Problem Chosen	Date	Team
A	**November 4 2023**	**464**

Maximum space diving altitude based on specific velocity altitude and aerodynamic heat

Abstract

To investigate the maximal feasible altitude for space skydiving while ensuring the safe return of the skydiver to the ground, this paper developed two models: a skydiving model and a parachute model, to analyze the dynamic behavior and force conditions of the skydiver throughout the descent.

In sky diving model, we used isothermal atmosphere to calculate the motion of the skydiver. In this process, the study innovatively introduces the Specific Velocity Altitude (SVA) as one of the evaluative metrics. In order to maximize the SVA, the maximum sky diving altitude is **19.46 km**. Jumping from this altitude ensures absolute safety for the skydiver, as it eliminates the risk of encountering severe instabilities cased by supersonic flight.

To challenge the limits of human free-fall, the skydiver must fly at supersonic speeds. But aerodynamic heat at supersonic speeds would be a major danger. To avoid being burned, the maximum sky diving altitude is **60.675 km**.

In parachute opening and landing model, we found that the velocity change at the end of the free fall is independent of the take-off height. To landing safely, the parachute should be opened at the altitude between **3.64km** and **19.93m**.

keyword: Space Diving Runge-Kutta Supersonic Flight Aerodynamic Heat Specific Velocity Altitude

Content

1. Introduction

The realm of space exploration has continuously captivated the human imagination, leading to a myriad of technological advancements that push the boundaries of our capabilities beyond Earth. As our understanding and fascination with space grow, so does our desire to experience it firsthand. This has led to the advent of Space Diving, an extreme sport that merges the thrill of skydiving with the awe-inspiring backdrop of Earth's upper atmosphere.

Space Diving, also known as high-altitude or stratospheric skydiving, is a daring activity where participants jump from spacecraft or high-altitude balloons at altitudes far surpassing those reached in traditional skydiving. The primary objective of this venture is to experience the maximum speed and the unparalleled sensation of freefall, testing the physical and mental limits of the diver as they plummet towards Earth under the sheer force of gravity.

The concept of Space Diving finds its roots in the pioneering work of Captain Joseph Kittinger during the 1960s. Kittinger's record-setting freefall jump from an altitude of 102,800 feet (31,333 meters) as part of the U.S[1]. Air Force's Project Excelsior was a research effort into survival techniques and equipment suitable for high-altitude bailout. His jump demonstrated the feasibility of surviving the harsh conditions of the upper atmosphere, paving the way for further exploration and adventure.

In recent years, several adventurers have completed successful Space Diving missions, capturing global attention and pushing the boundaries of human achievement. Notably, Felix Baumgartner's historic Red Bull

Stratos mission in 2012 broke numerous records, including the highest freefall, the fastest speed by a human in freefall, and the first supersonic skydive. Baumgartner achieved a maximum vertical speed of 1357.6 km/h, exceeding the speed of sound by 25% without the aid of a stabilization device, and jumped from the highest altitude ever of 38.9694 km.

Space Diving is inherently high-risk, requiring meticulous planning, extensive training, and specialized gear to protect the diver from extreme temperatures, pressures, and oxygen deprivation.

This paper aims to establish a physical model that accurately describes the entire process of a space dive, including freefall, parachute deployment, and landing. By understanding the underlying physics and forces involved, we aim to determine the highest safe altitude for space diving, ensuring the well-being and safety of the diver.

The outcomes of this research will not only contribute to the field of extreme sports but also have implications in aerospace engineering, emergency response planning, and future space tourism programs. By ensuring the safety of skydivers, we can facilitate the growth and advancement of this exciting activity while mitigating potential risks.

2. Restatement

There is a skydiver propelled vertically to a certain height, potentially beyond Earth's atmosphere, by a rocket. The skydiver then exits the rocket equipped with a spacesuit and a parachute, aiming to descend back to Earth's surface. Considering the combined mass of the skydiver, spacesuit, and parachute to be 190 kg, we should analyze the potential risks and challenges they might encounter during this descent, and give the maximum altitude from which a person could safely make it back to the surface.

3. Assumption

- Only consider the vertical motion of the skydiver. Vertical movement is directly related to the safety of athletes and should be considered first. In addition, since the air resistance can be considered collinear with the velocity, even if the skydiver has horizontal movement, it will not affect the vertical analysis.
- Assuming weather conditions are good during the skydiver's flight, the effects of lateral wind and turbulence are negligible.
- The ground environment is flat and broad, there are no unnecessary dangerous factors, as long as the landing speed is within the safe range, the skydiver can land safely.
- The skydiver is equipped with a hemispherical C-9 parachute with a diameter of 5.95 m. This is a very typical parachute and is well represented.
- When calculating aerodynamic heating, the stagnation temperature of the fluid is approximated to be the surface temperature of the spacesuit.

4. Physical Analysis of Model

4.1 The Dangers We Will Face

There are such dangers the skydiver will face:

A. Aerodynamic heat

B. Recoil force while opening the parachute

C. Landing velocity

D. Severe instabilities when experiencing the transonic or supersonic region

E. Flat Spin

F. Low Pressure

G. Extreme temperature

H. Huge air resistance

I. Unpredictable winds

J. Long Freefall Time.

When a skydiver takes off, his speed increases rapidly until it reaches a maximum and then decreases, during which he may experience a transonic or supersonic regime. In such a state, the drag coefficient of the air will change dramatically, causing serious instability [999], leading athletes to spin at high speed, and thus threatening flight safety. If an athlete takes off from a higher position, his maximum speed will rise linearly, and the threat of aerodynamic heat will no longer be negligible. Too much speed could burn the suit, and too much air resistance could tear the suit apart.

It is found in this paper that the relationship between skydiver's speed and altitude at the end of free fall has nothing to do with the skydiving altitude. The skydiver will have the same final velocity no matter what altitude he leaves the rocket. But there is a time window of the parachute opening: opening too early, the human body will not be able to withstand the huge recoil force; opening too late, the landing speed will be too large to land safely.

Due to the maturity of space suit production technology and other reasons, the latter several risk factors are basically under control, so this paper mainly considers the first four risk factors, and calculates the skydiving altitude and parachute opening height to avoid these risks.

Therefore, we divide the whole process of movement into two parts, one is the acceleration process at the high altitude of the Laplace Atmosphere, and the other is the process of opening the parachute and landing on the ground.

4.2 Skydiving Model:

4.2.1 Low Altitude Skydiving

This is not very relevant to the research in this article, but I think it is necessary to explain it in detail in order to understand the situation of skydiving. In the case of a low-altitude skydiving, the skydiving altitude is low enough that the air density can be assumed to be constant, resulting in a simple differential equation. And the typical way to deal with the low-altitude skydiver problem is to add a speed-dependent resistance to the standard freefall problem. In this case, Newton's second law gives the resultant acceleration as:

$$m\frac{dv}{dt} = -mg + kv^n \tag{3.1}$$

where v is the vertical velocity and g is the acceleration of the Earth's surface gravity. From equation, k in the resistance term is a constant, n is usually given a value of 1 or 2. So when the final velocity is reached, we obtain $\frac{dv}{dt} = 0$ and $k = \frac{mg}{v_t^n}$

In this article, we will take the value of n as 2 because it is a recognized value that is more suitable for larger and high-speed objects. When $n = 2$, the resistance term is usually given as follows:

$$F_r = \frac{1}{2}\rho_0 ADv^2 \tag{3.2}$$

where $\rho_0 = 1.23\ kg/m^3$ is the density of air at sea level, A is the area of the falling object perpendicular to the direction of velocity, D is the dimensionless drag coefficient, the value of which is determined by the shape of the object and the surrounding air flow, and the relevant data are determined by the relevant professional data website [2] gives, so we find:

$$v_t = \sqrt{\frac{2mg}{AD\rho_0}} \tag{3.3}$$

Using the definition of Eqs. (3.2) and (3.3) in Eq. (3.1) gives the solutions:

$$\begin{cases} v = -v_t \tanh\left(\dfrac{gt}{v_t} + c_1\right) \\ y = -\dfrac{{v_t}^2}{g}\ln\left[\cosh\left(\dfrac{gt}{v_t} + c_1\right)\right] + c_2 \end{cases} \tag{3.4}$$

where c_1 and c_2 are the selected integration constants that are applied to give the correct initial velocity and altitude, respectively. At the limit $k \to 0$, Eq. (3.4) returns to a simple free-fall solutions:

$$\begin{cases} v = v_0 - gt \\ y = y_0 + v_0 t - \dfrac{1}{2}gt^2 \end{cases} \tag{3.5}$$

where v_0 and y_0 are the initial velocity and altitude[3].

4.2.2 High Altitude Skydiving

i) Mechanical Analysis

In the extreme skydiver problem, the change in air density with altitude can be simulated by adding a height-dependent term to the resistance. The simplest atmospheric model is an Isothermal or Laplace

Atmosphere. This shows that the air pressure P decreases with height z as a single exponential function $P=P_0\cdot e^{-\frac{Mgz}{kT}}$, where $P_0=101300\ Pa$ and M is the mean molar mass of the air and k is the Boltzmann constant, from which the air density is obtained as $\rho=\rho_0\cdot e^{-\frac{Mgz}{kT}}$. Replacing it with ρ_0 in Eq. (3.2) gives the resistance as:

$$F_{universal}=\frac{GM_E m}{(R+z)^2} \tag{3.6}$$

where G is the gravitational constant, M_E is the mass of the earth, R is the radius of the earth, m is the mass of the skydiver, z is the height of the skydiver, and we define the equivalent gravitational acceleration as

$$g_{equal}=\frac{GM}{(R+z)^2} \tag{3.7}$$

So the Eq. (3.1) is transferred into:

$$m\frac{dv}{dt}=-mg_{equal}+\frac{1}{2}\rho_0 ADv^2\cdot e^{-\frac{Mgz}{kT}} \tag{3.8}$$

Although Eq. (3.8) assumes that the atmosphere is isothermal, which, strictly speaking, is not the case. But the coefficient $\frac{Mg}{RT}$ can be fitted into the experimental data from NASA [4]. The data table is in the appendix A. So Eq. (3.8) is a good description of the change in air pressure with altitude. Scatters in Figure 4.1 shows the change in air pressure in the standard atmosphere as a function of altitude. To a good approximation, this change can be described as an exponential function. The exponential least square fitting results in the coefficient $\frac{Mg}{RT}$ in equation (3.8) with a value of $1.3401\times10^{-4}\ m^{-1}$. The mean molecular mass of the air is 0.0288 kg/mol, which gives an effective temperature of about 254K in the atmosphere. The results are shown in Figure 4.1. In this paper, the inhomogeneous atmosphere described in Eq. (3.8) is called the Laplace atmosphere, and the coefficient $\frac{Mg}{RT}$ is called the characteristic height λ of the atmosphere, with a value of $7.4621\times10^{3}\ m$.

