(-\frac{mv_x^2}{2kT}\right) dv_x = n\sqrt{\frac{kT}{2\pi m}}. \tag{4}$$

5.2 Kinetic Energy Distribution

When the gas temperature is T, the average kinetic energy of the expelled gas molecules is [2]:

$$\bar{E}_k = \frac{1}{N_{\text{Leap}}} \int_0^\infty \frac{1}{2} mv_x^2 n_{v_x} v_x dv_x + \left(\frac{1}{\gamma - 1} - \frac{1}{2}\right) kT. \tag{5}$$

Where $\left(\frac{1}{\gamma-1} - \frac{1}{2}\right) kT$ is the contribution of the gas molecule velocity perpendicular to the x direction to the average kinetic energy. By substituting the equation 3 and 4 into the equation 5, we have:

$$\begin{aligned} E_k &= \frac{1}{n\sqrt{\frac{kT}{2\pi m}}} \int_0^\infty \frac{1}{2} mn\sqrt{\frac{m}{2\pi kT}} v_x^3 \exp\left(-\frac{mv_x^2}{2kT}\right) dv_x + \left(\frac{1}{\gamma - 1} - \frac{1}{2}\right) kT \\ &= \frac{1}{\sqrt{\frac{kT}{2\pi m}}} \frac{1}{2} m\sqrt{\frac{m}{2\pi kT}} \frac{1}{2\left(\frac{m}{2kT}\right)^2} + \left(\frac{1}{\gamma - 1} - \frac{1}{2}\right) kT \\ &= KT + \left(\frac{1}{\gamma - 1} - \frac{1}{2}\right) kT. \end{aligned} \tag{6}$$

5.3 Space Station Recoil Analysis

5.3.1 Initial Conditions and Assumptions

Now, let's consider the space station's recoil in our energy and momentum equations. Since the small hole is perpendicular to the x-axis, considering symmetry, after the gas is released, the container moves with a velocity in the negative x-direction. Let N be the number of gas molecules in the space station at time t, and u be the magnitude of the station's overall velocity. At time $t + dt$, the number of gas molecules in the station is $N + dN$, and the magnitude of the station's overall velocity is $u + du$.

5.3.2 Momentum Conservation

The conservation of momentum for the entire system during this process is expressed as follows [3] (M represents the mass of the space station, $M \gg Nm$):

$$(M + Nm)u = [(M + Nm + mdN)(u + du)] - mdN(u - \bar{v}_x). \tag{7}$$

The average velocity of the gas molecules escaping in the x-direction over the time interval dt is denoted as v_x. The conservation of momentum for the entire system during this process, after simplification and neglecting higher-order small quantities, is given by:

$$(M + Nm)du = -m\overline{v_x}dN. \tag{8}$$

5.3.3 Energy Conservation

The conservation of energy for the entire system during this process is expressed as follows:

$$\begin{aligned}\frac{1}{2}(M + Nm)u^2 + \frac{1}{\gamma - 1}NkT =& \frac{1}{2}(M + Nm + dNm)(u + du)^2 \\ &+ \frac{1}{\gamma - 1}(N + dN)k(T + dT) \\ &- \frac{1}{2}dNm\overline{(u - v_x)^2} - \left(\frac{1}{\gamma - 1} - \frac{1}{2}\right)dNkT.\end{aligned} \tag{9}$$

Simplifying and neglecting higher-order small quantities, we get:

$$(M + Nm)udu + \frac{1}{\gamma - 1}NkdT + \frac{1}{2}kTdN + mu\overline{v_x}dN - \frac{1}{2}m\overline{v_x^2}dN = 0. \tag{10}$$

5.3.4 System Evolution Equations

By substituting the equation 8 into the equation 10, we have:

$$\left(\frac{1}{2}m\overline{v_x^2} - \frac{1}{2}kT\right) dN = \frac{1}{\gamma - 1} NkdT. \tag{11}$$

According to 5.2, we know:

$$\frac{1}{2}m\overline{v_x^2} = kT. \tag{12}$$

Thus, we have:

$$\frac{1}{2}kT\ dN = \frac{1}{\gamma - 1} NkdT. \tag{13}$$

5.3.5 Final Solutions

The integral of both sides of equation 13 yields:

$$T = CN^{\frac{\gamma-1}{2}}, \tag{14}$$

$$C = T_0/N_0^{\frac{\gamma-1}{2}}. \tag{15}$$

From equation 4, the number of gas molecules leaked in time dt is:

$$dN = -\frac{N}{V}A\sqrt{\frac{kT}{2\pi m}}dt = -\frac{N^{\frac{\gamma+3}{4}}}{V}A\sqrt{\frac{kC}{2\pi m}}dt. \tag{16}$$

The integral of both sides gives our final solutions:

$$N = N_0\left(1 + \frac{A(\gamma-1)}{4V}\sqrt{\frac{kT_0}{2\pi m}}t\right)^{-\frac{4}{(\gamma-1)}}. \tag{17}$$

Similarly, we get:

$$T = CN^{\frac{\gamma-1}{2}} = T_0\left(1 + \frac{A(\gamma-1)}{4V}\sqrt{\frac{kT_0}{2\pi m}}t\right)^{-2}, \tag{18}$$

$$p = p_0\left(1 + \frac{A(\gamma-1)}{4V}\sqrt{\frac{kT_0}{2\pi m}}t\right)^{-\frac{2(\gamma+1)}{(\gamma-1)}}. \tag{19}$$

5.4 Dismiss the Space Station's Recoil

Actually, if we don't consider the recoil motion of the space station, the calculating result will be the same as 5.3.4.

We have the energy formula below:

$$\frac{1}{\gamma-1}(N+dN)k(T+dT)-\frac{1}{\gamma-1}NkT=dN\left(\left(\frac{1}{\gamma-1}-\frac{1}{2}\right)kT+kT\right). \tag{20}$$

After simplifying it, we will find that it's the same as equation 13.

5.5 Calculating Result

When we try to solve the equation 19,

$$p_f=p_0\left(1+\frac{A(\gamma-1)}{4V}\sqrt{\frac{kT_0}{2\pi m}}t\right)^{-\frac{2(\gamma+1)}{(\gamma-1)}} \tag{21}$$

We get:

$$t=\frac{4V}{A(\gamma-1)}\sqrt{\frac{2\pi m}{kT_0}}\cdot[(\frac{p_f}{p_0})^{-\frac{\gamma-1}{2(\gamma+1)}}-1] \tag{22}$$

After plugging in all the parameters, we can calculate the value of t to be:

$$t\approx 72958s\approx 7.3\times 10^4 s. \tag{23}$$

For now, we can visualize our results, according to equation 19 and 22:

Diameter (cm)	**Time (s)**	**Time (h)**
0.1	7.296×10^6	2.027×10^3
0.5	2.918×10^5	8.106×10^2
1.0	7.296×10^4	2.027×10^1
1.5	3.243×10^4	9.007×10^0
2.0	1.824×10^4	5.066×10^0
5.0	2.918×10^3	8.106×10^{-1}

Table 2: Time required for pressure reduction to $0.3p_0$ at different apertures in model 1

Based on equation 22, we get Table 2, which shows the time required for pressure reduction to $0.3p_0$ at different aperture sizes, noting that the duration time varies in a long range.

If we fix the other parameters and only change the aperture size, according to equation 19, we can draw Figure 4.

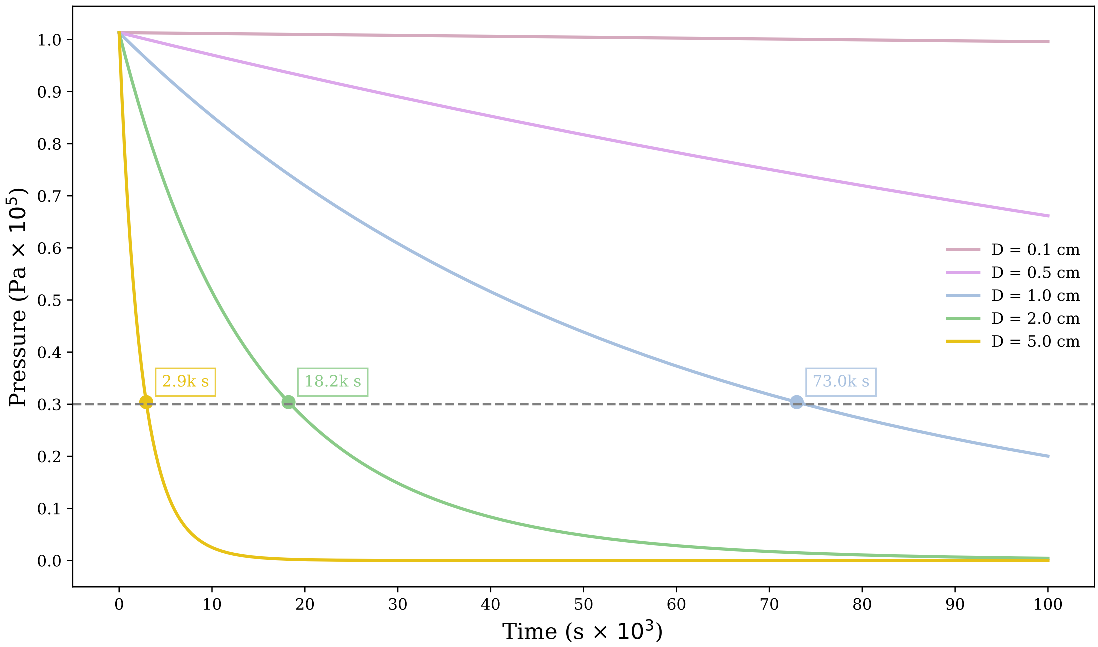

Figure 4: Pressure over Time for Different Hole Diameters using Model 1

From Figure 4, we can observe that the outflow rate decreases significantly with the increase of orifice diameter. At $D = 5.0cm$, the outflow is relatively rapid, while at $D = 0.1cm$, the outflow is extremely slow.

That's all we can do to deal with this aerodynamic model. However, it will be proved inaccurate later.

6 Model 2: Continuous Medium Model

6.1 Two Main Assumptions

6.1.1 Knudsen Number

First, we calculate the Knudsen number to determine the flow regime. The Knudsen number is defined as:

$$Kn = \frac{\lambda}{L} \tag{24}$$

where λ is the mean free path and L is the characteristic length.

The mean free path is calculated using:

$$\lambda = \frac{k_B T}{\sqrt{2}\pi d_0^2 p} \tag{25}$$

Input parameters:

- Boltzmann constant: $k_B = 1.38 \times 10^{-23}$ J/K
- Temperature: $T = 293.15$ K (20°C)
- Effective diameter of air molecules: $d_0 = 3.7 \times 10^{-10}$ m
- Pressure: $p = 101300$ Pa (1 atm)

Calculating the mean free path:

$$\lambda = \frac{(1.38 \times 10^{-23})(293.15)}{\sqrt{2}\pi(3.7 \times 10^{-10})^2(101300)} \approx 6.63 \times 10^{-8} \text{ m} \tag{26}$$

Taking the hole diameter as characteristic length: $L = 0.01$ m

Therefore, the Knudsen number is:

$$Kn = \frac{6.63 \times 10^{-8}}{0.01} \approx 6.63 \times 10^{-6} \tag{27}$$

Since $Kn < 0.001$, this falls within the continuum flow regime, allowing us to apply continuum fluid mechanics methods.

6.1.2 Critical Flow

At the orifice, we need to determine if the flow reaches sonic conditions. This requires comparing downstream pressure with critical pressure.

For an ideal gas, the critical pressure ratio is:

$$\frac{p^*}{p_0} = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{28}$$

where $\gamma = 1.4$ is the specific heat ratio for air. Calculating:

$$\frac{p^*}{p_0} = \left(\frac{2}{2.4}\right)^{3.5} \approx 0.528 \tag{29}$$

Therefore, the critical pressure is:

$$p^* = 0.528p_0 = 0.528 \times 101300 \approx 53486 \text{ Pa} \tag{30}$$

Since external environment is vacuum (pressure near 0), which is far below the critical pressure, the flow at the orifice definitely reaches sonic conditions, resulting in critical flow.