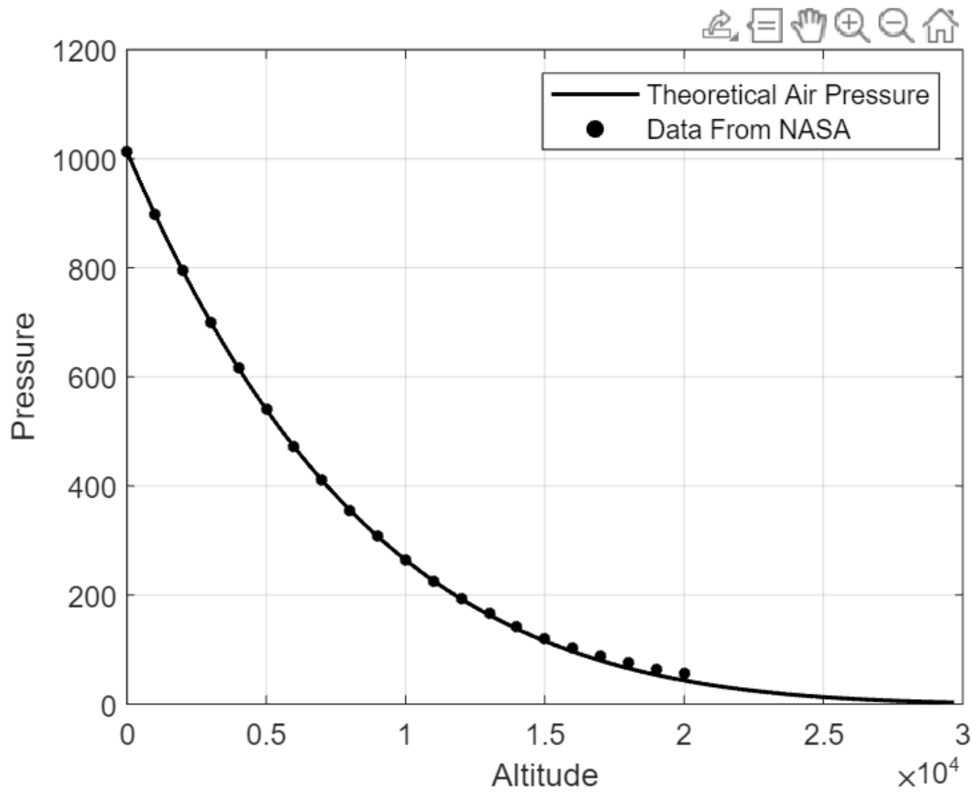


Figure 4.1

Thus we find Eq. (3.8) is transferred into[5]:

$$mv\frac{dv}{dz} = -mg_{equal} + k_0 v^2 e^{-\frac{z}{\lambda}} \tag{3.9}$$

This is an exact equation, the numerical solution of which can be obtained by Runge-Kutta method. Details are in the appendix B.

ii) Specific Velocity Altitude (SVA)

In this process, the skydiver will reach the maximum speed after the skydiving. However, the greater the speed reaches, the more difficult it is to control own posture, which is why Felix Baumgartner has a moment when he cannot control his own posture. To solve the problem, we have set up a metric H_v, which is the ratio of the skydiving altitude and the maximum speed:

$$H_v = \frac{z_0}{v_{\max}}$$

We denote it as specific velocity altitude (SVA). We should render SVA as big as possible to obtain the maximum skydiving altitude and the minimum maximum speed. Of course, this is for ordinary people who are not professionally trained, and the minimum value obtained in this case is the safest altitude for ordinary people. **This is the constraint D:**

$$\max(H_v)$$

iii) Aerodynamic Heat

For professionally trained pros, it is simple to control one's posture at high speeds, and we use the aerodynamic heat as a threshold to limit the height of the skydiving.[6] When a skydiver starts to free fall from a height, the descent is very fast and the air is rapidly compressed, causing the gas temperature to rise. This phenomenon is known as the aerothermal effect. Due to the aerothermal effect, skydivers need to wear specially designed skydiving clothing and gear to protect them from the heat. These garments often include thermal insulation to reduce the effects of temperature increases caused by aerodynamic heat on the body. But space skydiving is different from ordinary skydiving, because of its ultra-long stay time and possible supersonic process, the aerothermal effect cannot be simply ignored. The formula for the generation of aerothermal heat is[7]:

$$\frac{T_{surf}}{T_{environment}} = 1 + [(\gamma - 1)/2]v^2 \tag{3.10}$$

where T_{surf} is the temperature of the interface between the spacesuit and the atmosphere, $T_{environment}$ is the ambient temperature, $\gamma = 1.4$ is the ratio of the molar heat capacity at constant pressure to that at constant volume, v is the speed of the skydiver in Mach.

However, the actual atmosphere is divided into four layers. And the ambient temperature of each layer varies with altitude. The magnitude of the speed of sound also changes with altitude. We have to improve the Isothermal Atmospheric to the International Standard Atmospheric (ISAP)[8]. The speed of sound can be given by the following formula[9]:

$$v_{sound} = \sqrt{\frac{\gamma \cdot R \cdot T_{environment}}{M}} \tag{3.11}$$

where R is the Boltzmann constant and M is the mean molar mass of the air. The ambient temperature is given by the data of the International Standard Atmospheric. The ISAP is only applied here, because the calculation speed of the Isothermal Atmospheric is much faster than that of the ISAP, and the difference between the two is not very significant, as shown in Figure 4.2:

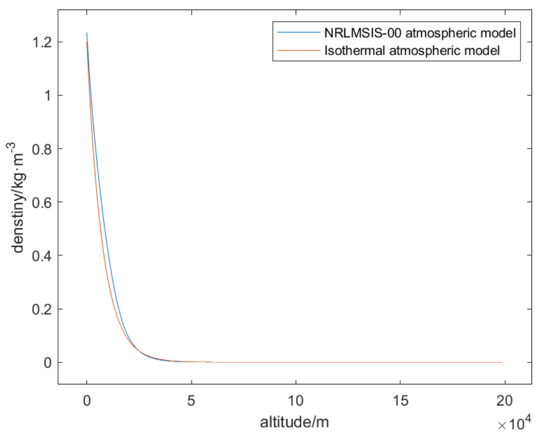


Figure 4.2

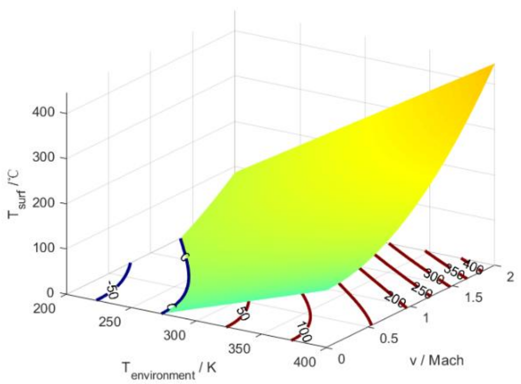


Figure 4.3

Figure 4.3 is an image of equation (3.10), where contour lines are marked on the T~v plane. We find that as the speed increased, the temperature rises dramatically, as well as the danger to skydivers. The upper limit of heat resistance for spacesuits is 150 degrees Celsius[10]. Due to the time limit, we fail to take into account the process of complete heat conduction into the spacesuit, thus take the temperature of the interface between the spacesuit and the atmosphere as the temperature of the spacesuit. The derivation process of part of the heat conduction is moved to future work. **Therefore, the constraint A is:**

$$T_{surf} \leq 150 \ ℃$$

4.3 Parachute Model

4.3.1 Parachute Opening Recoil Interaction

According to the work of Gyana R Behera, Arunangshu Mukhopadhyay and Monica Sikka[11], we assume that the parachute is C-9 parachute, which is a hemispherical parachute with a diameter of 5.95 meters. After opening the parachute, the maximum recoil force under the C-9 parachute is given by:

$$F_{recoil} = (144.17 \times 10^{-3}) m^{0.85} {v_s}^{1.87} \tag{3.12}$$

where m is the mass of the skydiver, v_s is the speed at the moment of opening the parachute.

At the same time, according to the work of Burton RR, Michaelson ED, and Leverett SD Jr[12], we know that the maximum acceleration that humans can tolerate is 5~15g. As for the problem discussed in this article, we take 15g, so as to ensure that the skydiving altitude is as large as possible. **This is the constraint B:**

$$a_{recoil} \leq 15g$$

4.3.2 Descent with Parachute and Landing

In the Laplace Atmosphere, the falling process of the parachute after opening is consistent with the force model of the high-altitude skydive:

$$m\frac{dv}{dt} = -mg_{equal} + F_r \tag{3.13}$$

However, since the area A perpendicular to the direction of velocity increases rapidly after the parachute is opened, it can be seen from Eq. (3.2) that F_r increases rapidly, so that the speed of the parachutist decreases.