6.2 Analysis

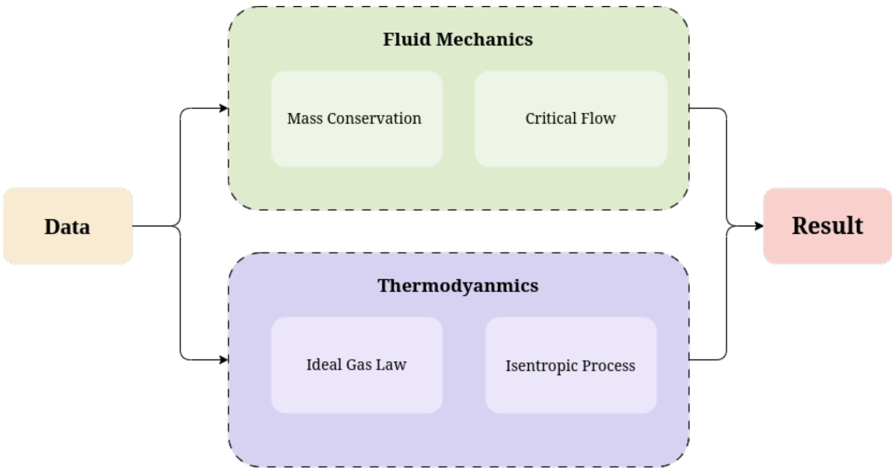

Figure 5: Model 2 overview diagram

6.2.1 Basic Governing Equations

To model the depressurization process of the space station, we start with several fundamental equations that govern compressible gas flow.

First, we consider the ideal gas law, which relates the pressure p, volume V, mass m, and temperature T:

$$pV = mRT \tag{31}$$

where R is the specific gas constant for air.

For an isentropic process, we have the following important relationships between pressure, density, and temperature:

$$\frac{p_2}{p_1} = \left(\frac{\rho_2}{\rho_1}\right)^{\gamma} = \left(\frac{T_2}{T_1}\right)^{\frac{\gamma}{\gamma-1}} \tag{32}$$

where γ is the specific heat ratio (approximately 1.4 for air).

6.2.2 Mass Conservation and Flow Rate Analysis

The mass conservation principle for our system states that the rate of mass change inside the space station equals the mass flow rate through the hole:

$$\frac{dm}{dt} = -\dot{m} \tag{33}$$

where the negative sign indicates mass leaving the system.

For isentropic flow, we can express the temperature-pressure relationship as:

$$\frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)^{\frac{\gamma-1}{\gamma}} \tag{34}$$

6.2.3 Critical Flow Conditions

At the discharge hole, the flow becomes critical (choked) when the local Mach number reaches unity. The critical pressure ratio for this condition is:

$$\left(\frac{p_e}{p}\right)_{critical} = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \approx 0.528 \tag{35}$$

Since we are venting to vacuum ($p_e = 0$), the flow will always be critical (choked). Under these conditions, the mass flow rate is given by [4]:

$$\dot{m} = C_d A p \sqrt{\frac{\gamma}{RT}} \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \tag{36}$$

where C_d is the discharge coefficient [5] and A is the hole area.

6.2.4 Temperature-Pressure Evolution

During the isentropic process, the relationship between temperature and pressure can be expressed as:

$$T = T_0 \left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{\gamma}} \tag{37}$$

where T_0 and p_0 are the initial temperature and pressure, respectively.

6.2.5 Derivation of the Time Evolution Equation

Combining the mass conservation equation with the ideal gas law yields:

$$\begin{aligned} m &= \frac{pV}{RT}, \\ \frac{d}{dt}\left(\frac{pV}{RT}\right) &= -\dot{m}. \end{aligned} \tag{38}$$

Substituting equations 36 and 37, we obtain:

$$\frac{d}{dt}\left(\frac{pV}{RT_0\left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{\gamma}}}\right) = -C_d A p \sqrt{\frac{\gamma}{RT_0\left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{\gamma}}}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}}. \tag{39}$$

6.2.6 Solution Process

After expanding and simplifying the left-hand side of equation 39, we get:

$$\frac{V}{RT_0} \cdot \frac{d}{dt}\left(p^{\frac{\gamma+1}{\gamma}} p_0^{-\frac{\gamma-1}{\gamma}}\right) = -C_d A p \sqrt{\frac{\gamma}{RT_0}} \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} p^{-\frac{\gamma-1}{2\gamma}} p_0^{\frac{\gamma-1}{2\gamma}}. \tag{40}$$

Further mathematical manipulation leads to:

$$\frac{V}{RT_0} \cdot \frac{\gamma+1}{\gamma} p^{\frac{1}{\gamma}} p_0^{-\frac{\gamma-1}{\gamma}} \frac{dp}{dt} = -C_d A p \sqrt{\frac{\gamma}{RT_0}} \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} p^{-\frac{\gamma-1}{2\gamma}} p_0^{\frac{\gamma-1}{2\gamma}}. \tag{41}$$

6.2.7 Final Integration and Result

To solve for the depressurization time, we separate variables and integrate:

$$\int_{p_0}^{p} \frac{\left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{2\gamma}}}{p} dp = -\frac{C_d A}{V}\sqrt{\gamma R T_0}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \int_0^t dt. \tag{42}$$

Through substitution $u = \left(\frac{p}{p_0}\right)^{\frac{\gamma+1}{2\gamma}}$, we obtain:

$$t = \frac{2V}{C_d A}\sqrt{\frac{1}{\gamma R T_0}}\left[1 - \left(\frac{p_f}{p_0}\right)^{\frac{\gamma+1}{2\gamma}}\right]. \tag{43}$$

6.2.8 Physical Interpretation and Implications

The final equation 43 reveals several important characteristics of the depressurization process:

- The evacuation time is directly proportional to the vessel volume (V)
- The time is inversely proportional to the hole area (A)
- The temperature dependence enters through the square root term
- The pressure ratio term shows how the depressurization rate changes with pressure