In order for a skydiver to land without injury, the following must be satisfied[13]:

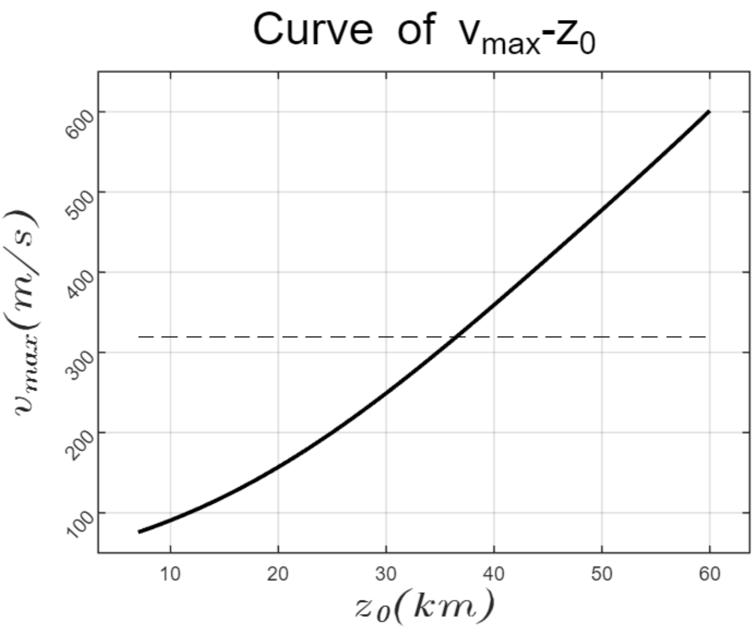
$$v_{ground} \leqslant 7m/s^2$$

This is constraint C.

5. Model Solution

5.1.1 Maximum Altitude at Low Speed

Based on the model of High Altitude Skydiving presented in 4.2.2, we can draw the curve of skydiving altitude and maximum velocity during free fall:

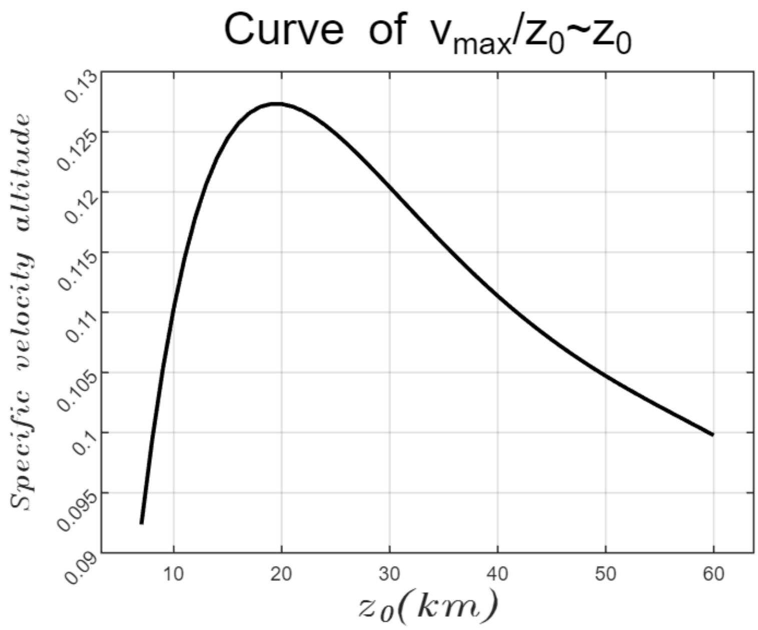


In the figure, the dashed lines indicate the estimated speed of sound. It is evident that there exists an approximately linear relationship between the maximum speed and the height of the jump. When the takeoff height exceeds approximately 35km, the maximum speed surpasses the speed of sound, thus posing challenges to attitude control.

To optimize the skydiving altitude and effectively manage the maximum speed, we propose a concept known as "specific speed height." This term is defined as follows:

$$H_v = z_0 / v_{\max}$$

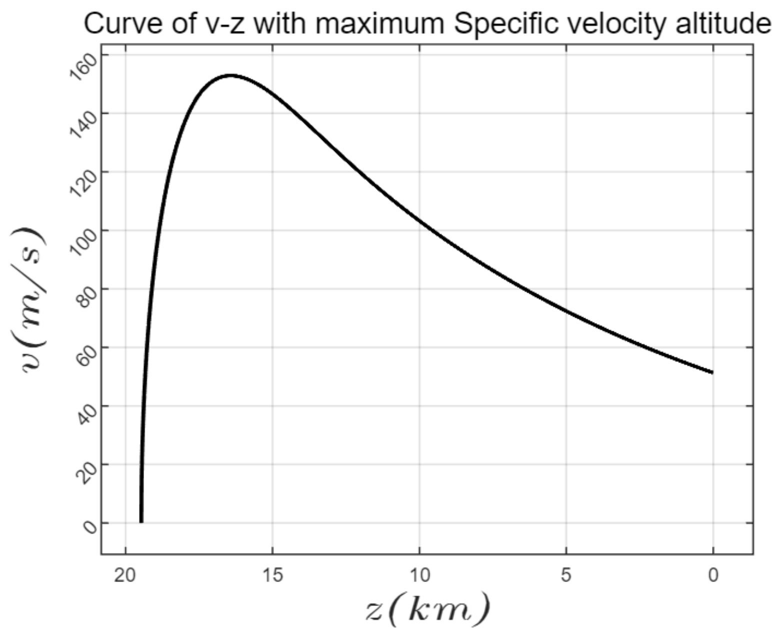
The relationship between the specific speed height and the skydiving altitude can be made as follows:



The figure exhibits an extreme value that represents the maximum specific velocity height. The corresponding jump height refers to the highest attainable altitude while maintaining the lowest possible maximum speed.

$$z_{0\max} = 19.467\ km$$

The correlation between the speed and height of the jump at the optimal specific speed height is as follows:



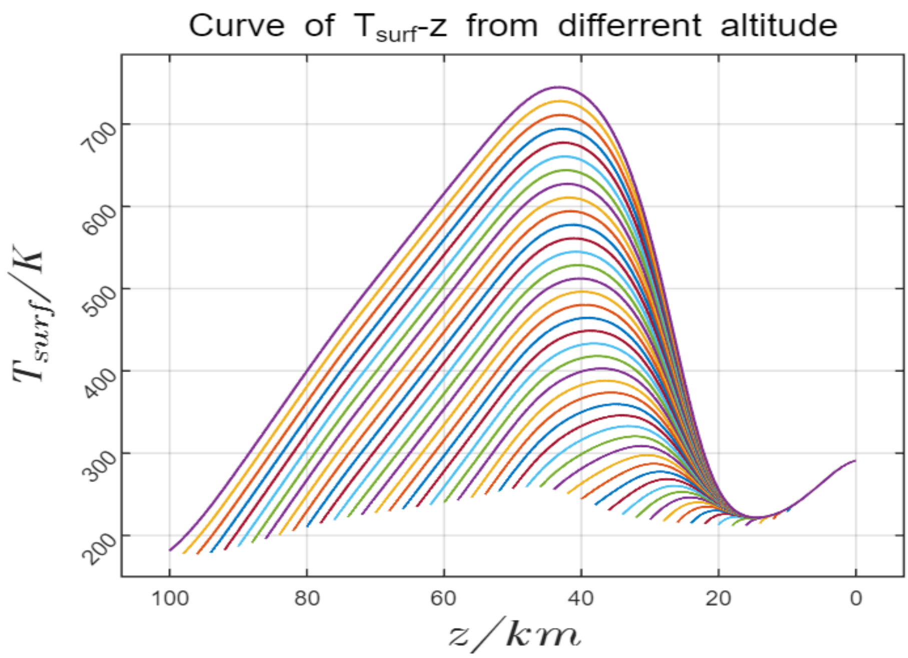
The maximum speed is recorded at 152m/s, roughly equivalent to 0.5 Mach, which falls significantly below the range of transonic or supersonic velocities. Releasing the rocket at this altitude ensures absolute safety for the skydiver, as it eliminates the risk of encountering severe instabilities or complications arising from unforeseen factors.

5.1.2 Maximum Altitude for Supersonic Flight

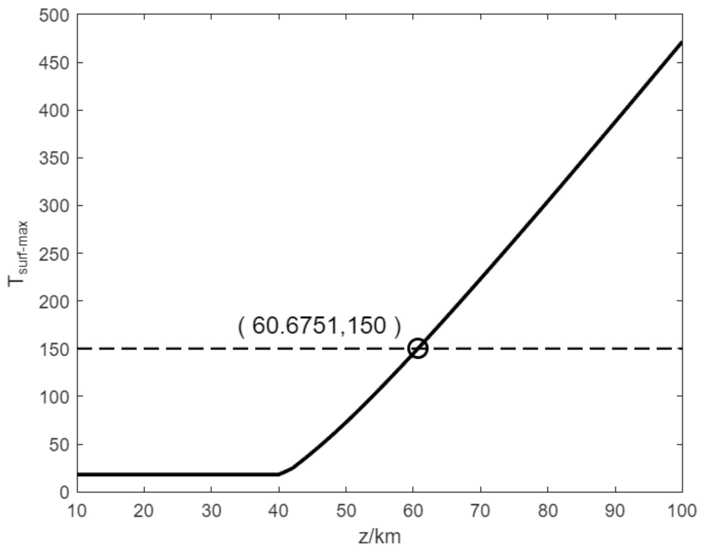
To challenge the limits of human free-fall, the skydiver must fly at supersonic speeds. Felix Baumgartner et al. have proven the feasibility of supersonic free fall. But aerodynamic heat at supersonic speeds would be a major danger.

According to the thesis provided by the paper[7], the relationship between stagnation temperature, ambient temperature and flight speed can be obtained as Figure 4.3.