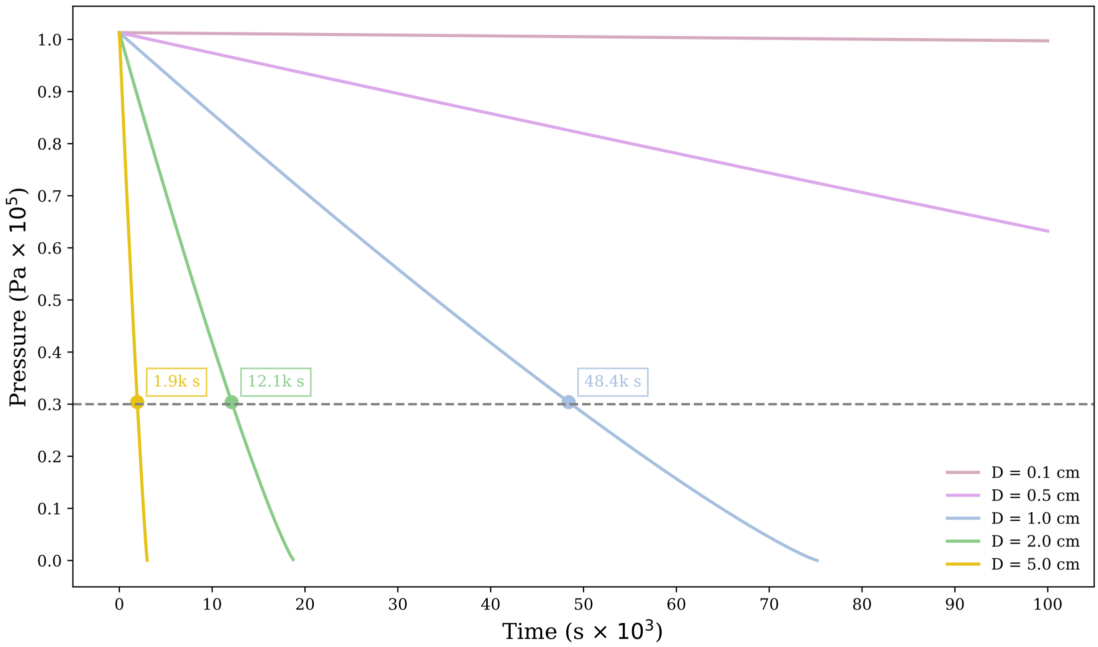

Figure 6: Pressure over Time for Different Hole Diameters using Model 2

From Figure 6, we can observe that the outflow rate decreases significantly with thein-crease of orifice diameter. This characteristic is the same as model 1. In addition, due to this unique continuous medium model, the rate of pressure decrease did not change significantly. In other words, the air pressure is approximately linearly decreasing.

Diameter (cm)	**Time (s)**	**Time (h)**
0.1	4.840×10^6	1.344×10^3
0.5	1.936×10^5	5.378×10^1
1.0	4.840×10^4	1.344×10^1
1.5	2.151×10^4	5.976×10^0
2.0	1.210×10^4	3.361×10^0
5.0	1.936×10^3	5.378×10^{-1}

Table 3: Time required for pressure reduction to $0.3p_0$ at different apertures in model 2

From Table 3, we can also notice that the discharge time is inversely proportional to the hole area and the time for discharge is inversely proportional to the square of the hole diameter. We can also notice that the discharge time decreases significantly, while the hole diameter increases.

7 Results Discussion

7.1 Discussion on the Relation Between Time and Diameters

Based on the expressions derived from the two models, since the area A of the small hole is in the denominator, the time t for gas evacuation is inversely proportional to the square of the diameter D of the hole. We can further illustrate it by plotting data points in Figure 7.

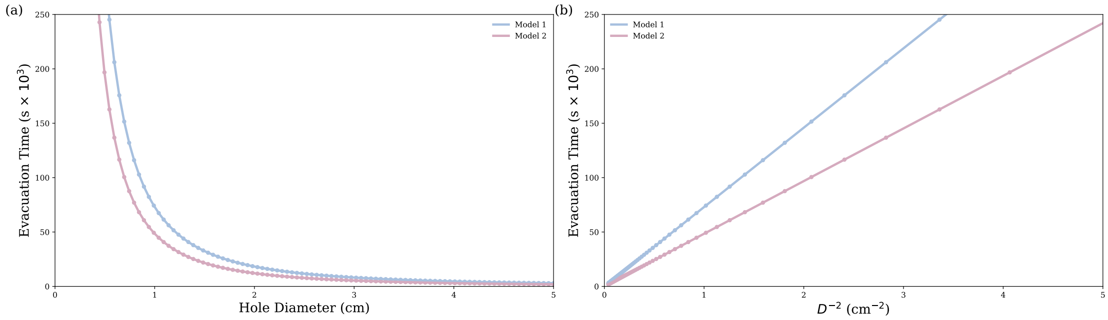

Figure 7: Pressure over Diameters for Different Models

7.2 Discussion on Two Models

7.2.1 Comparison on Figures

Here are two figures 8 and 9 showing the difference in discharge time and pressure decrease tendency between the two models.

For Figure 8, in order to clearly show the difference of the result under multiple scales, we remove the column representing the situation when the hole diameter is 0.1cm.

As Figure 8 shows, the time calculated by model 2 is much shorter than model 1's result for all the diameter conditions included in Figure 8. We can also observe that we have

$$\frac{p_{model2}}{p_{model1}} \approx \frac{2}{3} \tag{44}$$

in all 4 conditions, which is quite an interesting finding.

And as Figure 9 shows, the rate of pressure drop calculated from model 1 is always less than the the one in model 2. The drop rate calculated from model 1 decreases as the time move on and the one from model 2 seems not to decrease significantly with time.