Using NRLMSIS-00 atmospheric model and considering this problem within Karman line ($100km$), the relationship between stagnation temperature and altitude from different skydiving altitudes can be obtained:



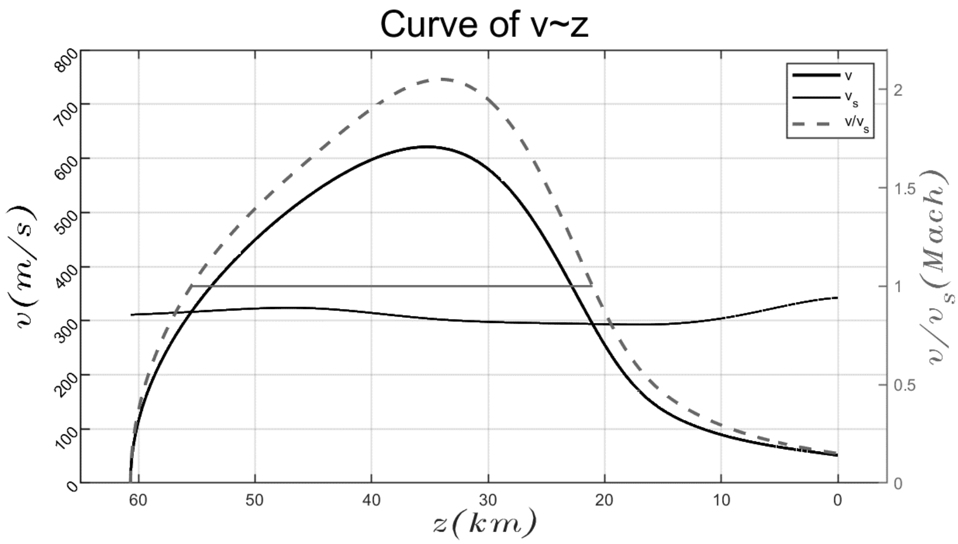
In the low-altitude part, all curves converge together, independent of the skydiving altitude. However, as the skydiving altitude increases, the maximum value of the stagnation temperature also increases rapidly. The relationship between maximum stagnation temperature and skydiving altitude can be obtained as follows:



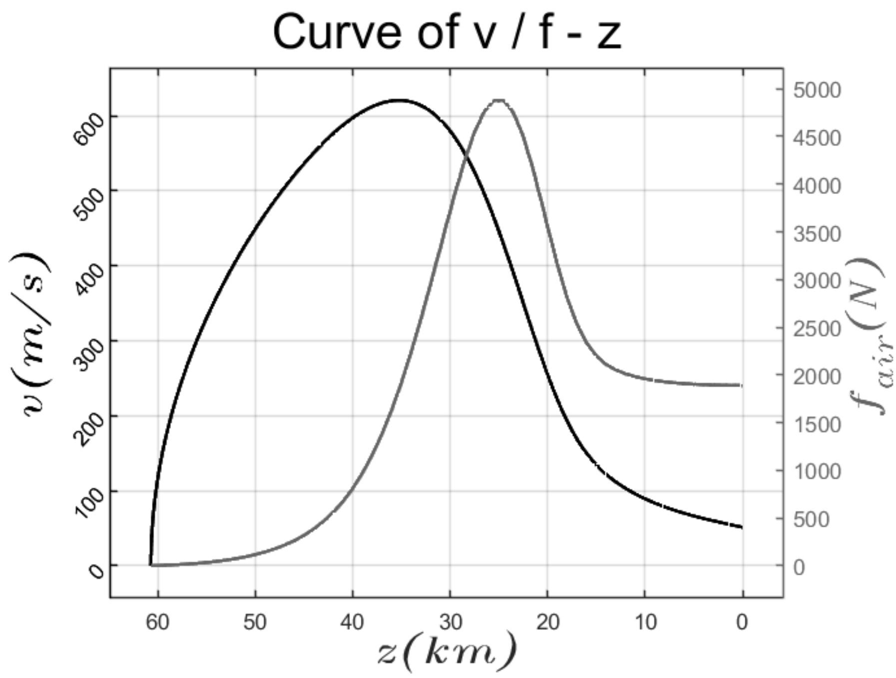
The dotted line marks the heat resistance limit $(150°C)$ of the spacesuit, so the intersection of the two lines is the maximum jump height when the stagnation temperature reaches the limit.

$$z_{0\max} = 60.6751\ km$$

If the skydiver leaves the rocket at this altitude, its speed changes as shown in the figure below:



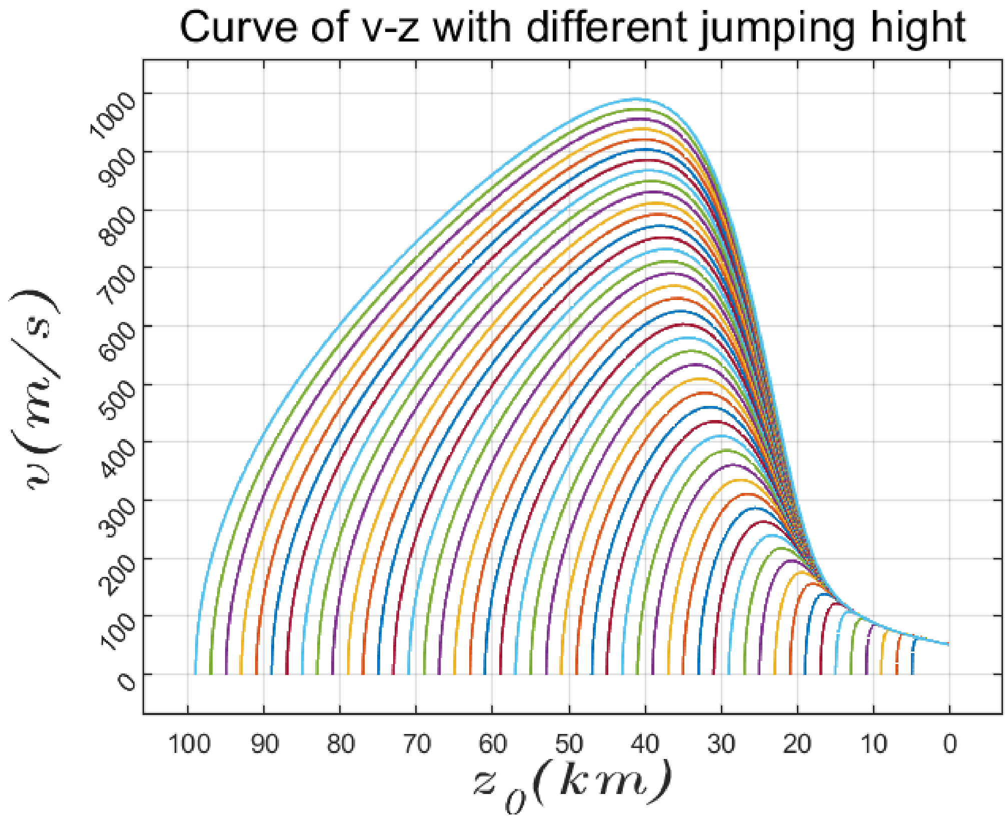
In this case, the maximum speed of the skydiver was as high as $2.05 Mach$, and more than half of the journey was supersonic. The change of air resistance is shown in the figure below:



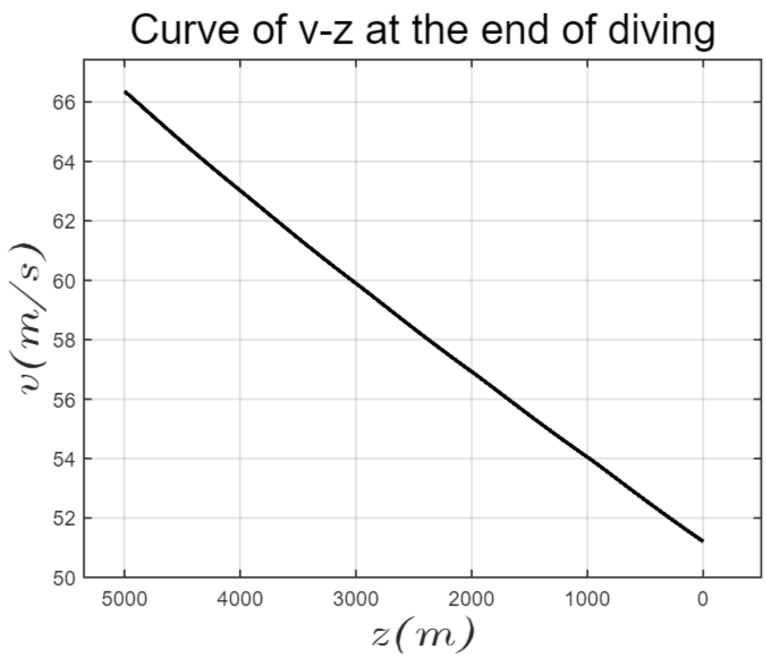
The peak of speed does not occur at the same height as the peak of air resistance because the lower atmosphere is denser. In addition, the peak of air resistance is high enough to produce a very destructive tear, which is $4.879\ kN$.

5.1.3 Opening Window of the Parachute

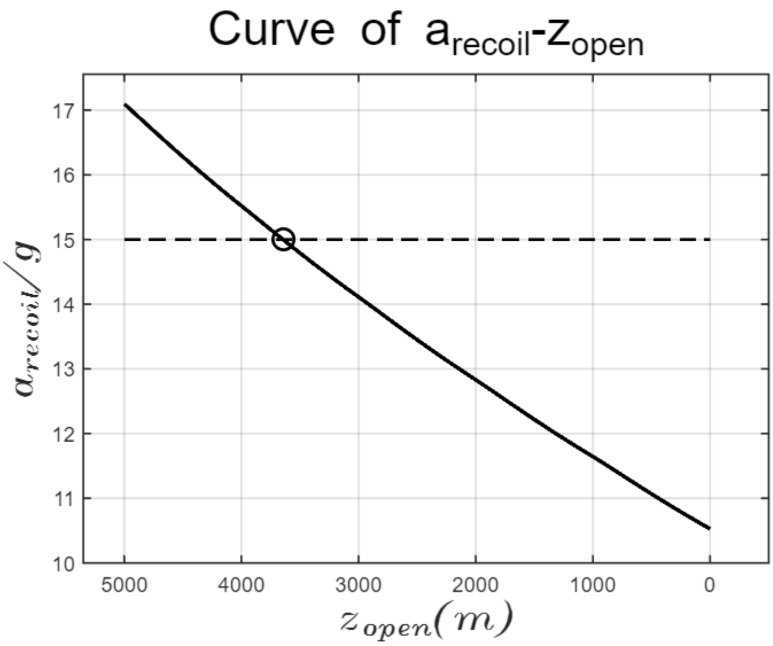
Through calculation, the velocity distribution line of different skydiving altitudes can be obtained as follows:



It is easy to find that the ends of all curves converge to the same curve, indicating that the velocity change of the end of the free fall is independent of the skydiving altitude. Using the curve of $z_0 = 10\ km$ as an approximation, the terminal velocity distribution can be obtained as follows:



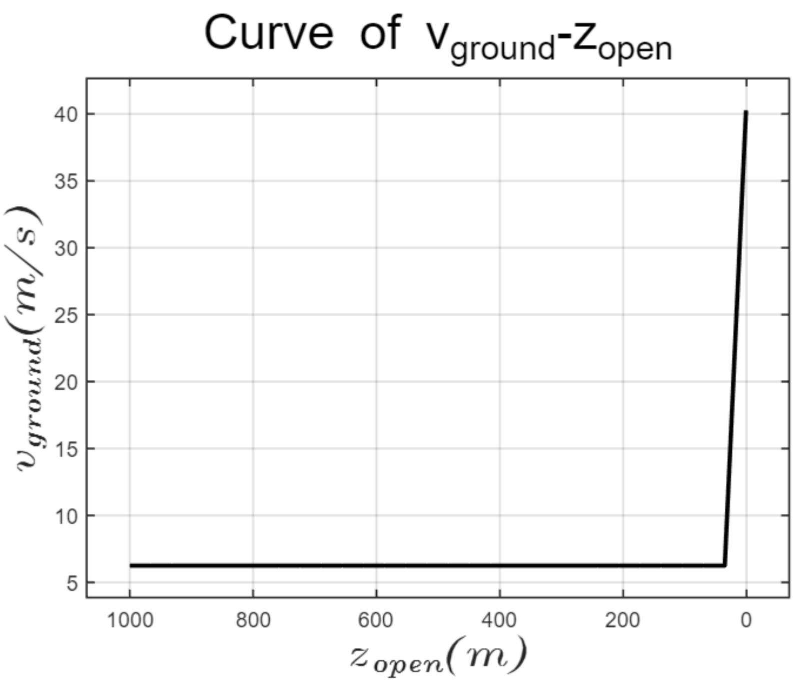
According to the calculation method in Section 4.3.1, the relationship between recoil force and parachute opening height can be seen as follows:



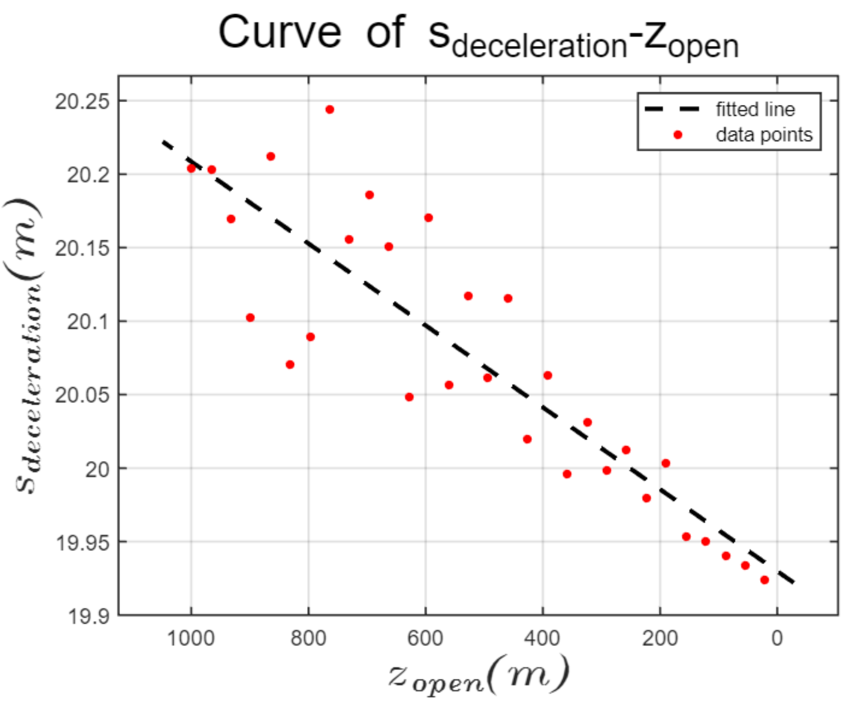
Due to the limitation $a_{recoil} < 15g$, the earliest opening height can be obtained:

$$z_{open-earliest} = 3.6456\ km$$

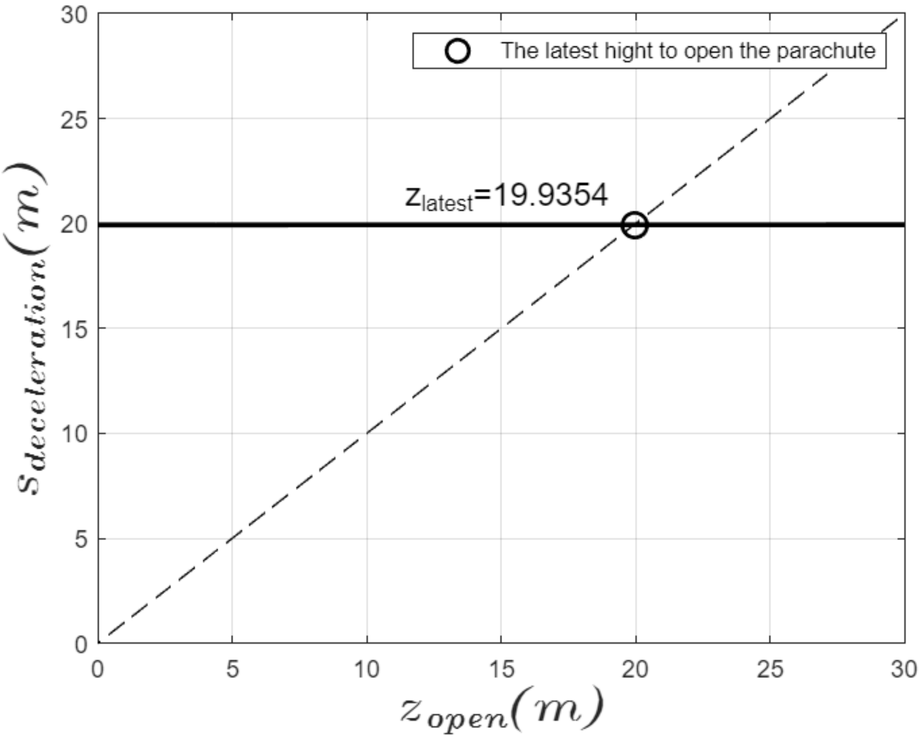
If the opening time is delayed, the landing speed may exceed the limit, as shown in the figure below:



When the height is sufficient, the landing speed is stable until the opening height cannot meet the distance required for deceleration. If the time velocity is considered to be stable when $dv/dz < 0.005$, the relationship between the opening height and deceleration distance is approximately linear:



The fitting curve obtained from the figure above can estimate the critical case when the height opening parachute is exactly equal to the deceleration distance.



The intersection of the dotted line and the solid line in the figure above is the latest altitude to open the parachute:

$$z_{open-latest} = 19.9354\ m$$

The corresponding landing speed:

$$v_{ground} = 6.2587\ m/s$$

The acceleration when opening the parachute:

$$a_{recoil} = 10.547g$$

The later the opening of the umbrella, the smaller the recoil force will be, so it can be considered that the best opening height is:

$$z_{open} = 19.9354\ m$$

6. Model Evaluation

6.1 Advantages of the model

6.1.1 Simplicity

The model is designed with simplicity in mind, which can make it easier to understand and apply. This is beneficial for users who may not have advanced knowledge in the field.

6.1.2 Authenticity

The model appears to incorporate authentic conditions or parameters, which suggests that it aims to closely replicate real-world scenarios. And several SCI papers are referred which ensures that the model is grounded in well-established scientific research. This enhances the model's reliability and applicability to practical situations.

6.1.3 Innovation

Using the fourth-order Runge-Kutta method (RK4), the numerical solution between the variables is obtained. And reasonably set up a metric, specific velocity altitude (SVA), to measure the maximum skydiving altitude. Apply ISAP to calculate T_{surf} on account of

6.2 Disadvantages

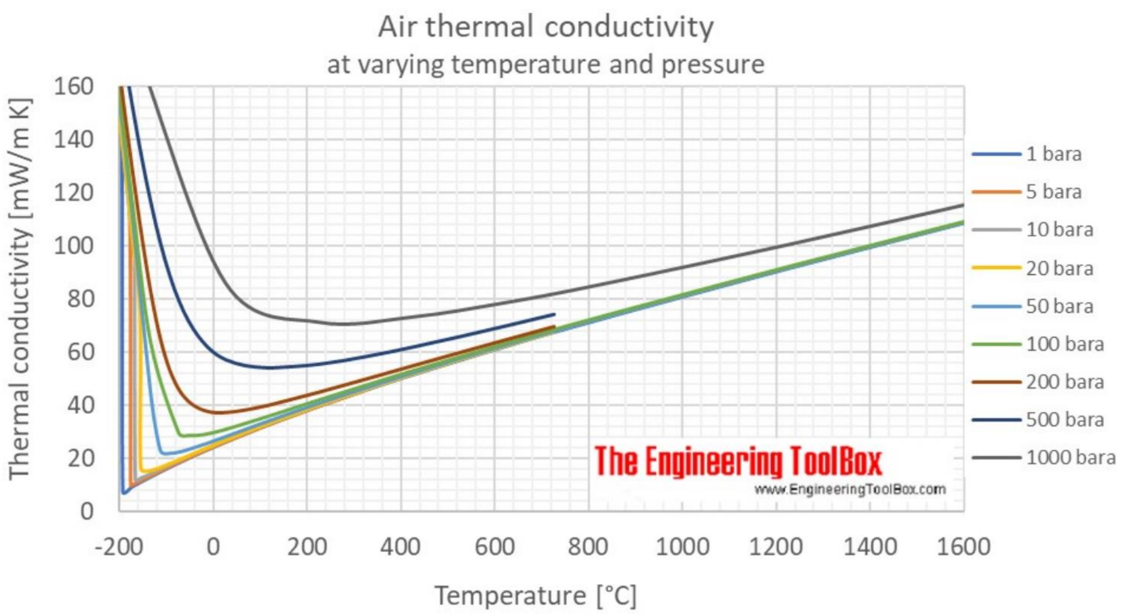
The model assumes that the air density is constant during low-altitude skydiving, which simplifies the differential equations. However, in reality, air density can vary due to weather conditions, altitude, and other environmental factors. This assumption could lead to inaccuracies in the model's predictions for scenarios where air density fluctuates. Besides, we ignore the influence of rotation on body posture control at high speed, which is a content worth considering.