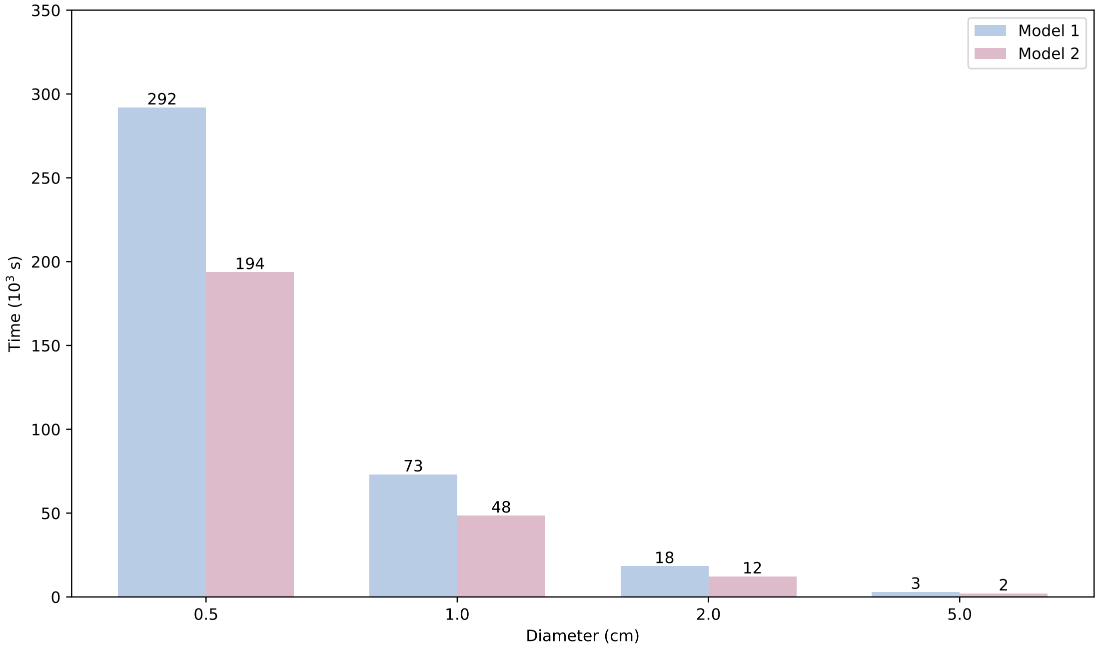

Figure 8: Comparison of leaking duration between model 1 and model 2

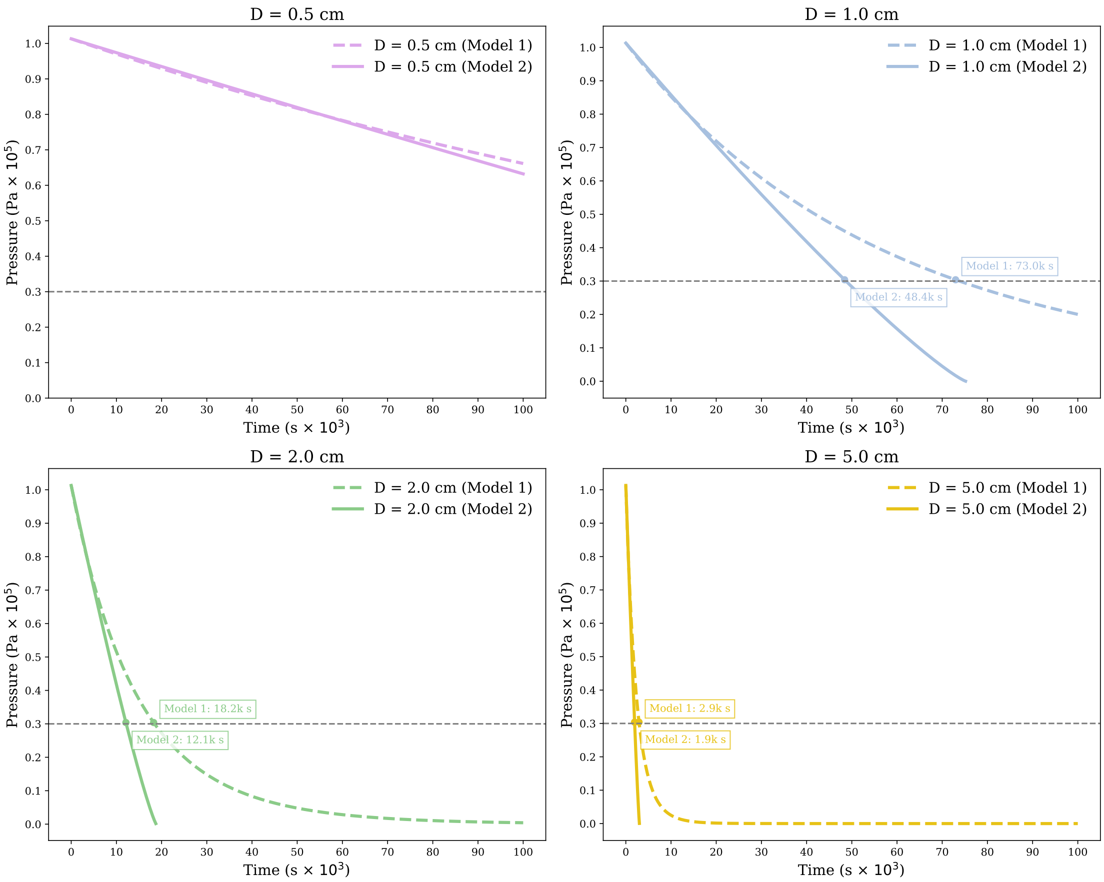

Figure 9: Comparison of pressure decrease between model 1 and model 2

7.2.2 Analysis

Through our analysis, the reason for such a result is mainly due to the fact that gas is in the actual continuous flow case, which is concluded by the analysis of the Knudsen Number. The rapid leakage of the gas, which reaches the speed of sound, allows for enhanced collisions between molecules, which leads to a more rapid escape from the space station module. Also in common with both models, the gas leaks in inverse square time, which is dramatically accelerated as the aperture size increases.

8 Model Evaluation

8.1 Strengths

1. We developed two complementary models based on different theoretical approaches (molecular dynamics and continuum mechanics) to analyze the space station depressurization process, and performed detailed calculations to determine evacuation times for various hole sizes.

2. We established a comprehensive theoretical foundation by analyzing the Knudsen number and critical flow conditions, which helped validate our modeling approaches and determine appropriate flow regimes for each model.

3. We modeled the subsequent pressure changes based on the initial conditions and flow characteristics, making reasonable simplifications while maintaining physical accuracy in both molecular and continuum approaches.