7. Future Work

Newton's law of cooling is the study of the gradual cooling of an object whose temperature is higher than its surroundings by transferring heat to the surrounding medium. Follow the rules, when there is a temperature difference between the medium surface and the environment, the heat lost per unit time per unit area is proportional to the temperature, and this proportional coefficient is called the heat transfer coefficient:

$$\frac{\partial Q}{\partial t} = k_0 (T - T_{environment}) \tag{5.1}$$

where k_0 is related to the medium surface temperature, surface finish, surface area and ambient temperature. k_0 is called the thermal conductivity. Figure 7.1 gives the relationship between k_0 and temperature as well as pressure[14].



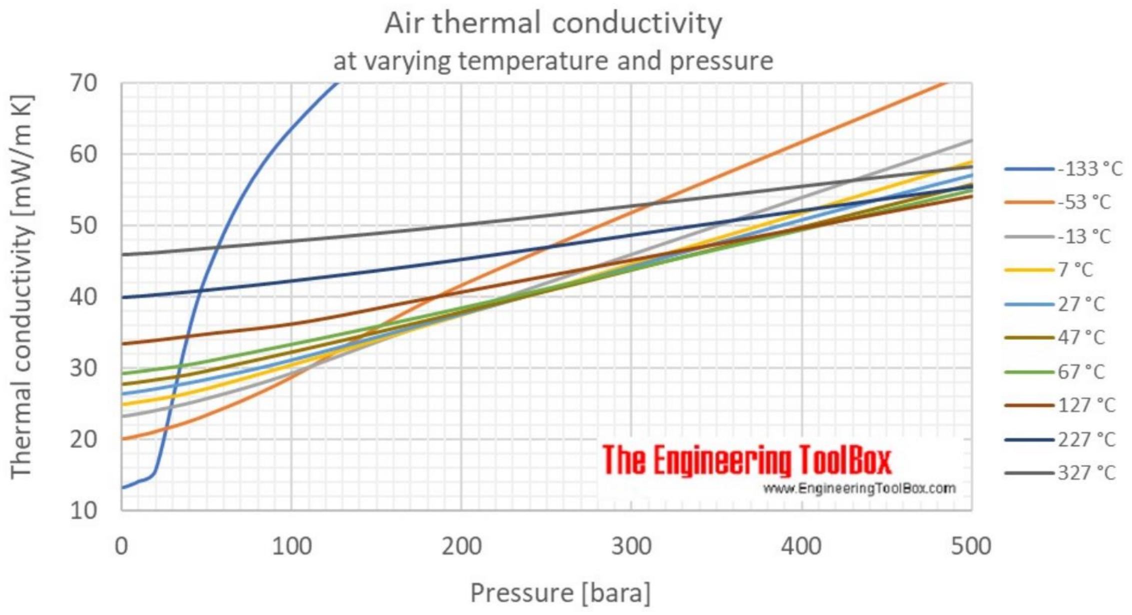


Figure 7-1

The temperature and pressure changes with altitude are substituted into k_0, and the heat absorption Q per unit time of the spacesuit is obtained by combining Eq.(5.1). Therefore, calculate the temperature change of the spacesuit by $Q = cm\Delta T$. Then, the theoretical model of the aerothermal part is improved. Due to time limit, further research is needed on the theory and data.

8. Reference

[1] Colino, José M;Barbero, Antonio J;Tapiador, Francisco J.Dynamics of a skydiver's epic free fall.[J].PHYSICS TODAY,2014,Vol.67(4): 64-65

[2] Engineering ToolBox, "Drag Coefficient, "http://www.engineeringtoolbox.com.libyc.nudt.edu.cn:80/drag-coefficient-d_627.html.

[3] Corvo, T.An Analytical Solution to the Extreme Skydiver Problem[J].Physics Teacher,2019,Vol.57(5): 287-289

[4] https://ntrs.nasa.gov/citations/19930091285
[5] Mohazzabi, Pirooz;Shea, James H..HIGH-ALTITUDE FREE FALL[J].American Journal of Physics,1996,Vol.64(10): 1242-1246
[6] Jared Leidich;Daniel Blignaut;Robert A. Eustace.Human Stratospheric Descent System: Architecture and Design for Safe, Stable Flight[A].AIAA aerodynamic decelerator systems technology conference and seminar[C],2015
[7] Gupta, Paresh;Rajput, S. P. S..Computational Fluid Dynamics and Thermal Analysis to Estimate the Skin Temperature of Cockpit Surface in Various Flight Profiles[J].Journal of Aerospace Engineering,2015,Vol.28(1): -1
[8] https://ccmc.gsfc.nasa.gov/models/NRLMSIS~00/
[9] José M Colino;Antonio J Barbero.Quantitative model of record stratospheric freefall[J].EUROPEAN JOURNAL OF PHYSICS,2013,Vol.34(4): 841-848
[10] Bue, Grant C.;Trevino, Luis A.;Tsioulos, Gus;Settles, Joseph;Colunga, Aaron;Vogel, Matthew;Vonau, Walt.Hollow Fiber Spacesuit Water Membrane Evaporator Development and Testing for Advanced Spacesuits[A].40th International Conference on Environmental Systems[C],2010
[11] Behera, Gyana R.;Mukhopadhyay, Arunangshu;Sikka, Monica Puri.Development of scheme to evaluate the performance of parachute canopy fabrics under tensile impact[J].Journal of Industrial Textiles,2022,Vol.51(2): 1
[12] Burton, R R;Leverett, S D;Michaelson, E D.Man at high sustained +Gz acceleration: a review.[J].Aerosp Med,1974,Vol.45(10): 1115-1136

[13] Wu Jinxue. Analysis on parachuting landing Acceleration [J]. Physics Teachers,2018,(2): 63-67

[14] https://www.engineeringtoolbox.com/air-properties-viscosity-conductivity-heat-capacity-d_1509.html

9. Appendix

A) The Standard Atmosphere Table

$Altitude\,(km)$	$\frac{p}{p_0}$	$Altitude\,(km)$	$\frac{p}{p_0}$
1	0.8870	11	0.2232
2	0.7845	12	0.1909
3	0.6918	13	0.1632
4	0.6082	14	0.1395
5	0.5330	15	0.1193
6	0.4655	16	0.1020
7	0.4051	17	0.0872
8	0.3512	18	0.07454
9	0.3032	19	0.06372
10	0.2606	20	0.05449

B) Runge-Kutta method

Runge-Kutta method is a set of iterative methods for solving numerical solutions to ordinary differential equations. The method can be seen as a generalization of Euler's method, providing greater accuracy and stability. The Runge-Kuta method improves the estimation of the slope of the function by calculating multiple intermediate points in each step, thereby improving the prediction accuracy of future points. The most commonly used Runge-Kuta method is the fourth-order Runge-Kuta method, and the steps are as follows:

Algorithm: Runge-Kutta
Input: $f(y,t)=0,\ y(t_0)$
Output: $y=y(t), 0\le t\le T$
While $t<T$:
1. $k_1=f(y_n,t)$;
2. $k_2=f\left(y_n+\Delta t\dfrac{k_1}{2},t+\dfrac{\Delta t}{2}\right)$;
3. $k_3=f\left(y_n+\Delta t\dfrac{k_2}{2},t_n+\dfrac{\Delta t}{2}\right)$
4. $k_4=f(y_n+\Delta tk_3,t+\Delta t)$
5. $t=t+\Delta t$
End
Return:y

The mathematical expression of this method is:

$$y_{n+1}=y_n+\frac{1}{6}h(k_1+2k_2+2k_3+k_4) \tag{5.2}$$

where Δt is the step size, y_n is the approximate solution of the current step, and y_{n+1} is the approximate solution of the next step.