8.2 Weaknesses

1. We used approximate treatments in several areas:

    - The discharge coefficient was assumed constant in the continuum model
    - The molecular dynamics model assumed idealized molecular collisions

2. We did not establish a complete thermal model incorporating:

    - Heat transfer between the gas and station walls
    - Temperature gradients near the orifice
    - Non-equilibrium thermal processes

3. While our models provided good theoretical results, there are practical limitations:

    - Limited experimental validation possibilities

9 Conclusion

All in all, through differential equations, we can calculate that the time taken for the gas to evacuate from the chamber to 0.3 atm is **about 48000 s** for a hole diameter of 1 cm. In further calculations, we find that the time taken for the gas evacuation further decreases as the hole diameter of the breach increases. At the same time, according to the calculation of the Knudsen number of air, we found that the aerodynamics model taken for our model 1 is not very accurate. Therefore, our discussion of the Knudsen number led us to conclude that the scenario is more suitable for analyzing the gas as a **continuous flow**. Therefore, we developed a second modeling approach that uses a **continuous medium model** of air to calculate the time evolution equation, which leads to a more accurate and reasonable time for the gas to leak to the required gas pressure.

After the above series of analyses and model comparisons, we can conclude the following:

- When the **Knudsen number** is **large**, the gas should be analyzed and discussed as a **particle flow**.
- When the **Knudsen number** is **small**, the gas should be analyzed as a **continuous flow**.
- The gas evacuation time **decreases quadratically** with increasing aperture size.
- From a more rigorous point of view, the gas in space **should be considered as a continuous flow** to obtain a more accurate time evolution equation.
- The time it takes for the gas to leak to 0.3 atm with an aperture of cm is **about 48000s**.

References

[1] A. V. Semikolenov, "Molecular flow of an ideal gas through a small hole channel into a vacuum," in AIP Conf. Proc., vol. 2383, no. 1, p. 020014, 25-Apr-2022. https://doi.org/10.1063/5.0074748

[2] M. Khorrami, Effusion of Gas Through a Small Hole, *Int J Thermophys* **42**, 14 (2021). https://doi.org/10.1007/s10765-020-02766-w

[3] Chinese Physics Society. (2018). Problem 5 of the 35th Chinese Physics Olympiad Final. Retrieved from https://cpho.pku.edu.cn/info/1053/1134.htm

[4] Pritchard, P. J., Leylegian, J. C. (2011). Fox and McDonald's Introduction to Fluid Mechanics (8th ed., p. 706). John Wiley Sons, Inc.

[5] BITSDE. (n.d.). §1-5 Orifice and Gap Flow. Retrieved from http://course.bitsde.com/web/209/CH02/23.HTM

A Appendix

A.1 Figure4.py

This code implements a molecular kinetic theory-based model for space station depressurization. It calculates the pressure decay using Maxwell velocity distribution and molecular dynamics principles. The model considers molecular flow through the hole and uses the Boltzmann constant and molecular mass in its calculations. The time required to reach 0.3 atmospheres is calculated by integrating the molecular flux equation.

Key features:

- Based on molecular kinetic theory
- Uses Boltzmann constant and molecular mass
- Assumes Maxwell velocity distribution
- Calculates pressure decay over time

```
import numpy as np
import matplotlib.pyplot as plt

# Define constants
P0 = 101325  # Initial pressure, in Pascal
V = np.pi * (4 / 2)**2 * 50  # Cylinder volume, in cubic meters
T0 = 293.15  # Temperature, in Kelvin
k = 1.380649e-23  # Boltzmann constant, in J/K
m = 4.81e-26  # Air molecule mass, in kg
t_max = 100000  # Maximum time, in seconds

# Define color scheme
COLORS = {
    'blue': '#A7C0DF',
    'pink': '#D5AABE',
    'green': '#8ACB88',
    'purple': '#DCA7EB',
    'yellow': '#E7C217'
}

def calculate_time(d):
    """Calculate time needed to reach target pressure"""
    A = np.pi * (d/2)**2
    t = -10 * V / (A * np.sqrt(k * T0 / (2 * np.pi * m))) * (1 - (0.3)**(-1/12))
    return t

def calculate_pressure(t, d):
```

```
    """Calculate pressure at given time point"""
    A = np.pi * (d/2)**2
    return P0 * (1 + (A / (10 * V)) * np.sqrt(k * T0 / (2 * np.pi * m)) * t)**(-12)

# Set different hole diameters
diameters = [0.1, 0.5, 1.0, 2.0, 5.0]  # cm
colors = [COLORS['pink'], COLORS['purple'], COLORS['blue'], COLORS['green'], COLORS['yellow']]

# Set global font to serif
plt.rcParams['font.family'] = 'serif'

# Create figure
plt.figure(figsize=(10, 6))

# Plot pressure evolution curves and intersection points
for d, color in zip(diameters, colors):
    d_m = d/100  # Convert to meters
    t_points = np.linspace(0, t_max, 1000)
    p_points = calculate_pressure(t_points, d_m)
    plt.plot(t_points/1e3, p_points/1e5, color=color,
             label=f'D = {d:.1f} cm', linewidth=2)

    # Calculate intersection point
    t_intersect = calculate_time(d_m)

    # Only plot intersection if it occurs within t_max
    if t_intersect <= t_max:
        p_intersect = calculate_pressure(t_intersect, d_m)

        # Plot intersection point
        plt.plot(t_intersect/1e3, p_intersect/1e5, 'o', color=color, markersize=8)

        # Add annotation
        plt.annotate(f'{t_intersect/1000:.1f}k s',
                    xy=(t_intersect/1e3, p_intersect/1e5),
                    xytext=(10, 10), textcoords='offset points',
                    color=color,
                    bbox=dict(facecolor='white', edgecolor=color, alpha=0.8))

# Add 0.3P  reference line
plt.axhline(y=0.3, color='gray', linestyle='--', label='0.3P ')

# Set figure properties
plt.xlabel('Time (s × $10^3$)', fontsize=14)
plt.ylabel('Pressure (Pa × $10^5$)', fontsize=14)

# Set ticks
plt.xticks(np.arange(0, t_max/1000+1, 10),
          ['%d' % (x) for x in np.arange(0, t_max/1000+1, 10)])
```

```python
plt.yticks(np.arange(0, 1.1, 0.1),
          ['%.1f' % (x) for x in np.arange(0, 1.1, 0.1)])

# Add legend
plt.grid(False)
plt.legend(frameon=False)

# Save and display figure
plt.tight_layout()
plt.savefig('model1_1.pdf', format='pdf', bbox_inches='tight')
plt.show()

```

A.2 Figure6.py

This code implements a continuum flow model based on compressible fluid dynamics. It uses the discharge coefficient and critical flow conditions to calculate the depressurization process. The model applies isentropic flow equations and considers the specific heat ratio of the gas. The time to reach 0.3 atmospheres is calculated using the derived analytical solution.

Key features:

- Based on compressible fluid dynamics
- Uses discharge coefficient
- Considers specific heat ratio
- Applies isentropic flow equations

```python
import numpy as np
import matplotlib.pyplot as plt

# Define constants
gamma = 1.4  # specific heat ratio
R = 287      # gas constant (J · kg ¹ · K ¹)
T0 = 293.15  # initial temperature (K)
P0 = 101300  # initial pressure (Pa)
Pf = 0.3 * P0  # final pressure (Pa)
Cd = 0.62    # discharge coefficient
D = 4        # station diameter (m)
L = 50       # station length (m)
V = np.pi * (D/2)**2 * L  # station volume (m³)
t_max = 100000  # maximum time, in seconds

```

```python
# Define color scheme
COLORS = {
    'blue': '#A7C0DF',
    'pink': '#D5AABE',
    'green': '#8ACB88',
    'purple': '#DCA7EB',
    'yellow': '#E7C217'
}

def calculate_time(d):
   """Calculate time needed to reach target pressure"""
   A = np.pi * (d/2)**2
   t = (2*V)/(Cd*A) * np.sqrt(1/(gamma*R*T0)) * (1 - (Pf/P0)**((gamma+1)/(2*gamma)))
   return t

def calculate_pressure(t, d):
   """Calculate pressure at given time point"""
   A = np.pi * (d/2)**2
   pressure_ratio = (1 - (Cd*A*np.sqrt(gamma*R*T0)/(2*V))*t)**(2*gamma/(gamma+1))
   return P0 * pressure_ratio

# Set different hole diameters (cm)
diameters = [0.1, 0.5, 1.0, 2.0, 5.0]  # cm
colors = [COLORS['pink'], COLORS['purple'], COLORS['blue'], COLORS['green'], COLORS['yellow']]

# Set global font to serif
plt.rcParams['font.family'] = 'serif'

# Create figure
plt.figure(figsize=(10, 6))

# Plot pressure evolution curves for each diameter
for d, color in zip(diameters, colors):
   t_points = np.linspace(0, t_max, 1000)
   p_points = [calculate_pressure(t, d/100) for t in t_points]
   plt.plot(t_points/1e3, np.array(p_points)/1e5, color=color,
            label=f'D = {d:.1f} cm', linewidth=2)

   # Calculate intersection point
   t_intersect = calculate_time(d/100)

   # Only plot intersection if it occurs within t_max
   if t_intersect <= t_max:
       p_intersect = calculate_pressure(t_intersect, d/100)

       # Plot intersection point
       plt.plot(t_intersect/1e3, p_intersect/1e5, 'o', color=color, markersize=8)

       # Add annotation
```

```python
    plt.annotate(f'{t_intersect/1000:.1f}k s',
                 xy=(t_intersect/1e3, p_intersect/1e5),
                 xytext=(10, 10), textcoords='offset points',
                 color=color,
                 bbox=dict(facecolor='white', edgecolor=color, alpha=0.8))

# Add gray dashed line for final pressure
plt.axhline(y=0.3, color='gray', linestyle='--')

# Set figure properties
plt.xlabel('Time (s × $10^3$)', fontsize=14)
plt.ylabel('Pressure (Pa × $10^5$)', fontsize=14)

# Set ticks
plt.xticks(np.arange(0, t_max/1000+1, 10), ['%d' % (x) for x in np.arange(0, t_max/1000+1,
    10)])
plt.yticks(np.arange(0, 1.1, 0.1), ['%.1f' % (x) for x in np.arange(0, 1.1, 0.1)])

# Add legend and disable grid
plt.grid(False)
plt.legend(frameon=False)

# Adjust layout and save
plt.tight_layout()
plt.savefig('model2_1.pdf', format='pdf', bbox_inches='tight')
plt.show()
```

A.3 Figure7.py

```python
import numpy as np
import matplotlib.pyplot as plt

# Constants
gamma = 1.4
R = 287
T0 = 293.15
P0 = 101300
Pf = 0.3 * P0
Cd = 0.62
D = 4
L = 50
V = np.pi * (D/2)**2 * L
k = 1.380649e-23
m = 4.81e-26

# Color scheme
COLORS = {
    'blue': '#A7C0DF',
```

```python
    'pink': '#D5AABE',
    'green': '#B7F5DE',
    'purple': '#DCA7EB',
    'yellow': '#EAE936'
}

def calculate_time_2(d):
    """Model 2"""
    A = np.pi * (d/2)**2
    t = (2*V)/(Cd*A) * np.sqrt(1/(gamma*R*T0)) * (1 - (Pf/P0)**((gamma+1)/(2*gamma)))
    return t

def calculate_time_3(d):
    """Model 1"""
    A = np.pi * (d/2)**2
    t = (4*V)/(A*(gamma-1)) * np.sqrt(2*np.pi*m/(k*T0)) * \
        ((Pf/P0)**(-((gamma-1)/(2*(gamma+1)))) - 1)
    return t

# Create diameter points
diameters = np.linspace(0.1, 5, 100) / 100  # 0.1-5 cm, converted to meters

# Calculate times and inverse squared diameters
times_2 = np.array([calculate_time_2(d) for d in diameters])
times_3 = np.array([calculate_time_3(d) for d in diameters])
inverse_d_squared = 1 / (diameters**2)  # Calculate d^(-2)

# Set font to serif
plt.rcParams['font.family'] = 'serif'

# Create figure
plt.figure(figsize=(10, 6))

# Plot Model 1
plt.plot(inverse_d_squared/10000, times_3/1000, color=COLORS['blue'], linewidth=2,
         label='Model 1')
plt.scatter(inverse_d_squared/10000, times_3/1000, color=COLORS['blue'], s=20)
# Plot Model 2
plt.plot(inverse_d_squared/10000, times_2/1000, color=COLORS['pink'], linewidth=2,
         label='Model 2')
plt.scatter(inverse_d_squared/10000, times_2/1000, color=COLORS['pink'], s=20)


# Set labels
plt.xlabel('$D^{-2}$ (cm$^{-2}$)', fontsize=14)
plt.ylabel('Evacuation Time (s × $10^3$)', fontsize=14)
plt.legend(frameon=False)

# Set axis limits
plt.xlim(0, 5)  # Adjust based on your needs
```

```
plt.ylim(0, 250)

# Customize ticks
plt.grid(False)

# Adjust layout and save
plt.tight_layout()
plt.savefig('time_vs_inverse_diameter_squared.pdf', format='pdf', bbox_inches='tight')
plt.show()

# Print comparison for selected inverse squared diameters
print("\nEvacuation time comparison:")
print("-" * 70)
print("D^(-2) (cm^-2)  Model 2 (s)   Model 1 (s)   Difference (%)")
print("-" * 70)
for d in [0.5, 1.0, 2.0, 5.0]:
    d_inv_sq = 1/(d**2)
    t2 = calculate_time_2(d/100)
    t3 = calculate_time_3(d/100)
    diff_percent = (t3 - t2) / t2 * 100
    print(f"{d_inv_sq:8.1f}      {t2:8.0f}      {t3:8.0f}      {diff_percent:8.1f}")
print("-" * 70)
```