C) Code in Matlab

```
% UPC
% A code for UPC. This is a mlx code. It can show the results calculated by the code lively.
% codline1 = 1

clear
close all
% Switch to NRLMSIS-00 atmospheric model
air = load("air.mat").air
air.density = air.density*1e3;air.altitude = air.altitude*1e3;

figure,plot(air.altitude,air.temperature)
gama = 1.4; R = 8.314; MA = 28.8e-3;
figure,plot(air.altitude,sqrt(gama*R*air.temperature/(MA)))
figure,plot(air.altitude,air.density)
hold on,plot(air.altitude,1.2*exp(-air.altitude/7.4621e3))
axis padded,legend('NRLMSIS-00 atmospheric model','Isothermal atmospheric model')


g = 9.8;
namda = 7.4621e3;
k0 = 0.72;
m = 190;
% Felix Baumgartner departure height: z0 = 38969.4;
% z0 = 60.6751e3;
z0 = 38969.4;
vztvf = freefall(z0);
v = vztvf(:,1); z = vztvf(:,2);vs = vztvf(:,4);

Fair = vztvf(:,5);
max_Fair = max(Fair)
% Felix Baumgartner's space-time curve and stress curve
fig = figure;
lcolor = [0,0,0]; rcolor = [0.4,0.4,0.4];
set(fig,'defaultAxesColorOrder',[lcolor; rcolor]);
yyaxis left;
plot(z/1000,v,'Color',lcolor,'LineWidth',2) ,hold on
plot(z/1000,vs,'-k','LineWidth',1.3), axis padded, set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z(km)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylabel('v(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v','~','z'],'FontSize',23)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)%,set(ylab,'Rotation',0),box on
ylim([0,600])

[xm,ym] = polyxpoly(z/1000,v./vs,[z(1),z(end)]/1000,[1,1]);
yyaxis right;
plot(z/1000,v./vs,'--','Color',rcolor,'LineWidth',2),hold on
plot(xm,ym,'-','Color',rcolor,'LineWidth',1.3),axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9)
ylabel('v/v_s(Mach)','FontName','ItalicT','FontWeight','bold','FontSize',16), grid on
legend('v','v_{s}','v/v_s'),ylim([0,1.3])
max(v./vs)

event1 = [38969,0;33446,310;27833,377.1;22961,290;7619,79;2567,53];
figure
% hold on
lcolor = [0,0,0]; rcolor = [0.4,0.4,0.4];
plot(z/1000,v,'Color',lcolor,'LineWidth',2) ,hold on
```

```
plot(event1(:,1)/1000,event1(:,2),'*k','MarkerSize',10)
plot(z/1000,vs,'-k','LineWidth',1.3), axis padded, set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z(km)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylabel('v(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v','~','z'],'FontSize',23),legend('theoratical line','real data','v_{s}','Location','best')
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)%,set(ylab,'Rotation',0),box on
% ylim([0,600])

fig = figure;
% lcolor = [0,0,0]; rcolor = [0.4,0.4,0.4];
set(fig,'defaultAxesColorOrder',[lcolor; rcolor]);
yyaxis left;
plot(z/1000,v,'Color',lcolor,'LineWidth',2) , axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z(km)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylabel('v(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v / f',' - ','z'],'FontSize',23)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)%,set(ylab,'Rotation',0),box on

yyaxis right;
plot(z/1000,Fair,'Color',rcolor,'LineWidth',2),axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9)
ylabel('f_{air}(N)','FontName','ItalicT','FontWeight','bold','FontSize',16)

t = vztvf(:,3);

figure
plot(t/60,v,'Color',[0,0,0],'LineWidth',2), axis padded, grid on , set(gca,'LineWidth',0.9)
xlabel('t(min)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylab = ylabel('v(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v','-','t'],'FontSize',23)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)

figure
plot(t/60,z/1000,'Color',[0,0,0],'LineWidth',2), axis padded, grid on , set(gca,'LineWidth',0.9)
xlabel('t(min)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylab = ylabel('z(km)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','z','-','t'],'FontSize',23)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)

% Calculate the relationship between maximum speed and takeoff height

z0 = 7e3:1.5e3:100e3;
vmax = zeros(length(z0),1);
vsmax = zeros(length(z0),1);
% Tmax = zeros(length(z0),1);
for i = 1:length(z0)
vzi = freefall2(z0(i));
vtemp = vzi(:,1); ztemp = vzi(:,2);
vmax(i) = max(vtemp);
zmax = ztemp(vtemp==max(vtemp));
Tmax = interp1(air.altitude,air.temperature,zmax);
vsmax(i) = sqrt(gama*R*Tmax/(MA));
end

% the relationship between maximum speed and takeoff height
figure
hold on
plot(z0'/1000,vmax,'Color',[0,0,0],'LineWidth',2)
% plot([z0(1),z0(end)]/1000,[319.29,319.29],'--k')
```

```
plot(z0'/1000,vsmax,'--k')
axis padded, grid on , box on , set(gca,'LineWidth',0.9)
xlabel('z_0(km)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylab = ylabel('v_{max}(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v_{max}','-','z_0'],'FontSize',23)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)
legend('v_{max}','v_{sound}','Location','northwest')
% Relation between specific speed height and skydiving altitude
figure
plot(z0'/1000,(z0'/1000)./vmax,'Color',[0,0,0],'LineWidth',2)
% plot([z0(1),z0(end)]/1000,[319.29,319.29],'--k')
axis padded, grid on , box on , set(gca,'LineWidth',0.9)
xlabel('z_0(km)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylabel('Specific velocity altitude','FontName','ItalicT','FontWeight','bold','FontSize',12);
title(['Curve of ','v_{max}/z_0','~','z_0'],'FontSize',23)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)
vzratio0 = (z0'/1000)./vmax;
zinterp = linspace(z0(1),z0(end),1000);
vzratio = interp1(z0,vzratio0,zinterp,'spline');

z0max = zinterp(vzratio == max(vzratio))
vz = freefall2(z0max);
v = vz(:,1); z = vz(:,2);
vmax_safe = max(v)
% Velocity height curve after takeoff at Maximum specific speed altitude
figure
plot(z/1000,v,'Color',[0,0,0],'LineWidth',2) , axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z(km)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylab = ylabel('v(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v','-','z with maximum Specific velocity altitude'],'FontSize',15)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)

% Calculate the distribution of pneumatic heating and the damage to the spacesuit
% T~[200,800] Mair~[0,2]
tscale = linspace(200,400);
mair = linspace(0,2);
[Ten,Mair] = meshgrid(tscale,mair);
Tair = (1+((1.4-1)/2)*Mair.^2).*Ten-273.15;
figure,surfl(Ten,Mair,Tair),hold on
contour(Ten,Mair,Tair,'ShowText','on','LineWidth',2.3)
view([393 26]),colormap jet,shading flat
zlim([0,max(Tair,[],'all')])
xlabel('T_{environment} / K'),ylabel('v / Mach'),zlabel('T_{surf} / ℃')


% Calculate the relationship between the maximum interface temperature and the takeoff height

z0 = 10e3:2e3:100e3;
Tmax = zeros(length(z0),1);
% vsmax = zeros(length(z0),1);
% Tmax = zeros(length(z0),1);
figure
hold on
for i = 1:length(z0)
vzi = freefall2(z0(i));
vtemp = vzi(:,1); ztemp = vzi(:,2);
Ttemp = interp1(air.altitude,air.temperature,ztemp);
vstemp = sqrt(gama*R*Ttemp/(MA));
```

```
vMach = vtemp./vstemp;
Tsurf = (1+((1.4-1)/2)*vMach.^2).*Ttemp;
Tmax(i) = max(Tsurf);
plot(ztemp/1000,Tsurf,'LineWidth',1.2)
end
axis padded, grid on , box on,set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z/km','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylab = ylabel('T_{surf}/K','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','T_{surf}','-','z from differrent altitude'],'FontSize',15)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)

tcritical = 150;
[zmax,tc] = polyxpoly([z0(1)/1000,z0(end)/1000],[tcritical,tcritical],z0/1000,Tmax-273.15);
figure,plot(z0'/1000,Tmax-273.15,'Color',[0,0,0],'LineWidth',2),hold on
plot([z0(1),z0(end)]/1000,[tcritical,tcritical],'--k','LineWidth',1.2),plot(zmax,tc,'ok','MarkerSize',10,'LineWidth',1.5)
xlabel('z/km'),ylabel('T_{surf-max}')
text(zmax-2.5,tc+25,['( ',num2str(zmax),',',num2str(tc),' )'],'HorizontalAlignment','right','FontSize',13)

% figure,plot(ztemp/1000,vtemp,'Color',[0,0,0],'LineWidth',2)
% figure,plot(ztemp/1000,vstemp,'Color',[0,0,0],'LineWidth',2)
% figure,plot(ztemp/1000,vMach,'Color',[0,0,0],'LineWidth',2)
% figure,plot(ztemp/1000,Ttemp,'Color',[0,0,0],'LineWidth',2)
% figure,plot(ztemp/1000,Tsurf,'Color',[0,0,0],'LineWidth',2)
```

```
% Velocity and height curves for different take-off heights
z0 = 5e3:2e3:100e3;
% vmax = zeros(length(z0),1);
figure
hold on%,set(gca,'XDir','reverse','LineWidth',0.9)
for i = 1:length(z0)
vzi = freefall2(z0(i));
plot(vzi(:,2)/1000,vzi(:,1),'LineWidth',1.1)
% vmax(i) = max(vzi(:,1));
end
axis padded, grid on , box on,set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z_0(km)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylabel('v(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v','-','z with different jumping hight'],'FontSize',17)
axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)

% The v-z diagram converges to the envelope at the end, so a line with smaller z0 can be used at z&lt; 100 cases to approximate all release heights at the end of the velocity distribution.
% Here the end of z0=10km is taken as an approximation.
% That is, the height of the jump is independent of the final speed-altitude relationship, as long as the last leg's speed-altitude relationship converges to the same regardless of the height of the jump.

% Using the end of z0=10km as an approximation, the following velocity height curve can be obtained
vzend0 = freefall2(10e3);
vend0 = vzend0(:,1); zend0 = vzend0(:,2);
hend = 5000;
vend = vend0(zend0<hend); zend = zend0(zend0<hend);
```

```
figure,plot(zend,vend,'Color',[0,0,0],'LineWidth',2)
axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z(m)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylab = ylabel('v(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v','-','z at the end of diving'],'FontSize',20)
% axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)


Fmax = 144.17e-3*m^0.85.*vend.^1.87;
amax = Fmax/m;
% Relation between parachute opening impact acceleration and parachute opening height
figure,plot(zend,amax/g,'Color',[0,0,0],'LineWidth',2)%,set(gca,'XDir','reverse','LineWidth',0.9),axis padded
axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z_{open}(m)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylab = ylabel('a_{recoil}/g','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','a_{recoil}','-','z_{open}'],'FontSize',23)
[zerliest,arecoil] = polyxpoly(zend,amax/g,[zend(1),zend(end)],[15,15])
hold on,plot([zend(1),zend(end)],[15,15],'--k','LineWidth',1.4)
plot(zerliest,arecoil,'ko','MarkerSize',10,'LineWidth',1.5)
% axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)

% The height velocity curve after opening the parachute at a certain height, the deceleration distance is about ten meters.
vzend0 = freefall2(10e3);
vend0 = vzend0(:,1); zend0 = vzend0(:,2);
hend = 1000;
vend = vend0(zend0<hend); zend = zend0(zend0<hend);
Fmax = 144.17e-3*m^0.85.*vend.^1.87;
amax = Fmax/m;

ztest = 30;
% A = 100;
A = 56.7678;
CDp = 0.7; rou0 = 1.2;
k1 = CDp*A*rou0;
vzpara = freefall2(ztest,interp1(zend,vend,ztest),k1);
% vzpara = freefall(ztest,0,k1);
figure,plot(vzpara(:,2),vzpara(:,1),'Color',[0,0,0],'LineWidth',2)%,axis padded,set(gca,'XDir','reverse','LineWidth',0.9)
axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z(m)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylab = ylabel('v(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v','-','z after opening the parachute'],'FontSize',19)
% axis padded,set(gca,'LineWidth',0.8,'YTickLabelRotation',47)

vzpara(end,1)
% Using the C-9 parachute provided in the paper, the speed can converge in about ten meters.
% figure,plot(vzpara(1:end-1,2),diff(vzpara(:,1))./diff(vzpara(:,2)))

ztest = linspace(zend(1),zend(end-1),30);vground = zeros(length(ztest),1);
for i = 1:length(ztest)
% A = 56.7678; CDp = 0.7; rou0 = 1.2;
% k1 = CDp*A*rou0;
vzpara = freefall2(ztest(i),interp1(zend,vend,ztest(i)),k1);
```

```
vground(i) = vzpara(end,1);
end

% The landing speed is related to the opening height. When the deceleration distance is sufficient, the landing speed can be reduced to the same safety threshold.
figure,plot(ztest,vground,'Color',[0,0,0],'LineWidth',2)
% ax = gca;chart = ax.Children(1);datatip(chart,35.06,6.25);
axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9)
xlabel('z_{open}'),ylabel('v_{ground}')
xlabel('z_{open}(m)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylabel('v_{ground}(m/s)','FontName','ItalicT','FontWeight','bold','FontSize',16);
title(['Curve of ','v_{ground}','-','z_{open}'],'FontSize',23)

ztest = zend(zend>20);
ztest = linspace(ztest(1),ztest(end),30);zopen = zeros(length(ztest),1);
for i = 1:length(ztest)
% A = 56.7678; CDp = 0.7; rou0 = 1.2;
% k1 = CDp*A*rou0;
vzpara = freefall2(ztest(i),interp1(zend,vend,ztest(i)),k1); ztemp = vzpara(:,2); %vtemp = vzpara(:,1);
dvdz = diff(vzpara(:,1))./diff(vzpara(:,2)); dvdz = [dvdz;dvdz(end)];
zopeni = ztemp(dvdz<0.005); zopeni = ztest(i)-zopeni(1);
zopen(i) = zopeni;
% vground(i) = vzpara(end,1);
end

% The deceleration distance is related to the opening height
figure,hold on
% cc = polyfit(ztest,zopen,1);
[inz,inop,~] = polyfunc(ztest,zopen,1);
% inz = linspace(-500,ztest(1)+100);inop = interp1(ztest,zopen,inz);
plot(inz,inop,'--k','LineWidth',2,'DisplayName','fitted line'),plot(ztest,zopen,'r.','MarkerSize',12,'DisplayName','data points')
axis padded, grid on , set(gca,'XDir','reverse','LineWidth',0.9),box on
% xlabel('z_{open}'),ylabel('v_{ground}')
xlabel('z_{open}(m)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylabel('s_{deceleration}(m)','FontName','ItalicT','FontWeight','bold','FontSize',16);
legend,title(['Curve of ','s_{deceleration}','-','z_{open}'],'FontSize',21)

[zlat,zlaty] = polyxpoly(inz,inop,inz,inz);

% Latest parachute opening height
zlat
figure,hold on
plot(zlat,zlaty,'ko','MarkerSize',10,'LineWidth',1.5),plot(inz,inop,'-k','LineWidth',2),plot(inz,inz,'--k')
xlim([0 30]),ylim([0 30])
grid on , box on%set(gca,'XDir','reverse','LineWidth',0.9)
% xlabel('z_{open}'),ylabel('v_{ground}')
xlabel('z_{open}(m)','FontName','ItalicT','FontWeight','bold','FontSize',16)
ylabel('s_{deceleration}(m)','FontName','ItalicT','FontWeight','bold','FontSize',16);
legend('The latest hight to open the parachute')%,title(['Curve of ',s_{deceleration}','-','z_{open}'],'FontSize',23)
text(zlat-0.8,zlat+1.3,['z_{latest}=',num2str(zlat)],'HorizontalAlignment','right','FontSize',13)
vzfinal = freefall(zlat,interp1(zend,vend,zlat),k1);
vgroundfinal = vzfinal(:,1); vgroundfinal = vgroundfinal(end)
% zend,amax/g
```

```
ashock_per_g = interp1(zend,amax/g,zlat)




function vz = freefall(z0,v0,k1)
% opts = spreadsheetImportOptions("NumVariables", 3);
% opts.Sheet = "Sheet1";opts.DataRange = "A2:C201";
% opts.VariableNames = ["altitude", "density", "temperature"];opts.VariableTypes = ["double", "double", "double"];
% air = readtable("altitude_and_air.xlsx", opts, "UseExcel", false);
% clear opts
air = load("air.mat").air;
air.density = air.density*1e3; air.altitude = air.altitude*1e3;

R_E = 6371e3; %ratio od Earth m
g0 = 9.8;
m = 190;
gama = 1.4; R = 8.314; MA = 28.8e-3; rou0 = air.density(1);

if nargin == 1
k0 = 0.72;
v0 = 0;
else
k0 = k1;
end

z = nan(10000,1);z(1) = z0;
v = nan(10000,1);v(1) = -v0;
t = nan(10000,1);t(1) = 0;
vs = nan(10000,1);vs(1) = sqrt(gama*R*interp1(air.altitude,air.temperature,z0)/(MA));
fair = nan(10000,1);fair(1) = k0/rou0*interp1(air.altitude,air.density,z0)*v0^2/m;

% z = [];z(1) = z0;
% v = [];v(1) = v0;
% t = [];t(1) = 0;
% vs = [];vs(1) = sqrt(gama*R*interp1(air.altitude,air.temperature,z0)/(MA));
% fair = [];fair(1) = k0/rou0*interp1(air.altitude,air.density,z0)*v0^2/m;

i = 2;
if v0 == 0
dt = 0.05*z0/40e3;
else
dt = (z0/v0)/500;
end
while z(i-1)>0
zi = z(i-1); vi = v(i-1);
gi = g0*R_E^2/(R_E+zi)^2;
% Ti = interp1(air.altitude,air.temperature,zi);
ai = -gi + k0/rou0*interp1(air.altitude,air.density,zi)*vi^2/m;

zi = zi + vi*dt;
vi = vi+ai*dt;

z(i) = zi;
v(i) = vi;
t(i) = t(i-1)+dt;
vs(i) = sqrt(gama*R*interp1(air.altitude,air.temperature,zi)/(MA));
fair(i) = k0/rou0*interp1(air.altitude,air.density,zi)*vi^2;

% z = [z;zi];
% v = [v;vi];
% t = [t;t(i-1)+dt];
% vs = [vs;sqrt(gama*R*interp1(air.altitude,air.temperature,zi)/(MA))];
% fair = [fair;k0/rou0*interp1(air.altitude,air.density,zi)*vi^2];

i = i+1;
end

v = cutend(v);
```

```
    z = cutend(z);
    t = cutend(t);
    vs = cutend(vs);
    fair = cutend(fair);
    n =
max([length(v),length(z),length(vs),length(t),length(fa
ir)]); n = n(1);
    v = addend(v,n);
    z = addend(z,n);
    t = addend(t,n);
    vs = addend(vs,n);
    fair = addend(fair,n);
    vz = [abs(v),z,t,vs,fair];

    end



    function vz = freefall2(z0,v0,k1)
    % opts = spreadsheetImportOptions("NumVari-
ables", 3);
    % opts.Sheet = "Sheet1";opts.DataRange =
"A2:C201";
    % opts.VariableNames = ["altitude", "density",
"temperature"];opts.VariableTypes = ["double",
"double", "double"];
    % air = readtable("altitude_and_air.xlsx", opts,
"UseExcel", false);
    % clear opts
    air = load("air.mat").air;
    air.density = air.density*1e3; air.altitude = air.al-
titude*1e3;

    R_E = 6371e3;
    g0 = 9.8;
    m = 190;
    rou0 = air.density(1); %gama = 1.4; R = 8.314; MA
= 28.8e-3;

    if nargin == 1
    k0 = 0.72;
    v0 = 0;
    else
    k0 = k1;
    end

    z = nan(10000,1);z(1) = z0;
    v = nan(10000,1);v(1) = -v0;
    % t = nan(10000,1);t(1) = 0;
    % vs = nan(10000,1);vs(1) = sqrt(gama*R*in-
terp1(air.altitude,air.temperature,z0)/(MA));
    % fair = nan(10000,1);fair(1) = k0/rou0*in-
terp1(air.altitude,air.density,z0)*v0^2/m;

    % z = [];z(1) = z0;
    % v = [];v(1) = v0;
    % t = [];t(1) = 0;
    % vs = [];vs(1) = sqrt(gama*R*interp1(air.alti-
tude,air.temperature,z0)/(MA));
    % fair = [];fair(1) = k0/rou0*interp1(air.alti-
tude,air.density,z0)*v0^2/m;

    i = 2;
    if v0 == 0
    % dt = 0.05*z0/40e3;
    dt = 0.1*z0/38969.4;
    else
    dt = (z0/v0)/500;
    end
    while z(i-1)>0
    zi = z(i-1); vi = v(i-1);
    gi = g0*R_E^2/(R_E+zi)^2;
    % Ti = interp1(air.altitude,air.temperature,zi);
    ai = -gi + k0/rou0*interp1(air.altitude,air.den-
sity,zi)*vi^2/m;

    zi = zi + vi*dt;
    vi = vi+ai*dt;

    z(i) = zi;
    v(i) = vi;
    % t(i) = t(i-1)+dt;
    % vs(i) = sqrt(gama*R*interp1(air.alti-
tude,air.temperature,zi)/(MA));
```

```
    % fair(i) = k0/rou0*interp1(air.altitude,air.den-
sity,zi)*vi^2;

    % z = [z;zi];
    % v = [v;vi];
    % t = [t;t(i-1)+dt];
    % vs = [vs;sqrt(gama*R*interp1(air.alti-
tude,air.temperature,zi)/(MA))];
    % fair = [fair;k0/rou0*interp1(air.alti-
tude,air.density,zi)*vi^2];

    i = i+1;
    end

    v = cutend(v);
    z = cutend(z);
    % t = cutend(t);
    % vs = cutend(vs);
    % fair = cutend(fair);
    n = max([length(v),length(z)]); n = n(1);
    v = addend(v,n);
    z = addend(z,n);
    % t = addend(t,n);
    % vs = addend(vs,n);
    % fair = addend(fair,n);
    vz = [abs(v),z];

    end

function newarry = cutend(oldarray)
while isnan(oldarray(end))
oldarray(end) = [];
end
newarry = oldarray;
end

function b = addend(a,n)
while length(a)<n
a = [a;a(end)];
end
b =a;
end

function [xx,yy,cc]=polyfunc(x,y,n1)
cc=polyfit(x,y,n1);
d=(max(x)-min(x))*0.05;
xx=linspace(min(x)-d,max(x)+d);
yy=zeros(1,length(xx));
%ch=strcat(ch2,'=');
for i=1:length(cc)
yy=yy+cc(i).*xx.^(length(cc)-i);
    %ch=strcat(ch,num2str(cc(i),n2),ch1,'^',num2st
r(length(cc)-i),'+');
end
end
```