A.4 Figure8.py

```
    import matplotlib.pyplot as plt
import numpy as np

diameters = [0.5, 1.0, 2.0, 5.0]

time_seconds_model1 = [291830, 72958, 18239, 2918]

time_seconds_model2 = [193607, 48402, 12100, 1936]

COLORS = {
    'blue': '#A7C0DF',
    'pink': '#D5AABE',
    'green': '#8ACB88',
    'purple': '#DCA7EB',
    'yellow': '#E7C217'
}

bar_width = 0.35
opacity = 0.8

index = np.arange(len(diameters))

plt.figure(figsize=(10, 6))
```

```

time_kseconds_model1 = np.array(time_seconds_model1) / 1e3
time_kseconds_model2 = np.array(time_seconds_model2) / 1e3

bars1 = plt.bar(index, time_kseconds_model1, bar_width, alpha=opacity, color=COLORS['blue'],
      label='Model 1')
bars2 = plt.bar(index + bar_width, time_kseconds_model2, bar_width, alpha=opacity, color=
     COLORS['pink'], label='Model 2')

plt.yscale('linear')
plt.ylim(0, 350)
plt.yticks(np.arange(0, 351, 50), [f'{int(t)}' for t in np.arange(0, 351, 50)])

plt.xlabel('Diameter (cm)')
plt.ylabel('Time ($10^3$ s)')
plt.xticks(index + bar_width / 2, diameters)
plt.legend()

def add_labels(bars):
    for bar in bars:
        yval = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2, yval, f'{yval:.0f}', va='bottom', ha='
     center')

add_labels(bars1)
add_labels(bars2)

plt.tight_layout()
plt.savefig('model3_1.pdf', format='pdf', bbox_inches='tight')
plt.show()
```

A.5 Figure9.py

```
import numpy as np
import matplotlib.pyplot as plt

gamma = 1.4
R = 287
T0 = 293.15
P0 = 101300
Pf = 0.3 * P0
Cd = 0.62
D = 4
L = 50
V = np.pi * (D/2) ** 2 * L
t_max = 100000

COLORS = {
```

```
    'blue': '#A7C0DF',
    'pink': '#D5AABE',
    'green': '#8ACB88',
    'purple': '#DCA7EB',
    'yellow': '#E7C217'
}

def calculate_time(d):
    A = np.pi * (d/2) ** 2
    t = (2*V)/(Cd*A) * np.sqrt(1/(gamma*R*T0)) * (1 - (Pf/P0)**((gamma + 1)/(2*gamma)))
    return t

def calculate_pressure(t, d):
    A = np.pi * (d/2) ** 2
    pressure_ratio = (1 - (Cd*A*np.sqrt(gamma*R*T0)/(2*V))*t)**(2*gamma/(gamma+1))
    return P0 * pressure_ratio

def calculate_time_2(d):
    A = np.pi * (d/2)**2
    t = -10 * V / (A * np.sqrt(k * T0 / (2 * np.pi * m))) * (1 - (0.3)**(-1/12))
    return t

def calculate_pressure_2(t, d):
    A = np.pi * (d/2)**2
    return P0 * (1 + (A / (10 * V)) * np.sqrt(k * T0 / (2 * np.pi * m)) * t)**(-12)

k = 1.380649e-23
m = 4.81e-26

diameters = [0.1, 0.5, 1.0, 2.0, 5.0]  # cm
colors = [COLORS['pink'], COLORS['purple'], COLORS['blue'], COLORS['green'], COLORS['yellow']]

plt.rcParams['font.family'] = 'serif'

plt.figure(figsize=(10, 6))

for d, color in zip(diameters, colors):
    d_m = d/100
    t_points = np.linspace(0, t_max, 1000)
    p_points = calculate_pressure_2(t_points, d_m)
    plt.plot(t_points/1e3, p_points/1e5, '--', color=color,
             label=f'D = {d:.1f} cm (Model 1)', linewidth=2)

for d, color in zip(diameters, colors):
    d_m = d/100
    t_points = np.linspace(0, t_max, 1000)
    p_points = calculate_pressure(t_points, d_m)
    plt.plot(t_points/1e3, np.array(p_points)/1e5, color=color,
             label=f'D = {d:.1f} cm (Model 2)', linewidth=2)
```

```python


plt.axhline(y=0.3, color='gray', linestyle='--')

plt.xlabel('Time (s × $10^3$)', fontsize=14)
plt.ylabel('Pressure (Pa × $10^5$)', fontsize=14)

plt.xticks(np.arange(0, t_max/1000+1, 10),
          ['%d' % (x) for x in np.arange(0, t_max/1000+1, 10)])
plt.yticks(np.arange(0, 1.1, 0.1),
          ['%.1f' % (x) for x in np.arange(0, 1.1, 0.1)])

plt.grid(False)
plt.legend(loc='center right', bbox_to_anchor=(1, 0.47), frameon=False)

plt.tight_layout()
plt.savefig('combined_model.pdf', format='pdf', bbox_inches='tight')
plt.show()
```